Application Load Balancers

Elastic Load Balancing

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Elastic Load Balancing: Application Load Balancers

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie 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 um Application Load Balancer? 1
Componentes do Application Load Balancer 1
Visão geral do Application Load Balancer 2
Benefícios da migração de um Classic Load Balancer 3
Serviços relacionados 4
Preços 5
Conceitos básicos 6
Antes de começar 6
Etapa 1: configurar seu grupo de destino 6
Etapa 2: escolher um tipo de balanceador de carga 7
Etapa 3: configurar o balanceador de carga e um receptor 8
Etapa 4: testar o balanceador de carga 9
Etapa 5: (opcional) excluir o balanceador de carga 9
Começando a usar o AWS CLI 11
Antes de começar 11
Criar um load balancer 12
Adicionar um receptor HTTPS 13
Adicionar roteamento baseado em caminho 14
Excluir o load balancer 15
Application Load Balancers 16
Sub-redes para seu balanceador de carga 17
Sub-redes de zona de disponibilidade 17
Sub-redes de zona local 18
Sub-redes de Outpost 18
Grupos de segurança do balanceador de carga 20
Estado do load balancer 20
Atributos do load balancer 21
Tipo de endereço IP 23
Pools de endereços IP IPAM 24
Conexões do balanceador de carga 25
Balanceamento de carga entre zonas 26
Nome DNS 26
Criar um balanceador de carga 27
Etapa 1: configurar um grupo de destino 6

Etapa 2: registrar destinos 29
Etapa 3: configurar um balanceador de carga e um receptor 29
Etapa 4: testar o balanceador de carga 9
Atualizar Zonas de disponibilidade 34
Atualizar grupos de segurança 35
Regras recomendadas 35
Atualizar os grupos de segurança associados 38
Atualizar o tipo de endereço IP 38
Atualizar os pools de endereços IP IPAM 39
Editar atributos do balanceador de carga 40
Tempo limite de inatividade da conexão 40
Duração da manutenção de atividade do cliente HTTP 41
Proteção contra exclusão 43
Modo de mitigação de dessincronização 44
Preservação de cabeçalho do host 46
Marcar um balanceador de carga 48
Excluir um balanceador de carga 49
Exibir o mapa de recursos 50
Componentes do mapa de recursos 51
Mudança de zona 52
Antes de começar 53
Balanceamento de carga entre zonas 54
Substituição administrativa 54
Habilitar mudança de zona 55
Iniciar uma mudança de zona 55
Atualizar uma mudança de zona 56
Cancelar uma mudança de zona 57
Reservas da LCU 57
Solicitar reserva 59
Atualizar ou encerrar a reserva 60
Monitore a reserva 60
Integrações de balanceadores de carga 61
Amazon CloudFront + AWS WAF 62
AWS Global Accelerator 63
AWS Config 63
AWS WAF 63

Receptores e regras 65
Configuração do receptor 65
Atributos do receptor 66
Ação padrão 68
Criar um receptor HTTP 69
Pré-requisitos 69
Adicionar um receptor HTTP 69
Certificados SSL 70
Certificado padrão 71
Lista de certificados 71
Renovação de certificado 72
Políticas de segurança 73
Políticas de segurança de TLS 75
Políticas de segurança do FIPS 100
Políticas compatíveis com FS 116
Criar um receptor HTTPS 122
Pré-requisitos 123
Adicionar um receptor HTTPS 123
Atualizar um receptor HTTPS 125
Substituir o certificado padrão 125
Adicionar certificados à lista de certificados 126
Remover certificados da lista de certificados 127
Atualizar a política de segurança 127
modificação do cabeçalho HTTP 128
Autenticação TLS mútua 129
Antes de começar 130
Cabeçalhos HTTP 133
Anuncie o nome do assunto da CA 134
Logs de conexão 135
Configurar o TLS mútuo 135
Compartilhar um armazenamento confiável 141
Configurar a autenticação de usuários 147
Preparação para usar um IdP compatível com OIDC 147
Preparação para usar o Amazon Cognito 148
Prepare-se para usar a Amazon CloudFront 150
Configurar a autenticação de usuários 150

Fluxo de autenticação ... 153
Verificação de assinatura e codificação de reivindicações de usuário ... 156
Timeout (Tempo limite) ... 158
Sair da autenticação ... 159
Regras do listener ... 159
Tipos de ação ... 160
Tipos de condição ... 169
Adicionar uma regra ... 176
Editar uma regra ... 178
Excluir uma regra ... 180
Cabeçalhos X-Forwarded ... 181
X-Forwarded-For ... 182
X-Forwarded-Proto ... 185
X-Forwarded-Port ... 186
modificação do cabeçalho ... 186
Renomear mTLS/TLS cabeçalhos ... 186
Adicionar cabeçalhos de resposta ... 188
Desativar cabeçalhos ... 190
Limitações ... 190
Ativar modificação do cabeç ... 191
Excluir um listener ... 194
Grupos de destino ... 195
Configuração de roteamento ... 196
Target type ... 196
Tipo de endereço IP ... 198
Versão do protocolo ... 199
Destinos registrados ... 200
Atributos do grupo de destino ... 201
Algoritmos de roteamento ... 203
Modificar o algoritmo de roteamento de um grupo de destino ... 205
Integridade do grupo de destino ... 205
Ações para estado não íntegro ... 206
Requisitos e considerações ... 206
Monitoramento ... 207
Exemplo ... 207
Como usar o failover de DNS do Route 53 para o seu balanceador de carga ... 209

Criar um grupo de destino ... 210
Atualizar configurações de íntegridade ... 212
Configurar verificações de integridade ... 213
Configurações de verificação de integridade ... 214
Status de integridade do destino ... 216
Códigos de motivo de verificação de integridade ... 218
Verificar a integridade do destino ... 219
Atualizar configurações de verificação de integridade ... 220
Editar atributos do grupo de destino ... 221
Atraso do cancelamento do registro ... 221
Modo de iniciação lenta ... 222
Balanceamento de carga entre zonas ... 223
Ponderações de destinos automáticos (ATW) ... 226
Sessões persistentes ... 230
Registrar destinos ... 237
Grupos de segurança de destino ... 238
Sub-redes compartilhadas ... 239
Registrar ou cancelar o registro de destinos ... 239
Usar funções do Lambda como destinos ... 242
Preparar a função do Lambda ... 243
Criar um grupo de destino para a função do Lambda ... 241
Receber eventos do balanceador de carga ... 245
Responder ao balanceador de carga ... 246
Cabeçalhos de vários valores ... 247
Habilitar verificações de integridade ... 249
Cancelar o registro da função do Lambda ... 251
Marcar um grupo de destino ... 251
Excluir um grupo de destino ... 252
Monitorar os balanceadores de carga ... 254
CloudWatch métricas ... 255
Métricas do Application Load Balancer ... 256
Dimensões de métrica para Application Load Balancers ... 278
Estatísticas para métricas do Application Load Balancer ... 279
Veja CloudWatch as métricas do seu balanceador de carga ... 280
Logs de acesso ... 282
Arquivos do log de acesso ... 283

Entradas do log de acesso ... 285
Exemplo de entradas de log ... 301
Processar arquivos de log de acesso ... 303
Habilitar logs de acesso ... 303
Desabilitar logs de acesso ... 314
Logs de conexão ... 315
Arquivos de log de conexão ... 316
Entradas de log de conexão ... 317
Exemplo de entradas de log ... 321
Processamento dos arquivos de log de conexão ... 322
Habilitar logs de conexão ... 322
Desabilitar logs de conexão ... 332
Rastreamento de solicitação ... 333
Sintaxe ... 333
Limitações ... 334
Solucionar problemas em seus balanceadores de carga ... 336
Um destino registrado não está em serviço ... 336
Os clientes não conseguem se conectar a um balanceador de carga voltado para a Internet ... 338
As solicitações enviadas para um domínio personalizado não são recebidas pelo balanceador de carga. ... 338
As solicitações HTTPS enviadas ao balanceador de carga retornam "NET::ERR_CERT_COMMON_NAME_INVALID" ... 339
O balanceador de carga mostra tempos elevados de processamento ... 339
O load balancer envia um código de resposta de 000 ... 340
O load balancer gera um erro de HTTP ... 340
HTTP 400: solicitação inválida ... 341
HTTP 401: Não autorizado ... 341
HTTP 403: negado ... 341
HTTP 405: método não permitido ... 341
HTTP 408: Request Timeout (HTTP 408: limite de tempo de solicitação) ... 342
HTTP 413: carga útil muito grande ... 342
HTTP 414: URI muito longo ... 342
HTTP 460 ... 342
HTTP 463 ... 342
HTTP 464 ... 342
HTTP 500: erro interno do servidor ... 343

HTTP 501: não implementado 343
HTTP 502: Bad Gateway (HTTP 502: gateway incorreto) 343
HTTP 503: Service Unavailable (HTTP 503: serviço indisponível) 344
HTTP 504: Gateway Timeout (HTTP 504: limite de tempo do gateway) 344
HTTP 505: versão incompatível 345
HTTP 507: armazenamento insuficiente 345
HTTP 561: Não autorizado 345
Um destino gera um erro HTTP 345
Um AWS Certificate Manager certificado não está disponível para uso 345
Não há compatibilidade com cabeçalhos de várias linhas. 346
Solucionar problemas de destinos não íntegros usando o mapa de recursos 346
Cotas 349
Balanceadores de cargas 349
Grupos de destino 350
Regras 350
Armazenamentos confiáveis 351
Certificados 351
Cabeçalhos HTTP 352
Unidades de capacidade do Load Balancer 352
Histórico de documentos 353
..... ccclxi

O que é um Application Load Balancer?

O Elastic Load Balancing distribui automaticamente seu tráfego de entrada em vários destinos, como EC2 instâncias, contêineres e endereços IP, em uma ou mais zonas de disponibilidade. Ele monitora a integridade dos destinos registrados e roteia o tráfego apenas para os destinos íntegros. O Elastic Load Balancing escala seu balanceador de carga conforme seu tráfego de entrada muda com o tempo. Ele pode ser dimensionado automaticamente para a vasta maioria das cargas de trabalho.

O Elastic Load Balancing oferece suporte aos seguintes balanceadores de carga: balanceadores de carga da aplicação, balanceadores de carga da rede, balanceadores de carga do gateway e balanceadores de carga clássicos. Você pode selecionar o tipo de balanceador de carga que melhor se adapte às suas necessidades. Este guia aborda os Application Load Balancers Para obter mais informações sobre os outros balanceadores de carga, consulte o Guia do usuário de Network Load Balancers, o Guia do usuário de Gateway Load Balancers e o Guia do usuário de Classic Load Balancers.

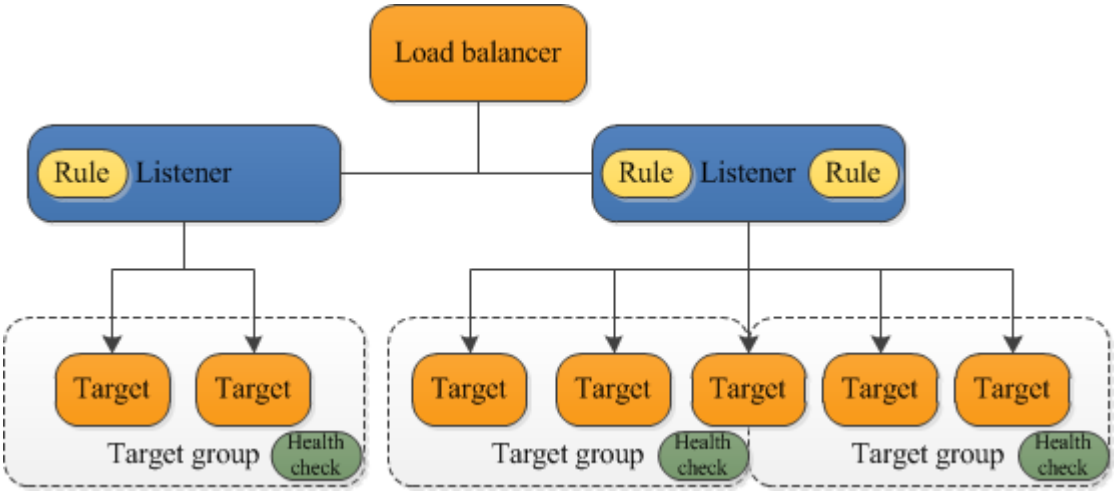
Componentes do Application Load Balancer

Um load balancer serve como ponto único de contato para os clientes. O balanceador de carga distribui o tráfego de entrada do aplicativo em vários destinos, como EC2 instâncias, em várias zonas de disponibilidade. Isso aumenta a disponibilidade do seu aplicativo. Você adiciona um ou mais listeners ao seu load balancer.

Um listener verifica a solicitações de conexão de clientes, usando o protocolo e a porta que você configurar. As regras que você define para um listener determinam como o load balancer roteia solicitações para seus destinos registrados. Cada regra consiste em uma prioridade, uma ou mais ações e uma ou mais condições. Quando as condições de uma regra forem atendidas, a ação será executada. É necessário definir uma regra padrão para cada listener e, opcionalmente, você poderá definir regras adicionais.

Cada grupo de destino encaminha solicitações para um ou mais destinos registrados, como EC2 instâncias, usando o protocolo e o número da porta que você especificar. Você pode registrar um destino com vários grupos de destino. Você pode configurar verificações de integridade em cada grupo de destino. As verificações de integridade são executadas em todos os destinos registrados a um grupo de destino especificado em uma regra de listeners para seu load balancer.

O diagrama a seguir ilustra os componentes básicos. Observe que cada listener contém uma regra padrão e um listener contém outra regra que roteia solicitações para um grupo de destino diferente. Um destino é registrado com dois grupos de destino.



Para obter mais informações, consulte a seguinte documentação do :

- balanceador de cargas
- Listeners
- Grupos de destino

Visão geral do Application Load Balancer

Um Application Load Balancer funciona na camada de aplicativos, a sétima camada do modelo Open Systems Interconnection (OSI). Depois que o load balancer recebe a solicitação, ele avalia as regras do listener em ordem de prioridade para determinar qual regra deve ser aplicada e, em seguida, seleciona um destino no grupo de destino para a ação da regra. Você pode configurar regras do listener para rotear as solicitações para diferentes grupos de destino com base no conteúdo do tráfego do aplicativo. O roteamento é realizado de forma independente para cada grupo de destino, até mesmo quando um destino é registrado com vários grupos de destino. Você pode configurar o algoritmo de roteamento usado no nível do grupo de destino. O algoritmo de roteamento padrão é o de ida e volta. Como alternativa, você pode especificar o algoritmo de roteamento de solicitações menos pendentes.

Você pode adicionar e remover destinos do balanceador de carga conforme suas necessidades mudarem, sem perturbar o fluxo geral de solicitações para sua aplicação. O Elastic Load Balancing

escala seu balanceador de carga à medida que o tráfego para sua aplicação muda com o tempo. O Elastic Load Balancing pode ser escalado para a vasta maioria de workloads automaticamente.

Você pode configurar verificações de integridade, que são usadas para monitorar a integridade dos destinos registrados para que o load balancer possa enviar solicitações apenas para os destinos íntegros.

Para obter mais informações, consulte Como o Elastic Load Balancing funciona no Manual do usuário do Elastic Load Balancing.

Benefícios da migração de um Classic Load Balancer

O uso de um Application Load Balancer em vez de um Classic Load Balancer oferece os seguintes benefícios:

- Suporte a Condições do caminho. Você pode configurar regras para seu listener que encaminhe as solicitações com base no URL da solicitação. Isso permite que você estruture seu aplicativo em serviços menores e roteie-as ao serviço correto com base no conteúdo do URL.
- Suporte a Condições do host. Você pode configurar regras para seu listener que encaminhem solicitações baseadas no campo do host no cabeçalho do HTTP. Isso permite que você roteie solicitações para vários domínios usando um único load balancer.
- Compatibilidade com roteamento baseado em campos na solicitação, como Condições de cabeçalho HTTP e métodos, parâmetros de consulta e endereços IP de origem.
- Support para rotear solicitações para vários aplicativos em uma única EC2 instância. Você pode registrar cada instância ou endereço IP com o mesmo grupo de destino usando portas diferentes.
- Compatibilidade para redirecionar solicitações de um URL para outro.
- Compatibilidade para retornar uma resposta HTTP personalizada.
- Suporte para registrar destinos por endereço IP, incluindo destinos fora da VPC para o load balancer.
- Compatibilidade para registrar as funções Lambda como destinos.
- Compatibilidade com o load balancer para autenticar os usuários de seus aplicativos por meio da identidade corporativa ou social desses usuários antes das solicitações de roteamento.
- Compatibilidade com aplicações em contêineres. O Amazon Elastic Container Service (Amazon ECS) pode selecionar uma porta não utilizada ao programar uma tarefa e registrá-la em um grupo de destino usando essa porta. Isso permite que você faça um uso eficiente dos seus clusters.

- Support para monitorar a integridade de cada serviço de forma independente, pois as verificações de saúde são definidas no nível do grupo-alvo e muitas CloudWatch métricas são relatadas no nível do grupo-alvo. Anexar um grupo de destino a um grupo do Auto Scaling permite que você escale cada serviço dinamicamente com base na demanda.
- Os logs de acesso contêm informações adicionais e são armazenados em formato compactado.
- Melhora no desempenho do load balancer.

Para obter mais informações sobre os recursos suportados por cada tipo de balanceador de carga, consulte Recursos do Elastic Load Balancing.

Serviços relacionados

O Elastic Load Balancing funciona com os serviços a seguir para melhorar a disponibilidade e a escalabilidade das suas aplicações.

- Amazon EC2 — Servidores virtuais que executam seus aplicativos na nuvem. Você pode configurar seu balanceador de carga para direcionar o tráfego para suas EC2 instâncias.
- Amazon EC2 Auto Scaling — Garante que você esteja executando o número desejado de instâncias, mesmo se uma instância falhar, e permite que você aumente ou diminua automaticamente o número de instâncias à medida que a demanda por suas instâncias muda. Se você habilitar o Auto Scaling com o Elastic Load Balancing, as instâncias iniciadas pelo Auto Scaling serão registradas automaticamente no grupo de destino, e as instâncias encerradas pelo Auto Scaling terão o registro cancelado automaticamente do grupo de destino.
- AWS Certificate Manager: ao criar um receptor HTTPS, você pode especificar certificados fornecidos pelo ACM. O load balancer usa certificados para encerrar conexões e descriptografar solicitações de clientes. Para obter mais informações, consulte Certificados SSL para o Application Load Balancer.
- Amazon CloudWatch — Permite monitorar seu balanceador de carga e agir conforme necessário. Para obter mais informações, consulte CloudWatch métricas para seu Application Load Balancer.
- Amazon ECS — Permite que você execute, interrompa e gerencie contêineres Docker em um cluster de EC2 instâncias. Você pode configurar o load balancer para rotear o tráfego para seus contêineres. Para obter mais informações, consulte Balanceamento de carga de serviço no Guia do desenvolvedor do Amazon Elastic Container Service.
- AWS Global Accelerator: melhora a disponibilidade e o desempenho da sua aplicação. Use um acelerador para distribuir o tráfego entre vários balanceadores de carga em uma ou mais

AWS regiões. Para obter mais informações, consulte o Guia do desenvolvedor do AWS Global Accelerator.

- Route 53 — Fornece uma maneira confiável e econômica de direcionar visitantes para sites, traduzindo nomes de domínio (como`www.example.com`) em endereços IP numéricos (como`192.0.2.1`) que os computadores usam para se conectar uns aos outros. AWS atribui URLs aos seus recursos, como balanceadores de carga. No entanto, você pode querer um URL que seja fácil para seus usuários se lembrarem. Por exemplo, você pode mapear o nome de domínio a um load balancer. Para obter mais informações, consulte Rotear tráfego para um balanceador de carga ELB no Guia do desenvolvedor do Amazon Route 53.
- AWS WAF— Você pode usar AWS WAF com seu Application Load Balancer para permitir ou bloquear solicitações com base nas regras de uma lista de controle de acesso à web (web ACL). Para obter mais informações, consulte AWS WAF.

Para ver informações sobre serviços integrados ao seu balanceador de carga, selecione seu balanceador de carga na guia Serviços integrados AWS Management Console e escolha a guia Serviços integrados.

Preços

Com o load balancer, você paga somente pelo que utilizar. Para obter mais informações, consulte Preço do Elastic Load Balancing.

Como começar a usar Application Load Balancers

Este tutorial fornece uma introdução prática aos Application Load Balancers por meio da AWS Management Console, uma interface baseada na web. Para criar seu primeiro Application Load Balancer, conclua as etapas a seguir.

Conteúdo

- Antes de começar
- Etapa 1: configurar seu grupo de destino
- Etapa 2: escolher um tipo de balanceador de carga
- Etapa 3: configurar o balanceador de carga e um receptor
- Etapa 4: testar o balanceador de carga
- Etapa 5: (opcional) excluir o balanceador de carga

Para demonstrações de configurações comuns do balanceador de carga, consulte Demonstrações do Elastic Load Balancing.

Antes de começar

- Decida quais duas zonas de disponibilidade você usará para suas EC2 instâncias. Configure sua nuvem privada virtual (VPC) com, pelo menos, uma sub-rede pública em cada uma destas Zonas de disponibilidade. Essas sub-redes públicas são usadas para configurar o load balancer. Em vez disso, você pode iniciar suas EC2 instâncias em outras sub-redes dessas zonas de disponibilidade.
- Execute pelo menos uma EC2 instância em cada zona de disponibilidade. Certifique-se de instalar um servidor web, como o Apache ou o Internet Information Services (IIS), em cada EC2 instância. Verifique se os security groups dessas instâncias permitem acesso HTTP na porta 80.

Etapa 1: configurar seu grupo de destino

Crie um grupo de destino, que é usado no roteamento da solicitação. A regra padrão para o seu listener roteia solicitações para os destinos registrados neste grupo de destino. O load balancer verifica a integridade dos destinos desse grupo de destino usando as configurações de verificação de integridade definidas para o grupo de destino.

Para configurar o grupo de destino usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Balanceamento de carga, selecione Grupos de destino.
3. Selecione Criar grupo de destino.
4. Em Configuração básica, mantenha o Tipo de destino como instância.
5. Em Nome do grupo de destino, digite um nome para o novo grupo de destino.
6. Mantenha o protocolo padrão (HTTP) e a porta (80).
7. Selecione a VPC que contém suas instâncias. Mantenha a versão do protocolo como HTTP1.
8. Para Health checks (Verificações de integridade), mantenha as configurações padrão.
9. Escolha Próximo.
10. Na página Registrar destinos, conclua as etapas a seguir. Essa é uma etapa opcional para criar um balanceador de carga. No entanto, você deve registrar esse destino se quiser testar o balanceador de carga e garantir que ele esteja roteando o tráfego para os destinos.
    a. Em Instâncias disponíveis, selecione uma ou mais instâncias.
    b. Mantenha a porta 80 padrão e escolha Incluir como pendente abaixo.
11. Selecione Criar grupo de destino.

Etapa 2: escolher um tipo de balanceador de carga

O Elastic Load Balancing oferece suporte para diferentes tipos de balanceadores de carga. Neste tutorial, você criará um Application Load Balancer.

Para criar um Application Load Balancer usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. Na barra de navegação, escolha uma região para seu balanceador de carga. Certifique-se de escolher a mesma região que você usou para suas EC2 instâncias.
3. No painel de navegação, em Load Balancing, selecione Load Balancers.
4. Selecione Criar load balancer.
5. Para Application Load Balancer, escolha Create (Criar).

Etapa 3: configurar o balanceador de carga e um receptor

Para criar um Application Load Balancer, primeiro você deve fornecer informações básicas para o balanceador de carga, como nome, esquema e tipo de endereço IP. Em seguida, você fornece informações sobre sua rede e um ou mais receptores. Um listener é um processo que verifica se há solicitações de conexão. Ele é configurado com um protocolo e uma porta para as conexões de clientes com o load balancer. Para obter mais informações sobre protocolos e portas suportados, consulte Configuração do receptor.

Para configurar seu load balancer e seu listener

1. Em Load balancer name (Nome do balanceador de carga), insira um nome para o seu balanceador de carga. Por exemplo, `my-alb`.
2. Para Esquema e Tipo de endereço IP, mantenha os valores padrão.
3. Para mapeamento de rede, selecione a VPC que você usou para suas EC2 instâncias. Selecione ao menos duas zonas de disponibilidade e uma sub-rede por zona. Para cada zona de disponibilidade que você usou para iniciar suas EC2 instâncias, selecione a zona de disponibilidade e, em seguida, selecione uma sub-rede pública para essa zona de disponibilidade.
4. Em Grupos de segurança, escolhemos o grupo de segurança padrão para a VPC que você selecionou na etapa anterior. Como alternativa, você pode escolher um grupo de segurança diferente. O grupo de segurança deve incluir regras que permitam que o balanceador de carga se comunique com destinos registrados tanto na porta do receptor quanto na porta de verificação de integridade. Para obter mais informações, consulte Regras de grupos de segurança.
5. Em Receptores e roteamento, mantenha o protocolo e a porta padrão e selecione seu grupo de destino na lista. Isso configura um receptor que aceita tráfego HTTP na porta 80 e encaminha o tráfego para o grupo de destino selecionado por padrão. Neste tutorial, você não está criando um listener HTTPS.
6. Em Ação padrão, selecione o grupo de destino que você criou e registrou na Etapa 1: configurar seu grupo de destino.
7. (Opcional) Adicione uma tag para caracterizar o balanceador de carga. As chaves de tag devem ser exclusivas de cada load balancer. Os caracteres permitidos são letras, espaços, números (em UTF-8) e os caracteres especiais a seguir: + - = . _ : / @. Não use espaços no início nem no fim. Os valores de tags diferenciam maiúsculas de minúsculas.

8. Revise sua configuração e escolha Create load balancer (Criar um balanceador de carga). Alguns atributos padrão são aplicados ao balanceador de carga durante a criação. Você pode visualizá-los e editá-los depois de criar o balanceador de carga. Para obter mais informações, consulte Atributos do load balancer.

Etapa 4: testar o balanceador de carga

Depois de criar o balanceador de carga, verifique se ele está enviando tráfego para suas EC2 instâncias.

Para testar seu load balancer

1. Após receber a notificação sobre a criação do load balancer com êxito, selecione Fechar.
2. No painel de navegação, em Balanceamento de carga, selecione Grupos de destino.
3. Selecione o grupo de destino recém-criado.
4. Escolha Destinos e verifique se a sua instância está pronta. Se o status de uma instância for `initial`, talvez seja porque a instância ainda está no processo de ser registrada ou ainda não passou pelo número mínimo de verificações de integridade para ser considerada íntegra. Após o status de pelo menos uma instância ser `healthy`, você pode testar seu load balancer.
5. No painel de navegação, em Load Balancing, selecione Load Balancers.
6. Selecione o load balancer recém-criado.
7. Escolha Descrição e copie o nome DNS do balanceador de carga (por exemplo, my-load-balancer -1234567890abcdef. elb.us-east-2.amazonaws.com). Cole o nome DNS no campo de endereço de um navegador da web conectado à Internet. Se tudo estiver funcionando, o navegador exibirá a página padrão do seu servidor.
8. (Opcional) Para definir outras regras do listener, consulte Adicione uma regra de ouvinte para seu Application Load Balancer.

Etapa 5: (opcional) excluir o balanceador de carga

Assim que o load balancer é disponibilizado, você será cobrado por cada hora ou hora parcial em que mantê-lo em execução. Quando não precisar mais do load balancer, pode excluí-lo. Assim que o load balancer for excluído, a cobrança será interrompida. Observe que a exclusão de um load balancer não afeta os destinos registrados com o load balancer. Por exemplo, suas EC2 instâncias continuam em execução após a exclusão do balanceador de carga criado neste guia.

Para excluir um balanceador de carga usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing, selecione Load Balancers.
3. Marque a caixa de seleção para o balanceador de carga e selecione Ações e Excluir.
4. Quando a confirmação for solicitada, escolha Sim, excluir.

Introdução aos Application Load Balancers usando o AWS CLI

Este tutorial fornece uma introdução prática aos Application Load Balancers por meio do. AWS CLI

Conteúdo

- Antes de começar
- Criar um load balancer
- Adicionar um receptor HTTPS
- Adicionar roteamento baseado em caminho
- Excluir o load balancer

Antes de começar

- Use o comando a seguir para verificar se você está executando uma versão da AWS CLI compatível com Application Load Balancers.

  ```
  aws elbv2 help
  ```

  Se você receber uma mensagem de erro de que o elbv2 não é uma opção válida, atualize sua AWS CLI. Para obter mais informações, consulte Instalando a versão mais recente do AWS CLI no Guia AWS Command Line Interface do Usuário.
- Execute suas EC2 instâncias em uma nuvem privada virtual (VPC). Certifique-se de que os security groups dessas instâncias permitam acesso na porta do listener e na porta de verificação de integridade. Para obter mais informações, consulte Grupos de segurança de destino.
- Decida se você criará um balanceador de IPv4 carga ou dualstack. Use IPv4 se quiser que os clientes se comuniquem com o balanceador de carga usando somente IPv4 endereços. Use o dualstack se quiser que os clientes se comuniquem com o balanceador de carga usando endereços. IPv4 IPv6 Você também pode usar o dualstack para se comunicar com destinos de back-end, como IPv6 aplicativos ou sub-redes dualstack, usando. IPv6
- Certifique-se de instalar um servidor web, como o Apache ou o Internet Information Services (IIS), em cada EC2 instância. Verifique se os security groups dessas instâncias permitem acesso HTTP na porta 80.

Criar um load balancer

Para criar seu primeiro load balancer, conclua as etapas a seguir.

Para criar um balanceador de carga

1. Use o create-load-balancercomando para criar um balanceador de carga. Você deve especificar duas sub-redes que não estejam na mesma Zona de disponibilidade.

```
aws elbv2 create-load-balancer --name my-load-balancer  \
--subnets subnet-0e3f5cac72EXAMPLE subnet-081ec835f3EXAMPLE --security-groups
 sg-07e8ffd50fEXAMPLE
```

   Use o create-load-balancercomando para criar um **dualstack** balanceador de carga.

```
aws elbv2 create-load-balancer --name my-load-balancer  \
--subnets subnet-0e3f5cac72EXAMPLE subnet-081ec835f3EXAMPLE --security-groups
 sg-07e8ffd50fEXAMPLE --ip-address-type dualstack
```

   O resultado inclui o Nome de recurso da Amazon (ARN) do load balancer, com o seguinte formato:

```
arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/my-load-
balancer/1234567890123456
```

2. Use o create-target-groupcomando para criar um grupo de destino, especificando a mesma VPC que você usou para EC2 suas instâncias.

   Você pode criar IPv4 e IPv6 direcionar grupos para associá-los aos balanceadores de carga de pilha dupla. O tipo de endereço IP do grupo de destino determina a versão do IP que o balanceador de carga usará para se comunicar e verificar a integridade dos destinos de backend.

```
aws elbv2 create-target-group --name my-targets --protocol HTTP --port 80 \
--vpc-id vpc-0598c7d356EXAMPLE --ip-address-type [ipv4 or ipv6]
```

   A saída inclui o ARN do grupo de destino, com este formato:

```
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/1234567890123456
```

3. Use o comando register-targets para registrar suas instâncias com o grupo de destino:

```
aws elbv2 register-targets --target-group-arn targetgroup-arn  \
--targets Id=i-0abcdef1234567890 Id=i-1234567890abcdef0
```

4. Use o comando create-listener para criar um listener para seu load balancer com uma regra padrão que encaminha solicitações ao seu grupo de destino:

```
aws elbv2 create-listener --load-balancer-arn loadbalancer-arn \
--protocol HTTP --port 80  \
--default-actions Type=forward,TargetGroupArn=targetgroup-arn
```

   A saída contém o ARN do listener, com o seguinte formato:

```
arn:aws:elasticloadbalancing:us-east-2:123456789012:listener/app/my-load-
balancer/1234567890123456/1234567890123456
```

5. (Opcional) Você pode verificar a integridade dos alvos registrados para seu grupo-alvo usando este describe-target-healthcomando:

```
aws elbv2 describe-target-health --target-group-arn targetgroup-arn
```

Adicionar um receptor HTTPS

Se você tiver um load balancer com um listener HTTP, pode adicionar um listener HTTPS como a seguir.

Para adicionar um listener HTTPS ao seu load balancer

1. Crie um certificado SSL para uso com o seu load balancer usando um dos seguintes métodos:

   - Crie ou importe o certificado usando AWS Certificate Manager (ACM). Para obter mais informações, consulte Solicitar um certificado público ou Importar certificados no Guia AWS Certificate Manager do usuário.

- Faça o upload do certificado usando AWS Identity and Access Management (IAM). Para obter mais informações, consulte Working with server certificates (Trabalho com certificados do servidor) no Guia de usuário do IAM.

2. Use o comando create-listener para criar o listener com uma regra padrão que encaminha solicitações para seu grupo de destino. Você deve especificar um certificado SSL ao criar um listener HTTPS. Observe que você pode especificar uma política SSL diferente da padrão usando a opção `--ssl-policy`.

```
aws elbv2 create-listener --load-balancer-arn loadbalancer-arn \
--protocol HTTPS --port 443  \
--certificates CertificateArn=certificate-arn \
--default-actions Type=forward,TargetGroupArn=targetgroup-arn
```

Adicionar roteamento baseado em caminho

Se você tiver um listener com uma regra padrão que encaminha solicitações a um grupo de destino, pode adicionar uma regra que encaminhe solicitações a outro grupo de destino com base no URL. Por exemplo, você pode rotear solicitações gerais para um grupo de destino e solicitar a exibição de imagens a outro grupo de destino.

Para adicionar uma regra para um listener com um padrão de caminho

1. Use o create-target-groupcomando para criar um grupo-alvo:

```
aws elbv2 create-target-group --name my-targets --protocol HTTP --port 80 \
--vpc-id vpc-0598c7d356EXAMPLE
```

2. Use o comando register-targets para registrar suas instâncias com o grupo de destino:

```
aws elbv2 register-targets --target-group-arn targetgroup-arn  \
--targets Id=i-0abcdef1234567890 Id=i-1234567890abcdef0
```

3. Use o comando create-rule para adicionar uma regra para o listener que encaminha solicitações ao grupo de destino se o URL contiver o padrão especificado:

```
aws elbv2 create-rule --listener-arn listener-arn --priority 10 \
--conditions Field=path-pattern,Values='/img/*' \
--actions Type=forward,TargetGroupArn=targetgroup-arn
```

Excluir o load balancer

Quando você não precisar mais de seu load balancer e grupo de destino, pode excluí-los da seguinte forma:

```
aws elbv2 delete-load-balancer --load-balancer-arn loadbalancer-arn
aws elbv2 delete-target-group --target-group-arn targetgroup-arn
```

Application Load Balancers

Um load balancer serve como ponto único de contato para os clientes. Os clientes enviam solicitações para o balanceador de carga, e o balanceador de carga as envia para destinos, como EC2 instâncias. Para configurar o load balancer, você cria grupos de destino e, em seguida, registra os destinos nesses grupos. Você também pode criar listeners para verificar a solicitações de conexão de clientes, além de regras dos listeners para rotear solicitações dos clientes para os destinos em um ou mais grupos de destino.

Para obter mais informações, consulte Como o Elastic Load Balancing funciona no Manual do usuário do Elastic Load Balancing.

Conteúdo

- Sub-redes para seu balanceador de carga
- Grupos de segurança do balanceador de carga
- Estado do load balancer
- Atributos do load balancer
- Tipo de endereço IP
- Pools de endereços IP IPAM
- Conexões do balanceador de carga
- Balanceamento de carga entre zonas
- Nome DNS
- Criar um Application Load Balancer
- Atualizar zonas de disponibilidade para o Application Load Balancer
- Grupos de segurança para seu Application Load Balancer
- Atualizar os tipos de endereço IP para o Application Load Balancer
- Atualize os pools de endereços IP IPAM para seu Application Load Balancer
- Editar atributos para o Application Load Balancer
- Marcar um Application Load Balancer
- Excluir um Application Load Balancer
- Exibir o mapa de recursos do Application Load Balancer
- Mudança de zona para o Application Load Balancer
- Reservas de capacidade para seu Application Load Balancer

- Integrações para seu Application Load Balancer

Sub-redes para seu balanceador de carga

Ao criar um Application Load Balancer, você deve habilitar as zonas que contêm seus destinos. Para habilitar uma zona, especifique uma sub-rede na zona. O Elastic Load Balancing cria um nó de balanceador de carga em cada zona que você especificar.

Considerações

- O balanceador de carga será mais eficaz se você garantir que cada zona de disponibilidade habilitada tenha ao menos um destino registrado.
- Se você registrar destinos em uma zona, mas não habilitá-la, esses destinos registrados não receberão tráfego do balanceador de carga.
- Se você habilitar várias zonas para seu balanceador de carga, elas precisarão ser do mesmo tipo. Por exemplo, você não pode habilitar uma zona de disponibilidade e uma zona local.
- Você pode especificar uma sub-rede que tenha sido compartilhada com você.

Os Application Load Balancers são compatíveis com os seguintes tipos de sub-redes.

Tipos de sub-redes

- Sub-redes de zona de disponibilidade
- Sub-redes de zona local
- Sub-redes de Outpost

Sub-redes de zona de disponibilidade

Você deve selecionar ao menos duas sub-redes de zona de disponibilidade. As seguintes restrições são aplicáveis:

- Cada sub-rede deve estar em uma zona de disponibilidade diferente.
- Para garantir que o balanceador de carga possa escalar corretamente, verifique se cada sub-rede da zona de disponibilidade do balanceador de carga tem um bloco CIDR com ao menos uma bitmask `/27` (por exemplo, `10.0.0.0/27`) e pelo menos oito endereços IP livres por sub-rede. Esses oito endereços IP são necessários para permitir que o balanceador de carga aumente a escala horizontalmente, se for o caso. Seu load balancer usa esses endereços IP para estabelecer

conexões com os destinos. Sem eles, seu Application Load Balancer pode ter dificuldades com as tentativas de substituição de nós, fazendo com que ele entre em um estado de falha.

Obs.: se uma sub-rede do Application Load Balancer ficar sem endereços IP utilizáveis ao tentar escalar, o Application Load Balancer será executado com capacidade insuficiente. Durante esse período, os nós antigos continuarão a fornecer tráfego, mas a tentativa paralisada de escalonamento poderá causar erros 5xx ou tempos limite ao tentar estabelecer uma conexão.

Sub-redes de zona local

Você pode especificar uma ou mais sub-redes de zona local. Os seguintes recursos não são suportados:

- Funções do Lambda como destinos
- Autenticação TLS mútua
- AWS WAF integração

Sub-redes de Outpost

Você pode especificar uma só sub-rede de Outpost. As seguintes restrições são aplicáveis:

- Um Outpost deve estar instalado e configurado no data center on-premises. É necessário ter uma conexão de rede confiável entre o Outpost e a região da AWS . Para obter mais informações, consulte o Guia do usuário do AWS Outposts.
- O balanceador de carga requer duas instâncias `large` no Outpost para os nós do balanceador de carga. Os tipos de instância compatíveis são apresentados na tabela a seguir. O balanceador de carga escala conforme necessário, redimensionando os nós um tamanho por vez (de `large` para `xlarge`, depois de `xlarge` para `2xlarge` e depois de `2xlarge` para `4xlarge`). Após escalar os nós para o maior tamanho de instância, se você precisar de capacidade adicional, o balanceador de carga adicionará instâncias `4xlarge` como nós do balanceador de carga. Se você não tiver capacidade de instância suficiente ou endereços IP disponíveis para escalar o balanceador de carga, o balanceador de carga relatará um evento para o AWS Health Dashboard e o estado do balanceador de carga será `active_impaired`.
- É possível registrar destinos por ID de instância ou por endereço IP. Se você registrar alvos na AWS região para o Posto Avançado, eles não serão usados.
- Os seguintes recursos não são suportados:

- AWS Global Accelerator integração
- Funções do Lambda como destinos
- Autenticação TLS mútua
- Sessões persistentes
- Autenticação de usuário
- AWS WAF integração

É possível implantar um Application Load Balancer em instâncias c5/c5d, m5/m5d ou r5/r5d em um Outpost. A tabela a seguir mostra o tamanho e o volume do EBS por tipo de instância que o balanceador de carga pode usar em um Outpost:

Tipo e tamanho de instância	Volume do EBS (GB)
c5/c5d	
grande	50
xlarge	50
2xlarge	50
4xlarge	100
m5/m5d	
grande	50
xlarge	50
2xlarge	100
4xlarge	100
r5/r5d	
grande	50
xlarge	100

Tipo e tamanho de instância	Volume do EBS (GB)
2xlarge	100
4xlarge	100

Grupos de segurança do balanceador de carga

Um security group atua como um firewall que controla o tráfego permitido de e para o load balancer. Você pode escolher as portas e os protocolos para permitir tráfego tanto de entrada quanto de saída.

As regras para os grupos de segurança que estão associados ao balanceador de carga devem permitir tráfego em ambas as direções tanto no receptor quanto nas portas de verificação de integridade. Sempre que você adicionar um listener a um load balancer ou atualizar a porta de verificação de integridade de um grupo de destino, será necessário revisar as regras do security group para garantir que elas permitam tráfego na nova porta em ambas as direções. Para obter mais informações, consulte Regras recomendadas.

Estado do load balancer

O load balancer pode estar em um dos seguintes estados:

`provisioning`

O load balancer está sendo configurado.

`active`

O load balancer está totalmente configurado e pronto para rotear o tráfego.

`active_impaired`

O balanceador de carga está roteando o tráfego, mas não tem os recursos necessários para escalar.

`failed`

O load balancer não pôde ser configurado.

Atributos do load balancer

Você pode configurar seu Application Load Balancer editando os atributos. Para obter mais informações, consulte Editar atributos do balanceador de carga.

A seguir estão os atributos do load balancer:

`access_logs.s3.enabled`

Indica se os logs de acesso armazenados no Amazon S3 estão habilitados. O padrão é `false`.

`access_logs.s3.bucket`

O nome do bucket do Amazon S3 para os logs de acesso. Esse atributo é necessário se os logs de acesso estiverem habilitados. Para obter mais informações, consulte Habilitar logs de acesso.

`access_logs.s3.prefix`

O prefixo para o local no bucket do Amazon S3.

`client_keep_alive.seconds`

O valor da manutenção de atividade do cliente, em segundos. O padrão é 3.600 segundos.

`deletion_protection.enabled`

Indica se a proteção contra exclusão está habilitada. O padrão é `false`.

`idle_timeout.timeout_seconds`

O valor de tempo limite de inatividade, em segundos. O padrão é 60 segundos.

`ipv6.deny_all_igw_traffic`

Bloqueia o acesso do gateway da Internet (IGW) ao balanceador de carga, impedindo o acesso não intencional ao balanceador de carga interno por meio de um gateway da Internet. Ele está configurado como `false` para balanceadores de carga voltados para a Internet e `true` para balanceadores de carga internos. Esse atributo não impede o acesso à Internet que não seja IGW (como, por meio de peering, AWS Direct Connect Transit Gateway ou). AWS VPN

`routing.http.desync_mitigation_mode`

Determina como o balanceador de carga processa solicitações que possam representar risco de segurança para a sua aplicação. Os valores possíveis são `monitor`, `defensive` e `strictest`. O padrão é `defensive`.

`routing.http.drop_invalid_header_fields.enabled`

Indica se os cabeçalhos HTTP com campos de cabeçalho que não sejam válidos serão removidos pelo balanceador de carga (`true`) ou roteados para destinos (`false`). O padrão é `false`. O Elastic Load Balancing exige que os nomes de cabeçalhos HTTP válidos estejam em conformidade com a expressão regular'`[-A-Za-z0-9]+`, conforme descrito no Registro de nomes de campos HTTP. Cada nome consiste em caracteres alfanuméricos ou hifens. Selecione `true` se quiser que os cabeçalhos HTTP que não estejam em conformidade com esse padrão sejam removidos das solicitações.

`routing.http.preserve_host_header.enabled`

Indica se o Application Load Balancer deve preservar o cabeçalho `Host` na solicitação HTTP e enviá-lo para o destino sem nenhuma alteração. Os valores possíveis são `true` e `false`. O padrão é `false`.

`routing.http.x_amzn_tls_version_and_cipher_suite.enabled`

Indica se os dois cabeçalhos (`x-amzn-tls-version` e `x-amzn-tls-cipher-suite`), que contêm informações sobre a versão negociada do TLS e o conjunto de cifras, serão adicionados à solicitação do cliente antes de enviá-la ao destino. O cabeçalho `x-amzn-tls-version` tem informações sobre a versão do protocolo TLS negociada com o cliente e o cabeçalho `x-amzn-tls-cipher-suite` tem informações sobre o pacote de criptografia negociado com o cliente. Ambos os cabeçalhos estão no formato OpenSSL. Os valores possíveis para o atributo são `true` e `false`. O padrão é `false`.

`routing.http.xff_client_port.enabled`

Indica se o cabeçalho `X-Forwarded-For` deve preservar a porta de origem que o cliente usou para se conectar ao balanceador de carga. Os valores possíveis são `true` e `false`. O padrão é `false`.

`routing.http.xff_header_processing.mode`

Permite que você modifique, preserve ou remova o cabeçalho `X-Forwarded-For` na solicitação HTTP antes que o Application Load Balancer envie a solicitação ao destino. Os valores possíveis são `append`, `preserve` e `remove`. O padrão é `append`.

- Se o valor for `append`, o Application Load Balancer adicionará o endereço IP do cliente (do último salto) ao cabeçalho `X-Forwarded-For` na solicitação HTTP antes de enviá-la aos destinos.

- Se o valor for `preserve`, o Application Load Balancer deverá preservar o cabeçalho `X-Forwarded-For` na solicitação HTTP e enviá-lo para o destino sem nenhuma alteração.
- Se o valor for `remove`, o Application Load Balancer removerá o cabeçalho `X-Forwarded-For` na solicitação HTTP e o enviará para o destino sem nenhuma alteração.

`routing.http2.enabled`

Indica se HTTP/2 está habilitado. O padrão é `true`.

`waf.fail_open.enabled`

Indica se um balanceador de carga AWS WAF habilitado deve encaminhar solicitações para destinos caso não consiga encaminhar a solicitação para o. AWS WAF Os valores possíveis são `true` e `false`. O padrão é `false`.

> **Note**
>
> O atributo `routing.http.drop_invalid_header_fields.enabled` foi introduzido para oferecer proteção contra a dessincronização HTTP. O atributo `routing.http.desync_mitigation_mode` foi adicionado para fornecer uma proteção mais abrangente contra a dessincronização HTTP para suas aplicações. Você não precisa usar os dois atributos e pode escolher qualquer um deles de acordo com os requisitos da sua aplicação.

Tipo de endereço IP

É possível definir os tipos de endereços IP que os clientes podem usar para acessar seus balanceadores de carga internos e voltados para a Internet.

Os Application Load Balancers são compatíveis com os seguintes tipos de endereços IP:

`ipv4`

Os clientes devem se conectar ao balanceador de carga usando IPv4 endereços (por exemplo, 192.0.2.1).

`dualstack`

Os clientes podem se conectar ao balanceador de carga usando IPv4 endereços (por exemplo, 192.0.2.1) e IPv6 endereços (por exemplo, 2001:0 db 8:85 a 3:0:0:8 a2e: 0370:7334).

`dualstack-without-public-ipv4`

Os clientes devem se conectar ao balanceador de carga usando IPv6 endereços (por exemplo, 2001:0 db 8:85 a 3:0:0:8 a2e: 0370:7334).

Considerações

- O balanceador de carga se comunica com os destinos com base no tipo de endereço IP do grupo de destino.
- Quando você habilita o modo dualstack para o balanceador de carga, o Elastic Load Balancing fornece um registro DNS AAAA para o balanceador de carga. Os clientes que se comunicam com o balanceador de carga usando IPv4 endereços resolvem o registro DNS A. Os clientes que se comunicam com o balanceador de carga usando IPv6 endereços resolvem o registro DNS AAAA.
- O acesso aos balanceadores de carga dualstack internos por meio do gateway da Internet é bloqueado para impedir o acesso não intencional à Internet. No entanto, isso não impede o acesso à Internet que não seja IGW (como, por meio de peering, AWS Direct Connect Transit Gateway ou). AWS VPN
- A autenticação do Application Load Balancer só é compatível com IPv4 a conexão com um provedor de identidade (IdP) ou endpoint do Amazon Cognito. Sem um IPv4 endereço público, o balanceador de carga não pode concluir o processo de autenticação, resultando em erros HTTP 500.

Para obter mais informações, consulte Atualizar os tipos de endereço IP para o Application Load Balancer.

Pools de endereços IP IPAM

Um pool de endereços IP IPAM é uma coleção de intervalos de endereços IP contíguos (ou) CIDRs, dentro do Amazon VPC IP Address Manager (IPAM). O uso de pools de endereços IP IPAM com seu Application Load Balancer permite que você organize IPv4 seus endereços de acordo com suas necessidades de roteamento e segurança. Os pools de endereços IP IPAM devem primeiro ser criados dentro do IPAM antes de poderem ser usados pelo Application Load Balancer. Para obter mais informações, consulte Traga seus endereços IP para o IPAM.

Considerações

- Os pools de endereços IP IPAM não são compatíveis com balanceadores de carga internos ou com o Dualstack sem o tipo de endereço IP público. IPv4
- Você não pode excluir um endereço IP em um pool de endereços IP IPAM se ele estiver sendo usado atualmente por um balanceador de carga.
- Durante a transição para um pool de endereços IP IPAM diferente, as conexões existentes são encerradas de acordo com a duração do keepalive do cliente HTTP do balanceador de carga.
- Os pools de endereços IP IPAM podem ser compartilhados em várias contas. Para obter mais informações, consulte Configurar opções de integração para seu IPAM

Os pools de endereços IP IPAM oferecem a opção de trazer alguns ou todos os seus intervalos de IPv4 endereços públicos AWS e usá-los com seus Application Load Balancers. Com um melhor controle da atribuição de endereços IP, você pode gerenciar e aplicar políticas e controles de segurança com mais eficiência, além de se beneficiar de custos mais baixos. Não há cobranças adicionais associadas ao uso de pools de endereços IP IPAM com seus Application Load Balancers. No entanto, pode haver cobranças associadas ao IPAM, dependendo do nível usado. Para obter mais informações, consulte os preços do Amazon VPC

Seu pool de endereços IP IPAM é sempre priorizado ao iniciar EC2 instâncias e Application Load Balancers, e quando seus endereços IP não estão mais em uso, eles ficam imediatamente disponíveis novamente. Se não houver mais endereços IP atribuíveis em seu pool de endereços IP IPAM, os endereços IP AWS gerenciados serão atribuídos. AWS endereços IP gerenciados têm custos adicionais. Para adicionar mais endereços IP, você pode adicionar novos intervalos de endereços IP a um pool de endereços IP IPAM existente.

Conexões do balanceador de carga

Ao processar uma solicitação, o balanceador de carga mantém duas conexões: uma com o cliente e outra com o destino. A conexão entre o balanceador de carga e o cliente também é chamada de conexão frontend. A conexão entre o balanceador de carga e o destino também é chamada de conexão de backend.

Balanceamento de carga entre zonas

Com os Application Load Balancers, o balanceamento de carga entre zonas é habilitado por padrão e não pode ser alterado por balanceador de carga. Para mais informações, consulte a seção Balanceamento de carga entre zonas no Guia do usuário do Elastic Load Balancing.

É possível desativar o balanceamento de carga entre zonas por grupo de destino. Para obter mais informações, consulte the section called “Desativar o balanceamento de carga entre zonas”.

Nome DNS

Cada Application Load Balancer recebe um nome de Sistema de Nomes de Domínio (DNS) padrão com a seguinte sintaxe: - .elb. *name id region*.amazonaws.com. Por exemplo, my-load-balancer -1234567890abcdef. elb.us-east-2.amazonaws.com.

Se preferir usar um nome DNS que seja mais fácil de lembrar, é possível criar um nome de domínio personalizado e associá-lo ao nome DNS do Application Load Balancer. Quando um cliente faz uma solicitação usando esse nome de domínio personalizado, o servidor de DNS a resolverá para o nome DNS do Application Load Balancer.

Primeiro, registre um nome de domínio com um registrador de nomes de domínio credenciado. Em seguida, use o serviço de DNS, como o registrador de domínios, para criar um registro de DNS e rotear solicitações ao Application Load Balancer. Para obter mais informações, consulte a documentação do serviço DNS. Por exemplo, se você usar o Amazon Route 53 como serviço de DNS, criará um registro de alias que apontará para o Application Load Balancer. Para obter mais informações, consulte Rotear tráfego para um balanceador de carga ELB no Guia do desenvolvedor do Amazon Route 53.

O Application Load Balancer tem um endereço IP por zona de disponibilidade habilitada. Esses são os endereços IP dos nós do Application Load Balancer. O nome DNS do Application Load Balancer resulta nesses endereços. Por exemplo, vamos supor que o nome de domínio personalizado do Application Load Balancer seja `example.applicationloadbalancer.com`. Use o comando dig ou nslookup a seguir para determinar os endereços IP dos nós do Application Load Balancer.

Linux ou Mac

```
$ dig +short example.applicationloadbalancer.com
```

Windows

```
C:\> nslookup example.applicationloadbalancer.com
```

O Application Load Balancer tem registros DNS para os nós. Você pode usar nomes DNS com a seguinte sintaxe para determinar os endereços IP dos nós do Application Load Balancer:. *az* *name*-... *id* coelho. *region*.amazonaws.com.

Linux ou Mac

```
$ dig +short us-east-2b.my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com
```

Windows

```
C:\> nslookup us-east-2b.my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com
```

Criar um Application Load Balancer

Um load balancer leva solicitações de clientes e as distribui em destinos em um grupo de destino.

Antes de começar, certifique-se de que você tenha uma nuvem privada virtual (VPC) com, no mínimo, uma sub-rede pública em cada uma das zonas usadas por seus destinos. Para obter mais informações, consulte the section called “Sub-redes para seu balanceador de carga”.

Para criar um balanceador de carga usando o AWS CLI, consulteIntrodução aos Application Load Balancers usando o AWS CLI.

Para criar um balanceador de carga usando o AWS Management Console, conclua as tarefas a seguir.

Tarefas

- Etapa 1: configurar um grupo de destino
- Etapa 2: registrar destinos
- Etapa 3: configurar um balanceador de carga e um receptor
- Etapa 4: testar o balanceador de carga

Etapa 1: configurar um grupo de destino

A configuração de um grupo-alvo permite que você registre alvos, como EC2 instâncias. O grupo de destino que você configura nesta etapa é usado como o grupo de destino na regra do receptor

quando você configura o balanceador de carga. Para obter mais informações, consulte Grupos de destino para seus Application Load Balancers.

Para configurar o grupo de destino usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Grupos de destino.
3. Selecione Criar grupo de destino.
4. Na seção Configuração básica, defina os seguintes parâmetros:

   a. Para Escolher um tipo de destino, selecione Instâncias para especificar destinos por ID de instância ou Endereços IP para especificar destinos apenas por endereço IP. Se o tipo de destino for uma função do Lambda, você poderá habilitar as verificações de integridade selecionando Habilitar na seção Verificações de integridade.

   b. Em Nome do grupo de destino, insira um nome para o grupo de destino.

   c. Modifique a Porta e o Protocolo conforme o necessário.

   d. Se o tipo de destino for Instâncias ou endereços IP, escolha IPv4ou IPv6como o tipo de endereço IP, caso contrário, vá para a próxima etapa.

      Observe que somente destinos que tenham o tipo de endereço IP selecionado podem ser incluídos nesse grupo de destino. Não é possível alterar o tipo de endereço IP após a criação do grupo de destino.

   e. Para VPC, selecione uma nuvem privada virtual (VPC) com os destinos que deseja incluir em seu grupo de destino.

   f. Em Versão do protocolo, selecione HTTP1quando o protocolo de solicitação for HTTP/1.1 ou HTTP/2; selecione quando o protocolo de solicitação for HTTP/2 ou gRPC; e selecione gRPC HTTP2, quando o protocolo de solicitação for gRPC.

5. Na seção Verificações de integridade, modifique as configurações padrão conforme necessário. Em Configurações avançadas de verificação de integridade, escolha a porta, a quantidade, o tempo limite, o intervalo e especifique os códigos de sucesso. Se as verificações de integridade excederem o número de Limite não íntegro, o balanceador de carga tornará o destino inoperante. Quando as verificações de integridade excederem o número de Limite íntegro, o balanceador de carga tornará o destino operacional novamente. Para obter mais informações, consulte Verificações de integridade para grupos de destino do Application Load Balancer.
6. (Opcional) Adicione uma ou mais tags, da seguinte forma:

a. Expanda a seção Tags.

b. Escolha Adicionar Tag.

c. Insira a chave e o valor da tag. Os caracteres permitidos são letras, espaços, números (em UTF-8) e os seguintes caracteres especiais: + - = . _ : / @. Não use espaços no início nem no fim. Os valores de tags diferenciam maiúsculas de minúsculas.

7. Escolha Próximo.

Etapa 2: registrar destinos

Você pode registrar EC2 instâncias, endereços IP ou funções Lambda como destinos em um grupo-alvo. Essa é uma etapa opcional para criar um balanceador de carga. No entanto, você deve registrar os destinos para garantir que o balanceador de carga roteie o tráfego para eles.

1. Na página Registrar destinos, adicione um ou mais destinos da seguinte forma:
   - Se o tipo de destino for Instâncias, selecione uma ou mais instâncias, insira uma ou mais portas e escolha Incluir como pendente abaixo.
   - Se o tipo de destino for Endereços IP, faça o seguinte:
     a. Selecione uma rede VPC na lista ou escolha Outros endereços IP privados.
     b. Insira o endereço IP manualmente ou encontre o endereço IP usando os detalhes da instância. É possível inserir até cinco endereços IP por vez.
     c. Insira as portas para rotear o tráfego para os endereços IP especificados.
     d. Escolha Incluir como pendente abaixo.
   - Se o tipo de destino for Lambda, selecione uma função do Lambda ou insira o ARN da função do Lambda e escolha Incluir conforme pendente abaixo.
2. Selecione Criar grupo de destino.

Etapa 3: configurar um balanceador de carga e um receptor

Para criar um Application Load Balancer, primeiro você deve fornecer informações básicas para o balanceador de carga, como nome, esquema e tipo de endereço IP. Em seguida, você fornece informações sobre sua rede e um ou mais receptores. Um listener é um processo que verifica se há solicitações de conexão. Ele é configurado com um protocolo e uma porta para as conexões de

clientes com o load balancer. Para obter mais informações sobre protocolos e portas suportados, consulte Configuração do receptor.

Para configurar seu balanceador de carga e receptor usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Load Balancers.
3. Selecione Criar load balancer.
4. Em Application Load Balancer, escolha Create (Criar).
5. Configuração básica
   a. Em Load balancer name (Nome do balanceador de carga), insira um nome para o seu balanceador de carga. Por exemplo, .**`my-alb`** O nome do seu Application Load Balancer deve ser exclusivo no conjunto de Application Load Balancers e Network Load Balancers para a região. Os nomes podem ter no máximo 32 caracteres, e podem conter somente caracteres alfanuméricos e hifens. Eles não podem começar ou terminar com hífen ou com `internal-`. Não é possível alterar o nome do seu Application Load Balancer após sua criação.
   b. Em Scheme (Esquema), escolha Internet-facing (Voltado para a Internet) ou Internal (Interno). Um balanceador de carga voltado para a Internet roteia solicitações de clientes até destinos na Internet. Um load balancer interno roteia solicitações a destinos usando endereços IP privados.
   c. Para o tipo de endereço IP IPv4, escolha Dualstack ou Dualstack sem público. IPv4 Escolha IPv4se seus clientes usam IPv4 endereços para se comunicar com o balanceador de carga. Escolha Dualstack se seus clientes usarem os dois IPv4 IPv6 endereços para se comunicar com o balanceador de carga. Escolha Dualstack sem público IPv4 se seus clientes usarem somente IPv6 endereços para se comunicar com o balanceador de carga.
6. Mapeamento de rede
   a. Para VPC, selecione a VPC que você usou para suas instâncias. EC2 Se você selecionou Voltado para a Internet para Esquema, somente VPCs com um gateway de Internet estão disponíveis para seleção.
   b. Para grupos de endereços IP IPAM, você pode escolher Usar pool IPAM para endereços públicos IPv4 . Para obter mais informações, consulte Pools de endereços IP IPAM
   c. Para zonas de disponibilidade e sub-redes, habilite zonas para seu balanceador de carga selecionando sub-redes da seguinte forma:

- Sub-redes de duas ou mais zonas de disponibilidade
- Sub-redes de uma ou mais zonas locais
- Uma sub-rede de Outpost

Para obter mais informações, consulte the section called “Sub-redes para seu balanceador de carga”.

Para balanceadores de carga internos, os IPv6 endereços IPv4 e são atribuídos a partir do CIDR da sub-rede.

Se você ativou o modo Dualstack para o balanceador de carga, selecione sub-redes com blocos CIDR e ambos. IPv4 IPv6

7. Em Security groups (Grupos de segurança), selecione um grupo de segurança existente ou crie um novo.

   O security group para o load balancer deve permitir que ele se comunique com destinos registrados tanto na porta do listener quanto na porta de verificação de integridade. O console pode criar um security group para seu load balancer em seu nome com regras que permitam essa comunicação. Você também pode criar um gruo de segurança e selecioná-lo. Para obter mais informações, consulte Regras recomendadas.

   (Opcional) Para criar um novo grupo de segurança para seu balanceador de carga, escolha Criar um novo grupo de segurança.

8. Em Receptores e roteamento, o receptor padrão aceita tráfego HTTP na porta 80. É possível manter o protocolo e a porta padrão ou escolher um protocolo diferente. Em Default action (Ação padrão), escolha o grupo de destino que você criou. É possível, opcionalmente, escolher Add listener (Adicionar listener) para adicionar outro listener (por exemplo, um listener de HTTPS).

9. (Opcional) Se estiver usando um receptor HTTPS

   Em Política de segurança, recomendamos que você sempre use a política de segurança predefinida mais recente.

   a. Para SSL/TLS Certificado padrão, as seguintes opções estão disponíveis:

      - Se você criou ou importou um certificado usando AWS Certificate Manager, selecione Do ACM e selecione o certificado em Selecionar um certificado.

- Se você tiver importado um certificado usando IAM, selecione Do IAM e selecione seu certificado em Selecionar um certificado.
- Se você tiver um certificado para importar, mas o ACM não estiver disponível na sua região, selecione Importar e selecione Para o IAM. Digite o nome do certificado no campo Nome do certificado. Em Chave privada do certificado, copie e cole o conteúdo do arquivo de chave privada (codificado por PEM). Em Corpo do certificado, copie e cole o conteúdo do arquivo do certificado de chave pública (codificado por PEM). Na Cadeia de certificados, copie e cole o conteúdo do arquivo da cadeia do certificado (codificado por PEM), exceto se estiver usando um certificado autoatribuído e se não for importante que os navegadores aceitem implicitamente o certificado.

b. (Opcional) Para habilitar a autenticação mútua, em Tratamento de certificados do cliente, ative a Autenticação mútua (mTLS).

   Quando habilitado, o modo TLS mútuo padrão é de passagem.

   Caso selecione Verificar com o armazenamento confiável:

   - Por padrão, as conexões com certificados expirados de cliente são rejeitadas. Para alterar esse comportamento, expanda Configurações avançadas de mTLS e, em Expiração do certificado do cliente, selecione Permitir certificados expirados de cliente.
   - Em Armazenamentos confiáveis, escolha um armazenamento confiável existente ou Novo armazenamento confiável.
     - Se você escolher Novo armazenamento confiável, forneça um Nome do armazenamento confiável, a Localização da autoridade de certificação de URI do S3 e, opcionalmente, uma Localização da lista de revogação do certificado de URI do S3.
   - (Opcional) Escolha se você deseja habilitar Anunciar nomes de assunto da TrustStore CA.

10. (Opcional) Você pode integrar outros serviços ao seu balanceador de carga durante a criação, em Otimizar com integrações de serviços.

    - Você pode optar por incluir proteções de segurança do AWS WAF no balanceador de carga, com uma ACL da Web existente ou criada automaticamente. Após a criação, a web ACLs pode ser gerenciada no AWS WAF console. Para obter mais informações, consulte Associar ou desassociar uma ACL da web a um AWS recurso no Guia do desenvolvedor.AWS WAF
    - É possível permitir que o AWS Global Accelerator crie um acelerador para você e associar seu balanceador de carga a ele. O nome do acelerador pode ter os seguintes caracteres

(até 64): a-z, A-Z, 0-9, . (ponto final) e - (hífen). Após a criação do acelerador, você poderá gerenciá-lo no console do AWS Global Accelerator. Para obter mais informações, consulte Add an accelerator when you create a load balancer no Guia do desenvolvedor do AWS Global Accelerator .

11. Marcar e criar

    a. (Opcional) Adicione uma tag para caracterizar o balanceador de carga. As chaves de tag devem ser exclusivas de cada load balancer. Os caracteres permitidos são letras, espaços, números (em UTF-8) e os caracteres especiais a seguir: + - = . _ : / @. Não use espaços no início nem no fim. Os valores de tags diferenciam maiúsculas de minúsculas.

    b. Revise sua configuração e escolha Create load balancer (Criar um balanceador de carga). Alguns atributos padrão são aplicados ao balanceador de carga durante a criação. Você pode visualizá-los e editá-los depois de criar o balanceador de carga. Para obter mais informações, consulte Atributos do load balancer.

Etapa 4: testar o balanceador de carga

Depois de criar seu balanceador de carga, você pode verificar se suas EC2 instâncias são aprovadas na verificação de integridade inicial. Em seguida, você pode verificar se o balanceador de carga está enviando tráfego para sua EC2 instância. Para excluir o balanceador de carga, consulte Excluir um Application Load Balancer.

Para testar o balanceador de carga

1. Após a criação do load balancer, selecione Close (Fechar).
2. No painel de navegação, selecione Grupos de destino.
3. Selecione o grupo de destino recém-criado.
4. Escolha Destinos e verifique se a sua instância está pronta. Se o status de uma instância for `initial`, normalmente isso indica que a instância ainda está em processo de registro. Esse status também pode indicar que a instância não foi aprovada no número mínimo de verificações de integridade para ser considerada íntegra. Após o status de pelo menos uma instância ser íntegro, você pode testar seu load balancer. Para obter mais informações, consulte Status de integridade do destino.
5. No painel de navegação, selecione Load Balancers.
6. Selecione o load balancer recém-criado.

7. Escolha Descrição e copie o nome DNS do balanceador de carga interno ou voltado para a Internet (por exemplo, my-load-balancer -1234567890abcdef. elb.us-east-2.amazonaws.com).

   - Para balanceadores de carga voltados para a Internet, cole o nome de DNS no campo de endereço de um navegador da Web conectado à Internet.
   - Para balanceadores de carga internos, cole o nome de DNS no campo de endereço de um navegador da Web que tenha conectividade privada com a VPC.

   Se tudo estiver configurado corretamente, o navegador exibirá a página padrão do seu servidor.

8. Se a página da Web não for exibida, consulte os documentos a seguir para obter ajuda adicional na configuração e etapas de solução de problemas.

   - Para problemas relacionados a DNS, consulte Rotear tráfego para um balanceador de carga do ELB no Guia do desenvolvedor do Amazon Route 53.
   - Para problemas relacionados ao balanceador de carga, consulte Solucionar problemas em seus Application Load Balancers.

Atualizar zonas de disponibilidade para o Application Load Balancer

Você pode habilitar ou desabilitar as Zonas de disponibilidade para o seu load balancer a qualquer momento. Depois de habilitar uma Zona de disponibilidade, o load balancer começa a rotear as solicitações para os destinos registrados nessa Zona de disponibilidade. Os balanceadores de carga de aplicativos têm o balanceamento de carga entre zonas ativado por padrão, resultando no encaminhamento das solicitações para todos os destinos registrados em todas as zonas de disponibilidade. Quando o balanceamento de carga entre zonas está desativado, o balanceador de carga só encaminha a solicitação para destinos na mesma zona de disponibilidade. Para obter mais informações, consulte Balanceamento de carga entre zonas. O load balancer é mais eficaz se você garantir que cada Zona de disponibilidade ativada tenha pelo menos um destino registrado.

Depois de desativar uma Zona de disponibilidade, os destinos nela permanecerão registrados no load balancer, mas o load balancer não roteará solicitações para elas.

Para atualizar as Zonas de disponibilidade usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.

2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Mapeamento de rede, escolha Editar sub-redes.
5. Para habilitar uma zona de disponibilidade, marque a caixa de seleção e selecione uma sub-rede. Se houver apenas uma sub-rede disponível, ela será selecionada para você.
6. Para alterar a sub-rede de uma zona de disponibilidade habilitada, escolha uma das outras sub-redes na lista.
7. Para desabilitar uma zona de disponibilidade, desmarque a caixa de seleção.
8. Escolha Salvar alterações.

Para atualizar as zonas de disponibilidade usando o AWS CLI

Use o comando set-sub-redes.

Grupos de segurança para seu Application Load Balancer

O grupo de segurança do seu Application Load Balancer controla o tráfego que tem permissão para acessar e deixar o balanceador de carga. Você deve garantir que seu load balancer consiga se comunicar com destinos registrados tanto na porta do listener quanto na porta de verificação de integridade. Sempre que você adicionar um listener ao load balancer ou atualizar a porta de verificação de integridade de um grupo de destino usado pelo load balancer para rotear as solicitações, será necessário verificar se os security groups associados ao load balancer permitem tráfego na nova porta em ambas as direções. Caso não façam isso, você poderá editar as regras para os grupos de segurança associados na ocasião ou associar diferentes grupos de segurança ao balanceador de carga. É possível escolher as portas e os protocolos que deseja permitir. Por exemplo, você pode abrir conexões ICMP (Internet Control Message Protocol) para o load balancer responder às solicitações de ping (no entanto, as solicitações de ping não são encaminhadas a nenhuma instância).

Regras recomendadas

As regras a seguir são recomendadas para um balanceador de carga voltado para a Internet.

Inbound

Source	Port Range	Comment
0.0.0.0/0	*listener*	Permite todo o tráfego de entrada na porta do listener do load balancer

Outbound

Destination	Port Range	Comment
instance security group	*instance listener*	Permitir tráfego de saída para instâncias na porta do ouvinte da instância
instance security group	*health check*	Permitir tráfego de saída para instâncias na porta de verificação de integridade

A regras a seguir são recomendadas para um balanceador de carga interno.

Inbound

Source	Port Range	Comment
VPC CIDR	*listener*	Permite tráfego de entrada do CIDR da VPC na porta do listener do load balancer

Outbound

Destination	Port Range	Comment
instance security group	*instance listener*	Permitir tráfego de saída para instâncias na porta do ouvinte da instância
instance security group	*health check*	Permitir tráfego de saída para instâncias na porta de verificação de integridade

As regras a seguir são recomendadas para um Application Load Balancer usado como destino de um Network Load Balancer.

Inbound

Source	Port Range	Comment
client IP addresses/ CIDR	*alb listener*	Permitir todo o tráfego de entrada de cliente na porta do receptor do balanceador de carga.
VPC CIDR	*alb listener*	Permitir o tráfego de entrada do cliente pela porta do AWS PrivateLink ouvinte do balanceador de carga
VPC CIDR	*alb listener*	Permitir tráfego de entrada de integridade proveniente do Network Load Balancer

Outbound

Destination	Port Range	Comment
instance security group	*instance listener*	Permitir tráfego de saída para instâncias na porta do ouvinte da instância
instance security group	*health check*	Permitir tráfego de saída para instâncias na porta de verificaç ão de integridade

Observe que os grupos de segurança do Application Load Balancer usam o rastreamento da conexão para monitorar as informações sobre o tráfego proveniente do Network Load Balancer. Isso acontece independentemente das regras do grupo de segurança definidas para seu Application Load Balancer. Para saber mais sobre o rastreamento de EC2 conexão da Amazon, consulte Rastreamento de conexões de grupos de segurança no Guia EC2 do usuário da Amazon.

Para garantir que seus destinos recebam tráfego exclusivamente do balanceador de carga, restrinja os grupos de segurança associados aos seus destinos para aceitar tráfego somente do balanceador de carga. Isso pode ser feito definindo o grupo de segurança do balanceador de carga como a origem na regra de entrada do grupo de segurança do destino.

Recomendamos também que você permita a entrada de tráfego ICMP para oferecer suporte ao Path MTU Discovery. Para obter mais informações, consulte Path MTU Discovery no Guia do EC2 usuário da Amazon.

Atualizar os grupos de segurança associados

Você pode atualizar os security groups associados ao seu load balancer a qualquer momento.

Para atualizar security groups usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Segurança, escolha Editar.
5. Para associar um security group ao seu load balancer, selecione-o. Para remover uma associação de grupo de segurança, escolha o ícone X para o grupo de segurança.
6. Escolha Salvar alterações.

Para atualizar grupos de segurança usando o AWS CLI

Use o comando set-security-groups.

Atualizar os tipos de endereço IP para o Application Load Balancer

Você pode configurar seu Application Load Balancer para que os clientes possam se comunicar com o balanceador de carga usando somente IPv4 endereços ou usando endereços IPv4 e IPv6 endereços (pilha dupla). O balanceador de carga se comunica com os destinos com base no tipo de endereço IP do grupo de destino. Para obter mais informações, consulte Tipo de endereço IP.

Requisitos para dualstack

- É possível definir o tipo de endereço IP ao criar o load balancer e atualizá-lo a qualquer momento.

- A nuvem privada virtual (VPC) e as sub-redes que você especifica para o balanceador de carga devem ter blocos CIDR associados. IPv6 Para obter mais informações, consulte IPv6os endereços no Guia EC2 do usuário da Amazon.
- As tabelas de rotas das sub-redes do balanceador de carga devem rotear o tráfego. IPv6
- Os grupos de segurança do balanceador de carga devem permitir o IPv6 tráfego.
- A rede ACLs das sub-redes do balanceador de carga deve permitir tráfego. IPv6

Como definir o tipo de endereço IP na criação

Defina as configurações conforme descrito em Criar um balanceador de carga.

Para atualizar o tipo de endereço IP usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Mapeamento de rede, escolha Editar tipo de endereço IP.
5. Para o tipo de endereço IP, escolha IPv4oferecer suporte somente a IPv4 endereços, Dualstack para oferecer suporte a IPv6 endereços IPv4 e ou Dualstack sem público para oferecer suporte somente a endereços. IPv4 IPv6
6. Escolha Salvar alterações.

Para atualizar o tipo de endereço IP usando o AWS CLI

Use o comando set-ip-address-type.

Atualize os pools de endereços IP IPAM para seu Application Load Balancer

Os pools de endereços IP IPAM devem primeiro ser criados dentro do IPAM antes de poderem ser usados pelo Application Load Balancer. Para obter mais informações, consulte Traga seus endereços IP para o IPAM.

Para definir os pools de endereços IP IPAM na criação

Defina as configurações conforme descrito em Criar um balanceador de carga.

Para atualizar os pools de endereços IP IPAM usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Mapeamento de rede, escolha Editar pools de IP.
5. Em Pools de IP, ative Usar pool IPAM para IPv4 endereços públicos.
6. Em Pool IPv4 IPAM público, escolha o pool IPAM que você deseja usar.
7. Escolha Salvar alterações.

Para atualizar os pools de endereços IP IPAM usando o AWS CLI

Use o comando modify-ip-pools.

Editar atributos para o Application Load Balancer

Depois de criar um Application Load Balancer, você pode editar seus atributos.

Atributos do load balancer

- Tempo limite de inatividade da conexão
- Duração da manutenção de atividade do cliente HTTP
- Proteção contra exclusão
- Modo de mitigação de dessincronização
- Preservação de cabeçalho do host

Tempo limite de inatividade da conexão

O tempo limite de inatividade da conexão é o período em que uma conexão existente de cliente ou destino pode permanecer inativa, sem que nenhum dado seja enviado ou recebido, antes que o balanceador de carga feche a conexão.

Para garantir que operações demoradas, como uploads de arquivo, tenham tempo para serem concluídas; envie pelo menos 1 byte de dados antes de decorrer cada período de tempo limite de inatividade e aumente a duração do período do tempo limite de inatividade conforme o necessário.

Recomendamos também que você configure o tempo limite de inatividade do seu aplicativo como um valor maior do que o tempo limite de inatividade configurado para o load balancer. Caso contrário, se a aplicação fechar a conexão TCP com o balanceador de carga incorretamente, o balanceador de carga poderá enviar uma solicitação à aplicação antes de receber o pacote indicando que a conexão foi fechada. Se isso acontecer, o balanceador de carga enviará um erro HTTP 502 Gateway inadequado para o cliente.

Os Application Load Balancers não oferecem suporte a quadros PING HTTP/2. Eles não redefinem o tempo limite de inatividade da conexão.

Por padrão, o Elastic Load Balancing define como 60 segundos ou 1 minuto o valor do tempo limite de inatividade para o balanceador de carga. Use o procedimento a seguir para definir um valor de tempo limite ocioso diferente.

Para atualizar o valor do tempo limite de inatividade da conexão usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Atributos, escolha Editar.
5. Em Configuração de tráfego, insira um valor para Tempo limite de inatividade da conexão. O intervalo válido é de 1 a 4.000 segundos.
6. Escolha Salvar alterações.

Para atualizar o valor do tempo limite de inatividade usando o AWS CLI

Use o comando modify-load-balancer-attributes com o atributo `idle_timeout.timeout_seconds`.

Duração da manutenção de atividade do cliente HTTP

A duração da manutenção de atividade do cliente HTTP é o tempo máximo em que um Application Load Balancer mantém uma conexão HTTP persistente com um cliente. Depois de decorrida a duração da manutenção de atividade do cliente HTTP configurado, o Application Load Balancer aceita mais uma solicitação e retorna uma resposta que fecha a conexão normalmente.

O tipo de resposta enviada pelo balanceador de carga depende da versão HTTP usada pela conexão do cliente.

- Para clientes conectados usando HTTP 1.x, o balanceador de carga envia um cabeçalho HTTP contendo o campo `Connection: close`.
- Para clientes conectados usando HTTP/2, o balanceador de carga envia um quadro `GOAWAY`.

Por padrão, o Application Load Balancer define o valor da duração da manutenção de atividade do cliente HTTP para balanceadores de carga como 3.600 segundos ou 1 hora. A duração da manutenção de atividade do cliente HTTP não pode ser desativada ou definida abaixo do mínimo de 60 segundos, mas você pode aumentar a duração da manutenção de atividade do cliente HTTP até um máximo de 604.800 segundos ou 7 dias. Um Application Load Balancer inicia o período de duração da manutenção de atividade do cliente HTTP quando uma conexão HTTP com um cliente é estabelecida inicialmente. O período de duração continua quando não há tráfego e não é reiniciado até que uma nova conexão seja estabelecida.

Quando o tráfego do balanceador de carga é deslocado de uma zona de disponibilidade prejudicada usando mudança de zona ou mudança automática de zona, os clientes com conexões abertas existentes podem continuar fazendo solicitações no local danificado até que os clientes se reconectem. Para oferecer suporte a uma recuperação mais rápida, considere definir um valor menor de duração da manutenção de atividade para limitar a quantidade de tempo que os clientes permanecem conectados a um balanceador de carga. Para obter mais informações, consulte Limit the time that clients stay connected to your endpoints no Guia do desenvolvedor do Amazon Application Recovery Controller (ARC).

> **Note**
>
> Quando o balanceador de carga muda o tipo de endereço IP do Application Load Balancer para `dualstack-without-public-ipv4`, o balanceador de carga espera que todas as conexões ativas sejam concluídas. Para diminuir o tempo necessário para trocar o tipo de endereço IP do seu Application Load Balancer, considere reduzir a duração do keepalive do cliente HTTP.

O Application Load Balancer atribui o valor de duração da manutenção de atividade do cliente HTTP durante a conexão inicial. Quando você atualiza a duração da manutenção de atividade do cliente HTTP, isso pode resultar em conexões simultâneas com diferentes valores de duração da manutenção de atividade do cliente HTTP. As conexões existentes mantêm o valor de duração da manutenção de atividade do cliente HTTP aplicado durante a conexão inicial. Novas conexões recebem o valor atualizado de duração da manutenção de atividade do cliente HTTP.

Para atualizar o valor de duração da manutenção de atividade do cliente usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Atributos, escolha Editar.
5. Em Configuração de tráfego, insira um valor para a Duração da manutenção de atividade do cliente HTTP. O intervalo válido é de 60 a 604.800 segundos.
6. Escolha Salvar alterações.

Para atualizar o valor da duração do keepalive do cliente usando o AWS CLI

Use o comando modify-load-balancer-attributes com o atributo `client_keep_alive.seconds`.

Proteção contra exclusão

Para evitar que seu load balancer seja excluído acidentalmente, é possível ativar a proteção contra exclusão. Por padrão, a proteção contra exclusão está desativada para seu load balancer.

Se você ativar a proteção contra exclusão para o load balancer, deverá desativá-la antes de excluir o load balancer.

Para habilitar a proteção contra exclusão usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Atributos, escolha Editar.
5. Em Configuração, ative a Proteção contra exclusão..
6. Escolha Salvar alterações.

Para desabilitar a proteção contra exclusão usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.

3. Selecione o load balancer.
4. Na guia Atributos, escolha Editar.
5. Na página Configuração, ative a Proteção contra exclusão.
6. Escolha Salvar alterações.

Para ativar ou desativar a proteção contra exclusão usando o AWS CLI

Use o comando modify-load-balancer-attributes com o atributo `deletion_protection.enabled`.

Modo de mitigação de dessincronização

O modo de mitigação de dessincronização protege sua aplicação contra problemas causados por dessincronização de HTTP. O balanceador de carga classifica cada solicitação com base em seu nível de ameaça, permite solicitações seguras e, em seguida, reduz o risco, conforme instruído pelo modo de mitigação especificado. Os modos de mitigação de dessincronização são: monitor (monitorado), defensive (defensivo) e strictest (mais rigoroso). O padrão é o modo defensivo, que fornece mitigação durável contra HTTP Desync, mantendo a disponibilidade da sua aplicação. Você pode alternar para o modo mais restrito a fim de garantir que sua aplicação receba somente solicitações que estejam em conformidade com a RFC 7230.

A biblioteca http_desync_guardian analisa solicitações HTTP para prevenir ataques de dessincronização de HTTP. Para obter mais informações, consulte HTTP Desync Guardian em. GitHub

Classificações

As classificações são as seguintes:

- Compatível: a solicitação está em conformidade com o RFC 7230 e não representa ameaças de segurança conhecidas.
- Aceitável: a solicitação não está em conformidade com o RFC 7230, mas não representa ameaças de segurança conhecidas.
- Ambígua: a solicitação não está em conformidade com o RFC 7230, mas representa um risco, pois vários servidores Web e proxies podem lidar com ela de formas diferentes.
- Grave: a solicitação representa um alto risco de segurança. O balanceador de carga bloqueia a solicitação, atende uma resposta 400 ao cliente e fecha a conexão do cliente.

Se uma solicitação não estiver em conformidade com o RFC 7230, o balanceador de carga incrementará a métrica `DesyncMitigationMode_NonCompliant_Request_Count`. Para obter mais informações, consulte Métricas do Application Load Balancer.

A classificação de cada solicitação está incluída nos logs de acesso do balanceador de carga. Se a solicitação não estiver em conformidade, os logs de acesso incluirão um código de motivo de classificação. Para obter mais informações, consulte Motivos de classificação.

Modos

A tabela a seguir descreve como os Application Load Balancers processam solicitações com base no modo e na classificação.

Classificação	Modo monitorado	Modo defensivo	Modo mais restrito
Compatível	Permitido	Permitido	Permitido
Aceitável	Permitido	Permitido	Bloqueado
Ambíguo	Permitido	Permitido[1]	Bloqueado
Grave	Permitido	Bloqueado	Bloqueado

[1] Encaminha as solicitações, mas fecha as conexões entre cliente e destino. Você poderá incorrer em cobranças adicionais se seu balanceador de carga receber um grande número de solicitações ambíguas no modo Defensivo. Isso ocorre porque o aumento do número de novas conexões por segundo contribui para as Load Balancer Capacity Units (LCU – Unidades de capacidade do balanceador de carga) usadas por hora. Você pode usar a métrica `NewConnectionCount` para comparar como seu balanceador de carga estabelece novas conexões no modo Monitorar e no modo Defensivo.

Para atualizar o modo de mitigação de dessincronização usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Atributos, escolha Editar.

5. Em Tratamento de pacotes, para o Modo de mitigação de dessincronização, escolha Defensivo, Mais rigoroso ou Monitorar.
6. Escolha Salvar alterações.

Para atualizar o modo de mitigação de dessincronização usando o AWS CLI

Use o modify-load-balancer-attributescomando com o `routing.http.desync_mitigation_mode` atributo definido como `monitordefensive`, `oustrictest`.

Preservação de cabeçalho do host

Quando você habilitar o atributo Preservar cabeçalho do host, o Application Load Balancer vai preservar o cabeçalho `Host` na solicitação HTTP e enviá-lo para o destino sem nenhuma modificação. Se o Application Load Balancer receber vários cabeçalhos `Host`, ele preservará todos eles. As regras do receptor são aplicadas somente ao primeiro cabeçalho `Host` recebido.

Por padrão, quando o atributo Preservar cabeçalho do host não estiver habilitado, o Application Load Balancer modificará o cabeçalho `Host` da seguinte maneira:

Quando a preservação de cabeçalho do host não estiver habilitada e a porta do receptor for uma porta não padrão: quando não estiver usando as portas padrão (portas 80 ou 443), anexaremos o número da porta ao cabeçalho do host, caso ele ainda não tenha sido anexado pelo cliente. Por exemplo, o cabeçalho `Host` na solicitação HTTP com `Host: www.example.com` seria modificado para `Host: www.example.com:8080` se a porta do receptor fosse uma porta não padrão, como `8080`.

Quando a preservação de cabeçalho do host não estiver habilitada e a porta do receptor for uma porta padrão (porta 80 ou 443): para portas padrão do receptor (porta 80 ou 443), não anexamos o número da porta ao cabeçalho do host de saída. Qualquer número de porta que já esteja no cabeçalho do host de entrada será removido.

A tabela a seguir mostra mais exemplos de como os Application Load Balancers processam os cabeçalhos do host na solicitação HTTP com base na porta do receptor.

Porta do receptor	Exemplo de solicitação	Cabeçalho do host na solicitação	Preservação de cabeçalho do host desabilitada (comportamento padrão)	Preservação de cabeçalho do host habilitada
A solicitação é enviada no HTTP/HTTPS ouvinte padrão.	`GET / index.html HTTP/1.1 Host: example.com`	example.com	example.com	example.com
A solicitação é enviada no receptor HTTP padrão e o cabeçalho do host tem uma porta (por exemplo, 80 ou 443).	`GET / index.html HTTP/1.1 Host: example.com:80`	example.com:80	example.com	example.com:80
A solicitação tem um caminho absoluto.	`GET https://dns_name/index.html HTTP/1.1 Host: example.com`	example.com	dns_name	example.com
A solicitação é enviada em uma porta de receptor não padrão (por exemplo, 8080)	`GET / index.html HTTP/1.1 Host: example.com`	example.com	example.com:8080	example.com
A solicitação é enviada em uma	`GET / index.ht`	example.com:8080	example.com:8080	example.com:8080

Porta do receptor	Exemplo de solicitação	Cabeçalho do host na solicitação	Preservação de cabeçalho do host desabilitada (comportamento padrão)	Preservação de cabeçalho do host habilitada
porta de receptor não padrão e o cabeçalho do host tem uma porta (por exemplo, 8080).	`ml HTTP/1.1 Host: example.com:8080`			

Para habilitar a preservação de cabeçalho do host usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Load Balancers.
3. Selecione o load balancer.
4. Na guia Atributos, escolha Editar.
5. Em Tratamento de pacotes, ative Preservar cabeçalho do host.
6. Escolha Salvar alterações.

Para habilitar a preservação do cabeçalho do host usando o AWS CLI

Use o modify-load-balancer-attributescomando com o `routing.http.preserve_host_header.enabled` atributo definido como`true`.

Marcar um Application Load Balancer

As tags ajudam a categorizar seus load balancers de diferentes formas, como por finalidade, por proprietário ou por ambiente.

Você pode adicionar várias tags para cada load balancer. Se você adicionar uma tag com uma chave que já esteja associada ao load balancer, o valor dessa tag será atualizado.

Quando você terminar com uma tag, poderá removê-la do seu load balancer.

Restrições

- Número máximo de tags por recurso: 50
- Comprimento máximo da chave: 127 caracteres Unicode
- Comprimento máximo de valor: 255 caracteres Unicode
- As chaves e valores das tags diferenciam maiúsculas de minúsculas. Os caracteres permitidos são letras, espaços e números representáveis em UTF-8, além dos seguintes caracteres especiais: + - = . _ : / @. Não use espaços no início nem no fim.
- Não use o `aws:` prefixo nos nomes ou valores de suas tags porque ele está reservado para AWS uso. Você não pode editar nem excluir nomes ou valores de tag com esse prefixo. As tags com esse prefixo não contam para as tags por limite de recurso.

Para atualizar as tags para um load balancer usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Tags, selecione Gerenciar tags e execute uma ou mais das ações a seguir:
   a. Para atualizar uma tag, edite os valores de Chave e Valor.
   b. Para adicionar uma nova tag, selecione Adicionar tag e, em seguida, insira os valores para Chave e Valor.
   c. Para excluir uma etiqueta, escolha Remove (Remover) ao lado dela.
5. Ao concluir a atualização de tags, selecione Salvar alterações.

Para atualizar as tags de um balanceador de carga usando o AWS CLI

Use os comandos add-tags e remove-tags.

Excluir um Application Load Balancer

Assim que o load balancer é disponibilizado, você será cobrado por cada hora ou hora parcial em que mantê-lo em execução. Quando não precisar mais do load balancer, pode excluí-lo. Assim que o load balancer for excluído, a cobrança será interrompida.

Você não pode excluir um load balancer se a proteção contra exclusão estiver habilitada. Para obter mais informações, consulte Proteção contra exclusão.

Observe que excluir um load balancer não afeta seus destinos registrados. Por exemplo, suas EC2 instâncias continuam em execução e ainda estão registradas em seus grupos-alvo. Para excluir seus grupos de destino, consulte Excluir um grupo de destino do Application Load Balancer.

Para excluir um load balancer usando o console

1. Se você tiver um registro DNS para seu domínio que aponte para o balanceador de carga, aponte-o para um novo local e aguarde até que a mudança de DNS entre em vigor antes de excluir o balanceador de carga.

   Exemplo:

   - Se o registro for um registro CNAME com Time-To-Live (TTL) de 300 segundos, aguarde pelo menos 300 segundos antes de seguir para a próxima etapa.
   - Se o registro for um registro de alias (A) do Route 53, aguarde pelo menos 60 segundos.
   - Se você estiver usando o Route 53, a alteração do registro levará 60 segundos para se propagar para todos os servidores globais de nome do Route 53. Adicione esse tempo ao valor do TTL do registro que está sendo atualizado.
2. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
3. No painel de navegação, selecione Balanceador de carga.
4. Selecione o balanceador de carga e, em seguida, Ações e Excluir balanceador de carga.
5. Quando a confirmação for solicitada, insira **`confirm`** e selecione Excluir.

Para excluir um balanceador de carga usando o AWS CLI

Use o comando delete-load-balancer.

Exibir o mapa de recursos do Application Load Balancer

O mapa de recursos do Application Load Balancer fornece uma exibição interativa da arquitetura do balanceador de carga, incluindo seus receptores, regras, grupos de destino e destinos associados. O mapa de recursos também destaca os relacionamentos e os caminhos de roteamento entre todos os recursos, produzindo uma representação visual da configuração do balanceador de carga.

Para exibir o mapa de recursos do Application Load Balancer usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Escolha a guia Mapa de recursos para exibir o mapa de recursos do balanceador de carga.

Componentes do mapa de recursos

Exibições do mapa

Há duas exibições disponíveis no mapa de recursos do Application Load Balancer: Visão geral e Mapa de destino não íntegro. A opção Visão geral é selecionada por padrão e exibe todos os recursos do balanceador de carga. Selecionar a opção Mapa de destino não íntegro exibirá somente os destinos não íntegros e os recursos associados a eles.

A exibição Mapa de destino não íntegro pode ser usada para solucionar problemas de destinos que estão falhando nas verificações de integridade. Para obter mais informações, consulte Solucionar problemas de destinos não íntegros usando o mapa de recursos.

Grupos de recursos

O mapa de recursos do Application Load Balancer contém quatro grupos de recursos, um para cada tipo de recurso. Os grupos de recursos são Receptores, Regras, Grupos de destino e Destinos.

Blocos de recursos

Cada recurso dentro de um grupo tem seu próprio bloco, que exibe detalhes sobre esse recurso específico.

- Passar o mouse sobre um bloco de recursos destaca os relacionamentos entre ele e outros recursos.
- Selecionar um bloco de recursos destaca as relações entre ele e outros recursos e exibe detalhes adicionais sobre esse recurso.
  - condições da regra: as condições de cada regra.
  - resumo de integridade do grupo de destino: o número de destinos registrados para cada estado de integridade.
  - status de integridade do destino: o status de integridade atual e a descrição do destino.

> **Note**
>
> Você pode desativar Mostrar detalhes do recurso para ocultar detalhes adicionais no mapa do recurso.

- Cada bloco de recursos contém um link que, quando selecionado, navega até a página de detalhes desse recurso.
  - Receptores: selecione protocol:port dos receptores. Por exemplo, `HTTP:80`.
  - Regras: selecione a ação das regras. Por exemplo, `Forward to target group`.
  - Grupos de destino: selecione o nome do grupo de destino. Por exemplo, `my-target-group`.
  - Destinos: selecione o ID dos destinos. Por exemplo, `i-1234567890abcdef0`.

Exportar o mapa de recursos

Selecionar Exportar oferece a opção de exportar a exibição atual do mapa de recursos do Application Load Balancer como PDF.

Mudança de zona para o Application Load Balancer

O deslocamento zonal e o deslocamento automático zonal são recursos do Amazon Application Recovery Controller (ARC). Com a mudança zonal, você pode afastar o tráfego de uma zona de disponibilidade prejudicada com uma única ação. Dessa forma, é possível continuar a operar em outras zonas de disponibilidade íntegras em uma Região da AWS.

Com o deslocamento automático zonal, você AWS autoriza a transferência do tráfego de recursos de um aplicativo de uma zona de disponibilidade durante eventos, em seu nome, para ajudar a reduzir o tempo de recuperação. AWS inicia uma mudança automática quando o monitoramento interno indica que há uma deficiência na zona de disponibilidade que pode afetar potencialmente os clientes. Quando AWS inicia um deslocamento automático, o tráfego do aplicativo para os recursos que você configurou para o deslocamento automático zonal começa a se afastar da Zona de Disponibilidade.

Quando você inicia uma mudança de zona, o balanceador de carga para de enviar novo tráfego do recurso para a zona de disponibilidade afetada. O ARC cria a mudança de zona imediatamente. Porém, pode demorar um pouco para que as conexões existentes e em andamento na zona de disponibilidade sejam concluídas, dependendo do comportamento do cliente e da reutilização da conexão. Dependendo das configurações de DNS e de outros fatores, as conexões existentes

podem ser concluídas em apenas alguns minutos ou levar mais tempo. Para obter mais informações, consulte Limit the time that clients stay connected to your endpoints no Guia do desenvolvedor do Amazon Application Recovery Controller (ARC).

Conteúdo

- Antes de começar uma mudança zonal
- Balanceamento de carga entre zonas
- Substituição administrativa de mudança de zona
- Ative a mudança de zona para seu Application Load Balancer
- Inicie uma mudança de zona para seu Application Load Balancer
- Atualize uma mudança de zona para seu Application Load Balancer
- Cancelar uma mudança de zona para seu Application Load Balancer

Antes de começar uma mudança zonal

- A mudança de zona está desativada por padrão e deve ser ativada em cada Application Load Balancer. Para obter mais informações, consulte Ative a mudança de zona para seu Application Load Balancer.
- Você pode iniciar uma mudança de zona para um balanceador de carga específico somente para uma única zona de disponibilidade. Você não pode iniciar uma mudança de zona para várias zonas de disponibilidade.
- AWS remove proativamente os endereços IP do balanceador de carga zonal do DNS quando vários problemas de infraestrutura afetam os serviços. Antes de iniciar uma mudança de zona, sempre verifique a capacidade atual da zona de disponibilidade. Se os balanceadores de carga estiverem com o balanceamento de carga entre zonas desativado e você usar uma mudança de zona para remover o endereço IP de um balanceador de carga de zona, a zona de disponibilidade afetada pela mudança de zona também perderá a capacidade de destino.

Para obter mais informações, consulte as melhores práticas para mudanças zonais no ARC no Guia do desenvolvedor do Amazon Application Recovery Controller (ARC).

Balanceamento de carga entre zonas

Quando uma mudança de zona é iniciada em um Application Load Balancer com o balanceamento de carga entre zonas ativado, todo o tráfego para destinos é bloqueado na zona de disponibilidade afetada e os endereços IP zonais são removidos do DNS.

Benefícios:

- Recuperação mais rápida de falhas na zona de disponibilidade.
- A capacidade de mover o tráfego para uma zona de disponibilidade saudável se forem detectadas falhas em uma zona de disponibilidade.
- Você pode testar a integridade do aplicativo simulando e identificando falhas para evitar tempo de inatividade não planejado.

Substituição administrativa de mudança de zona

Os destinos que pertencem a um Application Load Balancer incluem um novo status`AdministrativeOverride`, que é independente do `TargetHealth` estado.

Quando uma mudança de zona é iniciada para um Application Load Balancer, todos os alvos dentro da zona da qual está sendo deslocado são considerados administrativamente substituídos. O Application Load Balancer interrompe o roteamento de novos tráfegos para alvos substituídos administrativamente. As conexões existentes permanecem intactas até serem fechadas organicamente.

Os estados `AdministrativeOverride` possíveis são:

unknown

O estado não pode ser propagado devido a um erro interno

no_override

Nenhuma substituição está ativa no momento no destino

zonal_shift_active

A mudança de zona está ativa na zona de disponibilidade de destino

Ative a mudança de zona para seu Application Load Balancer

A mudança de zona está desativada por padrão e deve ser ativada em cada Application Load Balancer. Isso garante que você possa iniciar uma mudança de zona usando somente os Application Load Balancers específicos que você deseja. Para obter mais informações, consulte the section called “Mudança de zona”.

Para habilitar uma mudança de zona usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Load balancers (Balanceadores de carga).
3. Selecione o Application Load Balancer.
4. Na guia Atributos, escolha Editar.
5. Em Configuração de roteamento da Zona de Disponibilidade, para integração de deslocamento zonal ARC, escolha Habilitar.
6. Escolha Salvar alterações.

Para habilitar o deslocamento zonal usando o AWS CLI

Use o comando modify-load-balancer-attributes com o atributo `zonal_shift.config.enabled`.

Inicie uma mudança de zona para seu Application Load Balancer

As etapas deste procedimento explicam como iniciar uma mudança de zona usando o EC2 console da Amazon. Para verificar as etapas de como iniciar uma mudança de zona usando o console do ARC, consulte Starting a zonal shift no Guia do desenvolvedor do Amazon Application Recovery Controller (ARC).

Pré-requisito

Antes de começar, verifique se você ativou a mudança de zona para o balanceador de carga.

Para iniciar uma mudança de zona usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Load balancers (Balanceadores de carga).

3. Selecione o Application Load Balancer.
4. Na guia Integrações, em Amazon Application Recovery Controller (ARC), escolha Iniciar mudança zonal.
5. Selecione a zona de disponibilidade da qual você deseja remover o tráfego.
6. Escolha ou insira uma data de validade para a mudança de zona. Inicialmente, uma mudança de zona pode ser definida entre 1 minuto e 3 dias (72 horas).

   Todas as mudanças de zona são temporárias. Você deve definir uma validade, mas pode atualizar mudanças ativas posteriormente para definir uma nova validade.
7. Insira um comentário. Você pode atualizar a mudança de zona posteriormente para editar o comentário, se quiser.
8. Marque a caixa de seleção para reconhecer que iniciar uma mudança de zona reduz a capacidade do seu aplicativo ao afastar o tráfego da zona de disponibilidade.
9. Escolha Confirmar.

Para iniciar uma mudança zonal usando o AWS CLI

Use o start-zonal-shiftcomando Amazon Application Recovery Controller (ARC).

Atualize uma mudança de zona para seu Application Load Balancer

As etapas deste procedimento explicam como atualizar uma mudança de zona usando o EC2 console da Amazon. Para verificar as etapas de como atualizar uma mudança de zona usando o console do Amazon Application Recovery Controller (ARC), consulte Updating a zonal shift no Guia do desenvolvedor do Amazon Application Recovery Controller (ARC).

Atualizar uma mudança de zona usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Load balancers (Balanceadores de carga).
3. Selecione um Application Load Balancer com uma mudança de zona ativa.
4. Na guia Integrações, em Amazon Application Recovery Controller (ARC), escolha Atualizar mudança zonal.

   Isso abre o console ARC para continuar o processo de atualização.
5. Em Definir expiração da mudança de zona, opcionalmente, selecione ou insira uma expiração.

6. Em Comentário, opcionalmente, edite o comentário existente ou insira um novo.
7. Selecione Atualizar.

Para atualizar um deslocamento zonal usando o AWS CLI

Use o update-zonal-shiftcomando Amazon Application Recovery Controller (ARC).

Cancelar uma mudança de zona para seu Application Load Balancer

As etapas deste procedimento explicam como cancelar uma mudança de zona usando o EC2 console da Amazon. Para verificar as etapas de como cancelar uma mudança de zona usando o console do Amazon Application Recovery Controller (ARC), consulte Canceling a zonal shift no Guia do desenvolvedor do Amazon Application Recovery Controller (ARC).

Para cancelar uma mudança de zona usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Load balancers (Balanceadores de carga).
3. Selecione um Application Load Balancer com uma mudança de zona ativa.
4. Na guia Integrações, em Amazon Application Recovery Controller (ARC), escolha Cancelar mudança zonal.

   Isso abre o console ARC para continuar o processo de cancelamento.
5. Escolha Cancelar mudança de zona.
6. Quando a confirmação for solicitada, escolha Confirmar.

Para cancelar uma mudança de zona usando o AWS CLI

Use o cancel-zonal-shiftcomando Amazon Application Recovery Controller (ARC).

Reservas de capacidade para seu Application Load Balancer

As reservas da Unidade de Capacidade do Balanceador de Carga (LCU) permitem que você reserve uma capacidade estática mínima para seu balanceador de carga. Os Application Load Balancers escalam automaticamente para suportar cargas de trabalho detectadas e atender às necessidades de capacidade. Quando a capacidade mínima é configurada, seu balanceador de carga continua

aumentando ou diminuindo com base no tráfego recebido, mas também evita que a capacidade fique abaixo da capacidade mínima configurada.

Considere usar a reserva da LCU nas seguintes situações:

- Você tem um próximo evento que terá um tráfego repentino e incomum e deseja garantir que seu balanceador de carga possa suportar o aumento repentino de tráfego durante o evento.
- Você tem picos de tráfego imprevisíveis devido à natureza da sua carga de trabalho por um curto período.
- Você está configurando seu balanceador de carga para integrar ou migrar seus serviços em um horário de início específico e precisa começar com uma alta capacidade em vez de esperar que o auto-scaling entre em vigor.
- Você precisa manter uma capacidade mínima para atender aos contratos de nível de serviço ou aos requisitos de conformidade.
- Você está migrando cargas de trabalho entre balanceadores de carga e deseja configurar o destino de acordo com a escala da origem.

Estime a capacidade que você precisa

Ao determinar a quantidade de capacidade que você deve reservar para seu balanceador de carga, recomendamos realizar testes de carga ou revisar dados históricos da carga de trabalho que representem o tráfego futuro que você espera. Usando o console do Elastic Load Balancing, você pode estimar quanta capacidade precisa reservar com base no tráfego analisado.

Como alternativa, você pode utilizar a CloudWatch métrica `PeakLCUs` para determinar o nível de capacidade necessário. A `PeakLCUs` métrica considera os picos em seu padrão de tráfego que o balanceador de carga deve escalar em todas as dimensões de escalabilidade para suportar sua carga de trabalho. A `PeakLCUs` métrica é diferente da `ConsumedLCUs` métrica, que agrega apenas as dimensões de faturamento do seu tráfego. É recomendável usar a `PeakLCUs` métrica para garantir que sua reserva de LCU seja adequada durante o escalonamento do balanceador de carga. Ao estimar a capacidade, use um por minuto `Sum` de. `PeakLCUs`

Se você não tiver dados históricos da carga de trabalho para referenciar e não puder realizar o teste de carga, poderá estimar a capacidade necessária usando a calculadora de reservas da LCU. A calculadora de reservas da LCU usa dados com base nas cargas de trabalho históricas AWS observadas e pode não representar sua carga de trabalho específica. Para obter mais informações, consulte Calculadora de reserva de unidades de capacidade do Load Balancer.

Cotas para reservas de LCU

Sua conta tem cotas relacionadas a. LCUs Para obter mais informações, consulte the section called “Unidades de capacidade do Load Balancer”.

Solicitar reserva de unidade de capacidade do balanceador de carga para seu Application Load Balancer

Antes de usar a reserva da LCU, analise o seguinte:

- A capacidade é reservada em nível regional e distribuída uniformemente nas zonas de disponibilidade. Confirme se você tem metas distribuídas uniformemente suficientes em cada zona de disponibilidade antes de ativar a reserva de LCU.
- As solicitações de reserva da LCU são atendidas por ordem de chegada e dependem da capacidade disponível para uma zona naquele momento. A maioria das solicitações geralmente é atendida em alguns minutos, mas pode levar até algumas horas.
- Para atualizar uma reserva existente, a solicitação anterior deve ser provisionada ou falhar. Você pode aumentar a capacidade reservada quantas vezes precisar, mas só pode diminuir a capacidade reservada duas vezes por dia.
- Você continuará incorrendo em cobranças por qualquer capacidade reservada ou provisionada até que ela seja encerrada ou cancelada.

Solicite uma reserva de LCU

As etapas deste procedimento explicam como solicitar uma reserva de LCU em seu balanceador de carga.

Para solicitar uma reserva de LCU usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o nome do balanceador de carga.
4. Na guia Capacidade, escolha Editar reserva da LCU.
5. Selecione Estimativa baseada em referência histórica e, em seguida, selecione o balanceador de carga na lista suspensa.
6. Selecione o período de referência para ver o nível recomendado de LCU reservado.

7. Se você não tiver uma carga de trabalho de referência histórica, poderá escolher Estimativa manual e inserir o número de LCUs a serem reservadas.
8. Escolha Salvar.

Para solicitar uma reserva de LCU usando AWS CLI

Use o comando modify-capacity-reservation.

Atualize ou encerre as reservas da unidade de capacidade do balanceador de carga para seu Application Load Balancer

Atualizar ou encerrar uma reserva de LCU

As etapas deste procedimento explicam como atualizar ou encerrar uma reserva de LCU em seu balanceador de carga.

Para atualizar ou encerrar uma reserva de LCU usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o nome do balanceador de carga.
4. Na guia Capacidade, confirme se o status da reserva é Provisionado.
    a. Para atualizar a reserva da LCU, escolha Editar reserva da LCU.
    b. Para encerrar a reserva da LCU, escolha Cancelar capacidade.

Para atualizar ou encerrar uma reserva de LCU usando o AWS CLI

Use o comando modify-capacity-reservation.

Monitore a reserva da unidade de capacidade do Load Balancer para seu Application Load Balancer

Status da reserva

A reserva da LCU tem quatro status disponíveis:

- pendente - Indica a reserva que está em processo de provisionamento.

- provisionado - Indica que a capacidade reservada está pronta e disponível para uso.
- falhou - Indica que a solicitação não pode ser concluída no momento.
- rebalanceamento - Indica que uma zona de disponibilidade foi adicionada ou removida e o balanceador de carga está reequilibrando a capacidade.

LCU reservada

A `ReservedLCUs` métrica é relatada por minuto. A capacidade é reservada por hora. Por exemplo, se você tiver uma reserva de LCU de 6.000, o total de uma hora é de 6.000 e o total de um minuto `ReservedLCUs` é de 100. Para determinar sua utilização reservada da LCU, consulte a métrica. `PeakLCUs` Você pode definir CloudWatch alarmes para comparar o valor por minuto com o valor `Sum` da `PeakLCUs` capacidade reservada, ou o valor por hora `Sum` de`ReservedLCUs`, para determinar se você reservou capacidade suficiente para atender às suas necessidades.

Monitore a capacidade reservada

As etapas desse processo explicam como verificar o status de uma reserva de LCU no seu balanceador de carga.

Para visualizar o status de uma reserva de LCU usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o nome do balanceador de carga.
4. Na guia Capacidade, você pode ver o status da reserva e o valor da LCU reservada.

Para monitorar o status da reserva da LCU usando AWS CLI

Use o comando describe-capacity-reservation.

Integrações para seu Application Load Balancer

Você pode otimizar sua arquitetura do Application Load Balancer integrando-se a vários outros AWS serviços para aprimorar o desempenho, a segurança e a disponibilidade do seu aplicativo.

Integrações de balanceadores de carga

- Amazon CloudFront + AWS WAF
- AWS Global Accelerator
- AWS Config
- AWS WAF

Amazon CloudFront + AWS WAF

CloudFront A Amazon é um serviço web que ajuda a melhorar o desempenho, a disponibilidade e a segurança dos aplicativos que você usa AWS. CloudFront atua como um ponto de entrada único e distribuído para seus aplicativos web que usam balanceadores de carga de aplicativos. Ele amplia o alcance global do seu Application Load Balancer, permitindo que ele atenda usuários de forma eficiente a partir de pontos de presença próximos, otimizando a entrega de conteúdo e reduzindo a latência para usuários em todo o mundo. O armazenamento automático de conteúdo nesses locais periféricos reduz significativamente a carga no Application Load Balancer, melhorando seu desempenho e escalabilidade.

A integração com um clique disponível no console do Elastic Load Balancing cria CloudFront uma distribuição com as proteções de segurança AWS WAF recomendadas e a associa ao seu Application Load Balancer. As AWS WAF proteções bloqueiam contra explorações comuns da web antes de chegarem ao seu balanceador de carga. Você pode acessar a CloudFront distribuição e seu painel de segurança correspondente na guia Integrações do balanceador de carga no console. Para obter mais informações, consulte Gerenciar proteções AWS WAF de segurança no painel de CloudFront segurança no Amazon CloudFront Developer Guide e Introducing CloudFront Security Dashboard, uma experiência unificada de CDN e segurança em aws.amazon.com/blogs.

Como prática recomendada de segurança, configure os grupos de segurança do Application Load Balancer voltado para a Internet para permitir tráfego de entrada somente da lista de prefixos AWS gerenciada para CloudFront e remover quaisquer outras regras de entrada. Para obter mais informações, consulte Usar a lista de prefixos CloudFront gerenciados, Configurar CloudFront para adicionar um cabeçalho HTTP personalizado às solicitações e Configurar um Application Load Balancer para encaminhar somente solicitações que contenham um cabeçalho específico no CloudFront Amazon Developer Guide >.

> **Note**
>
> CloudFront só oferece suporte a certificados ACM na região us-east-1 do Leste dos EUA (Norte da Virgínia). Se o Application Load Balancer tiver um ouvinte HTTPS configurado com

um certificado ACM em uma região diferente de us-east-1, você precisará alterar a conexão de CloudFront origem de HTTPS para HTTP ou provisionar um certificado ACM na região Leste dos EUA (Norte da Virgínia) e anexá-lo à sua distribuição. CloudFront

AWS Global Accelerator

Para otimizar a disponibilidade, o desempenho e a segurança dos aplicativos, crie um acelerador para seu balanceador de carga. O acelerador direciona o tráfego pela rede AWS global para endereços IP estáticos que servem como endpoints fixos na região mais próxima do cliente. AWS Global Accelerator é protegido pelo Shield Standard, que minimiza o tempo de inatividade do aplicativo e a latência dos ataques S. DDo

Para obter mais informações, consulte Adicionar um acelerador ao criar um balanceador de carga no Guia do AWS Global Accelerator desenvolvedor.

AWS Config

Para otimizar o monitoramento e a conformidade do seu balanceador de carga, configure. AWS Config AWS Config fornece uma visão detalhada da configuração dos AWS recursos em sua AWS conta. Isso inclui como os recursos estão relacionados entre si e como foram configurados no passado, para que você possa ver como as configurações e os relacionamentos mudam com o tempo. AWS Config simplifica as auditorias, a conformidade e a solução de problemas.

Para obter mais informações, consulte O que é AWS Config? no Guia do AWS Config desenvolvedor.

AWS WAF

Você pode usar AWS WAF com seu Application Load Balancer para permitir ou bloquear solicitações com base nas regras de uma lista de controle de acesso à web (web ACL).

Por padrão, se o balanceador de carga não conseguir obter uma resposta AWS WAF, ele retornará um erro HTTP 500 e não encaminhará a solicitação. Se você precisar que seu balanceador de carga encaminhe solicitações aos destinos, mesmo que ele não consiga entrar em contato AWS WAF, você pode ativar a abertura de AWS WAF falha.

Web predefinida ACLs

Ao habilitar a AWS WAF integração, você pode optar por criar automaticamente uma nova Web ACL com regras predefinidas. A Web ACL predefinida inclui três regras AWS gerenciadas que oferecem proteções contra as ameaças de segurança mais comuns.

- `AWSManagedRulesAmazonIpReputationList`: o grupo de regras da lista de reputação de IP da Amazon bloqueia endereços IP normalmente associados a bots ou outras ameaças. Para obter mais informações, consulte o grupo de regras gerenciadas da lista de reputação de IP da Amazon no Guia do desenvolvedor do AWS WAF .
- `AWSManagedRulesCommonRuleSet`: O grupo de regras do conjunto de regras principais (CRS) fornece proteção contra a exploração de uma ampla gama de vulnerabilidades, incluindo algumas das vulnerabilidades comuns e de alto risco descritas em publicações do OWASP, como OWASP Top 10. Para obter mais informações, consulte Core rule set (CRS) managed rule group no Guia do desenvolvedor do AWS WAF .
- `AWSManagedRulesKnownBadInputsRuleSet`: o grupo de regras de entradas nocivas conhecidas bloqueia padrões de solicitação conhecidos como inválidos e associados à exploração ou à descoberta de vulnerabilidades. Para obter mais informações, consulte Known bad inputs managed rule group no Guia do desenvolvedor do AWS WAF .

Para obter mais informações, consulte Usando a web ACLs AWS WAF no Guia do AWS WAF desenvolvedor.

Receptores para seus Application Load Balancers

Um receptor é um processo que verifica solicitações de conexão usando o protocolo e a porta configurados por você. Antes de começar a usar seu Application Load Balancer, você deve adicionar ao menos um receptor. Se seu balanceador de carga não tiver receptores, ele não poderá receber tráfego dos clientes. As regras que você define para seus ouvintes determinam como o balanceador de carga encaminha as solicitações para os destinos que você registra, como EC2 instâncias.

Conteúdo

- Configuração do receptor
- Atributos do receptor
- Ação padrão
- Criar um receptor HTTP para seu Application Load Balancer
- Certificados SSL para o Application Load Balancer
- Políticas de segurança para o Application Load Balancer
- Criar um receptor HTTPS para seu Application Load Balancer
- Atualizar um receptor HTTPS para seu Application Load Balancer
- Autenticação mútua com TLS no Application Load Balancer
- Autenticar usuários usando um Application Load Balancer
- Regras do receptor para seu Application Load Balancer
- Cabeçalhos HTTP e Application Load Balancers
- Modificação do cabeçalho HTTP para seu Application Load Balancer
- Excluir um receptor para seu Application Load Balancer

Configuração do receptor

Os listeners são compatíveis com os seguintes protocolos e portas:

- Protocolos: HTTP, HTTPS
- Ports (Portas): 1-65535

Você pode usar um listener HTTPS para redirecionar o trabalho de criptografia e descriptografia ao seu load balancer, de forma que os aplicativos possam se concentrar na respectiva lógica de

negócios. Se o protocolo de listener for HTTPS, você deverá implantar pelo menos um certificado de servidor SSL no listener. Para obter mais informações, consulte Criar um receptor HTTPS para seu Application Load Balancer.

Se você precisar garantir que os destinos descriptografem o tráfego HTTPS em vez do balanceador de carga, é possível criar um Network Load Balancer com um receptor TCP na porta 443. Com um receptor TCP, o balanceador de carga transmite o tráfego criptografado para os destinos sem descriptografá-lo. Para obter mais informações, consulte o Guia do usuário de network load balancers.

WebSockets

Os Application Load Balancers fornecem suporte nativo para WebSockets. Você pode atualizar uma conexão HTTP/1.1 existente em uma conexão WebSocket (`wsouwss`) usando uma atualização de conexão HTTP. Quando você atualiza, a conexão TCP usada para solicitações (para o balanceador de carga e para o destino) se torna uma WebSocket conexão persistente entre o cliente e o destino por meio do balanceador de carga. Você pode usar WebSockets com ouvintes HTTP e HTTPS. As opções que você escolhe para seu ouvinte se aplicam às WebSocket conexões e ao tráfego HTTP. Para obter mais informações, consulte Como o WebSocket protocolo funciona no Amazon CloudFront Developer Guide.

HTTP/2

Application Load Balancers têm compatibilidade nativa para HTTP/2 com receptores HTTPS. Você pode enviar até 128 solicitações em paralelo usando uma conexão HTTP/2. Você pode usar a versão do protocolo para enviar a solicitação aos destinos usando HTTP/2. Para obter mais informações, consulte Versão do protocolo. Como HTTP/2 usa conexões front-end de forma mais eficiente, você pode perceber menos conexões entre clientes e o load balancer. Você não pode usar o recurso server-push do HTTP/2.

A autenticação TLS mútua para Application Load Balancers oferece suporte a HTTP/2 nos modos de passagem e verificação. Para obter mais informações, consulte Autenticação mútua com TLS no Application Load Balancer.

Para obter mais informações, consulte Roteamento de solicitação no Guia do usuário do Elastic Load Balancing.

Atributos do receptor

A seguir estão os atributos do ouvinte para Application Load Balancers:

`routing.http.request.x_amzn_mtls_clientcert_serial_number.header_name`

Permite modificar o nome do cabeçalho da solicitação HTTP X-Amzn-Mtls-Clientcert-Serial-Number.

`routing.http.request.x_amzn_mtls_clientcert_issuer.header_name`

Permite que você modifique o nome do cabeçalho da solicitação HTTP X-Amzn-Mtls-Clientcert-Issuer.

`routing.http.request.x_amzn_mtls_clientcert_subject.header_name`

Permite que você modifique o nome do cabeçalho da solicitação HTTP X-Amzn-Mtls-Clientcert-Subject.

`routing.http.request.x_amzn_mtls_clientcert_validity.header_name`

Permite que você modifique o nome do cabeçalho da solicitação HTTP X-Amzn-Mtls-Clientcert-Validity.

`routing.http.request.x_amzn_mtls_clientcert_leaf.header_name`

Permite que você modifique o nome do cabeçalho da solicitação HTTP X-Amzn-Mtls-Clientcert-Leaf.

`routing.http.request.x_amzn_mtls_clientcert.header_name`

Permite que você modifique o nome do cabeçalho da solicitação HTTP X-Amzn-Mtls-Clientcert.

`routing.http.request.x_amzn_tls_version.header_name`

Permite que você modifique o nome do cabeçalho da solicitação HTTP X-Amzn-Tls-Version.

`routing.http.request.x_amzn_tls_cipher_suite.header_name`

Permite que você modifique o nome do cabeçalho da solicitação HTTP X-Amzn-Tls-Cipher-Suite.

`routing.http.response.server.enabled`

Permite permitir ou remover o cabeçalho do servidor de resposta HTTP.

`routing.http.response.strict_transport_security.header_value`

Informa aos navegadores que o site só deve ser acessado usando HTTPS e que qualquer tentativa futura de acessá-lo usando HTTP deve ser automaticamente convertida em HTTPS.

`routing.http.response.access_control_allow_origin.header_value`

Especifica quais origens têm permissão para acessar o servidor.

`routing.http.response.access_control_allow_methods.header_value`

Retorna quais métodos HTTP são permitidos ao acessar o servidor de uma origem diferente.

`routing.http.response.access_control_allow_headers.header_value`

Especifica quais cabeçalhos podem ser usados durante a solicitação.

`routing.http.response.access_control_allow_credentials.header_value`

Indica se o navegador deve incluir credenciais, como cookies ou autenticação, ao fazer solicitações.

`routing.http.response.access_control_expose_headers.header_value`

Retorna quais cabeçalhos o navegador pode expor ao cliente solicitante.

`routing.http.response.access_control_max_age.header_value`

Especifica por quanto tempo os resultados de uma solicitação de preflight podem ser armazenados em cache, em segundos.

`routing.http.response.content_security_policy.header_value`

Especifica as restrições impostas pelo navegador para ajudar a minimizar o risco de certos tipos de ameaças à segurança.

`routing.http.response.x_content_type_options.header_value`

Indica se os tipos MIME anunciados nos cabeçalhos Content-Type devem ser seguidos e não alterados.

`routing.http.response.x_frame_options.header_value`

Indica se o navegador tem permissão para renderizar uma página em um quadro, iframe, incorporação ou objeto.

Ação padrão

Cada receptor tem uma ação padrão, também conhecida como regra padrão. A regra padrão não pode ser excluída e sempre é executada por último. Você pode criar regras adicionais. Essas regras consistem em uma prioridade, uma ou mais ações e uma ou mais condições. Você pode adicionar ou editar regras a qualquer momento. Para obter mais informações, consulte Regras do listener.

Criar um receptor HTTP para seu Application Load Balancer

Um receptor verifica se há solicitações de conexão. Você define um listener ao criar seu load balancer e você pode adicionar listeners ao seu load balancer a qualquer momento.

As informações dessa página ajudam você a criar um listener HTTP para o load balancer. Para adicionar um listener HTTPS ao seu load balancer, consulte Criar um receptor HTTPS para seu Application Load Balancer.

Pré-requisitos

- Para adicionar uma ação de encaminhamento à regra do listener padrão, você deve especificar um grupo de destino disponível. Para obter mais informações, consulte Criar um grupo de destino para o Application Load Balancer.
- Você pode especificar o mesmo grupo de destino em vários receptores, mas esses receptores devem pertencer ao mesmo balanceador de carga. Para usar um grupo de destino com um balanceador de carga, você deve verificar se ele não está sendo usado por um receptor para nenhum outro balanceador de carga.

Adicionar um receptor HTTP

Você configura um listener com um protocolo e uma porta para as conexões de clientes com o load balancer, e um grupo de destino para a regra do listener padrão. Para obter mais informações, consulte Configuração do receptor.

Para adicionar um listener HTTPS usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Receptores e regras, escolha Adicionar receptor.
5. Em Protocolo:Porta, escolha HTTP e mantenha a porta padrão ou insira outra porta.
6. Em Ação padrão, selecione uma das seguintes ações de roteamento e forneça as informações necessárias:

    - Encaminhar para grupos-alvo — Escolha um grupo-alvo. Para adicionar outro grupo-alvo, escolha Adicionar grupo-alvo, escolha um grupo-alvo, revise as porcentagens relativas e

atualize os pesos conforme necessário. Você deve ativar a aderência em nível de grupo se tiver ativado a aderência em qualquer um dos grupos-alvo.

- Redirecionar para URL — Insira o URL inserindo cada parte separadamente na guia Partes do URI ou inserindo o endereço completo na guia URL completo. Para Código de status, selecione temporário (HTTP 302) ou permanente (HTTP 301) com base em suas necessidades.
- Retornar resposta fixa — insira o código de resposta para retornar as solicitações canceladas do cliente. Opcionalmente, você pode especificar o tipo de conteúdo e o corpo da resposta.

7. (Opcional) Para adicionar tags, expanda as tags Listener. Escolha Adicionar nova tag e insira a chave e o valor da tag.
8. Escolha Adicionar.

Para adicionar um ouvinte HTTP usando o AWS CLI

Use o comando create-listener para criar o listener e a regra padrão, e o comando create-rule para definir as regras de listener adicionais.

Certificados SSL para o Application Load Balancer

Ao criar um receptor seguro para o Application Load Balancer, você deve implantar ao menos um certificado no balanceador de carga. O load balancer requer certificados X.509 (certificados de servidor SSL/TLS). Os certificados são uma forma digital de identificação emitida por uma autoridade certificadora (CA). Um certificado contém informações de identificação, período de validade, chave pública, número de série e a assinatura digital do emissor.

Quando você cria um certificado para uso com seu load balancer, é necessário especificar um nome de domínio. O nome de domínio no certificado deve corresponder ao registro de nome de domínio personalizado para que possamos verificar a conexão TLS. Se eles não coincidirem, o tráfego não será criptografado.

Você precisa especificar um nome de domínio totalmente qualificado (FQDN) para seu certificado, como `www.example.com` ou um nome de domínio de apex como `example.com`. Você também pode usar um asterisco (*) como um caractere curinga para proteger vários nomes de site no mesmo domínio. Quando você solicita um certificado-curinga, o asterisco (*) deve estar na posição mais à esquerda do nome do domínio e só pode proteger um nível de subdomínio. Por exemplo,`*.example.com` protege `corp.example.com` e `images.example.com`, mas não

pode proteger `test.login.example.com`. Note também que `*.example.com` protege apenas os subdomínios de `example.com`, mas não protege o domínio vazio ou apex (`example.com`). O nome-curinga será exibido no campo Assunto e na extensão Nome alternativo do assunto do certificado. Para obter mais informações sobre certificados públicos, consulte Solicitar um certificado público no Guia AWS Certificate Manager do usuário.

Recomendamos que você crie certificados para o seu balanceador de carga usando o AWS Certificate Manager (ACM). O ACM é compatível com comprimentos de chave de 2.048, 3.072 e 4.096, e com todos os certificados ECDSA. O ACM se integra ao Elastic Load Balancing para que você possa implantar o certificado em seu balanceador de carga. Para obter mais informações, consulte o Guia do usuário do AWS Certificate Manager.

Como alternativa, você pode usar SSL/TLS ferramentas para criar uma solicitação de assinatura de certificado (CSR) e, em seguida, obter a CSR assinada por uma CA para produzir um certificado e, em seguida, importar o certificado para o ACM ou fazer o upload do certificado no AWS Identity and Access Management (IAM). Para obter mais informações sobre a importação de certificados para o ACM, consulte Importação de certificados no Guia do usuário do AWS Certificate Manager . Para obter mais informações sobre a upload de certificados no IAM, consulte Trabalhar com certificados de servidor no Manual do usuário do IAM.

Certificado padrão

Quando você cria um listener HTTPS, deve especificar exatamente um certificado. Esse certificado é conhecido como o certificado padrão. É possível substituir o certificado padrão depois de criar o listener HTTPS. Para obter mais informações, consulte Substituir o certificado padrão.

Se você especificar certificados adicionais em uma lista de certificados, o certificado padrão será usado somente se um cliente se conectar sem usar o protocolo Server Name Indication (SNI) para especificar um nome de host ou se não houver certificados correspondentes na lista de certificados.

Se você não especificar certificados adicionais, mas precisar hospedar vários aplicativos seguros por meio de um único load balancer, poderá usar um certificado curinga ou adicionar um Subject Alternative Name (SAN) para cada domínio adicional ao seu certificado.

Lista de certificados

Depois de criar um ouvinte HTTPS, você pode adicionar certificados à lista de certificados. Se você criou o ouvinte usando o AWS Management Console, adicionamos o certificado padrão à lista de

certificados para você. Caso contrário, a lista de certificados estará vazia. O uso de uma lista de certificados permite que um load balancer ofereça suporte a vários domínios na mesma porta e forneça um certificado diferente para cada domínio. Para obter mais informações, consulte Adicionar certificados à lista de certificados.

O load balancer usa um algoritmo inteligente de seleção de certificado com suporte para SNI. Se o nome de host fornecido por um cliente corresponder a um único certificado na lista, o load balancer selecionará esse certificado. Se um nome de host fornecido por um cliente corresponder a vários certificados na lista, o load balancer selecionará o melhor certificado que o cliente puder comportar. A seleção do certificado se baseia nos critérios a seguir, na seguinte ordem:

- Algoritmo de chave pública (prefira ECDSA em relação a RSA)
- Expiração (de preferência não expirada)
- Algoritmo de hashing (prefira SHA em vez de MD5). Se houver vários certificados SHA, prefira o número SHA mais alto.
- Comprimento da chave (prefira o maior)
- Período de validade

As entradas no log de acesso do load balancer indicam o hostname especificado pelo cliente e o certificado apresentado ao cliente. Para obter mais informações, consulte Entradas do log de acesso.

Renovação de certificado

Cada certificado vem com um período de validade. Você deve garantir que renovou ou substituiu os certificados do load balancer antes do fim do período de validade. Isso inclui o certificado padrão e os certificados em uma lista de certificados. Renovar ou substituir um certificado não afeta as solicitações em andamento recebidas por um nó do load balancer e são pendentes de roteamento para um destino íntegro. Depois de um certificado ser renovado, as novas solicitações usarão o certificado renovado. Depois de o certificado ser substituído, as novas solicitações usarão o novo certificado.

Você pode gerenciar a renovação e a substituição do certificado da seguinte forma:

- Os certificados fornecidos AWS Certificate Manager e implantados em seu balanceador de carga podem ser renovados automaticamente. O ACM tenta renovar os certificados antes que eles expirem. Para obter mais informações, consulte Renovação gerenciada no Guia do usuário do AWS Certificate Manager .

- Se você tiver importado um certificado no ACM, deverá monitorar a data de validade do certificado e renová-lo antes que expire. Para obter mais informações, consulte Importar certificados no Manual do usuário do AWS Certificate Manager .
- Se você tiver importado um certificado para o IAM, precisará criar um novo certificado, importá-lo para o ACM ou IAM, adicionar o novo certificado ao balanceador de carga e remover o certificado expirado do seu balanceador de carga.

Políticas de segurança para o Application Load Balancer

O Elastic Load Balancing usa uma configuração de negociação com Secure Sockets Layer (SSL), conhecida como política de segurança, para negociar conexões SSL entre um cliente e o balanceador de carga. Uma política de segurança é uma combinação de cifras e protocolos. O protocolo estabelece uma conexão segura entre um cliente e um servidor, além de garantir que todos os dados passados entre o cliente e o load balancer sejam privados. A cifra é um algoritmo de criptografia que usa chaves de criptografia para criar uma mensagem codificada. Os protocolos usam várias cifras para criptografar dados pela Internet. Durante o processo de negociação de conexão, o cliente e o load balancer apresentam uma lista de cifras e protocolos que cada um suporta, em ordem de preferência. Por padrão, a primeira cifra na lista do servidor que corresponder a qualquer uma das cifras do cliente é selecionada para a conexão segura.

Considerações

- Os Application Load Balancers são compatíveis com renegociação de SSL apenas para conexões de destino.
- Os Application Load Balancers não são compatíveis com políticas de segurança personalizadas.
- Um ouvinte HTTPS exige uma política de segurança. Se você não especificar uma política de segurança ao criar o ouvinte, usaremos a política de segurança padrão. A política de segurança padrão depende de como você criou o ouvinte HTTPS:
  - Console — A política de segurança padrão é`ELBSecurityPolicy-TLS13-1-2-Res-2021-06`.
  - Outros métodos (por exemplo, o AWS CLI AWS CloudFormation, e o AWS CDK) — A política de segurança padrão é`ELBSecurityPolicy-2016-08`.
- Você pode escolher a política de segurança usada para conexões front-end, mas não para conexões backend. A política de segurança para conexões de back-end depende da política de segurança do ouvinte:

  - Se o ouvinte HTTPS usar uma política de segurança TLS 1.3, as conexões de back-end usarão a política. `ELBSecurityPolicy-TLS13-1-0-2021-06`
  - Se o ouvinte HTTPS usar uma política FIPS, as conexões de back-end usarão a política. `ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04`
  - Caso contrário, as conexões de back-end usarão a `ELBSecurityPolicy-2016-08` política.
- Para atender às normas de conformidade e segurança que exigem a desativação de determinadas versões do protocolo TLS ou para oferecer suporte a clientes legados que exigem cifras descontinuadas, use uma das políticas de segurança `ELBSecurityPolicy-TLS-`. Para exibir a versão do protocolo TLS das solicitações para o Application Load Balancer, habilite os logs de acesso para o balanceador de carga e examine as entradas correspondentes do log de acesso. Para obter mais informações, consulte Logs de acesso.
- Você pode restringir quais políticas de segurança estão disponíveis para os usuários em todo o seu Contas da AWS e AWS Organizations usando as chaves de condição do Elastic Load Balancing em suas políticas de IAM e controle de serviço (SCPs), respectivamente. Para obter mais informações, consulte Políticas de controle de serviço (SCPs) no Guia AWS Organizations do usuário.
- As políticas que oferecem suporte somente ao TLS 1.3 oferecem suporte ao Forward Secrecy (FS). Políticas que oferecem suporte a TLS 1.3 e TLS 1.2 que têm somente cifras no formato TLS_* e ECDHE_* também fornecem FS.
- Os Application Load Balancers suportam a retomada de TLS usando PSK (TLS 1.3) e IDs/session tíquetes de sessão (TLS 1.2 e anteriores). Há suporte para retomadas apenas em conexões com o mesmo endereço IP do Application Load Balancer. O recurso 0-RTT Data e a extensão early_data não estão implementados.
- Os Application Load Balancers oferecem suporte à extensão Extended Master Secret (EMS) para TLS 1.2.

Você pode descrever os protocolos e as cifras usando o describe-ssl-policies AWS CLI comando ou consultar as tabelas abaixo.

Políticas de segurança

- Políticas de segurança de TLS
  - Protocolos por política
  - Cifras por política
  - Políticas por cifra

- Políticas de segurança do FIPS
  - Protocolos por política
  - Cifras por política
  - Políticas por cifra
- Políticas compatíveis com FS
  - Protocolos por política
  - Cifras por política
  - Políticas por cifra

Políticas de segurança de TLS

Você pode usar as políticas de segurança do TLS para atender aos requisitos de conformidade e padrões de segurança que exigem a desativação de determinadas versões do protocolo TLS ou para oferecer suporte a clientes legados que exigem cifras descontinuadas.

As políticas que oferecem suporte somente ao TLS 1.3 oferecem suporte ao Forward Secrecy (FS). Políticas que oferecem suporte a TLS 1.3 e TLS 1.2 que têm somente cifras no formato TLS_* e ECDHE_* também fornecem FS.

Conteúdo

- Protocolos por política
- Cifras por política
- Políticas por cifra

Protocolos por política

A tabela a seguir descreve os protocolos compatíveis com cada política de segurança do TLS.

Políticas de segurança	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolítica- TLS13 -1-3-2021-06	Sim	Não	Nº	Nº

Políticas de segurança	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolítica- TLS13 -1-2-2021-06	Yes (Sim)	Yes (Sim)	Não	Nº
ELBSecurityPolítica- TLS13 -1-2-Res-2021-06	Yes (Sim)	Yes (Sim)	Não	Nº
ELBSecurityPolítica- TLS13 -1-2-Ext2-2021-06	Yes (Sim)	Yes (Sim)	Não	Nº
ELBSecurityPolítica- TLS13 -1-2-Ext1-2021-06	Yes (Sim)	Yes (Sim)	Não	Nº
ELBSecurityPolítica- TLS13 -1-1-2021-06	Yes (Sim)	Yes (Sim)	Yes (Sim)	Não
ELBSecurityPolítica- TLS13 -1-0-2021-06	Yes (Sim)	Yes (Sim)	Yes (Sim)	Yes (Sim)
ELBSecurityPolítica-TLS-1-2-EXT-2018-06	Não	Sim	Não	Nº
ELBSecurityPolítica-TLS-1-2-2017-01	Não	Sim	Não	Nº
ELBSecurityPolítica-TLS-1-1-2017-01	Não	Yes (Sim)	Yes (Sim)	Não

Políticas de segurança	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolítica-2016-08	Não	Yes (Sim)	Yes (Sim)	Yes (Sim)
ELBSecurityPolítica-2015-05	Não	Yes (Sim)	Yes (Sim)	Yes (Sim)

Cifras por política

A tabela a seguir descreve as cifras compatíveis com cada política de segurança do TLS.

Política de segurança	Cifras
ELBSecurityPolítica- TLS13 -1-3-2021-06	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384 • TLS_ 0_ 05_ CHACHA2 POLY13 SHA256
ELBSecurityPolítica- TLS13 -1-2-2021-06	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384 • TLS_ 0_ 05_ CHACHA2 POLY13 SHA256 • ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384
ELBSecurityPolítica- TLS13 -1-2-Res-2021-06	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384

Política de segurança	Cifras
	• TLS_ 0_ 05_ CHACHA2 POLY13 SHA256 • ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384
ELBSecurityPolítica- TLS13 -1-2-Ext2-2021-06	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384 • TLS_ 0_ 05_ CHACHA2 POLY13 SHA256 • ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-ECDSA- -SHA AES256 • ECDHE-RSA- -SHA AES256 • AES128-GCM- SHA256 • AES128-SHA256 • AES128-SHAH • AES256-GCM- SHA384 • AES256-SHA256 • AES256-SHAH

Política de segurança	Cifras
ELBSecurityPolítica- TLS13 -1-2-Ext1-2021-06	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384 • TLS_ 0_ 05_ CHACHA2 POLY13 SHA256 • ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • AES128-GCM- SHA256 • AES128-SHA256 • AES256-GCM- SHA384 • AES256-SHA256

Política de segurança	Cifras
ELBSecurityPolítica- TLS13 -1-1-2021-06	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384 • TLS_ 0_ 05_ CHACHA2 POLY13 SHA256 • ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-ECDSA- -SHA AES256 • ECDHE-RSA- -SHA AES256 • AES128-GCM- SHA256 • AES128-SHA256 • AES128-SHAH • AES256-GCM- SHA384 • AES256-SHA256 • AES256-SHAH

Política de segurança	Cifras
ELBSecurityPolítica- TLS13 -1-0-2021-06	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384 • TLS_ 0_ 05_ CHACHA2 POLY13 SHA256 • ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-ECDSA- -SHA AES256 • ECDHE-RSA- -SHA AES256 • AES128-GCM- SHA256 • AES128-SHA256 • AES128-SHAH • AES256-GCM- SHA384 • AES256-SHA256 • AES256-SHAH

Política de segurança	Cifras
ELBSecurityPolítica-TLS-1-2-EXT-2018-06	• ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-ECDSA- -SHA AES256 • ECDHE-RSA- -SHA AES256 • AES128-GCM- SHA256 • AES128-SHA256 • AES128-SHAH • AES256-GCM- SHA384 • AES256-SHA256 • AES256-SHAH

Política de segurança	Cifras
ELBSecurityPolítica-TLS-1-2-2017-01	• ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • AES128-GCM- SHA256 • AES128-SHA256 • AES256-GCM- SHA384 • AES256-SHA256

Política de segurança	Cifras
ELBSecurityPolítica-TLS-1-1-2017-01	• ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-ECDSA- -SHA AES256 • ECDHE-RSA- -SHA AES256 • AES128-GCM- SHA256 • AES128-SHA256 • AES128-SHAH • AES256-GCM- SHA384 • AES256-SHA256 • AES256-SHAH

Política de segurança	Cifras
ELBSecurityPolítica-2016-08	• ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-ECDSA- -SHA AES256 • ECDHE-RSA- -SHA AES256 • AES128-GCM- SHA256 • AES128-SHA256 • AES128-SHAH • AES256-GCM- SHA384 • AES256-SHA256 • AES256-SHAH

Política de segurança	Cifras
ELBSecurityPolítica-2015-05	• ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-ECDSA- -SHA AES256 • ECDHE-RSA- -SHA AES256 • AES128-GCM- SHA256 • AES128-SHA256 • AES128-SHAH • AES256-GCM- SHA384 • AES256-SHA256 • AES256-SHAH

Políticas por cifra

A tabela a seguir descreve as políticas de segurança do TLS compatíveis com cada cifra.

Nome da cifra	Políticas de segurança	Pacote de cifras
OpenSSL — TLS_AES_128_GCM_ SHA256 IANA — TLS_AES_128_GCM_ SHA256	• ELBSecurityPolítica- TLS13 -1-3-2021 -06 • ELBSecurityPolítica- TLS13 -1-2-2021 -06	1301

Nome da cifra	Políticas de segurança	Pacote de cifras
	• ELBSecurityPolítica- TLS13 -1-2-Res-2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06	
OpenSSL — TLS_AES_256_GCM_ SHA384 IANA — TLS_AES_256_GCM_ SHA384	• ELBSecurityPolítica- TLS13 -1-3-2021 -06 • ELBSecurityPolítica- TLS13 -1-2-2021 -06 • ELBSecurityPolítica- TLS13 -1-2-Res-2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06	1302

Nome da cifra	Políticas de segurança	Pacote de cifras
OpenSSL — TLS_ 0_ 05_ CHACHA2 POLY13 SHA256 IANA — TLS_ 0_ 05_ CHACHA2 POLY13 SHA256	• ELBSecurityPolítica- TLS13 -1-3-2021 -06 • ELBSecurityPolítica- TLS13 -1-2-2021 -06 • ELBSecurityPolítica- TLS13 -1-2-Res- 2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06	1303

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-ECDSA-AESOpenSSL — 128 GCM- SHA256 IANA — TLS_ECDHE_ECDSA_COM_AES_128_GCM_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-2021-06 • ELBSecurityPolítica- TLS13 -1-2-Res-2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext2-2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1-2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021-06 • ELBSecurityPolítica- TLS13 -1-0-2021-06 • ELBSecurityPolítica-TLS-1-2-EXT-2018-06 • ELBSecurityPolítica-TLS-1-2-2017-01 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	c02b

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-RSA-AESOpenSSL — 128 GCM- SHA256 IANA — TLS_ECDHE_RSA_COM_ AES_128_GCM_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-2021 -06 • ELBSecurityPolítica- TLS13 -1-2-Res- 2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2- EXT-2018-06 • ELBSecurityPolítica-TLS-1-2-2017-01 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	c02f

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-ECDSA-AESOpenSSL — 128-SHA256 IANA — TLS_ECDHE_ECDSA_COM_AES_128_CBC_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext2-2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1-2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021-06 • ELBSecurityPolítica- TLS13 -1-0-2021-06 • ELBSecurityPolítica-TLS-1-2-EXT-2018-06 • ELBSecurityPolítica-TLS-1-2-2017-01 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	c023

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-RSA-AESOpenSSL — 128-SHA256 IANA — TLS_ECDHE_RSA_COM_ AES_128_CBC_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-2021 -06 • ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2- EXT-2018-06 • ELBSecurityPolítica-TLS-1-2-2017-01 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	c027
ECDHE-ECDSA-AESOpenSSL — 128 SHA IANA: TLS_ECDHE_ECDSA_WI TH_AES_128_CBC_SHA	• ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2- EXT-2018-06 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	c009

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-RSA-AESOpenSSL — 128 SHA IANA: TLS_ECDHE_RSA_WITH _AES_128_CBC_SHA	• ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2- EXT-2018-06 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	c013
ECDHE-ECDSA-AESOpenSSL — 256 GCM- SHA384 IANA — TLS_ECDHE_ECDSA_CO M_AES_256_GCM_ SHA384	• ELBSecurityPolítica- TLS13 -1-2-2021 -06 • ELBSecurityPolítica- TLS13 -1-2-Res- 2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2- EXT-2018-06 • ELBSecurityPolítica-TLS-1-2-2017-01 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	c02c

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-RSA-AESOpenSSL — 256 GCM- SHA384 IANA — TLS_ECDHE_RSA_COM_ AES_256_GCM_ SHA384	• ELBSecurityPolítica- TLS13 -1-2-2021 -06 • ELBSecurityPolítica- TLS13 -1-2-Res- 2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2- EXT-2018-06 • ELBSecurityPolítica-TLS-1-2-2017-01 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	c030

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-ECDSA-AESOpenSSL — 256-SHA384 IANA — TLS_ECDHE_ECDSA_COM_AES_256_CBC_ SHA384	• ELBSecurityPolítica- TLS13 -1-2-2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext2-2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1-2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021-06 • ELBSecurityPolítica- TLS13 -1-0-2021-06 • ELBSecurityPolítica-TLS-1-2-EXT-2018-06 • ELBSecurityPolítica-TLS-1-2-2017-01 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	c024

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-RSA-AESOpenSSL — 256-SHA384 IANA — TLS_ECDHE_RSA_COM_ AES_256_CBC_ SHA384	• ELBSecurityPolítica- TLS13 -1-2-2021 -06 • ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2-EXT-2018-06 • ELBSecurityPolítica-TLS-1-2-2017-01 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	c028
ECDHE-ECDSA-AESOpenSSL — 256 SHA IANA: TLS_ECDHE_ECDSA_WI TH_AES_256_CBC_SHA	• ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2-EXT-2018-06 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	c00a

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-RSA-AESOpenSSL — 256 SHA IANA: TLS_ECDHE_RSA_WITH _AES_256_CBC_SHA	• ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2- EXT-2018-06 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	c014
AES128OpenSSL — -GCM- SHA256 IANA — TLS_RSA_COM_AES_12 8_GCM_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2- EXT-2018-06 • ELBSecurityPolítica-TLS-1-2-2017-01 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	9c

Nome da cifra	Políticas de segurança	Pacote de cifras
AES128OpenSSL — - SHA256 IANA — TLS_RSA_COM_AES_128_CBC_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2-EXT-2018-06 • ELBSecurityPolítica-TLS-1-2-2017-01 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	3c
AES128OpenSSL — -SHA IANA: TLS_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2-EXT-2018-06 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	2f

Nome da cifra	Políticas de segurança	Pacote de cifras
AES256OpenSSL — -GCM- SHA384 IANA — TLS_RSA_COM_AES_25 6_GCM_ SHA384	• ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2- EXT-2018-06 • ELBSecurityPolítica-TLS-1-2-2017-01 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	9d
AES256OpenSSL — - SHA256 IANA — TLS_RSA_COM_AES_25 6_CBC_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-2-Ext1 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2- EXT-2018-06 • ELBSecurityPolítica-TLS-1-2-2017-01 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	3d

Nome da cifra	Políticas de segurança	Pacote de cifras
AES256OpenSSL — -SHA IANA: TLS_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolítica- TLS13 -1-2-Ext2 -2021-06 • ELBSecurityPolítica- TLS13 -1-1-2021 -06 • ELBSecurityPolítica- TLS13 -1-0-2021 -06 • ELBSecurityPolítica-TLS-1-2-EXT-2018-06 • ELBSecurityPolítica-TLS-1-1-2017-01 • ELBSecurityPolítica-2016-08	35

Políticas de segurança do FIPS

> **Important**
>
> Todos os receptores seguros conectados a um Application Load Balancer devem usar políticas de segurança do FIPS ou políticas de segurança que não sejam do FIPS; elas não podem ser misturadas. Se um Application Load Balancer existente tiver dois ou mais receptores usando políticas que não sejam do FIPS e você quiser que eles usem políticas de segurança do FIPS, remova todos os receptores até que reste apenas um. Altere a política de segurança do receptor para FIPS e crie mais receptores usando políticas de segurança do FIPS. Como alternativa, você pode criar um Application Load Balancer com novos receptores usando somente políticas de segurança do FIPS.

O Federal Information Processing Standard (FIPS) é um padrão do governo dos EUA e do Canadá que especifica os requisitos de segurança para módulos criptográficos que protegem informações confidenciais. Para saber mais, consulte Federal Information Processing Standard (FIPS) 140 na página AWS Cloud Security Compliance.

Todas as políticas do FIPS utilizam o módulo criptográfico AWS-LC validado pelo FIPS. Para saber mais, consulte a página AWS-LC Cryptographic Module no site NIST Cryptographic Module Validation Program.

> **Important**
>
> As políticas `ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04` e `ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04` são fornecidas somente para compatibilidade legada. Embora utilizem criptografia FIPS usando o módulo FIPS14 0, podem não estar em conformidade com as diretrizes mais recentes do NIST para configuração de TLS.

Conteúdo

- Protocolos por política
- Cifras por política
- Políticas por cifra

Protocolos por política

A tabela a seguir descreve os protocolos compatíveis com cada política de segurança do FIPS.

Políticas de segurança	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolítica- TLS13 -1-3-FIPS-2023-04	Sim	Não	Nº	Nº
ELBSecurityPolítica- TLS13 -1-2-FIPS-2023-04	Yes (Sim)	Yes (Sim)	Não	Nº
ELBSecurityPolítica- TLS13 -1-2-RES-FIPS-2023-04	Yes (Sim)	Yes (Sim)	Não	Nº

Políticas de segurança	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04	Yes (Sim)	Yes (Sim)	Não	Nº
ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04	Yes (Sim)	Yes (Sim)	Não	Nº
ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04	Yes (Sim)	Yes (Sim)	Não	Nº
ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04	Yes (Sim)	Yes (Sim)	Yes (Sim)	Não
ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	Yes (Sim)	Yes (Sim)	Yes (Sim)	Yes (Sim)

Cifras por política

A tabela a seguir descreve as cifras compatíveis com cada política de segurança do FIPS.

Política de segurança	Cifras
ELBSecurityPolítica- TLS13 -1-3-FIPS-2023-04	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384
ELBSecurityPolítica- TLS13 -1-2-FIPS-2023-04	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384 • ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256

Política de segurança	Cifras
	• ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384
ELBSecurityPolítica- TLS13 -1-2-RES-FIPS-2023-04	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384 • ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384

Política de segurança	Cifras
ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384 • ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-RSA- -SHA AES256 • ECDHE-ECDSA- -SHA AES256 • AES128-GCM- SHA256 • AES128-SHA256 • AES128-SHAH • AES256-GCM- SHA384 • AES256-SHA256 • AES256-SHAH

Política de segurança	Cifras
ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384 • ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • AES128-GCM- SHA256 • AES128-SHA256 • AES256-GCM- SHA384 • AES256-SHA256
ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384 • ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-RSA- -SHA AES256 • ECDHE-ECDSA- -SHA AES256

Política de segurança	Cifras
ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384 • ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-RSA- -SHA AES256 • ECDHE-ECDSA- -SHA AES256 • AES128-GCM- SHA256 • AES128-SHA256 • AES128-SHAH • AES256-GCM- SHA384 • AES256-SHA256 • AES256-SHAH

Política de segurança	Cifras
ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	• TLS_AES_128_GCM_ SHA256 • TLS_AES_256_GCM_ SHA384 • ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-RSA- -SHA AES256 • ECDHE-ECDSA- -SHA AES256 • AES128-GCM- SHA256 • AES128-SHA256 • AES128-SHAH • AES256-GCM- SHA384 • AES256-SHA256 • AES256-SHAH

Políticas por cifra

A tabela a seguir descreve as políticas de segurança do FIPS compatíveis com cada cifra.

Nome da cifra	Políticas de segurança	Pacote de cifras
OpenSSL — TLS_AES_128_GCM_ SHA256	• ELBSecurityPolítica- TLS13 -1-3-FIPS -2023-04	1301

Nome da cifra	Políticas de segurança	Pacote de cifras
IANA — TLS_AES_128_GCM_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-RES-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	
OpenSSL — TLS_AES_256_GCM_ SHA384 IANA — TLS_AES_256_GCM_ SHA384	• ELBSecurityPolítica- TLS13 -1-3-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-RES-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	1302

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-ECDSA-AESOpenSSL — 128 GCM- SHA256 IANA — TLS_ECDHE_ECDSA_COM_AES_128_GCM_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-RES-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	c02b
ECDHE-RSA-AESOpenSSL — 128 GCM- SHA256 IANA — TLS_ECDHE_RSA_COM_AES_128_GCM_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-RES-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	c02f

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-ECDSA-AESOpenSSL — 128-SHA256 IANA — TLS_ECDHE_ECDSA_COM_AES_128_CBC_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-FIPS -2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS -2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS -2023-04	c023
ECDHE-RSA-AESOpenSSL — 128-SHA256 IANA — TLS_ECDHE_RSA_COM_AES_128_CBC_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-FIPS -2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS -2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS -2023-04	c027

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-ECDSA-AESOpenSSL — 128 SHA IANA: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	c009
ECDHE-RSA-AESOpenSSL — 128 SHA IANA: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	c013

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-ECDSA-AESOpenSSL — 256 GCM- SHA384 IANA — TLS_ECDHE_ECDSA_COM_AES_256_GCM_ SHA384	• ELBSecurityPolítica- TLS13 -1-2-RES-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	c02c
ECDHE-RSA-AESOpenSSL — 256 GCM- SHA384 IANA — TLS_ECDHE_RSA_COM_AES_256_GCM_ SHA384	• ELBSecurityPolítica- TLS13 -1-2-RES-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	c030

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-ECDSA-AESOpenSSL — 256-SHA384 IANA — TLS_ECDHE_ECDSA_COM_AES_256_CBC_ SHA384	• ELBSecurityPolítica- TLS13 -1-2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	c024
ECDHE-RSA-AESOpenSSL — 256-SHA384 IANA — TLS_ECDHE_RSA_COM_AES_256_CBC_ SHA384	• ELBSecurityPolítica- TLS13 -1-2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	c028

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-ECDSA-AESOpenSSL — 256 SHA IANA: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	c00a
ECDHE-RSA-AESOpenSSL — 256 SHA IANA: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-EXT0-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	c014
AES128OpenSSL — -GCM- SHA256 IANA — TLS_RSA_COM_AES_128_GCM_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	9c

Nome da cifra	Políticas de segurança	Pacote de cifras
AES128OpenSSL — - SHA256 IANA — TLS_RSA_COM_AES_128_CBC_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	3c
AES128OpenSSL — -SHA IANA: TLS_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	2f
AES256OpenSSL — -GCM- SHA384 IANA — TLS_RSA_COM_AES_256_GCM_ SHA384	• ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	9d

Nome da cifra	Políticas de segurança	Pacote de cifras
AES256OpenSSL — - SHA256 IANA — TLS_RSA_COM_AES_256_CBC_ SHA256	• ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-2-ext1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	3d
AES256OpenSSL — -SHA IANA: TLS_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolítica- TLS13 -1-2-ext2-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-1-FIPS-2023-04 • ELBSecurityPolítica- TLS13 -1-0-FIPS-2023-04	35

Políticas compatíveis com FS

As políticas de segurança com suporte do FS (Forward Secrecy) fornecem proteções adicionais contra a espionagem de dados criptografados, por meio do uso de uma chave de sessão aleatória exclusiva. Isso evita a decodificação dos dados capturados, mesmo que a chave secreta de longo prazo seja comprometida.

As políticas nesta seção oferecem suporte ao FS, e “FS” está incluído em seus nomes. No entanto, essas não são as únicas políticas que oferecem suporte ao FS. As políticas que oferecem suporte somente ao TLS 1.3 oferecem suporte ao FS. Políticas que oferecem suporte a TLS 1.3 e TLS 1.2 que têm somente cifras no formato TLS_* e ECDHE_* também fornecem FS.

Conteúdo

- Protocolos por política
- Cifras por política
- Políticas por cifra

Protocolos por política

A tabela a seguir descreve os protocolos compatíveis com cada política de segurança com suporte do FS.

Políticas de segurança	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolítica-FS-1-2-RES-2020-10	Não	Sim	Não	Nº
ELBSecurityPolítica-FS-1-2-RES-2019-08	Não	Sim	Não	Nº
ELBSecurityPolítica-FS-1-2-2019-08	Não	Sim	Não	Nº
ELBSecurityPolítica-FS-1-1-2019-08	Não	Yes (Sim)	Yes (Sim)	Não
ELBSecurityPolítica-FS-2018-06	Não	Yes (Sim)	Yes (Sim)	Yes (Sim)

Cifras por política

A tabela a seguir descreve as cifras para as quais cada política de segurança compatível com FS oferece suporte.

Política de segurança	Cifras
ELBSecurityPolítica-FS-1-2-RES-2020-10	• ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384
ELBSecurityPolítica-FS-1-2-RES-2019-08	• ECDHE-ECDSA- -GCM- AES128 SHA256

Política de segurança	Cifras
	• ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384
ELBSecurityPolítica-FS-1-2-2019-08	• ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-RSA- -SHA AES256 • ECDHE-ECDSA- -SHA AES256

Política de segurança	Cifras
ELBSecurityPolítica-FS-1-1-2019-08	• ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-RSA- -SHA AES256 • ECDHE-ECDSA- -SHA AES256
ELBSecurityPolítica-FS-2018-06	• ECDHE-ECDSA- -GCM- AES128 SHA256 • ECDHE-RSA- -GCM- AES128 SHA256 • ECDHE-ECDSA- - AES128 SHA256 • ECDHE-RSA- - AES128 SHA256 • ECDHE-ECDSA- -SHA AES128 • ECDHE-RSA- -SHA AES128 • ECDHE-ECDSA- -GCM- AES256 SHA384 • ECDHE-RSA- -GCM- AES256 SHA384 • ECDHE-ECDSA- - AES256 SHA384 • ECDHE-RSA- - AES256 SHA384 • ECDHE-RSA- -SHA AES256 • ECDHE-ECDSA- -SHA AES256

Políticas por cifra

A tabela a seguir descreve as políticas de segurança com suporte do FS, compatíveis com cada cifra.

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-ECDSA-AESOpenSSL — 128 GCM- SHA256 IANA — TLS_ECDHE_ECDSA_COM_AES_128_GCM_ SHA256	• ELBSecurityPolítica-FS-1-2-RES-2020-10 • ELBSecurityPolítica-FS-1-2-RES-2019-08 • ELBSecurityPolítica-FS-1-2-2019-08 • ELBSecurityPolítica-FS-1-1-2019-08 • ELBSecurityPolítica-FS-2018-06	c02b
ECDHE-RSA-AESOpenSSL — 128 GCM- SHA256 IANA — TLS_ECDHE_RSA_COM_AES_128_GCM_ SHA256	• ELBSecurityPolítica-FS-1-2-RES-2020-10 • ELBSecurityPolítica-FS-1-2-RES-2019-08 • ELBSecurityPolítica-FS-1-2-2019-08 • ELBSecurityPolítica-FS-1-1-2019-08 • ELBSecurityPolítica-FS-2018-06	c02f
ECDHE-ECDSA-AESOpenSSL — 128-SHA256 IANA — TLS_ECDHE_ECDSA_COM_AES_128_CBC_ SHA256	• ELBSecurityPolítica-FS-1-2-RES-2019-08 • ELBSecurityPolítica-FS-1-2-2019-08 • ELBSecurityPolítica-FS-1-1-2019-08 • ELBSecurityPolítica-FS-2018-06	c023
ECDHE-RSA-AESOpenSSL — 128-SHA256 IANA — TLS_ECDHE_RSA_COM_AES_128_CBC_ SHA256	• ELBSecurityPolítica-FS-1-2-RES-2019-08 • ELBSecurityPolítica-FS-1-2-2019-08 • ELBSecurityPolítica-FS-1-1-2019-08 • ELBSecurityPolítica-FS-2018-06	c027
ECDHE-ECDSA-AESOpenSSL — 128 SHA	• ELBSecurityPolítica-FS-1-2-2019-08 • ELBSecurityPolítica-FS-1-1-2019-08 • ELBSecurityPolítica-FS-2018-06	c009

Nome da cifra	Políticas de segurança	Pacote de cifras
IANA: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA		
ECDHE-RSA-AESOpenSSL — 128 SHA IANA: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolítica-FS-1-2-2019-08 • ELBSecurityPolítica-FS-1-1-2019-08 • ELBSecurityPolítica-FS-2018-06	c013
ECDHE-ECDSA-AESOpenSSL — 256 GCM- SHA384 IANA — TLS_ECDHE_ECDSA_COM_AES_256_GCM_ SHA384	• ELBSecurityPolítica-FS-1-2-RES-2020-10 • ELBSecurityPolítica-FS-1-2-RES-2019-08 • ELBSecurityPolítica-FS-1-2-2019-08 • ELBSecurityPolítica-FS-1-1-2019-08 • ELBSecurityPolítica-FS-2018-06	c02c
ECDHE-RSA-AESOpenSSL — 256 GCM- SHA384 IANA — TLS_ECDHE_RSA_COM_AES_256_GCM_ SHA384	• ELBSecurityPolítica-FS-1-2-RES-2020-10 • ELBSecurityPolítica-FS-1-2-RES-2019-08 • ELBSecurityPolítica-FS-1-2-2019-08 • ELBSecurityPolítica-FS-1-1-2019-08 • ELBSecurityPolítica-FS-2018-06	c030
ECDHE-ECDSA-AESOpenSSL — 256-SHA384 IANA — TLS_ECDHE_ECDSA_COM_AES_256_CBC_ SHA384	• ELBSecurityPolítica-FS-1-2-RES-2019-08 • ELBSecurityPolítica-FS-1-2-2019-08 • ELBSecurityPolítica-FS-1-1-2019-08 • ELBSecurityPolítica-FS-2018-06	c024

Nome da cifra	Políticas de segurança	Pacote de cifras
ECDHE-RSA-AESOpenSSL — 256-SHA384 IANA — TLS_ECDHE_RSA_COM_AES_256_CBC_ SHA384	• ELBSecurityPolítica-FS-1-2-RES-2019-08 • ELBSecurityPolítica-FS-1-2-2019-08 • ELBSecurityPolítica-FS-1-1-2019-08 • ELBSecurityPolítica-FS-2018-06	c028
ECDHE-ECDSA-AESOpenSSL — 256 SHA IANA: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolítica-FS-1-2-2019-08 • ELBSecurityPolítica-FS-1-1-2019-08 • ELBSecurityPolítica-FS-2018-06	c00a
ECDHE-RSA-AESOpenSSL — 256 SHA IANA: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolítica-FS-1-2-2019-08 • ELBSecurityPolítica-FS-1-1-2019-08 • ELBSecurityPolítica-FS-2018-06	c014

Criar um receptor HTTPS para seu Application Load Balancer

Um receptor verifica se há solicitações de conexão. Você define um listener ao criar seu load balancer e você pode adicionar listeners ao seu load balancer a qualquer momento.

Para criar um receptor HTTPS, você deve implantar pelo menos um certificado de servidor SSL no balanceador de carga. O load balancer usa um certificado de servidor para encerrar a conexão front-end e descriptografa solicitações dos clientes antes de enviá-las aos destinos. Você também deve especificar uma política de segurança que será usada para negociar conexões protegidas entre os clientes e o balanceador de carga.

Se precisar transmitir tráfego criptografado para destinos sem que o balanceador de carga o decodifique, você poderá criar um Network Load Balancer ou Classic Load Balancer com um receptor TCP na porta 443. Com um receptor TCP, o balanceador de carga transmite o tráfego criptografado para os destinos sem descriptografá-lo.

As informações dessa página ajudam você a criar um listener HTTPS para o load balancer. Para adicionar um listener HTTPS ao seu load balancer, consulte Criar um receptor HTTP para seu Application Load Balancer.

Pré-requisitos

- Para adicionar uma ação de encaminhamento à regra do listener padrão, você deve especificar um grupo de destino disponível. Para obter mais informações, consulte Criar um grupo de destino para o Application Load Balancer.
- Você pode especificar o mesmo grupo de destino em vários receptores, mas esses receptores devem pertencer ao mesmo balanceador de carga. Para usar um grupo de destino com um balanceador de carga, você deve verificar se ele não está sendo usado por um receptor para nenhum outro balanceador de carga.
- Os Application Load Balancers não oferecem suporte a ED25519 chaves.

Adicionar um receptor HTTPS

Você configura um listener com um protocolo e uma porta para as conexões de clientes com o load balancer, e um grupo de destino para a regra do listener padrão. Para obter mais informações, consulte Configuração do receptor.

Adicionar um listener HTTPS usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Receptores e regras, escolha Adicionar receptor.
5. Em Protocolo:Porta, escolha HTTPS e mantenha a porta padrão ou insira outra porta.
6. (Opcional) Para adicionar uma regra de autenticação, selecione Autenticar usuários, escolha um provedor de identidade e forneça as informações necessárias. Para obter mais informações, consulte Autenticar usuários usando um Application Load Balancer.
7. Em Ação de roteamento, selecione uma das seguintes ações de roteamento e forneça as informações necessárias:

   - Encaminhar para grupos-alvo — Escolha um grupo-alvo. Para adicionar outro grupo-alvo, escolha Adicionar grupo-alvo, escolha um grupo-alvo, revise as porcentagens relativas e

atualize os pesos conforme necessário. Você deve ativar a aderência em nível de grupo se tiver ativado a aderência em qualquer um dos grupos-alvo.

- Redirecionar para URL — Insira o URL inserindo cada parte separadamente na guia Partes do URI ou inserindo o endereço completo na guia URL completo. Para Código de status, selecione temporário (HTTP 302) ou permanente (HTTP 301) com base em suas necessidades.
- Retornar resposta fixa — insira o código de resposta para retornar as solicitações canceladas do cliente. Opcionalmente, você pode especificar o tipo de conteúdo e o corpo da resposta.

8. Em Política de segurança, selecionamos a política de segurança recomendada. Você pode selecionar uma política de segurança diferente conforme necessário.
9. Em SSL/TLS Certificado padrão, escolha o certificado padrão. Também adicionamos o certificado padrão à lista SNI. Você pode selecionar um certificado usando uma das seguintes opções:

   - Do ACM — Escolha um certificado do Certificado (do ACM), que exibe os certificados disponíveis do. AWS Certificate Manager
   - Do IAM — Escolha um certificado do Certificado (do IAM), que exibe os certificados para os quais você importou AWS Identity and Access Management.
   - Importar certificado — Escolha um destino para seu certificado; Importar para o ACM ou Importar para o IAM. Para a chave privada do certificado, copie e cole o conteúdo do arquivo de chave privada (codificado por PEM). Para Corpo do certificado, copie e cole o conteúdo do arquivo de certificado de chave pública (codificado por PEM). Para a cadeia de certificados, copie e cole o conteúdo do arquivo da cadeia de certificados (codificado por PEM), a menos que você esteja usando um certificado autoassinado e não seja importante que os navegadores aceitem implicitamente o certificado.

10. (Opcional) Para ativar a autenticação mútua, em Tratamento de certificados do cliente, ative a autenticação mútua (mTLS).

    O modo padrão é de passagem. Se você selecionar Verificar com armazenamento confiável:

    - Por padrão, as conexões com certificados expirados de cliente são rejeitadas. Para alterar esse comportamento, expanda Configurações avançadas de mTLS e, em Expiração do certificado do cliente, selecione Permitir certificados expirados de cliente.
    - Para Armazenamento confiável, escolha um repositório confiável existente ou escolha Novo armazenamento confiável e forneça as informações necessárias.

11. (Opcional) Para adicionar tags, expanda as tags Listener. Escolha Adicionar nova tag e insira a chave e o valor da tag.
12. Escolha Adicionar.
13. Para adicionar certificados à lista de certificados opcionais, consulteAdicionar certificados à lista de certificados.

Para adicionar um ouvinte HTTPS usando o AWS CLI

Use o comando create-listener para criar o listener e a regra padrão, e o comando create-rule para definir as regras de listener adicionais.

Atualizar um receptor HTTPS para seu Application Load Balancer

Depois de criar um listener HTTPS, você pode substituir o certificado padrão, atualizar a lista de certificados ou substituir a política de segurança.

Tarefas

- Substituir o certificado padrão
- Adicionar certificados à lista de certificados
- Remover certificados da lista de certificados
- Atualizar a política de segurança
- modificação do cabeçalho HTTP

Substituir o certificado padrão

Você pode substituir o certificado padrão do listener usando o procedimento a seguir. Para obter mais informações, consulte Certificados SSL para o Application Load Balancer.

Como substituir o certificado padrão usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Receptores e regras, escolha o texto na coluna Protocolo:Porta para abrir a página de detalhes do receptor.

5. Na guia Certificados, escolha Alterar padrão.
6. Na tabela Certificados do ACM e do IAM, selecione um novo certificado padrão.
7. Escolha Salvar como padrão.

Para substituir o certificado padrão usando o AWS CLI

Use o comando modify-listener.

Adicionar certificados à lista de certificados

Você pode adicionar certificados à lista de certificados do listener usando o procedimento a seguir. Se você criou o ouvinte usando o AWS Management Console, adicionamos o certificado padrão à lista de certificados para você. Caso contrário, a lista de certificados estará vazia. Adicionar o certificado padrão à lista de certificados garante que esse certificado seja usado com o protocolo SNI, mesmo que seja substituído como certificado padrão. Para obter mais informações, consulte Certificados SSL para o Application Load Balancer.

Para adicionar certificados à lista de certificados usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Receptores e regras, escolha o texto na coluna Protocolo:Porta para abrir a página de detalhes do receptor.
5. Na guia Certificados, selecione Adicionar certificado.
6. Para adicionar certificados que já são gerenciados pelo ACM ou pelo IAM, marque as caixas de seleção dos certificados e escolha Incluir como pendente abaixo.
7. Se você tiver um certificado que não seja gerenciado pelo ACM ou pelo IAM, escolha Importar certificado, preencha o formulário e escolha Importar.
8. Escolha Adicionar certificados pendentes.

Para adicionar um certificado à lista de certificados usando o AWS CLI

Use o comando add-listener-certificates.

Remover certificados da lista de certificados

Você pode remover certificados da lista de certificados de um listener HTTPS usando o procedimento a seguir. Depois de remover um certificado, o ouvinte não poderá mais criar conexões usando esse certificado. Para garantir que os clientes não sejam afetados, adicione um novo certificado à lista e confirme se as conexões estão funcionando antes de remover um certificado da lista.

Para remover o certificado padrão de um listener TLS, consulte Substituir o certificado padrão.

Para remover certificados da lista de certificados usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Receptores e regras, selecione o texto na coluna Protocolo:Porta para abrir a página de detalhes do receptor.
5. Na guia Certificados, marque as caixas de seleção para os certificados e escolha Remover.
6. Quando a confirmação for solicitada, insira **`confirm`** e escolha Rejeitar.

Para remover um certificado da lista de certificados usando o AWS CLI

Use o comando remove-listener-certificates.

Atualizar a política de segurança

Quando você cria um listener HTTPS, pode selecionar a política de segurança que atenda às suas necessidades. Quando uma nova política de segurança for adicionada, você poderá atualizar seu receptor HTTPS para usar a nova política de segurança. Os Application Load Balancers não são compatíveis com políticas de segurança personalizadas. Para obter mais informações, consulte Políticas de segurança para o Application Load Balancer.

A atualização da política de segurança pode resultar em interrupções se o balanceador de carga estiver lidando com um alto volume de tráfego. Para diminuir a possibilidade de interrupções quando seu balanceador de carga está lidando com um grande volume de tráfego, crie um balanceador de carga adicional para ajudar a lidar com o tráfego ou solicite uma reserva de LCU.

Usando políticas FIPS em seu Application Load Balancer

Todos os receptores seguros conectados a um Application Load Balancer devem usar políticas de segurança do FIPS ou políticas de segurança que não sejam do FIPS; elas não podem ser misturadas. Se um Application Load Balancer existente tiver dois ou mais receptores usando políticas que não sejam do FIPS e você quiser que eles usem políticas de segurança do FIPS, remova todos os receptores até que reste apenas um. Altere a política de segurança do receptor para FIPS e crie mais receptores usando políticas de segurança do FIPS. Como alternativa, você pode criar um Application Load Balancer com novos receptores usando somente políticas de segurança do FIPS.

Para atualizar a política de segurança usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Receptores e regras, selecione o texto na coluna Protocolo:Porta para abrir a página de detalhes do receptor.
5. Na página Detalhes, escolha Ações e Editar receptor.
6. Na seção Configurações de receptor seguro, em Política de segurança, escolha uma nova política de segurança.
7. Escolha Salvar alterações.

Para atualizar a política de segurança usando o AWS CLI

Use o comando modify-listener.

modificação do cabeçalho HTTP

A modificação do cabeçalho HTTP permite renomear cabeçalhos específicos gerados pelo balanceador de carga, inserir cabeçalhos de resposta específicos e desativar o cabeçalho de resposta do servidor. Os Application Load Balancers oferecem suporte à modificação de cabeçalhos para cabeçalhos de solicitação e resposta.

Para obter mais informações, consulte Habilite a modificação do cabeçalho HTTP para seu Application Load Balancer.

Autenticação mútua com TLS no Application Load Balancer

A autenticação mútua com TLS é uma variação do Transport Layer Security (TLS). O TLS tradicional estabelece comunicações seguras entre um servidor e um cliente, nas quais o servidor precisa fornecer sua identidade aos clientes. Com o TLS mútuo, um balanceador de carga negocia a autenticação mútua entre o cliente e o servidor enquanto negocia o TLS. Ao usar o TLS mútuo com o Application Load Balancer, você simplifica o gerenciamento da autenticação e reduz a carga nas aplicações.

Ao usar o TLS mútuo com o Application Load Balancer, o balanceador de carga pode gerenciar a autenticação do cliente para ajudar a garantir que somente clientes confiáveis se comuniquem com suas aplicações de backend. Quando você usa esse recurso, o Application Load Balancer autentica clientes com certificados de uma autoridade de certificação (CA) terceirizada ou usando o AWS Private Certificate Authority (PCA), opcionalmente, com verificações de revogação. O Application Load Balancer transmite informações de certificado do cliente para o backend, que suas aplicações podem usar para autorização. Ao usar o TLS mútuo no Application Load Balancer, você pode obter autenticação integrada, escalável e gerenciada para entidades baseadas em certificados, que usam bibliotecas estabelecidas.

O TLS mútuo para Application Load Balancers fornece as seguintes duas opções para validar seus certificados de cliente X.509v3:

Observação: não há suporte para certificados de cliente X.509v1.

- Passagem de TLS mútuo: ao usar o modo de passagem de TLS mútuo, o Application Load Balancer envia toda a cadeia de certificados do cliente para o destino usando cabeçalhos HTTP. Em seguida, usando a cadeia de certificados do cliente, você pode implementar a autenticação do balanceador de carga correspondente e a lógica de autorização de destino na aplicação.
- Verificação de TLS mútuo: ao usar o modo de verificação de TLS mútuo, o Application Load Balancer executa a autenticação de certificado de cliente X.509 para clientes quando um balanceador de carga negocia conexões TLS.

Para começar a usar o TLS mútuo no Application Load Balancer com a passagem, você só precisa configurar o receptor para aceitar qualquer certificado dos clientes. Para usar o TLS mútuo com a verificação, faça o seguinte:

- Crie um recurso de armazenamento confiável.

- Faça upload do seu pacote de autoridade de certificação (CA) e, opcionalmente, das listas de revogação.
- Anexe o armazenamento confiável ao receptor que está configurado para verificar os certificados do cliente.

Para obter step-by-step os procedimentos para configurar o modo de verificação mútua de TLS com seu Application Load Balancer, consulte. Configuração de um TLS mútuo em um Application Load Balancer

Antes de começar a configurar o TLS mútuo no Application Load Balancer

Antes de começar a configurar o TLS mútuo no Application Load Balancer, esteja ciente do seguinte:

Cotas

Os Application Load Balancers incluem certos limites relacionados à quantidade de armazenamentos confiáveis, certificados de CA e listas de revogação de certificados em uso em sua conta. AWS

Para obter mais informações, consulte Quotas for your Application Load Balancers.

Requisitos de certificados

Os Application Load Balancers são compatíveis com os seguintes recursos para certificados usados com a autenticação de TLS mútuo:

- Certificado compatível: X.509v3
- Chaves públicas com suporte: RSA 2K a 8K ou ECDSA secp256r1, secp384r1, secp521r1
- Algoritmos de assinatura suportados: SHA256 384.512 com RSA/SHA256, 384, 512 with EC/SHA 256.384.512 hash com RSASSA-PSS com MGF1

Pacotes de certificados de CA

O seguinte se aplica aos pacotes de autoridade de certificação (CA):

- Os Application Load Balancers fazem o upload de cada pacote de certificados de autoridade de certificação (CA) em lote. Os Application Load Balancers não oferecem suporte ao upload de certificados individuais. Se precisar adicionar novos certificados, você deverá fazer upload do arquivo do pacote de certificados.
- Para substituir um pacote de certificados CA, use a ModifyTrustStoreAPI.

Pedido de certificado para passagem

Ao usar a passagem de TLS mútuo, o Application Load Balancer insere cabeçalhos para apresentar a cadeia de certificados do cliente aos destinos de backend. A ordem de apresentação começa com os certificados folha e termina com o certificado raiz.

Retomada da sessão

Não há suporte para a retomada da sessão ao usar os modos de passagem ou de verificação de TLS mútuo com um Application Load Balancer.

Cabeçalhos HTTP

Os Application Load Balancers usam cabeçalhos `X-Amzn-Mtls` para enviar informações do certificado quando negociam conexões de clientes usando TLS mútuo. Para obter mais informações e exemplos de cabeçalhos, consulte Cabeçalhos HTTP e TLS mútuo.

Arquivos de certificado de CA

Os arquivos de certificado de CA devem atender aos seguintes requisitos:

- O arquivo do certificado deve usar o formato PEM (Privacy Enhanced Mail).
- O conteúdo do certificado deve estar dentro dos limites `-----BEGIN CERTIFICATE-----` e `-----END CERTIFICATE-----`.
- Os comentários devem ser precedidos por um caractere # e não devem conter nenhum caractere `-`.
- Não pode haver linhas em branco.

Exemplo de certificado que não é aceito (inválido):

```
# comments

Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 01
    Signature Algorithm: ecdsa-with-SHA384
        Issuer: C=US, O=EXAMPLE, OU=EXAMPLE, CN=EXAMPLE
        Validity
            Not Before: Jan 11 23:57:57 2024 GMT
            Not After : Jan 10 00:57:57 2029 GMT
        Subject: C=US, O=EXAMPLE, OU=EXAMPLE, CN=EXAMPLE
        Subject Public Key Info:
```

```
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (384 bit)
                pub:
                    00:01:02:03:04:05:06:07:08
                ASN1 OID: secp384r1
                NIST CURVE: P-384
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment, Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Subject Key Identifier:
                00:01:02:03:04:05:06:07:08
            X509v3 Subject Alternative Name:
                URI:EXAMPLE.COM
    Signature Algorithm: ecdsa-with-SHA384
         00:01:02:03:04:05:06:07:08
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

Exemplos de certificados que são aceitos (válidos):

1. Certificado único (codificado por PEM):

```
# comments
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

2. Vários certificados (codificados por PEM):

```
# comments
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
# comments
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

Cabeçalhos HTTP e TLS mútuo

Esta seção descreve os cabeçalhos HTTP que os Application Load Balancers usam para enviar informações de certificado ao negociar conexões com clientes usando TLS mútuo. Os cabeçalhos `X-Amzn-Mtls` específicos usados pelo Application Load Balancer dependem do modo TLS mútuo especificado: modo de passagem ou modo de verificação.

Para obter informações sobre outros cabeçalhos HTTP compatíveis com os Application Load Balancers, consulte Cabeçalhos HTTP e Application Load Balancers.

Cabeçalho HTTP para o modo de passagem

Para TLS mútuo no modo de passagem, os Application Load Balancers usam o cabeçalho a seguir.

X-Amzn-Mtls-Clientcert

Esse cabeçalho contém o formato PEM codificado por URL de toda a cadeia de certificados do cliente apresentada na conexão, tendo `+=/` como caracteres seguros.

Exemplo de conteúdo do cabeçalho:

```
X-Amzn-Mtls-Clientcert: -----BEGIN%20CERTIFICATE-----%0AMIID<...reduced...>do0g
%3D%3D%0A-----END%20CERTIFICATE-----%0A-----BEGIN%20CERTIFICATE-----
%0AMIID1<...reduced...>3eZlyKA%3D%3D%0A-----END%20CERTIFICATE-----%0A
```

Cabeçalhos HTTP para o modo de verificação

Para TLS mútuo no modo de verificação, os Application Load Balancers usam os cabeçalhos a seguir.

X-Amzn-Mtls-Clientcert-Serial-Number

Esse cabeçalho contém uma representação hexadecimal do número de série do certificado folha.

Exemplo de conteúdo do cabeçalho:

```
X-Amzn-Mtls-Clientcert-Serial-Number: 03A5B1
```

X-Amzn-Mtls-Clientcert-Issuer

Esse cabeçalho contém uma representação em RFC2253 sequência do nome distinto (DN) do emissor.

Exemplo de conteúdo do cabeçalho:

```
X-Amzn-Mtls-Clientcert-Issuer:
 CN=rootcamtls.com,OU=rootCA,O=mTLS,L=Seattle,ST=Washington,C=US
```

X-Amzn-Mtls-Clientcert-Subject

Esse cabeçalho contém uma representação em RFC2253 sequência do nome distinto (DN) do assunto.

Exemplo de conteúdo do cabeçalho:

```
X-Amzn-Mtls-Clientcert-Subject: CN=client_.com,OU=client-3,O=mTLS,ST=Washington,C=US
```

X-Amzn-Mtls-Clientcert-Validity

Esse cabeçalho contém um formato ISO86 01 da `notAfter` data `notBefore` e.

Exemplo de conteúdo do cabeçalho:

```
X-Amzn-Mtls-Clientcert-Validity:
 NotBefore=2023-09-21T01:50:17Z;NotAfter=2024-09-20T01:50:17Z
```

X-Amzn-Mtls-Clientcert-Leaf

Esse cabeçalho contém um formato PEM codificado por URL do certificado folha, tendo `+=/` como caracteres seguros.

Exemplo de conteúdo do cabeçalho:

```
X-Amzn-Mtls-Clientcert-Leaf: -----BEGIN%20CERTIFICATE-----%0AMIIG<...reduced...>NmrUlw
%0A-----END%20CERTIFICATE-----%0A
```

Anuncie o nome do assunto da Autoridade Certificadora (CA)

Os nomes de assunto da Advertising Certificate Authority (CA) aprimoram o processo de autenticação ajudando os clientes a determinar quais certificados serão aceitos durante a autenticação TLS mútua.

Quando você ativa Anunciar nomes de assunto da CA, o Application Load Balancer anunciará a lista de nomes de assuntos de Autoridades Certificadoras CAs () em que confia, com base no

armazenamento confiável ao qual está associado. Quando um cliente se conecta a um destino por meio do Application Load Balancer, o cliente recebe a lista de nomes de assuntos confiáveis da CA.

Durante o handshake TLS, quando o Application Load Balancer solicita um certificado de cliente, ele inclui uma lista de CA Distinguished Names DNs () confiáveis em sua mensagem de Solicitação de Certificado. Isso ajuda os clientes a selecionar certificados válidos que correspondam aos nomes de assunto da CA anunciados, simplificando o processo de autenticação e reduzindo os erros de conexão.

Você pode ativar o nome do assunto Anuncie CA em ouvintes novos e existentes. Para obter mais informações, consulte Adicionar um receptor HTTPS.

Logs de conexão para Application Load Balancers

O Elastic Load Balancing fornece logs de conexão que capturam atributos sobre as solicitações enviadas aos Application Load Balancers. Os logs de conexão contêm informações como o endereço IP e a porta do cliente, informações sobre o certificado do cliente, resultados da conexão e cifras TLS que estão sendo usadas. Esses logs de conexão podem então ser usados para revisar padrões de solicitação e outras tendências.

Para saber mais sobre os logs de conexão, consulte Logs de conexão para o Application Load Balancer

Configuração de um TLS mútuo em um Application Load Balancer

Esta seção inclui os procedimentos de como configurar o modo de verificação de TLS mútuo para autenticação em Application Load Balancers.

Para usar o modo de passagem de TLS mútuo, você só precisa configurar o receptor para aceitar qualquer certificado de clientes. Ao usar a passagem de TLS mútuo, o Application Load Balancer envia toda a cadeia de certificados do cliente para o destino usando cabeçalhos HTTP, o que permite implementar a lógica correspondente de autenticação e autorização na aplicação. Para obter mais informações, consulte Criar um receptor HTTPS para seu Application Load Balancer.

Ao usar o TLS mútuo no modo de verificação, o Application Load Balancer executa a autenticação de certificado de cliente X.509 para clientes quando um balanceador de carga negocia conexões TLS.

Para utilizar o modo de verificação de TLS mútuo, faça o seguinte:

- Crie um recurso de armazenamento confiável.

- Faça upload do seu pacote de autoridade de certificação (CA) e, opcionalmente, das listas de revogação.
- Anexe o armazenamento confiável ao receptor que está configurado para verificar os certificados do cliente.

Siga os procedimentos desta seção para configurar o modo de verificação de TLS mútuo no Application Load Balancer no AWS Management Console. Para configurar o TLS mútuo usando operações de API em vez do console, consulte o Guia de referência da API do Application Load Balancer.

Tarefas

- Criar um armazenamento confiável
- Associar um armazenamento confiável
- Exibir detalhes do armazenamento confiável
- Modificar um armazenamento confiável
- Excluir um armazenamento confiável

Criar um armazenamento confiável

Há três maneiras de criar um armazenamento confiável: ao criar um Application Load Balancer, ao criar um receptor seguro e ao usar o console do armazenamento confiável. Quando você adiciona um armazenamento confiável ao criar um balanceador de carga ou receptor, o armazenamento confiável é automaticamente associado ao novo receptor. Ao criar um armazenamento confiável usando o console, é você quem deve associá-lo a um receptor.

Esta seção aborda a criação de um armazenamento confiável usando o console, mas as etapas usadas ao criar um Application Load Balancer ou receptor são as mesmas. Para obter mais informações, consulte Configure a load balancer and a listener e Create an HTTPS listener.

Pré-requisitos:

- Para criar um armazenamento confiável, você deve ter um pacote de certificados da sua autoridade de certificação (CA).

Para criar um armazenamento confiável usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.

2. No painel de navegação, escolha Armazenamentos confiáveis.
3. Escolha Criar repositório fiduciário.
4. Configuração do armazenamento confiável

   a. Em Nome do armazenamento confiável, insira um nome para o armazenamento confiável.

   b. Em Pacote da autoridade de certificação, insira o caminho do Amazon S3 para o pacote de certificados da CA que o armazenamento confiável deve usar.

      Opcional: use a Versão do objeto para selecionar uma versão anterior do pacote de certificados da CA. Caso contrário, a versão atual será usada.
5. (Opcional) Para revogações, você pode adicionar uma lista de revogação de certificados ao seu repositório confiável.

   - Escolha Adicionar nova CRL e insira a localização da lista de revogação de certificados no Amazon S3.

     Opcional: use a Versão do objeto para selecionar uma versão anterior da lista de revogação de certificados. Caso contrário, a versão atual será usada.
6. (Opcional) Para as tags da Trust Store, você pode inserir até 50 tags para aplicar à sua Trust Store.
7. Escolha Criar repositório fiduciário.

Associar um armazenamento confiável

Depois de criar um armazenamento confiável, você deve associá-lo a um receptor antes que o Application Load Balancer possa começar a usá-lo. Você pode ter somente um armazenamento confiável associado a cada um dos receptores seguros, mas um armazenamento confiável pode ser associado a vários receptores.

Esta seção aborda a associação de um armazenamento confiável a um receptor existente. Como alternativa, você pode associar um armazenamento confiável ao criar um Application Load Balancer ou um receptor. Para obter mais informações, consulte Configure a load balancer and a listener e Create an HTTPS listener.

Para associar um armazenamento confiável usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.

2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Receptores e regras, escolha o link na coluna Protocol:Port para abrir a página de detalhes do receptor seguro.
5. Na guia Segurança, escolha Editar configurações de receptor seguro.
6. Se o TLS mútuo não estiver habilitado, selecione Autenticação mútua (mTLS) em Tratamento de certificados do cliente e escolha Verificar com armazenamento confiável.
7. Em Trust store, escolha o trust store.
8. Escolha Salvar alterações.

Exibir detalhes do armazenamento confiável

Pacotes de certificados de CA

O pacote de certificados da CA é um componente obrigatório do armazenamento confiável. É uma coleção de certificados raiz e intermediários confiáveis que foram validados por uma autoridade de certificação. Esses certificados validados garantem que o cliente possa confiar que o certificado apresentado pertence ao balanceador de carga.

Você pode exibir o conteúdo do pacote de certificados de CA atual no armazenamento confiável a qualquer momento.

Exibir um pacote de certificados de CA

Para exibir um pacote de certificado de CA usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Armazenamentos confiáveis.
3. Selecione o armazenamento confiável para exibir a página de detalhes.
4. Escolha Ações e Obter pacote de CA.
5. Escolha Compartilhar link ou Download.

Listas de revogação de certificados

Opcionalmente, você pode criar uma lista de revogação de certificados para um armazenamento confiável. As listas de revogação são divulgadas pelas autoridades de certificação e contêm dados

de certificados que foram revogados. Os Application Load Balancers só oferecem suporte a listas de revogação de certificados no formato PEM.

Quando uma lista de revogação de certificados é adicionada a um armazenamento confiável, ela recebe um ID de revogação. O IDs aumento de revogação para cada lista de revogação adicionada ao armazenamento confiável e elas não podem ser alteradas. Se uma lista de revogação de certificados for excluída de um armazenamento confiável, o ID de revogação também será excluído e não será reutilizado durante toda a vida útil do armazenamento confiável.

> **Note**
>
> Os Application Load Balancers não podem revogar certificados que tenham um número de série negativo em uma lista de revogação de certificados.

Exibir uma lista de revogação de certificados

Para exibir uma lista de revogação usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Armazenamentos confiáveis.
3. Selecione o armazenamento confiável para exibir a página de detalhes.
4. Na guia Listas de revogação de certificados, selecione Ações e Obter lista de revogação.
5. Escolha Compartilhar link ou Download.

Modificar um armazenamento confiável

Um armazenamento confiável pode conter somente um pacote de certificados de CA por vez, mas você pode substituir o pacote de certificados de CA a qualquer momento após a criação do armazenamento confiável.

Substituir um pacote de certificados de CA

Para substituir um pacote de certificados de CA usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Armazenamentos confiáveis.

3. Selecione o repositório confiável.
4. Escolha Ações e Substituir pacote de CA.
5. Na página Substituir pacote de CA, em Pacote de autoridade de certificação, insira a localização do Amazon S3 do pacote de CA desejado.
6. (Opcional) Use a Versão do objeto para selecionar uma versão anterior da lista de revogação de certificados. Caso contrário, a versão atual será usada.
7. Selecione Substituir pacote de CA.

Adicionar uma lista de revogação de certificados

Para adicionar uma lista de revogações usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Armazenamentos confiáveis.
3. Selecione o armazenamento confiável para exibir a página de detalhes.
4. Na guia Listas de revogação de certificados, selecione Ações e Adicionar lista de revogação.
5. Na página Adicionar lista de revogação, em Lista de revogação de certificados, insira a localização do Amazon S3 da lista de revogação de certificados desejada
6. (Opcional) Use a Versão do objeto para selecionar uma versão anterior da lista de revogação de certificados. Caso contrário, a versão atual será usada.
7. Selecione Adicionar lista de revogação

Excluir uma lista de revogação de certificados

Para excluir uma lista de revogação usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Armazenamentos confiáveis.
3. Selecione o repositório confiável.
4. Na guia Listas de revogação de certificados, escolha Ações e, em seguida, Excluir lista de revogação.
5. Quando a confirmação for solicitada, insira **`confirm`**.
6. Escolha Excluir.

Excluir um armazenamento confiável

Quando não precisar mais usar um armazenamento confiável, poderá excluí-lo.

Observação: você não pode excluir um armazenamento confiável atualmente associado a um receptor.

Para excluir um armazenamento confiável usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Armazenamentos confiáveis.
3. Selecione o armazenamento confiável para exibir a página de detalhes.
4. Escolha Ações e, em seguida, Excluir armazenamento confiável.
5. Confirme a exclusão digitando `confirm`.
6. Selecione Excluir

Compartilhar o armazenamento confiável do Elastic Load Balancing para Application Load Balancers

O Elastic Load Balancing se integra com AWS Resource Access Manager (AWS RAM) para permitir o compartilhamento do armazenamento confiável. AWS RAM é um serviço que permite que você compartilhe com segurança seus recursos de armazenamento fiduciário do Elastic Load Balancing Contas da AWS entre e dentro de sua organização ou unidades organizacionais (). OUs Caso tenha várias contas, poderá criar um armazenamento confiável uma vez e usar o AWS RAM para torná-lo utilizável por outras contas. Se sua conta for gerenciada por AWS Organizations, você poderá compartilhar armazenamentos fiduciários com todas as contas da organização ou somente com contas dentro de unidades organizacionais especificadas (OUs).

Com AWS RAM, você compartilha recursos de sua propriedade criando um compartilhamento de recursos. Um compartilhamento de atributos especifica os atributos a serem compartilhados, e os consumidores com os quais compartilhá-los. Nesse modelo, o Conta da AWS proprietário da loja fiduciária (proprietário) a compartilha com outros Contas da AWS (consumidores). Os consumidores podem associar armazenamentos de confiança compartilhados aos receptores do Application Load Balancer da mesma forma que associam armazenamentos confiáveis em suas próprias contas.

O proprietário de um armazenamento confiável pode compartilhar um armazenamento confiável com:

- Específico Contas da AWS dentro ou fora de sua organização em AWS Organizations

- Uma unidade organizacional dentro de sua organização em AWS Organizations
- Toda a sua organização em AWS Organizations

Conteúdo

- Pré-requisitos para o compartilhamento de armazenamentos confiáveis
- Permissões para armazenamentos confiáveis compartilhados
- Compartilhar um armazenamento confiável
- Parar de compartilhar um armazenamento confiável
- Faturamento e medição

Pré-requisitos para o compartilhamento de armazenamentos confiáveis

- Você deve criar um compartilhamento de recursos usando AWS Resource Access Manager. Para obter mais informações, consulte Create a resource share no Guia do usuário do AWS RAM .
- Para compartilhar uma loja fiduciária, você deve possuí-la em seu Conta da AWS. Não é possível compartilhar um armazenamento confiável que tenha sido compartilhado com você.
- Para compartilhar um armazenamento confiável com sua organização ou unidade organizacional no AWS Organizations, é preciso habilitar o compartilhamento no AWS Organizations. Para obter mais informações, consulte Habilitar o compartilhamento com o AWS Organizations no Guia do usuário do AWS RAM .

Permissões para armazenamentos confiáveis compartilhados

Proprietários de armazenamentos confiáveis

- Proprietários de armazenamentos confiáveis podem criar um armazenamento confiável.
- Proprietários de armazenamentos confiáveis podem usar um armazenamento confiável com balanceadores de carga na mesma conta.
- Os proprietários de lojas fiduciárias podem compartilhar uma loja fiduciária com outras AWS contas ou AWS Organizations.
- Os proprietários de lojas fiduciárias podem cancelar o compartilhamento de uma loja confiável de qualquer AWS conta ou AWS Organizations.
- Proprietários de armazenamentos confiáveis não podem impedir que balanceadores de carga usem um armazenamento confiável na mesma conta.

- Os proprietários de armazenamentos confiáveis podem listar todos os Application Load Balancers usando um armazenamento confiável compartilhado.
- Os proprietários de armazenamentos confiáveis podem excluir um armazenamento confiável se não houver associações atuais.
- Os proprietários de armazenamentos confiáveis podem excluir associações com um armazenamento confiável compartilhado.
- Os proprietários de lojas fiduciárias recebem CloudTrail registros quando uma loja confiável compartilhada é usada.

Consumidores de armazenamentos confiáveis

- Os consumidores de armazenamentos confiáveis podem exibir armazenamentos confiáveis compartilhados.
- Os consumidores de armazenamentos confiáveis podem criar ou modificar receptores usando um armazenamento confiável na mesma conta.
- Os consumidores de armazenamentos confiáveis podem criar ou modificar receptores usando um armazenamento confiável compartilhado.
- Os consumidores de armazenamentos confiáveis não podem criar um receptor usando um armazenamento confiável que não é mais compartilhado.
- Os consumidores de armazenamentos confiáveis não podem modificar um armazenamento confiável compartilhado.
- Os consumidores de armazenamentos confiáveis podem exibir o ARN de um armazenamento confiável compartilhado quando associados a um receptor.
- Os consumidores da loja confiável recebem CloudTrail registros ao criar ou modificar um ouvinte usando uma loja confiável compartilhada.

Permissões gerenciadas

Ao compartilhar um armazenamento confiável, o compartilhamento de recursos usa permissões gerenciadas para controlar quais ações são permitidas pelo consumidor do armazenamento confiável. Você pode usar as permissões gerenciadas padrão `AWSRAMPermissionElasticLoadBalancingTrustStore`, que incluem todas as permissões disponíveis, ou criar suas próprias permissões gerenciadas pelo cliente. As permissões `DescribeTrustStores`, `DescribeTrustStoreRevocations` e `DescribeTrustStoreAssociations` estão sempre habilitadas e não podem ser removidas.

As seguintes permissões são compatíveis com compartilhamentos de recursos do armazenamento confiável:

balanceamento elástico de carga: CreateListener

Pode anexar um armazenamento confiável compartilhado a um novo receptor.

balanceamento elástico de carga: ModifyListener

Pode anexar um armazenamento confiável compartilhado a um receptor existente.

balanceamento elástico de carga: GetTrustStoreCaCertificatesBundle

Pode baixar o pacote de certificados de CA associado ao armazenamento confiável compartilhado.

balanceamento elástico de carga: GetTrustStoreRevocationContent

Pode baixar o arquivo de revogação associado ao armazenamento confiável compartilhado.

elasticloadbalancing: (Padrão) DescribeTrustStores

Pode listar todos os armazenamentos confiáveis pertencentes e compartilhados com a conta.

elasticloadbalancing: (Padrão) DescribeTrustStoreRevocations

Pode listar todo o conteúdo de revogação de um determinado armazenamento confiável.

elasticloadbalancing: (Padrão) DescribeTrustStoreAssociations

Pode listar todos os recursos na conta do consumidor do armazenamento confiável que estão associados ao armazenamento confiável compartilhado.

Compartilhar um armazenamento confiável

Para compartilhar um armazenamento confiável, é necessário adicioná-lo a um compartilhamento de recursos. Um compartilhamento de recursos é um recurso do AWS RAM que permite que você compartilhe seus recursos entre Contas da AWS. Um compartilhamento de recursos especifica os recursos a serem compartilhados, os consumidores com os quais compartilhá-los e quais ações as entidades principais poderão executar. Ao compartilhar um repositório fiduciário usando o EC2 console da Amazon, você o adiciona a um compartilhamento de recursos existente. Para adicionar o armazenamento confiável a um novo compartilhamento de recursos, primeiramente você deve criar o compartilhamento de recursos usando o console do AWS RAM.

Ao compartilhar um repositório confiável que você possui com outras pessoas Contas da AWS, você permite que essas contas associem seus ouvintes do Application Load Balancer aos armazenamentos confiáveis em sua conta.

Se você faz parte de uma organização AWS Organizations e o compartilhamento dentro de sua organização está ativado, os consumidores em sua organização recebem automaticamente acesso ao armazenamento confiável compartilhado. Caso contrário, os consumidores receberão um convite para participar do compartilhamento de recursos e terão acesso ao armazenamento confiável compartilhado depois de aceitar o convite.

Você pode compartilhar uma loja fiduciária de sua propriedade usando o EC2 console da Amazon, o AWS RAM console ou AWS CLI o.

Para compartilhar uma loja fiduciária que você possui usando o EC2 console da Amazon

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Balanceamento de carga, escolha Armazenamentos confiáveis.
3. Selecione o nome do armazenamento confiável para exibir a página de detalhes.
4. Na guia Compartilhamento, escolha Compartilhar armazenamento confiável.
5. Na página Compartilhar armazenamento confiável, em Compartilhamentos de recursos, selecione com quais compartilhamentos de recursos seu armazenamento confiável será compartilhado.
6. (Opcional) Caso precise criar um novo compartilhamento de recursos, selecione o link Criar um compartilhamento de recursos no console do RAM.
7. Selecione Compartilhar armazenamento confiável.

Para compartilhar um repositório fiduciário que você possui usando o AWS RAM console

Consulte Criar um compartilhamento de recursos no Manual do usuário do AWS RAM .

Para compartilhar uma loja fiduciária que você possui usando o AWS CLI

Use o comando create-resource-share.

Parar de compartilhar um armazenamento confiável

Para interromper o compartilhamento de um armazenamento confiável de sua propriedade, remova-o do compartilhamento de recursos. As associações existentes persistem após você parar de compartilhar seu armazenamento confiável, no entanto, novas associações com um

armazenamento confiável compartilhado anteriormente não são permitidas. Quando o proprietário do armazenamento confiável ou o consumidor do armazenamento confiável excluir uma associação, ela será excluída de ambas as contas. Se um consumidor de armazenamento confiável quiser sair de um compartilhamento de recursos, ele deverá pedir ao proprietário do compartilhamento de recursos que remova a conta.

> ⚠ Excluir associações
>
> Os proprietários de lojas confiáveis podem excluir com força as associações existentes de lojas confiáveis usando o DeleteTrustStoreAssociationcomando. Quando uma associação é excluída, qualquer receptor do balanceador de carga que usa o armazenamento confiável não pode mais verificar os certificados do cliente e falhará nos handshakes de TLS.

Você pode parar de compartilhar uma loja fiduciária usando o EC2 console da Amazon, o AWS RAM console ou AWS CLI o.

Para parar de compartilhar uma loja fiduciária que você possui usando o EC2 console da Amazon

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Balanceamento de carga, escolha Armazenamentos confiáveis.
3. Selecione o nome do armazenamento confiável para exibir a página de detalhes.
4. Na guia Compartilhamento, em Compartilhamento de recursos, selecione os compartilhamentos de recursos com os quais interromper o compartilhamento.
5. Escolha Remover.

Para parar de compartilhar um repositório fiduciário que você possui usando o AWS RAM console

Consulte Atualização de um compartilhamento de atributos no Guia do usuário do AWS RAM .

Para parar de compartilhar uma loja confiável que você possui usando o AWS CLI

Use o comando disassociate-resource-share.

Faturamento e medição

Os armazenamentos confiáveis compartilhados têm a mesma taxa padrão de armazenamento confiável, cobrada por hora, por associação de armazenamento confiável com um Application Load Balancer.

Para obter mais informações, incluindo a taxa específica por região, consulte os preços do Elastic Load Balancing

Autenticar usuários usando um Application Load Balancer

Você pode configurar um Application Load Balancer para autenticar usuários com segurança conforme eles acessem suas aplicações. Com isso, você pode redirecionar o trabalho de autenticação de usuários para seu load balancer para que seus aplicativos possam se concentrar na respectiva lógica de negócios.

Os casos de uso a seguir são comportados:

- Autentique usuários por meio de um provedor de identidade (IdP) compatível com OpenID Connect (OIDC).
- Autentique usuários por meio de redes sociais IdPs, como Amazon, Facebook ou Google, por meio dos grupos de usuários suportados pelo Amazon Cognito.
- Autentique usuários por meio de identidades corporativas, usando SAML, OpenID Connect (OIDC) OAuth ou por meio dos grupos de usuários suportados pelo Amazon Cognito.

Preparação para usar um IdP compatível com OIDC

Faça o seguinte se você estiver usando um IdP compatível com OIDC com seu Application Load Balancer:

- Crie um novo aplicativo OIDC em seu IdP. O DNS do IdP deve ser resolvível publicamente.
- Você deve configurar um ID de cliente e um segredo de cliente.
- Obtenha os seguintes endpoints publicados pelo IdP: autorização, token e informações de usuário. Você pode localizar essa informação na configuração.
- Os certificados de endpoints do IdP devem ser emitidos por uma autoridade de certificação pública confiável.
- As entradas de DNS dos endpoints devem ser resolvíveis publicamente, mesmo que sejam resolvidas em endereços IP privados.
- Permita um dos seguintes redirecionamentos URLs em seu aplicativo IdP, qualquer que seja usado pelos usuários, em que DNS é o nome de domínio do seu balanceador de carga e CNAME é o alias de DNS do seu aplicativo:
  - *DNS*https://oauth2/idresponse

- *CNAME*https://oauth2/idresponse

Preparação para usar o Amazon Cognito

Regiões disponíveis

A integração do Amazon Cognito para Application Load Balancers está disponível nas seguintes regiões:

- Leste dos EUA (Norte da Virgínia)
- Leste dos EUA (Ohio)
- Oeste dos EUA (Norte da Califórnia)
- Oeste dos EUA (Oregon)
- Canadá (Central)
- Oeste do Canadá (Calgary)
- Europa (Estocolmo)
- Europa (Milão)
- Europa (Frankfurt)
- Europa (Zurique)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- Europa (Espanha)
- América do Sul (São Paulo)
- Ásia-Pacífico (Hong Kong)
- Ásia-Pacífico (Tóquio)
- Ásia-Pacífico (Seul)
- Asia Pacific (Osaka)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Hyderabad)
- Ásia-Pacífico (Singapura)

- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Jacarta)
- Ásia-Pacífico (Melbourne)
- Oriente Médio (Emirados Árabes Unidos)
- Oriente Médio (Bahrein)
- África (Cidade do Cabo)
- Israel (Tel Aviv)

Faça o seguinte se você estiver usando grupos de usuários do Amazon Cognito com seu Application Load Balancer:

- Criar um grupo de usuários. Para obter mais informações, consulte Grupos de usuários do Amazon Cognito no Guia do desenvolvedor do Amazon Cognito.
- Crie um cliente de grupo de usuários. Você deve configurar o cliente para gerar um segredo do cliente, usar o fluxo de concessão de código e oferecer suporte aos mesmos OAuth escopos usados pelo balanceador de carga. Para obter mais informações, consulte Configurar um cliente de aplicativo de grupo de usuários no Guia do desenvolvedor do Amazon Cognito.
- Crie um domínio de grupo de usuários. Para obter mais informações, consulte Configurar um domínio de grupo de usuários no Guia do Desenvolvedor do Amazon Cognito.
- Verifique se o escopo solicitado retorna um token de ID. Por exemplo, o escopo padrão, `openid` retorna um token de ID, mas o `aws.cognito.signin.user.admin` escopo não.
- Para federar com um IdP social ou corporativo, habilite o IdP na seção de federação. Para obter mais informações, consulte Login do grupo de usuários com um provedor de identidade terceirizado no Guia do Desenvolvedor do Amazon Cognito.
- Permita o seguinte redirecionamento URLs no campo URL de retorno de chamada do Amazon Cognito, em que DNS é o nome de domínio do seu balanceador de carga e CNAME é o alias de DNS do seu aplicativo (se você estiver usando um):
  - *DNS*https://oauth2/idresponse
  - *CNAME*https://oauth2/idresponse
- Permita o domínio do grupo de usuários no URL de retorno de chamada do aplicativo do IdP. Use o formato de seu IdP. Por exemplo:
  - https://*domain-prefix*.auth. *region*.amazoncognito. com/saml2/idpresponse
  - *user-pool-domain*https://saml2/idresponse

O URL de retorno de chamada nas configurações do aplicativo cliente deve usar todas as letras minúsculas.

Para permitir que um usuário configure um balanceador de carga para usar o Amazon Cognito para autenticar usuários, você deve conceder permissão ao usuário para chamar a ação `cognito-idp:DescribeUserPoolClient`.

Prepare-se para usar a Amazon CloudFront

Ative as seguintes configurações se você estiver usando uma CloudFront distribuição na frente do seu Application Load Balancer:

- Encaminhar cabeçalhos de solicitação (todos) — Garante que as respostas CloudFront não sejam armazenadas em cache para solicitações autenticadas. Isso impede que sejam exibidos no cache após a expiração da sessão de autenticação. Como alternativa, para reduzir esse risco enquanto o armazenamento em cache está ativado, os proprietários de uma CloudFront distribuição podem definir que o valor time-to-live (TTL) expire antes que o cookie de autenticação expire.
- Encaminhamento e armazenamento em cache de string de consulta (todos): garante que o balanceador de carga tenha acesso aos parâmetros da string de consulta necessários para autenticar o usuário com o IdP.
- Encaminhamento de cookies (todos) — Garante o CloudFront encaminhamento de todos os cookies de autenticação para o balanceador de carga.
- Ao configurar a autenticação do OpenID Connect (OIDC) em conjunto com a CloudFront Amazon, certifique-se de que a porta HTTPS 443 seja usada consistentemente em todo o caminho de conexão. Caso contrário, falhas de autenticação podem ocorrer porque o redirecionamento do OIDC do cliente URLs não corresponde ao número da porta do URI gerado originalmente.

Configurar a autenticação de usuários

Você configura a autenticação de usuários criando uma ação de autenticação para uma ou mais regras do listener. Os tipos de ação `authenticate-cognito` e `authenticate-oidc` são comportados somente por listeners HTTPS. Para obter descrições dos campos correspondentes, consulte AuthenticateCognitoActionConfige AuthenticateOidcActionConfigna versão de referência da API Elastic Load Balancing 2015-12-01.

O load balancer envia um cookie de sessão para o cliente a fim de manter o status de autenticação. Esse cookie sempre contém o atributo `secure`, pois a autenticação do usuário requer um listener

HTTPS. Esse cookie contém o atributo `SameSite=None` com solicitações CORS (cross-origin resource sharing, compartilhamento de recursos de origem cruzada).

Para um balanceador de carga compatível com várias aplicações que exigem autenticação independente de cliente, cada regra de receptor com uma ação de autenticação deve ter um nome de cookie exclusivo. Isso garante que os clientes sempre sejam autenticados com o IdP antes de serem roteados para o grupo de destino especificado na regra.

Os Application Load Balancers não são compatíveis com valores de cookie codificados por URL.

Por padrão, o campo `SessionTimeout` é definido com 7 dias. Se desejar sessões menores, pode configurar um tempo limite de sessão de 1 segundo. Para obter mais informações, consulte Tempo limite de sessão.

Defina o campo `OnUnauthenticatedRequest` de acordo com seu aplicativo. Por exemplo:

- Aplicações que exigem que o usuário faça login usando uma identidade social ou corporativa: viabilizado pela opção padrão, `authenticate`. Se o usuário não estiver conectado, o load balancer redirecionará a solicitação para o endpoint de autorização do IdP e o IdP solicitará que o usuário faça login usando sua interface de usuário.
- Aplicações que oferecem uma visualização personalizada para um usuário que está conectado ou uma visualização geral para um usuário que não está conectado: para viabilizar esse tipo de aplicação, use a opção `allow`. Se o usuário estiver conectado, o load balancer fornecerá as solicitações do usuário e o aplicativo poderá oferecer uma visualização personalizada. Se o usuário não estiver conectado, o load balancer encaminhará a solicitação sem as solicitações do usuário e o aplicativo poderá oferecer uma visualização geral.
- Aplicativos de página única com carregamento JavaScript a cada poucos segundos — se você usar a `deny` opção, o balanceador de carga retornará um erro HTTP 401 não autorizado para chamadas AJAX que não têm informações de autenticação. Porém, se o usuário tiver informações de autenticação expiradas, ele redirecionará o cliente para o endpoint de autorização do IdP.

O load balancer precisa se comunicar com o endpoint de token do IdP (`TokenEndpoint`) e o endpoint de informações do usuário do Idp (`UserInfoEndpoint`). Os Application Load Balancers só são compatíveis com IPv4 a comunicação com esses endpoints. Se seu IdP usa endereços públicos, garanta que os grupos de segurança do seu balanceador de carga e a rede da ACLs sua VPC permitam acesso aos endpoints. Ao usar um balanceador de carga interno ou o tipo de endereço IP `dualstack-without-public-ipv4`, um gateway NAT pode permitir que o

balanceador de carga se comunique com os endpoints. Para obter mais informações, consulte Fundamentos de gateway NAT no Guia do usuário da Amazon VPC.

Use o comando create-rule a seguir para configurar a autenticação de usuários.

```
aws elbv2 create-rule --listener-arn listener-arn --priority 10 \
--conditions Field=path-pattern,Values="/login" --actions file://actions.json
```

Veja a seguir um exemplo do arquivo `actions.json` que especifica uma ação `authenticate-oidc` e uma ação `forward`. `AuthenticationRequestExtraParams` permite que você transmita parâmetros extras para um IdP durante a autenticação. Siga a documentação fornecida pelo seu provedor de identidade para determinar os campos que são compatíveis

```
[{
    "Type": "authenticate-oidc",
    "AuthenticateOidcConfig": {
        "Issuer": "https://idp-issuer.com",
        "AuthorizationEndpoint": "https://authorization-endpoint.com",
        "TokenEndpoint": "https://token-endpoint.com",
        "UserInfoEndpoint": "https://user-info-endpoint.com",
        "ClientId": "abcdefghijklmnopqrstuvwxyz123456789",
        "ClientSecret": "12345678901234567890123456789O",
        "SessionCookieName": "my-cookie",
        "SessionTimeout": 3600,
        "Scope": "email",
        "AuthenticationRequestExtraParams": {
            "display": "page",
            "prompt": "login"
        },
        "OnUnauthenticatedRequest": "deny"
    },
    "Order": 1
},
{
    "Type": "forward",
    "TargetGroupArn": "arn:aws:elasticloadbalancing:region-code:account-
id:targetgroup/target-group-name/target-group-id",
    "Order": 2
}]
```

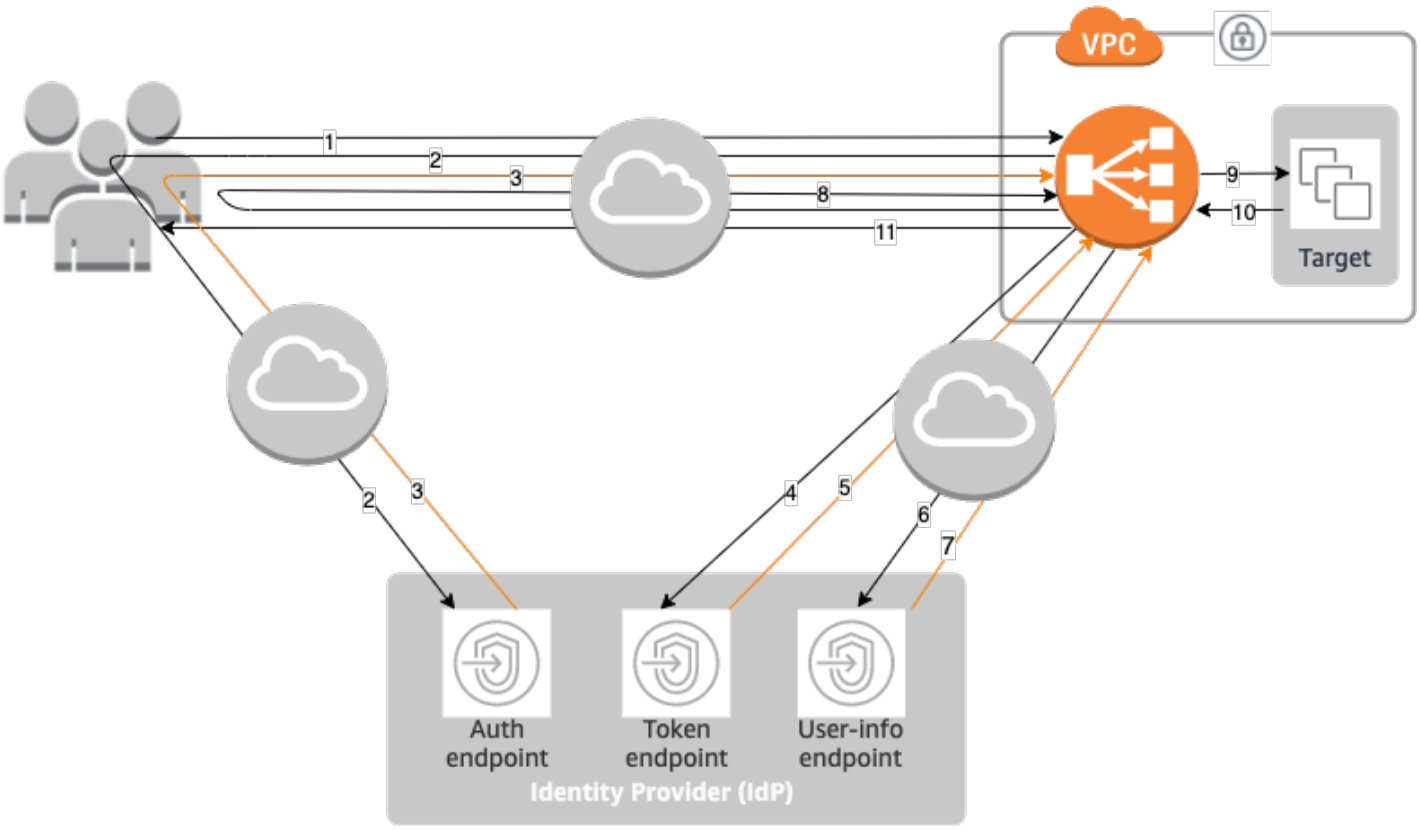
Veja a seguir um exemplo de configuração do arquivo `actions.json`, que especifica uma ação `authenticate-cognito` e uma ação `forward`.

```
[{
    "Type": "authenticate-cognito",
    "AuthenticateCognitoConfig": {
        "UserPoolArn": "arn:aws:cognito-idp:region-code:account-id:userpool/user-pool-
id",
        "UserPoolClientId": "abcdefghijklmnopqrstuvwxyz123456789",
        "UserPoolDomain": "userPoolDomain1",
        "SessionCookieName": "my-cookie",
        "SessionTimeout": 3600,
        "Scope": "email",
        "AuthenticationRequestExtraParams": {
            "display": "page",
            "prompt": "login"
        },
        "OnUnauthenticatedRequest": "deny"
    },
    "Order": 1
},
{
    "Type": "forward",
    "TargetGroupArn": "arn:aws:elasticloadbalancing:region-code:account-
id:targetgroup/target-group-name/target-group-id",
    "Order": 2
}]
```

Para obter mais informações, consulte Regras do receptor para seu Application Load Balancer.

Fluxo de autenticação

O diagrama de rede a seguir é uma representação visual de como um Application Load Balancer usa o OIDC para autenticar usuários.



Os itens numerados abaixo destacam e explicam os elementos apresentados no diagrama de rede anterior.

1. O usuário envia uma solicitação HTTPS para um site hospedado atrás de um Application Load Balancer. Quando as condições para uma regra com uma ação de autenticação são atendidas, o load balancer verifica a existência de um cookie de sessão de autenticação nos cabeçalhos de solicitação.
2. Se o cookie não estiver presente, o load balancer redirecionará o usuário para o endpoint de autorização do IdP para que o IdP possa autenticar o usuário.
3. Depois que o usuário é autenticado, o IdP envia o usuário de volta para o balanceador de carga com um código de concessão de autorização.
4. O balanceador de carga apresenta o código de concessão de autorização ao endpoint do token do IdP.
5. Ao receber um código de concessão de autorização válido, o IdP fornece o token de ID e o token de acesso ao Application Load Balancer.
6. Em seguida, o Application Load Balancer envia o token de acesso ao endpoint de informações do usuário.

7. O endpoint de informações do usuário troca o token de acesso pelas reivindicações do usuário.
8. O Application Load Balancer redireciona o usuário com o cookie de sessão de autenticação `AWSELB` para o URI original. Como a maioria dos navegadores restringe o tamanho dos cookies a 4 K, o balanceador de carga fragmenta cookies com tamanho superior a 4 K em vários cookies. Se o tamanho total das solicitações do usuário e do token de acesso recebidos do IdP for superior a 11K bytes, o load balancer retornará um erro HTTP 500 para o cliente e incrementará a métrica `ELBAuthUserClaimsSizeExceeded`.
9. O Application Load Balancer valida o cookie e encaminha as informações do usuário para destinos no conjunto de cabeçalhos HTTP `X-AMZN-OIDC-*`. Para obter mais informações, consulte Verificação de assinatura e codificação de reivindicações de usuário.
10. O destino envia uma resposta de volta ao Application Load Balancer.
11. O Application Load Balancer envia a resposta final ao usuário.

Cada nova solicitação passa pelas etapas de 1 a 11, enquanto as solicitações subsequentes passam pelas etapas de 9 a 11. Ou seja, todas as solicitações subsequentes começam na etapa 9, desde que o cookie não tenha expirado.

O cookie `AWSALBAuthNonce` é adicionado ao cabeçalho da solicitação depois que o usuário fizer autenticação no IdP. Isso não muda a forma como o Application Load Balancer processa as solicitações de redirecionamento do IdP.

Se o IdP fornecer um token de atualização válido no token de ID, o load balancer salvará o token de atualização e o usará para atualizar as solicitações do usuário toda vez que o token de acesso expirar, até o momento em que a sessão expirar ou a atualização do IdP falhar. Se o usuário encerrar a sessão, a atualização falhará e o load balancer o redirecionará para o endpoint de autorização do IdP. Isso permite que o load balancer suspenda as sessões depois que o usuário encerra a sessão. Para obter mais informações, consulte Tempo limite de sessão.

> **Note**
>
> A expiração do cookie é diferente da expiração da sessão de autenticação. A expiração do cookie é um atributo do cookie e que está definido para 7 dias. A duração real da sessão de autenticação é determinada pelo tempo limite da sessão configurado no Application Load Balancer para o recurso de autenticação. O tempo limite dessa sessão está incluído no valor do cookie Auth, que também é criptografado.

Verificação de assinatura e codificação de reivindicações de usuário

Depois que o load balancer autentica com êxito um usuário, envia as solicitações do usuário recebidas do IdP para o destino. O load balancer assina a solicitação do usuário para que os aplicativos possam verificar a assinatura e verificar se as solicitações foram enviadas pelo load balancer.

O load balancer adiciona os seguintes cabeçalhos HTTP:

`x-amzn-oidc-accesstoken`

O token de acesso do endpoint de token, em texto simples.

`x-amzn-oidc-identity`

O campo de assunto (`sub`) do endpoint de informações do usuário, em texto simples.

Obs.: a subreivindicação é a melhor maneira de identificar um usuário específico.

`x-amzn-oidc-data`

As solicitações do usuário, no formato de JSON Web Token (JWT).

Os tokens de acesso e as reivindicações de usuário são diferentes dos tokens de ID. Os tokens de acesso e as reivindicações de usuário só permitem o acesso aos recursos do servidor, enquanto os tokens de ID contêm informações adicionais para autenticar um usuário. O Application Load Balancer cria um token de acesso ao autenticar um usuário e apenas transmite os tokens de acesso e as declarações para o backend, porém não transmite as informações do token de ID.

Esses tokens seguem o formato JWT, mas não são tokens de ID. O formato JWT inclui um cabeçalho, carga e assinatura que são codificados em URL base64 e incluem caracteres de preenchimento no final. Um Application Load Balancer usa ES256 (ECDSA usando P-256 e SHA256) para gerar a assinatura JWT.

O cabeçalho JWT contém um objeto JSON com os seguintes campos:

```
{
   "alg": "algorithm",
   "kid": "12345678-1234-1234-1234-123456789012",
   "signer": "arn:aws:elasticloadbalancing:region-code:account-id:loadbalancer/
app/load-balancer-name/load-balancer-id",
```

```
   "iss": "url",
   "client": "client-id",
   "exp": "expiration"
}
```

A carga JWT é um objeto JSON que contém o usuário as solicitações do usuário recebidas do endpoint de informações do usuário do IdP.

```
{
   "sub": "1234567890",
   "name": "name",
   "email": "alias@example.com",
   ...
}
```

Se quiser que o balanceador de carga criptografe suas declarações de usuário, você deve configurar seu grupo de destino para usar HTTPS. Além disso, como prática de segurança, recomendamos que você restrinja seus destinos para receber apenas tráfego do seu Application Load Balancer. Você pode fazer isso configurando o grupo de segurança de seus destinos para fazer referência ao ID do grupo de segurança do balanceador de carga.

Para garantir a segurança, você deve verificar a assinatura antes de fazer qualquer autorização com base nas declarações e validar se o campo `signer` no cabeçalho JWT contém o ARN esperado do Application Load Balancer.

Para obter a chave pública, obtenha o ID de chave de cabeçalho JWT e use-o para procurar a chave pública do endpoint. O endpoint para cada região da AWS é o seguinte:

```
https://public-keys.auth.elb.region.amazonaws.com/key-id
```

Para AWS GovCloud (US), os endpoints são os seguintes:

```
https://s3-us-gov-west-1.amazonaws.com/aws-elb-public-keys-prod-us-gov-west-1/key-id
https://s3-us-gov-east-1.amazonaws.com/aws-elb-public-keys-prod-us-gov-east-1/key-id
```

AWS fornece uma biblioteca que você pode usar para verificar a JWTs assinatura do Amazon Cognito, Application Load Balancers e outros compatíveis com OIDC. IDPs Para obter mais informações, consulte AWS JWT Verify.

Timeout (Tempo limite)

Tempo limite de sessão

O token de atualização e o tempo limite de sessão funcionam juntos da seguinte forma:

- Se o tempo limite da sessão for menor do que a expiração do token de acesso, o load balancer respeitará o tempo limite da sessão. Se o usuário tiver uma sessão ativa com o IdP, talvez o usuário não seja solicitado a fazer login novamente. Caso contrário, o utilizador será redirecionado para fazer login.
  - Se o tempo limite de sessão do IdP for superior ao tempo limite de sessão do Application Load Balancer, o usuário não precisará fornecer credenciais para fazer login novamente. Em vez disso, o IdP o redirecionará para o Application Load Balancer com um novo código de concessão de autorização. Os códigos de autorização são de uso único, mesmo que não haja um novo login.
  - Se o tempo limite de sessão do IdP for igual ou inferior ao tempo limite de sessão do Application Load Balancer, o usuário será solicitado a fornecer credenciais para fazer login novamente. Após o login do usuário, o IdP o redirecionará para o Application Load Balancer com um novo código de concessão de autorização, e o restante do fluxo de autenticação continuará até que a solicitação chegue ao backend.
- Se o tempo limite da sessão for superior ao tempo de expiração do token de acesso e o IdP não for compatível com tokens de atualização, o balanceador de carga manterá a sessão de autenticação até sua expiração. Em seguida, ele forçará o usuário a fazer login novamente.
- Se o tempo limite de sessão for maior do que o tempo de expiração do token de acesso e o IdP comportar tokens de atualização, o load balancer atualizará a sessão de usuário toda vez que o token de acesso expirar. O load balancer fará com que o usuário faça login novamente apenas depois que a sessão de autenticação expirar ou o fluxo de atualização falhar.

Tempo limite de login do cliente

O cliente deve iniciar e concluir o processo de autenticação em até 15 minutos. Se um cliente não conseguir concluir a autenticação durante os 15 minutos, ele receberá um erro HTTP 401 do balanceador de carga. Não é possível alterar nem remover esse tempo limite.

Por exemplo, se um usuário carregar a página de login por meio do Application Load Balancer, ele deverá concluir o processo de login em até 15 minutos. Se o usuário esperar e tentar fazer login

após a expiração do tempo limite de 15 minutos, o balanceador de carga retornará um erro HTTP 401. O usuário precisará atualizar a página e tentar fazer login novamente.

Sair da autenticação

Quando um aplicativo precisa encerrar a sessão de um usuário autenticado, deve definir o tempo de expiração do cookie de sessão de autenticação como –1 e redirecionar o cliente para o endpoint de logout do IdP (se o IdP comportar um). Para evitar que os usuários reutilizem um cookie excluído, é recomendável configurar um tempo de expiração o mais razoável possível para o token de acesso. Se um cliente fornecer ao balanceador de carga um cookie de sessão que tenha um token de acesso expirado com um token de atualização não NULL, o balanceador de carga entrará em contato com o IdP para determinar se o usuário ainda está conectado.

As páginas de destino de logout do cliente não estão autenticadas. Isso significa que eles não podem estar por trás de uma regra do Application Load Balancer que exija autenticação.

- Quando uma solicitação for enviada ao destino, a aplicação deverá definir a expiração como -1 para todos os cookies de autenticação. Os Application Load Balancers são compatíveis com cookies de até 16 K e, portanto, podem criar até 4 fragmentos para enviar ao cliente.
  - Se o IdP tiver um endpoint de logout, ele deverá enviar um redirecionamento para o endpoint de logout do IdP, por exemplo, o endpoint LOGOUT documentado no Guia do desenvolvedor do Amazon Cognito.
  - Se o IdP não tiver um endpoint de logout, a solicitação retornará à página inicial de logout do cliente e o processo de login será reiniciado.
- Supondo que o IdP tenha um endpoint de logout, o IdP deverá expirar os tokens de acesso e os tokens de atualização e redirecionar o usuário de volta para a página inicial de logout do cliente.
- As solicitações subsequentes seguirão o fluxo original de autenticação.

Regras do receptor para seu Application Load Balancer

As regras de ouvinte do seu Application Load Balancer determinam como ele encaminha as solicitações para os destinos. Quando um ouvinte recebe uma solicitação, ele avalia a solicitação em relação a cada regra em ordem de prioridade, começando pela regra de menor numeração. Cada regra inclui condições a serem atendidas e as ações a serem executadas quando as condições da regra forem atendidas. Esse mecanismo de roteamento flexível permite que você implemente padrões sofisticados de distribuição de tráfego, ofereça suporte a vários aplicativos ou

microsserviços em um único balanceador de carga e personalize o tratamento de solicitações com base nos requisitos específicos do seu aplicativo.

Princípios básicos das regras

- Cada regra consiste em uma prioridade, uma ou mais ações e uma ou mais condições.
- Cada ação de regra tem um tipo e as informações necessárias para realizar a ação.
- Cada condição de regra tem um tipo e as informações necessárias para avaliar a condição.
- Ao criar um listener, você define as ações para a regra padrão. A regra padrão não pode ter condições. Se nenhuma das condições das outras regras for atendida, a ação da regra padrão será executada.
- As regras são avaliadas em ordem de prioridade, do valor mais baixo para o valor mais alto. A regra padrão é avaliada por último. Você não pode alterar a prioridade da regra padrão.
- Cada regra deve incluir exatamente uma das seguintes ações: `forward`, `redirect` ou `fixed-response` e deve ser a última ação a ser executada.
- Cada regra diferente da regra padrão pode, opcionalmente, incluir uma das seguintes condições: `host-header`http-request-method,`path-pattern`, e. `source-ip` Opcionalmente, também pode incluir uma ou ambas as seguintes condições: `http-header` e. `query-string`
- Você pode especificar até três strings de comparação por condição e até cinco por regra.

Conteúdo

- Tipos de ação para regras de ouvinte
- Tipos de condição para regras de ouvinte
- Adicione uma regra de ouvinte para seu Application Load Balancer
- Edite uma regra de ouvinte para seu Application Load Balancer
- Exclua uma regra de ouvinte para seu Application Load Balancer

Tipos de ação para regras de ouvinte

As ações determinam como um balanceador de carga trata as solicitações quando as condições de uma regra de ouvinte são satisfeitas. Cada regra deve ter pelo menos uma ação que especifique como lidar com as solicitações correspondentes. Cada ação de regra tem um tipo e informações de configuração. Os Application Load Balancers oferecem suporte aos seguintes tipos de ação para regras de ouvinte.

Tipos de ação

`authenticate-cognito`

[Receptores HTTPS] Use o Amazon Cognito para autenticar usuários. Para obter mais informações, consulte Configurar a autenticação de usuários.

`authenticate-oidc`

[Listeners HTTPS] Usa um provedor de identidade compatível com OpenID Connect (OIDC) para autenticar usuários. Para obter mais informações, consulte Configurar a autenticação de usuários.

`fixed-response`

Retorna uma resposta HTTP personalizada. Para obter mais informações, consulte Ações de resposta fixa.

`forward`

Encaminha as solicitações para os grupos de destino especificados. Para obter mais informações, consulte Ações de encaminhamento.

`redirect`

Redireciona solicitações de um URL para outro. Para obter mais informações, consulte Ações de redirecionamento.

Princípios básicos da ação

- Cada regra deve incluir exatamente uma das seguintes ações de roteamento:`forward`,`redirect`, ou`fixed-response`, e deve ser a última ação a ser executada.
- Um ouvinte HTTPS pode ter uma regra com uma ação de autenticação do usuário e uma ação de roteamento.
- Quando há várias ações, a ação com a menor prioridade é executada primeiro.
- Se a versão do protocolo for gRPC ou HTTP/2, as únicas ações compatíveis serão ações `forward`.

Ações de resposta fixa

Uma `fixed-response` ação descarta as solicitações do cliente e retorna uma resposta HTTP personalizada. Você pode usar essa ação para retornar um código de resposta 2XX, 4XX e 5XX e uma mensagem opcional.

Quando uma ação de `fixed-response` é executada, a ação e o URL do destino do redirecionamento são registrados no logs de acesso. Para obter mais informações, consulte Entradas do log de acesso. A contagem de ações de `fixed-response` com êxito é relatada na métrica `HTTP_Fixed_Response_Count`. Para obter mais informações, consulte Métricas do Application Load Balancer.

Example Exemplo de ação de resposta fixa para o AWS CLI

Você pode especificar uma ação ao criar ou modificar uma regra. Para obter mais informações, consulte os comandos create-rule e modify-rule. A ação a seguir envia uma resposta fixa com o código de status e o corpo da mensagem especificados.

```
[
  {
      "Type": "fixed-response",
      "FixedResponseConfig": {
          "StatusCode": "200",
          "ContentType": "text/plain",
          "MessageBody": "Hello world"
      }
  }
]
```

Ações de encaminhamento

Uma ação de `forward` faz o roteamento das solicitações para seu grupo de destino. Antes de adicionar uma ação `forward`, crie o grupo de destino e adicione destinos a ele. Para obter mais informações, consulte Criar um grupo de destino para o Application Load Balancer.

Se especificar vários grupos de destino para uma ação `forward`, você deverá especificar um peso para cada grupo de destino. Cada peso de grupo de destino é um valor de 0 a 999. As solicitações que correspondem a uma regra de listener com grupos de destino ponderados são distribuídas para esses grupos de destino com base em seus pesos. Por exemplo, se você especificar dois grupos de destino, cada um com um peso de 10, cada grupo de destino receberá metade das solicitações. Se você especificar dois grupos de destino, um com peso de 10 e o outro com peso de 20, o grupo de destino com peso de 20 receberá duas vezes mais solicitações do que o outro grupo de destino.

Se você configurar uma regra para distribuir o tráfego entre grupos de destino ponderados e um dos grupos de destino estiver vazio ou tiver apenas alvos não íntegros, o balanceador de carga não fará o failover automaticamente para um grupo-alvo com alvos íntegros.

Por padrão, configurar uma regra para distribuir o tráfego entre grupos de destino ponderados não garante que as sticky sessions sejam honradas. Para garantir que as sticky sessions sejam honradas, habilite a perdurabilidade do grupo de destino para a regra. Quando o balanceador de carga encaminha pela primeira vez uma solicitação para um grupo-alvo ponderado, ele gera um cookie chamado AWSALBTG que codifica informações sobre o grupo-alvo selecionado, criptografa o cookie e inclui o cookie na resposta ao cliente. O cliente deve incluir o cookie recebido nas solicitações subsequentes ao load balancer. Quando o load balancer recebe uma solicitação que corresponde a uma regra com a perdurabilidade do grupo de destino habilitada e contém o cookie, a solicitação é roteada para o grupo de destino especificado no cookie.

Os Application Load Balancers não são compatíveis com valores de cookie codificados por URL.

Com solicitações de CORS (cross-origin resource sharing, compartilhamento de recursos de origem cruzada), alguns navegadores exigem `SameSite=None; Secure` para habilitar a perdurabilidade. Nesse caso, o Elastic Load Balancing gera um segundo cookie AWSALBTGCORS, que inclui as mesmas informações do cookie de aderência original mais esse atributo. `SameSite` Os clientes recebem ambos os cookies.

Example Exemplo de ação de encaminhamento com um grupo de destino

Você pode especificar uma ação ao criar ou modificar uma regra. Para obter mais informações, consulte os comandos create-rule e modify-rule. A ação a seguir encaminha solicitações para o grupo de destino especificado.

```
[
  {
      "Type": "forward",
      "ForwardConfig": {
          "TargetGroups": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067"
              }
          ]
      }
  }
]
```

Example Exemplo de ação de encaminhamento com dois grupos ponderados de destino

A ação a seguir encaminha solicitações para os dois grupos de destino especificados, com base no peso de cada grupo de destino.

```
[
  {
      "Type": "forward",
      "ForwardConfig": {
          "TargetGroups": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/blue-targets/73e2d6bc24d8a067",
                  "Weight": 10
              },
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/green-targets/09966783158cda59",
                  "Weight": 20
              }
          ]
      }
  }
]
```

Example Exemplo de ação de encaminhamento com perdurabilidade habilitada

Se você tiver uma regra de encaminhamento com vários grupos de destino e um ou mais grupos de destino tiver sessões persistentes habilitadas, você deverá habilitar a perdurabilidade do grupo de destino.

A ação a seguir encaminha solicitações para os dois grupos de destino especificados, com a perdurabilidade do grupo de destino habilitada. As solicitações que não contêm os cookies de perdurabilidade são roteadas com base no peso de cada grupo de destino.

```
[
  {
      "Type": "forward",
      "ForwardConfig": {
          "TargetGroups": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/blue-targets/73e2d6bc24d8a067",
```

```
                "Weight": 10
            },
            {
                "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/green-targets/09966783158cda59",
                "Weight": 20
            }
        ],
        "TargetGroupStickinessConfig": {
            "Enabled": true,
            "DurationSeconds": 1000
        }
    }
  }
]
```

Ações de redirecionamento

Uma `redirect` ação redireciona as solicitações do cliente de uma URL para outra. Você pode configurar redirecionamentos como temporários (HTTP 302) ou permanentes (HTTP 301) com base em suas necessidades.

Um URI consiste nos seguintes componentes:

```
protocol://hostname:port/path?query
```

Você deve modificar pelo menos um dos seguintes componentes para evitar um loop de redirecionamento: protocolo, nome do host, porta ou caminho. Todos os componentes que você não modificar manterão seus valores originais.

protocolo

O protocolo (HTTP or HTTPS). Você pode redirecionar HTTP para HTTP, HTTP para HTTPS e HTTPS para HTTPS. Você não pode redirecionar HTTPS para HTTP.

hostname

O nome do host. Um nome de host não diferencia maiúsculas de minúsculas, pode ter até 128 caracteres e consiste em caracteres alfanuméricos, curingas (* e ?) e hifens (-).

porta

A porta (1 a 65535).

caminho

O caminho absoluto, começando com a "/" inicial. Um caminho diferencia maiúsculas de minúsculas, pode ter até 128 caracteres e consiste em caracteres alfanuméricos, curingas (* e ?), & (usando &) e nos seguintes caracteres especiais: _-.$/~'"@:+.

consulta

Os parâmetros da consulta. O tamanho máximo é 128 caracteres.

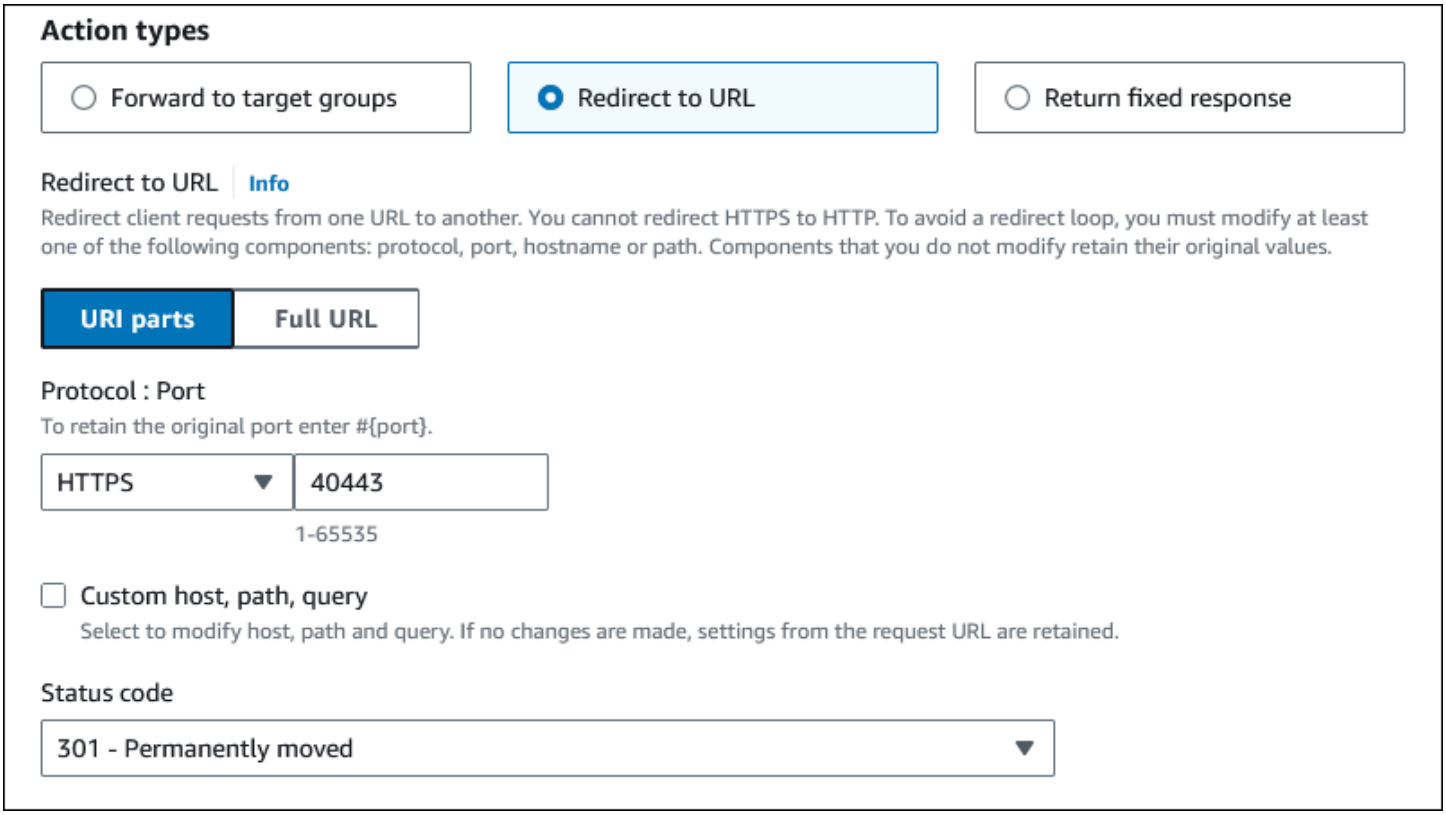
Você pode reutilizar os componentes do URI do URL original no URL de destino usando as seguintes palavras-chave reservadas:

- `#{protocol}` – mantém o protocolo. Use no protocolo e nos componentes de consulta.
- `#{host}` – mantém o domínio. Use no nome de host, no caminho e nos componentes de consulta.
- `#{port}` – mantém a porta. Use na porta, no caminho e nos componentes de consulta.
- `#{path}` – mantém o caminho. Use no caminho e nos componentes de consulta.
- `#{query}` – mantém os parâmetros da consulta. Use no componente de consulta.

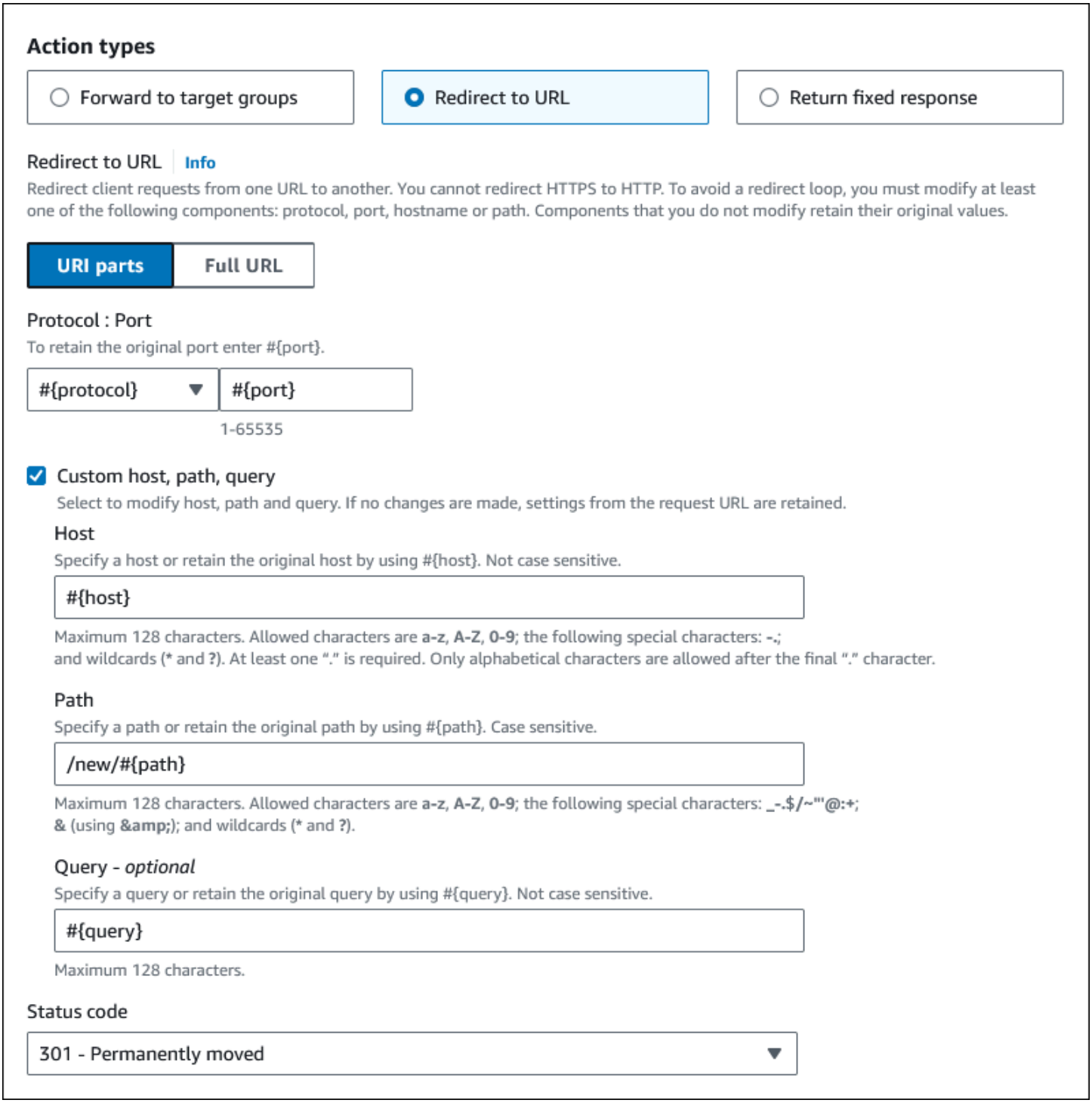
Quando uma ação de `redirect` é executada, a ação é registrada nos logs de acesso. Para obter mais informações, consulte Entradas do log de acesso. A contagem de ações de `redirect` com êxito é relatada na métrica `HTTP_Redirect_Count`. Para obter mais informações, consulte Métricas do Application Load Balancer.

Example Exemplo de ações de redirecionamento usando o console

A regra a seguir define um redirecionamento permanente para um URL que usa o protocolo HTTPS e a porta especificada (40443), mas mantém o nome do host, o caminho e os parâmetros de consulta originais. Esta tela é equivalente a "https://#{host}:40443/#{path}?#{query}".



A seguinte regra define um redirecionamento permanente para um URL que usa o protocolo, a porta, nome de host e os parâmetros de consulta originais, e usa a palavra-chave `#{path}` para criar um caminho modificado. Esta tela é equivalente a "#{protocol}://#{host}:#{port}/new/#{path}?#{query}".



Example Exemplo de ação de redirecionamento para o AWS CLI

Você pode especificar uma ação ao criar ou modificar uma regra. Para obter mais informações, consulte os comandos create-rule e modify-rule. A ação a seguir redireciona uma solicitação HTTP para uma solicitação HTTPS na porta 443, com o mesmo nome de host, caminho e string de consulta que a solicitação HTTP.

```
[
  {
      "Type": "redirect",
      "RedirectConfig": {
          "Protocol": "HTTPS",
          "Port": "443",
          "Host": "#{host}",
          "Path": "/#{path}",
          "Query": "#{query}",
          "StatusCode": "HTTP_301"
      }
  }
]
```

Tipos de condição para regras de ouvinte

As condições definem os critérios que as solicitações recebidas devem atender para que uma regra de ouvinte entre em vigor. Se uma solicitação corresponder às condições de uma regra, ela será tratada conforme especificado pelas ações da regra. Cada condição de regra possui um tipo e informações de configuração. Os Application Load Balancers oferecem suporte aos seguintes tipos de condições para regras de ouvinte.

Tipos de condição

`host-header`

Rota com base no nome do host de cada solicitação. Para obter mais informações, consulte Condições do host.

`http-header`

Rota com base nos cabeçalhos HTTP de cada solicitação. Para obter mais informações, consulte Condições de cabeçalho HTTP.

`http-request-method`

Rota com base no método de solicitação HTTP de cada solicitação. Para obter mais informações, consulte Condições do método de solicitação HTTP.

`path-pattern`

Rota com base nos padrões de caminho na solicitação URLs. Para obter mais informações, consulte Condições do caminho.

`query-string`

Rota com base em key/value pares ou valores nas cadeias de caracteres de consulta. Para obter mais informações, consulte Condições de string de consulta.

`source-ip`

Rota com base no endereço IP de origem de cada solicitação. Para obter mais informações, consulte Condições de endereço IP de origem.

Princípios básicos da condição

- Cada regra pode, opcionalmente, incluir até uma de cada uma das seguintes condições: `host-header`, `http-request-method`, `path-pattern` e `source-ip`. Cada regra também pode, opcionalmente, incluir uma ou mais de cada uma das seguintes condições: `http-header` e `query-string`.
- Você pode especificar até três avaliações de correspondência por condição. Por exemplo, para cada condição `http-header`, você pode especificar até três strings para serem comparadas ao valor do cabeçalho HTTP na solicitação. A condição é atendida se uma das strings corresponder ao valor do cabeçalho HTTP. Para exigir que todas as strings sejam uma correspondência, crie uma condição por avaliação de correspondência.
- Você pode especificar até cinco avaliações de correspondência por regra. Por exemplo, você pode criar uma regra com cinco condições em que cada condição tenha uma avaliação de correspondência.
- Você pode incluir caracteres curinga nas avaliações de correspondência para as condições `http-header`, `host-header`, `path-pattern` e `query-string`. Existe um limite de cinco caracteres curinga por regra.
- As regras são aplicadas apenas a caracteres ASCII visíveis; caracteres de controle (0x00 a 0x1f e 0x7f) são excluídos.

Demonstrações

Para demonstrações, consulte Roteamento avançado de solicitação.

Condições de cabeçalho HTTP

Você pode usar condições de cabeçalho HTTP para configurar regras que roteiam solicitações com base nos cabeçalhos HTTP da solicitação. Você pode especificar os nomes dos campos de

cabeçalho HTTP padrão ou personalizados. O nome do cabeçalho e a avaliação de correspondência não diferenciam maiúsculas de minúsculas. Os caracteres curinga a seguir são compatíveis com as strings de comparação: * (corresponde a 0 ou mais caracteres) e ? (corresponde exatamente a 1 caractere). Caracteres curinga não são compatíveis com o nome do cabeçalho.

Quando o atributo Application Load Balancer `routing.http.drop_invalid_header_fields` estiver ativado, ele eliminará os nomes dos cabeçalhos que não estão em conformidade com as expressões regulares (). `A-Z,a-z,0-9` Nomes de cabeçalho que não estejam em conformidade com as expressões regulares também podem ser adicionados.

Example Exemplo de condição de cabeçalho HTTP para o AWS CLI

Você pode especificar condições ao criar ou modificar uma regra. Para obter mais informações, consulte os comandos create-rule e modify-rule. A condição a seguir é atendida por solicitações com um cabeçalho User-Agent que corresponda a uma das strings especificadas.

```
[
  {
      "Field": "http-header",
      "HttpHeaderConfig": {
          "HttpHeaderName": "User-Agent",
          "Values": ["*Chrome*", "*Safari*"]
      }
  }
]
```

Condições do método de solicitação HTTP

Você pode usar condições do método de solicitação HTTP para configurar regras que roteiam solicitações com base no método de solicitação HTTP da solicitação. Você pode especificar métodos HTTP padrão ou personalizados. A avaliação de correspondência faz distinção entre maiúsculas e minúsculas. Caracteres curinga não são compatíveis; portanto, o nome do método deve ser uma correspondência exata.

Recomendamos que você roteie as solicitações GET e HEAD da mesma maneira, porque a resposta a uma solicitação HEAD pode ser armazenada em cache.

Example Exemplo de condição do método HTTP para o AWS CLI

Você pode especificar condições ao criar ou modificar uma regra. Para obter mais informações, consulte os comandos create-rule e modify-rule. A condição a seguir é atendida por solicitações que usam o método especificado.

```
[
  {
      "Field": "http-request-method",
      "HttpRequestMethodConfig": {
          "Values": ["CUSTOM-METHOD"]
      }
  }
]
```

Condições do host

Você pode usar as condições do host para definir regras que roteiam solicitações com base no nome do host no cabeçalho de host (também conhecido como roteamento baseado em host). Isso permite que você ofereça suporte a vários subdomínios e a diferentes domínios de nível superior usando um só balanceador de carga.

Um nome de host não diferencia maiúsculas de minúsculas, pode ter até 128 caracteres e conter qualquer um dos caracteres a seguir:

- A-Z, a-z, 0-9
- - .
- * (corresponde a 0 ou mais caracteres)
- ? (corresponde a exatamente 1 caractere)

É necessário incluir pelo menos um caractere ".". Você pode incluir somente caracteres alfabéticos após o "." final.

Por exemplo, os hostnames

- example.com
- test.example.com
- *.example.com

A regra *.example.com corresponde a test.example.com, mas não corresponde a example.com.

Example Exemplo de condição de cabeçalho de host para o AWS CLI

Você pode especificar condições ao criar ou modificar uma regra. Para obter mais informações, consulte os comandos create-rule e modify-rule. A condição a seguir é atendida por solicitações com um cabeçalho de host que corresponda à string especificada.

```
[
  {
      "Field": "host-header",
      "HostHeaderConfig": {
          "Values": ["*.example.com"]
      }
  }
]
```

Condições do caminho

Você pode usar as condições de caminho para definir regras que roteiam solicitações com base no URL da solicitação (também conhecido como roteamento baseado em caminho).

O padrão de caminho é aplicado apenas ao caminho do URL, não aos seus parâmetros de consulta. Ele é aplicado somente a caracteres ASCII visíveis; caracteres de controle (0x00 a 0x1f e 0x7f) são excluídos.

A avaliação da regra é realizada somente após a normalização de URI.

O padrão do caminho diferencia maiúsculas de minúsculas, pode ter até 128 caracteres e conter qualquer um dos caracteres a seguir.

- A-Z, a-z, 0-9
- _ - . $ / ~ " ' @ : +
- & (usando &)
- * (corresponde a 0 ou mais caracteres)
- ? (corresponde a exatamente 1 caractere)

Se a versão do protocolo for gRPC, as condições podem ser específicas de um pacote, serviço ou método.

Exemplos de padrões de caminho HTTP

- `/img/*`
- `/img/*/pics`

Exemplos de padrões de caminho gRPC

- /package
- /package.service
- /package.service/method

O caminho padrão é usado para rotear as solicitações, mas não as altera. Por exemplo, se uma regra tiver um padrão de caminho `/img/*`, a regra encaminhará uma solicitação de `/img/picture.jpg` ao grupo de destino especificado como uma solicitação para `/img/picture.jpg`.

Example Exemplo de condição de padrão de caminho para o AWS CLI

Você pode especificar condições ao criar ou modificar uma regra. Para obter mais informações, consulte os comandos create-rule e modify-rule. A condição a seguir é atendida por solicitações com um URL que contém a string especificada.

```
[
  {
      "Field": "path-pattern",
      "PathPatternConfig": {
          "Values": ["/img/*"]
      }
  }
]
```

Condições de string de consulta

Você pode usar condições da sequência de caracteres de consulta para configurar regras que roteiam solicitações com base em key/value pares ou valores na sequência de caracteres de consulta. A avaliação de correspondência não diferencia maiúsculas de minúsculas. Os caracteres curinga a seguir são compatíveis: * (corresponde a 0 ou mais caracteres) e ? (corresponde exatamente a 1 caractere).

Example Exemplo de condição de sequência de caracteres de consulta para o AWS CLI

Você pode especificar condições ao criar ou modificar uma regra. Para obter mais informações, consulte os comandos create-rule e modify-rule. A condição a seguir é satisfeita por solicitações com uma string de consulta que inclui um key/value par de “version=v1" ou qualquer chave definida como “example”.

```
[
  {
      "Field": "query-string",
      "QueryStringConfig": {
          "Values": [
            {
                "Key": "version",
                "Value": "v1"
            },
            {
                "Value": "*example*"
            }
          ]
      }
  }
]
```

Condições de endereço IP de origem

Você pode usar condições de endereço IP de origem para configurar regras que roteiam solicitações com base no endereço IP de origem da solicitação. O endereço IP deve ser especificado no formato CIDR. Você pode usar ambos IPv4 e IPv6 endereços. Caracteres curinga não são compatíveis. Você não pode especificar o CIDR `255.255.255.255/32` para a condição da regra de IP de origem.

Se um cliente estiver por trás de um proxy, este é o endereço IP do proxy e não o endereço IP do cliente.

Essa condição não é satisfeita pelos endereços no X-Forwarded-For cabeçalho. Para pesquisar endereços no X-Forwarded-For cabeçalho, use uma `http-header` condição.

Example Exemplo de condição de IP de origem para o AWS CLI

Você pode especificar condições ao criar ou modificar uma regra. Para obter mais informações, consulte os comandos create-rule e modify-rule. A condição a seguir é atendida por solicitações com um endereço IP de origem em um dos blocos CIDR especificados.

```
[
  {
      "Field": "source-ip",
      "SourceIpConfig": {
          "Values": ["192.0.2.0/24", "198.51.100.10/32"]
      }
  }
]
```

Adicione uma regra de ouvinte para seu Application Load Balancer

Você define uma regra padrão ao criar um ouvinte. Você pode definir regras adicionais a qualquer momento.

Para adicionar uma regra usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Nas guias Ouvintes e regras, selecione o texto na coluna Protocol:Porta para abrir as páginas de detalhes do ouvinte.
5. Na guia Regras, selecione Adicionar regra.
6. (Opcional) Para especificar um nome para sua regra, expanda Nome e tags e insira o nome. Para adicionar mais tags, selecione Add additional tag (Adicionar outra tag).
7. Para cada condição, escolha Adicionar condição, escolha o tipo de condição e forneça os valores de condição necessários:

   - Cabeçalho do host — Insira o cabeçalho do host. Por exemplo: `*.example.com`.

     Máximo de 128 caracteres. Não diferencia maiúsculas de minúsculas. Os caracteres permitidos são a-z, A-Z, 0-9 e os seguintes caracteres especiais: -_.; e curingas (* e ?). É necessário incluir pelo menos um caractere ".". Você pode incluir somente caracteres alfabéticos após o "." final.

   - Caminho — Insira o caminho. Por exemplo: `/item/*`.

     Máximo de 128 caracteres. Diferencia maiúsculas e minúsculas. Os caracteres permitidos são a-z, A-Z, 0-9 e os seguintes caracteres especiais: _-.$/~"'@:+; &; e curingas (* e ?).

- Cadeia de caracteres de consulta — Insira pares chave-valor ou valores sem chaves.

  Máximo de 128 caracteres. Não diferencia maiúsculas de minúsculas. Os caracteres permitidos são a-z, A-Z, 0-9 e os seguintes caracteres especiais: _-.$/~"'@:+&()!,;=; e curingas (* e ?).

- Método de solicitação HTTP — Insira o método de solicitação HTTP.

  Máximo de 40 caracteres. Diferencia maiúsculas e minúsculas. Os caracteres permitidos são A-Z e os seguintes caracteres especiais: -_. Curingas não são compatíveis.

- Cabeçalho HTTP: digite o nome do cabeçalho e adicione uma ou mais strings de comparação.
  - Nome do cabeçalho HTTP: a regra avaliará as solicitações que contêm esse cabeçalho para confirmar os valores correspondentes.

    Máximo de 40 caracteres. Não diferencia maiúsculas de minúsculas. Os caracteres permitidos são a-z, A-Z, 0-9 e os seguintes caracteres especiais: *?-!#$%&'+.^_`|~. Curingas não são compatíveis.
  - Valor do cabeçalho HTTP: insira cadeias de caracteres para comparação com o valor do cabeçalho HTTP.

    Máximo de 128 caracteres. Não diferencia maiúsculas de minúsculas. Os caracteres permitidos são a-z, A-Z, 0-9; espaços; os seguintes caracteres especiais:!” #$%&' () +,. /:; <=>@ [] ^_` {|} ~-; e curingas (* e?).
- IP de origem: defina o endereço IP de origem no formato CIDR.

  Ambos IPv4 e IPv6 CIDRs são permitidos. Curingas não são compatíveis.

8. (Opcional) Para adicionar uma regra de autenticação, selecione Autenticar usuários, escolha um provedor de identidade e forneça as informações necessárias. Para obter mais informações, consulte Autenticar usuários usando um Application Load Balancer.
9. Em Ação de roteamento, selecione uma das seguintes ações de roteamento e forneça as informações necessárias:

   - Encaminhar para grupos-alvo — Escolha um grupo-alvo. Para adicionar outro grupo-alvo, escolha Adicionar grupo-alvo, escolha um grupo-alvo, revise as porcentagens relativas e atualize os pesos conforme necessário. Você deve ativar a aderência em nível de grupo se tiver ativado a aderência em qualquer um dos grupos-alvo.
   - Redirecionar para URL — Insira o URL inserindo cada parte separadamente na guia Partes do URI ou inserindo o endereço completo na guia URL completo. Para Código de

status, selecione temporário (HTTP 302) ou permanente (HTTP 301) com base em suas necessidades.

- Retornar resposta fixa — insira o código de resposta para retornar as solicitações canceladas do cliente. Opcionalmente, você pode especificar o tipo de conteúdo e o corpo da resposta.

10. Escolha Próximo.
11. Em Prioridade, insira um valor de 1 a 50.000. As regras são avaliadas em ordem de prioridade, do valor mais baixo para o valor mais alto.
12. Escolha Próximo.
13. Na página Review and create (Revisar e criar), escolha Create (Criar).

Para adicionar uma regra usando o AWS CLI

Use o comando create-rule para criar a regra. Use o comando describe-rules para visualizar informações sobre a regra.

Edite uma regra de ouvinte para seu Application Load Balancer

Você pode editar a ação e as condições de uma regra de ouvinte a qualquer momento. As atualizações de regras não entram em vigor imediatamente, portanto, as solicitações podem ser roteadas usando a configuração de regra anterior por um curto período após a atualização de uma regra. Todas as solicitações em trânsito são concluídas.

Tarefas

- Modificar a ação padrão
- Atualizar as prioridades das regras
- Gerenciar as tags de regras

Modificar a ação padrão

A ação padrão é atribuída a uma regra chamada Padrão. Você pode manter o tipo de regra atual e alterar as informações necessárias ou pode alterar o tipo de regra e fornecer as novas informações obrigatórias.

Para modificar a ação padrão usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.

2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Receptores e regras, selecione o texto na coluna Protocolo:Porta para abrir a página de detalhes do receptor.
5. Na guia Regras, na seção Regras do ouvinte, marque a caixa de seleção da regra padrão. Selecione Ações e Editar regra.
6. Na seção Ação padrão, atualize a regra conforme necessário.

Para modificar a ação padrão usando o AWS CLI

Use o comando modify-listener.

Atualizar as prioridades das regras

As regras são avaliadas em ordem de prioridade, do valor mais baixo para o valor mais alto. A regra padrão é avaliada por último. Você pode alterar a prioridade de uma regra não padrão a qualquer momento. Você não pode alterar a prioridade da regra padrão.

Para atualizar as prioridades das regras usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Load Balancers.
3. Selecione o load balancer.
4. Na guia Receptores e regras, selecione o texto na coluna Protocolo:Porta para abrir a página de detalhes do receptor.
5. Na guia Regras, selecione a regra do ouvinte e escolha Ações, Repriorizar regras.
6. Na seção Regras do ouvinte, a coluna Prioridade exibe as prioridades atuais da regra. Para atualizar a prioridade de uma regra, insira um valor de 1 a 50.000.
7. Escolha Salvar alterações.

Para atualizar as prioridades das regras usando o AWS CLI

Use o comando set-rule-priorities.

Gerenciar as tags de regras

As tags ajudam você a categorizar seus receptores e regras de maneiras diferentes. Por exemplo, você pode marcar um recurso por finalidade, proprietário ou ambiente. As chaves de tag devem ser exclusivas para cada regra. Se você adicionar uma tag a uma chave que já está associada a uma regra, o valor da tag será atualizado.

Quando não precisar mais de uma tag, você poderá removê-la.

Para gerenciar as tags de uma regra usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Load Balancers.
3. Escolha o nome do balanceador de carga para abrir sua página de detalhes.
4. Na guia Receptores e regras, selecione o texto na coluna Protocolo:Porta para abrir a página de detalhes do receptor.
5. Na guia Regras, selecione o texto na coluna Etiqueta de nome para abrir a página de detalhes da regra.
6. Na página de detalhes, escolha Gerenciar tags.
7. Na página Gerenciar tags, é possível realizar uma ou mais das seguintes ações:
   a. Para adicionar uma nova tag, escolha Adicionar nova tag e insira valores para Chave e Valor.
   b. Para excluir uma tag, escolha Remover ao lado da tag.
   c. Para atualizar uma tag, insira novos valores para Chave ou Valor.
8. Escolha Salvar alterações.

Para gerenciar as tags de uma regra usando o AWS CLI

Use os comandos add-tags e remove-tags.

Exclua uma regra de ouvinte para seu Application Load Balancer

Você pode excluir as regras não padrão para um listener a qualquer momento. Você não pode excluir a regra padrão para um ouvinte. Quando você exclui um listener, todas as regras são excluídas.

Para excluir uma regra usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Receptores e regras, selecione o texto na coluna Protocolo:Porta para abrir a página de detalhes do receptor.
5. Selecione a regra.
6. Escolha Actions (Ações), Delete rule (Excluir regra).
7. Quando a confirmação for solicitada, insira **`confirm`** e selecione Excluir.

Para excluir uma regra usando o AWS CLI

Use o comando delete-rule.

Cabeçalhos HTTP e Application Load Balancers

As solicitações HTTP e as respostas HTTP usam campos de cabeçalho para enviar informações sobre as mensagens HTTP. Os cabeçalhos HTTP são adicionados automaticamente. Os campos de cabeçalho são pares de nome-valor separados por dois pontos e separados por um retorno de carro (CR) e um avanço de linha (LF). Um conjunto padrão de campos de cabeçalho HTTP está definido na RFC 2616, Cabeçalhos de mensagem. Também há a disponibilidade de cabeçalhos HTTP não padrão que são adicionados automaticamente e amplamente usados pelas aplicações. Alguns dos cabeçalhos HTTP não padrão possuem um prefixo `X-Forwarded`. Os Application Load Balancers são compatíveis com os seguintes cabeçalhos `X-Forwarded`.

Para obter mais informações sobre conexões HTTP, consulte Roteamento de solicitação no Manual do usuário do Elastic Load Balancing.

Cabeçalhos X-Forwarded

- X-Forwarded-For
- X-Forwarded-Proto
- X-Forwarded-Port

X-Forwarded-For

O cabeçalho da solicitação `X-Forwarded-For` ajuda você a identificar o endereço IP de um cliente quando usar um load balancer HTTP ou HTTPS. Como os balanceadores de carga interceptam o tráfego entre clientes e servidores, os logs de acesso do seu servidor vão conter apenas o endereço IP do balanceador de carga. Para ver o endereço IP do cliente, use o atributo `routing.http.xff_header_processing.mode`. Esse atributo permite que você modifique, preserve ou remova o cabeçalho `X-Forwarded-For` na solicitação HTTP antes que o Application Load Balancer envie a solicitação ao destino. Os valores possíveis para esse atributo são `append`, `preserve` e `remove`. O valor padrão desse atributo é `append`.

> **Important**
>
> O cabeçalho `X-Forwarded-For` deve ser usado com cuidado devido ao potencial de riscos à segurança. As entradas só podem ser consideradas confiáveis se adicionadas por sistemas devidamente protegidos na rede.

Anexar

Por padrão, o Application Load Balancer armazena o endereço IP do cliente no cabeçalho de solicitação `X-Forwarded-For` e encaminha o cabeçalho para o seu servidor. Se o cabeçalho de solicitação `X-Forwarded-For` não estiver incluído na solicitação original, o balanceador de carga criará um com o endereço IP do cliente como o valor da solicitação. Caso contrário, o balanceador de carga anexará o endereço IP do cliente ao cabeçalho existente e encaminhará o cabeçalho para o seu servidor. O cabeçalho de solicitação `X-Forwarded-For` pode conter vários endereços IP separados por vírgula.

O cabeçalho de solicitação `X-Forwarded-For` leva a seguinte forma:

```
X-Forwarded-For: client-ip-address
```

Veja a seguir um exemplo de cabeçalho de solicitação `X-Forwarded-For` para um cliente com o endereço IP `203.0.113.7`.

```
X-Forwarded-For: 203.0.113.7
```

Veja a seguir um exemplo de cabeçalho de `X-Forwarded-For` solicitação para um cliente com um IPv6 endereço de`2001:DB8::21f:5bff:febf:ce22:8a2e`.

```
X-Forwarded-For: 2001:DB8::21f:5bff:febf:ce22:8a2e
```

Quando o atributo de preservação da porta do cliente (`routing.http.xff_client_port.enabled`) estiver habilitado no balanceador de carga, o cabeçalho `X-Forwarded-For` da solicitação incluirá o `client-port-number` anexado ao `client-ip-address`, separado por dois pontos. Em seguida, o cabeçalho adotará a seguinte forma:

```
IPv4 -- X-Forwarded-For: client-ip-address:client-port-number
```

```
IPv6 -- X-Forwarded-For: [client-ip-address]:client-port-number
```

Pois IPv6, observe que quando o balanceador de carga anexa o `client-ip-address` ao cabeçalho existente, ele coloca o endereço entre colchetes.

Veja a seguir um exemplo de cabeçalho de `X-Forwarded-For` solicitação para um cliente com IPv4 endereço `12.34.56.78` e número de porta de`8080`.

```
X-Forwarded-For: 12.34.56.78:8080
```

Veja a seguir um exemplo de cabeçalho de `X-Forwarded-For` solicitação para um cliente com IPv6 endereço `2001:db8:85a3:8d3:1319:8a2e:370:7348` e número de porta de`8080`.

```
X-Forwarded-For: [2001:db8:85a3:8d3:1319:8a2e:370:7348]:8080
```

Preservar

O modo `preserve` no atributo garante que o cabeçalho `X-Forwarded-For` na solicitação HTTP não seja modificado de nenhuma forma antes do envio para os destinos.

Remover

O modo `remove` no atributo remove o cabeçalho `X-Forwarded-For` na solicitação HTTP antes do envio para os destinos.

> **Note**
>
> Se você habilitar o atributo de preservação da porta do cliente (`routing.http.xff_client_port.enabled`) e também selecionar `preserve` ou `remove` para o atributo `routing.http.xff_header_processing.mode`, o Application Load Balancer substituirá o atributo de preservação da porta do cliente. Dependendo do modo selecionado, ele mantém o cabeçalho `X-Forwarded-For` inalterado ou o remove antes de enviá-lo para os destinos.

A tabela a seguir mostra exemplos do cabeçalho `X-Forwarded-For` que o destino recebe quando você seleciona o modo `append`, `preserve` ou `remove`. Neste exemplo, o endereço IP do último salto é `127.0.0.1`.

Descrição da solicitação	Exemplo de solicitação	XFF no modo **`append`**	XFF no modo **`preserve`**	XFF no modo **`remove`**
A solicitação é enviada sem cabeçalho XFF.	`GET / index.html HTTP/1.1 Host: example.com`	`X-Forwarded-For: 127.0.0.1`	Não está presente	Não está presente
A solicitação é enviada com um cabeçalho XFF e um endereço IP do cliente.	`GET / index.html HTTP/1.1 Host: example.com X-Forwarded-For: 127.0.0.4`	`X-Forwarded-For: 127.0.0.4, 127.0.0.1`	`X-Forwarded-For: 127.0.0.4`	Não está presente
A solicitação é enviada com um cabeçalho XFF e vários endereços IP do cliente.	`GET / index.html HTTP/1.1 Host: example.com`	`X-Forwarded-For: 127.0.0.4, 127.0.0.8, 127.0.0.1`	`X-Forwarded-For: 127.0.0.4, 127.0.0.8`	Não está presente

Descrição da solicitação	Exemplo de solicitação	XFF no modo **append**	XFF no modo **preserve**	XFF no modo **remove**
	`X-Forwarded-For: 127.0.0.4, 127.0.0.8`			

Para modificar, preservar ou remover o cabeçalho X-Forwarded-For usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Atributos, escolha Editar.
5. Na seção Configuração de tráfego, em Tratamento de pacotes, para X-Forwarded-For cabeçalho, escolha Anexar (padrão), Preservar ou Remover.
6. Escolha Salvar alterações.

Para modificar, preservar ou remover o X-Forwarded-For cabeçalho usando o AWS CLI

Use o comando modify-load-balancer-attributes com o atributo `routing.http.xff_header_processing.mode`.

X-Forwarded-Proto

O cabeçalho da solicitação `X-Forwarded-Proto` ajuda você a identificar o protocolo (HTTP ou HTTPS) que um cliente usou para se conectar ao seu load balancer. Os logs de acesso do servidor contêm apenas o protocolo usado entre o servidor e o load balancer; eles não contêm informações sobre o protocolo usado entre o cliente e o load balancer. Para determinar o protocolo usado entre o cliente e o balanceador de carga, use o cabeçalho de solicitação `X-Forwarded-Proto`. O Elastic Load Balancing armazena o protocolo usado entre o cliente e o balanceador de carga no cabeçalho da solicitação `X-Forwarded-Proto` e encaminha o cabeçalho para seu servidor.

O aplicativo ou o site podem usar o protocolo armazenado no cabeçalho da solicitação `X-Forwarded-Proto` para renderizar uma resposta que redireciona para o URL apropriado.

O cabeçalho de solicitação `X-Forwarded-Proto` leva a seguinte forma:

```
X-Forwarded-Proto: originatingProtocol
```

O exemplo a seguir contém um cabeçalho de solicitação `X-Forwarded-Proto` para uma solicitação originada do cliente como solicitação de HTTPS:

```
X-Forwarded-Proto: https
```

X-Forwarded-Port

O cabeçalho de solicitação `X-Forwarded-Port` ajuda a identificar a porta de destino que o cliente usou para se conectar ao load balancer.

Modificação do cabeçalho HTTP para seu Application Load Balancer

A modificação do cabeçalho HTTP é suportada pelos Application Load Balancers, tanto para cabeçalhos de solicitação quanto para cabeçalhos de resposta. Sem precisar atualizar o código do seu aplicativo, a modificação do cabeçalho permite que você tenha mais controle sobre o tráfego e a segurança do seu aplicativo.

Para ativar a modificação do cabeçalho, consulteAtivar modificação do cabeç.

Renomear mTLS/TLS cabeçalhos

O recurso de renomeação do cabeçalho permite que você configure os nomes dos cabeçalhos mTLS e TLS que o Application Load Balancer gera e adiciona às solicitações.

Essa capacidade de modificar cabeçalhos HTTP permite que seu Application Load Balancer ofereça suporte facilmente a aplicativos que usam cabeçalhos de solicitação e resposta formatados especificamente.

Cabeçalho	Descrição
X-Amzn-Mtls-Clientcert-Serial-Number	Garante que o alvo possa identificar e verificar o certificado específico apresentado pelo cliente durante o handshake TLS.

Cabeçalho	Descrição
X-Amzn-Mtls-Clientcert-Issuer	Ajuda o alvo a validar e autenticar o certificado do cliente identificando a autoridade de certifica ção que emitiu o certificado.
X-Amzn-Mtls-Clientcert-Subject	Fornece ao alvo informações detalhadas sobre a entidade para a qual o certificado do cliente foi emitido, o que ajuda na identificação, autenticação, autorização e registro durante a autenticação do mTLS.
X-Amzn-Mtls-Clientcert-Validity	Permite que o alvo verifique se o certificado do cliente que está sendo usado está dentro do período de validade definido, garantindo que o certificado não tenha expirado ou seja usado prematuramente.
X-Amzn-Mtls-Clientcert-Leaf	Fornece o certificado do cliente usado no handshake do mTLS, permitindo que o servidor autentique o cliente e valide a cadeia de certificados. Isso garante que a conexão seja segura e autorizada.
X-Amzn-Mtls-Clientcert	Carrega o certificado completo do cliente. Permitir que o alvo verifique a autenticidade do certificado, valide a cadeia de certificados e autentique o cliente durante o processo de handshake do mTLS.
X-Amzn-TLS-Version	Indica a versão do protocolo TLS usada para uma conexão. Isso facilita a determinação do nível de segurança da comunicação, a solução de problemas de conexão e a garantia da conformidade.

Cabeçalho	Descrição
X-Amzn-TLS-Cipher-Suite	Indica a combinação de algoritmos criptográficos usados para proteger uma conexão no TLS. Isso permite que o servidor avalie a segurança da conexão, ajudando na solução de problemas de compatibilidade e garantindo a conformidade com as políticas de segurança.

Adicionar cabeçalhos de resposta

Usando cabeçalhos de inserção, você pode configurar seu Application Load Balancer para adicionar cabeçalhos relacionados à segurança às respostas. Com esses atributos, você pode inserir cabeçalhos, incluindo HSTS, CORS e CSP.

Por padrão, esses cabeçalhos estão vazios. Quando isso acontece, o Application Load Balancer não modifica esse cabeçalho de resposta.

Quando você ativa um cabeçalho de resposta, o Application Load Balancer adiciona o cabeçalho com o valor configurado a todas as respostas. Se a resposta do destino incluir o cabeçalho de resposta HTTP, o balanceador de carga atualizará o valor do cabeçalho para ser o valor configurado. Caso contrário, o balanceador de carga adiciona o cabeçalho de resposta HTTP à resposta com o valor configurado.

Cabeçalho	Descrição
Strict-Transport-Security	Impõe conexões somente HTTPS pelo navegador por um período especificado, ajudando a proteger contra man-in-the-middle ataques, downgrades de protocolo e erros do usuário, garantindo que todas as comunicações entre o cliente e o alvo sejam criptografadas.
Access-Control-Allow-Origin	Controla se os recursos em um alvo podem ser acessados de diferentes origens. Isso permite

Cabeçalho	Descrição
	interações seguras entre origens, evitando o acesso não autorizado.
Access-Control-Allow-Methods	Especifica os métodos HTTP que são permitidos ao fazer solicitações de origem cruzada para o destino. Ele fornece controle sobre quais ações podem ser executadas de diferentes origens.
Access-Control-Allow-Headers	Especifica quais cabeçalhos personalizados ou não simples podem ser incluídos em uma solicitação de origem cruzada. Esse cabeçalho dá aos alvos o controle sobre quais cabeçalhos podem ser enviados por clientes de diferentes origens.
Access-Control-Allow-Credentials	Especifica se o cliente deve incluir credenciais como cookies, autenticação HTTP ou certificados de cliente em solicitações de origem cruzada.
Access-Control-Expose-Headers	Permite que o alvo especifique quais cabeçalhos de resposta adicionais podem ser acessados pelo cliente em solicitações de origem cruzada.
Access-Control-Max-Age	Define por quanto tempo o navegador pode armazenar em cache o resultado de uma solicitação de preflight, reduzindo a necessidade de repetidas verificações de preflight. Isso ajuda a otimizar o desempenho reduzindo o número de solicitações OPTIONS necessárias para determinadas solicitações de origem cruzada.

Cabeçalho	Descrição
Content-Security-Policy	Recurso de segurança que evita ataques de injeção de código, como o XSS, controlando quais recursos, como scripts, estilos, imagens etc. podem ser carregados e executados por um site.
X-Content-Type-Options	Com a diretiva no-sniff, aprimora a segurança da web impedindo que os navegadores adivinhem o tipo MIME de um recurso. Ele garante que os navegadores interpretem o conteúdo apenas de acordo com o tipo de conteúdo declarado
X-Frame-Options	Mecanismo de segurança de cabeçalho que ajuda a evitar ataques de click-jacking controlando se uma página da web pode ser incorporada em quadros. Valores como DENY e SAMEORIGIN podem garantir que o conteúdo não seja incorporado em sites maliciosos ou não confiáveis.

Desativar cabeçalhos

Usando cabeçalhos de desativação, você pode configurar seu Application Load Balancer para desativar `server:awselb/2.0` o cabeçalho das respostas. Isso reduz a exposição de informações específicas do servidor e adiciona uma camada extra de proteção ao seu aplicativo.

O nome do atributo é`routing.http.response.server.enabled`. Os valores disponíveis são `true` ou`false`. O valor padrão é `true`.

Limitações

- Os valores do cabeçalho podem conter os seguintes caracteres
  - Caracteres alfanuméricos:`a-z`,`A-Z`, e `0-9`
  - Caracteres especiais: `_ :;.,\/'?!(){}[]@<>=-+*#&`|~^%`

- O valor do atributo não pode exceder 1K bytes de tamanho.
- O Elastic Load Balancing realiza validações básicas de entrada para verificar se o valor do cabeçalho é válido. No entanto, a validação não pode confirmar se o valor é compatível com um cabeçalho específico.
- Definir um valor vazio para qualquer atributo fará com que o Application Load Balancer volte ao comportamento padrão.

Habilite a modificação do cabeçalho HTTP para seu Application Load Balancer

A modificação do cabeçalho é desativada por padrão e deve ser ativada em cada ouvinte.

Para habilitar a modificação do cabeçalho usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o Application Load Balancer.
4. Na guia Ouvintes e regras, selecione o protocolo e a porta para abrir a página de detalhes do seu ouvinte.
5. Na guia Atributos, selecione Editar.

   Os atributos do ouvinte são organizados em grupos. Você escolherá quais recursos ativar.
6. [ouvintes HTTPS] Nomes de cabeçalho modificáveis mTLS/TLS

   a. Expanda os nomes de mTLS/TLS cabeçalhos modificáveis.

   b. Permita que os cabeçalhos da solicitação sejam modificados e forneçam nomes para eles. Para obter mais informações, consulte the section called “Renomear mTLS/TLS cabeçalhos”.
7. Adicionar cabeçalhos de resposta

   a. Expanda Adicionar cabeçalhos de resposta.

   b. Ative os cabeçalhos de resposta para adicionar e fornecer valores para eles. Para obter mais informações, consulte the section called “Adicionar cabeçalhos de resposta”.
8. Cabeçalho de resposta do servidor ALB

   - Ative ou desative o cabeçalho do servidor.

9. Escolha Salvar alterações.

Para habilitar a modificação do cabeçalho usando o AWS CLI

Use o modify-listener-attributescomando com os seguintes atributos:

`routing.http.request.x_amzn_mtls_clientcert_serial_number.header_name`

Modifique o nome do cabeçalho de X-Amzn-Mtls-Clientcert-Serial-Number.

`routing.http.request.x_amzn_mtls_clientcert_issuer.header_name`

Modifique o nome do cabeçalho de X-Amzn-Mtls-Clientcert-Issuer.

`routing.http.request.x_amzn_mtls_clientcert_subject.header_name`

Modifique o nome do cabeçalho de X-Amzn-Mtls-Clientcert-Subject.

`routing.http.request.x_amzn_mtls_clientcert_validity.header_name`

Modifique o nome do cabeçalho de X-Amzn-Mtls-Clientcert-Validity.

`routing.http.request.x_amzn_mtls_clientcert_leaf.header_name`

Modifique o nome do cabeçalho de X-Amzn-Mtls-Clientcert-Leaf.

`routing.http.request.x_amzn_mtls_clientcert.header_name`

Modifique o nome do cabeçalho de X-Amzn-Mtls-Clientcert.

`routing.http.request.x_amzn_tls_version.header_name`

Modifique o nome do cabeçalho de X-Amzn-Tls-Version.

`routing.http.request.x_amzn_tls_cipher_suite.header_name`

Modifique o nome do cabeçalho de X-Amzn-Tls-Cipher-Suite.

`routing.http.response.server.enabled`

Indica se o cabeçalho do servidor de resposta HTTP deve ser permitido ou removido.

`routing.http.response.strict_transport_security.header_value`

Adicione o cabeçalho Strict-Transport-Security para informar aos navegadores que o site só deve ser acessado usando HTTPS e que qualquer tentativa futura de acessá-lo usando HTTP deve ser automaticamente convertida em HTTPS.

`routing.http.response.access_control_allow_origin.header_value`

Adicione o cabeçalho Access-Control-Allow-Origin para especificar quais origens têm permissão para acessar o servidor.

`routing.http.response.access_control_allow_methods.header_value`

Adicione o cabeçalho Access-Control-Allow-Methods para especificar quais métodos HTTP são permitidos ao acessar o servidor de uma origem diferente.

`routing.http.response.access_control_allow_headers.header_value`

Adicione o cabeçalho Access-Control-Allow-Headers para especificar quais cabeçalhos são permitidos durante uma solicitação de origem cruzada.

`routing.http.response.access_control_allow_credentials.header_value`

Adicione o cabeçalho Access-Control-Allow-Credentials para indicar se o navegador deve incluir credenciais como cookies ou autenticação em solicitações de origem cruzada.

`routing.http.response.access_control_expose_headers.header_value`

Adicione o cabeçalho Access-Control-Expose-Headers para indicar quais cabeçalhos o navegador pode expor ao cliente solicitante.

`routing.http.response.access_control_max_age.header_value`

Adicione o cabeçalho Access-Control-Max-Age para especificar por quanto tempo os resultados de uma solicitação de preflight podem ser armazenados em cache, em segundos.

`routing.http.response.content_security_policy.header_value`

Adicione o cabeçalho Content-Security-Policy para especificar as restrições impostas pelo navegador para ajudar a minimizar o risco de certos tipos de ameaças à segurança.

`routing.http.response.x_content_type_options.header_value`

Adicione o cabeçalho X-Content-Type-Options para indicar se os tipos MIME anunciados nos cabeçalhos Content-Type devem ser seguidos e não alterados.

`routing.http.response.x_frame_options.header_value`

Adicione o cabeçalho X-Frame-Options para indicar se o navegador tem permissão para renderizar uma página em um quadro, iframe, incorporação ou objeto.

Excluir um receptor para seu Application Load Balancer

Você pode excluir um listener a qualquer momento. Quando excluir um load balancer, todos os seus listeners serão excluídos.

Para excluir um listener usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Balanceador de carga.
3. Selecione o load balancer.
4. Na guia Receptores e regras, marque a caixa de seleção do receptor e escolha Gerenciar receptor, Excluir receptor.
5. Quando a confirmação for solicitada, insira **`confirm`** e selecione Excluir.

Para excluir um ouvinte usando o AWS CLI

Use o comando delete-listener.

Grupos de destino para seus Application Load Balancers

Os grupos de destino encaminham solicitações para alvos registrados individuais, como EC2 instâncias, usando o protocolo e o número da porta que você especificar. Você pode registrar um destino com vários grupos de destino. Você pode configurar verificações de integridade em cada grupo de destino. As verificações de integridade são executadas em todos os destinos registrados a um grupo de destino especificado em uma regra de listeners para seu load balancer.

Cada grupo de destino é usado para rotear solicitações para um ou mais destinos registrados. Ao criar cada regra do listener, especifique um grupo de destino e condições. Quando uma condição da regra é atendida, o tráfego é encaminhado para o grupo de destino correspondente. Você pode criar grupos de destino diferentes para tipos de solicitações diferentes. Por exemplo, você pode criar um grupo de destino para solicitações gerais e outros grupos de destino para solicitações para os microsserviços do aplicativo. Você só pode usar cada grupo de destino com um balanceador de carga. Para obter mais informações, consulte Componentes do Application Load Balancer.

Você define as configurações de verificação de integridade para seu load balancer por grupo de destino. Cada grupo de destino usa as configurações de verificação de integridade padrão, a menos que você as substitua ao criar o grupo de destino ou as modifique posteriormente. Após especificar um grupo de destino em uma regra para um listener, o load balancer monitora continuamente a integridade de todos os destinos registrados com o grupo de destino que estiverem em uma Zona de disponibilidade habilitada para o load balancer. O load balancer roteia solicitações para os destinos registrados que são íntegros.

Conteúdo

- Configuração de roteamento
- Target type
- Tipo de endereço IP
- Versão do protocolo
- Destinos registrados
- Atributos do grupo de destino
- Algoritmos de roteamento
- Integridade do grupo de destino
- Criar um grupo de destino para o Application Load Balancer
- Atualizar configurações de integridade do grupo de destino do Application Load Balancer

- Verificações de integridade para grupos de destino do Application Load Balancer
- Editar atributos do grupo de destino para o Application Load Balancer
- Registre destinos com o grupo de destino do Application Load Balancer
- Usar funções do Lambda como destino de um Application Load Balancer
- Tags para o grupo de destino do Application Load Balancer
- Excluir um grupo de destino do Application Load Balancer

Configuração de roteamento

Por padrão, um load balancer roteia solicitações para seus destinos usando o protocolo e o número da porta que você especificou ao criar o grupo de destino. Como alternativa, você pode substituir a porta usada para rotear o tráfego para um destino quando registrá-lo no grupo de destino.

Os grupos de destino são compatíveis com os seguintes protocolos e portas:

- Protocolos: HTTP, HTTPS
- Ports (Portas): 1-65535

Quando um grupo de destino estiver configurado com o protocolo HTTPS ou usar as verificações de integridade de HTTPS, se algum receptor HTTPS estiver usando uma política de segurança do TLS 1.3, a política de segurança `ELBSecurityPolicy-TLS13-1-0-2021-06` será usada para conexões de destino. Caso contrário, a política de segurança `ELBSecurityPolicy-2016-08` será usada. O balanceador de carga estabelecerá conexões TLS com os destinos usando certificados instalados nos destinos. O load balancer não valida esses certificados. Portanto, é possível usar certificados autoassinados ou certificados que tenham expirado. Como o balanceador de carga e seus destinos estão em uma nuvem privada virtual (VPC), o tráfego entre o balanceador de carga e os destinos é autenticado no nível do pacote, portanto, não corre o risco man-in-the-middle de ataques ou falsificação, mesmo que os certificados nos destinos não sejam válidos. O tráfego que sai não AWS terá essas mesmas proteções, e etapas adicionais podem ser necessárias para proteger ainda mais o tráfego.

Target type

Durante a criação de um grupo de destino, você especifica seu tipo de destino, que determina o tipo de destino especificado ao registrar destinos com esse grupo de destino. Depois de criar um grupo de destino, você não pode mudar o tipo de destino dele.

Os possíveis tipos de destino são os seguintes:

`instance`

Os destinos são especificados por ID de instância.

`ip`

Os destinos são endereços IP.

`lambda`

O destino é uma função Lambda.

Quando o tipo de destino é `ip`, você pode especificar os endereços IP de um dos seguintes blocos CIDR:

- As sub-redes da VPC para o grupo de destino
- 10.0.0.0/8 (RFC 1918)
- 100.64.0.0/10 (RFC 6598)
- 172.16.0.0/12 (RFC 1918)
- 192.168.0.0/16 (RFC 1918)

> **Important**
>
> Você não pode especificar publicamente endereços IP roteáveis.

Todos os blocos CIDR compatíveis permitem que você registre os seguintes destinos em um grupo de destino:

- Instâncias em uma VPC emparelhada com a VPC do balanceador de carga (mesma região ou região diferente).
- AWS recursos que são endereçáveis por endereço IP e porta (por exemplo, bancos de dados).
- Recursos locais vinculados AWS por meio AWS Direct Connect de uma conexão Site-to-Site VPN.

> **Note**
>
> Para Application Load Balancers implantados em uma zona local, os destinos `ip` devem estar na mesma zona local para receber tráfego.
> Para obter mais informações, consulte O que são Zonas AWS Locais?

Se você especificar destinos usando um ID de instância, o tráfego é roteado para instâncias usando o endereço IP privado especificado na interface de rede principal para a instância. Se você especificar destinos usando endereços IP, você pode rotear o tráfego para uma instância com qualquer endereço IP privado de uma ou mais interfaces de rede. Isso permite que vários aplicativos em uma instância usem a mesma porta. Cada interface de rede pode ter seu próprio security group.

Se o tipo de destino do seu grupo de destino for `lambda`, você poderá registrar uma única função Lambda. Quando o load balancer recebe uma solicitação para a função Lambda, ele invoca a função Lambda. Para obter mais informações, consulte Usar funções do Lambda como destino de um Application Load Balancer.

Você pode configurar o Amazon Elastic Container Service (Amazon ECS) como destino do seu Application Load Balancer. Para obter mais informações, consulte Use an Application Load Balancer for Amazon ECS no Amazon Elastic Container Service Developer Guide.

Tipo de endereço IP

Ao criar um novo grupo de destino, você pode selecionar o tipo de endereço IP dele. Isso controla a versão do IP usada para comunicação com os destinos e para a verificação do status de integridade deles.

Os grupos-alvo dos seus Application Load Balancers oferecem suporte aos seguintes tipos de endereço IP:

`ipv4`

O balanceador de carga se comunica com os destinos usando. IPv4

`ipv6`

O balanceador de carga se comunica com os destinos usando. IPv6

Considerações

- O balanceador de carga se comunica com os destinos com base no tipo de endereço IP do grupo de destino. Os destinos de um IPv4 grupo-alvo devem aceitar o IPv4 tráfego do balanceador de carga e os destinos de um IPv6 grupo-alvo devem aceitar o IPv6 tráfego do balanceador de carga.
- Você não pode usar um IPv6 grupo-alvo com um balanceador de `ipv4` carga.
- Você não pode registrar uma função Lambda com um IPv6 grupo-alvo.

Versão do protocolo

Por padrão, os Application Load Balancers enviam solicitações aos destinos usando HTTP/1.1. Você pode usar a versão do protocolo para enviar solicitações aos destinos usando HTTP/2 ou gRPC.

A tabela a seguir resume o resultado das combinações de protocolo de solicitação e versão de protocolo do grupo de destino.

Protocolo de solicitação	Versão do protocolo	Resultado
HTTP/1.1	HTTP/1.1	Bem-sucedida
HTTP/2	HTTP/1.1	Bem-sucedida
gRPC	HTTP/1.1	Erro
HTTP/1.1	HTTP/2	Erro
HTTP/2	HTTP/2	Bem-sucedida
gRPC	HTTP/2	Sucesso se os destinos forem compatíveis com gRPC
HTTP/1.1	gRPC	Erro
HTTP/2	gRPC	Sucesso se for uma solicitaç ão POST
gRPC	gRPC	Bem-sucedida

Considerações sobre a versão do protocolo gRPC

- O único protocolo de receptor compatível é HTTPS.
- O único tipo de ação compatível com as regras do receptor é `forward`.
- Só há compatibilidade com os tipos de destino `instance` e `ip`.
- O balanceador de carga analisa as solicitações do gRPC e encaminha as chamadas do gRPC para os grupos de destino adequados com base no pacote, serviço e método.
- O balanceador de carga é compatível com streaming unário no lado do cliente, streaming no lado do servidor e streaming bidirecional.
- Você deve fornecer um método de verificação de integridade personalizado com o formato `/package.service/method`.
- É necessário especificar os códigos de status a serem usados ao verificar uma resposta bem-sucedida de um destino.
- Não é possível usar funções do Lambda como destino.

Considerações sobre a versão do protocolo HTTP/2

- O único protocolo de receptor compatível é HTTPS.
- O único tipo de ação compatível com as regras do receptor é `forward`.
- Só há compatibilidade com os tipos de destino `instance` e `ip`.
- O balanceador de carga é compatível com streaming unário no lado do cliente, streaming no lado do servidor e streaming bidirecional. O número máximo de streams por conexão HTTP/2 do cliente é 128.

Destinos registrados

O seu load balancer serve como um ponto único de contato para clientes e distribui o tráfego de entrada nos destinos íntegros registrados. Você pode registrar cada destino com um ou mais grupos de destino.

Se a demanda da seu aplicativo aumentar, você pode registrar destinos adicionais com um ou mais grupos de destino, a fim de dar conta da demanda. O balanceador de carga inicia o roteamento do tráfego para um destino recém-registrado assim que o processo de registro é concluído e o destino passa pela verificação de integridade inicial, independentemente do limite configurado.

Se a demanda na seu aplicativo diminuir, ou se você precisar fazer manutenção nos seus destinos, você pode cancelar o registro dos destinos dos seus grupos de destino. Cancelar o registro de um destino o remove do seu grupo de destino, mas não afeta o destino de outra forma. O load balancer interrompe as solicitações de roteamento ao destino assim que o registro dele for cancelado. O destino entra no estado `draining` até que as solicitações em andamento tenham sido concluídas. Você pode registrar o destino com o grupo de destino novamente quando estiver pronto para retomar o recebimento de solicitações.

Se você estiver registrando destinos por ID de instância, poderá usar o balanceador de carga com um grupo do Auto Scaling. Depois que você anexar um grupo de destino a um grupo do Auto Scaling, o Auto Scaling registrará os destinos no grupo de destino para você quando ele os iniciar. Para obter mais informações, consulte Como anexar um balanceador de carga ao seu grupo de Auto Scaling no Guia do usuário do Amazon Auto EC2 Scaling.

Limites

- Você não pode registrar os endereços IP de outro Application Load Balancer na mesma VPC. Se o outro Application Load Balancer estiver em uma VPC emparelhada à VPC do balanceador de carga, você poderá registrar seus endereços IP.
- Você não pode registrar instâncias por ID de instância se elas estiverem em uma VPC pareada com a VPC do balanceador de carga (mesma região ou região diferente). Você poderá registrar essas instâncias pelo endereço IP.

Atributos do grupo de destino

Você pode configurar um grupo de destino editando os atributos. Para obter mais informações, consulte Editar atributos do grupo de destino.

Os seguintes atributos do grupo de destino são compatíveis se o tipo de grupo de destino for `instance` ou `ip`:

`deregistration_delay.timeout_seconds`

A quantidade de tempo que o Elastic Load Balancing deve aguardar antes de cancelar o registro de um destino. O intervalo é de 0 a 3.600 segundos. O valor de padrão é de 300 segundos.

`load_balancing.algorithm.type`

O algoritmo de balanceamento de carga determina como o load balancer seleciona os destinos ao rotear as solicitações. O valor é `round_robin`, `least_outstanding_requests` ou `weighted_random`. O padrão é `round_robin`.

`load_balancing.algorithm.anomaly_mitigation`

Disponível somente quando `load_balancing.algorithm.type` for `weighted_random`. Indica se a mitigação de anomalias está habilitada. O valor é on ou `off`. O padrão é `off`.

`load_balancing.cross_zone.enabled`

Indica se o balanceamento de carga entre zonas está habilitado. O valor é `true`, `false` ou `use_load_balancer_configuration`. O padrão é `use_load_balancer_configuration`.

`slow_start.duration_seconds`

O período, em segundos, durante o qual o load balancer envia a um destino recém-registrado uma parcela de tráfego com aumento linear ao grupo de destino. O intervalo é de 30 a 900 segundos (15 minutos). O padrão é 0 segundos (desativado).

`stickiness.enabled`

Indica se sticky sessions estão habilitadas. O valor é `true` ou `false`. O padrão é `false`.

`stickiness.app_cookie.cookie_name`

O nome do cookie da aplicação. O nome do cookie do aplicativo não pode ter os seguintes prefixos:`AWSALB`,`AWSALBAPP`, ou`AWSALBTG`; eles são reservados para uso pelo balanceador de carga.

`stickiness.app_cookie.duration_seconds`

O período de expiração de cookie baseado em aplicação, em segundos. Após esse período, o cookie será considerado antigo. O valor mínimo é 1 segundo e o valor máximo é 7 dias (604.800 segundos). O valor padrão é de 1 dia (86.400 segundos).

`stickiness.lb_cookie.duration_seconds`

O período de expiração do cookie baseado em duração, em segundos. Após esse período, o cookie será considerado antigo. O valor mínimo é 1 segundo e o valor máximo é 7 dias (604.800 segundos). O valor padrão é de 1 dia (86.400 segundos).

`stickiness.type`

O tipo de perdurabilidade. Os valores possíveis são `lb_cookie` e `app_cookie`.

`target_group_health.dns_failover.minimum_healthy_targets.count`

O número mínimo de destinos que devem estar íntegros. Se o número de destinos íntegros estiver abaixo desse valor, marque o nó como não íntegro no DNS, para que o tráfego seja roteado somente para nós íntegros. Os valores possíveis são `off` ou um número inteiro de 1 até o número máximo de destinos. Quando`off`, a falha de DNS é desativada, o que significa que, mesmo que todos os destinos no grupo-alvo não estejam íntegros, o nó não é removido do DNS. O padrão é um.

`target_group_health.dns_failover.minimum_healthy_targets.percentage`

O percentual mínimo de destinos que devem estar íntegros. Se a porcentagem de destinos íntegros for menor do que esse valor, marque o nó como não íntegro no DNS, para que o tráfego seja roteado somente para nós íntegros. Os valores possíveis são `off` ou um número inteiro de 1 a 100. Quando`off`, a falha de DNS é desativada, o que significa que, mesmo que todos os destinos no grupo-alvo não estejam íntegros, o nó não é removido do DNS. O padrão é `off`.

`target_group_health.unhealthy_state_routing.minimum_healthy_targets.count`

O número mínimo de destinos que devem estar íntegros. Se o número de destinos íntegros for menor do que desse valor, envie tráfego para todos os alvos, incluindo alvos não íntegros. O intervalo é de 1 ao número máximo de destinos. O padrão é um.

`target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage`

O percentual mínimo de destinos que devem estar íntegros. Se a porcentagem de destinos íntegros for menor do que valor, envie tráfego para todos os destinos, incluindo destinos não íntegros. Os valores possíveis são `off` ou um número inteiro de 1 a 100. O padrão é `off`.

O seguinte atributo do grupo de destino é compatível se o tipo de grupo de destino for `lambda`:

`lambda.multi_value_headers.enabled`

Indica se os cabeçalhos da solicitação e resposta trocados entre o load balancer e a função Lambda incluem matrizes de valores ou strings. Os valores possíveis são `true` ou `false`. O valor padrão é `false`. Para obter mais informações, consulte Cabeçalhos de vários valores.

Algoritmos de roteamento

Um algoritmo de roteamento é o método usado pelo balanceador de carga para determinar quais destinos receberão solicitações. Por padrão, o algoritmo de roteamento de ida e volta é usado para

rotear solicitações no nível do grupo de destino. As solicitações menos pendentes e os algoritmos de roteamento aleatório ponderado também estão disponíveis com base nas necessidades da aplicação. Um grupo de destino só pode ter um algoritmo de roteamento ativo por vez, no entanto, o algoritmo de roteamento pode ser atualizado sempre que necessário.

Se você habilitar as sessões persistentes, o algoritmo de roteamento selecionado será usado para a seleção inicial de destino. Solicitações futuras do mesmo cliente serão encaminhadas para o mesmo destino, ignorando o algoritmo de roteamento selecionado.

Ida e volta

- O algoritmo de roteamento de ida e volta direciona as solicitações uniformemente entre destinos íntegros no grupo de destino, em uma ordem sequencial.
- Esse algoritmo é comumente usado quando as solicitações recebidas têm complexidade semelhante, os destinos registrados são semelhantes em capacidade de processamento ou se você precisa distribuir as solicitações igualmente entre os destinos.

Solicitações menos pendentes

- O algoritmo de roteamento de solicitações menos pendentes encaminha as solicitações para os destinos com o menor número de solicitações em andamento.
- Esse algoritmo é comumente usado quando as solicitações recebidas variam em complexidade, os destinos registrados variam em capacidade de processamento.
- Quando um balanceador de carga compatível com HTTP/2 estiver usando destinos que sejam compatíveis apenas com HTTP/1.1, ele converterá a solicitação em várias solicitações HTTP/1.1. Nessa configuração, o algoritmo de solicitações menos pendentes tratará cada solicitação HTTP/2 como várias solicitações.
- Ao usar WebSockets, o destino é selecionado usando o algoritmo de solicitações menos pendentes. Uma vez selecionado, o balanceador de carga cria uma conexão com o destino e envia todas as mensagens por essa conexão.
- O algoritmo de roteamento de solicitações menos pendentes não pode ser usado com o modo de início lento.

Aleatório ponderado

- O algoritmo de roteamento aleatório ponderado direciona as solicitações uniformemente entre destinos íntegros no grupo de destino, em ordem aleatória.

- Esse algoritmo oferece suporte à mitigação de anomalias de ponderações de destinos automáticos (ATW).
- O algoritmo de roteamento aleatório ponderado não pode ser usado com o modo de início lento.
- O algoritmo de roteamento aleatório ponderado não pode ser usado com sessões persistentes.

Modificar o algoritmo de roteamento de um grupo de destino

É possível modificar o algoritmo de roteamento do grupo de destino a qualquer momento.

Para modificar o algoritmo de roteamento usando o novo console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na página de detalhes dos grupos de destino, na guia Atributos, escolha Editar.
5. Na página Editar atributos do grupo de destino, na seção Configuração de tráfego, em Algoritmo de balanceamento de carga, escolha Ida e volta, Solicitações menos pendentes ou Ponderado aleatório.
6. Escolha Salvar alterações.

Para modificar o algoritmo de roteamento usando o AWS CLI

Use o comando modify-target-group-attributes com o atributo `load_balancing.algorithm.type`.

Integridade do grupo de destino

Por padrão, um grupo de destino é considerado íntegro desde que tenha pelo menos um destino íntegro. Se você tiver uma frota grande, não é suficiente ter apenas um destino íntegro distribuindo o tráfego. Em vez disso, você pode especificar uma contagem ou percentual mínimo de destinos que devem estar íntegros e quais ações o balanceador de carga executa quando os destinos íntegros ficarem abaixo do limite especificado. Isso melhora a disponibilidade do seu aplicativo.

Conteúdo

- Ações para estado não íntegro

- Requisitos e considerações
- Monitoramento
- Exemplo
- Como usar o failover de DNS do Route 53 para o seu balanceador de carga

Ações para estado não íntegro

Você pode configurar os limites íntegros para as seguintes ações:

- Failover de DNS — Quando os destinos íntegros em uma zona ficam abaixo do limite, marcamos os endereços IP do nó do balanceador de carga da zona como não íntegros no DNS. Portanto, quando os clientes resolvem o nome DNS do balanceador de carga, o tráfego é roteado somente para zonas íntegras.
- Failover de roteamento — quando os destinos íntegros em uma zona ficam abaixo do limite, o balanceador de carga envia tráfego para todos os destinos que estão disponíveis para o nó do balanceador de carga, incluindo destinos não íntegros. Isso aumenta a probabilidade de sucesso da conexão de um cliente, especialmente quando os destinos temporariamente são reprovados nas verificações de integridade, e reduz o risco de sobrecarga dos destinos íntegros.

Requisitos e considerações

- Você não pode usar esse recurso com grupos de destino nos quais o destino seja uma função do Lambda. Se o Application Load Balancer for o destino de um Network Load Balancer ou Global Accelerator, não configure um limite para o failover de DNS.
- Se você especificar os dois tipos de limites para uma ação (contagem e percentual), o balanceador de carga executará a ação quando um dos limites for violado.
- Se você especificar limites para ambas as ações, o limite para failover de DNS deverá ser maior ou igual ao limite para failover de roteamento, de modo que o failover de DNS ocorra com o failover de roteamento ou antes dele.
- Se você especificar o limite como um percentual, calcularemos o valor dinamicamente com base no número total de destinos registrados nos grupos de destino.
- O número total de destinos depende do balanceamento de carga entre zonas estar ativado ou desativado. Se o balanceamento de carga entre zonas estiver desativado, cada nó enviará tráfego somente para os destinos na sua própria zona, o que significa que os limites se aplicarão ao número de destinos em cada zona habilitada separadamente. Se o balanceamento de carga entre

zonas estiver ativado, cada nó enviará tráfego a todos os destinos em todas as zonas habilitadas, o que significa que os limites especificados se aplicarão ao número total de destinos em todas as zonas habilitadas. Para obter mais informações, consulte Balanceamento de carga entre zonas para grupos de destino do Application Load Balancer.

- Quando ocorre um failover de DNS, ele afeta todos os grupos-alvo associados ao balanceador de carga. Verifique se você tem capacidade suficiente nas zonas restantes para processar esse tráfego adicional, especialmente se o balanceamento de carga entre zonas estiver desativado.
- Com o failover de DNS, removemos os endereços IP das zonas não íntegras do nome do host DNS do balanceador de carga. No entanto, o cache DNS do cliente local pode conter esses endereços IP até que o time-to-live (TTL) no registro DNS expire (60 segundos).
- Com o failover de DNS, se houver vários grupos-alvo vinculados a um Application Load Balancer e um grupo-alvo não estiver íntegro em uma zona, as verificações de integridade do DNS serão bem-sucedidas se pelo menos um outro grupo-alvo estiver íntegro nessa zona.
- Com o failover de DNS, se todas as zonas do balanceador de carga forem consideradas não íntegras, o balanceador de carga enviará tráfego para todas as zonas, incluindo as zonas não íntegras.
- Além da existência de destinos íntegros em número suficiente, há outros fatores que podem levar ao failover de DNS, como a integridade da zona.

Monitoramento

Para monitorar a integridade de seus grupos-alvo, consulte CloudWatch as métricas da saúde do grupo-alvo.

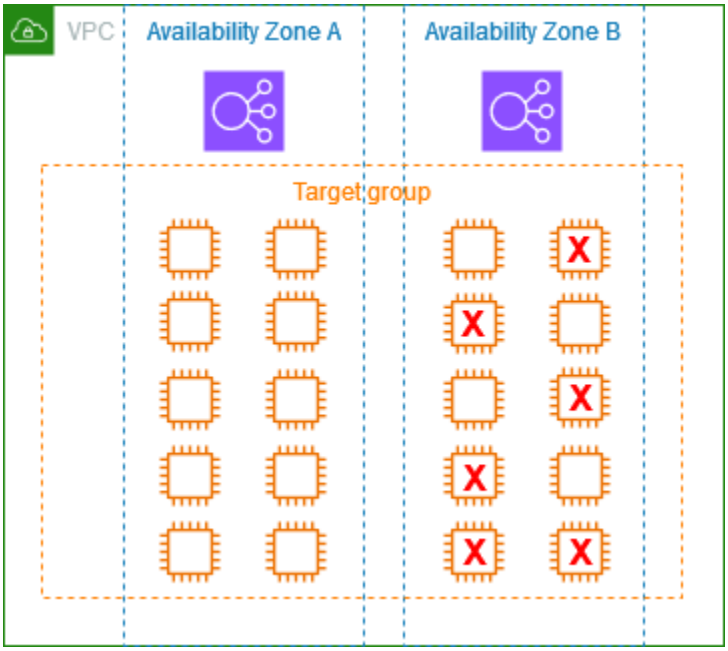
Exemplo

O exemplo a seguir demonstra como as configurações de integridade do grupo de destino são aplicadas.

Cenário

- Um balanceador de carga compatível com duas zonas de disponibilidade, A e B
- Cada zona de disponibilidade contém 10 destinos registrados
- O grupo de destino tem as seguintes configurações de integridade:
  - Failover de DNS: 50%

  - Failover de roteamento: 50%
- Seis destinos apresentam falha na zona de disponibilidade B



Se o balanceamento de carga entre zonas estiver desativado

- O nó do balanceador de carga em cada zona de disponibilidade só pode enviar tráfego para os 10 destinos em sua zona de disponibilidade.
- Há 10 destinos íntegros na zona de disponibilidade A, o que atende ao percentual necessário de destinos íntegros. O balanceador de carga continua distribuindo o tráfego entre os 10 destinos íntegros.
- Há apenas 4 destinos íntegros na zona de disponibilidade B, o que representa 40% dos destinos do nó do balanceador de carga na zona de disponibilidade B. Como isso é inferior ao percentual necessário de destinos íntegros, o balanceador de carga executará as seguintes ações:
  - Failover de DNS: a zona de disponibilidade B será marcada como não íntegra no DNS. Como os clientes não conseguem resolver o nome do balanceador de carga para o nó do balanceador de carga na zona de disponibilidade B e a zona de disponibilidade A está íntegra, os clientes enviam novas conexões para a zona de disponibilidade A.
  - Failover de roteamento: quando novas conexões são enviadas explicitamente para a zona de disponibilidade B, o balanceador de carga distribui o tráfego para todos os destinos na zona de disponibilidade B, incluindo os destinos não íntegros. Isso evita interrupções entre os destinos íntegros restantes.

Se o balanceamento de carga entre zonas estiver ativado

- Cada nó do balanceador de carga pode enviar tráfego para todos os 20 destinos registrados em ambas as zonas de disponibilidade.
- Há 10 destinos íntegros na zona de disponibilidade A e 4 destinos íntegros na zona de disponibilidade B, totalizando 14 destinos íntegros. Isso representa 70% dos destinos para os nós do balanceador de carga em ambas as zonas de disponibilidade, o que atende ao percentual necessário de destinos íntegros.
- O balanceador de carga distribui tráfego entre os 14 destinos íntegros nas duas zonas de disponibilidade.

Como usar o failover de DNS do Route 53 para o seu balanceador de carga

Se você usa o Route 53 para rotear consultas de DNS para seu balanceador de carga, também poderá configurar o failover de DNS para o seu balanceador de carga usando o Route 53. Em uma configuração de failover, o Route 53 verifica a integridade dos destinos dos grupos de destino do balanceador de carga para determinar se eles estão disponíveis. Se não houver destinos íntegros registrados no balanceador de carga ou se o próprio balanceador de carga não estiver íntegro, o Route 53 roteará o tráfego para outro recurso disponível, como um balanceador de carga íntegro ou um site estático no Amazon S3.

Por exemplo, vamos supor que você tenha uma aplicação Web para `www.example.com` e deseja instâncias redundantes em execução por trás de dois balanceadores de carga que residam em diferentes regiões. Você deseja que o tráfego seja roteado primariamente para o balanceador de carga em uma região e quer usar o balanceador de carga na outra região como backup durante falhas. Se você configurar o failover de DNS, poderá especificar os balanceadores de carga primário e secundário (backup). O Route 53 direcionará o tráfego para o balanceador de carga primário, se estiver disponível, ou para o balanceador de carga secundário, em caso contrário.

Como funciona a avaliação da saúde alvo

- Se avaliar a integridade do alvo estiver definido `Yes` em um registro de alias para um Application Load Balancer, o Route 53 avaliará a integridade do recurso especificado pelo valor. `alias target` O Route 53 usa as verificações de saúde do grupo-alvo.
- Se todos os grupos-alvo vinculados a um Application Load Balancer estiverem íntegros, o Route 53 marcará o registro de alias como íntegro. Se você configurou um limite para um grupo-alvo e ele atinge esse limite, ele passa nas verificações de saúde. Caso contrário, se um grupo-

alvo contiver pelo menos um alvo saudável, ele será aprovado nas verificações de saúde. Se as verificações de saúde forem aprovadas, o Route 53 retornará registros de acordo com sua política de roteamento. Se uma política de roteamento de failover for usada, o Route 53 retornará o registro principal.

- Se algum dos grupos de destino anexados a um Application Load Balancer não estiver íntegro, o registro do alias falhará na verificação de integridade do Route 53 (abertura de falha). Se estiver usando avaliar a integridade do alvo, a política de roteamento de failover redireciona o tráfego para o recurso secundário.
- Se todos os grupos de destino anexados a um Application Load Balancer estiverem vazios (sem destinos), o Route 53 considerará o registro não íntegro (abertura de falha). Se estiver usando avaliar a integridade do alvo, a política de roteamento de failover redireciona o tráfego para o recurso secundário.

Para obter mais informações, consulte Uso dos limites de integridade do grupo-alvo do balanceador de carga para melhorar a disponibilidade no AWS blog e Configuração do failover de DNS no Guia do desenvolvedor do Amazon Route 53.

Criar um grupo de destino para o Application Load Balancer

Você registra seus destinos com um grupo de destino. Por padrão, o load balancer envia solicitações para destinos registrados usando a porta e o protocolo especificados por você para o grupo de destino. Você pode substituir essa porta ao registrar cada destino no grupo de destino.

Depois de criar um grupo de destino, você pode adicionar tags.

Para rotear o tráfego aos destino em um grupo de destino, especifique o grupo de destino em uma ação quando você criar um listener ou uma regra para o listener. Para obter mais informações, consulte Regras do receptor para seu Application Load Balancer. Você pode especificar o mesmo grupo de destino em vários receptores, mas esses receptores devem pertencer ao mesmo Application Load Balancer. Para usar um grupo de destino com um balanceador de carga, você deve verificar se ele não está sendo usado por um receptor para qualquer outro balanceador de carga.

Você pode adicionar ou remover destinos do seu grupo de destino a qualquer momento. Para obter mais informações, consulte Registre destinos com o grupo de destino do Application Load Balancer. Você também pode modificar as configurações de verificação de integridade para seu grupo de destino. Para obter mais informações, consulte Atualizar as configurações de verificação de integridade de um grupo de destino do Application Load Balancer.

Para criar um grupo de destino usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Selecione Criar grupo de destino.
4. Em Escolher um tipo de destino, selecione Instâncias para registrar destinos por ID de instância, Endereços IP para registrar destinos por endereço IP ou Função do Lambda para registrar uma função do Lambda como destino.
5. Em Nome do grupo de destino, digite um nome para o novo grupo de destino. Esse nome deve ser exclusivo por região e por conta, pode ter o máximo de 32 caracteres, deve conter apenas caracteres alfanuméricos ou hifens, e não deve iniciar nem terminar com hífen.
6. (Opcional) Nos itens Protocolo e Porta, modifique os valores padrão conforme o necessário.
7. Se o tipo de destino for Instâncias ou endereços IP, escolha IPv4ou IPv6como o tipo de endereço IP, caso contrário, vá para a próxima etapa.

   Observe que somente destinos que tenham o tipo de endereço IP selecionado podem ser incluídos nesse grupo de destino. O tipo de endereço IP não pode ser alterado após a criação do grupo de destino.
8. Em VPC, selecione uma nuvem privada virtual (VPC). Observe que, para os tipos de destino de endereços IP, os VPCs disponíveis para seleção são aqueles que oferecem suporte ao tipo de endereço IP que você escolheu na etapa anterior.
9. (Opcional) Em Versão do protocolo, modifique os valores padrão conforme necessário.
10. (Opcional) Na seção Verificações de integridade, modifique as configurações padrão conforme necessário.
11. Se o tipo de destino for função do Lambda, será possível habilitar as verificações de integridade selecionando Habilitar na seção Verificações de integridade.
12. (Opcional) Adicione uma ou mais tags, da seguinte forma:

    a. Expanda a seção Tags.

    b. Escolha Adicionar Tag.

    c. Insira a chave e o valor da etiqueta.
13. Escolha Próximo.
14. (Opcional) Adicione um ou mais destinos da seguinte forma:

- Se o tipo de destino for Instâncias, selecione uma ou mais instâncias, insira uma ou mais portas e escolha Incluir como pendente abaixo.

  Observação: as instâncias devem ter um IPv6 endereço principal atribuído para serem registradas em um IPv6 grupo-alvo.

- Se o tipo de destino for Endereços IP, faça o seguinte:

  a. Selecione uma rede VPC na lista ou escolha Outros endereços IP privados.

  b. Insira o endereço IP manualmente ou encontre o endereço IP usando os detalhes da instância. É possível inserir até cinco endereços IP por vez.

  c. Insira as portas para rotear o tráfego para os endereços IP especificados.

  d. Escolha Incluir como pendente abaixo.

- Se o tipo de destino for uma função do Lambda, especifique uma única função do Lambda ou ignore essa etapa e especifique uma função do Lambda posteriormente.

15. Selecione Criar grupo de destino.
16. (Opcional) É possível especificar o grupo de destino em uma regra de listener. Para obter mais informações, consulte Adicionar uma regra.

Para criar um grupo-alvo usando o AWS CLI

Use o create-target-groupcomando para criar o grupo-alvo, o comando add-tags para marcar seu grupo-alvo e o comando register-targets para adicionar alvos.

Atualizar configurações de integridade do grupo de destino do Application Load Balancer

Você pode modificar as configurações de integridade do grupo de destino conforme exibido a seguir.

Para modificar as configurações de integridade do grupo de destino usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Balanceamento de carga, selecione Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na guia Atributos, escolha Editar.

5. Verifique se o balanceamento de carga entre zonas está ativado ou desativado. Atualize essa configuração conforme necessário para garantir que você tenha capacidade suficiente para processar o tráfego adicional se uma zona falhar.
6. Expanda os requisitos de integridade do grupo de destino.
7. Em Tipo de configuração, recomendamos que você escolha Configuração unificada, que define o mesmo limite para ambas as ações.
8. Em Requisitos de estado íntegro, execute uma das seguintes ações:
   - Escolha Contagem mínima de destinos íntegros e, em seguida, insira um número de 1 até o número máximo de destinos para seu grupo de destino.
   - Escolha Porcentagem mínima de destinos íntegros e, em seguida, insira um número de 1 a 100.
9. Escolha Salvar alterações.

Para modificar as configurações de saúde do grupo-alvo usando o AWS CLI

Use o comando modify-target-group-attributes. O exemplo a seguir define como 50% o limite de integridade de ambas as ações de estado não íntegro.

```
aws elbv2 modify-target-group-attributes \
--target-group-arn arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \
--attributes
 Key=target_group_health.dns_failover.minimum_healthy_targets.percentage,Value=50 \

 Key=target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage,Value=50
```

Verificações de integridade para grupos de destino do Application Load Balancer

Seu Application Load Balancer envia periodicamente solicitações para seus destinos registrados para testar o status deles. Esses testes se chamam verificações de integridade.

Cada nó do load balancer só roteia solicitações para os destinos íntegros nas Zonas de disponibilidade habilitadas para o load balancer. Cada nó do load balancer verifica a integridade de cada destino usando as configurações de verificação de integridade para os grupos de destino

em que o destino é registrado. Após o destino ser registrado, ele deverá ser aprovado em uma verificação de integridade para ser considerado íntegro. Após cada verificação de integridade ser concluída, o nó do load balancer fechará a conexão estabelecida para a verificação de integridade.

Se um grupo de destino contiver somente destinos registrados não íntegros, o balanceador de carga encaminhará as solicitações para todos esses destinos, independentemente do status de integridade. Isso significa que se todos os destinos falharem nas verificações de integridade ao mesmo tempo em todas as zonas de disponibilidade habilitadas, o balanceador de carga apresentará falha ao abrir. O efeito da falha na abertura é permitir o tráfego para todos os destinos em todas as zonas de disponibilidade habilitadas, independentemente do seu estado de integridade, mas com base no algoritmo de balanceamento de carga.

As verificações de saúde não são compatíveis WebSockets.

Para obter mais informações, consulte the section called “Integridade do grupo de destino”.

Configurações de verificação de integridade

Você pode configurar verificações de integridade para os destinos em um grupo de destino conforme descrito na tabela a seguir. Os nomes das configurações usados na tabela são os nomes usados na API. O balanceador de carga envia uma solicitação de verificação de integridade para cada destino registrado a cada HealthCheckIntervalSecondssegundo, usando a porta, o protocolo e o caminho de verificação de integridade especificados. Cada solicitação de verificação de integridade é independente e o resultado dura por todo o intervalo. O tempo necessário para o destino responder não afeta o intervalo da próxima solicitação de verificação de integridade. Se as verificações de integridade excederem as falhas UnhealthyThresholdCountconsecutivas, o balanceador de carga colocará o alvo fora de serviço. Quando as verificações de integridade excedem os sucessos HealthyThresholdCountconsecutivos, o balanceador de carga coloca o alvo de volta em serviço.

Observe que quando você cancela o registro de um alvo, isso diminui, HealthyHostCountmas não aumenta. UnhealthyHostCount

Configuração	Descrição
HealthCheckProtocol	O protocolo que o load balancer usa ao executar verificações de integridade nos destinos. Em Application Load Balancers, os possíveis protocolos são HTTP e HTTPS. O padrão é o protocolo HTTP.

Configuração	Descrição
	Esses protocolos usam o método HTTP GET para enviar solicitações de verificação de integridade.
HealthCheckPort	A porta que o load balancer usa ao executar verificações de integridade nos destinos. O padrão é usar a porta em que cada destino recebe o tráfego do load balancer.
HealthCheckPath	O destino para verificações de integridade nos destinos. Se a versão do protocolo for HTTP/1.1 ou HTTP/2, especifique um URI válido (/path?query). O padrão é /. Se a versão do protocolo for gRPC, especifiq ue o caminho de um método personalizado de verificação de integridade com o formato `/package.service/method` . O padrão é `/AWS.ALB/healthcheck` .
HealthCheckTimeoutSeconds	O tempo, em segundos, durante o qual ausência de resposta de um destino significa uma falha na verificação de integridade. O intervalo é de 2 a 120 segundos. O padrão é de 5 segundos se o tipo de destino é `instance` ou `ip` e de 30 segundos se o tipo de destino é `lambda`.
HealthCheckIntervalSeconds	A quantia aproximada de tempo, em segundos, entre as verificações de integridade de um destino individual. O intervalo é de 5 a 300 segundos. O padrão é de 30 segundos se o tipo de destino é `instance` ou `ip` e de 35 segundos se o tipo de destino é `lambda`.

Configuração	Descrição
HealthyThresholdCount	O número de verificações de integridade bem-sucedidas consecutivas necessárias antes de considerar íntegro um destino não íntegro. O intervalo é de 2 a 10. O padrão é 5.
UnhealthyThresholdCount	O número de verificações de integridade consecutivas exigido antes considerar um destino não íntegro. O intervalo é de 2 a 10. O padrão é 2.
Matcher	O códigos a serem usados ao verificar uma resposta bem-sucedida de um destino. Eles são chamados de códigos de sucesso no console. Se a versão do protocolo for HTTP/1.1 ou HTTP/2, os valores possíveis são de 200 a 499. Você pode especificar valores múltiplos (por exemplo, "200,202") ou um intervalo valores (por exemplo, "200-299"). O valor padrão é 200. Se a versão do protocolo for gRPC, os valores possíveis são de 0 a 99. Você pode especific ar valores múltiplos (por exemplo, "0,1") ou um intervalo valores (por exemplo, "0-5"). O valor padrão é 12.

Status de integridade do destino

Antes que o load balancer envie uma solicitação de verificação de integridade para um destino, você deverá registrá-lo com um grupo de destino, especificar o grupo de destino em uma regra do listener e garantir que a Zona de disponibilidade do destino esteja habilitado para o load balancer. Antes de um destino receber solicitações do load balancer, ele deverá ser aprovado nas verificações de

integridade iniciais. Após o destino ser aprovado nas verificações de integridade iniciais, o status será `Healthy`.

A tabela a seguir descreve os valores possíveis para o status de integridade de um destino registrado.

Valor	Descrição
`initial`	O load balancer está no processo de registro do destino ou executando as verificações de integridade iniciais no destino. Códigos de motivo relacionados: `Elb.RegistrationInProgress` | `Elb.InitialHealthChecking`
`healthy`	O destino é íntegro. Códigos de motivo relacionados: nenhum
`unhealthy`	O destino não respondeu a uma verificação de integridade ou falhou em uma verificação de integridade. Códigos de motivo relacionados: `Target.ResponseCodeMismatch` | `Target.Timeout` | `Target.FailedHealthChecks` | `Elb.InternalError`
`unused`	O destino não está registrado em um grupo de destino, o grupo de destino não é usado em uma regra do listener, o destino está em uma zona de disponibilidade desativada ou o destino está no estado parado ou encerrado. Códigos de motivo relacionados: `Target.NotRegistered` | `Target.NotInUse` | `Target.InvalidState` | `Target.IpUnusable`
`draining`	O destino está cancelando o registro e está acontecendo drenagem da conexão.

Valor	Descrição
	Código de motivo relacionado: `Target.DeregistrationInProgress`
`unavailable`	As verificações de integridade estão desativadas para o grupo de destino. Código de motivo relacionado: `Target.HealthCheckDisabled`

Códigos de motivo de verificação de integridade

Se o status de um destino for qualquer valor diferente de `Healthy`, a API retornará um código de motivo e uma descrição do problema; o console exibirá a mesma descrição. Os códigos de motivo que começarem com `Elb` são originados no load balancer, e os códigos de motivo que começarem com `Target` são originados no destino. Para obter mais informações sobre as possíveis causas de falhas na verificação de integridade, consulte Solução de problemas.

Código do motivo	Descrição
`Elb.InitialHealthChecking`	Verificações de integridade iniciais em andamento
`Elb.InternalError`	As verificações de integridade falharam devido a um erro interno
`Elb.RegistrationInProgress`	O registro do destino está em andamento
`Target.DeregistrationInProgress`	O cancelamento do registro do destino está em andamento
`Target.FailedHealthChecks`	Verificações de integridade com falha
`Target.HealthCheckDisabled`	As verificações de integridade estão desativadas
`Target.InvalidState`	O destino está no estado interrompido

Código do motivo	Descrição
	O destino está no estado encerrado O destino está no estado encerrado ou interrompido O destino está em um estado inválido
`Target.IpUnusable`	O endereço IP não pode ser usado como um destino, uma vez que está sendo usado por um load balancer.
`Target.NotInUse`	O grupo de destino não está configurado para receber tráfego do load balancer O destino está em uma Zona de disponibilidade que não está habilitada para o load balancer
`Target.NotRegistered`	O destino não está registrado no grupo de destino
`Target.ResponseCodeMismatch`	As verificações de integridade apresentaram falhas com estes códigos: [código]
`Target.Timeout`	Solicitação expirada

Verificar a integridade dos destinos do Application Load Balancer

Você pode verificar a integridade dos destinos registrados com seus grupos de destino.

Para verificar a integridade dos seus destinos usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na guia Destinos, a coluna Status indica o status de cada destino.
5. Se o status for qualquer valor diferente de `Healthy`, a coluna Detalhes do status conterá mais informações. Para obter ajuda com falhas na verificação de integridade, consulte Solução de problemas.

Para verificar a saúde de seus alvos usando o AWS CLI

Use o comando describe-target-health. O resultado desse comando contém o estado de integridade do destino. Se o status for qualquer valor diferente de `Healthy`, a saída também inclui um código de motivo.

Como receber notificações por e-mail sobre destinos não íntegros

Use CloudWatch alarmes para acionar uma função Lambda para enviar detalhes sobre alvos não íntegros. Para step-by-step obter instruções, consulte a seguinte postagem no blog: Identificação de alvos não íntegros do seu balanceador de carga.

Atualizar as configurações de verificação de integridade de um grupo de destino do Application Load Balancer

Você pode atualizar as configurações de verificação de integridade do grupo de destino a qualquer momento.

Para atualizar as configurações de verificação de integridade de um grupo de destino usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na guia Detalhes do grupo, na seção Configurações da verificação de integridade, escolha Editar.
5. Na página Editar configurações da verificação de integridade, modifique as configurações conforme necessário e escolha Salvar alterações.

Para modificar as configurações de verificação de saúde de um grupo-alvo usando o AWS CLI

Use o comando modify-target-group.

Editar atributos do grupo de destino para o Application Load Balancer

Depois de criar um grupo de destino para o Application Load Balancer, você pode editar os atributos do grupo de destino dele.

Atributos do grupo de destino

- Atraso do cancelamento do registro
- Modo de iniciação lenta
- Balanceamento de carga entre zonas para grupos de destino do Application Load Balancer
- Ponderações de destinos automáticos (ATW)
- Sessões persistentes para seu Application Load Balancer

Atraso do cancelamento do registro

O Elastic Load Balancing interrompe o envio de solicitações aos destinos cujo registro esteja sendo cancelado. Por padrão, o Elastic Load Balancing aguarda 300 segundos antes de concluir o processo de cancelamento do registro, o que pode ajudar na conclusão das solicitações em trânsito para o destino. Para alterar o tempo que o Elastic Load Balancing aguarda, atualize o valor de atraso de cancelamento de registro. .

O estado inicial de um destino que terá o registro cancelado é `draining`. Depois de decorrido o retardo de cancelamento do registro, processo será concluído e o estado do destino será `unused`. Se o destino for parte de um grupo do Auto Scaling, ele poderá ser encerrado e substituído.

Se um destino cujo registro esteja sendo cancelado não tiver solicitações em trânsito nem conexões ativas, o Elastic Load Balancing concluirá imediatamente o processo de cancelamento de registro, sem aguardar o término do tempo de espera. No entanto, mesmo que o cancelamento do registro de destino seja concluído, o status do destino será exibido como `draining` até que o tempo limite de atraso do cancelamento do registro termine. Depois que o tempo limite expirar, o destino passará para um estado `unused`.

Se cancelar o registro de um destino encerrar a conexão antes de o retardo de cancelamento do registro passar, o cliente receberá uma resposta de erro de nível 500.

Para atualizar o valor de retardo de cancelamento do registro usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na guia Detalhes do grupo, na seção Atributos, escolha Editar.
5. Na página Editar atributos, altere o valor do Atraso do cancelamento do registro conforme o necessário.
6. Escolha Salvar alterações.

Para atualizar o valor do atraso de cancelamento de registro usando o AWS CLI

Use o comando modify-target-group-attributes com o atributo `deregistration_delay.timeout_seconds`.

Modo de iniciação lenta

Por padrão, um destino começa a receber toda sua parte de solicitações assim que for registrado com um grupo de destino e enviar uma verificação de integridade inicial. Usar o modo de iniciação lenta oferece tempo para que os destinos aqueçam antes que o load balancer envie toda a parte de solicitações.

Com a iniciação lenta habilitada para um grupo de destino, os destinos entrarão no modo de iniciação lenta quando forem considerados íntegros pelo grupo de destino. Um destino sai do modo de iniciação lenta quando a duração da iniciação lenta configurada expira ou o destino se torna não íntegro. O load balancer aumenta linearmente o número de solicitações enviadas a um destino no modo de iniciação lenta. Assim que um destino íntegro deixa o modo de iniciação lenta, o balanceador de carga pode enviar uma parcela total de solicitações para esse destino.

Considerações

- Quando você habilita a iniciação lenta para um grupo de destino, os destinos íntegros que já estão registrados no grupo não entram no modo de iniciação lenta.
- Ao habilitar a iniciação lenta para um grupo de destino vazio e registrar destinos usando uma única operação de registro, esses destinos não entram no modo de iniciação rápida. Os destinos recém-

registrados entram no modo de iniciação lenta somente quando há pelo menos um destino íntegro que não esteja no modo de iniciação lenta.

- Se você cancelar o registro de um destino no modo de iniciação lenta, o destino sai do modo. Se você registrar o mesmo destino novamente, ele entrará no modo de iniciação lenta quando for considerado íntegro pelo grupo de destino.
- Se um destino no modo de iniciação lenta se tornar não íntegro, o destino sairá do modo de iniciação lenta. Quando o destino se tornar íntegro, ele entrará novamente no modo de iniciação lenta.
- Você não pode ativar o modo de início lento ao usar as solicitações menos pendentes ou algoritmos de roteamento aleatório ponderado.

Para atualizar o valor de duração da iniciação lenta usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na guia Detalhes do grupo, na seção Atributos, escolha Editar.
5. Na página Editar atributos, altere o valor da Duração da iniciação lenta conforme necessário. Para desativar o modo de iniciação lenta, defina a duração para 0.
6. Escolha Salvar alterações.

Para atualizar o valor da duração do início lento usando o AWS CLI

Use o comando modify-target-group-attributes com o atributo `slow_start.duration_seconds`.

Balanceamento de carga entre zonas para grupos de destino do Application Load Balancer

Os nós do load balancer distribuem solicitações de clientes para destinos registrados. Quando o balanceamento de carga entre zonas estiver ativado, cada nó do balanceador de carga distribuirá o tráfego aos destinos registrados em todas as zonas de disponibilidade registradas. Quando o balanceamento de carga entre zonas estiver desativado, cada nó do balanceador de carga distribuirá o tráfego somente para os destinos registrados na respectiva zona de disponibilidade. Isso poderá ser usado se os domínios de falha de zona tiverem preferência em relação aos regionais, garantindo

que uma zona íntegra não seja afetada por uma zona não íntegra ou para melhorias gerais na latência.

Com os Application Load Balancers, o balanceamento de carga entre zonas sempre está ativado por balanceador de carga e não pode ser desativado. Para grupos de destino, o padrão é usar a configuração do balanceador de carga, mas você pode substituir o padrão ativando ou desativando explicitamente o balanceamento de carga entre zonas em nível de grupo de destino.

Considerações

- Não há compatibilidade com persistência do destino quando o balanceamento de carga entre zonas estiver desativado.
- Não há compatibilidade com funções do Lambda quando o balanceamento de carga entre zonas estiver desativado.
- A tentativa de desativar o balanceamento de carga entre zonas por meio da API `ModifyTargetGroupAttributes` se algum destino tiver um parâmetro `AvailabilityZone` definido como `all` resultará em um erro.
- Ao registrar destino, o parâmetro `AvailabilityZone` é obrigatório. Só é permitido usar valores específicos de zona de disponibilidade quando o balanceamento de carga entre zonas estiver desativado. Caso contrário, o parâmetro será ignorado e tratado como `all`.

Práticas recomendadas

- Planeje a capacidade de destino suficiente em todas as zonas de disponibilidade que você espera utilizar, por grupo de destino. Se você não conseguir planejar a capacidade suficiente em todas as zonas de disponibilidade participantes, recomendamos que você mantenha o balanceamento de carga entre zonas ativado.
- Ao configurar seu Application Load Balancer com vários grupos de destino, certifique-se de que todos os grupos de destino estejam participando das mesmas zonas de disponibilidade na região configurada. Isso evita que uma zona de disponibilidade fique vazia enquanto o balanceamento de carga entre zonas estiver desativado, pois acionará um Erro 503 para todas as solicitações HTTP que entrarem na zona de disponibilidade vazia.
- Evite criar sub-redes vazias. Os Application Load Balancers expõem endereços IP de zona por meio do DNS para as sub-redes vazias, o que acionará Erros 503 para solicitações HTTP.
- Pode haver ocorrências nas quais um grupo de destino com o balanceamento de carga entre zonas desativado tenha capacidade de destino suficiente por zona de disponibilidade, mas todos

os destinos em uma zona de disponibilidade não estejam íntegros. Quando houver pelo menos um grupo de destino com todos os destinos não íntegros, os endereços IP dos nós do balanceador de carga serão removidos do DNS. Depois que o grupo de destino tiver pelo menos um destino íntegro, os endereços IP serão restaurados para o DNS.

Desativar o balanceamento de carga entre zonas

Você pode desativar o balanceamento de carga entre zonas para seus grupos de destino do Application Load Balancer a qualquer momento.

Para desativar o balanceamento de carga entre zonas usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Balanceamento de carga, escolha Grupos de destino.
3. Selecione o nome do grupo de destino para abrir a página de detalhes dele.
4. Na guia Atributos, selecione Editar.
5. Na página Editar atributos do grupo de destino, selecione Desativado para Balanceamento de carga entre zonas.
6. Escolha Salvar alterações.

Para desativar o balanceamento de carga entre zonas usando o AWS CLI

Use o modify-target-group-attributescomando e defina o `load_balancing.cross_zone.enabled` atributo como`false`.

```
aws elbv2 modify-target-group-attributes --target-group-arn my-targetgroup-arn --
attributes Key=load_balancing.cross_zone.enabled,Value=false
```

Esta é uma resposta de exemplo:

```
{
    "Attributes": [
        {
            "Key": "load_balancing.cross_zone.enabled",
            "Value": "false"
        },
    ]
}
```

Ativar o balanceamento de carga entre zonas

Você pode ativar o balanceamento de carga entre zonas para seus grupos de destino do Application Load Balancer a qualquer momento. A configuração de balanceamento de carga entre zonas por grupo de destino substitui a configuração por balanceador de carga.

Para ativar o balanceamento de carga entre zonas usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Balanceamento de carga, escolha Grupos de destino.
3. Selecione o nome do grupo de destino para abrir a página de detalhes dele.
4. Na guia Atributos, selecione Editar.
5. Na página Editar atributos do grupo de destino, selecione Ativado para Balanceamento de carga entre zonas.
6. Escolha Salvar alterações.

Para ativar o balanceamento de carga entre zonas usando o AWS CLI

Use o modify-target-group-attributescomando e defina o `load_balancing.cross_zone.enabled` atributo como`true`.

```
aws elbv2 modify-target-group-attributes --target-group-arn my-targetgroup-arn --
attributes Key=load_balancing.cross_zone.enabled,Value=true
```

Esta é uma resposta de exemplo:

```
{
    "Attributes": [
        {
            "Key": "load_balancing.cross_zone.enabled",
            "Value": "true"
        },
    ]
}
```

Ponderações de destinos automáticos (ATW)

As ponderações de destinos automáticos (ATW) monitoram constantemente os destinos que executam suas aplicações, detectando desvios significativos de desempenho, conhecidos como

anomalias. As ATW fornecem a capacidade de ajustar dinamicamente a quantidade de tráfego roteado para os destinos, por meio da detecção de anomalias de dados em tempo real.

As ponderações de destinos automáticos (ATW) realizam automaticamente a detecção de anomalias em cada Application Load Balancer da conta. Quando destinos anômalos são identificados, as ATW podem tentar estabilizá-los automaticamente reduzindo a quantidade de tráfego que roteiam, o que é conhecido como mitigação de anomalias. As ATW otimizam continuamente a distribuição de tráfego para maximizar as taxas de êxito por destino e, ao mesmo tempo, minimizar as taxas de falha do grupo de destino.

Considerações:

- Atualmente, a detecção de anomalias monitora os códigos de resposta HTTP 5xx provenientes de seus destinos e as falhas de conexão com eles. A detecção de anomalias está sempre ativada e não pode ser desativada.
- As ATW não são compatíveis ao usar o Lambda como destino.

Detecção de anomalias

As ATW monitoram a detecção de anomalias de qualquer destino que esteja exibindo um desvio significativo no comportamento de outros destinos no grupo de destino. Esses desvios, chamados de anomalias, são determinados comparando os erros percentuais de um destino com os erros percentuais de outros destinos no grupo de destino. Esses erros podem ser tanto erros de conexão quanto códigos de erro HTTP. Destinos que relatam significativamente mais do que seus pares são então considerados anômalos.

A detecção de anomalias requer um mínimo de três destinos íntegros no grupo de destino. Quando um destino é registrado em um grupo de destino, ele precisa primeiro passar pelas verificações de integridade para começar a receber tráfego. Quando o destino está recebendo o destino, as ATW começam a monitorar o destino e publicam continuamente o resultado da anomalia. Em destinos sem anomalias, o resultado da anomalia é `normal`. Em destinos com anomalias, o resultado da anomalia é `anomalous`.

A detecção de anomalias das ATW funciona independentemente das verificações de integridade do grupo de destino. Um destino pode passar por todas as verificações de integridade do grupo de destino, mas ainda assim ser marcado como anômalo devido a uma taxa de erro elevada. Destinos que se tornam anômalos não afetam o status da verificação de integridade do grupo de destino.

Status de detecção de anomalia

As ATW publicam continuamente o status das detecções de anomalias que realiza nos destinos. Você pode ver o status atual a qualquer momento usando o AWS Management Console ou AWS CLI.

Para exibir o status da detecção de anomalias usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na página de detalhes dos grupos de destino, escolha a guia Destinos.
5. Na tabela de Destinos registrados, você pode exibir o status de anomalias de cada destino na coluna Resultado da detecção de anomalias.

   Se nenhuma anomalia for detectada, o resultado é `normal`.

   Se anomalias forem detectadas, o resultado é `anomalous`.

Para visualizar os resultados da detecção de anomalias usando o AWS CLI

Use o describe-target-healthcomando com o valor do `Include.member.N` atributo definido como`AnomalyDetection`.

Mitigação de anomalias

> **Important**
>
> A função de mitigação de anomalias das ATW só está disponível ao usar o algoritmo de roteamento aleatório ponderado.

A mitigação de anomalias das ATW direciona o tráfego para longe de destinos anômalos automaticamente, dando a eles a oportunidade de se recuperarem.

Durante a mitigação:

- As ATW ajustam periodicamente a quantidade de tráfego roteado para destinos anômalos. Atualmente, o período é a cada cinco segundos.
- As ATW reduzem a quantidade de tráfego roteado para destinos anômalos até a quantidade mínima necessária para realizar a mitigação de anomalias.
- Os destinos que não são mais detectados como anômalos terão gradualmente mais tráfego roteado para eles até atingirem a paridade com outros destinos normais no grupo de destino.

Ativar a mitigação de anomalias das ATW

É possível ativar a mitigação de anomalias a qualquer momento.

Para ativar a mitigação de anomalias usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na página de detalhes dos grupos de destino, na guia Atributos, escolha Editar.
5. Na página Editar atributos do grupo de destino, na seção Configuração de tráfego, em Algoritmo de balanceamento de carga, verifique se a opção Aleatório ponderado está selecionada.

   Observação: quando o algoritmo aleatório ponderado é selecionado inicialmente, a detecção de anomalias está ativada por padrão.
6. Em Mitigação de anomalias, verifique se a opção Ativar mitigação de anomalias está selecionada.
7. Escolha Salvar alterações.

Para ativar a mitigação de anomalias usando o AWS CLI

Use o comando modify-target-group-attributes com o atributo `load_balancing.algorithm.anomaly_mitigation`.

Status de mitigação de anomalias

Sempre que o ATW estiver realizando a mitigação em um alvo, você pode visualizar o status atual a qualquer momento usando o ou. AWS Management Console AWS CLI

Para exibir o status de mitigação de anomalias usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na página de detalhes dos grupos de destino, escolha a guia Destinos.
5. Na tabela de Destinos registrados, você pode consultar o status de mitigação de anomalias de cada destino na coluna Mitigação em vigor.

   Se a mitigação não estiver em andamento, o status é `yes`.

   Se a mitigação estiver em andamento, o status é no.

Para visualizar o status de mitigação de anomalias usando o AWS CLI

Use o describe-target-healthcomando com o valor do `Include.member.N` atributo definido como`AnomalyDetection`.

Sessões persistentes para seu Application Load Balancer

Por padrão, um Application Load Balancer roteia cada solicitação de modo independente para um destino registrado com base no algoritmo de balanceamento de carga escolhido. No entanto, você pode usar o recurso sessão persistente (também conhecido como afinidade de sessão) para habilitar o balanceador de carga a vincular a sessão de um usuário a um destino específico. Isso garante que todas as solicitações do usuário durante a sessão sejam enviadas para o mesmo destino. Esse recurso é útil para servidores que mantêm as informações de estado para fornecer uma experiência contínua aos clientes. Para usar sessões persistentes, o cliente deve ser compatível com cookies.

Os Application Load Balancers são compatíveis a cookies baseados em duração e cookies baseados em aplicação. As sessões persistentes são habilitadas por grupo de destino. Você pode usar uma combinação de persistência baseada em duração, persistência baseada em aplicação e ausência de persistência em seus grupos de destino.

O segredo para o gerenciamento de sessões persistentes é determinar por quanto tempo o balanceador de carga deve rotear consistentemente a solicitação do usuário para o mesmo destino. Se sua aplicação tiver seu próprio cookie de sessão, você poderá usar a persistência baseada em aplicação, de forma que o cookie da sessão do balanceador de carga acompanhe a duração especificada pelo cookie de sessão da aplicação. Se sua aplicação não tiver seu próprio cookie de sessão, você poderá usar a persistência baseada na duração para gerar um cookie de sessão do balanceador de carga com uma duração que você especificar.

O conteúdo desses cookies gerados pelo balanceador de carga é criptografado usando uma chave alternante. Você não pode descriptografar ou modificar os cookies gerados pelo balanceador de carga.

Para os dois tipos de persistência, o Application Load Balancer redefine a expiração dos cookies que ele gera após cada solicitação. Se um cookie expirar, a sessão não continuará persistente e o cliente deverá remover o cookie de seu repositório de cookies.

Requisitos

- Um HTTP/HTTPS balanceador de carga.
- Pelo menos uma instância íntegra em cada Zona de disponibilidade.

Considerações

- Não há compatibilidade com sessões persistentes se o balanceamento de carga entre zonas estiver desabilitado. A tentativa de habilitar sessões persistentes enquanto o balanceamento de carga entre zonas estiver desabilitado falhará.
- Para cookies baseados em aplicação, os nomes dos cookies devem ser especificados individualmente para cada grupo de destino. No entanto, para cookies baseados em duração, `AWSALB` é o único nome usado em todos os grupos de destino.
- Se você estiver usando várias camadas de Application Load Balancers, poderá habilitar sessões persistentes em todas as camadas com cookies baseados em aplicação. No entanto, com cookies baseados em duração, você só poderá habilitar sessões persistentes em uma camada, porque `AWSALB` é o único nome disponível.
- Se o Application Load Balancer receber um cookie de persistência `AWSALBCORS` e `AWSALB` baseado em duração, o valor em `AWSALBCORS` terá precedência.
- A persistência baseada em aplicação não funciona com grupos de destino ponderados.

- Se você tiver uma ação de encaminhamento com vários grupos de destino e um ou mais grupos de destino tiver sessões persistentes habilitadas, você deverá habilitar a persistência por grupo de destino.
- WebSocket as conexões são inerentemente pegajosas. Se o cliente solicitar um upgrade de conexão para WebSockets, o destino que retorna um código de status HTTP 101 para aceitar o upgrade de conexão é o destino usado na WebSockets conexão. Depois que a WebSockets atualização for concluída, a aderência baseada em cookies não será usada.
- Os Application Load Balancers usam o atributo `Expires` no cabeçalho do cookie em vez do atributo `Max-Age`.
- Os Application Load Balancers não são compatíveis com valores de cookie codificados por URL.
- Se o Application Load Balancer receber uma nova solicitação enquanto o destino estiver sendo esgotado devido ao cancelamento do registro, a solicitação será roteada para um destino íntegro.

Persistência com base em duração

A persistência baseada na duração encaminha as solicitações para o mesmo destino em um grupo de destino usando um cookie gerado pelo balanceador de carga (`AWSALB`). O cookie é usado para mapear a sessão para o destino. Se sua aplicação não tiver seu próprio cookie de sessão, você poderá especificar sua própria duração de persistência e gerenciar por quanto tempo seu balanceador de carga deve rotear consistentemente a solicitação do usuário para o mesmo destino.

Quando um balanceador de carga receber uma solicitação de um cliente pela primeira vez, ele roteará a solicitação para um destino (com base no algoritmo escolhido) e gerará um cookie com o nome `AWSALB`. Ele codifica informações sobre o destino selecionado, criptografa o cookie e inclui o cookie na resposta ao cliente. O cookie gerado pelo balanceador de carga tem sua própria expiração de 7 dias, que não é configurável.

Nas solicitações subsequentes, o cliente deverá incluir o cookie `AWSALB`. Quando o balanceador de carga receber uma solicitação de um cliente contendo o cookie, ele a detectará e encaminhará a solicitação para o mesmo destino. Se o cookie estiver presente, mas não puder ser decodificado, ou se ele se referir a um destino cujo registro foi cancelado ou não está íntegro, o balanceador de carga seleciona um novo destino e atualiza o cookie com informações sobre o novo destino.

Para solicitações de compartilhamento de recursos de origem cruzada (CORS), alguns navegadores exigem `SameSite=None; Secure` para habilitar a persistência. Para oferecer suporte a esses navegadores, o balanceador de carga sempre gera um segundo cookie de persistência,

`AWSALBCORS`, que inclui as mesmas informações do cookie de persistência original, além do atributo `SameSite`. Os clientes recebem os dois cookies, incluindo solicitações não relacionadas ao CORS.

Para habilitar a persistência com base em duração usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na guia Detalhes do grupo, na seção Atributos, escolha Editar.
5. Na página Editar atributos, faça o seguinte:
   a. Selecione Persistência.
   b. Em Tipo de persistência, selecione Cookie gerado pelo balanceador de carga.
   c. Em Duração da perdurabilidade, especifique um valor entre um segundo e sete dias.
   d. Escolha Salvar alterações.

Para ativar a aderência com base na duração usando o AWS CLI

Use o modify-target-group-attributescomando com `stickiness.enabled` os `stickiness.lb_cookie.duration_seconds` atributos e.

Use o comando a seguir para habilitar a persistência com base em duração.

```
aws elbv2 modify-target-group-attributes --target-group-arn ARN --attributes
 Key=stickiness.enabled,Value=true
 Key=stickiness.lb_cookie.duration_seconds,Value=time-in-seconds
```

Sua saída deve ser similar ao exemplo a seguir.

```
    {
       "Attributes": [
          ...
           {
                "Key": "stickiness.enabled",
                "Value": "true"
             },
             {
```

```
                "Key": "stickiness.lb_cookie.duration_seconds",
                "Value": "86500"

            },
         ...
        ]
    }
```

Persistência com base em aplicação

A persistência baseada em aplicação oferece a flexibilidade de definir seus próprios critérios de persistência entre cliente e destino. Quando você ativa a persistência baseada em aplicação, o balanceador de carga encaminha a primeira solicitação para um destino dentro do grupo de destino com base no algoritmo escolhido. Espera-se que o destino defina um cookie personalizado de aplicação que corresponda ao cookie configurado no balanceador de carga para viabilizar a persistência. Esse cookie personalizado pode incluir qualquer um dos atributos de cookie exigidos pela aplicação.

Quando o Application Load Balancer receber o cookie personalizado de aplicação do destino, ele gerará automaticamente um novo cookie criptografado de aplicação para capturar informações de persistência. Esse cookie de aplicação gerado pelo balanceador de carga captura informações de persistência para cada grupo de destino que esteja com a persistência baseada em aplicações habilitada.

O cookie de aplicação gerado pelo balanceador de carga não copia os atributos do cookie personalizado definido pelo destino. Ele tem seu próprio prazo de validade de 7 dias, que não é configurável. Na resposta ao cliente, o Application Load Balancer valida somente o nome com o qual o cookie personalizado foi configurado no grupo de destino e não o valor ou o atributo de expiração do cookie personalizado. Desde que o nome corresponda, o balanceador de carga enviará os dois cookies, o cookie personalizado definido pelo destino e o cookie de aplicação gerado pelo balanceador de carga, em resposta ao cliente.

Nas solicitações subsequentes, os clientes precisarão devolver os dois cookies para manter a persistência. O balanceador de carga descriptografa o cookie de aplicação e verifica se a duração configurada da persistência ainda é válida. Em seguida, ele usa as informações do cookie para enviar a solicitação para o mesmo destino dentro do grupo de destino a fim de manter a persistência. O balanceador de carga também transfere o cookie personalizado de aplicação para o destino sem inspecioná-lo ou modificá-lo. Nas respostas subsequentes, a expiração do cookie de aplicação gerado pelo balanceador de carga e a duração da persistência configurada no balanceador de carga

serão redefinidas. Para manter a persistência entre o cliente e o destino, a expiração do cookie e a duração da persistência não devem expirar.

Se um destino falhar ou deixar de ser íntegro, o balanceador de carga interromperá as solicitações de roteamento para esse destino e escolherá um novo destino íntegro com base no algoritmo de balanceamento de carga escolhido. O balanceador de carga trata a sessão como “aderida” ao novo destino íntegro e continua a rotear solicitações para o novo destino íntegro, mesmo que o destino com falha retorne.

Com solicitações de compartilhamento de recursos de origem cruzada (CORS), para habilitar a persistência, o balanceador de carga só adiciona os atributos `SameSite=None; Secure` ao cookie de aplicação gerado pelo balanceador de carga se a versão de agente do usuário for Chromium80 ou superior.

Como a maioria dos navegadores limita os cookies a 4 K, o balanceador de carga fragmenta cookies de aplicação com tamanho superior a 4 K em vários cookies. Os Application Load Balancers são compatíveis com cookies de até 16 K e, portanto, podem criar até 4 fragmentos que enviam ao cliente. O nome do cookie do aplicativo que o cliente vê começa com “AWSALBAPP-” e inclui um número de fragmento. Por exemplo, se o tamanho do cookie for de 0 a 4K, o cliente verá AWSALBAPP -0. Se o tamanho do cookie for de 4 a 8k, o cliente verá AWSALBAPP -0 e AWSALBAPP -1 e assim por diante.

Para habilitar a persistência baseada em aplicação usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na guia Detalhes do grupo, na seção Atributos, escolha Editar.
5. Na página Editar atributos, faça o seguinte:

    a. Selecione Persistência.
    b. Em Tipo de persistência, selecione Cookie baseado em aplicação.
    c. Em Duração da perdurabilidade, especifique um valor entre um segundo e sete dias.
    d. Em Nome do cookie do aplicativo, insira um nome para o cookie baseado em aplicação.

        Não use `AWSALB`, `AWSALBAPP` ou `AWSALBTG` no nome do cookie. Eles estão reservados para uso pelo balanceador de carga.

e. Escolha Salvar alterações.

Para ativar a aderência baseada em aplicativos usando o AWS CLI

Use o modify-target-group-attributescomando com os seguintes atributos:

- `stickiness.enabled`
- `stickiness.type`
- `stickiness.app_cookie.cookie_name`
- `stickiness.app_cookie.duration_seconds`

Use o comando a seguir para habilitar a persistência com base em aplicação.

```
aws elbv2 modify-target-group-attributes --target-group-arn ARN --attributes
 Key=stickiness.enabled,Value=true Key=stickiness.type,Value=app_cookie
 Key=stickiness.app_cookie.cookie_name,Value=my-cookie-name
 Key=stickiness.app_cookie.duration_seconds,Value=time-in-seconds
```

Sua saída deve ser similar ao exemplo a seguir.

```
    {
        "Attributes": [
            ...
            {
                "Key": "stickiness.enabled",
                "Value": "true"
            },
            {
                "Key": "stickiness.app_cookie.cookie_name",
                "Value": "MyCookie"
            },
            {
                "Key": "stickiness.type",
                "Value": "app_cookie"
            },
            {
                "Key": "stickiness.app_cookie.duration_seconds",
                "Value": "86500"
            },
```

```
            ...
        ]
    }
```

Rebalanceamento manual

Ao aumentar a escala verticalmente, se o número de destinos aumentar consideravelmente, há potencial para uma distribuição desigual da carga devido à persistência. Nesse cenário, você poderá reequilibrar a carga em seus destinos usando as duas opções a seguir:

- Defina uma expiração no cookie gerado pela aplicação que seja anterior à data e hora atuais. Isso impedirá que os clientes enviem o cookie para o Application Load Balancer, que reiniciará o processo de estabelecimento da persistência.
- Defina uma duração muito curta na configuração de persistência baseada em aplicação do balanceador de carga, por exemplo, 1 segundo. Isso forçará o Application Load Balancer a restabelecer a persistência mesmo que o cookie definido pelo destino não tenha expirado.

Registre destinos com o grupo de destino do Application Load Balancer

Você registra seus destinos com um grupo de destino. Quando você cria um grupo de destino, você especifica o tipo de destino, que determina como você registra seus destinos. Por exemplo, você pode registrar instâncias IDs, endereços IP ou funções Lambda. Para obter mais informações, consulte Grupos de destino para seus Application Load Balancers.

Se a demanda em seus destinos atualmente registrados aumentar, você pode registrar destinos adicionais para lidar com a demanda. Quando seu destino estiver pronto para lidar com solicitações, registre-o com seu grupo de destino. O load balancer inicia as solicitações de roteamento ao destino assim que o processo de registro for concluído e o destino passar nas verificações de integridade iniciais.

Se a demanda em seus destinos registrados diminuir, ou se você precisar fazer manutenção em um destino, poderá cancelar o registro do seu grupo de destino. O load balancer interrompe as solicitações de roteamento para um destino assim que você cancela o registro dele. Quando o destino estiver pronto para receber as solicitações, você poderá registrá-lo com o grupo de destino novamente.

Quando você cancelar o registro de um destino, o load balancer esperará até que as solicitações em andamento sejam concluídas. Isso é conhecido como drenagem de conexão. O status de um destino é `draining` enquanto a drenagem de conexão estiver em andamento.

Quando você cancelar o registro de um destino registrado por endereço IP, deverá aguardar que o atraso de cancelamento de registro seja concluído para registrar o mesmo endereço IP novamente.

Se você estiver registrando destinos por ID de instância, poderá usar o balanceador de carga com um grupo do Auto Scaling. Após anexar um grupo de destino a um grupo do Auto Scaling e o grupo aumentar a escala horizontalmente, as instâncias iniciadas pelo grupo do Auto Scaling serão registradas automaticamente no grupo de destino. Se você desanexar o grupo de destino do grupo do Auto Scaling, as instâncias terão o registro automaticamente cancelado do grupo de destino. Para obter mais informações, consulte Como anexar um balanceador de carga ao seu grupo de Auto Scaling no Guia do usuário do Amazon Auto EC2 Scaling.

Ao encerrar um aplicativo em um destino, você deve primeiro cancelar o registro do alvo de seu grupo-alvo e dar tempo para que as conexões existentes sejam drenadas. Você pode monitorar o status do cancelamento de registro usando o comando describe-target-health CLI ou atualizando a visualização do grupo-alvo no. AWS Management Console Depois de confirmar que o registro do alvo foi cancelado, você pode continuar interrompendo ou encerrando o aplicativo. Essa sequência evita que os usuários experimentem erros 5XX quando os aplicativos são encerrados enquanto ainda processam o tráfego.

Grupos de segurança de destino

Ao registrar EC2 instâncias como destinos, você deve garantir que os grupos de segurança de suas instâncias permitam que o balanceador de carga se comunique com suas instâncias na porta do ouvinte e na porta de verificação de integridade.

Regras recomendadas

Inbound		
Source	Port Range	Comment
load balancer security group	*instance listener*	Permitir tráfego do load balancer na porta do ouvinte da instância

load balancer security group	*health check*	Permitir tráfego do load balancer na porta de verificação de integridade

Recomendamos também que você permita a entrada de tráfego ICMP para oferecer suporte ao Path MTU Discovery. Para obter mais informações, consulte Path MTU Discovery no Guia do EC2 usuário da Amazon.

Sub-redes compartilhadas

Os participantes podem criar um Application Load Balancer em uma VPC compartilhada.
Os participantes não podem registrar um destino executado em uma sub-rede que não seja compartilhada com eles.

Registrar ou cancelar o registro de destinos

O tipo de destino do seu grupo de destino determina como você registra os destinos com esse grupo de destino. Para obter mais informações, consulte Target type.

Conteúdo

- Registrar ou cancelar o registro de destinos por ID de Instância
- Registrar ou cancelar o registro de destinos por endereço IP
- Registrar ou cancelar o registro de uma função do Lambda
- Como registrar ou cancelar o registro de destinos usando a AWS CLI

Registrar ou cancelar o registro de destinos por ID de Instância

> **Note**
>
> Ao registrar destinos por ID de instância para um grupo de IPv6 destino, os destinos devem ter um IPv6 endereço principal atribuído. Para saber mais, consulte os IPv6 endereços no Guia do EC2 usuário da Amazon

A instância deve estar na nuvem privada virtual (VPC) que você especificou para o grupo de destino. A instância também deve estar no estado `running` quando você registrá-la.

Para registrar ou cancelar o registro de destinos por ID de instância usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Escolha a guia Destinos.
5. Para registrar instâncias, escolha Registrar destinos. Selecione uma ou mais instâncias, insira a porta padrão da instância conforme necessário e escolha Incluir como pendente abaixo. Após terminar de adicionar instâncias, escolha Registrar destinos pendentes.

   Observações:

   - As instâncias devem ter um IPv6 endereço principal atribuído para serem registradas em um IPv6 grupo-alvo.
   - AWS GovCloud (US) Region s não suportam a atribuição de um IPv6 endereço principal usando o console. Você deve usar a API para atribuir IPv6 endereços principais em AWS GovCloud (US) Region s.
6. Para cancelar o registro de instâncias, selecione as instâncias e escolha Cancelar registro.

Registrar ou cancelar o registro de destinos por endereço IP

IPv4 alvos

Os endereços IP que você registra devem ser de um dos seguintes blocos CIDR:

- As sub-redes da VPC para o grupo de destino
- 10.0.0.0/8 (RFC 1918)
- 100.64.0.0/10 (RFC 6598)
- 172.16.0.0/12 (RFC 1918)
- 192.168.0.0/16 (RFC 1918)

Você não pode registrar os endereços IP de outro Application Load Balancer na mesma VPC. Se o outro Application Load Balancer estiver em uma VPC emparelhada à VPC do balanceador de carga, você poderá registrar seus endereços IP.

IPv6 alvos

- Os endereços IP que você registra devem estar dentro do bloco CIDR da VPC ou dentro de um bloco CIDR da VPC emparelhado.

Para registrar ou cancelar o registro de destinos por endereço IP usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Escolha a guia Destinos.
5. Para registrar endereços IP, escolha Registrar destinos. Para cada endereço IP, selecione a rede, insira o endereço IP e a porta e, em seguida, escolha Incluir como pendente abaixo.
6. Opcional: se o endereço IP estiver fora da VPC selecionada, você deverá especificar uma zona de disponibilidade.
7. Quando você concluir a especificação de endereços, escolha Registrar destinos pendentes.
8. Para cancelar o registro de endereços IP, selecione os endereços IP e escolha Cancelar registro. Se você tiver vários endereços IP registrados, poderá ser útil para adicionar um filtro ou alterar a ordem de classificação.

Registrar ou cancelar o registro de uma função do Lambda

Você só pode registrar uma função do Lambda com cada grupo de destino. O Elastic Load Balancing precisa ter permissão para invocar cada função do Lambda. Se não precisar mais enviar tráfego para sua função Lambda, você poderá cancelar o registro. Depois de cancelar o registro de uma função Lambda, as solicitações em andamento falham com erros HTTP 5XX. Para substituir uma função Lambda, é melhor criar um grupo de destino. Para obter mais informações, consulte Usar funções do Lambda como destino de um Application Load Balancer.

Para registrar ou cancelar o registro de funções do Lambda usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.

4. Escolha a guia Destinos.
5. Se não houver qualquer função Lambda registrada, escolha Register (Registrar). Selecione a função Lambda e escolha Register (Registrar).
6. Para cancelar o registro de uma função Lambda, escolha Deregister (Cancelar registro). Quando a confirmação for solicitada, escolha Cancelar registro.

Como registrar ou cancelar o registro de destinos usando a AWS CLI

Use o comando register-targets para adicionar destinos e o comando deregister-targets para remover destinos.

Usar funções do Lambda como destino de um Application Load Balancer

Você pode registrar suas funções Lambda como destinos e configurar uma regra de listener para encaminhar solicitações ao grupo de destino para sua função Lambda. Quando o load balancer encaminha a solicitação para um grupo de destino com uma função Lambda como um destino, ele invoca sua função Lambda e transmite o conteúdo da solicitação para a função Lambda, no formato JSON.

Limites

- A função do Lambda e o grupo de destino devem estar na mesma conta e na mesma região.
- O tamanho máximo do corpo da solicitação que você pode enviar para uma função Lambda é de 1 MB. Para limites de tamanho relacionados, consulte Limites de cabeçalho HTTP.
- O tamanho máximo da resposta JSON que a função Lambda pode enviar é de 1 MB.
- WebSockets não são suportados. Solicitações de atualização são rejeitadas com um código HTTP 400.
- Não há compatibilidade com zonas locais.
- Não há suporte para ponderações de destinos automáticos (ATW).

Conteúdo

- Preparar a função do Lambda
- Criar um grupo de destino para a função do Lambda

- Receber eventos do balanceador de carga
- Responder ao balanceador de carga
- Cabeçalhos de vários valores
- Habilitar verificações de integridade
- Cancelar o registro da função do Lambda

Para uma demonstração, consulte Destino do Lambda no Application Load Balancer.

Preparar a função do Lambda

As recomendações a seguir se aplicam se você estiver usando sua função do Lambda com um Application Load Balancer.

Permissões para invocar a função do Lambda

Se criar o grupo de destino e registrar a função Lambda usando o AWS Management Console, o console adicionará as permissões necessárias à sua política de função Lambda em seu nome. Caso contrário, depois de criar o grupo-alvo e registrar a função usando o AWS CLI, você deverá usar o comando add-permission para conceder permissão ao Elastic Load Balancing para invocar sua função Lambda. Recomendamos que você use as chaves de condição `aws:SourceAccount` e `aws:SourceArn` para restringir a invocação da função ao grupo de destino especificado. Para obter mais informações, consulte O problema de "confused deputy" no Guia do usuário do IAM.

```
aws lambda add-permission \
--function-name lambda-function-arn-with-alias-name \
--statement-id elb1 \
--principal elasticloadbalancing.amazonaws.com \
--action lambda:InvokeFunction \
--source-arn target-group-arn \
--source-account target-group-account-id
```

Versionamento da função do Lambda

É possível registrar uma função Lambda por grupo de destino. Para garantir que você possa alterar sua função Lambda e que o load balancer sempre invoque a versão atual da função Lambda, crie um alias de função e inclua o alias no ARN da função ao registrar a função Lambda com o load balancer. Para obter mais informações, consulte aliases de AWS Lambda funções no Guia do AWS Lambda desenvolvedor.

Tempo-limite da função

O load balancer aguarda até que sua função Lambda responda ou expire. Recomendamos que você configure o tempo-limite da função Lambda com base no tempo de execução esperado. Para obter informações sobre o valor de tempo limite padrão e como alterá-lo, consulte Configurar o tempo limite da função Lambda. Para obter informações sobre o valor máximo de tempo limite que você pode configurar, consulte AWS Lambda cotas.

Criar um grupo de destino para a função do Lambda

Crie um grupo de destino, que é usado no roteamento da solicitação. Se o conteúdo da solicitação corresponder a uma regra de listener com uma ação para encaminhá-la para esse grupo de destino, o load balancer invocará a função Lambda registrada.

Para criar um grupo de destino e registrar a função do Lambda usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Selecione Criar grupo de destino.
4. Em Selecionar um tipo de destino, escolha Função do Lambda.
5. Em Nome do grupo de destino, digite um nome para o novo grupo de destino.
6. (Opcional) Para habilitar as verificações de integridade, escolha Habilitar na seção Verificação de integridade.
7. (Opcional) Adicione uma ou mais tags, da seguinte forma:
   a. Expanda a seção Tags.
   b. Escolha Adicionar Tag.
   c. Insira a chave e o valor da etiqueta.
8. Escolha Próximo.
9. Especifique uma única função do Lambda ou omita essa etapa e especifique uma função do Lambda posteriormente.
10. Selecione Criar grupo de destino.

Para criar um grupo de destino e registrar a função do Lambda usando a AWS CLI

Use os comandos create-target-groupe register-targets.

Receber eventos do balanceador de carga

O load balancer oferece suporte a invocações do Lambda para solicitações por protocolos HTTP e HTTPS. O load balancer envia um evento no formato JSON. O load balancer adiciona os seguintes cabeçalhos a todas as solicitações: `X-Amzn-Trace-Id`, `X-Forwarded-For`, `X-Forwarded-Port` e `X-Forwarded-Proto`.

Se o cabeçalho `content-encoding` estiver presente, o balanceador de carga Base64 codifica o corpo e define `isBase64Encoded` como `true`.

Se o cabeçalho `content-encoding` não estiver presente, a codificação Base64 dependerá do tipo de conteúdo. Para os seguintes tipos, o balanceador de carga envia o corpo como está e define `isBase64Encoded` como`false`: text/*,. application/json, application/javascript, and application/xml Caso contrário, o balanceador de carga Base64 codificará o corpo e definirá `isBase64Encoded` como `true`.

O comando a seguir é um exemplo de evento.

```
{
    "requestContext": {
        "elb": {
            "targetGroupArn":
 "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-target-
group/6d0ecf831eec9f09"
        }
    },
    "httpMethod": "GET",
    "path": "/",
    "queryStringParameters": {parameters},
    "headers": {
        "accept": "text/html,application/xhtml+xml",
        "accept-language": "en-US,en;q=0.8",
        "content-type": "text/plain",
        "cookie": "cookies",
        "host": "lambda-846800462-us-east-2.elb.amazonaws.com",
        "user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6)",
        "x-amzn-trace-id": "Root=1-5bdb40ca-556d8b0c50dc66f0511bf520",
        "x-forwarded-for": "72.21.198.66",
        "x-forwarded-port": "443",
        "x-forwarded-proto": "https"
```

```
    },
    "isBase64Encoded": false,
    "body": "request_body"
}
```

Responder ao balanceador de carga

A resposta da função do Lambda deve incluir o status de codificação Base64, o código do status e os cabeçalhos. É possível omitir o corpo.

Para incluir um conteúdo binário no corpo da resposta, você deve codificar o conteúdo em Base64 e definir `isBase64Encoded` como `true`. O load balancer decodifica o conteúdo para recuperar o conteúdo binário e o envia ao cliente no corpo da resposta HTTP.

O balanceador de carga não respeita hop-by-hop cabeçalhos, como `Connection` ou. `Transfer-Encoding` É possível omitir o cabeçalho `Content-Length` porque o load balancer o calcula antes de enviar respostas aos clientes.

Veja a seguir um exemplo de resposta de uma função do Lambda com base em nodejs.

```
{
    "isBase64Encoded": false,
    "statusCode": 200,
    "statusDescription": "200 OK",
    "headers": {
        "Set-cookie": "cookies",
        "Content-Type": "application/json"
    },
    "body": "Hello from Lambda (optional)"
}
```

Para modelos de função Lambda que funcionam com Application Load Balancers, consulte application-load-balancer-serverless-app no github. Como alternativa, abra o console do Lambda, escolha Aplicações, Criar uma aplicação e selecione uma das seguintes opções no AWS Serverless Application Repository:

- Alb-lambda-Target - S3 UploadFileto
- Alb-lambda-Target- BinaryResponse
- ALB-Lambda-Target - IP WhatisMy

Cabeçalhos de vários valores

Se as solicitações de um cliente ou respostas de uma função do Lambda contiverem cabeçalhos de vários valores ou contiverem o mesmo cabeçalho várias vezes, ou parâmetros de consulta com vários valores para a mesma chave, você poderá habilitar o suporte para a sintaxe de cabeçalho de vários valores. Após habilitar cabeçalhos de vários valores, os cabeçalhos e os parâmetros de consulta trocados entre o load balancer e a função do Lambda usam matrizes em vez de strings. Se você não habilitar a sintaxe de cabeçalho de vários valores e um cabeçalho ou um parâmetro de consulta tiver vários valores, o load balancer usará o último valor recebido.

Conteúdo

- Solicitações com cabeçalhos de vários valores
- Respostas com cabeçalhos de vários valores
- Habilitar cabeçalhos de vários valores

Solicitações com cabeçalhos de vários valores

Os nomes dos campos usados para cabeçalhos e parâmetros de string de consulta diferem dependendo da ativação de cabeçalhos de vários valores para o grupo de destino.

A solicitação de exemplo a seguir tem dois parâmetros de consulta com a mesma chave:

```
http://www.example.com?&myKey=val1&myKey=val2
```

Com o formato padrão, o load balancer usa o último valor enviado pelo cliente e envia um evento que inclui parâmetros de string de consulta que usam `queryStringParameters`. Por exemplo:

```
"queryStringParameters": { "myKey": "val2"},
```

Se você ativar cabeçalhos de vários valores, o load balancer usará os dois valores de chave enviados pelo cliente e enviará um evento que inclui parâmetros de string de consulta usando `multiValueQueryStringParameters`. Por exemplo:

```
"multiValueQueryStringParameters": { "myKey": ["val1", "val2"] },
```

Da mesma forma, suponha que o cliente envie uma solicitação com dois cookies no cabeçalho:

```
"cookie": "name1=value1",
"cookie": "name2=value2",
```

Com o formato padrão, o load balancer usa o último cookie enviado pelo cliente e envia um evento que inclui cabeçalhos que usam `headers`. Por exemplo:

```
"headers": {
    "cookie": "name2=value2",
    ...
},
```

Se você ativar cabeçalhos de vários valores, o load balancer usará os dois cookies enviados pelo cliente e enviará um evento que inclui cabeçalhos que usam `multiValueHeaders`. Por exemplo:

```
"multiValueHeaders": {
    "cookie": ["name1=value1", "name2=value2"],
    ...
},
```

Se os parâmetros de consulta forem codificados em URL, o load balancer não os decodificará. Decodifique-os na função do Lambda.

Respostas com cabeçalhos de vários valores

Os nomes dos campos usados para cabeçalhos diferem dependendo da ativação de cabeçalhos de vários valores para o grupo de destino. Você deve usar `multiValueHeaders`, se tiver ativado cabeçalhos de vários valores e `headers` de outra forma.

Com o formato padrão, é possível especificar um único cookie:

```
{
  "headers": {
      "Set-cookie": "cookie-name=cookie-value;Domain=myweb.com;Secure;HttpOnly",
      "Content-Type": "application/json"
  },
}
```

Se você habilitar os cabeçalhos de vários valores, será necessário especificar vários cookies da seguinte maneira:

```
{
  "multiValueHeaders": {
      "Set-cookie": ["cookie-name=cookie-
value;Domain=myweb.com;Secure;HttpOnly","cookie-name=cookie-value;Expires=May 8,
 2019"],
      "Content-Type": ["application/json"]
  },
}
```

O balanceador de carga pode enviar os cabeçalhos para o cliente em uma ordem diferente da ordem especificada na carga de resposta do Lambda. Portanto, não conte com o retorno dos cabeçalhos em uma ordem específica.

Habilitar cabeçalhos de vários valores

Você pode habilitar ou desabilitar cabeçalhos de vários valores para um grupo de destino com o tipo de destino `lambda`.

Para habilitar cabeçalhos de vários valores usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na guia Detalhes do grupo, na seção Atributos, escolha Editar.
5. Marque ou desmarque Cabeçalhos de vários valores.
6. Escolha Salvar alterações.

Para habilitar cabeçalhos de vários valores usando o AWS CLI

Use o comando modify-target-group-attributes com o atributo `lambda.multi_value_headers.enabled`.

Habilitar verificações de integridade

Por padrão, as verificações de integridade estão desabilitadas para grupos de destino do tipo `lambda`. Você pode habilitar as verificações de integridade para implementar o failover de DNS com o Amazon Route 53. A função Lambda pode verificar a integridade de um serviço de downstream

antes de responder à solicitação de verificação de integridade. Se a resposta da função do Lambda indicar uma falha na verificação de integridade, essa falha será transmitida para o Route 53. É possível configurar o Route 53 para executar o failover para uma pilha de backup da aplicação.

Você será cobrado por verificações de integridade como por qualquer invocação da função Lambda.

A seguir, o formato do evento de verificação de integridade enviado à sua função Lambda. Para verificar se um evento é um evento de verificação de integridade, verifique o valor do campo do agente de usuário. O agente de usuário para verificações de integridade é `ELB-HealthChecker/2.0`.

```
{
    "requestContext": {
        "elb": {
            "targetGroupArn":
 "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-target-
group/6d0ecf831eec9f09"
        }
    },
    "httpMethod": "GET",
    "path": "/",
    "queryStringParameters": {},
    "headers": {
        "user-agent": "ELB-HealthChecker/2.0"
    },
    "body": "",
    "isBase64Encoded": false
}
```

Para habilitar verificações de integridade para um grupo de destino usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na guia Detalhes do grupo, na seção Configurações da verificação de integridade, escolha Editar.
5. Em Verificação de integridade, selecione Habilitar.
6. Escolha Salvar alterações.

Para habilitar verificações de saúde para um grupo-alvo usando o AWS CLI

Use o comando modify-target-group com a opção `--health-check-enabled`.

Cancelar o registro da função do Lambda

Se não precisar mais enviar tráfego para sua função Lambda, você poderá cancelar o registro. Depois de cancelar o registro de uma função Lambda, as solicitações em andamento falham com erros HTTP 5XX.

Para substituir uma função Lambda, recomendamos criar um grupo de destino, registrar a nova função com o novo grupo de destino e atualizar as regras do listener para usar o novo grupo de destino em vez do existente.

Para cancelar o registro da função do Lambda usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na guia Targets (Destinos), selecione Deregister (Cancelar registro).
5. Quando a confirmação for solicitada, escolha Cancelar registro.

Para cancelar o registro da função Lambda usando o AWS CLI

Use o comando deregister-targets.

Tags para o grupo de destino do Application Load Balancer

As tags ajudam a categorizar seus grupos de destino de diferentes formas, como por finalidade, por proprietário ou por ambiente.

Você pode adicionar várias tags a um grupo de destino. As chaves de tag devem ser exclusivas para cada grupo de destino. Se você adicionar uma tag com uma chave que já esteja associada ao grupo de destino, o valor dessa tag será atualizado.

Quando não precisar mais de uma tag, você poderá removê-la.

Restrições

- Número máximo de tags por recurso: 50
- Comprimento máximo da chave: 127 caracteres Unicode
- Comprimento máximo de valor: 255 caracteres Unicode
- As chaves e os valores de tags diferenciam maiúsculas de minúsculas. Os caracteres permitidos são letras, espaços e números representáveis em UTF-8, além dos seguintes caracteres especiais: + - = . _ : / @. Não use espaços no início nem no fim.
- Não use o `aws:` prefixo nos nomes ou valores das tags porque ele está reservado para AWS uso. Você não pode editar nem excluir nomes ou valores de tag com esse prefixo. As tags com esse prefixo não contam para as tags por limite de recurso.

Para atualizar as tags de um grupo de destino usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Escolha o nome do grupo de destino para abrir sua página de detalhes.
4. Na guia Tags, selecione Gerenciar tags e execute uma ou mais das ações a seguir:
   a. Para atualizar uma tag, insira novos valores para Chave e Valor.
   b. Para adicionar uma nova tag, escolha Adicionar tag e insira uma Chave e um Valor.
   c. Para excluir uma tag, escolha Remover ao lado da tag.
5. Ao concluir a atualização de tags, selecione Salvar alterações.

Para atualizar as tags de um grupo-alvo usando o AWS CLI

Use os comandos add-tags e remove-tags.

Excluir um grupo de destino do Application Load Balancer

Você pode excluir um grupo de destino se ele não for mencionado pelas ações de encaminhamento de nenhuma regra de receptor. A exclusão de um grupo de destino não afeta os destinos registrados no grupo de destino. Se você não precisar mais de uma EC2 instância registrada, poderá interrompê-la ou encerrá-la.

Para excluir um grupo de destino usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, em Load Balancing (Balanceamento de carga), escolha Grupos de destino.
3. Selecione o grupo de destino e escolha Actions (Ações), Delete (Excluir).
4. Quando a confirmação for solicitada, escolha Sim, excluir.

Para excluir um grupo-alvo usando o AWS CLI

Use o comando delete-target-group.

Monitorar seus Application Load Balancers

Você pode usar os recursos a seguir para monitorar seus load balancers, analisar os padrões de tráfego e solucionar problemas com seu load balancers e destinos.

CloudWatch métricas

Você pode usar CloudWatch a Amazon para recuperar estatísticas sobre pontos de dados para seus balanceadores de carga e destinos como um conjunto ordenado de dados de séries temporais, conhecido como métricas. Essas métricas podem ser usadas para verificar se o sistema está executando conforme o esperado. Para obter mais informações, consulte CloudWatch métricas para seu Application Load Balancer.

Logs de acesso

Você pode usar os logs de acesso para capturar informações detalhadas sobre as solicitações feitas ao seu balanceador de carga e armazená-las como arquivos de log no Amazon S3. Você pode usar esses logs de acesso para analisar padrões de tráfego e solucionar problemas com seus destinos. Para obter mais informações, consulte Logs de acesso para seu Application Load Balancer.

Logs de conexão

Você pode usar os logs de conexão para capturar atributos sobre as solicitações enviadas ao balanceador de carga e armazená-las como arquivos de log no Amazon S3. Você pode usar esses logs de conexão para determinar o endereço IP e a porta do cliente, as informações do certificado do cliente, os resultados da conexão e as cifras TLS que estão sendo usadas. Esses logs de conexão podem então ser usados para revisar padrões de solicitação e outras tendências. Para obter mais informações, consulte Logs de conexão para o Application Load Balancer.

Rastreamento de solicitação

Você pode usar o rastreamento de solicitações para rastrear solicitações HTTP. O load balancer adicionará um cabeçalho com um identificador de rastreamento para cada solicitação receber. Para obter mais informações, consulte Solicitar rastreamento para seu Application Load Balancer.

CloudTrail troncos

Você pode usar AWS CloudTrail para capturar informações detalhadas sobre as chamadas feitas para a API do Elastic Load Balancing e armazená-las como arquivos de log no Amazon S3. Você

pode usar esses CloudTrail registros para determinar quais chamadas foram feitas, o endereço IP de origem da chamada, quem fez a chamada, quando a chamada foi feita e assim por diante. Para obter mais informações, consulte Registrar chamadas de API para uso do Elastic Load Balancing. CloudTrail

CloudWatch métricas para seu Application Load Balancer

O Elastic Load Balancing publica pontos de dados na Amazon CloudWatch para seus balanceadores de carga e seus alvos. CloudWatchpermite que você recupere estatísticas sobre esses pontos de dados como um conjunto ordenado de dados de séries temporais, conhecido como métricas. Considere uma métrica como uma variável a ser monitorada, e os pontos de dados como os valores dessa variável ao longo do tempo. Por exemplo, você pode monitorar o número total de destinos íntegros de um load balancer ao longo de um período especificado. Cada ponto de dados tem um time stamp associado e uma unidade de medida opcional.

É possível usar métricas para verificar se o sistema está executando conforme o esperado. Por exemplo, você pode criar um CloudWatch alarme para monitorar uma métrica específica e iniciar uma ação (como enviar uma notificação para um endereço de e-mail) se a métrica estiver fora do que você considera um intervalo aceitável.

O Elastic Load Balancing reporta métricas CloudWatch somente quando as solicitações estão fluindo pelo balanceador de carga. Se houver solicitações passando pelo balanceador de carga, o Elastic Load Balancing vai medir e enviar suas métricas em intervalos de 60 segundos. Se não há solicitações passando pelo load balancer ou não há dados para uma métrica, a métrica não é reportada.

As métricas dos Application Load Balancers excluem solicitações de verificação de integridade.

Para obter mais informações, consulte o Guia CloudWatch do usuário da Amazon.

Conteúdo

- Métricas do Application Load Balancer
- Dimensões de métrica para Application Load Balancers
- Estatísticas para métricas do Application Load Balancer
- Veja CloudWatch as métricas do seu balanceador de carga

Métricas do Application Load Balancer

- balanceador de cargas
- LCUs
- Destinos
- Integridade do grupo de destino
- Funções do Lambda
- Autenticação de usuário

O namespace `AWS/ApplicationELB` inclui as métricas a seguir para load balancers.

Métrica	Descrição
`ActiveConnectionCount`	O número total de conexões TCP simultâneas ativas de clientes com o load balancer e do load balancer com destinos. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é `Sum`. Dimensões • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`
`BYoIPUtilPercentage`	A porcentagem de uso do pool de IP. Critérios de emissão de relatórios: o BYo IP está ativado no balanceador de carga. Estatísticas: a única estatística significativa é `Average`. Dimensões • `LoadBalancer` , `TargetGroup` • `LoadBalancer` , `TargetGroup` , `AvailabilityZone`

Métrica	Descrição
ClientTLSNegotiationErrorCount	O número de conexões TLS iniciadas pelo cliente que não estabeleceram uma sessão com o load balancer devido a um erro de TLS. As causas possíveis incluem uma incompatibilidade de cifras ou protocolos ou uma falha do cliente ao verificar o certificado do servidor e fechar a conexão. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer
DesyncMitigationMode_NonCompliant_Request_Count	O número de solicitações que não estão em conformidade com a RFC 7230. Critérios de relatório: há um valor diferente de zero Estatísticas: a estatística mais útil é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer

Métrica	Descrição
`DroppedInvalidHeaderRequestCount`	O número de solicitações em que o load balancer removeu cabeçalhos HTTP com campos de cabeçalho que não são válidos antes de rotear a solicitação. O load balancer removerá esses cabeçalhos somente se o `routing.http.drop_invalid_header_fields.enabled` atributo estiver definido como `true`. Critérios de relatório: há um valor diferente de zero Estatísticas: todas Dimensões • `AvailabilityZone` , `LoadBalancer`
`ForwardedInvalidHeaderRequestCount`	O número de solicitações roteadas pelo load balancer que tinha cabeçalhos HTTP com campos de cabeçalho que não são válidos. O load balancer encaminhará solicitações com esses cabeçalhos somente se o `routing.http.drop_invalid_header_fields.enabled` atributo estiver definido como `false`. Critérios de relatório: sempre relatado Estatísticas: todas Dimensões • `AvailabilityZone` , `LoadBalancer`

Métrica	Descrição
`GrpcRequestCount`	O número de solicitações gRPC processadas em e. IPv4 IPv6 Critérios de relatório: há um valor diferente de zero Estatísticas: a estatística mais útil é `Sum`. `Minimum`, `Maximum` e `Average` retornam 1. Dimensões • `LoadBalancer` , `TargetGroup` • `AvailabilityZone` , `LoadBalancer` , `TargetGroup` • `TargetGroup` • `AvailabilityZone` , `TargetGroup`
`HTTP_Fixed_Response_Count`	O número de ações de resposta fixa que foram bem-sucedidas. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é `Sum`. Dimensões • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`
`HTTP_Redirect_Count`	O número de ações de redirecionamento que foram bem-sucedidas. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é `Sum`. Dimensões • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`

Métrica	Descrição
HTTP_Redirect_Url_Limit_Exceeded_Count	O número de ações de redirecionamento que não foram concluídas porque o URL no cabeçalho de localização de resposta era maior de 8K. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer
HTTPCode_ELB_3XX_Count	O número de códigos de redirecionamento 3XX HTTP originados pelo load balancer. Essa contagem não inclui códigos de resposta gerados pelos destinos. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer

Métrica	Descrição
`HTTPCode_ELB_4XX_Count`	O número de códigos de erro do cliente 4XX HTTP originados pelo load balancer. Essa contagem não inclui códigos de resposta gerados pelos destinos. Erros de cliente são gerados quando solicitações estão malformadas ou incompletas. Essas solicitações não foram recebidas pelo destino, exceto no caso em que o load balancer retorna um código de erro HTTP 460. Essa contagem não inclui códigos de resposta gerados pelos destinos. Critérios de relatório: há um valor diferente de zero Estatísticas: a estatística mais útil é `Sum`. `Minimum`, `Maximum` e `Average` retornam 1. Dimensões • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`
`HTTPCode_ELB_5XX_Count`	O número de códigos de erro do servidor 5XX HTTP originados pelo load balancer. Essa contagem não inclui códigos de resposta gerados pelos destinos. Critérios de relatório: há um valor diferente de zero Estatísticas: a estatística mais útil é `Sum`. `Minimum`, `Maximum` e `Average` retornam 1. Dimensões • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`

Métrica	Descrição
HTTPCode_ELB_500_Count	O número de códigos de erro do HTTP 500 originados pelo load balancer. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer
HTTPCode_ELB_502_Count	O número de códigos de erro do HTTP 502 originados pelo load balancer. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer
HTTPCode_ELB_503_Count	O número de códigos de erro do HTTP 503 originados pelo load balancer. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer

Métrica	Descrição
HTTPCode_ELB_504_Count	O número de códigos de erro do HTTP 504 originados pelo load balancer. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer
IPv6ProcessedBytes	O número total de bytes processados pelo balanceador de IPv6 carga. Essa contagem está incluída em ProcessedBytes . Critérios de relatório: há um valor diferente de zero Estatísticas: a estatística mais útil é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer
IPv6RequestCount	O número de IPv6 solicitações recebidas pelo balanceador de carga. Critérios de relatório: há um valor diferente de zero Estatísticas: a estatística mais útil é Sum. Minimum, Maximum e Average retornam 1. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer

Métrica	Descrição
`LowReputationPacketsDropped`	O número de pacotes retirados de fontes maliciosas conhecidas. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é `Sum`. Dimensões • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`
`LowReputationRequestsDenied`	O número de solicitações HTTP negadas com uma resposta HTTP 403. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é `Sum`. Dimensões • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`
`NewConnectionCount`	O número total de novas conexões TCP estabelecidas de clientes com o load balancer e do load balancer com destinos. Critérios de relatório: há um valor diferente de zero Estatísticas: a estatística mais útil é `Sum`. Dimensões • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`

Métrica	Descrição
NonStickyRequestCount	O número de solicitações em que o load balancer escolheu um novo destino porque não foi possível usar um sticky session. Por exemplo, a solicitação foi a primeira solicitação de um novo cliente e nenhum cookie de perdurabilidade foi apresentado, um cookie de perdurabilidade foi apresentado, mas não especificou um destino registrado com esse grupo de destino, o cookie de perdurabilidade estava malformado ou expirado ou um erro interno impedia o load balancer de ler o cookie de perdurabilidade. Reporting criteria: a perdurabilidade está habilitada no grupo de destino. Estatísticas: a única estatística significativa é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer
ProcessedBytes	O número total de bytes processados pelo balanceador de carga em IPv4 e IPv6 (cabeçalho HTTP e carga útil HTTP). Essa contagem inclui tráfego de e para clientes e funções Lambda, tráfego em conexões Websocket e tráfego de um provedor de identidade (IdP) se a autenticação do usuário estiver ativada. Critérios de relatório: há um valor diferente de zero Estatísticas: a estatística mais útil é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer

Métrica	Descrição
`RejectedConnectionCount`	O número de conexões que foram rejeitadas porque o load balancer atingiu o número máximo de conexões. Critérios de relatório: há um valor diferente de zero Estatísticas: a estatística mais útil é `Sum`. Dimensões • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`
`RequestCount`	O número de solicitações processadas IPv4 repetidamente IPv6. Essa métrica só é incrementada para solicitações nas quais o nó do balanceador de carga tenha conseguido escolher um destino. Solicitações rejeitadas antes da escolha de um destino não são refletidas nessa métrica. Critérios de emissão de relatórios: relatados se houver destinos registrados. Estatísticas: a estatística mais útil é `Sum`. Dimensões • `LoadBalancer` • `LoadBalancer` , `AvailabilityZone` • `LoadBalancer` , `TargetGroup` • `LoadBalancer` , `AvailabilityZone` , `TargetGroup`

Métrica	Descrição
`RuleEvaluations`	O número de regras avaliadas pelo balanceador de carga durante o processamento de solicitações. A regra padrão não é contada. As 10 avaliações gratuitas de regras por solicitação estão incluídas nessa contagem. Critérios de relatório: há um valor diferente de zero Estatísticas: a estatística mais útil é `Sum`. Dimensões • `LoadBalancer`

O `AWS/ApplicationELB` namespace inclui as seguintes métricas para unidades de capacidade do balanceador de carga (LCU).

Métrica	Descrição
`ConsumedLCUs`	O número de unidades de capacidade do balanceador de carga (LCU) usadas pelo balanceador de carga. Você paga pelo número LCUs que usa por hora. Quando a reserva da LCU estiver ativa, o LCUs Consumed informará `0` se o uso está abaixo da capacidad e reservada e informará os valores acima `0` se o uso exceder a reservada. LCUs Para obter mais informações, consulte Preço do Elastic Load Balancing. Critérios de relatório: sempre relatado Estatísticas: todas Dimensões • `LoadBalancer`

Métrica	Descrição
`PeakLCUs`	O número máximo de unidades de capacidade do balanceador de carga (LCU) usadas pelo balanceador de carga em um determinado momento. Aplicável somente ao usar o LCU Reservation. Critérios de emissão de relatórios: Sempre Estatísticas: as estatísticas mais úteis são `Sum` e `Max`. Dimensões • `LoadBalancer`
`ReservedLCUs`	Uma métrica de cobrança que relata a capacidade reservada por minuto. O total reservado LCUs em qualquer período é o valor pelo LCUs qual você será cobrado. Por exemplo, se 500 LCUs forem reservados por uma hora, a métrica por minuto será LCUs 8,33. Para obter mais informações, consulte Monitore a reserva. Critérios de relatório: há um valor diferente de zero Estatísticas: todas Dimensões • `LoadBalancer`

O namespace `AWS/ApplicationELB` inclui as métricas a seguir para destinos.

Métrica	Descrição
`AnomalousHostCount`	O número de hosts detectados com anomalias. Critérios de relatório: sempre relatado Estatísticas: As únicas estatísticas significativas são `Minimum` `Maximum` e.

Métrica	Descrição
	Dimensões • `TargetGroup` , `LoadBalancer` • `TargetGroup` , `AvailabilityZone` , `LoadBalancer`
`HealthyHostCount`	O número de destinos considerados íntegros. Critérios de emissão de relatórios: relatados se houver destinos registrados. Estatísticas: as estatísticas mais úteis são `Average`, `Minimum` e `Maximum`. Dimensões • `LoadBalancer` , `TargetGroup` • `LoadBalancer` , `AvailabilityZone` , `TargetGroup`
`HTTPCode_Target_2XX_Count` , `HTTPCode_Target_3XX_Count` , `HTTPCode_Target_4XX_Count` , `HTTPCode_Target_5XX_Count`	O número de códigos de resposta HTTP gerados pelos destinos. Isso não inclui códigos de resposta gerados pelo load balancer. Critérios de emissão de relatórios: relatados se houver destinos registrados. Estatísticas: a estatística mais útil é `Sum`. `Minimum`, `Maximum` e `Average` retornam 1. Dimensões • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer` • `TargetGroup` , `LoadBalancer` • `TargetGroup` , `AvailabilityZone` , `LoadBalancer`

Métrica	Descrição
`MitigatedHostCount`	O número de destinos sob mitigação. Critérios de relatório: sempre relatado Estatísticas: as estatísticas mais úteis são `Average`, `Minimum` e `Maximum`. Dimensões • `TargetGroup` , `LoadBalancer` • `TargetGroup` , `AvailabilityZone` , `LoadBalancer`
`RequestCountPerTarget`	A contagem média de solicitações por destino, em um grupo de destino. Você deve especificar o grupo de destino usando a dimensão `TargetGroup` . Essa métrica não se aplica se o destino é uma função Lambda. Essa contagem usa o número total de solicitações recebidas pelo grupo de destino, dividido pelo número de destinos íntegros no grupo de destino. Se não houver destinos íntegros no grupo de destino, ele será dividido pelo número total de destinos registrados. Critérios de relatório: sempre relatado Estatísticas: a única estatística válida é `Sum`. Isso representa a média, e não a soma. Dimensões • `TargetGroup` • `TargetGroup` , `AvailabilityZone` • `LoadBalancer` , `TargetGroup` • `LoadBalancer` , `AvailabilityZone` , `TargetGroup`

Métrica	Descrição
TargetConnectionErrorCount	O número de conexões que não foram estabelecidas com êxito entre o load balancer e o destino. Essa métrica não se aplica se o destino é uma função Lambda. Essa métrica não é incrementada para conexões de verificação de integridade malsucedidas. Critérios de relatório: há um valor diferente de zero Estatísticas: a estatística mais útil é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer • TargetGroup , LoadBalancer • TargetGroup , AvailabilityZone , LoadBalancer
TargetResponseTime	O tempo decorrido, em segundos, após a solicitação deixar o balanceador de carga até que o destino comece a enviar os cabeçalhos de resposta. Isso equivale ao campo target_processing_time nos logs de acesso. Critérios de relatório: há um valor diferente de zero Estatística: as estatísticas mais úteis são Average e pNN.NN (percentis). Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer • TargetGroup , LoadBalancer • TargetGroup , AvailabilityZone , LoadBalancer

Métrica	Descrição
TargetTLSNegotiationErrorCount	O número de conexões TLS iniciadas pelo load balancer que não estabeleceram uma sessão com o destino. Entre as causas possíveis está uma diferença de cifras ou protocolos. Essa métrica não se aplica se o destino é uma função Lambda. Critérios de relatório: há um valor diferente de zero Estatísticas: a estatística mais útil é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer • TargetGroup , LoadBalancer • TargetGroup , AvailabilityZone , LoadBalancer
UnHealthyHostCount	O número de destinos considerados sem integridade. Quando você cancela o registro de um alvo, isso diminui, HealthyHostCount mas não aumenta. UnhealthyHostCount Critérios de emissão de relatórios: relatados se houver destinos registrados. Estatísticas: as estatísticas mais úteis são Average, Minimum e Maximum. Dimensões • LoadBalancer , TargetGroup • LoadBalancer , AvailabilityZone , TargetGroup

Métrica	Descrição
`ZonalShiftedHostCount`	O número de alvos considerados desativados devido à mudança de zona. Critérios de emissão de relatórios: relatado quando há um valor Estatísticas: a estatística mais útil é `Sum`. Dimensões • `LoadBalancer` , `TargetGroup` . • `AvailabilityZone` , `LoadBalancer` , `TargetGroup` .

O namespace `AWS/ApplicationELB` inclui as métricas a seguir para a integridade do grupo de destino. Para obter mais informações, consulte the section called “Integridade do grupo de destino”.

Métrica	Descrição
`HealthyStateDNS`	O número de zonas que atendem aos requisitos de estado íntegro do DNS. Estatísticas: a estatística mais útil é `Max`. Dimensões • `LoadBalancer` , `TargetGroup` • `AvailabilityZone` , `LoadBalancer` , `TargetGroup`
`HealthyStateRouting`	O número de zonas que atendem aos requisitos de estado íntegro do roteamento. Estatísticas: a estatística mais útil é `Max`. Dimensões • `LoadBalancer` , `TargetGroup` • `AvailabilityZone` , `LoadBalancer` , `TargetGroup`

Métrica	Descrição
UnhealthyRoutingRequestCount	O número de solicitações roteadas usando a ação de failover de roteamento (falha na abertura). Estatísticas: a estatística mais útil é Sum. Dimensões • LoadBalancer , TargetGroup • AvailabilityZone , LoadBalancer , TargetGroup
UnhealthyStateDNS	O número de zonas que não atendem aos requisitos de estado íntegro do DNS e, portanto, foram marcadas como não íntegras no DNS. Estatísticas: a estatística mais útil é Min. Dimensões • LoadBalancer , TargetGroup • AvailabilityZone , LoadBalancer , TargetGroup
UnhealthyStateRouting	O número de zonas que não atendem aos requisitos de estado íntegro do roteamento e, portanto, o balanceador de carga distribui o tráfego para todos os destinos na zona, incluindo destinos não íntegros. Estatísticas: a estatística mais útil é Min. Dimensões • LoadBalancer , TargetGroup • AvailabilityZone , LoadBalancer , TargetGroup

O namespace AWS/ApplicationELB inclui as seguintes métricas para funções Lambda que são registradas como destinos.

Métrica	Descrição
LambdaInternalError	O número de solicitações para uma função Lambda que falharam por um problema com o load balancer interno ou AWS Lambda. Para obter os códigos de motivo de erro, verifique o campo error_reason do log de acesso. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é Sum. Dimensões • TargetGroup • TargetGroup , LoadBalancer
LambdaTargetProcessedBytes	O número total de bytes processados pelo load balancer para solicitações e respostas de uma função Lambda. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é Sum. Dimensões • LoadBalancer
LambdaUserError	O número de solicitações para uma função Lambda que falhou por um problema com a função Lambda. Por exemplo, o load balancer não tinha permissão para invocar a função, o load balancer recebeu o JSON da função que está malformada ou não possui campos obrigatórios, ou o tamanho do corpo ou da resposta da solicitação excedia o tamanho máximo de 1 MB. Para obter os códigos de motivo de erro, verifique o campo error_reason do log de acesso. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é Sum.

Métrica	Descrição
	Dimensões • `TargetGroup` • `TargetGroup` , `LoadBalancer`

O namespace `AWS/ApplicationELB` inclui as seguintes métricas de autenticação do usuário.

Métrica	Descrição
`ELBAuthError`	O número de autenticações de usuário que não podiam ser concluídas como uma ação de autenticação não configurada, o load balancer não pode estabelecer uma conexão com o IdP, ou o load balancer não pode encerrar o fluxo de autenticação devido a um erro interno. Para obter os códigos de motivo de erro, verifique o campo error_reason do log de acesso. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é `Sum`. Dimensões • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`
`ELBAuthFailure`	O número de autenticações de usuário que não podiam ser concluídas porque o IdP negou ao usuário ou um código de autorizaç ão foi usado mais de uma vez. Para obter os códigos de motivo de erro, verifique o campo error_reason do log de acesso. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é `Sum`. Dimensões • `LoadBalancer`

Métrica	Descrição
	• `AvailabilityZone` , `LoadBalancer`
`ELBAuthLatency`	O tempo decorrido, em milissegundos, para consultar o IdP das informações de token de ID e de usuário. Se uma ou mais dessas operações falharem, este é o tempo da falha. Critérios de relatório: há um valor diferente de zero Estatísticas: Todas as estatísticas são significativas. Dimensões • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`
`ELBAuthRefreshTokenSuccess`	O número de vezes que o load balancer atualizou com sucesso as solicitações do usuário usando um token de atualização fornecido pelo IdP. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é `Sum`. Dimensões • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`

Métrica	Descrição
ELBAuthSuccess	O número de ações de autenticação que foram bem-sucedidas. Essa métrica é incrementada ao final de fluxo de trabalho de autenticação, após o load balancer ter recuperado as solicitações do usuário de IdP. Critérios de relatório: há um valor diferente de zero Estatísticas: a estatística mais útil é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer
ELBAuthUserClaimsSizeExceeded	O número de vezes que um IdP configurado retornou solicitações do usuário que excederam 11K bytes de tamanho. Critérios de relatório: há um valor diferente de zero Estatísticas: a única estatística significativa é Sum. Dimensões • LoadBalancer • AvailabilityZone , LoadBalancer

Dimensões de métrica para Application Load Balancers

Para filtrar as métricas do seu Application Load Balancer, use as dimensões a seguir.

Dimensão	Descrição
AvailabilityZone	Filtra os dados de métrica por zona de disponibilidade.

Dimensão	Descrição
`LoadBalancer`	Filtra os dados da métrica por load balancer. Especifique o balancead or de carga da seguinte forma: app/ load-balancer-name/123456789 0123456 (a parte final do ARN do balanceador de carga).
`TargetGroup`	Filtra os dados da métrica por grupo de destino. Especifique o grupo-alvo da seguinte forma: targetgroup/ target-group-name/123456789 0123456 (a parte final do ARN do grupo-alvo).

Estatísticas para métricas do Application Load Balancer

CloudWatch fornece estatísticas com base nos pontos de dados métricos publicados pelo Elastic Load Balancing. As estatísticas são agregações de dados de métrica ao longo de um período especificado. Quando você solicita estatísticas, o fluxo de dados apresentado é identificado pelo nome da métrica e pela dimensão. Dimensão é um par de nome-valor que identifica exclusivamente uma métrica. Por exemplo, você pode solicitar estatísticas de todas as EC2 instâncias íntegras por trás de um balanceador de carga lançado em uma zona de disponibilidade específica.

As estatísticas `Minimum` e `Maximum` refletem os valores mínimos e máximos dos pontos de dados relatados por cada um dos nós do load balancer em cada janela de amostragem. Por exemplo, suponha que haja dois nós de balanceador de carga que compõem o Application Load Balancer. Um nó tem `HealthyHostCount` com `Minimum` de 2, `Maximum` de 10 e `Average` de 6, enquanto o outro nó tem `HealthyHostCount` com `Minimum` de 1, `Maximum` de 5 e `Average` de 3. Assim, o load balancer tem `Minimum` de 1, `Maximum` de 10 e `Average` de cerca de 4.

Recomendamos monitorar `UnHealthyHostCount` diferentes de zero na estatística `Minimum` e ativar alarmes de valores diferentes de zero para mais de um ponto de dados. O uso de `Minimum` detectará quando os destinos forem considerados não íntegros por cada nó e zona de disponibilidade do seu balanceador de carga. O alarme para `Average` ou `Maximum` é útil se você quiser ser alertado sobre possíveis problemas, e recomendamos que os clientes revisem essa métrica e investiguem ocorrências diferentes de zero. A mitigação automática de falhas pode ser feita seguindo as melhores práticas de uso da verificação de integridade do balanceador de carga no Amazon EC2 Auto Scaling ou no Amazon Elastic Container Service (Amazon ECS).

A estatística `Sum` é o valor agregado entre todos os nós do load balancer. Como as métricas incluem vários relatórios por período, `Sum` só será aplicável às métricas agregadas em todos os nós do load balancer.

A estatística `SampleCount` é o número de amostras medidas. Como as métricas são obtidas com base em intervalos de amostragem e eventos, essa estatística normalmente não é útil. Por exemplo, com `HealthyHostCount`, `SampleCount` se baseia no número de amostras que cada nó do load balancer relata, não no número de hosts íntegros.

Um percentil indica a posição relativa de um valor no dataset. É possível especificar qualquer percentil usando até duas casas decimais (por exemplo, p95.45). Por exemplo, 95º percentil significa que 95% dos dados está abaixo desse valor e 5% está acima. Percentis geralmente são usados para isolar anomalias. Por exemplo, vamos supor que um aplicativo atende à maioria das solicitações de um cache em 1-2 ms, mas em 100-200 ms se o cache estiver vazio. O máximo reflete o caso mais lento, cerca de 200 ms. A média não indica a distribuição dos dados. Percentis fornecem uma visão mais significativa da performance do aplicativo. Ao usar o 99º percentil como acionador ou CloudWatch alarme do Auto Scaling, você pode ter como meta que no máximo 1% das solicitações demorem mais do que 2 ms para serem processadas.

Veja CloudWatch as métricas do seu balanceador de carga

Você pode visualizar as CloudWatch métricas dos seus balanceadores de carga usando o EC2 console da Amazon. Essas métricas são exibidas como gráficos de monitoramento. O monitoramento de gráficos mostrará pontos de dados se o load balancer estiver ativo e recebendo solicitações.

Como alternativa, você pode visualizar as métricas do seu load balancer usando o console do CloudWatch.

Para visualizar as métricas usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. Para visualizar métricas filtradas por grupo de destino, faça o seguinte:
   a. No painel de navegação, selecione Grupos de destino.
   b. Selecione o grupo de destino e, em seguida, selecione a guia Monitoramento.
   c. (Opcional) Para filtrar os resultados de acordo com o horário, selecione um período na opção Exibindo os dados de.
   d. Para obter uma visualização maior de uma única métrica, selecione seu gráfico.
3. Para visualizar métricas filtradas por load balancer, faça o seguinte:
   a. No painel de navegação, selecione Load Balancers.

b. Selecione seu load balancer e, em seguida, selecione a guia Monitoramento.

c. (Opcional) Para filtrar os resultados de acordo com o horário, selecione um período na opção Exibindo os dados de.

d. Para obter uma visualização maior de uma única métrica, selecione seu gráfico.

Para visualizar métricas usando o CloudWatch console

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Selecione o namespace ApplicationELB.
4. (Opcional) Para visualizar uma métrica em todas as dimensões, digite o nome no campo de pesquisa.
5. (Opcional) Para filtrar por dimensão, selecione uma das seguintes ações:

   - Para exibir somente as métricas relatadas para os seus load balancers, escolha Conforme as métricas do AppELB. Para visualizar uma métrica de um só balanceador de carga, digite o nome no campo de pesquisa.
   - Para exibir somente as métricas relatadas para os grupos de destino, selecione Conforme AppELB, conforme as métricas do TG. Para visualizar uma métrica para um só grupo de destino, digite o nome no campo de pesquisa.
   - Para exibir somente as métricas relatadas para os load balancers por zona de disponibilidade, selecione Conforme AppELB, conforme as métricas de AZ. Para visualizar uma métrica de um só balanceador de carga, digite o nome no campo de pesquisa. Para visualizar uma métrica de uma só zona de disponibilidade, digite o nome no campo de pesquisa.
   - Para exibir somente as métricas relatadas para os load balancers por zona de disponibilidade e grupo de destino, selecione Conforme AppELB, conforme as métricas de TG. Para visualizar uma métrica de um só balanceador de carga, digite o nome no campo de pesquisa. Para visualizar uma métrica para um só grupo de destino, digite o nome no campo de pesquisa. Para visualizar uma métrica de uma só zona de disponibilidade, digite o nome no campo de pesquisa.

Para visualizar métricas usando o AWS CLI

Use o comando list-metrics para listar as métricas disponíveis:

```
aws cloudwatch list-metrics --namespace AWS/ApplicationELB
```

Para obter as estatísticas de uma métrica usando o AWS CLI

Use o get-metric-statisticscomando a seguir para obter estatísticas para a métrica e a dimensão especificadas. CloudWatch trata cada combinação exclusiva de dimensões como uma métrica separada. Você não consegue recuperar estatísticas usando combinações de dimensões que não tenham sido especialmente publicadas. Você deve especificar as mesmas dimensões usadas ao criar as métricas.

```
aws cloudwatch get-metric-statistics --namespace AWS/ApplicationELB \
--metric-name UnHealthyHostCount --statistics Average  --period 3600 \
--dimensions Name=LoadBalancer,Value=app/my-load-balancer/50dc6c495c0c9188 \
Name=TargetGroup,Value=targetgroup/my-targets/73e2d6bc24d8a067 \
--start-time 2016-04-18T00:00:00Z --end-time 2016-04-21T00:00:00Z
```

A seguir está um exemplo de saída:

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-04-18T22:00:00Z",
            "Average": 0.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2016-04-18T04:00:00Z",
            "Average": 0.0,
            "Unit": "Count"
        },
        ...
    ],
    "Label": "UnHealthyHostCount"
}
```

Logs de acesso para seu Application Load Balancer

O Elastic Load Balancing fornece logs de acesso que capturam informações detalhadas sobre as solicitações enviadas ao seu balanceador de carga. Cada log contém informações como a hora

em que a solicitação foi recebida, o endereço IP do cliente, latências, caminhos de solicitação e respostas do servidor. Você pode usar esses logs de acesso para analisar padrões de tráfego e solucionar problemas.

O registro de logs de acesso é um recurso opcional do Elastic Load Balancing que está desabilitado por padrão. Após habilitar os logs de acesso para seu balanceador de carga, o Elastic Load Balancing capturará os logs e os armazenará como arquivos compactados no bucket do Amazon S3 que você especificar. Você pode desabilitar os logs de acesso a qualquer momento.

Você receberá cobranças pelos custos de armazenamento do Amazon S3, mas não haverá cobranças pela largura de banda usada pelo Elastic Load Balancing para enviar arquivos de log para o Amazon S3. Para obter mais informações sobre os custos de armazenamento, consulte Preços do Amazon S3.

Conteúdo

- Arquivos do log de acesso
- Entradas do log de acesso
- Exemplo de entradas de log
- Processar arquivos de log de acesso
- Habilitar os logs de acesso para seu Application Load Balancer
- Desabilite os logs de acesso para seu Application Load Balancer

Arquivos do log de acesso

O Elastic Load Balancing publica um arquivo de log para cada nó do balanceador de carga a cada 5 minutos. A entrega de logs, no final das contas, é consistente. O load balancer pode distribuir vários logs para o mesmo período. Isso normalmente acontece se o site tiver alto tráfego.

Os nomes dos arquivos dos logs de acesso usa o seguinte formato:

```
bucket[/prefix]/AWSLogs/aws-account-id/elasticloadbalancing/region/yyyy/mm/dd/aws-account-id_elasticloadbalancing_region_app.load-balancer-id_end-time_ip-address_random-string.log.gz
```

bucket

O nome do bucket do S3.

prefix

(Opcional) O prefixo (hierarquia lógica) no bucket. O prefixo especificado não pode incluir a string `AWSLogs`. Para mais informações, consulte Organizar objetos usando prefixos.

`AWSLogs`

Adicionamos a parte do nome do arquivo que começa com `AWSLogs` após o nome do bucket e o prefixo opcional que você especificar.

aws-account-id

O ID da AWS conta do proprietário.

região

A Região para seu load balancer e o bucket do S3.

aaaa/mm/dd

A data em que o log foi entregue.

load-balancer-id

O ID de recursos do load balancer. Se o ID de recursos contiver barras (/), elas são substituídos por pontos (.).

end-time

A data e a hora em que o intervalo de registro terminou. Por exemplo, a hora final de 20140215T2340Z contém entradas para solicitações feitas entre 23h35 e 23h40 no horário UTC ou Zulu.

ip-address

O endereço IP do nó do load balancer que processou a solicitação. Para um load balancer interno, esse é um endereço IP privado.

random-string

Uma string aleatória gerada pelo sistema.

Veja um exemplo de um nome de arquivo de log com um prefixo:

```
s3://amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/
elasticloadbalancing/us-east-2/2022/05/01/123456789012_elasticloadbalancing_us-
```

```
east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

Veja um exemplo de um nome de arquivo de log sem um prefixo:

```
s3://amzn-s3-demo-logging-bucket/AWSLogs/123456789012/elasticloadbalancing/
us-east-2/2022/05/01/123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

Você pode armazenar os arquivos de log no bucket pelo tempo que desejar, mas também pode definir regras do ciclo de vida do Amazon S3 para arquivar ou excluir os arquivos de log automaticamente. Para obter mais informações, consulte Gerenciamento do ciclo de vida de objetos no Guia do usuário do Amazon S3.

Entradas do log de acesso

O Elastic Load Balancing registra em log as solicitações enviadas ao balanceador de carga, inclusive aquelas que nunca chegaram aos destinos. Por exemplo, se um cliente enviar uma solicitação mal formada ou não houver destinos íntegros para responder a solicitação, a solicitação mesmo assim será registrada. O Elastic Load Balancing não registra em log solicitações de verificação de integridade.

Cada entrada de registro contém os detalhes de uma única solicitação (ou conexão, no caso de WebSockets) feita ao balanceador de carga. Para WebSockets, uma entrada é gravada somente após o fechamento da conexão. Se a conexão atualizada não puder ser estabelecida, a entrada será a mesma de uma solicitação HTTP ou HTTPS.

> **Important**
>
> O Elastic Load Balancing registra as solicitações na base do melhor esforço. Recomendamos que você use logs de acesso para compreender a natureza das solicitações, não como uma contabilidade completa de todas as solicitações.

Conteúdo

- Sintaxe
- Ações executadas

- Motivos de classificação
- Códigos de motivo de erro

Sintaxe

A tabela a seguir descreve os campos de uma entrada no log de acesso, em ordem. Todos os campos são delimitados por espaços. Quando novos campos são introduzidos, eles são adicionados no final da entrada de log. Você deve ignorar quaisquer campos no final da entrada de log que não era esperada.

Campo	Descrição
type	O tipo de solicitação ou conexão. Os valores possíveis são as seguintes (ignorar todos os outros valores): • `http` — HTTP • `https`: HTTP por TLS • `h2`: HTTP/2 por TLS • `grpcs`: gRPC por TLS • `ws` — WebSockets • `wss`— WebSockets sobre TLS
time	A hora em que o load balancer gerou uma resposta para o cliente, no formato ISO 8601. Pois WebSockets, esse é o momento em que a conexão é fechada.
elb	O ID de recursos do load balancer. Se você estiver analisando entradas de registro de acesso, observe que os recursos IDs podem conter barras (/).
client:port	O endereço IP e porta do cliente solicitante. Se houver um proxy na frente do balanceador de carga, esse campo conterá o endereço IP do proxy.
target:port	O endereço IP e porta do destino que processou essa solicitação.

Campo	Descrição
	Se o cliente não enviar uma solicitação completa, o load balancer não poderá despachar a solicitação a um destino e esse valor será definido como -. Se o destino for uma função Lambda, esse valor é definido como -. Se a solicitação for bloqueada por AWS WAF, esse valor será definido como -.
request_processing_time	O tempo total (em segundos, com precisão de milissegundos) decorrido desde o momento em que o balanceador de carga recebeu a solicitação até o momento em que ele a enviou a um destino. Esse valor será configurado como -1 se o load balancer não conseguir despachar a solicitação a um destino. Isso pode acontecer se o destino fechar a conexão antes de o tempo limite de inatividade ou se o cliente enviar uma solicitação malformada. Esse valor também pode ser definido como -1 se uma conexão TCP não puder ser estabelecida com o destino antes de atingir o tempo limite da conexão TCP de 10 segundos. Se AWS WAF estiver habilitado para seu Application Load Balancer ou o tipo de destino for uma função Lambda, o tempo necessário para o cliente enviar os dados necessários para solicitações POST será contabilizado. `request_processing_time`

Campo	Descrição
target_processing_time	O tempo total (em segundos, com precisão de milissegundos) decorrido desde o momento em que o load balancer enviou a solicitação a um destino até que o destino começar a enviar os cabeçalhos de resposta. Esse valor será configurado como -1 se o load balancer não conseguir despachar a solicitação a um destino. Isso pode acontecer se o destino fechar a conexão antes de o tempo limite de inatividade ou se o cliente enviar uma solicitação malformada. Esse valor também pode ser configurado como –1 se o destino registrad o não responder antes do tempo limite de inatividade. Se não AWS WAF estiver habilitado para seu Application Load Balancer, o tempo necessário para o cliente enviar os dados necessários para solicitações POST será contabilizado. `target_processing_time`
response_processing_time	O tempo total decorrido (em segundos, com precisão de milissegundos) desde o momento em que o load balancer recebeu o cabeçalho de resposta do destino até que ele começou a enviar a resposta ao cliente. Isso inclui o tempo de fila no load balancer e o tempo de aquisição de conexão do load balancer ao cliente. Esse valor será definido como -1 se o balanceador de carga não receber uma resposta de um destino. Isso pode acontecer se o destino fechar a conexão antes de o tempo limite de inatividade ou se o cliente enviar uma solicitação malformada.
elb_status_code	O código de status da resposta gerada pelo balanceador de carga, regra de resposta fixa ou código de resposta AWS WAF personalizado para ações de bloco.
target_status_code	O código de status da resposta do destino. Esse valor só será registrad o se tiver sido estabelecida uma conexão ao destino e o destino tiver enviado uma resposta. Caso contrário, ele será definido como -.

Campo	Descrição
received_bytes	O tamanho da solicitação, em bytes, recebida do cliente (solicitante). Para solicitações HTTP, isso inclui os cabeçalhos. Pois WebSockets, esse é o número total de bytes recebidos do cliente na conexão.
sent_bytes	O tamanho da resposta, em bytes, enviada ao cliente (solicitante). Para solicitações HTTP, isso inclui os cabeçalhos e o corpo da resposta. Pois WebSockets, esse é o número total de bytes enviados ao cliente na conexão. Os cabeçalhos TCP e a carga útil do handshake TLS não estão incluídos no. `sent_bytes` Portanto, `sent_bytes` não `DataTransfer-Out-Bytes` coincidirá AWS Cost Explorer.
"solicitação"	A linha de solicitação do cliente entre aspas duplas e registrada no seguinte formato: método HTTP + protocol://host:port/uri + versão HTTP. O load balancer preserva o URL enviado pelo cliente, da forma como se encontra, ao gravar o URI da solicitação. Ele não define o tipo de conteúdo para o arquivo do log de acesso. Ao processar esse campo, considere como o cliente enviou o URL.
"user_agent"	Uma string usuário-agente que identifica o cliente que originou a solicitação entre aspas duplas. A string consiste em um ou mais identificadores de produto, produto[/versão]. Se a string tiver mais de 8 KB, ela ficará truncada.
ssl_cipher	[Listener HTTPS] A cifra do SSL. Esse valor é definido como -, se o listener não for um listener HTTPS.
ssl_protocol	[Listener HTTPS] O protocolo SSL. Esse valor é definido como -, se o listener não for um listener HTTPS.
target_group_arn	O Nome de recurso da Amazon (ARN) do grupo de destino.
"trace_id"	O conteúdo do cabeçalho X-Amzn-Trace-Id em aspas duplas.

Campo	Descrição
"domain_name"	[Listener HTTPS] O domínio SNI fornecido pelo cliente durante o handshake do TLS em aspas duplas. Esse valor será definido como - se o cliente não oferecer suporte a SNI ou o domínio não corresponder a um certificado e o certificado padrão for apresentado ao cliente.
"chosen_cert_arn"	[Listener HTTPS] O ARN do certificado apresentado ao cliente em aspas duplas. Esse valor é configurado como `session-reused` se a sessão for reutilizada. Esse valor é definido como -, se o listener não for um listener HTTPS.
matched_rule_priority	O valor de prioridade da regra que corresponde à solicitação. Se uma regra corresponde, este é um valor de 1 a 50.000. Se nenhuma regra corresponde e a ação padrão for executada, o valor é 0. Se ocorrer um erro durante a avaliação de regras, ele é definido como -1. Para qualquer outro erro, ele é definido como -.
request_creation_time	A hora em que o load balancer recebeu a solicitação do cliente, no formato ISO 8601.
"actions_executed"	As ações executadas ao processar a solicitação em aspas duplas. Esse valor é uma lista separada por vírgulas que pode incluir os valores descritos em Ações executadas. Se nenhuma ação foi executada, como para uma solicitação malformada, esse valor será definido como -.
"redirect_url"	O URL do destino do redirecionamento para o cabeçalho de localização da resposta HTTP, entre aspas duplas. Se nenhuma ação de redirecio namento foi realizada, o valor é definido como -.
"error_reason"	O código de motivo, entre aspas duplas. Se a solicitação falhou, esse é um dos códigos de erro descritas em Códigos de motivo de erro. Se as ações realizadas não incluírem uma ação de autenticação ou o destino não for uma função do Lambda, esse valor será definido como -.

Campo	Descrição
"target:port_list"	Uma lista delimitada por espaços de endereços IP e portas para os destinos que processaram esta solicitação, entre aspas duplas. Atualmente, essa lista pode conter um item e corresponde ao campo target:port. Se o cliente não enviar uma solicitação completa, o load balancer não poderá despachar a solicitação a um destino e esse valor será definido como -. Se o destino for uma função Lambda, esse valor é definido como -. Se a solicitação for bloqueada por AWS WAF, esse valor será definido como -.
"target_status_code_list"	Uma lista delimitada por espaços de códigos de status das respostas dos destinos, entre aspas duplas. Atualmente, essa lista pode conter um item e corresponde ao campo target_status_code. Esse valor só será registrado se tiver sido estabelecida uma conexão ao destino e o destino tiver enviado uma resposta. Caso contrário, ele será definido como -.
"classification"	A classificação para mitigação de dessincronização, entre aspas duplas. Se a solicitação não estiver em conformidade com a RFC 7230, os valores possíveis são Aceitável, Ambíguo e Severo. Se a solicitação estiver em conformidade com a RFC 7230, esse valor será definido como -.
"classification_reason"	O código de motivo da classificação, entre aspas duplas. Se a solicitação não estiver em conformidade com a RFC 7230, esse é um dos códigos de classificação descritos em Motivos de classificação. Se a solicitação estiver em conformidade com a RFC 7230, esse valor será definido como -.

Campo	Descrição
conn_trace_id	O ID de rastreabilidade da conexão é um ID opaco exclusivo usado para identificar cada conexão. Depois que uma conexão for estabelecida com um cliente, as solicitações subsequentes desse cliente conterão esse ID em suas respectivas entradas de log de acesso. Esse ID atua como uma chave estrangeira para criar um link entre os logs de conexão e acesso.

Ações executadas

O load balancer armazena as ações executadas no campo actions_executed do log de acesso.

- `authenticate`: o balanceador de carga validou a sessão, autenticou o usuário e adicionou as informações do usuário aos cabeçalhos da solicitação, conforme especificado pela configuração da regra.
- `fixed-response`: o balanceador de carga emitiu uma resposta fixa, conforme especificado pela configuração da regra.
- `forward`: o balanceador de carga encaminhou a solicitação para um destino, conforme especificado pela configuração da regra.
- `redirect`: o balanceador de carga redirecionou a solicitação para outro URL, conforme especificado pela configuração da regra.
- `waf`: o balanceador de carga encaminhou a solicitação ao AWS WAF para determinar se a solicitação deve ser encaminhada para o destino. Se essa for a ação final, AWS WAF determinou que a solicitação deve ser rejeitada. Por padrão, as solicitações rejeitadas por AWS WAF serão registradas como “403" no campo. `elb_status_code` Quando AWS WAF estiver configurado para rejeitar solicitações com um Código de Resposta Personalizado, o `elb_status_code` campo refletirá o código de resposta configurado.
- `waf-failed`— O balanceador de carga tentou encaminhar a solicitação para AWS WAF, mas esse processo falhou.

Motivos de classificação

Se uma solicitação não estiver em conformidade com a RFC 7230, o balanceador de carga armazenará um dos seguintes códigos no campo classification_reason do log de acesso. Para obter mais informações, consulte Modo de mitigação de dessincronização.

Código	Descrição	Classificação
`AmbiguousUri`	O URI de solicitação contém caracteres de controle.	Ambíguo
`BadContentLength`	O cabeçalho Content-Length (Comprimento de conteúdo) contém um valor que não pode ser analisado ou não é um número válido.	Grave
`BadHeader`	Um cabeçalho contém um caractere nulo ou retorno de carro.	Grave
`BadTransferEncoding`	O cabeçalho Transfer-Encoding (Codificação de transferência) contém um valor inválido.	Grave
`BadUri`	O URI de solicitação contém um caractere nulo ou retorno de carro.	Grave
`BadMethod`	O método de solicitação está malformado.	Grave
`BadVersion`	A versão da solicitação está malformada.	Grave
`BothTeClPresent`	A solicitação contém um cabeçalho Transfer-Coding (Codificação de transferência) e um cabeçalho Content-Length (Comprimento de conteúdo).	Ambíguo
`DuplicateContentLength`	Há vários cabeçalhos Content-Length (Comprimento de conteúdo) com o mesmo valor.	Ambíguo
`EmptyHeader`	Um cabeçalho está vazio ou há uma linha com apenas espaços.	Ambíguo
`GetHeadZeroContentLength`	Há um cabeçalho Content-Length (Comprimento de conteúdo) com um valor de 0 para uma solicitação GET ou HEAD.	Aceitável

Código	Descrição	Classificação
`MultipleContentLength`	Há vários cabeçalhos Content-Length (Comprimento de conteúdo) com valores diferentes.	Grave
`MultipleTransferEncodingChunked`	Há vários cabeçalhos Transfer-Coding (Codificação de transferência): cabeçalhos em bloco.	Grave
`NonCompliantHeader`	Um cabeçalho contém um caractere não ASCII ou de controle.	Aceitável
`NonCompliantVersion`	A versão de solicitação contém um valor incorreto.	Aceitável
`SpaceInUri`	O URI de solicitação contém um espaço que não é codificado por URL.	Aceitável
`SuspiciousHeader`	Há um cabeçalho que pode ser normalizado para Transfer-Enconding (Codificação de transferência) ou Content-Length (Comprimento de conteúdo) usando técnicas comuns de normalização de texto.	Ambíguo
`SuspiciousTeClPresent`	A solicitação contém um cabeçalho Transfer-Encoding e um cabeçalho Content-Length, com pelo menos um deles sendo suspeito.	Grave
`UndefinedContentLengthSemantics`	Há um cabeçalho Content-Length definido para uma solicitação GET ou HEAD.	Ambíguo
`UndefinedTransferEncodingSemantics`	Há um cabeçalho Transfer-Encoding definido para uma solicitação GET ou HEAD.	Ambíguo

Códigos de motivo de erro

Se o load balancer não puder concluir uma ação de autenticação, ele armazenará um dos seguintes códigos de motivo no campo error_reason do log de acesso. O balanceador de carga também incrementa a métrica correspondente CloudWatch . Para obter mais informações, consulte Autenticar usuários usando um Application Load Balancer.

Código	Descrição	Métrica
`AuthInvalidCookie`	O cookie de autenticação não é válido.	`ELBAuthFailure`
`AuthInvalidGrantError`	O código de concessão de autorização do endpoint de token não é válido.	`ELBAuthFailure`
`AuthInvalidIdToken`	O token de ID não é válido.	`ELBAuthFailure`
`AuthInvalidStateParam`	O parâmetro de estado não é válido.	`ELBAuthFailure`
`AuthInvalidTokenResponse`	A resposta do endpoint de token não é válida.	`ELBAuthFailure`
`AuthInvalidUserinfoResponse`	A resposta do endpoint de informações do usuário não é válida.	`ELBAuthFailure`
`AuthMissingCodeParam`	A resposta de autenticação do endpoint de autorização não possui um parâmetro de consulta denominado "code".	`ELBAuthFailure`
`AuthMissingHostHeader`	A resposta de autenticação do endpoint de autorização não possui um campo de cabeçalho de host.	`ELBAuthError`

Código	Descrição	Métrica
`AuthMissingStateParam`	A resposta de autenticação do endpoint de autorização não possui um parâmetro de consulta denominado "state".	`ELBAuthFailure`
`AuthTokenEpRequestFailed`	Há uma resposta de erro (não 2XX) do endpoint de token.	`ELBAuthError`
`AuthTokenEpRequestTimeout`	O balanceador de carga não consegue se comunicar com o endpoint do token ou o endpoint do token não está respondendo em 5 segundos.	`ELBAuthError`
`AuthUnhandledException`	O load balancer encontrou uma exceção não gerenciada.	`ELBAuthError`
`AuthUserinfoEpRequestFailed`	Há uma resposta de erro (não 2XX) do endpoint de informações do usuário do IdP.	`ELBAuthError`
`AuthUserinfoEpRequestTimeout`	O balanceador de carga não consegue se comunicar com o endpoint de informações do usuário do IdP ou o endpoint de informaçõ es do usuário não está respondendo em 5 segundos.	`ELBAuthError`
`AuthUserinfoResponseSizeExceeded`	O tamanho das solicitações retornadas pelo IdP excedeu 11K bytes.	`ELBAuthUserClaimsSizeExceeded`

Se houver falha em uma solicitação para um grupo de destino ponderado, o load balancer armazenará um dos códigos de erro a seguir no campo error_reason do log de acesso.

Código	Descrição
`AWSALBTGCookieInvalid`	O AWSALBTG cookie, que é usado com grupos-alvo ponderados, não é válido. Por exemplo, o load balancer retorna esse erro quando os valores de cookie são codificados por URL.
`WeightedTargetGroupsUnhandledException`	O load balancer encontrou uma exceção não gerenciada.

Se uma solicitação para uma função Lambda falhar, o load balancer armazena um dos seguintes códigos de motivo no campo error_reason do log de acesso. O balanceador de carga também incrementa a métrica correspondente CloudWatch . Para obter mais informações, consulte a ação Invoke (Invocar) do Lambda.

Código	Descrição	Métrica
`LambdaAccessDenied`	O load balancer não tinha permissão para invocar a função Lambda.	`LambdaUserError`
`LambdaBadRequest`	Houve falha na invocação do Lambda porque os cabeçalhos ou o corpo da solicitação do cliente não continham somente caracteres UTF-8.	`LambdaUserError`
`LambdaConnectionError`	O balanceador de carga não pode se conectar ao Lambda.	`LambdaInternalError`
`LambdaConnectionTimeout`	A tentativa de conexão com o Lambda esgotou o tempo limite.	`LambdaInternalError`
`LambdaEC2AccessDeniedException`	A Amazon EC2 negou acesso ao Lambda durante a inicialização da função.	`LambdaUserError`

Código	Descrição	Métrica
`LambdaEC2ThrottledException`	A Amazon EC2 limitou o Lambda durante a inicialização da função.	`LambdaUserError`
`LambdaEC2UnexpectedException`	A Amazon EC2 encontrou uma exceção inesperada durante a inicialização da função.	`LambdaUserError`
`LambdaENILimitReachedException`	O Lambda não conseguiu criar uma interface de rede na VPC especificada na configuração da função do Lambda porque o limite para interfaces de rede foi excedido.	`LambdaUserError`
`LambdaInvalidResponse`	A resposta da função Lambda está malformada ou não possui campos obrigatórios.	`LambdaUserError`
`LambdaInvalidRuntimeException`	Não há compatibilidade com a versão especificada do runtime do Lambda.	`LambdaUserError`
`LambdaInvalidSecurityGroupIDException`	O ID do grupo de segurança especificado na configuração da função Lambda não é válido.	`LambdaUserError`
`LambdaInvalidSubnetIDException`	O ID de sub-rede especificado na configuração da função Lambda não é válido.	`LambdaUserError`
`LambdaInvalidZipFileException`	O Lambda não conseguiu descompactar o arquivo zip da função especificada.	`LambdaUserError`

Código	Descrição	Métrica
`LambdaKMSAccessDeniedException`	O Lambda não conseguiu descriptografar variáveis de ambiente porque o acesso à chave do KMS foi negado. Verifique as permissões do KMS da função Lambda.	`LambdaUserError`
`LambdaKMSDisabledException`	O Lambda não conseguiu descriptografar variáveis de ambiente porque a chave do KMS especificada está desabilitada. Verifique as configurações da chave do KMS da função Lambda.	`LambdaUserError`
`LambdaKMSInvalidStateException`	O Lambda não conseguiu descriptografar variáveis de ambiente porque o estado da chave do KMS não é válido. Verifique as configurações da chave do KMS da função Lambda.	`LambdaUserError`
`LambdaKMSNotFoundException`	O Lambda não conseguiu descriptografar variáveis de ambiente porque a chave do KMS não foi encontrada. Verifique as configurações da chave do KMS da função Lambda.	`LambdaUserError`
`LambdaRequestTooLarge`	O tamanho do corpo da solicitação excedeu 1 MB.	`LambdaUserError`
`LambdaResourceNotFound`	Não foi possível encontrar a função Lambda.	`LambdaUserError`
`LambdaResponseTooLarge`	O tamanho da resposta excedeu 1 MB.	`LambdaUserError`
`LambdaServiceException`	O Lambda encontrou um erro interno.	`LambdaInternalError`

Código	Descrição	Métrica
`LambdaSubnetIPAddressLimitReachedException`	O Lambda não conseguiu configurar o acesso à VPC para a função do Lambda porque há uma ou mais sub-redes sem endereços IP disponíveis.	`LambdaUserError`
`LambdaThrottling`	A função Lambda foi limitada porque houve muitas solicitações.	`LambdaUserError`
`LambdaUnhandled`	A função Lambda encontrou uma exceção não gerenciada.	`LambdaUserError`
`LambdaUnhandledException`	O load balancer encontrou uma exceção não gerenciada.	`LambdaInternalError`
`LambdaWebsocketNotSupported`	WebSockets não são compatíveis com o Lambda.	`LambdaUserError`

Se o balanceador de carga encontrar um erro ao encaminhar solicitações para AWS WAF, ele armazenará um dos seguintes códigos de erro no campo error_reason do log de acesso.

Código	Descrição
`WAFConnectionError`	O balanceador de carga não pode se conectar a. AWS WAF
`WAFConnectionTimeout`	A conexão com o AWS WAF tempo limite foi atingido.
`WAFResponseReadTimeout`	Uma solicitação para atingir o AWS WAF tempo limite.
`WAFServiceError`	AWS WAF retornou um erro 5XX.
`WAFUnhandledException`	O load balancer encontrou uma exceção não gerenciada.

Exemplo de entradas de log

A seguir estão exemplo de entradas de log. Observe que o texto de exemplo aparece em várias linhas apenas para facilitar a leitura.

Entrada HTTP de exemplo

A seguir está uma entrada no log de exemplo para um listener do HTTP (porta 80 para porta 80):

```
http 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 10.0.0.1:80 0.000 0.001 0.000 200 200 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337262-36d228ad5d99923122bbe354" "-" "-"
0 2018-07-02T22:22:48.364000Z "forward" "-" "-" "10.0.0.1:80" "200" "-" "-"
 TID_1234abcd5678ef90
```

Exemplo de entrada HTTPS

A seguir está uma entrada no log de exemplo para um listener HTTPS (porta 443 para porta 80):

```
https 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 10.0.0.1:80 0.086 0.048 0.037 200 200 0 57
"GET https://www.example.com:443/ HTTP/1.1" "curl/7.46.0" ECDHE-RSA-AES128-GCM-SHA256
 TLSv1.2
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337281-1d84f3d73c47ec4e58577259" "www.example.com" "arn:aws:acm:us-
east-2:123456789012:certificate/12345678-1234-1234-1234-123456789012"
1 2018-07-02T22:22:48.364000Z "authenticate,forward" "-" "-" "10.0.0.1:80" "200" "-"
 "-" TID_1234abcd5678ef90
```

Entrada HTTP/2 de exemplo

A seguir está um exemplo de entrada de log para um fluxo de HTTP/2.

```
h2 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
10.0.1.252:48160 10.0.0.66:9000 0.000 0.002 0.000 200 200 5 257
"GET https://10.0.2.105:773/ HTTP/2.0" "curl/7.46.0" ECDHE-RSA-AES128-GCM-SHA256
 TLSv1.2
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
```

```
"Root=1-58337327-72bd00b0343d75b906739c42" "-" "-"
1 2018-07-02T22:22:48.364000Z "redirect" "https://example.com:80/" "-" "10.0.0.66:9000"
 "200" "-" "-" TID_1234abcd5678ef90
```

Exemplo de WebSockets entrada

Veja a seguir um exemplo de entrada de registro para uma WebSockets conexão.

```
ws 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
10.0.0.140:40914 10.0.1.192:8010 0.001 0.003 0.000 101 101 218 587
"GET http://10.0.0.30:80/ HTTP/1.1" "-" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
1 2018-07-02T22:22:48.364000Z "forward" "-" "-" "10.0.1.192:8010" "101" "-" "-"
 TID_1234abcd5678ef90
```

Exemplo de WebSockets entrada segura

Veja a seguir um exemplo de entrada de registro para uma WebSockets conexão segura.

```
wss 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
10.0.0.140:44244 10.0.0.171:8010 0.000 0.001 0.000 101 101 218 786
"GET https://10.0.0.30:443/ HTTP/1.1" "-" ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2
arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
1 2018-07-02T22:22:48.364000Z "forward" "-" "-" "10.0.0.171:8010" "101" "-" "-"
 TID_1234abcd5678ef90
```

Exemplo de entradas para as funções Lambda

A seguir, há um exemplo de entrada de log para uma solicitação de função Lambda que foi bem-sucedida:

```
http 2018-11-30T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 - 0.000 0.001 0.000 200 200 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
0 2018-11-30T22:22:48.364000Z "forward" "-" "-" "-" "-" "-" "-" TID_1234abcd5678ef90
```

A seguir, há um exemplo de entrada de log para uma solicitação de função Lambda que não foi bem-sucedida:

```
http 2018-11-30T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 - 0.000 0.001 0.000 502 - 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
0 2018-11-30T22:22:48.364000Z "forward" "-" "LambdaInvalidResponse" "-" "-" "-" "-"
 TID_1234abcd5678ef90
```

Processar arquivos de log de acesso

Os arquivos de log de acesso são compactados. Se você baixar os arquivos, deverá descompactá-los para visualizar as informações.

Se houver uma grande demanda no seu site, o load balancer poderá gerar arquivos de log com gigabytes de dados. Talvez você não consiga processar uma quantidade tão grande de dados usando o line-by-line processamento. Assim, pode ter de usar ferramentas analíticas que forneçam soluções de processamento paralelo. Por exemplo, você pode usar as ferramentas analíticas a seguir para analisar e processar logs de acesso:

- O Amazon Athena é um serviço de consultas interativas que facilita a análise de dados no Amazon S3 usando SQL padrão. Para obter mais informações, consulte Como consultar logs do Application Load Balancer no Guia do usuário do Amazon Athena.
- Loggly
- Splunk
- Sumo logic

Habilitar os logs de acesso para seu Application Load Balancer

Ao habilitar os logs de acesso para seu balanceador de carga, você deve especificar o nome do bucket do S3 no qual o balanceador de carga armazenará os logs. O bucket deve ter uma política de bucket que conceda permissão para o Elastic Load Balancing gravar no bucket.

Tarefas

- Etapa 1: Crie um bucket do S3

- Etapa 2: Anexe uma política ao seu bucket do S3
- Etapa 3: Configurar logs de acesso
- Etapa 4: Verificar permissões do bucket
- Solução de problemas

Etapa 1: Crie um bucket do S3

Quando você habilitar os logs de acesso, deverá especificar um bucket do S3 para os logs de acesso. É possível usar um bucket existente ou criar um bucket especificamente para logs de acesso. O bucket deve atender aos seguintes requisitos:

Requisitos

- O bucket deve estar localizado na mesma região que o load balancer. O bucket e o balanceador de carga podem pertencer a contas diferentes.
- A única opção de criptografia compatível no lado do servidor são as chaves gerenciadas pelo Amazon S3 (SSE-S3). Para obter mais informações, consulte Chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3).

Para criar um bucket do S3 usando o console do Amazon S3

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Escolha Criar bucket.
3. Na página Criar bucket, faça o seguinte:
   a. Para Nome do bucket, insira um nome para o bucket. Esse nome deve ser exclusivo entre todos os nomes de buckets existentes no Amazon S3. Em algumas regiões, talvez haja restrições adicionais quanto a nomes de buckets. Para obter mais informações, consulte Restrições e limitações do bucket no Guia do usuário do Amazon S3.
   b. Em Região da AWS , selecione a região em que você criou seu balanceador de carga.
   c. Em Criptografia padrão, escolha Chaves gerenciadas pelo Amazon S3 (SSE-S3).
   d. Escolha Criar bucket.

Etapa 2: Anexe uma política ao seu bucket do S3

O bucket do S3 deve ter uma política de bucket que conceda permissão para que o Elastic Load Balancing grave os logs de acesso no bucket. As políticas de bucket são um conjunto de instruções JSON gravadas na linguagem de políticas de acesso para definir permissões de acesso para o seu bucket. Cada instrução inclui informações sobre uma única permissão e contém uma série de elementos.

Se estiver usando um bucket que já tem uma política anexada, você poderá adicionar a instrução para os logs de acesso do Elastic Load Balancing à política. Se você fizer isso, recomendamos que avalie o conjunto resultante de permissões para garantir que eles são apropriadas para os usuários que precisam de acesso ao bucket para logs de acesso.

Políticas de bucket disponíveis

A política de bucket que você usará depende da Região da AWS e do tipo de zona.

Regiões disponíveis a partir de agosto de 2022

Esta política concede permissões ao serviço de entrega de logs especificado. Use essa política para balanceadores de carga nas seguintes regiões:

- Ásia-Pacífico (Hyderabad)
- Ásia-Pacífico (Malásia)
- Ásia-Pacífico (Melbourne)
- Ásia-Pacífico (Taipei)
- Ásia-Pacífico (Tailândia)
- Oeste do Canadá (Calgary)
- Europa (Espanha)
- Europa (Zurique)
- Israel (Tel Aviv)
- Oriente Médio (Emirados Árabes Unidos)
- México (Central)

JSON

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/
*"
    }
  ]
}
```

Para`Resource`, insira o ARN do local para os registros de acesso, usando o formato mostrado no exemplo de política. Sempre inclua o ID da conta com o balanceador de carga no caminho do recurso do ARN do bucket do S3. Isso garante que somente os balanceadores de carga da conta especificada possam gravar registros de acesso no bucket do S3.

O ARN especificado dependerá de você planejar ou não incluir um prefixo ao habilitar os logs de acesso na etapa 3.

Exemplo de ARN do bucket do S3 com um prefixo

O nome do bucket do S3 é amzn-s3-demo-logging-bucket e o prefixo é. logging-prefix

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

Exemplo de ARN do bucket do S3 sem prefixo

O nome do bucket do S3 éamzn-s3-demo-logging-bucket. Não há parte do prefixo no ARN do bucket do S3.

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

Regiões disponíveis antes de agosto de 2022

Esta política concede permissões para o ID de conta do Elastic Load Balancing especificado. Use essa política para balanceadores de carga nas regiões listadas abaixo.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/
AWSLogs/123456789012/*"
        }
    ]
}
```

Para`Principal`, *111122223333* substitua pelo ID da conta do Elastic Load Balancing para a região do balanceador de carga:

- Leste dos EUA (N. da Virgínia): 127311923021
- Leste os EUA (Ohio): 033677994240
- Oeste dos EUA (N. da Califórnia): 027434742980
- Oeste dos EUA (Oregon): 797873946194
- África (Cidade do Cabo): 098369216593
- Ásia-Pacífico (Hong Kong): 754344448648
- Ásia-Pacífico (Jacarta) — 589379963580
- Ásia-Pacífico (Mumbai): 718504428378
- Ásia-Pacífico (Osaka): 383597477331
- Ásia-Pacífico (Seul): 600734575887
- Ásia-Pacífico (Singapura): 114774131450
- Ásia-Pacífico (Sydney): 783225319266
- Ásia-Pacífico (Tóquio): 582318560864
- Canadá (Central): 985666609251
- Europa (Frankfurt): 054676820928

- Europa (Irlanda): 156460612806
- Europa (Londres): 652711504416
- Europa (Milão): 635631232127
- Europa (Paris): 009996457667
- Europa (Estocolmo): 897822967062
- Oriente Médio (Bahrein): 076674570225
- América do Sul (São Paulo): 507241528517

Para`Resource`, insira o ARN do local para os registros de acesso, usando o formato mostrado no exemplo de política. Sempre inclua o ID da conta com o balanceador de carga no caminho do recurso do ARN do bucket do S3. Isso garante que somente os balanceadores de carga da conta especificada possam gravar registros de acesso no bucket do S3.

O ARN especificado dependerá de você planejar ou não incluir um prefixo ao habilitar os logs de acesso na etapa 3.

Exemplo de ARN do bucket do S3 com um prefixo

O nome do bucket do S3 é amzn-s3-demo-logging-bucket e o prefixo é. logging-prefix

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

Exemplo de ARN do bucket do S3 sem prefixo

O nome do bucket do S3 éamzn-s3-demo-logging-bucket. Não há parte do prefixo no ARN do bucket do S3.

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

AWS GovCloud (US) Regiões

Esta política concede permissões para o ID de conta do Elastic Load Balancing especificado. Use essa política para balanceadores de carga nas AWS GovCloud (US) regiões.

JSON

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws-us-gov:iam::111122223333:root"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws-us-gov:s3:::amzn-s3-demo-bucket/prefix/
AWSLogs/123456789012/*"
        }
    ]
}
```

Para`Principal`, *111122223333* substitua pelo ID da conta do Elastic Load Balancing para a região do balanceador de carga:

- AWS GovCloud (Oeste dos EUA) — 048591011584
- AWS GovCloud (Leste dos EUA) — 190560391635

Para`Resource`, insira o ARN do local para os registros de acesso, usando o formato mostrado no exemplo de política. Sempre inclua o ID da conta com o balanceador de carga no caminho do recurso do ARN do bucket do S3. Isso garante que somente os balanceadores de carga da conta especificada possam gravar registros de acesso no bucket do S3.

O ARN do bucket do S3 que você especifica depende se você planeja incluir um prefixo ao ativar a etapa 3 do link de registros de acesso.

Exemplo de ARN do bucket do S3 com um prefixo

O nome do bucket do S3 é amzn-s3-demo-logging-bucket e o prefixo é. logging-prefix

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

Exemplo de ARN do bucket do S3 sem prefixo

O nome do bucket do S3 éamzn-s3-demo-logging-bucket. Não há parte do prefixo no ARN do bucket do S3.

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

Zonas de Outposts

A política a seguir concede permissões ao serviço de entrega de logs especificado. Use essa política para balanceadores de carga em zonas de Outposts.

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "logdelivery.elb.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*",
    "Condition": {
        "StringEquals": {
            "s3:x-amz-acl": "bucket-owner-full-control"
        }
    }
}
```

Para`Resource`, insira o ARN do local para os registros de acesso, usando o formato mostrado no exemplo de política. Sempre inclua o ID da conta com o balanceador de carga no caminho do recurso do ARN do bucket do S3. Isso garante que somente os balanceadores de carga da conta especificada possam gravar registros de acesso no bucket do S3.

O ARN do bucket do S3 que você especifica depende se você planeja incluir um prefixo ao habilitar os registros de acesso na etapa 3.

Exemplo de ARN do bucket do S3 com um prefixo

O nome do bucket do S3 é amzn-s3-demo-logging-bucket e o prefixo é. logging-prefix

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

Exemplo de ARN do bucket do S3 sem prefixo

O nome do bucket do S3 éamzn-s3-demo-logging-bucket. Não há parte do prefixo no ARN do bucket do S3.

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

⚠ Aprimorar a segurança

Use as sugestões a seguir para aprimorar a segurança do seu bucket do S3.

Revise sua política de bucket

- Use o caminho completo do recurso, incluindo a parte do ID da conta do ARN do bucket do S3. Não use curingas (*) na parte do ID da conta do ARN do bucket do S3.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
```

- Use `aws:SourceArn` para garantir que somente balanceadores de carga da região e da conta especificadas possam usar seu bucket.

```
"Condition": {
    "ArnLike": {
        "aws:SourceArn":
 "arn:aws:elasticloadbalancing:region:123456789012:loadbalancer/*"
    }
}
```

- Use `aws:SourceOrgId` with `aws:SourceArn` para garantir que somente balanceadores de carga da organização especificada possam usar seu bucket.

```
"Condition": {
    "StringEquals": {
        "aws:SourceOrgId": "o-1234567890"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:elasticloadbalancing:*:*:loadbalancer/*"
    }
}
```

- Se você tiver uma `Deny` declaração para impedir o acesso aos diretores de serviço, exceto aqueles explicitamente permitidos, não se esqueça de adicionar `logdelivery.elasticloadbalancing.amazonaws.com` à lista de diretores de serviço permitidos. Por exemplo, se você usou a `aws:PrincipalServiceNamesList` condição, adicione `logdelivery.elasticloadbalancing.amazonaws.com` o seguinte:

```
{
```

```
  "Effect": "Deny",
  "Principal": "*",
  "Condition": {
      "StringNotEqualsIfExists": {
          "aws:PrincipalServiceNamesList": [
              "logdelivery.elasticloadbalancing.amazonaws.com",
              "service.amazonaws.com"
          ]
      }
  }
}
```

Se você usou o `NotPrincipal` elemento, adicione `logdelivery.elasticloadbalancing.amazonaws.com` o seguinte. Observe que recomendamos que você use a chave de `aws:PrincipalServiceNamesList` condição `aws:PrincipalServiceName` ou para permitir explicitamente os principais de serviço em vez de usar o `NotPrincipal` elemento. Para obter mais informações, consulte NotPrincipal.

```
{
  "Effect": "Deny",
  "NotPrincipal": {
    "Service": [
       "logdelivery.elasticloadbalancing.amazonaws.com",
       "service.amazonaws.com"
    ]
  }
},
```

Para anexar uma política de bucket para logs de acesso ao seu bucket usando o console do Amazon S3

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Selecione o nome do bucket para abrir sua página de detalhes.
3. Escolha Permissions (Permissões) e, em seguida, escolha Bucket policy (Política de bucket), Edit (Editar).
4. Crie ou atualize a política de bucket para conceder as permissões necessárias.
5. Escolha Salvar alterações.

Etapa 3: Configurar logs de acesso

Siga o procedimento a seguir para configurar logs de acesso a fim de capturar informações de solicitação e entregar arquivos de log ao bucket do S3.

Requisitos

O bucket deverá atender aos requisitos descritos na etapa 1 e você deverá anexar uma política de bucket, conforme descrito na etapa 2. Se você incluir um prefixo, ele não deverá incluir a string "AWSLogs".

Para habilitar os logs de acesso ao seu load balancer usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Load Balancers.
3. Selecione o nome do balanceador de carga para abrir sua página de detalhes.
4. Na guia Atributos, escolha Editar.
5. Em Monitoramento, ative os Logs de acesso.
6. Para URI do S3, insira o URI do S3 para seus arquivos de log. O URI especificado dependerá de você estar ou não usando um prefixo.
   - URI com um prefixo: s3:///*amzn-s3-demo-logging-bucket*`logging-prefix`
   - URI sem prefixo: s3://*amzn-s3-demo-logging-bucket*
7. Escolha Salvar alterações.

Para habilitar os registros de acesso usando o AWS CLI

Use o comando modify-load-balancer-attributes.

Gerenciar o bucket do S3 para os logs de acesso

Certifique-se de desabilitar os registros de acesso antes de excluir o bucket que você configurou para os logs de acesso. Caso contrário, se houver um novo bucket com o mesmo nome e a política de bucket necessária criada em uma Conta da AWS que não seja de sua propriedade, o Elastic Load Balancing poderá gravar os logs de acesso do seu balanceador de carga nesse novo bucket.

Etapa 4: Verificar permissões do bucket

Após o registro de acesso em logs ser habilitado para seu balanceador de carga, o Elastic Load Balancing validará o bucket do S3 e criará um arquivo de teste para garantir que a política do bucket especifique as permissões necessárias. Você pode usar o console do Amazon S3 para verificar se o arquivo de teste foi criado. O arquivo de teste não é um arquivo de log de acesso real; ele não contêm registros de exemplo.

Para verificar se um arquivo de teste foi criado no bucket usando o console do Amazon S3

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Selecione o nome do bucket que você especificou para logs de acesso.
3. Localize o arquivo de teste, `ELBAccessLogTestFile`. O local dependerá de você estar ou não usando um prefixo.
   - Localização com um prefixo:*amzn-s3-demo-logging-bucket*//*logging-prefix*/AWSLogs/*123456789012*ELBAccessLogTestFile
   - Localização sem prefixo:*amzn-s3-demo-logging-bucket*//AWSLogs/*123456789012*ELBAccessLogTestFile

Solução de problemas

Se você receber um erro de acesso negado, as possíveis causas serão:

- A política do bucket não concede ao Elastic Load Balancing permissão para gravar logs de acesso no bucket. Confira se está usando a política de bucket correta para a região. Confira se o ARN do recurso usa o mesmo nome de bucket que você especificou ao habilitar os logs de acesso. Confira se o ARN do recurso não inclui um prefixo se você não tiver especificado um prefixo ao habilitar os logs de acesso.
- O bucket usa uma opção de criptografia que não é aceita no lado do servidor. O bucket deve usar chaves gerenciadas pelo Amazon S3 (SSE-S3).

Desabilite os logs de acesso para seu Application Load Balancer

Você pode desabilitar os logs de acesso para seu load balancer a qualquer momento. Após desabilitar os log de acesso, seus logs de acesso permanecerão no seu bucket do S3 até que você

os exclua. Para obter mais informações, consulte Criação, configuração e trabalho com buckets do S3 no Guia do usuário do Amazon S3.

Para desabilitar os logs de acesso usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Load Balancers.
3. Selecione o nome do balanceador de carga para abrir sua página de detalhes.
4. Na guia Atributos, escolha Editar.
5. Em Monitoramento, desative os Logs de acesso.
6. Escolha Salvar alterações.

Para desativar os registros de acesso usando o AWS CLI

Use o comando modify-load-balancer-attributes.

Logs de conexão para o Application Load Balancer

O Elastic Load Balancing fornece logs de conexão que capturam informações detalhadas sobre as solicitações enviadas ao balanceador de carga. Cada log contém informações como o endereço IP e a porta do cliente, a porta do receptor, a cifra e o protocolo TLS usados, a latência do handshake do TLS, o status da conexão e os detalhes do certificado do cliente. Você pode usar esses logs de conexão para analisar padrões de solicitação e solucionar problemas.

Os logs de conexão são um recurso opcional do Elastic Load Balancing que está desabilitado por padrão. Após habilitar os logs de conexão para o balanceador de carga, o Elastic Load Balancing capturará os logs e os armazenará como arquivos compactados no bucket do Amazon S3 que você especificar. Você pode desabilitar os logs de conexão a qualquer momento.

Você receberá cobranças pelos custos de armazenamento do Amazon S3, mas não haverá cobranças pela largura de banda usada pelo Elastic Load Balancing para enviar arquivos de log para o Amazon S3. Para obter mais informações sobre os custos de armazenamento, consulte Preços do Amazon S3.

Conteúdo

- Arquivos de log de conexão
- Entradas de log de conexão

- Exemplo de entradas de log
- Processamento dos arquivos de log de conexão
- Habilitar os logs de conexão para o Application Load Balancer
- Desabilitar os logs de conexão para o Application Load Balancer

Arquivos de log de conexão

O Elastic Load Balancing publica um arquivo de log para cada nó do balanceador de carga a cada 5 minutos. A entrega de logs, no final das contas, é consistente. O load balancer pode distribuir vários logs para o mesmo período. Isso normalmente acontece se o site tiver alto tráfego.

Os nomes dos arquivos dos logs de conexão usam o seguinte formato:

```
bucket[/prefix]/AWSLogs/aws-account-id/elasticloadbalancing/region/yyyy/mm/dd/
conn_log_aws-account-id_elasticloadbalancing_region_app.load-balancer-id_end-time_ip-
address_random-string.log.gz
```

bucket

O nome do bucket do S3.

prefix

(Opcional) O prefixo (hierarquia lógica) no bucket. O prefixo especificado não pode incluir a string `AWSLogs`. Para mais informações, consulte Organizar objetos usando prefixos.

`AWSLogs`

Adicionamos a parte do nome do arquivo que começa com `AWSLogs` após o nome do bucket e o prefixo opcional que você especificar.

aws-account-id

O ID da AWS conta do proprietário.

região

A Região para seu load balancer e o bucket do S3.

aaaa/mm/dd

A data em que o log foi entregue.

load-balancer-id

O ID de recursos do load balancer. Se o ID de recursos contiver barras (/), elas são substituídos por pontos (.).

end-time

A data e a hora em que o intervalo de registro terminou. Por exemplo, a hora final de 20140215T2340Z contém entradas para solicitações feitas entre 23h35 e 23h40 no horário UTC ou Zulu.

ip-address

O endereço IP do nó do load balancer que processou a solicitação. Para um load balancer interno, esse é um endereço IP privado.

random-string

Uma string aleatória gerada pelo sistema.

Veja um exemplo de um nome de arquivo de log com um prefixo:

```
s3://amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/
elasticloadbalancing/us-east-2/2022/05/01/
conn_log.123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

Veja um exemplo de um nome de arquivo de log sem um prefixo:

```
s3://amzn-s3-demo-logging-bucket/AWSLogs/123456789012/elasticloadbalancing/us-
east-2/2022/05/01/conn_log.123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

Você pode armazenar os arquivos de log no bucket pelo tempo que desejar, mas também pode definir regras do ciclo de vida do Amazon S3 para arquivar ou excluir os arquivos de log automaticamente. Para obter mais informações, consulte Gerenciamento do ciclo de vida de objetos no Guia do usuário do Amazon S3.

Entradas de log de conexão

Cada tentativa de conexão tem uma entrada em um arquivo de log de conexão. A forma como as solicitações do cliente são enviadas é determinada pela conexão ser persistente ou não persistente.

As conexões não persistentes têm uma única solicitação, que cria uma única entrada no log de acesso e no log de conexão. As conexões persistentes têm várias solicitações, o que cria várias entradas no log de acesso e uma única entrada no log de conexão.

Conteúdo

- Sintaxe
- Códigos de motivo de erro

Sintaxe

A tabela a seguir descreve os campos de uma entrada no log de conexão, em ordem. Todos os campos são delimitados por espaços. Quando novos campos são introduzidos, eles são adicionados no final da entrada de log. Você deve ignorar quaisquer campos no final da entrada de log que não era esperada.

Campo	Descrição
timestamp	O horário, em formato ISO 8601, em que o balanceador de carga estabeleceu com êxito ou não conseguiu estabelecer uma conexão.
client_ip	O endereço IP do cliente solicitante.
client_port	A porta do cliente solicitante.
listener_port	A porta do receptor do balanceador de carga que está recebendo a solicitação do cliente.
tls_protocol	[Ouvinte HTTPS] O SSL/TLS protocolo usado durante apertos de mão. Esse campo está definido - para não SSL/TLS solicitações.
tls_cipher	[Ouvinte HTTPS] O SSL/TLS protocolo usado durante apertos de mão. Esse campo está definido - para não SSL/TLS solicitações.
tls_handshake_latency	[Receptor HTTPS] O tempo total em segundos, com precisão de milissegundos, decorreu ao estabelecer um handshake. Este campo está definido como - quando: • A solicitação recebida não é uma SSL/TLS solicitação.

Campo	Descrição
	• O handshake não foi estabelecido com êxito.
leaf_client_cert_subject	[Receptor HTTPS] O nome do assunto do certificado folha do cliente. Este campo está definido como - quando: • A solicitação recebida não é uma SSL/TLS solicitação. • O receptor do balanceador de carga não está configurado com o mTLS habilitado. • O servidor não consegue obter load/parse o certificado de cliente Leaf.
leaf_client_cert_validity	[Receptor HTTPS] A validade, com `not-before` e `not-after` no formato ISO 8601, do certificado folha do cliente. Este campo está definido como - quando: • A solicitação recebida não é uma SSL/TLS solicitação. • O receptor do balanceador de carga não está configurado com o mTLS habilitado. • O servidor não consegue obter load/parse o certificado de cliente Leaf.
leaf_client_cert_serial_number	[Receptor HTTPS] O número de série do certificado folha do cliente. Este campo está definido como - quando: • A solicitação recebida não é uma SSL/TLS solicitação. • O receptor do balanceador de carga não está configurado com o mTLS habilitado. • O servidor não consegue obter load/parse o certificado de cliente Leaf.
tls_verify_status	[Receptor HTTPS] O status da solicitação de conexão. Esse valor é `Success` se a conexão for estabelecida com exito. Em uma conexão malsucedida, o valor é `Failed:$error_code` .

Campo	Descrição
conn_trace_id	O ID de rastreabilidade da conexão é um ID opaco exclusivo usado para identificar cada conexão. Depois que uma conexão é estabelecida com um cliente, as solicitações subsequentes desse cliente contêm esse ID em suas respectivas entradas de registro de acesso. Esse ID atua como uma chave estrangeira para criar um link entre os logs de conexão e acesso.

Códigos de motivo de erro

Se o balanceador de carga não conseguir estabelecer uma conexão, ele armazenará um dos seguintes códigos de motivo no log de conexão.

Código	Descrição
`ClientCertMaxChainDepthExceeded`	A profundidade máxima da cadeia de certificados do cliente foi excedida
`ClientCertMaxSizeExceeded`	O tamanho máximo do certificado do cliente foi excedido
`ClientCertCrlHit`	O certificado do cliente foi revogado pela CA
`ClientCertCrlProcessingError`	Erro de processamento da CRL
`ClientCertUntrusted`	O certificado do cliente não é confiável
`ClientCertNotYetValid`	O certificado do cliente ainda não é válido

Código	Descrição
`ClientCertExpired`	O certificado do cliente está expirado
`ClientCertTypeUnsupported`	O tipo de certificado do cliente não é compatível
`ClientCertInvalid`	O certificado do cliente é inválido
`ClientCertPurposeInvalid`	A finalidade do certificado do cliente é inválida
`ClientCertRejected`	O certificado do cliente foi rejeitado pela validação do servidor personalizado
`UnmappedConnectionError`	Erro de conexão de runtime não mapeado

Exemplo de entradas de log

A seguir estão exemplos de entradas de log de conexão. Observe que o texto de exemplo aparece em várias linhas apenas para facilitar a leitura.

Veja a seguir um exemplo de entrada de registro para uma conexão bem-sucedida com um ouvinte HTTPS com o modo de verificação mútua de TLS ativado na porta 443.

```
2023-10-04T17:05:15.514108Z 203.0.113.1 36280 443 TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256
 4.036
"CN=amazondomains.com,O=endEntity,L=Seattle,ST=Washington,C=US"
 NotBefore=2023-09-21T22:43:21Z;NotAfter=2026-06-17T22:43:21Z
FEF257372D5C14D4 Success TID_3180a73013c8ca4bac2f731159d4b0fe
```

Veja a seguir um exemplo de entrada de registro de uma falha na conexão com um ouvinte HTTPS com o modo de verificação mútua de TLS ativado na porta 443.

```
2023-10-04T17:05:15.514108Z 203.0.113.1 36280 443 TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256
 -
"CN=amazondomains.com,O=endEntity,L=Seattle,ST=Washington,C=US"
 NotBefore=2023-09-21T22:43:21Z;NotAfter=2026-06-17T22:43:21Z
FEF257372D5C14D4 Failed:ClientCertUntrusted TID_1c71a68d70587445ad5127ff8b2687d7
```

Processamento dos arquivos de log de conexão

Os arquivos de log de conexão são compactados. Se você abrir os arquivos usando o console do Amazon S3, eles serão descompactados e as informações serão exibidas. Se você baixar os arquivos, deverá descompactá-los para visualizar as informações.

Se houver uma grande demanda no seu site, o load balancer poderá gerar arquivos de log com gigabytes de dados. Talvez você não consiga processar uma quantidade tão grande de dados usando o line-by-line processamento. Assim, pode ter de usar ferramentas analíticas que forneçam soluções de processamento paralelo. Por exemplo, você pode usar as seguintes ferramentas analíticas para analisar e processar logs de conexão:

- O Amazon Athena é um serviço de consultas interativas que facilita a análise de dados no Amazon S3 usando SQL padrão.
- Loggly
- Splunk
- Sumo logic

Habilitar os logs de conexão para o Application Load Balancer

Ao habilitar os logs de conexão para o balanceador de carga, você deve especificar o nome do bucket do S3 no qual o balanceador de carga armazenará os logs. O bucket deve ter uma política de bucket que conceda permissão para o Elastic Load Balancing gravar no bucket.

Tarefas

- Etapa 1: Crie um bucket do S3
- Etapa 2: Anexe uma política ao seu bucket do S3
- Etapa 3: configurar logs de conexão
- Etapa 4: Verificar permissões do bucket
- Solução de problemas

Etapa 1: Crie um bucket do S3

Quando você habilitar os logs de conexão, deverá especificar um bucket do S3 para eles. É possível usar um bucket existente ou criar um bucket especificamente para logs de conexão. O bucket deve atender aos seguintes requisitos:

Requisitos

- O bucket deve estar localizado na mesma região que o load balancer. O bucket e o balanceador de carga podem pertencer a contas diferentes.
- A única opção de criptografia compatível no lado do servidor são as chaves gerenciadas pelo Amazon S3 (SSE-S3). Para obter mais informações, consulte Chaves de criptografia gerenciadas pelo Amazon S3 (SSE-S3).

Para criar um bucket do S3 usando o console do Amazon S3

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Escolha Criar bucket.
3. Na página Criar bucket, faça o seguinte:
   a. Para Nome do bucket, insira um nome para o bucket. Esse nome deve ser exclusivo entre todos os nomes de buckets existentes no Amazon S3. Em algumas regiões, talvez haja restrições adicionais quanto a nomes de buckets. Para obter mais informações, consulte Restrições e limitações do bucket no Guia do usuário do Amazon S3.
   b. Em Região da AWS , selecione a região em que você criou seu balanceador de carga.
   c. Em Criptografia padrão, escolha Chaves gerenciadas pelo Amazon S3 (SSE-S3).
   d. Escolha Criar bucket.

Etapa 2: Anexe uma política ao seu bucket do S3

O bucket do S3 deve ter uma política de bucket que conceda permissão para que o Elastic Load Balancing grave os logs de conexão no bucket. As políticas de bucket são um conjunto de instruções JSON gravadas na linguagem de políticas de acesso para definir permissões de acesso para o seu bucket. Cada instrução inclui informações sobre uma única permissão e contém uma série de elementos.

Se estiver usando um bucket existente que já tenha uma política anexada, você poderá adicionar a instrução para os logs de conexão do Elastic Load Balancing à política. Se você fizer isso, recomendamos que avalie o conjunto resultante de permissões para garantir que sejam apropriadas para os usuários que precisam de acesso ao bucket para logs de conexão.

Políticas de bucket disponíveis

A política de bucket que você usará depende da Região da AWS e do tipo de zona.

⚠ Aumente a segurança usando um bucket ARNs S3 preciso.

- Use o caminho completo do recurso, não apenas o ARN do bucket do S3.
- Inclua a parte do ID da conta do ARN do bucket do S3.
- Não use curingas (*) na parte do ID da conta do ARN do bucket do S3.

Regiões disponíveis a partir de agosto de 2022

Esta política concede permissões ao serviço de entrega de logs especificado. Use essa política para balanceadores de carga nas seguintes regiões:

- Ásia-Pacífico (Hyderabad)
- Ásia-Pacífico (Malásia)
- Ásia-Pacífico (Melbourne)
- Ásia-Pacífico (Tailândia)
- Oeste do Canadá (Calgary)
- Europa (Espanha)
- Europa (Zurique)
- Israel (Tel Aviv)
- Oriente Médio (Emirados Árabes Unidos)
- México (Central)

JSON

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/
*"
    }
  ]
}
```

Para`Resource`, insira o ARN do local para os registros de acesso, usando o formato mostrado no exemplo de política. Sempre inclua o ID da conta com o balanceador de carga no caminho do recurso do ARN do bucket do S3. Isso garante que somente os balanceadores de carga da conta especificada possam gravar registros de acesso no bucket do S3.

O ARN do bucket do S3 que você especifica depende se você planeja incluir um prefixo ao habilitar os registros de acesso na etapa 3.

Exemplo de ARN do bucket do S3 com um prefixo

O nome do bucket do S3 é amzn-s3-demo-logging-bucket e o prefixo é. logging-prefix

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

Exemplo de ARN do bucket do S3 sem prefixo

O nome do bucket do S3 éamzn-s3-demo-logging-bucket. Não há parte do prefixo no ARN do bucket do S3.

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

Uso de NotPrincipal quando Effect for Deny

Se a política de bucket do Amazon S3 usar `Effect` com o valor `Deny` e incluir `NotPrincipal` conforme mostrado no exemplo a seguir, certifique-se de que `logdelivery.elasticloadbalancing.amazonaws.com` esteja incluído na lista de `Service`.

```
{
  "Effect": "Deny",
  "NotPrincipal": {
    "Service": [
       "logdelivery.elasticloadbalancing.amazonaws.com",
       "example.com"
    ]
  }
},
```

Regiões disponíveis antes de agosto de 2022

Essa política concede permissões para a conta especificada do Elastic Load Balancing. Use essa política para balanceadores de carga nas regiões listadas abaixo.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/
AWSLogs/123456789012/*"
        }
    ]
}
```

Para`Principal`, *111122223333* substitua pelo ID da conta do Elastic Load Balancing para a região do balanceador de carga:

- Leste dos EUA (N. da Virgínia): 127311923021
- Leste os EUA (Ohio): 033677994240
- Oeste dos EUA (N. da Califórnia): 027434742980
- Oeste dos EUA (Oregon): 797873946194

- África (Cidade do Cabo): 098369216593
- Ásia-Pacífico (Hong Kong): 754344448648
- Ásia-Pacífico (Jacarta) — 589379963580
- Ásia-Pacífico (Mumbai): 718504428378
- Ásia-Pacífico (Osaka): 383597477331
- Ásia-Pacífico (Seul): 600734575887
- Ásia-Pacífico (Singapura): 114774131450
- Ásia-Pacífico (Sydney): 783225319266
- Ásia-Pacífico (Tóquio): 582318560864
- Canadá (Central): 985666609251
- Europa (Frankfurt): 054676820928
- Europa (Irlanda): 156460612806
- Europa (Londres): 652711504416
- Europa (Milão): 635631232127
- Europa (Paris): 009996457667
- Europa (Estocolmo): 897822967062
- Oriente Médio (Bahrein): 076674570225
- América do Sul (São Paulo): 507241528517

Para`Resource`, insira o ARN do local para os registros de acesso, usando o formato mostrado no exemplo de política. Sempre inclua o ID da conta com o balanceador de carga no caminho do recurso do ARN do bucket do S3. Isso garante que somente os balanceadores de carga da conta especificada possam gravar registros de acesso no bucket do S3.

O ARN do bucket do S3 que você especifica depende se você planeja incluir um prefixo ao habilitar os registros de acesso na etapa 3.

Exemplo de ARN do bucket do S3 com um prefixo

O nome do bucket do S3 é amzn-s3-demo-logging-bucket e o prefixo é. logging-prefix

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

Exemplo de ARN do bucket do S3 sem prefixo

O nome do bucket do S3 éamzn-s3-demo-logging-bucket. Não há parte do prefixo no ARN do bucket do S3.

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

AWS GovCloud (US) Regiões

Essa política concede permissões para a conta especificada do Elastic Load Balancing. Use essa política para balanceadores de carga em Zonas de Disponibilidade ou Zonas AWS GovCloud (US) Locais nas Regiões na lista abaixo.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws-us-gov:iam::111122223333:root"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws-us-gov:s3:::amzn-s3-demo-bucket/prefix/
AWSLogs/123456789012/*"
        }
    ]
}
```

Para `Principal` *111122223333* substituir pelo ID da conta do Elastic Load Balancing para a região do balanceador de carga:

- AWS GovCloud (Oeste dos EUA) — 048591011584
- AWS GovCloud (Leste dos EUA) — 190560391635

Para`Resource`, insira o ARN do local para os registros de acesso, usando o formato mostrado no exemplo de política. Sempre inclua o ID da conta com o balanceador de carga no caminho

do recurso do ARN do bucket do S3. Isso garante que seus balanceadores de carga da conta especificada possam gravar registros de acesso no bucket do S3.

O ARN do bucket do S3 que você especifica depende se você planeja incluir um prefixo ao habilitar os registros de acesso.

Exemplo de ARN do bucket do S3 com um prefixo

O nome do bucket do S3 é amzn-s3-demo-logging-bucket e o prefixo é. logging-prefix

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

Exemplo de ARN do bucket do S3 sem prefixo

O nome do bucket do S3 éamzn-s3-demo-logging-bucket. Não há parte do prefixo no ARN do bucket do S3.

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

Zonas de Outposts

A política a seguir concede permissões ao serviço de entrega de logs especificado. Use essa política para balanceadores de carga em zonas de Outposts.

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "logdelivery.elb.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
    "Condition": {
        "StringEquals": {
            "s3:x-amz-acl": "bucket-owner-full-control"
        }
    }
}
```

Para`Resource`, insira o ARN do local para os registros de acesso. Sempre inclua o ID da conta com o balanceador de carga no caminho do recurso do ARN do bucket do S3. Isso garante que somente os balanceadores de carga da conta especificada possam gravar registros de acesso no bucket do S3.

O ARN especificado dependerá de você planejar ou não incluir um prefixo ao habilitar os logs de acesso na etapa 3.

Exemplo de ARN do bucket do S3 com um prefixo

O nome do bucket do S3 é amzn-s3-demo-logging-bucket e o prefixo é. logging-prefix

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

Exemplo de ARN do bucket do S3 sem prefixo

O nome do bucket do S3 éamzn-s3-demo-logging-bucket. Não há parte do prefixo no ARN do bucket do S3.

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

Uso de NotPrincipal quando Effect for Deny

Se a política de bucket do Amazon S3 usar `Effect` com o valor `Deny` e incluir `NotPrincipal` conforme mostrado no exemplo a seguir, certifique-se de que `logdelivery.elasticloadbalancing.amazonaws.com` esteja incluído na lista de `Service`.

```
{
  "Effect": "Deny",
  "NotPrincipal": {
    "Service": [
       "logdelivery.elasticloadbalancing.amazonaws.com",
       "example.com"
    ]
  }
},
```

Para anexar uma política de bucket para logs de conexão ao seu bucket usando o console do Amazon S3

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Selecione o nome do bucket para abrir sua página de detalhes.
3. Escolha Permissions (Permissões) e, em seguida, escolha Bucket policy (Política de bucket), Edit (Editar).
4. Crie ou atualize a política de bucket para conceder as permissões necessárias.

5. Escolha Salvar alterações.

Etapa 3: configurar logs de conexão

Siga o procedimento a seguir para configurar logs de conexão a fim de capturar e entregar arquivos de log ao bucket do S3.

Requisitos

O bucket deverá atender aos requisitos descritos na etapa 1 e você deverá anexar uma política de bucket, conforme descrito na etapa 2. Se você especificar um prefixo, ele não deverá incluir a string "AWSLogs".

Para habilitar os logs de conexão no balanceador de carga usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Load Balancers.
3. Selecione o nome do balanceador de carga para abrir sua página de detalhes.
4. Na guia Atributos, escolha Editar.
5. Em Monitoramento, ative os Logs de conexão.
6. Para URI do S3, insira o URI do S3 para seus arquivos de log. O URI especificado dependerá de você estar ou não usando um prefixo.
   - URI com um prefixo: `s3://bucket-name/prefix`
   - URI sem prefixo: `s3://bucket-name`
7. Escolha Salvar alterações.

Para ativar os registros de conexão usando o AWS CLI

Use o comando modify-load-balancer-attributes.

Para gerenciar o bucket do S3 para os logs de conexão

Certifique-se de desabilitar os logs de conexão antes de excluir o bucket que você configurou para os logs de conexão. Caso contrário, se houver um novo bucket com o mesmo nome e a política de bucket necessária, mas criado em uma Conta da AWS que não seja de sua propriedade, o Elastic Load Balancing poderá gravar os logs de conexão do balanceador de carga nesse novo bucket.

Etapa 4: Verificar permissões do bucket

Depois que os logs de conexão são habilitados para o balanceador de carga, o Elastic Load Balancing validará o bucket do S3 e criará um arquivo de teste para garantir que a política do bucket especifique as permissões necessárias. Você pode usar o console do Amazon S3 para verificar se o arquivo de teste foi criado. O arquivo de teste não é um arquivo de log de conexão real; ele não contêm registros de exemplo.

Para verificar se o Elastic Load Balancing criou um arquivo de teste no seu bucket do S3

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Selecione o nome do bucket que você especificou para os logs de conexão.
3. Localize o arquivo de teste, `ELBConnectionLogTestFile`. O local dependerá de você estar ou não usando um prefixo.
   - Localização com um prefixo:*amzn-s3-demo-logging-bucket*//*prefix*/AWSLogs/*123456789012*ELBConnectionLogTestFile
   - Localização sem prefixo:*amzn-s3-demo-logging-bucket*//AWSLogs/*123456789012*ELBConnectionLogTestFile

Solução de problemas

Se você receber um erro de acesso negado, as possíveis causas serão:

- A política do bucket não concede ao Elastic Load Balancing permissão para gravar logs de conexão no bucket. Confira se está usando a política de bucket correta para a região. Confira se o ARN do recurso usa o mesmo nome de bucket que você especificou ao habilitar os logs de conexão. Confira se o ARN do recurso não inclui um prefixo se você não tiver especificado um ao habilitar os logs de conexão.
- O bucket usa uma opção de criptografia que não é aceita no lado do servidor. O bucket deve usar chaves gerenciadas pelo Amazon S3 (SSE-S3).

Desabilitar os logs de conexão para o Application Load Balancer

Você pode desabilitar logs de conexão para o balanceador de carga a qualquer momento. Depois de desabilitar os logs de conexão, eles permanecerão no bucket do S3 até que você os exclua. Para

obter mais informações, consulte Criação, configuração e trabalho com buckets no Guia do usuário do Amazon S3.

Para desabilitar os logs de conexão usando o console

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Load Balancers.
3. Selecione o nome do balanceador de carga para abrir sua página de detalhes.
4. Na guia Atributos, escolha Editar.
5. Em Monitoramento, desative os Logs de conexão.
6. Escolha Salvar alterações.

Para desativar os registros de conexão usando o AWS CLI

Use o comando modify-load-balancer-attributes.

Solicitar rastreamento para seu Application Load Balancer

Quando o load balancer recebe uma solicitação de um cliente, ele adiciona ou atualiza o cabeçalho X-Amzn-Trace-Id, antes de enviar a solicitação ao destino. Todos os serviços ou aplicativos entre o load balancer e o destino também podem adicionar ou atualizar esse cabeçalho.

Você pode usar o rastreamento de solicitação para rastrear solicitações HTTP de clientes para destinos ou outros serviços. Se você habilitar os logs de acesso, o conteúdo do cabeçalho X-Amzn-Trace-Id será registrado. Para obter mais informações, consulte Logs de acesso para seu Application Load Balancer.

Sintaxe

O cabeçalho X-Amzn-Trace-Id contém campos com o seguinte formato:

```
Field=version-time-id
```

Campo

O nome do campo. Os valores suportados são `Root` e `Self`.

um aplicativo pode adicionar campos arbitrários para as suas próprias finalidades. O load balancer preserva esses campos, mas não os usa.

version

O número da versão. Esse valor é 1.

horário

A hora de referência (epoch), em segundos. Esse valor tem 8 dígitos hexadecimais.

id

O identificador de rastreamento. Esse valor é de 24 dígitos hexadecimais.

Exemplos

Se o cabeçalho X-Amzn-Trace-Id não estiver presente em uma solicitação de entrada, o load balancer deverá gerar um cabeçalho com o campo `Root` e encaminhar a solicitação. Por exemplo:

```
X-Amzn-Trace-Id: Root=1-67891233-abcdef012345678912345678
```

Se o cabeçalho X-Amzn-Trace-Id estiver presente e contiver um campo `Root`, o load balancer inserirá um campo `Self` e encaminhará a solicitação. Por exemplo:

```
X-Amzn-Trace-Id: Self=1-67891233-12456789abcdef012345678;Root=1-67891233-
abcdef012345678912345678
```

Se um aplicativo adicionar um cabeçalho com um campo `Root` e um campo personalizado, o load balancer preservará os dois campos, inserirá um campo `Self` e encaminhará a solicitação:

```
X-Amzn-Trace-Id: Self=1-67891233-12456789abcdef012345678;Root=1-67891233-
abcdef012345678912345678;CalledFrom=app
```

Se o cabeçalho X-Amzn-Trace-Id estiver presente e contiver um campo `Self`, o load balancer atualizará o valor do campo `Self`.

Limitações

- O load balancer atualiza o cabeçalho quando recebe uma solicitação recebida, não quando recebe uma resposta.

- Se os cabeçalhos HTTP tiverem mais de 7 KB, o load balancer reescreverá o cabeçalho X-Amzn-Trace-Id com um campo `Root`.
- Com WebSockets, você pode rastrear somente até que a solicitação de upgrade seja bem-sucedida.

Solucionar problemas em seus Application Load Balancers

As informações a seguir podem ajudar na solução de problemas com seu Application Load Balancer.

Problemas

- Um destino registrado não está em serviço
- Os clientes não conseguem se conectar a um balanceador de carga voltado para a Internet
- As solicitações enviadas para um domínio personalizado não são recebidas pelo balanceador de carga.
- As solicitações HTTPS enviadas ao balanceador de carga retornam “NET::ERR_CERT_COMMON_NAME_INVALID”
- O balanceador de carga mostra tempos elevados de processamento
- O load balancer envia um código de resposta de 000
- O load balancer gera um erro de HTTP
- Um destino gera um erro HTTP
- Um AWS Certificate Manager certificado não está disponível para uso
- Não há compatibilidade com cabeçalhos de várias linhas.
- Solucionar problemas de destinos não íntegros usando o mapa de recursos

Um destino registrado não está em serviço

Se um destino estiver levando mais tempo que o esperado para entrar no estado `InService`, ele pode estar falhando nas verificações de integridade. O destino não entrará em serviço até ser aprovado em uma verificação de integridade. Para obter mais informações, consulte Verificações de integridade para grupos de destino do Application Load Balancer.

Verifique se a sua instância está falhando nas verificações de integridade e verifique os seguintes problemas:

Um security group não permite o tráfego

> O security group associado a uma instância deve permitir tráfego do load balancer usando a porta de verificação de integridade e o protocolo de verificação de integridade. Você pode adicionar uma regra ao security group da instância para permitir todo o tráfego do security group do load

balancer. Além disso, o security group para seu load balancer deve permitir o tráfego para as instâncias.

Uma lista de controle de acesso (ACL) à rede não permite o tráfego

A Network ACL associada às sub-redes para suas instâncias deve permitir tráfego de entrada na porta de verificação de integridade e tráfego de saída nas portas efêmeras (1024-65535). A Network ACL associada às sub-redes para os nós do seu load balancer devem permitir tráfego de entrada nas portas efêmeras e tráfego de saída na verificação de integridade e nas portas efêmeras.

O caminho de ping não existe

Crie uma página de destino para a verificação de integridade e especifique seu caminho como caminho de ping.

A conexão expira

Primeiro, verifique se você pode se conectar ao destino diretamente de dentro da rede usando o endereço IP privado do destino e o protocolo de verificação de integridade. Se você não conseguir se conectar, verifique se a instância está superutilizada e adicione mais destinos ao seu grupo de destino se estiver muito ocupado para responder. Se você puder se conectar, é possível que a página de destino não esteja respondendo antes do período do tempo limite da verificação de integridade. Escolha uma página de destino mais simples para a verificação de integridade ou ajuste as configurações de verificação de integridade.

O destino não retorna um código de resposta bem-sucedido

Por padrão, o código de sucesso é 200, mas você também pode especificar códigos de sucesso adicionais ao configurar as verificações de integridade. Confirme os códigos de sucesso que o load balancer está esperando e se seu aplicativo está configurada para retornar esses códigos com sucesso.

O código de resposta do destino estava mal formado ou houve um erro na conexão com o destino

Verifique se a aplicação responde às solicitações de verificação de integridade do balanceador de carga. Algumas aplicações exigem configuração adicional para responder às verificações de integridade, como uma configuração de host virtual para responder ao cabeçalho do host HTTP enviado pelo balanceador de carga. O valor do cabeçalho do host contém o endereço IP privado do destino, seguido pela porta de verificação de integridade quando não estiver usando uma porta padrão. Se o destino usar uma porta de verificação de integridade padrão, o valor do cabeçalho do host conterá apenas o endereço IP privado do destino. Por exemplo, se o endereço

IP privado do seu destino for `10.0.0.10` e a porta de verificação de integridade for `8080`, o cabeçalho HTTP Host enviado pelo balanceador de carga nas verificações de integridade será `Host: 10.0.0.10:8080`. Se o endereço IP privado do seu destino for `10.0.0.10` e a porta de verificação de integridade for `80`, o cabeçalho HTTP Host enviado pelo balanceador de carga nas verificações de integridade será `Host: 10.0.0.10`. Pode ser necessário realizar uma configuração de host virtual para responder a esse host ou uma configuração padrão para verificar a integridade da aplicação com sucesso. As solicitações de verificação de integridade têm os seguintes atributos: `User-Agent` é definido como `ELB-HealthChecker/2.0`, o terminador de linha para campos de cabeçalho de mensagem é a sequência CRLF e o cabeçalho termina na primeira linha vazia seguida por um CRLF.

Os clientes não conseguem se conectar a um balanceador de carga voltado para a Internet

Se o balanceador de carga não estiver respondendo às solicitações, verifique os seguintes problemas possíveis:

Seu balanceador de carga voltado para a Internet está anexado a uma sub-rede privada

É necessário que você especifique sub-redes públicas para o seu balanceador de carga. Uma sub-rede pública tem uma rota para o Internet Gateway para sua Virtual Private Cloud (VPC).

Um security group ou Network ACL não permite o tráfego

O grupo de segurança do balanceador de carga e qualquer rede ACLs das sub-redes do balanceador de carga deve permitir tráfego de entrada dos clientes e tráfego de saída para os clientes nas portas do ouvinte.

As solicitações enviadas para um domínio personalizado não são recebidas pelo balanceador de carga.

Se o balanceador de carga não estiver recebendo solicitações enviadas para um domínio personalizado, verifique os seguintes problemas:

O nome de domínio personalizado não corresponde ao endereço IP do balanceador de carga.

- Confirme para qual endereço IP o nome de domínio personalizado é resolvido usando uma interface da linha de comando.

    - Linux, macOS ou Unix: você pode usar o comando `dig` no Terminal. Ex.`dig example.com`
    - Windows: você pode usar o comando `nslookup` no Prompt de comando. Ex.`nslookup example.com`
  - Confirme para qual endereço IP o nome DNS dos balanceadores de carga é resolvido usando uma interface da linha de comando.
  - Compare os resultados das duas saídas. É necessário que os endereços IP correspondam.

Se estiver usando o Route 53 para hospedar seu domínio personalizado, consulte Meu domínio não está disponível na Internet no Guia do desenvolvedor do Amazon Route 53.

As solicitações HTTPS enviadas ao balanceador de carga retornam “NET::ERR_CERT_COMMON_NAME_INVALID”

Se as solicitações HTTPS estiverem recebendo `NET::ERR_CERT_COMMON_NAME_INVALID` do balanceador de carga, verifique as seguintes causas possíveis:

- O nome de domínio usado na solicitação HTTPS não corresponde ao nome alternativo especificado no certificado do ACM associado aos receptores.
- O nome DNS padrão dos balanceadores de carga está em uso. Não é possível usar o nome DNS padrão para fazer solicitações HTTPS, pois um certificado público não pode ser solicitado para o domínio `*.amazonaws.com`.

O balanceador de carga mostra tempos elevados de processamento

O balanceador de carga conta os tempos de processamento de maneira diferente com base na configuração.

- Se AWS WAF estiver associado ao seu Application Load Balancer e um cliente enviar uma solicitação HTTP POST, o tempo de envio dos dados para solicitações POST será refletido no `request_processing_time` campo nos registros de acesso do balanceador de carga. Espera-se esse comportamento para solicitações HTTP POST.
- Se não AWS WAF estiver associado ao seu Application Load Balancer e um cliente enviar uma solicitação HTTP POST, o tempo de envio dos dados para solicitações POST será refletido no

`target_processing_time` campo nos registros de acesso do balanceador de carga. Espera-se esse comportamento para solicitações HTTP POST.

O load balancer envia um código de resposta de 000

Com conexões HTTP/2, se o número de solicitações atendidas por meio de uma conexão exceder 10.000, o balanceador de carga enviará um quadro GOAWAY e fechará a conexão com um TCP FIN.

O load balancer gera um erro de HTTP

Os seguintes erros de HTTP são gerados pelo load balancer. O load balancer envia o código HTTP para o cliente, salva a solicitação no log de acesso e incrementa a métrica `HTTPCode_ELB_4XX_Count` ou `HTTPCode_ELB_5XX_Count`.

Erros

- HTTP 400: solicitação inválida
- HTTP 401: Não autorizado
- HTTP 403: negado
- HTTP 405: método não permitido
- HTTP 408: Request Timeout (HTTP 408: limite de tempo de solicitação)
- HTTP 413: carga útil muito grande
- HTTP 414: URI muito longo
- HTTP 460
- HTTP 463
- HTTP 464
- HTTP 500: erro interno do servidor
- HTTP 501: não implementado
- HTTP 502: Bad Gateway (HTTP 502: gateway incorreto)
- HTTP 503: Service Unavailable (HTTP 503: serviço indisponível)
- HTTP 504: Gateway Timeout (HTTP 504: limite de tempo do gateway)
- HTTP 505: versão incompatível

- HTTP 507: armazenamento insuficiente
- HTTP 561: Não autorizado

HTTP 400: solicitação inválida

Causas possíveis:

- O cliente enviou uma solicitação malformada que não atende às especificações de HTTP.
- O cabeçalho de solicitação excedeu 16 K por linha de solicitação, 16 K por cabeçalho único ou 64 K para o cabeçalho da solicitação inteira.
- O cliente fechou a conexão antes de enviar o corpo completo da solicitação.

HTTP 401: Não autorizado

Você configurou uma regra de listener para autenticar usuários, mas uma das seguintes afirmações é verdadeira:

- Você configurou `OnUnauthenticatedRequest` para rejeitar usuários não autenticados ou o IdP negou acesso.
- O tamanho das solicitações retornadas pelo IdP excedeu o tamanho máximo permitido pelo load balancer.
- Um cliente enviou uma solicitação HTTP/1.0 sem um cabeçalho de host e o load balancer não conseguiu gerar uma URL de redirecionamento.
- O escopo solicitado não retorna um token de ID.
- Você não conclui o processo de login antes da expiração do tempo limite de login do cliente. Para obter mais informações, consulte Tempo limite de login do cliente.

HTTP 403: negado

Você configurou uma lista de controle de acesso à AWS WAF web (web ACL) para monitorar solicitações ao seu Application Load Balancer e ela bloqueou uma solicitação.

HTTP 405: método não permitido

O cliente usou o método TRACE, que não é compatível com Application Load Balancers.

HTTP 408: Request Timeout (HTTP 408: limite de tempo de solicitação)

O cliente não enviou dados antes que o tempo limite de inatividade expirasse. Enviar um keep-alive do TCP. não impede esse limite. Envie pelo menos 1 byte de dados antes que transcorra cada período de tempo limite de inatividade. Aumente a duração do período do tempo limite de inatividade conforme o necessário.

HTTP 413: carga útil muito grande

Causas possíveis:

- O destino é uma função do Lambda e o corpo da solicitação excede 1 MB.
- O cabeçalho de solicitação excedeu 16 K por linha de solicitação, 16 K por cabeçalho único ou 64 K para o cabeçalho da solicitação inteira.

HTTP 414: URI muito longo

A URL da solicitação ou os parâmetros da string de consulta são muito grandes.

HTTP 460

O load balancer recebeu uma solicitação de um cliente, mas o cliente encerrou a conexão com ele antes de decorrido o tempo limite de inatividade.

Verifique se o período de tempo de espera do cliente é maior do que o período de tempo limite de inatividade para o load balancer. Verifique se seu destino fornece uma resposta ao cliente antes do fim do tempo limite do cliente ou aumente o período do tempo limite do cliente de acordo com o tempo limite de inatividade do load balancer, se o cliente for compatível com este.

HTTP 463

O balanceador de carga recebeu um cabeçalho de solicitação X-Forwarded-For com muitos endereços IP. O limite superior para endereços IP é 30.

HTTP 464

O balanceador de carga recebeu um protocolo de solicitação de entrada que é incompatível com a configuração da versão do protocolo do grupo de destino.

Causas possíveis:

- O protocolo de solicitação é HTTP/1.1, enquanto a versão do protocolo do grupo de destino é gRPC ou HTTP/2.
- O protocolo de solicitação é gRPC, enquanto a versão do protocolo do grupo de destino é HTTP/1.1.
- O protocolo de solicitação é HTTP/2 e a solicitação não é POST, enquanto a versão do protocolo do grupo de destino é gRPC.

HTTP 500: erro interno do servidor

Causas possíveis:

- Você configurou uma lista de controle de acesso à AWS WAF web (Web ACL) e houve um erro ao executar as regras da Web ACL.
- O load balancer não consegue se comunicar com o endpoint de token do IdP ou o endpoint de informações do usuário do IdP.
  - Verifique se é possível resolver o DNS do IdP publicamente.
  - Verifique se os grupos de segurança do seu balanceador de carga e a rede ACLs da sua VPC permitem acesso externo a esses endpoints.
  - Verifique se a VPC tem acesso à Internet. Se você tiver um load balancer interno, use um gateway NAT para permitir acesso à internet.
- A reivindicação do usuário recebida do IdP tem tamanho superior a 11 KB.
- O endpoint do token IdP ou o endpoint de informações do usuário do IdP está levando mais de 5 segundos para responder.

HTTP 501: não implementado

O load balancer recebeu um cabeçalho Transfer-Encoding (Codificação de transferência) com um valor não compatível. Os valores compatíveis para Transfer-Encoding (Codificação de transferência) são `chunked` e `identity`. Como alternativa, você pode usar o cabeçalho Content-Encoding.

HTTP 502: Bad Gateway (HTTP 502: gateway incorreto)

Causas possíveis:

- O load balancer recebeu um TCP RST do destino ao tentar estabelecer uma conexão.
- O load balancer recebeu uma resposta inesperada do destino, como "ICMP Destination unreachable (Host unreachable)" (Destino ICMP inacessível (Host inacessível)), ao tentar estabelecer uma conexão. Verifique se o tráfego é permitido das sub-redes do load balancer para os destinos na porta de destino.
- O destino fechou a conexão com um TCP RST ou TCP FIN enquanto o load balancer tinha uma solicitação pendente para o destino. Verifique se a duração de keep-alive do destino é mais curta que o valor do tempo limite de inatividade do load balancer.
- A resposta de destino é malformada ou contém cabeçalhos HTTP inválidos.
- O cabeçalho de resposta de destino excedeu 32 K para o cabeçalho de resposta inteiro.
- O período de atraso no cancelamento do registro decorrido para uma solicitação processada por um destino que foi cancelado. Aumente o período de atraso para que operações demoradas possam ser concluídas.
- O destino é uma função Lambda e o corpo da resposta excede 1 MB.
- O destino é uma função Lambda que não respondeu antes que seu tempo limite configurado fosse atingido.
- O destino é uma função do Lambda que retornou um erro ou a função passou por controle de utilização pelo serviço Lambda.
- O balanceador de carga encontrou um erro do handshake do SSL ao se conectar a um destino.

Para obter mais informações, consulte Como solucionar erros HTTP 502 do Application Load Balancer no AWS Support Knowledge Center.

HTTP 503: Service Unavailable (HTTP 503: serviço indisponível)

Os grupos-alvo do balanceador de carga não têm destinos registrados ou todos os destinos registrados estão em um `unused` estado.

HTTP 504: Gateway Timeout (HTTP 504: limite de tempo do gateway)

Causas possíveis:

- O load balancer não conseguiu estabelecer uma conexão com o destino antes da expiração do tempo limite de conexão (10 segundos).
- O load balancer estabeleceu uma conexão com o destino, mas o destino não respondeu antes de decorrido o tempo limite de inatividade.

- A ACL ou SecurityGroup as políticas da rede não permitiam o tráfego dos destinos para os nós do balanceador de carga nas portas efêmeras (1024-65535).
- O destino retorna um cabeçalho content-length maior do que o corpo da entidade. O load balancer atingiu o tempo limite enquanto aguardava pelos bytes faltantes.
- O destino é uma função do Lambda e o serviço do Lambda não respondeu antes da expiração do tempo limite da conexão.
- O balanceador de carga encontrou um tempo limite do handshake de SSL (10 segundos) ao se conectar a um destino.

HTTP 505: versão incompatível

O balanceador de carga recebeu uma solicitação inesperada de versão HTTP. Por exemplo, o balanceador de carga estabeleceu uma conexão HTTP/1, mas recebeu uma solicitação HTTP/2.

HTTP 507: armazenamento insuficiente

O URL de redirecionamento é muito longo.

HTTP 561: Não autorizado

Você configurou uma regra do listener para autenticar usuários, mas o IdP retornou um código de erro ao autenticar o usuário. Verifique seus logs de acesso para ver o código do motivo do erro relacionado.

Um destino gera um erro HTTP

O load balancer encaminhará respostas HTTP válidas dos destinos para o cliente, incluindo erros de HTTP. Os erros HTTP gerados por um destino são registrados nas métricas `HTTPCode_Target_4XX_Count` e `HTTPCode_Target_5XX_Count`.

Um AWS Certificate Manager certificado não está disponível para uso

Ao decidir usar um ouvinte HTTPS com seu Application Load Balancer AWS Certificate Manager , é necessário validar a propriedade do domínio antes de emitir um certificado. Se essa etapa for omitida durante a configuração, o certificado permanecerá no estado `Pending Validation` e não estará disponível para uso até que seja validado.

- Se estiver usando a validação de e-mail, consulte Validação de e-mail no Guia do usuário do AWS Certificate Manager .
- Se estiver usando a validação de DNS, consulte Validação de DNS no Guia do usuário do AWS Certificate Manager .

Não há compatibilidade com cabeçalhos de várias linhas.

Os Application Load Balancers não são compatíveis com cabeçalhos de várias linhas, incluindo o cabeçalho do tipo de mídia `message/http`. Quando um cabeçalho de várias linhas é fornecido, o Application Load Balancer acrescenta um caractere de dois pontos, "`:`", antes de transmiti-lo para o destino.

Solucionar problemas de destinos não íntegros usando o mapa de recursos

Se os destinos do Application Load Balancer estiverem falhando nas verificações de integridade, você poderá usar o mapa de recursos para encontrar destinos não íntegros e realizar ações com base no código do motivo da falha. Para obter mais informações, consulte Exibir o mapa de recursos do Application Load Balancer.

O mapa de recursos fornece duas exibições: Visão geral e Mapa de destino não íntegro. A opção Visão geral é selecionada por padrão e exibe todos os recursos do balanceador de carga. Selecionar a exibição Mapa de destino não íntegro mostrará somente os destinos não íntegros em cada grupo de destino associado ao Application Load Balancer.

> **Note**
>
> Você deve habilitar a opção Mostrar detalhes do recurso para exibir o resumo da verificação de integridade e as mensagens de erro de todos os recursos aplicáveis no mapa de recursos. Quando não habilitado, você deve selecionar cada recurso para exibir os detalhes.

A coluna Grupos de destino exibe um resumo dos destinos íntegros e não íntegros de cada grupo de destino. Isso pode ajudar a determinar se todos os destinos estão falhando nas verificações de integridade ou somente destinos específicos. Se todos os destinos em um grupo de destino falharem nas verificações de integridade, averigue a configuração do grupo de destino. Selecione o nome de um grupo de destino para abrir a página de detalhes em uma nova guia.

A coluna Destinos exibe o TargetID e o status atual da verificação de integridade de cada destino. Quando um destino não está íntegro, o código do motivo da falha da verificação de integridade é exibido. Quando um único destino falha em uma verificação de integridade, verifique se o destino tem recursos suficientes e confirme se as aplicações em execução no destino estão disponíveis. Selecione um ID de destino para abrir a página de detalhes em uma nova guia.

Selecionar Exportar oferece a opção de exportar a exibição atual do mapa de recursos do Application Load Balancer como PDF.

Verifique se a instância está falhando nas verificações de integridade e, com base no código do motivo da falha, verifique os seguintes problemas:

- Não íntegro: incompatibilidade de resposta HTTP
  - Verifique se a aplicação em execução no destino está enviando a resposta HTTP correta às solicitações de verificação de integridade do Application Load Balancer.
  - Como alternativa, você pode atualizar a solicitação de verificação de integridade do Application Load Balancer para corresponder à resposta da aplicação em execução no destino.
- Não íntegro: a solicitação atingiu o tempo limite
  - Verifique se os grupos de segurança e as listas de controle de acesso (ACL) à rede associados aos seus destinos e ao Application Load Balancer não estão bloqueando a conectividade.
  - Verifique se o destino tem recursos suficientes disponíveis para aceitar conexões do Application Load Balancer.
  - Verifique o status de todas as aplicações em execução no destino.
  - As respostas da verificação de integridade do Application Load Balancer podem ser exibidas nos logs da aplicação de cada destino. Para obter mais informações, consulte Health check reason codes.
- Insalubre: FailedHealthChecks
  - Verifique o status de todas as aplicações em execução no destino.
  - Verifique se o destino está recebendo tráfego na porta de verificação de integridade.

> **Ao usar um receptor HTTPS**
>
> Você escolhe a política de segurança usada para conexões frontend. A política de segurança usada em conexões de backend é selecionada automaticamente com base na política de segurança de frontend em uso.

- Se o receptor HTTPS estiver usando uma política de segurança TLS 1.3 para conexões frontend, a política de segurança `ELBSecurityPolicy-TLS13-1-0-2021-06` será usada para conexões backend.
- Se seu receptor HTTPS não estiver usando uma política de segurança TLS 1.3 para conexões frontend, a política de segurança `ELBSecurityPolicy-2016-08` será usada para conexões backend.

Para obter mais informações, consulte Políticas de segurança.

- Verifique se o destino está fornecendo um certificado e uma chave de servidor no formato correto especificado pela política de segurança.
- Verifique se o destino oferece suporte a uma ou mais cifras correspondentes e a um protocolo fornecido pelo Application Load Balancer para estabelecer handshakes de TLS.

Cotas para seus Application Load Balancers

Sua AWS conta tem cotas padrão, anteriormente chamadas de limites, para cada AWS serviço. A menos que especificado de outra forma, cada cota é específica da região . Você pode solicitar o aumento de algumas cotas, porém, algumas delas não podem ser aumentadas.

Para visualizar as cotas para os Application Load Balancers, abra o console do Service Quotas. No painel de navegação, selecione Serviços da AWS e Elastic Load Balancing. Você também pode usar o comando describe-account-limits(AWS CLI) para o Elastic Load Balancing.

Para solicitar um aumento da cota, consulte Requesting a quota increase no Guia do usuário do Service Quotas. Se a cota ainda não estiver disponível nas Cotas de Serviço, envie uma solicitação para aumentar a cota de serviço.

Cotas

- Balanceadores de cargas
- Grupos de destino
- Regras
- Armazenamentos confiáveis
- Certificados
- Cabeçalhos HTTP
- Unidades de capacidade do Load Balancer

Balanceadores de cargas

Sua AWS conta tem as seguintes cotas relacionadas aos Application Load Balancers.

Name	Padrão	Ajustável
Application Load Balancers por região	50	Sim
Certificados por Application Load Balancer (exceto certificados padrão)	25	Sim
Receptores por Application Load Balancer	50	Sim

Name	Padrão	Ajustável
Grupos-alvo por ação por Application Load Balancer	5	Não
Grupos de destino por Application Load Balancer	100	Não
Metas por Application Load Balancer	1.000	Sim

Grupos de destino

As cotas a seguir são para grupos de destino.

Name	Padrão	Ajustável
Grupos de destino por região	3.000*	Sim
Destinos por grupo de destino por região (instâncias ou endereços IP)	1.000	Sim
Destinos por grupo de destino por região (funções do Lambda)	1	Não
Load balancers por grupo de destino	1	Não

* Essa cota é compartilhada por Application Load Balancers e Network Load Balancers.

Regras

As cotas a seguir são para regras.

Name	Padrão	Ajustável
Regras por Application Load Balancer (exceto regras padrão)	100	Sim
Valores de condição por regra	5	Não

Name	Padrão	Ajustável
Condição: curingas por regra	5	Não
Avaliações de correspondência por regra	5	Não

Armazenamentos confiáveis

As cotas a seguir servem para armazenamentos confiáveis.

Name	Padrão	Ajustável
Armazenamentos confiáveis por conta	20	Sim
Número de receptores usando mTLS no modo de verificação, por balanceador de carga.	2	Não

Certificados

As cotas a seguir se aplicam aos certificados, incluindo nomes de certificados de CA de publicidade e listas de revogação de certificados.

Name	Padrão	Ajustável
Tamanho do certificados de CA	16 KB	Não
Certificados de CA por armazenamento confiável	25	Sim
Tamanho do assunto dos certificados CA por armazenamento confiável	10.000	Sim
Profundidade máxima da cadeia de certificados	4	Não
Entradas de revogação por armazenamento confiável	500.000	Sim
Tamanho do arquivo da lista de revogações	50 MB	Não

Name	Padrão	Ajustável
Listas de revogação por armazenamento confiável	30	Sim
Tamanho da mensagem TLS	64 K	Não

Cabeçalhos HTTP

Os cabeçalhos HTTP têm os seguintes limites de tamanho.

Name	Padrão	Ajustável
Linha de solicitação	16 K	Não
Cabeçalho único	16 K	Não
Cabeçalho de resposta inteiro	32 K	Não
Cabeçalho da solicitação inteira	64 K	Não

Unidades de capacidade do Load Balancer

As cotas a seguir são para Load Balancer Capacity Units (LCU).

Name	Padrão	Ajustável
Unidades de capacidade reservadas do Application Load Balancer (LCUs) por Application Load Balancer	15.000	Sim
Unidades de capacidade reservadas do Application Load Balancer (LCU) por região	0	Sim

Histórico do documento dos Application Load Balancers

A tabela a seguir descreve as versões dos Application Load Balancers.

Alteração	Descrição	Data
modificação do cabeçalho HTTP	Esta versão adiciona suporte à modificação do cabeçalho HTTP para todos os códigos de resposta. Anteriormente, esse recurso era limitado aos códigos de resposta 2xx e 3xx.	28 de fevereiro de 2025
Reserva da unidade de capacidade	Esta versão adiciona suporte para definir uma capacidade mínima para seu balanceador de carga.	20 de novembro de 2024
Mapa de recursos	Esta versão adiciona suporte para exibir os recursos e relacionamentos do balanceador de carga em um formato visual.	8 de março de 2024
WAF com um clique	Esta versão adiciona suporte para configurar o comportamento do seu balanceador de carga se ele se integrar com um clique. AWS WAF	6 de fevereiro de 2024
TLS mútuo	Esta versão adiciona suporte para autenticação TLS mútua.	26 de novembro de 2023
Ponderações de destino automáticas	Esta versão adiciona suporte ao algoritmo de ponderações de destino automáticas.	26 de novembro de 2023

Encerramento de TLS com FIPS 140-3	Esta versão adiciona políticas de segurança que usam módulos criptográficos do FIPS 140-3 ao encerrar conexões TLS.	20 de novembro de 2023
Registre alvos usando IPv6	Esta versão adiciona suporte para registrar instâncias como destinos quando abordadas por IPv6.	2 de outubro de 2023
Políticas de segurança compatíveis com TLS 1.3	Esta versão adiciona suporte a políticas de segurança predefinidas com TLS 1.3.	22 de março de 2023
Mudança de zona	Esta versão adiciona suporte para rotear o tráfego para fora de uma única zona de disponibilidade prejudicada por meio da integração com o Controlador de Recuperação de Aplicações (ARC) da Amazon.	28 de novembro de 2022
Desativar o balanceamento de carga entre zonas	Esta versão adiciona suporte para desativar o balanceamento de carga entre zonas.	28 de novembro de 2022
Integridade do grupo de destino	Esta versão adiciona suporte para configurar a contagem ou a porcentagem mínima de destinos que devem estar íntegros e quais ações o balanceador de carga executará quando o limite não for atingido.	28 de novembro de 2022

Balanceamento de carga entre zonas	Esta versão adiciona suporte para configurar o balanceamento de carga entre zonas em nível de grupo de destino.	17 de novembro de 2022
IPv6 grupos-alvo	Esta versão adiciona suporte para configurar grupos de IPv6 destino para Application Load Balancers.	23 de novembro de 2021
IPv6 balanceadores de carga internos	Esta versão adiciona suporte para configurar grupos de IPv6 destino para Application Load Balancers.	23 de novembro de 2021
AWS PrivateLink e endereços IP estáticos	Esta versão adiciona suporte para usar AWS PrivateLink e expor endereços IP estáticos ao encaminhar o tráfego diretamente dos Network Load Balancers para os Application Load Balancers.	27 de setembro de 2021
Preservação da porta do cliente	Esta versão adiciona um atributo para preservar a porta de origem que o cliente usou para estabelecer conexão com o balanceador de carga.	29 de julho de 2021

Cabeçalhos de TLS	Esta versão adiciona um atributo para indicar se os cabeçalhos de TLS, que contêm informações sobre a versão negociada do TLS e o conjunto de cifras, serão adicionados à solicitação do cliente antes de enviá-la ao destino.	21 de julho de 2021
Certificados adicionais do ACM	Esta versão é compatível com certificados RSA com comprimentos de chave de 2.048, 3.072 e 4.096 bits, e com todos os certificados ECDSA.	14 de julho de 2021
Persistência com base em aplicação	Esta versão adiciona um cookie baseado em aplicação para oferecer suporte a sessões persistentes para seu balanceador de carga.	8 de fevereiro de 2021
Política de segurança para FS compatível com TLS versão 1.2	Esta versão adiciona uma política de segurança para Forward Secrecy (FS – Sigilo de encaminhamento) compatível com TLS versão 1.2.	24 de novembro de 2020
Compatibilidade com falha na abertura do WAF	Esta versão adiciona suporte para configurar o comportamento do seu balanceador de carga se ele se integrar com o AWS WAF	13 de novembro de 2020

Compatibilidade com gRPC e HTTP/2	Esta versão adiciona suporte para cargas de trabalho gRPC e HTTP/2. end-to-end	29 de outubro de 2020
Compatibilidade com Outpost	Você pode provisionar um Application Load Balancer no AWS Outposts.	8 de setembro de 2020
Modo de mitigação de dessincronização	Esta versão adiciona suporte ao modo de mitigação de dessincronização.	17 de agosto de 2020
Solicitações menos pendentes	Esta versão agora comporta o algoritmo de solicitações menos pendentes.	25 de novembro de 2019
Grupos de destino ponderados	Esta versão adiciona suporte a ações de encaminhamento com vários grupos de destino. As solicitações são distribuí das para esses grupos de destino com base no peso especificado para cada grupo de destino.	19 de novembro de 2019
New attribute (Novo atributo)	Esta versão adiciona suporte ao atributo routing.http.drop_ invalid_header_fields.enabled.	15 de novembro de 2019
Políticas de segurança para FS	Essa versão adiciona suporte para três políticas adicionai s de segurança de sigilo de encaminhamento predefinidas.	8 de outubro de 2019
Roteamento avançado de solicitação	Essa versão adiciona suporte para tipos de condição adicionais das regras do listener.	27 de março de 2019

Funções do Lambda como destino	Esta versão inclui o suporte para registrar funções Lambda como destino.	29 de novembro de 2018
Ações de redirecionamento	Esta versão inclui suporte para que o load balancer redirecione solicitações para um URL diferente.	25 de julho de 2018
Ações de resposta fixa	Esta versão inclui suporte para que o load balancer retorne uma resposta HTTP personalizada.	25 de julho de 2018
Políticas de segurança para o FS e TLS 1.2	Essa versão agora comporta duas outras políticas de segurança predefinidas.	6 de junho de 2018
Autenticação de usuário	Essa versão agora oferece compatibilidade com o load balancer para autenticar os usuários de seus aplicativos usando a identidade corporativa ou social desses usuários antes das solicitações de roteamento.	30 de maio de 2018
Permissões em nível de recurso	Essa versão agora comporta permissões em nível de recursos e chaves de condição de marcação.	10 de maio de 2018

Modo de iniciação lenta	Essa versão adiciona suporte para o modo de iniciação lenta, que aumenta gradualmente a parte de solicitações que o load balancer envia para um destino recém-registrado enquanto ele aquece.	24 de março de 2018
Suporte a SNI	Esta versão acrescenta suporte a SNI (Server Name Indication, indicação de nome de servidor).	10 de outubro de 2017
Endereços IP como destinos	Esta versão inclui o suporte para registrar endereços IP como destinos.	31 de agosto de 2017
Roteamento baseado em host	Esta versão agora comporta solicitações de roteamento com base nos nomes de host no cabeçalho de host.	5 de abril de 2017
Políticas de segurança para TLS 1.1 e TLS 1.2	Esta versão inclui políticas de segurança para TLS 1.1 e TLS 1.2.	6 de fevereiro de 2017
IPv6 apoio	Esta versão adiciona suporte para IPv6 endereços.	25 de janeiro de 2017
Rastreamento de solicitação	Esta versão adiciona suporte ao rastreamento de solicitação.	22 de novembro de 2016
Suporte de percentis para a métrica TargetResponseTime	Esta versão adiciona suporte às novas estatísticas percentuais suportadas pela Amazon. CloudWatch	17 de novembro de 2016

Novo tipo de balanceador de carga	Esta versão do Elastic Load Balancing introduz os Application Load Balancers.	11 de agosto de 2016

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.