



Guia do Desenvolvedor

AWS Elastic Beanstalk



AWS Elastic Beanstalk: Guia do Desenvolvedor

Copyright © 2024 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 desprestige a Amazon. Todas as outras marcas comerciais que não são propriedade da Amazon pertencem aos respectivos proprietários, os quais podem ou não ser afiliados, estar conectados ou ser patrocinados pela Amazon.

Table of Contents

O que é o AWS Elastic Beanstalk?	1
Definição de preços	2
Para onde ir em seguida	2
Conceitos básicos	4
Configurando: Crie uma AWS conta	4
Inscreva-se para um Conta da AWS	4
Crie um usuário com acesso administrativo	5
Etapa 1: Criar	6
Criar um aplicativo e um ambiente	6
AWS recursos criados para o aplicativo de exemplo	11
Etapa 2: Explorar	12
Etapa 3: implantar uma nova versão	14
Etapa 4: Configurar	17
Fazer uma alteração de configuração	17
Verificar a alteração de configuração	18
Etapa 5: limpar	19
Próximas etapas	20
Conceitos	23
Aplicativo	23
Versão do aplicativo	23
Ambiente	23
Nível do ambiente	23
Configuração de ambientes	24
Configuração salva	24
Plataforma	24
Ambientes de servidor da web	24
Ambientes de operador	26
Considerações sobre design	27
Escalabilidade	28
Segurança	28
Armazenamento persistente	29
Tolerância a falhas	30
Entrega de conteúdo	31
Atualizações e correções de software	31

Conectividade	31
Permissões	33
Função de serviço	34
Perfil da instância	44
Política de usuário	45
Plataformas	46
Glossário de plataformas	46
Modelo de responsabilidade compartilhada	50
Política de suporte de plataforma	51
Ramificações desativadas da plataforma	51
Após o período de carência de 90 dias	52
Cronograma da plataforma	53
Recursos de planejamento	54
Próximos lançamentos de filiais da plataforma	54
Programação de ramificações desativadas da plataforma	55
Histórico de ramificações de plataforma retiradas	55
Histórico do servidor e do sistema operacional	60
Plataformas compatíveis	62
Plataformas compatíveis	62
Plataformas Linux	63
Versões compatíveis do Amazon Linux	64
Lista de plataformas Linux do Elastic Beanstalk	65
Estender plataformas Linux	66
Como trabalhar com o Docker	92
As ramificações da plataforma Docker	92
Ramificação da plataforma Docker	94
Ramificação da plataforma gerenciada pelo ECS	119
Contêineres pré-configurados	149
Configuração de ambientes	157
Trabalho com Go	171
Conceitos básicos	172
Ambiente de desenvolvimento	179
A Plataforma Go	180
Tutorial para o Go	189
Como trabalhar com Java	194
Conceitos básicos	195

Ambiente de desenvolvimento	207
A Plataforma Tomcat	209
Plataforma Java SE	227
Adicionar um banco de dados	238
Toolkit do Eclipse	247
Recursos	267
Trabalhar com .NET Core no Linux	267
Conceitos básicos	268
Ambiente de desenvolvimento	276
O .NET Core na plataforma Linux	276
Tutorial: .NET Core no Linux	283
A AWS Toolkit for Visual Studio	291
Migração do Windows para o Linux	316
Trabalhando com o.NET no servidor Windows	317
Conceitos básicos	320
Ambiente de desenvolvimento	329
A Plataforma .NET	330
Tutorial - ASP.NET Core	344
Adicionar um banco de dados	356
A AWS Toolkit for Visual Studio	360
Migrar o aplicativo local	395
Recursos	395
Trabalho com Node.js	396
Conceitos básicos	396
Ambiente de desenvolvimento	398
A plataforma Node.js	401
Tutorial – Express	418
Tutorial: Express com clustering	430
Tutorial: Node.js com DynamoDB	448
Adicionar um banco de dados	460
Recursos	463
Como trabalhar com PHP	463
Ambiente de desenvolvimento	465
A Plataforma PHP	468
Tutorial - Laravel	478
Tutorial – CakePHP	488

Tutorial - Symfony	496
Tutorial – produção de HA	503
Tutorial - HA WordPress	515
Tutorial – Drupal de alta disponibilidade	533
Adicionar um banco de dados	552
Como trabalhar com Python	556
Ambiente de desenvolvimento	556
A Plataforma Python	559
Tutorial – flask	568
Tutorial - Django	577
Adicionar um banco de dados	591
Recursos	594
Trabalhar com Ruby	594
Ambiente de desenvolvimento	595
A Plataforma Ruby	598
Tutorial – rails	605
Tutorial – sinatra	614
Adicionar um banco de dados	620
Tutoriais e amostras	623
Gerenciar aplicativos	626
Console de gerenciamento de aplicativos	628
Gerenciar versões de aplicativos	629
Ciclo de vida da versão	633
Atribuir tags em versões do aplicativo	636
Criar um pacote de origem	638
Criar pacote de origem a partir da linha de comando	639
Criar pacote de origem com o Git	640
Compactar arquivos no Mac OS X Finder ou Windows Explorer	640
Criar pacote de origem para um aplicativo .NET	643
Testar pacote de origem	644
Marcar recursos	645
Propagação de tags para modelos de execução	647
Recursos que podem ser marcados	648
Marcar aplicativos	648
Gerenciar ambientes	652
Console de gerenciamento de ambiente	653

Visão geral do ambiente	654
Ações do ambiente	657
Eventos	659
Integridade	660
Logs	660
Monitorar	661
alarmes	662
Atualizações gerenciadas	662
Tags	663
Configuração	664
Criar ambientes	666
O assistente de criação de novo ambiente	674
Clonar um ambiente	698
Encerrar um ambiente	701
Com a AWS CLI	703
Com API	705
URL "Iniciar agora"	709
Compose Environments	715
Implantações	718
Como escolher uma política de implantação	719
Implantar uma nova versão do aplicativo	722
Reimplantar a versão anterior	722
Outras formas de implantar seu aplicativo	723
Opções de implantação	723
Implantações azuis/verdes	732
Alterações de configuração	735
Atualizações contínuas	736
Atualizações imutáveis	742
Atualizações da plataforma	746
Método 1: Atualizar a versão da plataforma do ambiente	751
Método 2: Executar uma implantação azul/verde	753
Atualizações gerenciadas	754
Atualizar um ambiente legado	762
Migrar para AL2023/AL2	764
Perguntas frequentes sobre descontinuação de plataformas	782
Cancelar uma atualização	787

Reconstruir um ambiente	788
Reconstrução de um ambiente em execução	789
Reconstruir um ambiente encerrado	789
Tipos de ambiente	792
Ambiente escalável e com balanceamento de carga	792
Ambiente de instância única	792
Alterar tipo de ambiente	793
Ambientes de operador	794
Daemon SQS do ambiente de operador	797
Filas de mensagens mortas	798
Tarefas periódicas	799
Usar o Amazon CloudWatch para escalabilidade automática em níveis do ambiente de operador	801
Configurar ambientes de operador	801
Links de ambiente	805
Configurar ambientes	808
Configurar usando o console	810
Página de configuração	810
Página Revisar alterações	812
Instâncias do Amazon EC2	813
Tipos de instância do Amazon EC2	814
Configurar instâncias do Amazon EC2 para seu ambiente	815
Configurar instâncias do EC2 da AWS para seu ambiente usando a AWS CLI	823
Recomendações para ambientes de primeiro grupo Graviton arm64	827
O namespace <code>aws:autoscaling:launchconfiguration</code>	829
IMDS	830
Grupo de Auto Scaling	833
Suporte à instância spot	834
Configuração do grupo de Auto Scaling usando o console do Elastic Beanstalk	838
Configuração do grupo de Auto Scaling usando a CLI do EB	842
Opções de configuração	843
Gatilhos	844
Ações programadas	847
Configuração de verificação de integridade	852
Load balancer	853
Classic Load Balancer	855

Application Load Balancer	867
Application Load Balancer compartilhado	887
Network Load Balancer	907
Configurar logs de acesso	920
Banco de dados	920
Ciclo de vida do banco de dados	921
Adição de uma instância de banco de dados do Amazon RDS ao seu ambiente usando o console	922
Conectar ao banco de dados	924
Configuração de uma instância de banco de dados do RDS integrada, usando o console ...	925
Configuração de uma instância de banco de dados do RDS integrada usando arquivos de configuração	926
Desacoplamento de uma instância de banco de dados do RDS usando o console	927
Desacoplamento de uma instância de banco de dados do RDS usando arquivos de configuração	930
Segurança	932
Configurar a segurança do ambiente	932
Namespaces de configuração de segurança do ambiente	935
Marcação em ambientes	935
Adicionar tags durante a criação do ambiente	936
Gerenciar tags de um ambiente existente	937
Propriedades de ambientes e configurações de software	939
Definir configurações específicas da plataforma	940
Configurar propriedades do ambiente (variáveis de ambiente)	941
Namespaces de configurações de software	943
Acessar propriedades de ambiente	945
Depuração	946
Visualização de logs	949
Notificações	952
Configurar notificações usando o console do Elastic Beanstalk	953
Configurar notificações usando opções de configuração	954
Configurar permissões para enviar notificações	956
Amazon VPC	958
Definir as configurações da VPC no console do Elastic Beanstalk	959
O namespace aws:ec2:vpc	962
Migrar do EC2-Classical para uma VPC	963

Nome de domínio	968
Configurar ambientes (avançado)	970
Opções de configuração	971
Precedência	971
Valores recomendados	972
Antes da criação do ambiente	974
Durante a criação	980
Depois da criação	987
Opções gerais	998
Opções específicas para a plataforma	1084
Opções personalizadas	1097
.Ebextensions	1098
Configurações de opção	1101
Servidor do Linux	1103
Servidor do Windows	1121
Recursos personalizados	1130
Configurações salvas	1159
Marcação de configurações salvas	1165
env.yaml	1167
Imagem personalizada	1170
Criar uma AMI do personalizada	1171
Limpeza de uma AMI personalizada	1175
AMI baseada em plataforma descontinuada	1176
Arquivos estáticos	1182
Configurar arquivos estáticos usando o console	1183
Configurar arquivos estáticos usando opções de configuração	1184
HTTPS	1185
Criar um certificado	1187
Upload de um certificado	1190
Encerrar no load balancer	1191
Encerrar na instância	1195
Criptografia de ponta a ponta	1232
Passagem de TCP	1236
Armazenar chaves com segurança	1237
Redirecionamento de HTTP para HTTPS	1239
Monitoramento de ambiente	1241

Console de monitoramento	1241
Grafos de monitoramento	1242
Personalizar o console de monitoramento	1243
Relatórios de integridade básica	1244
Cores de integridade	1245
Verificações de integridade do Elastic Load Balancing	1246
Verificações de integridade do ambiente de nível do operador e instância única	1247
Verificações adicionais	1247
CloudWatch Métricas da Amazon	1247
Monitoramento e relatório de integridade aprimorada	1249
Agente de integridade do Elastic Beanstalk	1252
Fatores que determinam a integridade de instâncias e ambientes	1253
Personalização de regra de verificação de integridade	1256
Funções de integridade aperfeiçoada	1256
Autorização de integridade aprimorada	1257
Eventos de integridade aprimorada	1258
Comportamento dos relatórios de integridade aprimorada durante as atualizações, as implantações e a escalabilidade	1259
Habilitar integridade aprimorada	1260
Console de integridade	1264
Status e cores de integridade	1270
Métricas de instância	1273
Regras de integridade aperfeiçoada	1276
CloudWatch	1281
Usuários da API	1290
Formato de log de integridade aprimorado	1292
Notificações e solução de problemas	1296
Gerenciar alarmes	1299
Ver histórico de alterações	1302
Visualizar eventos	1304
Monitorar instâncias	1306
Visualizar logs da instância	1309
Local de log em instâncias do Amazon EC2	1311
Local de log no Amazon S3	1312
Configurações de rotação de log no Linux	1313
Extensão da configuração padrão da tarefa de log	1314

Fazer streaming de arquivos de log para o Amazon CloudWatch Logs	1317
Integração de produtos da AWS	1319
Visão geral da arquitetura	1319
CloudFront	1320
CloudTrail	1321
Informações do Elastic Beanstalk no CloudTrail	1321
Noções básicas sobre entradas de arquivo de log do Elastic Beanstalk	1322
CloudWatch	1323
CloudWatch Logs	1324
Pré-requisitos para streaming de logs de instância para o CloudWatch Logs	1326
Como o Elastic Beanstalk configura o CloudWatch Logs	1327
Fazer streaming de logs de instâncias para o CloudWatch Logs	1333
Solucionar problemas de integração do CloudWatch Logs	1335
Streaming de integridade do ambiente	1336
EventBridge	1339
Monitorar um recurso Elastic Beanstalk com o EventBridge	1340
Eventos demonstrativos de padrões de eventos do Elastic Beanstalk	1343
Eventos demonstrativos do Elastic Beanstalk	1345
Mapeamento de campo de evento do Elastic Beanstalk	1347
AWS Config	1350
Configurar o AWS Config	1351
Configuração do AWS Config para registrar recursos do Elastic Beanstalk	1351
Visualizar os detalhes da configuração do Elastic Beanstalk no console do AWS Config ...	1352
Avaliar recursos do Elastic Beanstalk usando regras do AWS Config	1356
DynamoDB	1357
ElastiCache	1357
Amazon EFS	1358
Arquivos de configuração	1359
Sistemas de arquivos criptografados	1360
Aplicativos de exemplo	1360
Limpeza dos sistemas de arquivos	1361
IAM	1361
Perfis de instância	1362
Perfis de serviço	1366
Usar funções vinculadas a serviços	1380
Políticas de usuário	1393

Formato ARN	1401
Recursos e condições	1403
Controle de acesso com base em tags	1448
Exemplo de políticas gerenciadas	1452
Exemplo de políticas específicas de recursos	1456
Amazon RDS	1466
Amazon RDS na VPC padrão	1468
Amazon RDS no EC2 Classic	1475
Credenciais do Amazon RDS e Secrets Manager	1481
Limpar uma instância externa do Amazon RDS	1481
Amazon S3	1482
Conteúdo do bucket do Amazon S3 do Elastic Beanstalk	1482
Excluir objetos no bucket do Amazon S3 do Elastic Beanstalk	1483
Excluir o bucket do Amazon S3 do Elastic Beanstalk	1484
Amazon VPC	1485
VPC pública	1487
VPC pública/privada	1488
VPC privada	1488
Bastion hosts	1490
Amazon RDS	1495
VPC endpoints	1503
Configurar a máquina de desenvolvimento	1506
Criar uma pasta de projeto	1506
Configurar o controle da origem	1507
Configurar um repositório remoto	1507
Instalar a EB CLI	1508
Instalar o AWS CLI	1508
EB CLI	1509
Instalar a EB CLI	1510
Instalar a CLI do EB usando scripts de configuração	1511
Instalação manual	1511
Configurar a EB CLI	1521
Ignorar arquivos que usam .ebignore	1524
Usar perfis nomeados	1525
Implantar um artefato, em vez da pasta do projeto	1525
Definições de configuração e precedência	1525

Metadados da instância	1526
Conceitos básicos da EB CLI	1527
Eb create	1527
Eb status	1528
Eb health	1529
Eb events	1530
Eb logs	1530
Eb open	1530
Eb deploy	1531
Eb config	1532
Eb terminate	1532
CodeBuild	1533
Criar um aplicativo	1534
Criar e implantar o código do aplicativo	1534
Usar a EB CLI com o Git	1536
Associar ambientes do Elastic Beanstalk a ramificações Git	1537
Implantação de alterações	1537
Uso dos submódulos do Git	1538
Para atribuir tags do Git à versão do aplicativo	1539
CodeCommit	1539
Pré-requisitos	1540
Criar um repositório CodeCommit com a CLI do EB	1540
Implantar a partir do repositório CodeCommit	1541
Configurar ramificações e ambientes adicionais	1543
Usar um repositório CodeCommit existente	1544
Monitorar integridade	1545
Leitura da saída	1548
Visualização de integridade interativa	1551
Opções de visualização de integridade interativa	1552
Composição de ambientes	1553
Solução de problemas	1555
Solução de problemas de implantações	1556
Comandos da EB CLI	1559
eb abort	1560
eb appversion	1561
eb clone	1566

eb codesource	1568
eb config	1570
eb console	1580
eb create	1580
eb deploy	1597
eb events	1599
eb health	1601
eb init	1603
eb labs	1607
eb list	1607
eb local	1609
eb logs	1612
eb open	1617
eb platform	1618
eb printenv	1628
eb restore	1629
eb scale	1630
eb setenv	1631
eb ssh	1633
eb status	1636
eb swap	1637
eb tags	1639
eb terminate	1643
eb upgrade	1645
eb use	1646
Opções comuns	1647
EB CLI 2.6 (desativada)	1648
Diferenças da versão 3 da EB CLI	1648
Migrar para o CLI 3 do EB e o CodeCommit	1649
CLI da API do EB (desativada)	1650
Converter scripts da CLI da API do Elastic Beanstalk	1650
Segurança	1655
Proteção de dados	1656
Criptografia de dados	1657
Privacidade entre redes	1658
Identity and Access Management	1658

AWS políticas gerenciadas	1659
Registro em log e monitoramento	1672
Relatórios de integridade aprimorada	1672
Logs de instâncias do Amazon EC2	1672
Notificações do ambiente	1673
Alertas do Amazon CloudWatch	1673
Logs do AWS CloudTrail	1673
Depuração do AWS X-Ray	1673
Validação de conformidade	1673
Resiliência	1674
Segurança da infraestrutura	1675
Modelo de responsabilidade compartilhada	1675
Práticas recomendadas de segurança	1676
Práticas recomendadas de segurança preventiva	1676
Práticas recomendadas de segurança de detecção	1677
Solução de problemas	1679
Usar a ferramenta Systems Manager	1679
Orientações gerais	1681
Categorias	1681
Conectividade	1682
Criação de ambiente	1682
Implantações	1683
Integridade	1684
Configuração	1684
Docker	1685
Perguntas frequentes	1685
Recursos	1687
Aplicações de exemplo	1688
Histórico da plataforma	1689
Plataformas personalizadas	1689
Histórico do documento	1706
.....	mdccvii

O que é o AWS Elastic Beanstalk?

A Amazon Web Services (AWS) consiste em mais de cem serviços, cada um dos quais expõe uma área de funcionalidade. Embora a variedade de serviços ofereça flexibilidade para você decidir como vai gerenciar a infraestrutura da AWS, pode ser um desafio descobrir quais serviços usar e como provisioná-los.

Com o Elastic Beanstalk, é possível implantar e gerenciar rapidamente aplicações na Nuvem AWS sem se preocupar com a infraestrutura que as executa. O Elastic Beanstalk reduz a complexidade de gerenciamento sem restringir as escolhas nem o controle. Basta fazer upload da aplicação e o Elastic Beanstalk automaticamente gerencia os detalhes de provisão de capacidade, balanceamento de carga, escalabilidade e monitoramento do status da aplicação.

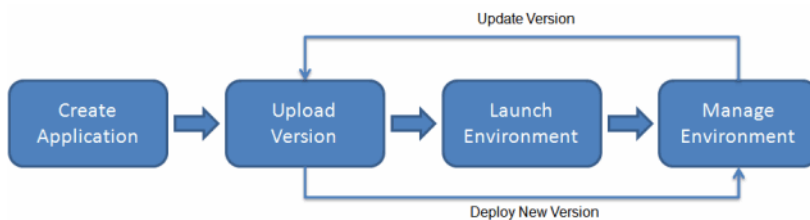
O Elastic Beanstalk é compatível com aplicações desenvolvidas em Go, Java, .NET, Node.js, PHP, Python e Ruby. Quando a aplicação é implantada, o Elastic Beanstalk constrói a versão da plataforma suportada selecionada e provisiona um ou mais recursos da AWS, como instâncias do Amazon EC2, para executar a aplicação.

É possível interagir com o Elastic Beanstalk usando o console do Elastic Beanstalk, a AWS Command Line Interface (AWS CLI), ou eb, uma CLI de alto nível projetada especificamente para o Elastic Beanstalk.

Para saber mais sobre como implantar uma aplicação Web demonstrativa usando o Elastic Beanstalk, consulte [Conceitos básicos da AWS: Implantar uma aplicação Web](#).

Também é possível executar a maioria das tarefas de implantação, como alterar o tamanho da frota de instâncias do Amazon EC2 ou monitorar a aplicação, diretamente da interface da Web do Elastic Beanstalk (console).

Para usar o Elastic Beanstalk, crie uma aplicação, faça upload de uma versão dela na forma de um pacote de origem (por exemplo, arquivo Java .war) no Elastic Beanstalk e forneça algumas informações sobre a aplicação. O Elastic Beanstalk inicia automaticamente um ambiente e cria e configura os recursos da AWS necessários para iniciar o código. Depois que o ambiente for iniciado, você poderá gerenciá-lo e implantar novas versões do aplicativo. O diagrama a seguir ilustra o fluxo de trabalho do Elastic Beanstalk.



Depois que você criar e implantar a aplicação, as informações sobre ela, incluindo métricas, eventos e status do ambiente, ficarão disponíveis no console do Elastic Beanstalk, nas APIs ou nas interfaces de linhas de comando, inclusive a AWS CLI unificada.

Definição de preços

Não há custo adicional para o Elastic Beanstalk. Você paga apenas pelos recursos subjacentes da AWS consumidos pela aplicação. Para obter detalhes sobre definição de preço, consulte a [página de detalhes do serviço Elastic Beanstalk](#).

Para onde ir em seguida

Este guia contém informações conceituais sobre o serviço da Web do Elastic Beanstalk, bem como informações sobre como usar o serviço para implantar aplicações Web. Seções separadas descrevem como usar o console do Elastic Beanstalk, as ferramentas de Interface de Linhas de Comando (CLI) e a API para implantar e gerenciar seus ambientes do Elastic Beanstalk. Este guia também documenta como o Elastic Beanstalk é integrado a outros serviços fornecidos pela Amazon Web Services.

Recomendamos que você leia primeiro [Conceitos básicos do Elastic Beanstalk](#) para saber como começar a usar o Elastic Beanstalk. Os Conceitos básicos orientam você na criação, visualização e atualização da aplicação do Elastic Beanstalk, além de editar e encerrar o ambiente do Elastic Beanstalk. Os Conceitos básicos também descrevem diferentes maneiras de acessar o Elastic Beanstalk.

Para saber mais sobre uma aplicação do Elastic Beanstalk e seus componentes, consulte as páginas a seguir.

- [Conceitos do Elastic Beanstalk](#)
- [Glossário de plataformas do Elastic Beanstalk](#)
- [Modelo de responsabilidade compartilhada para manutenção da plataforma do Elastic Beanstalk](#)

- [Política de suporte às plataformas do Elastic Beanstalk](#)

Conceitos básicos do Elastic Beanstalk

Para ajudar você a entender como AWS Elastic Beanstalk funciona, este tutorial explica como criar, explorar, atualizar e excluir um aplicativo do Elastic Beanstalk. Leva menos de uma hora para ser concluído.

Não há custo para usar o Elastic Beanstalk, AWS mas os recursos que ele cria para este tutorial estão ativos (e não são executados em um sandbox). Você incorre nas taxas de uso padrão para esses recursos até que os encerre no final deste tutorial. O total de cobranças geralmente é inferior a um dólar. Para obter informações sobre como minimizar as cobranças, consulte [nível gratuito da AWS](#).

Tópicos

- [Configurando: Crie uma AWS conta](#)
- [Etapa 1: criar um aplicativo de exemplo](#)
- [Etapa 2: explorar seu ambiente](#)
- [Etapa 3: implantar uma nova versão do aplicativo](#)
- [Etapa 4: configurar seu ambiente](#)
- [Etapa 5: limpar](#)
- [Próximas etapas](#)

Configurando: Crie uma AWS conta

Se você ainda não é AWS cliente, precisa criar uma AWS conta. A inscrição permite que você acesse o Elastic Beanstalk AWS e outros serviços necessários.

Inscreva-se para um Conta da AWS

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

Para se inscrever em um Conta da AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga as instruções on-line.

Parte do procedimento de inscrição envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

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

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando <https://aws.amazon.com/> e selecionando Minha conta.

Crie um usuário com acesso administrativo

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

Proteja seu Usuário raiz da conta da AWS

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

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

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

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

Crie um usuário com acesso administrativo

1. Habilitar o IAM Identity Center.

Para obter instruções, consulte [Habilitar AWS IAM Identity Center](#) no Guia do usuário do AWS IAM Identity Center .

2. No IAM Identity Center, conceda acesso administrativo a um usuário.

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

Faça login como usuário com acesso administrativo

- Para fazer login com seu usuário do Centro de Identidade do IAM, use a URL de login que foi enviada ao seu endereço de e-mail quando você criou o usuário do Centro do Usuário do IAM.

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

Atribua acesso a usuários adicionais

1. No IAM Identity Center, crie um conjunto de permissões que siga as melhores práticas de aplicação de permissões com privilégios mínimos.

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

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

Para obter instruções, consulte [Adicionar grupos](#) no Guia AWS IAM Identity Center do usuário.

Etapa 1: criar um aplicativo de exemplo

Nesta etapa, crie uma aplicação a partir de uma aplicação demonstrativa preexistente. O Elastic Beanstalk é compatível com plataformas para diferentes linguagens de programação, servidores de aplicações e contêineres do Docker. Você escolhe uma plataforma ao criar o aplicativo.

Criar um aplicativo e um ambiente

Para criar seu exemplo de aplicação, você usará o assistente do console Create application (Criar aplicação). Ele cria uma aplicação do Elastic Beanstalk e inicia um ambiente nela. Um ambiente é a coleção de AWS recursos necessários para executar o código do aplicativo.

Como criar um aplicativo de exemplo

1. Abra o [console do Elastic Beanstalk](#).

2. Escolha Criar aplicativo.
3. Em Application name (Nome da aplicação), insira **getting-started-app**.
4. Opcionalmente, adicione [tags de aplicativo](#).
5. Em Plataforma (Plataforma), escolha uma plataforma.
6. Escolha Next.
7. A página Configurar acesso ao serviço é exibida.
8. Escolha Usar um perfil de serviço existente em Perfil de serviço.
9. Em seguida, vamos nos concentrar na lista suspensa Perfil de instância do EC2. Os valores exibidos nessa lista suspensa podem variar, dependendo de a conta ter criado anteriormente um novo ambiente.

Escolha uma das opções a seguir com base nos valores exibidos na lista.

- Se `aws-elasticbeanstalk-ec2-role` for exibido na lista suspensa, selecione-o na lista suspensa Perfil de instância do EC2.
- Se outro valor for exibido na lista e for o perfil padrão de instância do EC2 para seus ambientes, selecione-o na lista suspensa Perfil de instância do EC2.
- Se a lista suspensa Perfil de instância do EC2 não mostrar nenhum valor para seleção, expanda o procedimento que se segue, Criar um perfil do IAM para ser o perfil de instância do EC2.

Conclua as etapas em Criar perfil do IAM para ser o perfil de instância do EC2 para criar um perfil do IAM que você possa selecionar posteriormente para ser o perfil de instância do EC2. Depois, retorne para esta etapa.

Agora que você criou um perfil do IAM e atualizou a lista, ele será exibido como uma opção na lista suspensa. Selecione o perfil do IAM que você acabou de criar na lista suspensa Perfil de instância do EC2.

10. Escolha Skip to Review (Pular para revisão) na página Configure service access (Configurar acesso a serviço).

Essa opção ignora as etapas opcionais.

11. A página Review (Revisão) exibe um resumo de todas as suas escolhas.

Escolha Save (Salvar) na parte inferior da página.

Criar um perfil do IAM para ser o perfil de instância do EC2

Configure service access [Info](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role

Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

Para criar um perfil do IAM para ser o perfil de instância do EC2


1. Escolha Visualizar detalhes de permissão. Essa opção exibe a lista suspensa Perfil de instância do EC2.

Uma janela modal intitulada Visualizar permissões de perfil de instância é exibida. Essa janela lista os perfis gerenciados que deverão ser anexados ao novo perfil de instância do EC2 que você criar. Ela também fornece um link para iniciar o console do IAM.

2. Escolha o link Console do IAM exibido na parte superior da janela.
3. No painel de navegação do console do IAM, escolha Roles (Perfis).
4. Selecione Criar função.
5. Em Tipo de entidade confiável, selecione Serviço da AWS .
6. Em Use case (Caso de uso), selecione EC2.
7. Escolha Next.
8. Anexe as políticas gerenciadas apropriadas. Role a janela modal Visualizar permissões de perfil de instância para ver as políticas gerenciadas. As políticas também estão listadas aqui:

- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

9. Escolha Next.
10. Insira um nome para a função.
11. (Opcional) Adicione tags à função.
12. Selecione Criar função.
13. Retorne à janela do console do Elastic Beanstalk que está aberta.
14. Feche a janela modal Visualizar permissões de perfil de instância.

 Important

Não feche a página do navegador que exibe o console do Elastic Beanstalk.

15. Escolha



(atualizar), ao lado da lista suspensa Perfil de instância do EC2.

Essa opção atualiza a lista suspensa para que o perfil que você acabou de criar seja exibido na lista suspensa.

Fluxo de trabalho do Elastic Beanstalk

Para implantar e executar o aplicativo de exemplo em AWS recursos, o Elastic Beanstalk executa as seguintes ações. Essas ações levam cerca de cinco minutos para serem concluídas.

1. Cria um aplicativo do Elastic Beanstalk chamado `getting-started-app`.
2. Lança um ambiente chamado `GettingStartedApp-env` com estes AWS recursos:
 - Uma instância do Amazon Elastic Compute Cloud (Amazon EC2) (máquina virtual)
 - Um grupo de segurança do Amazon EC2
 - Um bucket do Amazon Simple Storage Service (Amazon S3)
 - CloudWatch Alarmes da Amazon
 - Uma AWS CloudFormation pilha
 - Um nome de domínio

Para obter detalhes sobre esses AWS recursos, consulte [the section called “AWS recursos criados para o aplicativo de exemplo”](#).

3. Cria uma nova versão do aplicativo chamada Sample Application (Aplicativo de exemplo). Este é o arquivo da aplicação demonstrativa padrão do Elastic Beanstalk.
4. Implanta o código do aplicativo de exemplo no ambiente GettingStartedApp-env.

Durante o processo de criação do ambiente, o console rastreia o progresso e exibe eventos.

The screenshot displays the AWS Elastic Beanstalk console for an environment named 'Gettingstarted-env'. The environment is in a 'Healthy' state, indicated by a green checkmark and 'Ok' status. The environment ID is 'e-irkuacn9ny' and the application name is 'GettingStarted'. The platform is 'Node.js 16 running on 64bit Amazon Linux 2/5.6.3'. The console shows a list of 20 events, including the successful launch of the environment and the deployment of the application.

Environment overview

Health: ✔ Ok

Environment ID: e-irkuacn9ny

Domain: Gettingstarted-env.eba-w2pdx9as.us-east-1.elasticbeanstalk.com

Application name: GettingStarted

Platform

Platform: Node.js 16 running on 64bit Amazon Linux 2/5.6.3

Update

Running version: Sample Application

Events (20)

Time	Type	Details
January 8, 2023 19:40:13 (UTC-5)	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 46 seconds ago and took 2 minutes.
January 8, 2023 19:39:29 (UTC-5)	INFO	Successfully launched environment: Gettingstarted-env
January 8, 2023 19:39:28 (UTC-5)	INFO	Application available at Gettingstarted-env.eba-w2pdx9as.us-east-1.elasticbeanstalk.com.
January 8, 2023 19:39:13 (UTC-5)	INFO	Added instance [i-0b1530c3cabd58083] to your environment.
January 8, 2023 19:38:56 (UTC-5)	INFO	Instance deployment completed successfully.
January 8, 2023 19:38:28 (UTC-5)	INFO	Waiting for EC2 instances to launch. This may take a few minutes.
January 8, 2023 19:37:13 (UTC-5)	INFO	Environment health has transitioned to Pending. Initialization in progress (running for 20 seconds). There are no instances.
January 8, 2023 19:37:11 (UTC-5)	INFO	Created security group named: awseb-e-irkuacn9ny-stack-AWSEBSecurityGroup-1TQD00YHCNM7W
January 8, 2023 19:36:55 (UTC-5)	INFO	Created security group named: sg-0d8a4193f0512fe9a
January 8, 2023 19:36:55 (UTC-5)	INFO	Created target group named: arn:aws:elasticloadbalancing:us-east-1:164656829171:targetgroup/awseb-AWSEB-EURAPI3GVX2H/d33ef00e2dc5b0c8
January 8, 2023 19:36:34 (UTC-5)	INFO	Using elasticbeanstalk-us-east-1-164656829171 as Amazon S3 storage bucket for environment data.
January 8, 2023 19:36:33 (UTC-5)	INFO	createEnvironment is starting.

Quando todos os recursos são iniciados e as instâncias do EC2 que executam o aplicativo passam nas verificações de integridade, a integridade do ambiente muda para Ok. Agora você pode usar o site do seu aplicativo web.

AWS recursos criados para o aplicativo de exemplo

Quando você cria o aplicativo de exemplo, o Elastic Beanstalk cria os seguintes recursos: AWS

- Instância do EC2: uma máquina virtual do Amazon EC2 configurada para executar aplicações Web na plataforma de sua escolha.

Cada plataforma executa um conjunto diferente de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da Web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou nginx como um proxy reverso que processa tráfego da web na frente do aplicativo web, encaminha solicitações para ele, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic

Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

Etapa 2: explorar seu ambiente

Para ter uma visão geral do ambiente da sua aplicação do Elastic Beanstalk, use a página Environment overview (Visão geral do ambiente) no console do Elastic Beanstalk.

Como visualizar a visão geral do ambiente

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

A parte superior na página Environment overview (Visão geral do ambiente) mostra informações de nível superior sobre seu ambiente. Isso inclui o nome, o URL, o status de integridade atual, o nome da versão da aplicação implantada no momento e a versão da plataforma na qual a aplicação está sendo executada. Abaixo do painel de visão geral, você pode ver os cinco eventos mais recentes do ambiente na guia Events (Eventos). As outras guias exibem outros detalhes importantes sobre o ambiente.

Para saber mais sobre níveis de ambiente, plataformas, versões de aplicações e outros conceitos do Elastic Beanstalk, consulte [Conceitos](#).

The screenshot displays the AWS Elastic Beanstalk console interface for an environment named 'Gettingstarted-env'. The left sidebar shows navigation options like 'Applications', 'Environments', and 'Change history'. The main content area is divided into several sections: 'Environment overview' showing 'Health' as 'Ok', 'Environment ID' as 'e-irkuacn9ny', and 'Domain' as 'Gettingstarted-env.eba-w2pdx9as.us-east-1.elasticbeanstalk.com'; 'Platform' showing 'Node.js 16 running on 64bit Amazon Linux 2/5.7.0'; and 'Events (38)' with a table of recent events. The events table has columns for 'Time', 'Type', and 'Details'. Two events are visible, both with 'INFO' type, indicating successful configuration and environment updates.

Time	Type	Details
March 28, 2023 20:01:06 (UTC-4)	INFO	Environment health has transitioned from Info to Ok. Configuration update completed 47 seconds ago and took 14 minutes.
March 28, 2023 20:00:06 (UTC-4)	INFO	Environment update completed successfully.

Enquanto o Elastic Beanstalk AWS cria seus recursos e inicia seu aplicativo, o ambiente está em um estado. Pending As mensagens de status sobre eventos de execução são adicionadas continuamente à visão geral.

O Domain (Domínio), ou URL, do ambiente encontra-se na parte superior da página Environment overview (Visão geral do ambiente), abaixo de Health (Integridade) do ambiente. Este é o URL do aplicativo web no qual o ambiente é executado. Escolha esse URL para acessar a página de Parabéns do aplicativo de exemplo. O painel de navegação do lado esquerdo lista um link Go to environment (Ir para o ambiente) que inicia a mesma página de aplicação.

Também encontra-se listada no painel de navegação esquerdo a Configuration (Configuração), que mostra a página de Configuration overview (Visão geral da configuração). Essa página exibe um resumo dos valores das opções de configuração do ambiente, agrupados por categoria.

As guias exibidas na metade inferior da página contém informações mais detalhadas sobre o ambiente e fornecem acesso a recursos adicionais:

- **Events (Eventos):** mostra informações ou mensagens de erro do serviço do Elastic Beanstalk e de outros serviços cujos recursos são utilizados por esse ambiente.
- **Health (Integridade):** mostra o status e as informações detalhadas de integridade sobre as instâncias do Amazon EC2 que executam a aplicação.
- **Logs:** recupere e baixe os logs do Amazon EC2 no seu ambiente. Você pode recuperar logs completos ou as atividades recentes. Os logs recuperados ficam disponíveis por 15 minutos.

- **Monitoring (Monitoramento):** mostra as estatísticas do ambiente, por exemplo, a latência média e a utilização de CPU.
- **Alarms (Alarmes):** mostra os alarmes que você configurou para as métricas do ambiente. Você pode adicionar, modificar ou excluir alarmes nessa página.
- **Managed Updates (Atualizações gerenciadas):** mostra informações sobre as atualizações gerenciadas da plataforma e das substituições de instâncias já concluídas ou que ocorrerão em breve.
- **Tags:** mostra as tags de ambiente e permite que você as gerencie. Tags são pares de chave-valor que são aplicadas ao seu ambiente.

Note

O painel de navegação no lado esquerdo do console lista links com o mesmo nome das guias. Selecionar qualquer desses links exibirá o conteúdo da guia correspondente.

Etapa 3: implantar uma nova versão do aplicativo


Periodicamente, pode ser necessário implantar uma nova versão do aplicativo. Você pode implantar uma nova versão do aplicativo a qualquer momento, desde que não haja outras operações de atualização em andamento no ambiente.

A versão do aplicativo com a qual você iniciou este tutorial é chamada de Sample Application (Aplicativo de exemplo).

Para atualizar a versão do aplicativo

1. Faça download do aplicativo de exemplo que corresponde à plataforma do ambiente. Use um dos aplicativos a seguir.
 - Docker: [docker.zip](#)
 - Docker com vários contêineres — [2.zip docker-multicontainer-v](#)
 - Docker pré-configurado (Glassfish) — [1.zip docker-glassfish-v](#)
 - Go: [go.zip](#)
 - Corretto: [corretto.zip](#)
 - Tomcat: [tomcat.zip](#)


- .NET Core no Linux — [dotnet-core-linux.zip](#)
 - .NET Core — [dotnet-asp-windows.zip](#)
 - Node.js: [nodejs.zip](#)
 - PHP: [php.zip](#)
 - Python: [python.zip](#)
 - Ruby: [ruby.zip](#)
2. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
 3. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note


Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.


4. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
5. Escolha Choose file (Escolher arquivo) e faça upload do pacote de origem do aplicativo de exemplo que você obteve por download.

Upload and deploy ✕

 To deploy a previous version, go to the [Application Versions page](#).

Upload application:

 **Choose file**

File name : **java-tomcat-v3.zip** 

Version label:

Sample Application-2

► **Deployment Preferences**

The application version will be deployed using the **All at once** policy.

Current number of instances: **1**

Cancel **Deploy**

O console preenche automaticamente o Version label (Rótulo da versão) com um novo rótulo exclusivo. Se você digitar seu próprio rótulo de versão, certifique-se de que ele seja exclusivo.

6. Escolha Implantar.

Embora o Elastic Beanstalk implante o arquivo nas instâncias do Amazon EC2, você pode visualizar o status de implantação na visão geral do ambiente. O status de Environment Health (Integridade do ambiente) fica acinzentado enquanto a versão do aplicativo é atualizada. Quando a implantação estiver concluída, o Elastic Beanstalk executará uma verificação de integridade da aplicação. Quando o aplicativo responde à verificação de integridade, ela é considerada íntegra e o status retorna para verde. A visão geral do ambiente exibe a nova Running Version (Versão em execução): o nome fornecido como Version label (Rótulo da versão).

A nova versão da aplicação do Elastic Beanstalk também é carregada e adicionada à tabela de versões da aplicação. Para ver a tabela, escolha Versões do aplicativo abaixo getting-started-appno painel de navegação.

Etapa 4: configurar seu ambiente

Você pode configurar o ambiente para melhor atender ao aplicativo. Por exemplo, se você tiver uma aplicação que exija muito do computador, altere o tipo de instância do Amazon Elastic Compute Cloud (Amazon EC2) que a executa. Para aplicar alterações de configuração, o Elastic Beanstalk executa uma atualização de ambiente.

Algumas mudanças de configuração são simples e ocorrem de maneira rápida. Algumas alterações exigem a exclusão e a recriação de AWS recursos, o que pode levar alguns minutos. Quando você altera as configurações, o Elastic Beanstalk avisa sobre possíveis períodos de inatividade da aplicação.

Fazer uma alteração de configuração

Neste exemplo de alteração de configuração, você edita as configurações de capacidade do ambiente. Você configura um ambiente escalável e com carga balanceada que tenha entre duas e quatro instâncias do Amazon EC2 em seu grupo de Auto Scaling e, depois, verifica se a alteração ocorreu. O Elastic Beanstalk cria uma instância adicional do Amazon EC2, adicionando à única instância criada inicialmente. Depois, o Elastic Beanstalk associa as duas instâncias ao balanceador de carga do ambiente. Como resultado, a capacidade de resposta do aplicativo é aprimorada e sua disponibilidade é aumentada.

Como alterar a capacidade do seu ambiente

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar).
5. Feche a seção Instances (instâncias) para poder ver mais facilmente a seção Capacity (Capacidade). Na seção Auto Scaling Group (Grupo do Auto Scaling), altere Environment type (Tipo de ambiente) para Load balanced (Carga balanceada).
6. Na linha Instances (Instâncias), altere Max (Máximo) para **4** e altere Min (Mínimo) para **2**.

7. Para salvar as alterações, escolha Aplicar na parte inferior da página.
8. Um aviso informa que essa atualização substitui todas as suas instâncias atuais. Selecione a opção Confirmar.
9. A página Environment overview (Visão geral do ambiente) será exibida, mostrando a guia Events (Eventos).

A atualização do ambiente pode levar alguns minutos. Para descobrir se está concluído, procure o evento Successfully deployed new configuration to environment (Nova configuração implantada com êxito no ambiente) na lista de eventos. Isso confirma que a contagem de instância mínima de Auto Scaling foi definida como 2. O Elastic Beanstalk inicia automaticamente a segunda instância.

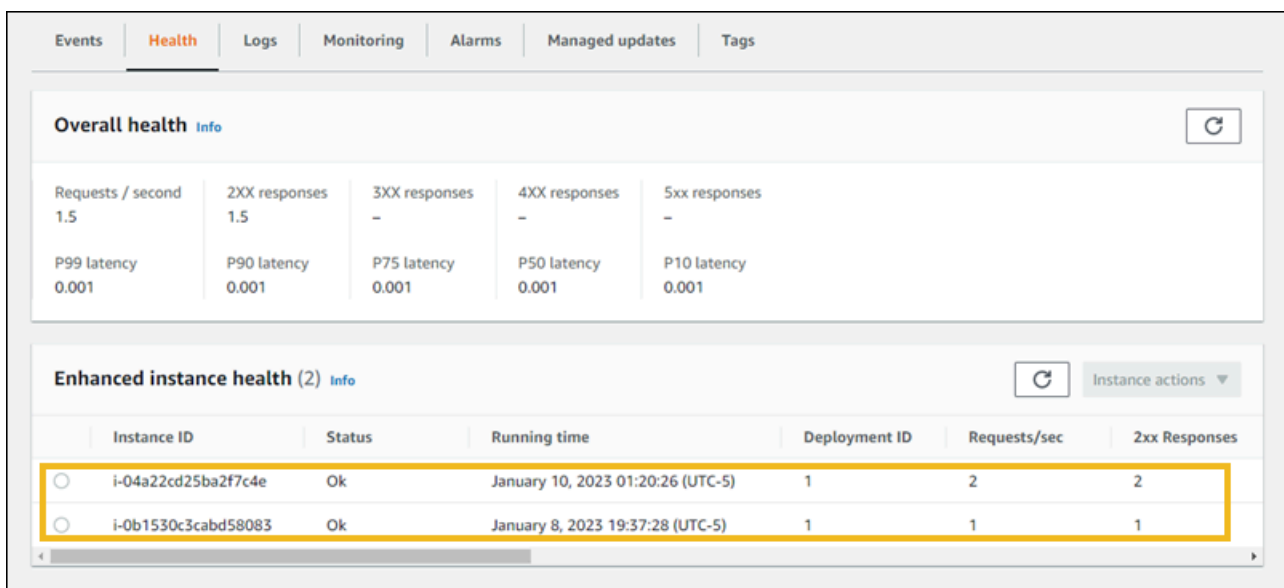
Verificar a alteração de configuração

Quando a atualização do ambiente estiver concluída e o ambiente estiver pronto, verifique a alteração.

Como verificar o aumento da capacidade

1. Escolha Health (Integridade) na lista de guias ou no painel de navegação esquerdo.
2. Veja a seção Enhanced instance health (Integridade aprimorada de instâncias).

Você pode ver que duas instâncias do Amazon EC2 estão listadas. A capacidade do ambiente aumentou para duas instâncias.



The screenshot displays the AWS Elastic Beanstalk Health page. The 'Overall health' section shows a table of performance metrics:

Requests / second	2XX responses	3XX responses	4XX responses	5xx responses
1.5	1.5	-	-	-

P99 latency	P90 latency	P75 latency	P50 latency	P10 latency
0.001	0.001	0.001	0.001	0.001

The 'Enhanced instance health (2)' section shows a table of instance details:

Instance ID	Status	Running time	Deployment ID	Requests/sec	2xx Responses
i-04a22cd25ba2f7c4e	Ok	January 10, 2023 01:20:26 (UTC-5)	1	2	2
i-0b1530c3cabd58083	Ok	January 8, 2023 19:37:28 (UTC-5)	1	1	1

Etapa 5: limpar

Parabéns! Você implantou com sucesso um aplicativo de amostra AWS na nuvem, fez o upload de uma nova versão e modificou sua configuração para adicionar uma segunda instância do Auto Scaling. Para garantir que você não seja cobrado por nenhum serviço que não esteja usando, exclua todas as versões do aplicativo e encerre o ambiente. Isso também exclui os AWS recursos que o ambiente criou para você.

Excluir o aplicativo e todos os recursos associados

1. Excluir todas as versões do aplicativo.
 - a. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
 - b. No painel de navegação, escolha Aplicativos e, em seguida, escolha getting-started-app.
 - c. No painel de navegação, encontre o nome do aplicativo e escolha Application versions (Versões do aplicativo).
 - d. Na página Application versions (Versões do aplicativo), selecione todas as versões do aplicativo que você deseja excluir.
 - e. Escolha Ações e, em seguida, escolha Excluir.
 - f. Ative Delete versions from Amazon S3 (Excluir versões do Amazon S3).
 - g. Escolha Delete (Excluir) e Done (Concluído).
2. Encerrar o ambiente.
 - a. No painel de navegação, escolha e, em seguida getting-started-app, escolha GettingStartedApp-env na lista de ambientes.
 - b. Escolha Actions e, em seguida, Terminate Environment.
 - c. Confirme que você deseja encerrar GettingStartedApp-env digitando o nome do ambiente e, em seguida, escolha Encerrar.
3. Exclua o getting-started-app aplicativo.
 - a. No painel de navegação, escolha o. getting-started-app
 - b. Escolha Ações e Excluir aplicativo.
 - c. Confirme que você deseja excluir getting-started-app digitando o nome do aplicativo e escolha Excluir.

Próximas etapas

Agora que você já sabe como criar uma aplicação e um ambiente do Elastic Beanstalk, recomendamos que você leia [Conceitos](#). Este tópico fornece informações sobre os componentes e a arquitetura do Elastic Beanstalk e descreve considerações de design importantes para a aplicação do Elastic Beanstalk.

Além do console do Elastic Beanstalk, é possível usar as ferramentas a seguir para criar e gerenciar ambientes do Elastic Beanstalk.

EB CLI

A EB CLI é uma ferramenta da linha de comando para criar e gerenciar ambientes. Para mais detalhes, consulte [Usar a Interface de Linhas de Comando do Elastic Beanstalk \(CLI do EB\)](#).

AWS SDK for Java

O AWS SDK for Java fornece uma API Java que você pode usar para criar aplicativos que usam serviços de AWS infraestrutura. Com o AWS SDK for Java, você pode começar em minutos com um único pacote disponível para download que inclui a biblioteca AWS Java, exemplos de código e documentação.

AWS SDK for Java Isso requer o J2SE Development Kit 5.0 ou posterior. Você pode fazer download do software Java mais recente em <http://developers.sun.com/downloads/>. O SDK também exige os pacotes de terceiros, o Apache Commons (Codec, HttpClient e Logging) e o Saxon-HE, que estão incluídos no diretório de terceiros do SDK.

Para obter mais informações, consulte [AWS SDK para Java](#).

AWS Toolkit for Eclipse

AWS Toolkit for Eclipse É um plug-in de código aberto para o Eclipse Java IDE. Você pode usá-lo para criar projetos web AWS Java pré-configurados com o e AWS SDK for Java, em seguida, implantar os aplicativos web no Elastic Beanstalk. O plug-in do Elastic Beanstalk é implementado na Eclipse Web Tools Platform (WTP). O toolkit oferece um modelo de aplicativo Web de amostra para log de viagem que demonstra o uso do Amazon S3 e Amazon SNS.

Para garantir que você tenha todas as dependências WTP, recomendamos começar com a distribuição do Java EE do Eclipse. Você pode fazer download dele em <http://eclipse.org/downloads/>.

Para obter mais informações sobre como usar o plugin do Elastic Beanstalk for Eclipse, consulte [AWS Toolkit for Eclipse](#). Para começar a criar a aplicação do Elastic Beanstalk usando o Eclipse, consulte [Criar e implantar aplicações Java no Elastic Beanstalk](#).

AWS SDK for .NET

AWS SDK for .NET Isso permite que você crie aplicativos que usam serviços de AWS infraestrutura. Com o AWS SDK for .NET, você pode começar em minutos com um único pacote disponível para download que inclui a AWS biblioteca.NET, exemplos de código e documentação.

Para obter mais informações, consulte [AWS SDK for .NET](#). Para as versões suportadas do .NET Framework e do Visual Studio, consulte o [Guia do Desenvolvedor AWS SDK for .NET](#).

AWS Toolkit for Visual Studio

Com o AWS Toolkit for Visual Studio plug-in, você pode implantar um aplicativo.NET existente no Elastic Beanstalk. Você também pode criar projetos usando os AWS modelos pré-configurados com o. AWS SDK for .NET

Para obter os pré-requisitos e informações de instalação, consulte [AWS Toolkit for Visual Studio](#). Para começar a criar a aplicação do Elastic Beanstalk usando o Visual Studio, consulte [Criar e implantar aplicações .NET no Elastic Beanstalk](#).

AWS SDK para JavaScript em Node.js

O AWS SDK do Node.js permite que você crie aplicativos com base nos serviços de AWS infraestrutura. JavaScript Com o AWS SDK do Node.js, você pode começar em minutos com um único pacote disponível para download que inclui a biblioteca AWS Node.js, exemplos de código e documentação. JavaScript

Para obter mais informações, consulte o [AWS SDK JavaScript em Node.js](#).

AWS SDK for PHP

AWS SDK for PHP Isso permite que você crie aplicativos com base nos serviços de AWS infraestrutura. Com o AWS SDK for PHP, você pode começar em minutos com um único pacote disponível para download que inclui a biblioteca AWS PHP, exemplos de código e documentação.

O AWS SDK for PHP requer PHP 5.2 ou posterior. Para obter mais detalhes de download, consulte <http://php.net/>.

Para obter mais informações, consulte o [AWS SDK for PHP](#)

AWS SDK for Python (Boto)

Com o AWS SDK for Python (Boto), você pode começar em minutos com um único pacote disponível para download que inclui a biblioteca AWS Python, exemplos de código e documentação. Você pode criar aplicativos em Python com base nas APIs que eliminam a complexidade da codificação diretamente nas interfaces de web service.

A all-in-one biblioteca fornece APIs de Python fáceis de usar para desenvolvedores que ocultam muitas das tarefas de nível inferior associadas à programação para a AWS nuvem, incluindo autenticação, novas tentativas de solicitação e tratamento de erros. O SDK fornece exemplos práticos em Python de como usar as bibliotecas para criar aplicativos.

Para informações sobre o Boto, o código demonstrativo, a documentação, as ferramentas e recursos adicionais, consulte o [Centro de desenvolvedores Python](#).

AWS SDK for Ruby

Você pode começar em minutos com um único pacote disponível para download, completo com a biblioteca AWS Ruby, exemplos de código e documentação. Você pode criar aplicativos em Ruby com base nas APIs que eliminam a complexidade da codificação diretamente nas interfaces de web services.

A all-in-one biblioteca fornece APIs Ruby amigáveis para desenvolvedores que ocultam muitas das tarefas de nível inferior associadas à programação para a AWS nuvem, incluindo autenticação, novas tentativas de solicitação e tratamento de erros. O SDK fornece exemplos práticos em Ruby de como usar as bibliotecas para criar aplicativos.

Para obter informações sobre o SDK, o código demonstrativo, a documentação, as ferramentas e recursos adicionais, consulte o [Centro do desenvolvedor para Ruby](#).

Conceitos do Elastic Beanstalk

O AWS Elastic Beanstalk permite que você gerencie todos os recursos que executam sua aplicação como ambientes. Veja a seguir alguns conceitos-chave do Elastic Beanstalk.

Aplicativo

Uma aplicação do Elastic Beanstalk é uma coleção lógica de componentes do Elastic Beanstalk, incluindo ambientes, versões e configurações de ambiente. No Elastic Beanstalk, uma aplicação é conceitualmente semelhante a uma pasta.

Versão do aplicativo

No Elastic Beanstalk, uma versão da aplicação se refere a uma iteração rotulada específica do código implantável de uma aplicação Web. Uma versão do aplicativo aponta para um objeto do Amazon Simple Storage Service (Amazon S3) que contém o código implantável, como um arquivo WAR Java. Uma versão do aplicativo faz parte de um aplicativo. Os aplicativos podem ter várias versões e cada uma delas é única. Em um ambiente em execução, é possível implantar qualquer versão do aplicativo já carregada no aplicativo ou fazer upload e implantar imediatamente uma nova versão do aplicativo. Você pode fazer upload de várias versões do aplicativo para testar diferenças entre uma versão do seu aplicativo Web e outra.

Ambiente

Um ambiente é um conjunto de recursos da AWS que executam uma versão da aplicação. Cada ambiente executa somente uma versão do aplicativo por vez, no entanto, você pode executar a mesma versão ou diferentes versões do aplicativo em vários ambientes ao mesmo tempo. Ao criar um ambiente, o Elastic Beanstalk provisiona os recursos necessários para executar a versão da aplicação especificada.

Nível do ambiente

Ao iniciar um ambiente do Elastic Beanstalk, primeiro escolha um nível de ambiente. O nível de ambiente designa o tipo de aplicação que o ambiente executa e determina quais recursos são

provisionados pelo Elastic Beanstalk para oferecer suporte a ele. Um aplicativo que atende a solicitações HTTP é executado em um [nível de ambiente de servidor da web](#). Um ambiente de back-end que extrai tarefas de uma fila do Amazon Simple Queue Service (Amazon SQS) é executado em uma [camada de ambiente de operador](#).

Configuração de ambientes

Uma configuração de ambiente identifica um conjunto de parâmetros e configurações que definem como um ambiente e seus recursos associados se comportam. Quando você atualiza as definições de configuração de um ambiente, o Elastic Beanstalk aplica automaticamente as alterações aos recursos existentes ou exclui e implanta novos recursos (dependendo do tipo de alteração).

Configuração salva

Uma configuração salva é um modelo que você pode usar como um ponto de partida para a criação de configurações exclusivas de ambiente. É possível criar e modificar as configurações salvas e aplicá-las a ambientes, usando o console do Elastic Beanstalk, a EB CLI, a AWS CLI ou a API. A API e a AWS CLI referem-se a configurações salvas como modelos de configuração.

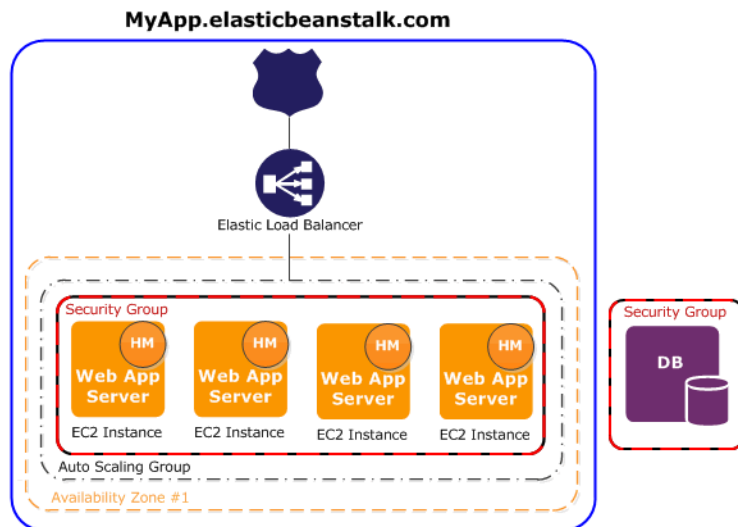
Plataforma

Uma plataforma é uma combinação de um sistema operacional, tempo de execução da linguagem de programação, servidor Web, servidor de aplicações e componentes do Elastic Beanstalk. Projete e segmente a aplicação Web para uma plataforma. O Elastic Beanstalk oferece uma variedade de plataformas para criação de aplicações.

Para obter mais detalhes, consulte [Plataformas do Elastic Beanstalk](#).

Ambientes de servidor da web

O diagrama a seguir mostra uma arquitetura demonstrativa do Elastic Beanstalk para um nível de ambiente de servidor Web e mostra como os componentes desse tipo de nível de ambiente funcionam em conjunto.



O ambiente é a essência do aplicativo. No diagrama, o ambiente é mostrado na linha sólida de nível superior. Quando você cria um ambiente, o Elastic Beanstalk provisiona os recursos necessários para executar a aplicação. Os recursos da AWS criados para um ambiente incluem um balanceador de carga elástico (ELB no diagrama), um grupo do Auto Scaling e uma ou mais instâncias do Amazon Elastic Compute Cloud (Amazon EC2).

Cada ambiente tem um CNAME (URL) que aponta para um load balancer. O ambiente tem um URL, como `myapp.us-west-2.elasticbeanstalk.com`. O alias desse URL no [Amazon Route 53](#) está definido como um URL do Elastic Load Balancing: algo como `abcdef-123456.us-west-2.elb.amazonaws.com`, usando um registro CNAME. O [Amazon Route 53](#) é um web service de Domain Name System (DNS) altamente disponível e dimensionável. Isso proporciona um roteamento seguro e confiável para a infraestrutura. O nome de domínio que você registrou com o seu provedor DNS encaminhará as solicitações para o CNAME.

O load balancer fica na frente das instâncias do Amazon EC2, que fazem parte de um grupo de Auto Scaling. O Auto Scaling do Amazon EC2 inicia automaticamente as instâncias adicionais do Amazon EC2 para acomodar a crescente carga na aplicação. Se diminuir a carga na aplicação, o Auto Scaling do Amazon EC2 interromperá as instâncias, mas sempre deixará pelo menos uma em execução.

A pilha de software em execução nas instâncias do Amazon EC2 depende do tipo de contêiner. O tipo de contêiner define a topologia da infraestrutura e a pilha de software a serem usadas nesse ambiente. Por exemplo, o ambiente do Elastic Beanstalk com um contêiner Apache Tomcat usa o sistema operacional Amazon Linux, o servidor Web Apache e o software Apache Tomcat. Para ver a lista de tipos de contêiner compatíveis, consulte [Plataformas compatíveis com Elastic Beanstalk](#).

Cada instância do Amazon EC2 que executa a aplicação usa um desses tipos de contêiner. Além disso, um componente de software chamado gerenciador de hosts (HM) é executado em cada instância do Amazon EC2. O gerenciador de host é responsável pelo seguinte:

- Implantação do aplicativo
- Agregação de eventos e métricas de recuperação por meio do console, API ou linha de comando
- Geração de eventos no nível de instância
- Monitoramento dos arquivos de log do aplicativo quanto a erros críticos
- Monitoramento do servidor de aplicativos
- Correção dos componentes de instância
- Rotação de arquivos de log do aplicativo e sua publicação no Amazon S3

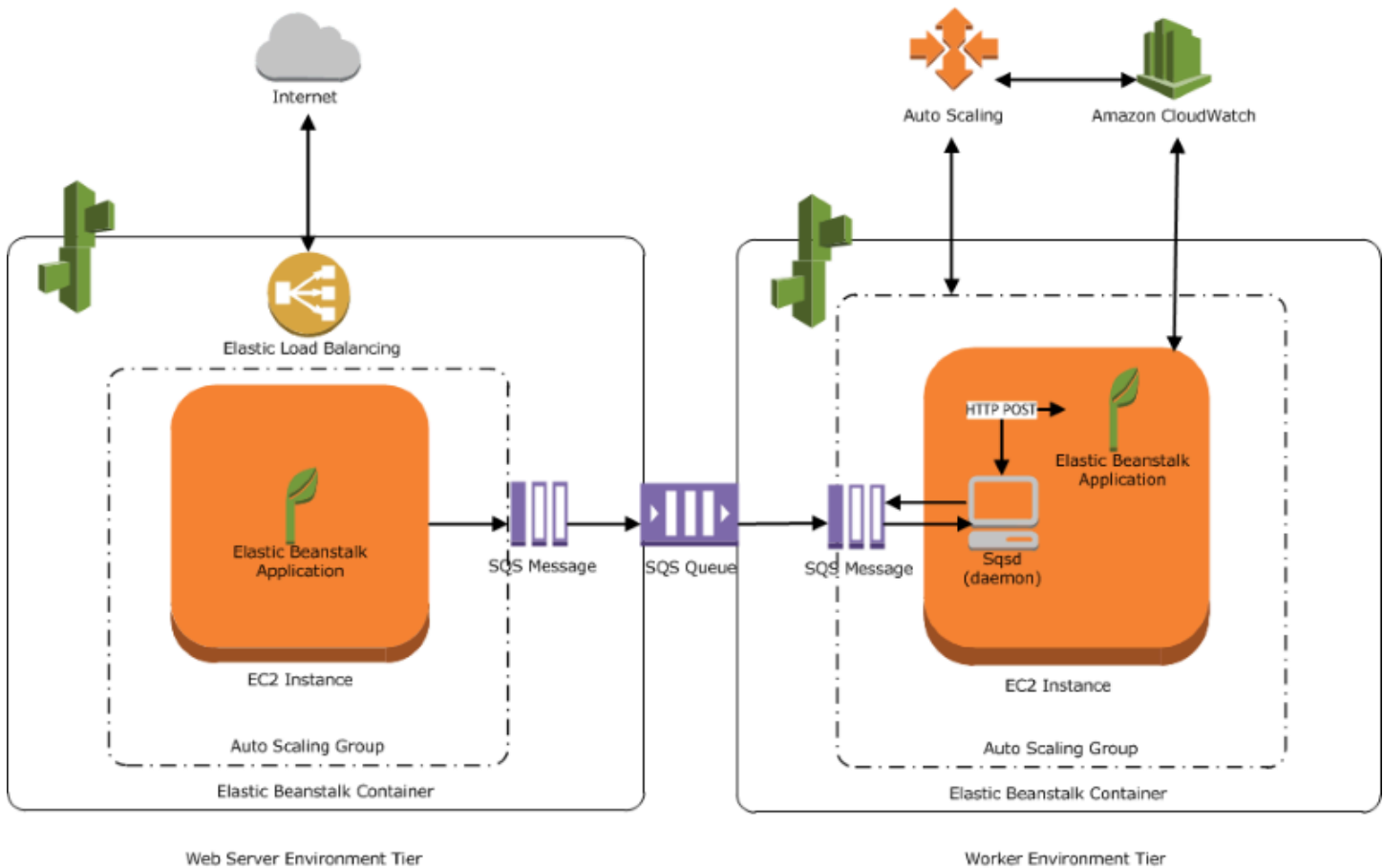
O gerenciador de hosts relata métricas, erros e eventos e o status da instância do servidor, que estão disponíveis no console do Elastic Beanstalk, nas APIs e nas CLIs.

As instâncias do Amazon EC2 mostradas no diagrama fazem parte de um grupo de segurança. Um security group define as regras de firewall para suas instâncias. Por padrão, o Elastic Beanstalk define um grupo de segurança, que permite que qualquer pessoa se conecte usando a porta 80 (HTTP). Você pode definir mais de um security group. Por exemplo, você pode definir um grupo de segurança para seu servidor de banco de dados. Para obter mais informações sobre os grupos de segurança do Amazon EC2 e como configurá-los para a aplicação do Elastic Beanstalk, consulte [Grupos de segurança](#).

Ambientes de operador

Os recursos da AWS criados para um nível de ambiente de operador incluem um grupo do Auto Scaling, uma ou mais instâncias do Amazon EC2 e uma função do IAM. Para o nível de ambiente de operador, o Elastic Beanstalk também cria e provisiona uma fila do Amazon SQS se você ainda não tiver uma. Quando você inicia um nível de ambiente de operador, o Elastic Beanstalk instala os arquivos de suporte necessários da sua linguagem de programação escolhida e um daemon em cada instância do EC2 no grupo de Auto Scaling. O daemon lê mensagens de uma fila do Amazon SQS. O daemon envia os dados de cada mensagem que lê para o aplicativo web em execução no ambiente de operador para processamento. Se você tiver várias instâncias no ambiente de operador, cada instância terá seu próprio daemon, mas todas farão a leitura da mesma fila do Amazon SQS.

O diagrama a seguir mostra os diferentes componentes e suas interações nos ambientes e serviços da AWS.



O Amazon CloudWatch é usado para monitoramento de alarmes e integridade. Para obter mais informações, acesse [Relatórios de integridade básica](#).

Para ver mais detalhes sobre como funciona o nível de ambiente de operador, consulte [Ambientes de operador do Elastic Beanstalk](#).

Considerações sobre design

Como as aplicações implantadas usando o AWS Elastic Beanstalk são executadas nos recursos do Nuvem AWS, é importante ter em mente vários fatores importantes para otimizar a aplicação: escalabilidade, segurança, armazenamento persistente, tolerância a falhas, entrega de conteúdo, atualizações e correções de software, e conectividade. Cada um desses fatores é abordado separadamente neste tópico. Para ver uma lista completa de whitepapers técnicos da AWS, abrangendo tópicos como arquitetura, segurança e economia, consulte os [whitepapers sobre computação em nuvem da AWS](#).

Escalabilidade

Quando estiver operando em um ambiente de hardware físico, não em um ambiente de nuvem, você pode abordar a escalabilidade de uma das duas maneiras. Você pode aumentar a escala na vertical ou aumentar a escala na horizontal. A abordagem de aumentar a escala na vertical exige que você invista em hardware poderoso, que possa suportar as demandas crescentes de seus negócios. A abordagem de aumentar a escala na horizontal exige que você adote um modelo de investimento distribuído. Dessa maneira, suas aquisições de hardware e de aplicações podem ser mais direcionadas, seus conjuntos de dados são federados e seu design é orientado a serviço. A abordagem de aumentar a escala vertical pode ser cara e ainda há o risco de que a demanda supere a capacidade. Nesse aspecto, a abordagem de aumentar a escala na horizontal geralmente é mais eficaz. Entretanto, quando usar essa abordagem, você deve ser capaz de prever a demanda em intervalos regulares e implantar a infraestrutura em blocos para atender a essa demanda. Por isso, essa abordagem, muitas vezes, pode resultar em capacidade não utilizada e exigir monitoramento cuidadoso.

Migrando para a nuvem, você pode alinhar bem a infraestrutura com a demanda, aproveitando a elasticidade da nuvem. A elasticidade ajuda a otimizar a aquisição e a liberação de recursos. Com ela, a escala da infraestrutura pode aumentar e diminuir rapidamente, de acordo com a flutuação de demanda. Para usá-la, configure a autoescalabilidade para aumentar ou diminuir a escala com base nas métricas dos recursos de ambiente (uso dos servidores ou E/S de rede, por exemplo). Por exemplo, você pode definir métricas como utilização do servidor ou E/S de rede. Pode usar a autoescalabilidade para adicionar automaticamente capacidade computacional sempre que o uso aumentar e removê-la sempre que o uso cair. Você pode publicar métricas do sistema (por exemplo, CPU, memória, E/S de disco e E/S de rede) no Amazon CloudWatch. Depois, você pode usar o CloudWatch para configurar alarmes para acionar as ações de autoescalabilidade ou enviar notificações com base nessas métricas. Para obter mais instruções sobre como configurar a autoescalabilidade, consulte [Grupo do Auto Scaling para o seu ambiente do Elastic Beanstalk](#).

Também recomendamos que você crie todas as aplicações do Elastic Beanstalk o mais sem estado possível, usando componentes com baixo acoplamento e tolerantes a falhas, cuja escala possa ser aumentada conforme necessário. Para obter mais informações sobre como projetar arquiteturas de aplicação escaláveis para a AWS, consulte [AWSFramework Well-Architected](#).

Segurança

A segurança na AWS é uma [responsabilidade compartilhada](#). A Amazon Web Services protege os recursos físicos do ambiente e garante que a nuvem seja um lugar seguro para a execução de

aplicações. Você é responsável pela segurança dos dados que chegam e saem do ambiente do Elastic Beanstalk e pela segurança da aplicação.

Configure o SSL para proteger os fluxos entre a aplicação e os clientes. Para configurar o SSL, você precisa de um certificado gratuito do AWS Certificate Manager (ACM). Se já tiver um certificado emitido por uma autoridade de certificação (CA) externa, você poderá usar o ACM para importar esse certificado. Caso contrário, importe-o usando a AWS CLI.

Se o ACM não estiver [disponível na Região da AWS](#), você poderá adquirir um certificado de uma CA externa, como a VeriSign ou a Entrust. Depois, use a AWS Command Line Interface (AWS CLI) para carregar um certificado autoassinado ou de terceiros e uma chave privada para o AWS Identity and Access Management (IAM). A chave pública do certificado autentica o seu servidor para o navegador. Também serve como base para criar a chave de sessão compartilhada que criptografa os dados nas duas direções. Para obter instruções sobre como criar, carregar e atribuir um certificado SSL para o ambiente, consulte [Configurar HTTPS para o seu ambiente do Elastic Beanstalk](#).

Quando você configura um certificado SSL para o ambiente, os dados são criptografados entre o cliente e o balanceador de carga do Elastic Load Balancing do ambiente. Por padrão, a criptografia é terminada no balanceador de carga, e o tráfego entre o balanceador de carga e as instâncias do Amazon EC2 não é criptografado.

Armazenamento persistente

As aplicações do Elastic Beanstalk são executadas nas instâncias do Amazon EC2 que não têm armazenamento local persistente. Quando as instâncias do Amazon EC2 são terminadas, o sistema de arquivos local não é salvo. Novas instâncias do Amazon EC2 começam com um sistema de arquivos padrão. Recomendamos que você configure sua aplicação para armazenar dados em uma origem dos dados persistente. A AWS oferece vários serviços de armazenamento persistente que você pode usar em sua aplicação. Veja-os na tabela a seguir.

Serviços de armazenam ento	Documentação do serviço	Integração do Elastic Beanstalk
Amazon S3	Documentação do Amazon Simple Storage Service	Usar o Elastic Beanstalk com o Amazon S3
Amazon Elastic File System	Documentação do Amazon Elastic File System	Usar o Elastic Beanstalk com o Amazon Elastic File System

Serviços de armazenam ento	Documentação do serviço	Integração do Elastic Beanstalk
Amazon Elastic Block Store	Amazon Elastic Block Store Guia de recursos: Elastic Block Store	
Amazon DynamoDB	Documentação do Amazon DynamoDB	Usar o Elastic Beanstalk com o Amazon DynamoDB
Amazon Relational Database Service (RDS)	Documentação do Amazon Relational Database Service	Usar o Elastic Beanstalk com o Amazon RDS

Note

O Elastic Beanstalk cria um usuário webapp para você configurar como proprietário dos diretórios de aplicações nas instâncias do EC2. Para as versões da plataforma Amazon Linux 2 lançadas a partir de [3 de fevereiro de 2022](#), o Elastic Beanstalk atribui ao usuário webapp um valor de uid (id do usuário) e gid (id do grupo) de 900 para novos ambientes. Ele faz o mesmo para os ambientes existentes após uma atualização de versão da plataforma. Essa abordagem mantém uma permissão consistente de acesso ao armazenamento do sistema de arquivos permanente para o usuário webapp.

Na situação improvável de que outro usuário ou processo já esteja usando 900, o sistema operacional usa, por padrão, outro valor para o uid e o gid do usuário webapp. Execute o comando `id webapp` do Linux em suas instâncias do EC2 para verificar os valores uid e gid atribuídos ao usuário webapp.

Tolerância a falhas

Como regra geral, seja pessimista ao projetar a arquitetura para a nuvem. Aproveite a elasticidade que ela oferece. Sempre projete, implemente e implante visando a recuperação automatizada de falha. Use várias zonas de disponibilidade para suas instâncias do Amazon EC2 e para o Amazon RDS. As zonas de disponibilidade são conceitualmente semelhantes a datacenters lógicos. Use o Amazon CloudWatch para ter mais visibilidade da integridade da aplicação do Elastic Beanstalk e tomar as medidas apropriadas em caso de degradação da performance ou falha de hardware.

Defina as configurações de Auto Scaling para manter seu conjunto de instâncias do Amazon EC2 em um tamanho fixo, de modo que as instâncias não íntegras do Amazon EC2 sejam substituídas por novas. Se você estiver usando o Amazon RDS, defina o período de retenção de backups, para que o Amazon RDS possa fazer backups automatizados.

Entrega de conteúdo

Quando os usuários se conectam ao seu site, suas solicitações podem ser roteadas por meio de várias redes individuais. Como resultado, os usuários podem experimentar uma performance insatisfatória devido à alta latência. O Amazon CloudFront pode ajudar a reduzir os problemas de latência, distribuindo o conteúdo da Web, como imagens e vídeo, por uma rede de locais de borda em todo o mundo. Os usuários finais são roteados para o local da borda mais próximo, de modo que o conteúdo seja entregue com a melhor performance possível. O CloudFront também funciona perfeitamente com o Amazon S3, que armazena por muito tempo as versões originais e definitivas dos seus arquivos. Para obter mais informações sobre o Amazon CloudFront, consulte o [Guia do desenvolvedor do Amazon CloudFront](#).

Atualizações e correções de software

O AWS Elastic Beanstalk lança regularmente [atualizações de plataforma](#) para fornecer correções, atualizações de software e novos recursos. O Elastic Beanstalk oferece várias opções de como lidar com atualizações de plataforma. As [atualizações da plataforma gerenciadas](#) atualizam automaticamente o ambiente para a versão mais recente da plataforma durante uma janela de manutenção programada, enquanto sua aplicação está em manutenção. Em ambientes criados a partir de 25 de novembro de 2019 usando o console do Elastic Beanstalk, as atualizações gerenciadas são habilitadas por padrão sempre que possível. Você também pode iniciar atualizações manualmente usando o console do Elastic Beanstalk ou a CLI do EB.

Conectividade

O Elastic Beanstalk precisa ser capaz de se conectar a instâncias do ambiente para concluir as implantações. Na implantação de uma aplicação do Elastic Beanstalk em uma Amazon VPC, a configuração necessária para permitir a conectividade depende do tipo de ambiente da Amazon VPC que você cria:

- Para ambientes de instância única, nenhuma configuração adicional é necessária. Isso porque, com esses ambientes, o Elastic Beanstalk atribui a cada instância do Amazon EC2 um endereço de IP elástico público, o que permite que a instância se comunique diretamente com a Internet.

- Para os ambientes com balanceamento de carga e escaláveis em uma Amazon VPC com sub-redes públicas e privadas, faça o seguinte:
 - Crie um balanceador de carga na sub-rede pública para rotear o tráfego de entrada da Internet para as instâncias do Amazon EC2.
 - Crie um dispositivo de Tradução de Endereço de Rede (NAT) para rotear o tráfego de saída das instâncias do Amazon EC2 em sub-redes privadas para a Internet.
 - Crie regras de roteamento de entrada e de saída para as instâncias do Amazon EC2 dentro da sub-rede privada.
 - Se você estiver usando uma instância NAT, configure os grupos de segurança para a instância NAT e as instâncias do Amazon EC2 para habilitar a comunicação com a Internet.
- Para um ambiente escalável e com balanceamento de carga em uma Amazon VPC que tenha uma sub-rede pública, nenhuma configuração adicional é necessária. Isso porque, com esse ambiente, suas instâncias do Amazon EC2 são configuradas com um endereço IP público que permite que as instâncias se comuniquem com a Internet.

Para obter mais informações sobre como usar o Elastic Beanstalk com a Amazon VPC, consulte [Usar o Elastic Beanstalk com Amazon VPC](#).

Funções de serviço, perfis de instância e políticas de usuário

Quando você cria um ambiente, o AWS Elastic Beanstalk solicita o fornecimento das seguintes funções do AWS Identity and Access Management (IAM):

- [Service role](#) (Função de serviço): o Elastic Beanstalk assume uma função de serviço para usar outros Serviços da AWS em seu nome.
- [Instance profile](#) (Perfil da instância): o Elastic Beanstalk aplica o perfil das instâncias às instâncias em seu ambiente. Isso permite que ele faça o seguinte:
 - Recupere [versões da aplicação](#) do Amazon Simple Storage Service (Amazon S3).
 - Carregue logs para o Amazon S3.
 - Execute outras tarefas que variam dependendo da plataforma e do tipo de ambiente.

Função de serviço

Quando você cria um ambiente no console do Elastic Beanstalk ou usa a EB CLI do Elastic Beanstalk, as funções de serviço necessárias são criadas e recebem [políticas gerenciadas](#). Essas políticas incluem todas as permissões necessárias. Agora, suponha que a função de serviço já exista em sua conta e você crie um novo ambiente no console do Elastic Beanstalk ou usando a CLI do Elastic Beanstalk. Se isso acontecer, a função de serviço existente será atribuída automaticamente ao novo ambiente.

Perfil da instância

Se a sua conta da AWS não tiver um perfil de instância do EC2, você deverá criar um usando o serviço IAM. Depois, você poderá atribuir o perfil de instância do EC2 aos novos ambientes que criar. O assistente Criar ambiente fornece informações que guiam você pelo serviço IAM para que possa criar um perfil de instância do EC2 com as permissões necessárias. Depois de criar o perfil de instância, você pode retornar ao console para selecioná-lo como o perfil de instância do EC2 e continuar as etapas para criar seu ambiente.

Note

Anteriormente, o Elastic Beanstalk criava um perfil de instância padrão do EC2 denominado `aws-elasticbeanstalk-ec2-role` a primeira vez que uma conta da AWS criava um ambiente. Esse perfil de instância incluía as políticas gerenciadas padrão.

Se sua conta já tiver esse perfil de instância, ele permanecerá disponível para você atribuí-lo aos seus ambientes.

Porém, as diretrizes de segurança recentes da AWS não permitem que um serviço da AWS crie automaticamente perfis com políticas de confiança para outros serviços da AWS, neste caso o EC2. Por causa dessas diretrizes de segurança, o Elastic Beanstalk não cria mais um perfil de instância padrão `aws-elasticbeanstalk-ec2-role`.

Políticas de usuário

Além das funções que atribui ao seu ambiente, você também pode criar [políticas de usuário](#) e aplicá-las a usuários e grupos do IAM em sua conta. A aplicação de políticas de usuário permite que os usuários criem e gerenciem aplicações e ambientes do Elastic Beanstalk. O Elastic Beanstalk também fornece políticas gerenciadas para acesso total e acesso somente leitura. Para obter mais informações sobre essas políticas, consulte [the section called “Políticas de usuário”](#).

Perfis de instância e políticas de usuário adicionais

Você pode criar seus próprios perfis de instância e políticas de usuário para cenários avançados. Se suas instâncias precisarem acessar serviços que não estejam incluídos nas políticas padrão, você poderá criar uma nova política ou acrescentar políticas adicionais à política padrão. Se a política gerenciada for muito permissiva para o que você precisa, também é possível criar políticas de usuário mais restritivas. Para mais informações sobre permissões da AWS, consulte o [Guia do usuário do IAM](#).

Tópicos

- [Função de serviço do Elastic Beanstalk](#)
- [Perfil de instância do Elastic Beanstalk](#)
- [Política de usuário do Elastic Beanstalk](#)

Função de serviço do Elastic Beanstalk

Uma função de serviço é uma função do IAM que o console do Elastic Beanstalk assume ao chamar outros serviços em seu nome. Por exemplo, o Elastic Beanstalk usa uma função de serviço ao chamar as APIs do Amazon Elastic Compute Cloud (Amazon EC2), Elastic Load Balancing e Amazon EC2 Auto Scaling para coletar informações. A função de serviço que o Elastic Beanstalk usa é a função que você especificou ao criar o ambiente do Elastic Beanstalk.

Existem duas políticas gerenciadas que são anexadas ao perfil de serviço. Essas políticas fornecem as permissões que permitem que o Elastic Beanstalk acesse os recursos necessários da AWS para criar e gerenciar seus ambientes. Uma das políticas gerenciadas fornece permissões para [monitoramento avançado de integridade](#) e compatibilidade com o Amazon SQS no nível do operador e a outra fornece as permissões adicionais necessárias para [atualizações gerenciadas da plataforma](#).

AWSElasticBeanstalkEnhancedHealth

Essa política concede todas as permissões de que o Elastic Beanstalk precisa para monitorar a integridade do ambiente. Ela também inclui ações do Amazon SQS para permitir que o Elastic Beanstalk monitore a atividade das filas nos ambientes de operador.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:DescribeInstanceHealth",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetHealth",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:GetConsoleOutput",
        "ec2:AssociateAddress",
        "ec2:DescribeAddresses",
        "ec2:DescribeSecurityGroups",
        "sqs:GetQueueAttributes",
        "sqs:GetQueueUrl",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeAutoScalingInstances",
        "autoscaling:DescribeScalingActivities",
        "autoscaling:DescribeNotificationConfigurations",
        "sns:Publish"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

```
}
```

AWS**ElasticBeanstalkManagedUpdatesCustomerRolePolicy**

Esta política concede permissões ao Elastic Beanstalk para atualizar ambientes em seu nome a fim de realizar atualizações gerenciadas de plataforma.

Agrupamentos de permissão no nível de serviço

Esta política é agrupada em declarações com base no conjunto de permissões fornecidas.

- *ElasticBeanstalkPermissions*: esse grupo de permissões destina-se a acionar as ações de serviço do Elastic Beanstalk (APIs do Elastic Beanstalk).
- *AllowPassRoleToElasticBeanstalkAndDownstreamServices* - Esse grupo de permissões permite que qualquer função seja transferida para o Elastic Beanstalk e para outros serviços downstream, como o AWS CloudFormation.
- *ReadOnlyPermissions*: esse grupo de permissões destina-se a coletar informações sobre o ambiente em execução.
- **OperationPermissions*: os grupos com este padrão de nomenclatura destinam-se a acionar as operações necessárias para executar atualizações na plataforma.
- **BroadOperationPermissions*: os grupos com este padrão de nomenclatura destinam-se a acionar as operações necessárias para executar atualizações na plataforma. Eles também incluem permissões amplas para oferecer suporte a ambientes herdados.
- **TagResource*— Grupos com esse padrão de nomenclatura são para chamadas que usam as APIs tag-on-create para anexar tags em recursos que estão sendo criados em um ambiente Elastic Beanstalk.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ElasticBeanstalkPermissions",
      "Effect": "Allow",
      "Action": [
        "elasticbeanstalk:*"
      ],
      "Resource": "*"
    }
  ],
}
```

```
{
  "Sid": "AllowPassRoleToElasticBeanstalkAndDownstreamServices",
  "Effect": "Allow",
  "Action": "iam:PassRole",
  "Resource": "arn:aws:iam::*:role/*",
  "Condition": {
    "StringEquals": {
      "iam:PassedToService": [
        "elasticbeanstalk.amazonaws.com",
        "ec2.amazonaws.com",
        "ec2.amazonaws.com.cn",
        "autoscaling.amazonaws.com",
        "elasticloadbalancing.amazonaws.com",
        "ecs.amazonaws.com",
        "cloudformation.amazonaws.com"
      ]
    }
  }
},
{
  "Sid": "ReadOnlyPermissions",
  "Effect": "Allow",
  "Action": [
    "autoscaling:DescribeAccountLimits",
    "autoscaling:DescribeAutoScalingGroups",
    "autoscaling:DescribeAutoScalingInstances",
    "autoscaling:DescribeLaunchConfigurations",
    "autoscaling:DescribeLoadBalancers",
    "autoscaling:DescribeNotificationConfigurations",
    "autoscaling:DescribeScalingActivities",
    "autoscaling:DescribeScheduledActions",
    "ec2:DescribeAccountAttributes",
    "ec2:DescribeAddresses",
    "ec2:DescribeAvailabilityZones",
    "ec2:DescribeImages",
    "ec2:DescribeInstanceAttribute",
    "ec2:DescribeInstances",
    "ec2:DescribeKeyPairs",
    "ec2:DescribeLaunchTemplates",
    "ec2:DescribeLaunchTemplateVersions",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSnapshots",
    "ec2:DescribeSpotInstanceRequests",
    "ec2:DescribeSubnets",
```

```

        "ec2:DescribeVpcClassicLink",
        "ec2:DescribeVpcs",
        "elasticloadbalancing:DescribeInstanceHealth",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth",
        "logs:DescribeLogGroups",
        "rds:DescribeDBEngineVersions",
        "rds:DescribeDBInstances",
        "rds:DescribeOrderableDBInstanceOptions",
        "sns:ListSubscriptionsByTopic"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "EC2BroadOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersion",
        "ec2:CreateSecurityGroup",
        "ec2>DeleteLaunchTemplate",
        "ec2>DeleteLaunchTemplateVersions",
        "ec2>DeleteSecurityGroup",
        "ec2:DisassociateAddress",
        "ec2:ReleaseAddress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress"
    ],
    "Resource": "*"
},
{
    "Sid": "EC2RunInstancesOperationPermissions",
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "*",
    "Condition": {
        "ArnLike": {

```

```

        "ec2:LaunchTemplate": "arn:aws:ec2:*:*:launch-template/*"
    }
}
},
{
    "Sid": "EC2TerminateInstancesOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "ec2:TerminateInstances"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
        "StringLike": {
            "ec2:ResourceTag/aws:cloudformation:stack-id": [
                "arn:aws:cloudformation:*:*:stack/awseb-e-*",
                "arn:aws:cloudformation:*:*:stack/eb-*"
            ]
        }
    }
}
},
{
    "Sid": "ECSBroadOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "ecs:CreateCluster",
        "ecs:DescribeClusters",
        "ecs:RegisterTaskDefinition"
    ],
    "Resource": "*"
},
{
    "Sid": "ECSDeleteClusterOperationPermissions",
    "Effect": "Allow",
    "Action": "ecs:DeleteCluster",
    "Resource": "arn:aws:ecs:*:*:cluster/awseb-*"
},
{
    "Sid": "ASGOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "autoscaling:AttachInstances",
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:CreateLaunchConfiguration",
        "autoscaling:CreateOrUpdateTags",

```

```

        "autoscaling:DeleteLaunchConfiguration",
        "autoscaling:DeleteAutoScalingGroup",
        "autoscaling:DeleteScheduledAction",
        "autoscaling:DetachInstances",
        "autoscaling:DeletePolicy",
        "autoscaling:PutScalingPolicy",
        "autoscaling:PutScheduledUpdateGroupAction",
        "autoscaling:PutNotificationConfiguration",
        "autoscaling:ResumeProcesses",
        "autoscaling:SetDesiredCapacity",
        "autoscaling:SuspendProcesses",
        "autoscaling:TerminateInstanceInAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup"
    ],
    "Resource": [
        "arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/
awseb-e-*",
        "arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/
eb-*",
        "arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/awseb-
e-*",
        "arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/eb-*"
    ]
},
{
    "Sid": "CFNOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "cloudformation:*"
    ],
    "Resource": [
        "arn:aws:cloudformation:*:*:stack/awseb-*",
        "arn:aws:cloudformation:*:*:stack/eb-*"
    ]
},
{
    "Sid": "ELBOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "elasticloadbalancing:AddTags",
        "elasticloadbalancing:ApplySecurityGroupsToLoadBalancer",
        "elasticloadbalancing:ConfigureHealthCheck",
        "elasticloadbalancing>CreateLoadBalancer",
        "elasticloadbalancing>DeleteLoadBalancer",

```



```

        "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
        "elasticloadbalancing:DeregisterTargets",
        "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
        "elasticloadbalancing:RegisterTargets"
    ],
    "Resource": [
        "arn:aws:elasticloadbalancing:*:*:targetgroup/awseb-*",
        "arn:aws:elasticloadbalancing:*:*:targetgroup/eb-*",
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/awseb-*",
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/eb-*",
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/*/awseb-*/**",
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/*/eb-*/**"
    ]
},
{
    "Sid": "CWLogsOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogGroup",
        "logs>DeleteLogGroup",
        "logs:PutRetentionPolicy"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/elasticbeanstalk/*"
},
{
    "Sid": "S3ObjectOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
        "s3:PutObject",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionAcl"
    ],
    "Resource": "arn:aws:s3:::elasticbeanstalk-*/**"
},
{
    "Sid": "S3BucketOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "s3:GetBucketLocation",

```

```

        "s3:GetBucketPolicy",
        "s3:ListBucket",
        "s3:PutBucketPolicy"
    ],
    "Resource": "arn:aws:s3:::elasticbeanstalk-*"
},
{
    "Sid": "SNSOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "sns:CreateTopic",
        "sns:GetTopicAttributes",
        "sns:SetTopicAttributes",
        "sns:Subscribe"
    ],
    "Resource": "arn:aws:sns:*:*:ElasticBeanstalkNotifications-*"
},
{
    "Sid": "SQSOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "sqs:GetQueueAttributes",
        "sqs:GetQueueUrl"
    ],
    "Resource": [
        "arn:aws:sqs:*:*:awseb-e-*",
        "arn:aws:sqs:*:*:eb-*"
    ]
},
{
    "Sid": "CWPutMetricAlarmOperationPermissions",
    "Effect": "Allow",
    "Action": [
        "cloudwatch:PutMetricAlarm"
    ],
    "Resource": [
        "arn:aws:cloudwatch:*:*:alarm:awseb-*",
        "arn:aws:cloudwatch:*:*:alarm:eb-*"
    ]
},
{
    "Sid": "AllowECSTagResource",
    "Effect": "Allow",
    "Action": [

```

```
        "ecs:TagResource"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "ecs:CreateAction": [
                "CreateCluster",
                "RegisterTaskDefinition"
            ]
        }
    }
}
]
```

Você pode criar um ambiente Elastic Beanstalk com qualquer das abordagens a seguir. Cada seção descreve como a abordagem lida com o perfil de serviço.

Console do Elastic Beanstalk

Se você criar um ambiente usando o console do Elastic Beanstalk, o Elastic Beanstalk solicita a criação de uma função de serviço chamada `aws-elasticbeanstalk-service-role`. Quando criado por meio do Elastic Beanstalk, esse perfil inclui uma política de confiança que permite ao Elastic Beanstalk assumir o perfil de serviço. As duas políticas gerenciadas descritas anteriormente neste tópico também estão vinculadas ao perfil.

Interface de linha de comando do Elastic Beanstalk (CLI do EB)

Você pode criar um ambiente usando o comando [the section called “eb create”](#) da interface da linha de comando do Elastic Beanstalk (CLI do EB). Se você não especificar um perfil de serviço por meio da opção `--service-role`, o Elastic Beanstalk cria o mesmo perfil de serviço padrão `aws-elasticbeanstalk-service-role`. Se a função de serviço padrão já existir, o Elastic Beanstalk a usará para o novo ambiente. Quando criado por meio do Elastic Beanstalk, esse perfil inclui uma política de confiança que permite ao Elastic Beanstalk assumir o perfil de serviço. As duas políticas gerenciadas descritas anteriormente neste tópico também estão vinculadas ao perfil.

API do Elastic Beanstalk

Você pode criar um ambiente usando a ação `CreateEnvironment` da API do Elastic Beanstalk. Se você não especificar um perfil de serviço, o Elastic Beanstalk criará um perfil vinculado ao serviço de monitoramento. Esse é um tipo exclusivo de função de serviço predefinida pelo Elastic Beanstalk

para incluir todas as permissões que o serviço exige para chamar outros Serviços da AWS em seu nome. A função vinculada a serviço é associada à sua conta. O Elastic Beanstalk cria essa função uma só vez e a reutiliza ao criar ambientes adicionais. Você também pode usar o IAM para criar antecipadamente a função vinculada ao serviço de monitoramento para sua conta. Quando sua conta tem um perfil vinculado ao serviço de monitoramento, é possível usá-lo para criar um ambiente usando a API, o console ou a CLI do Elastic Beanstalk. Para obter instruções sobre como usar funções vinculadas a serviços com ambientes do Elastic Beanstalk, consulte [Usar funções vinculadas ao serviço para o Elastic Beanstalk](#).

Para obter mais informações sobre as funções de serviço, consulte [Gerenciar funções de serviço do Elastic Beanstalk](#).

Perfil de instância do Elastic Beanstalk

Um perfil de instância é um perfil do IAM aplicado às instâncias do Amazon EC2 que são iniciadas no ambiente do Elastic Beanstalk. Ao criar um ambiente do Elastic Beanstalk, você especifica o perfil de instância que é usado quando as instâncias do EC2 fazem o seguinte:

- Recuperam [versões de aplicações](#) do Amazon Simple Storage Service (Amazon S3)
- Gravam logs no Amazon S3
- Em [ambientes integrados do AWS X-Ray](#), fazem upload dos dados de depuração para o X-Ray
- Em ambientes do Docker gerenciados pelo ESC, coordenam implantações de contêiner com o Amazon Elastic Container Service (Amazon ECS)
- Em ambientes de operador, leem a partir de uma fila do Amazon Simple Queue Service (Amazon SQS)
- Em ambientes de operador, escolhem o líder com o Amazon DynamoDB
- Em ambientes de operador, publicam métricas de integridade de instância no Amazon CloudWatch

O Elastic Beanstalk fornece um conjunto de políticas gerenciadas que permitem que as instâncias do EC2 do ambiente realizem as operações necessárias. As políticas gerenciadas que se seguem são as necessárias para os casos de uso básicos.

- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

Essas políticas são associadas ao perfil de instância que você cria quando inicia um ambiente no console do Elastic Beanstalk pela primeira vez.

Se sua aplicação Web exigir acesso a outros Serviços da AWS adicionais, acrescente instruções ou políticas gerenciadas ao perfil da instância que permitam acesso a esses serviços.

Para obter mais informações sobre os perfis da instância, consulte [Gerenciar perfis de instância do Elastic Beanstalk](#).

Política de usuário do Elastic Beanstalk

A fim de evitar o uso de sua conta raiz ou o compartilhamento de credenciais, crie usuários do IAM para cada pessoa que use o Elastic Beanstalk. Como uma prática recomendada de segurança, conceda apenas as permissões para que esses usuários acessem os serviços e os recursos de que precisam.

O Elastic Beanstalk requer permissões não apenas para suas próprias ações de API, mas também para vários outros serviços da AWS. O Elastic Beanstalk usa permissões de usuário para iniciar recursos em um ambiente. Esses recursos incluem instâncias do EC2, um balanceador de carga do Elastic Load Balancing e um grupo do Auto Scaling. O Elastic Beanstalk também usa permissões de usuário para salvar logs e modelos no Amazon Simple Storage Service (Amazon S3), enviar notificações para o Amazon SNS, atribuir perfis de instância e publicar métricas no CloudWatch. O Elastic Beanstalk requer permissões do AWS CloudFormation para orquestrar implantações e atualizações de recursos. Ele também exige permissões do Amazon RDS para criar bancos de dados quando necessário e permissões do Amazon SQS para criar filas para ambientes de operador.

Para obter mais informações sobre políticas de usuário, consulte [Gerenciar políticas de usuário do Elastic Beanstalk](#).

Plataformas do Elastic Beanstalk

AWS Elastic Beanstalk fornece uma variedade de plataformas nas quais você pode criar seus aplicativos. Projete a aplicação Web para uma dessas plataformas, e o Elastic Beanstalk implanta o código na versão de plataforma selecionada para criar um ambiente de aplicação ativo.

O Elastic Beanstalk oferece plataformas para diferentes linguagens de programação, servidores de aplicações e contêineres do Docker. Algumas plataformas possuem várias versões simultaneamente compatíveis.

Tópicos

- [Glossário de plataformas do Elastic Beanstalk](#)
- [Modelo de responsabilidade compartilhada para manutenção da plataforma do Elastic Beanstalk](#)
- [Política de suporte às plataformas do Elastic Beanstalk](#)
- [Cronograma de lançamento da plataforma Elastic Beanstalk](#)
- [Plataformas compatíveis com Elastic Beanstalk](#)
- [Plataformas Linux do Elastic Beanstalk](#)
- [Implantar aplicações do Elastic Beanstalk a partir de contêineres do Docker](#)
- [Criar e implantar aplicações Go no Elastic Beanstalk](#)
- [Criar e implantar aplicações Java no Elastic Beanstalk](#)
- [Trabalhar com .NET Core no Linux](#)
- [Criar e implantar aplicações .NET no Elastic Beanstalk](#)
- [Implantar aplicações Node.js no Elastic Beanstalk](#)
- [Criar e implantar aplicações PHP no Elastic Beanstalk](#)
- [Como trabalhar com Python](#)
- [Criar e implantar aplicações Ruby no Elastic Beanstalk](#)

Glossário de plataformas do Elastic Beanstalk

Veja a seguir os termos-chave relacionados a plataformas do AWS Elastic Beanstalk e seu ciclo de vida.

Tempo de execução

Os programas de software de tempo de execução específicos de cada linguagem de programação (estrutura, bibliotecas, intérprete, vm, etc.) que são necessários para executar o código do aplicativo.

Componentes do Elastic Beanstalk

Componentes de software que o Elastic Beanstalk adiciona a uma plataforma para habilitar a funcionalidade Elastic Beanstalk. Por exemplo, o agente de integridade aprimorada é necessário para reunir e relatar informações de integridade.

Plataforma

Uma combinação de um sistema operacional (SO), tempo de execução, servidor Web, servidor de aplicações e componentes do Elastic Beanstalk. As plataformas fornecem componentes que estão disponíveis para executar seu aplicativo.

Versão da plataforma

Uma combinação de versões específicas de um sistema operacional (SO), tempo de execução, servidor Web, servidor de aplicações e componentes do Elastic Beanstalk. Crie um ambiente do Elastic Beanstalk com base em uma versão da plataforma e implanta a aplicação nele.

Uma plataforma tem um número de versão semântica do formulário X.Y.Z, em que X é a versão principal, Y é a versão secundária, e Z é a versão do patch.

Uma versão de plataforma pode estar em um dos seguintes estados:

- **Compatível:** uma versão de plataforma que consiste inteiramente em componentes compatíveis. Todos os componentes não atingiram o fim da vida útil (EOL), conforme designado por seus respectivos fornecedores (proprietários - AWS ou terceiros - ou comunidades). Eles recebem patches regulares ou atualizações secundárias de seus fornecedores. O Elastic Beanstalk disponibiliza versões de plataforma compatíveis para a criação do ambiente.
- **Retirada:** uma versão de plataforma com um ou mais componentes retirados, que atingiram o fim da vida útil, conforme designado por seus fornecedores. Versões retiradas da plataforma não estão disponíveis para uso em ambientes do Elastic Beanstalk para clientes novos ou existentes.

Para obter detalhes sobre componentes desativados, consulte [the section called “Política de suporte de plataforma”](#).

Ramificação da plataforma

Uma linha de versões de plataforma que compartilha versões específicas (geralmente principais) de alguns de seus componentes, como sistema operacional (SO), tempo de execução ou componentes do Elastic Beanstalk. Por exemplo: Python 3.6 em execução no Amazon Linux de 64 bits; IIS 10.0 em execução no Windows Server 2016 de 64 bits. Cada versão da plataforma sucessiva na ramificação é uma atualização para a anterior.

A versão mais recente da plataforma em cada ramificação da plataforma está disponível incondicionalmente para a criação do ambiente. Versões da plataforma anteriores na ramificação ainda são compatíveis: é possível criar um ambiente com base em uma versão de plataforma anterior se você a usou em um ambiente nos últimos 30 dias. Mas essas versões anteriores da plataforma não possuem os componentes mais atualizados e não são recomendadas para uso.

Uma ramificação de plataforma pode estar em um dos seguintes estados:

- **Compatível:** uma ramificação de plataforma atual. Ela consiste inteiramente em componentes compatíveis. Ela recebe atualizações contínuas da plataforma e é recomendada para uso em ambientes de produção. Para obter uma lista de ramificações de plataforma suportadas, consulte [Plataformas suportadas com o Elastic Beanstalk](#) no Guia Plataformas do AWS Elastic Beanstalk.
- **Beta:** uma ramificação da plataforma de pré-visualização e de pré-lançamento. Ela é de natureza experimental. Ela pode receber atualizações contínuas da plataforma por um tempo, mas não tem suporte a longo prazo. Uma ramificação de plataforma beta não é recomendada para uso em ambientes de produção. Use-a apenas para avaliação. Para obter uma lista de ramificações de plataforma beta, consulte [Versões de plataforma do Elastic Beanstalk em beta público](#) no Guia Plataformas do AWS Elastic Beanstalk.
- **Obsoleta:** uma ramificação de plataforma com um ou mais componentes defasados. Ela recebe atualizações contínuas da plataforma, mas não é recomendada para uso em ambientes de produção. Para obter uma lista de ramificações de plataforma defasadas, consulte [Versões de plataforma do Elastic Beanstalk programadas para desativação](#) no Guia Plataformas do AWS Elastic Beanstalk.
- **Retirada:** uma ramificação de plataforma com um ou mais componentes retirados. Ela não recebe mais atualizações de plataforma e não é recomendada para uso em ambientes de produção. As ramificações de plataforma retiradas não estão listadas no guia Plataformas do AWS Elastic Beanstalk. O Elastic Beanstalk não disponibiliza versões de ramificações de plataforma retiradas para criação de ambiente.

Um componente compatível não tem data de desativação programada pelo fornecedor (proprietário ou comunidade). O fornecedor pode ser a AWS ou um terceiro. Um componente defasado tem uma data de desativação programada pelo fornecedor. Um componente desativado atingiu o fim da vida útil (EOL) e não tem mais suporte pelo fornecedor. Para obter detalhes sobre componentes desativados, consulte [the section called “Política de suporte de plataforma”](#).

Se seu ambiente usar uma ramificação de plataforma defasada ou desativada, recomendamos atualizá-la para uma versão de plataforma em uma ramificação de plataforma compatível. Para obter mais detalhes, consulte [the section called “Atualizações da plataforma”](#).

Atualização da plataforma

Um lançamento de novas versões da plataforma que contêm as atualizações para alguns componentes da plataforma: sistema operacional, tempo de execução, servidor Web, servidor de aplicações e componentes do Elastic Beanstalk. As atualizações de plataforma seguem a taxonomia de versão semântica e podem ter vários níveis:

- **Atualização principal:** uma atualização que tenha alterações incompatíveis com as versões existentes da plataforma. Talvez seja necessário modificar o aplicativo para executar corretamente em uma nova versão principal. Uma atualização principal tem um novo número de versão de plataforma principal.
- **Atualização secundária:** uma atualização que adiciona funcionalidade compatível com a versão anterior de uma plataforma existente. Não é necessário modificar o aplicativo para executar corretamente em uma nova versão secundária. Uma atualização secundária tem um novo número de versão de plataforma secundária.
- **Atualização de patch:** uma atualização que consiste em versões de manutenção (correções de erros, atualizações de segurança e melhorias de performance) que são compatíveis com uma versão anterior da plataforma. Uma atualização de patch tem um novo número de versão de plataforma de patch.

Managed Updates

Um recurso do Elastic Beanstalk que aplica automaticamente patches e atualizações secundárias aos componentes do sistema operacional (SO), tempo de execução, servidor Web, servidor de aplicações e Elastic Beanstalk para uma versão de plataforma compatível com o Elastic Beanstalk. Uma atualização gerenciada aplica uma versão mais recente da plataforma na mesma ramificação da plataforma ao seu ambiente. Você pode configurar as atualizações gerenciadas para aplicar somente atualizações de patch ou atualizações de versão secundária e de patch. Você também pode desabilitar atualizações gerenciadas completamente.

Para obter mais informações, consulte [Atualizações gerenciadas de plataforma](#).

Modelo de responsabilidade compartilhada para manutenção da plataforma do Elastic Beanstalk

AWS e nossos clientes compartilham a responsabilidade de alcançar um alto nível de segurança e conformidade de componentes de software. Esse modelo compartilhado reduz os encargos operacionais.

Para obter detalhes, consulte o [Modelo de Responsabilidade AWS Compartilhada](#).

AWS Elastic Beanstalk ajuda você a cumprir sua parte do modelo de responsabilidade compartilhada, fornecendo um recurso de atualizações gerenciadas. Esse recurso aplica automaticamente atualizações secundárias e patches para uma versão da plataforma compatível do Elastic Beanstalk. Se uma atualização gerenciada falhar, o Elastic Beanstalk notificará você sobre a falha a fim de garantir que você esteja ciente disso e possa tomar medidas imediatas.

Para ter mais informações, consulte [Atualizações gerenciadas de plataforma](#).

Além disso, o Elastic Beanstalk faz o seguinte:

- Publica a [política de suporte da plataforma](#) e a programação de desativação para os próximos 12 meses.
- Libera atualizações de patch, secundárias e principais dos componentes do sistema operacional (SO), runtime, servidor de aplicações e servidor Web normalmente em até 30 dias após sua disponibilidade. O Elastic Beanstalk é responsável por criar atualizações para seus respectivos componentes que estão presentes em suas versões de plataforma compatíveis. Todas as outras atualizações vêm diretamente de seus fornecedores (proprietários ou da comunidade).

Anunciamos todas as atualizações para nossas plataformas suportadas em nossas [notas de versão](#) no guia de notas de versão AWS Elastic Beanstalk . Também fornecemos uma lista de todas as plataformas suportadas e seus componentes, juntamente com um histórico da plataforma, no guia Plataformas AWS Elastic Beanstalk . Para obter mais informações, consulte [Plataformas compatíveis](#).

Você é responsável por fazer o seguinte:

- Atualize todos os componentes que você controla (identificados como Cliente no [Modelo de Responsabilidade AWS Compartilhada](#)). Isso inclui garantir a segurança do aplicativo, seus dados e qualquer componente que o aplicativo exigir e que tenha sido obtido por download.
- Certifique-se de que os ambientes do Elastic Beanstalk estejam em execução em uma versão compatível da plataforma e migre qualquer ambiente em execução em uma versão retirada da plataforma para uma versão compatível.
- Resolva todos os problemas que surgirem em tentativas de atualização gerenciadas com falha e tente fazer a atualização novamente.
- Aplique patch ao sistema operacional, runtime, servidor de aplicações e servidor Web por conta própria se optou por não utilizar as atualizações gerenciadas do Elastic Beanstalk. Você pode fazer isso [aplicando as atualizações de plataforma manualmente](#) ou aplicar patch diretamente nos componentes em todos os recursos de ambiente relevantes.
- [Gerencie a segurança e a conformidade de todos AWS os serviços que você usa fora do Elastic Beanstalk de acordo com AWS o Modelo de Responsabilidade Compartilhada.](#)

Política de suporte às plataformas do Elastic Beanstalk

AWS Elastic Beanstalk fornece uma variedade de plataformas para execução de aplicativos AWS. O Elastic Beanstalk é compatível com ramificações de plataformas que ainda recebem atualizações secundárias e de patches contínuas de seus fornecedores (proprietários ou comunidade). Para obter uma definição de termos relacionados, consulte [Glossário de plataformas do Elastic Beanstalk](#).

Ramificações desativadas da plataforma

Quando um componente de uma ramificação de plataforma compatível é marcado como End of Life (EOL) por seu fornecedor, o Elastic Beanstalk marca a ramificação da plataforma como descontinuada. Os componentes de uma ramificação da plataforma incluem o seguinte: sistema operacional (SO), versão da linguagem de execução, servidor de aplicativos ou servidor web.

Depois que uma ramificação da plataforma é marcada como desativada, as seguintes políticas se aplicam:

- O Elastic Beanstalk deixa de fornecer atualizações de manutenção, incluindo atualizações de segurança.
- O Elastic Beanstalk não fornece mais suporte técnico para filiais desativadas da plataforma.

- O Elastic Beanstalk não disponibiliza mais a ramificação da plataforma para novos clientes do Elastic Beanstalk para implantações em novos ambientes. Há um período de carência de 90 dias a partir da data de retirada publicada para clientes existentes com ambientes ativos sendo executados em ramificações de plataforma desativadas.

Note

Uma ramificação de plataforma descontinuada não estará disponível no console do Elastic Beanstalk. No entanto, ele estará disponível por meio do AWS CLI EB CLI e da EB API para clientes que tenham ambientes existentes com base na ramificação descontinuada da plataforma. Os clientes existentes também podem usar os [consoles do ambiente Clone](#) e do ambiente [Rebuild](#).

Para obter uma lista das filiais da plataforma que estão programadas para serem desativadas, consulte o [Programação de ramificações desativadas da plataforma](#) tópico de programação da plataforma Elastic Beanstalk a seguir.

Para obter mais informações sobre o que esperar quando a ramificação da plataforma do seu ambiente for desativada, consulte [Perguntas frequentes sobre descontinuação de plataformas](#).

Após o período de carência de 90 dias

Nossa política para filiais de plataformas desativadas não remove o acesso aos ambientes nem exclui recursos. No entanto, os clientes existentes que executam um ambiente do Elastic Beanstalk em uma filial de plataforma desativada devem estar cientes dos riscos de fazer isso. Esses ambientes podem acabar em uma situação imprevisível, porque o Elastic Beanstalk não consegue fornecer atualizações de segurança, suporte técnico ou hotfixes para filiais desativadas da plataforma devido ao fato de o fornecedor marcar o EOL de seus componentes.

Por exemplo, uma vulnerabilidade de segurança prejudicial e crítica pode surgir em um ambiente sendo executado em uma ramificação de plataforma retirada. Ou uma ação da API do EB pode parar de funcionar no ambiente se, com o tempo, tornar-se incompatível com o serviço Elastic Beanstalk. Quanto mais tempo um ambiente baseado em uma ramificação de plataforma retirada permanecer ativo, maior a chance de que esses tipos de riscos ocorram. Para continuar a obter os benefícios de importantes melhorias de funcionalidade, performance e segurança oferecidas por fornecedores de componentes nas versões mais recentes, é altamente recomendável atualizar todos os ambientes do Elastic Beanstalk para uma versão compatível da plataforma.

Se seu aplicativo tiver problemas ao ser executado em uma ramificação de plataforma descontinuada e você não conseguir migrá-lo para uma plataforma compatível, precisará considerar outras alternativas. Soluções alternativas incluem o encapsulamento da aplicação em uma imagem do Docker para executá-la como um contêiner do Docker. Isso permitiria que um cliente usasse qualquer uma de nossas soluções Docker, como nossas plataformas Docker Elastic Beanstalk AL2023/AL2 ou outros serviços baseados em Docker, como Amazon ECS ou Amazon EKS. As alternativas que não são do Docker incluem nosso AWS CodeDeploy serviço, que permite a personalização completa dos tempos de execução que você deseja.

Cronograma de lançamento da plataforma Elastic Beanstalk

Para garantir que seus aplicativos sejam executados em um ambiente seguro e suportado, o Elastic Beanstalk fornece atualizações regulares para suas plataformas gerenciadas, conforme descrito no tópico anterior. [Modelo de responsabilidade compartilhada](#) Além do lançamento mensal de novas versões de ramificações da plataforma, nossa manutenção de lançamentos também inclui os seguintes processos:

- Lançamento de novas ramificações de plataforma — elas normalmente introduzem uma nova versão principal de uma linguagem de tempo de execução, sistema operacional ou servidor de aplicativos.
- Aposentadoria das filiais da plataforma — Devemos aposentar uma filial da plataforma quando um de seus componentes atingir o fim da vida útil (EOL). Para obter mais informações sobre nossa política para agências aposentadas, consulte [Política de suporte às plataformas do Elastic Beanstalk](#)

Tópicos

- [Recursos de planejamento](#)
- [Próximos lançamentos de filiais da plataforma](#)
- [Programação de ramificações desativadas da plataforma](#)
- [Histórico de ramificações de plataforma retiradas](#)
- [Histórico do servidor e do sistema operacional descontinuados](#)

Recursos de planejamento

Além dos cronogramas a seguir, há recursos adicionais que podem ajudá-lo a planejar a manutenção e o suporte para seu aplicativo executado em uma plataforma do Elastic Beanstalk. Para obter mais informações sobre os componentes da nossa plataforma, datas importantes e anúncios de lançamento, consulte os seguintes recursos:

- [AWS Elastic Beanstalk Guia de plataformas](#) — Este guia fornece uma lista detalhada de componentes para cada uma de nossas ramificações de plataforma. Ele também fornece um histórico da plataforma por data de lançamento com os mesmos detalhes. Este guia pode informá-lo quando componentes específicos da ramificação da sua plataforma foram alterados. Se seu aplicativo começar a se comportar de forma diferente, você também poderá cruzar a data da ocorrência no guia de plataformas para ver se houve alguma alteração na plataforma que possa ter afetado seu aplicativo.
- [AWS Elastic Beanstalk Notas de lançamento](#) — Nossas notas de lançamento anunciam todos os lançamentos da nossa plataforma, tanto menores quanto principais. Isso inclui nossas atualizações mensais de plataforma, lançamentos de segurança, hotfixes e anúncios de aposentadoria. Você pode assinar nossos feeds RSS na documentação das notas de lançamento.

Próximos lançamentos de filiais da plataforma

A tabela a seguir lista as próximas filiais da plataforma Elastic Beanstalk e sua data prevista de lançamento. Essas datas são provisórias e estão sujeitas a alterações.

Versão em tempo de execução//ramificação da plataforma	Sistema operacional	Data de lançamento prevista
.NET 8	Amazon Linux 2023	Junho de 2024
PHP 8.3	Amazon Linux 2023	Setembro de 2024
Python 3.12	Amazon Linux 2023	Setembro de 2024
Ruby 3.3	Amazon Linux 2023	Novembro de 2024

Programação de ramificações desativadas da plataforma

A tabela a seguir lista as filiais da plataforma Elastic Beanstalk que estão programadas para serem descontinuadas, pois alguns de seus componentes estão chegando ao fim da vida útil (EOL).

Para obter uma lista mais detalhada das ramificações da plataforma em desativação que inclui seus componentes específicos, consulte a [desativação das versões da plataforma](#) no guia de AWS Elastic Beanstalk plataformas.

Versão em tempo de execução//ramificação da plataforma	Sistema operacional	Data de aposentadoria prevista
Python 3.7 AL2	Amazon Linux 2	30 de setembro de 2024
Node.js 14 AL2	Amazon Linux 2	30 de setembro de 2024
Node.js 16 AL2	Amazon Linux 2	30 de setembro de 2024
Ruby 2.7 AL2	Amazon Linux 2	30 de setembro de 2024
Ruby 3.0 AL2	Amazon Linux 2	30 de setembro de 2024
Corretto 8 with Tomcat 8.5 AL2	Amazon Linux 2	30 de setembro de 2024
Corretto 11 with Tomcat 8.5 AL2	Amazon Linux 2	30 de setembro de 2024
PHP 8.0 AL2	Amazon Linux 2	30 de setembro de 2024

Histórico de ramificações de plataforma retiradas

As tabelas a seguir listam as ramificações da plataforma Elastic Beanstalk que já estão desativadas. Você pode ver um histórico detalhado dessas ramificações da plataforma e seus componentes no guia [Histórico da plataforma](#) do AWS Elastic Beanstalk Platforms.

Amazon Linux 2 (AL2)


Versão em tempo de execução// ramificação da plataforma	Data de retirada		
Corretto 11 with Tomcat 7 AL2	29 de junho de 2022		
Corretto 8 with Tomcat 7 AL2	29 de junho de 2022		
Node.js 12 AL2	23 de dezembro de 2022		
Node.js 10 AL2	29 de junho de 2022		
PHP 7.4 AL2	9 de junho de 2023		
PHP 7.3 AL2	29 de junho de 2022		
PHP 7.2 AL2	29 de junho de 2022		
Ruby 2.6 AL2	23 de dezembro de 2022		
Ruby 2.5 AL2	29 de junho de 2022		

Amazon Linux AMI (AL1)

Versão em tempo de execução// ramificação da plataforma	Data de retirada		
Single Container Docker	18 de julho de 2022		

Versão em tempo de execução// ramificação da plataforma	Data de retirada		
Multicontainer Docker	18 de julho de 2022		
Preconfigured Docker - GlassFish 5.0 with Java 8	18 de julho de 2022		
Go 1	18 de julho de 2022		
Java 8	18 de julho de 2022		
Java 7	18 de julho de 2022		
Java 8 with Tomcat 8.5	18 de julho de 2022		
Java 7 with Tomcat 7	18 de julho de 2022		
Node.js	18 de julho de 2022		
PHP 7.2 - 7.3	18 de julho de 2022		
Python 3.6	18 de julho de 2022		
Ruby 2.4, 2.5, 2.6 with Passenger	18 de julho de 2022		
Ruby 2.4, 2.5, 2.6 with Puma	18 de julho de 2022		
Go 1.3–1.10	31 de outubro de 2020		

Versão em tempo de execução// ramificação da plataforma	Data de retirada		
Java 6	31 de outubro de 2020		
Node.js 4.x–8.x	31 de outubro de 2020		
PHP 5.4–5.6	31 de outubro de 2020		
PHP 7.0–7.1	31 de outubro de 2020		
Python 2.6, 2.7, 3.4	31 de outubro de 2020		
Ruby 1.9.3	31 de outubro de 2020		
Ruby 2.0–2.3	31 de outubro de 2020		

 Note

Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para ter mais informações, consulte [Perguntas frequentes sobre descontinuação de plataformas](#).

Windows Server

Versão em tempo de execução// ramificação da plataforma	Data de retirada		
IIS 8.5 em execução no	29 de junho de 2022		

Versão em tempo de execução// ramificação da plataforma	Data de retirada		
Windows Server (e Core) de 64 bits 2012 R2 versão 0.1.0			
IIS 8.5 em execução no Windows Server (e Core) de 64 bits 2012 R2 versão 1.2.0	29 de junho de 2022		
IIS 10.0 em execução no Windows Server de 64 bits 2016 (e Core) versão 1.2.0	29 de junho de 2022		
ISS 8 em execução no Windows Server de 64 bits 2012 de versão de plataforma R1	22 de junho de 2022		
IIS 8 em execução no Windows Server de 64 bits 2012 R1 versão 0.1.0	22 de junho de 2022		

Versão em tempo de execução// ramificação da plataforma	Data de retirada		
IIS 8 em execução no Windows Server de 64 bits 2012 R1 versão 1.2.0	22 de junho de 2022		

Note

Para obter mais informações sobre a descontinuação das ramificações da plataforma Windows 2012 R2, consulte [Windows Server 2012 R2 platform branches retired](#) nas AWS Elastic Beanstalk Release Notes.

Histórico do servidor e do sistema operacional descontinuados

As tabelas a seguir fornecem um histórico dos sistemas operacionais, servidores de aplicativos e servidores web que não são mais suportados pelas plataformas do Elastic Beanstalk. Todas as ramificações da plataforma que utilizavam esses componentes agora estão desativadas. As datas refletem a data de aposentadoria da última filial da plataforma Elastic Beanstalk que incluiu o componente.

Sistema operacional

Versão do SO	Data de desativação da plataforma		
Windows Server 2012 R2 running IIS 8.5	4 de dezembro de 2023		

Versão do SO	Data de desativação da plataforma		
Windows Server Core 2012 R2 running IIS 8.5	4 de dezembro de 2023		
Amazon Linux AMI (AL1)	18 de julho de 2022		
Windows Server 2012 R1	22 de junho de 2022		
Windows Server 2008 R2	28 de outubro de 2019		

Servidores de aplicações

Versão do servidor do aplicativo	Data de desativação da plataforma		
Tomcat 7	29 de junho de 2022 para plataformas Amazon Linux 2 (AL2) 18 de junho de 2022 para plataformas Amazon Linux AMI (AL1)		
Tomcat 8	31 de outubro de 2020		
Tomcat 6	31 de outubro de 2020		

Servidores da web

Versão do servidor web	Data de desativação da plataforma		
IIS 8 em execução no	22 de junho de 2022		

Versão do servidor web	Data de desativação da plataforma		
Windows Server de 64 bits			
Servidor HTTP Apache 2.2	31 de outubro de 2020		
Nginx 1.12.2	31 de outubro de 2020		

Plataformas compatíveis com Elastic Beanstalk

AWS Elastic Beanstalk fornece uma variedade de plataformas nas quais você pode criar seus aplicativos. Projete a aplicação Web para uma dessas plataformas, e o Elastic Beanstalk implanta o código na versão de plataforma selecionada para criar um ambiente de aplicação ativo.

O Elastic Beanstalk oferece plataformas para linguagens de programação (Go, Java, Node.js, PHP, Python, Ruby), servidores de aplicações (Tomcat, Passenger, Puma) e contêineres do Docker. Algumas plataformas possuem várias versões simultaneamente compatíveis.

O Elastic Beanstalk provisiona os recursos necessários para executar a aplicação, incluindo uma ou mais instâncias do Amazon EC2. A pilha de software em execução nas instâncias do Amazon EC2 depende da versão da plataforma específica que você selecionou para o ambiente.

É possível personalizar e configurar o software do qual seu aplicativo depende na plataforma Linux. Saiba mais em [Personalizar software em servidores Linux](#) e [Personalizar software em Windows Servers](#). Há notas de release detalhadas disponíveis para releases recentes nas [Notas de release do AWS Elastic Beanstalk](#).

Plataformas compatíveis

O guia de AWS Elastic Beanstalk plataformas lista todas as versões atuais da ramificação da plataforma na seção Plataformas suportadas do [Elastic Beanstalk](#). O guia de plataformas também lista um histórico da plataforma para cada plataforma, que inclui uma lista das versões anteriores da plataforma de filiais. Para ver o histórico da plataforma de cada plataforma, selecione um dos links a seguir.

- [Docker](#)

- [Go](#)
- [Java SE](#)
- [Tomcat \(executando Java SE\)](#)
- [.NET Core no Linux](#)
- [.NET no Windows Server](#)
- [Node.js](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)

O nome da pilha de soluções para uma ramificação da plataforma

[Você pode usar o nome da pilha de soluções para uma determinada versão da ramificação da plataforma para iniciar um ambiente com a EB CLI, a API do ElasticBeanstalk ou a CLI.AWS O guia de AWS Elastic Beanstalk plataformas lista o nome da pilha de soluções na versão da ramificação da plataforma nas seções Plataformas suportadas pelo Elastic Beanstalk e Histórico da plataforma.](#)

Para recuperar todos os nomes da pilha de soluções que você pode usar para criar um ambiente, use a [ListAvailableSolutionStacks](#)API ou a [aws elasticbeanstalk list-available-solution-stacks](#)na CLI AWS .

Plataformas Linux do Elastic Beanstalk

A maioria das plataformas compatíveis com o Elastic Beanstalk são baseadas no sistema operacional Linux. Especificamente, essas plataformas são baseadas no Amazon Linux, uma distribuição do Linux fornecida pela AWS. As plataformas do Elastic Beanstalk Linux usam instâncias do Amazon Elastic Compute Cloud (Amazon EC2), e essas instâncias executam o Amazon Linux.

As plataformas Linux do Elastic Beanstalk fornecem muitas funcionalidades prontas. Você pode estender as plataformas de várias maneiras para oferecer suporte ao seu aplicativo. Para obter mais detalhes, consulte [the section called “Estender plataformas Linux”](#).

Tópicos

- [Versões compatíveis do Amazon Linux](#)
- [Lista de plataformas Linux do Elastic Beanstalk](#)
- [Estender as plataformas Linux do Elastic Beanstalk](#)

Versões compatíveis do Amazon Linux

O AWS Elastic Beanstalk oferece suporte a plataformas baseadas no Amazon Linux 2 e Amazon Linux 2023.

Em [19 de outubro de 2023](#), o Elastic Beanstalk oferece plataformas AL2023 para todas as linguagens de programação compatíveis com as plataformas Amazon Linux 2. O Beanstalk também oferece suporte ao Docker e às plataformas Docker baseadas em ECS no Amazon Linux 2 e no Amazon Linux 2023.

Para obter mais informações sobre Amazon Linux 2 e Amazon Linux 2023, consulte o seguinte:

- Amazon Linux 2 – [Amazon Linux](#) no Guia do usuário Amazon EC2 para as instâncias Linux.
- Amazon Linux 2023 – [O que é o Amazon Linux 2023?](#) no Guia do usuário Amazon Linux 2023.

Para obter detalhes sobre as versões de plataforma compatíveis, consulte [Plataformas compatíveis com Elastic Beanstalk](#).

Note

É possível migrar sua aplicação de uma ramificação da plataforma AL1 ou AL2 do Elastic Beanstalk para a ramificação equivalente da plataforma AL2023. Para obter mais informações, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Amazon Linux 2023

AWS anunciou a [disponibilidade geral](#) do Amazon Linux 2023 em março de 2023. O guia do usuário Amazon Linux 2023 resume as principais diferenças entre o Amazon Linux 2 e o Amazon Linux 2023. Para obter mais informações, consulte [Comparando Amazon Linux 2 e Amazon Linux 2023](#) no guia do usuário.

Há um alto grau de compatibilidade entre as plataformas Elastic Beanstalk Amazon Linux 2 e Amazon Linux 2023. Embora haja algumas diferenças a serem observadas:

- Instance Metadata Service Version 1 (IMDSv1) – A configuração padrão da opção do [DisableIMDSv1](#) está nas plataformas AL2023 `true`. O padrão está `false` nas plataformas AL2.

- ferramenta de instância pkg-repo — A ferramenta [pkg-repo](#) não está disponível para ambientes executados nas plataformas AL2023. No entanto, você pode aplicar atualizações de pacote e sistema operacional manualmente a uma instância AL2023. Para obter mais informações, consulte [Gerenciamento de pacotes e atualizações do sistema operacional](#) no Guia do usuário do Amazon Linux 2023
- Configuração Apache HTTPd — O arquivo `httpd.conf` Apache para plataformas AL2023 tem algumas configurações diferentes das do AL2:
 - Negar acesso a todo o sistema de arquivos do servidor por padrão. Essas configurações são descritas em Proteger arquivos do servidor por padrão na página [Dicas de segurança](#) do site Apache.
 - Impeça que os usuários substituam os recursos de segurança que você configurou. A configuração nega o acesso à configuração `.htaccess` em todos os diretórios, exceto aqueles especificamente habilitados. Essa configuração é descrita em Protegendo as configurações do sistema na página [Dicas de segurança](#) do site Apache. A página [Tutorial do Apache HTTP Server: arquivos.htaccess](#) afirma que essa configuração pode ajudar a melhorar o desempenho.
 - Negue acesso a arquivos com padrão de nome `.ht*`. Essa configuração impede que os clientes da Web exibam arquivos `.htaccess` e `.htpasswd`.

Você pode alterar qualquer uma das definições de configuração acima para seu ambiente. Para obter mais informações, consulte [Estender as plataformas Linux do Elastic Beanstalk](#). Expanda o tópico Reverse Proxy para ver a seção Configurando o Apache HTTPD.

Lista de plataformas Linux do Elastic Beanstalk

A lista a seguir fornece as plataformas Linux compatíveis com o Elastic Beanstalk para diferentes linguagens de programação, bem como para contêineres Docker. O Elastic Beanstalk oferece plataformas baseadas no Amazon Linux 2 e Amazon Linux 2023 para todos eles. Para saber mais sobre uma plataforma, selecione o link correspondente.

- [Docker \(e ECS Docker\)](#)
- [Go](#)
- [Tomcat \(executando Java SE\)](#)
- [Java SE](#)
- [.NET Core no Linux](#)
- [Node.js](#)

- [PHP](#)
- [Python](#)
- [Ruby](#)

Estender as plataformas Linux do Elastic Beanstalk

As [plataformas Linux do AWS Elastic Beanstalk](#) oferecem muitas funcionalidades de imediato para oferecer suporte ao desenvolvimento e execução de aplicações. Quando necessário, você pode estender as plataformas de várias maneiras para configurar opções, instalar software, adicionar arquivos e executar comandos, fornecer instruções de compilação e runtime e adicionar scripts de inicialização executados em vários estágios de provisionamento das instâncias do Amazon Elastic Compute Cloud (Amazon EC2) do seu ambiente.

Buildfile e Procfile

Algumas plataformas permitem que você personalize a maneira como cria ou prepara seu aplicativo e especifique os processos que executam seu aplicativo. Cada tópico de plataforma individual menciona especificamente Buildfile e/ou Procfile se a plataforma oferecer suporte a eles. Procure sua plataforma específica em [Plataformas](#).

Para todas as plataformas compatíveis, a sintaxe e a semântica são idênticas e descritas nesta página. Tópicos individuais da plataforma mencionam o uso específico desses arquivos para criar e executar aplicativos em suas respectivas linguagens.

Buildfile

Para especificar um comando personalizado de compilação e configuração para o aplicativo, coloque um arquivo chamado Buildfile no diretório raiz da origem do aplicativo. O nome do arquivo diferencia maiúsculas de minúsculas. Use a seguinte sintaxe para o Buildfile.

```
<process_name>: <command>
```

O comando no Buildfile deve corresponder à expressão regular: `^[A-Za-z0-9_-]+:\s*[\s]*.*$`.

O Elastic Beanstalk não monitora a aplicação executada com um Buildfile. Use um Buildfile para comandos que são executados por breves períodos e são encerrados após a conclusão das tarefas. Para processos de aplicativo de longa execução que não devem ser encerrados, use o [Procfile](#).

Todos os caminhos no Buildfile são relativos à raiz do pacote de origem. No seguinte exemplo de um Buildfile, o `build.sh` é um script de shell localizado na raiz do pacote de origem.

Example Buildfile

```
make: ./build.sh
```

Se você quiser fornecer etapas de compilação personalizadas, recomendamos que você use os hooks de plataforma `predeploy` para qualquer coisa, exceto os comandos mais simples, em vez de um Buildfile. Os hooks de plataforma permitem scripts mais avançados e melhor tratamento de erros. Os hooks de plataforma são descritos na próxima seção.

Procfile

Para especificar comandos personalizados para iniciar e executar o aplicativo, coloque um arquivo chamado `Procfile` no diretório raiz da origem do aplicativo. O nome do arquivo diferencia maiúsculas de minúsculas. Use a seguinte sintaxe para o `Procfile`. Você pode especificar um ou mais comandos.

```
<process_name1>: <command1>  
<process_name2>: <command2>  
...
```

Cada linha no `Procfile` deve corresponder à expressão regular: `^[A-Za-z0-9_-]+:\s*[\s].*$`.

Use um `Procfile` para processos de aplicações de longa execução que não devem ser fechadas. O Elastic Beanstalk espera que os processos sejam executados a partir do `Procfile` para serem executados continuamente. O Elastic Beanstalk monitora esses processos e reinicia qualquer processo que seja encerrado. Para processos de curta execução, use um [Buildfile](#).

Todos os caminhos no `Procfile` são relativos à raiz do pacote de origem. O exemplo do `Procfile` a seguir define três processos. O primeiro, chamado `web` no exemplo, é o principal aplicativo web.

Example Procfile

```
web: bin/myserver  
cache: bin/mycache  
foo: bin/fooapp
```

O Elastic Beanstalk configura o servidor de proxy para encaminhar solicitações à sua aplicação Web principal na porta 5000, e esse número de porta pode ser configurado. Um uso comum para um Procfile é passar esse número de porta para sua aplicação como um argumento de comando. Para obter detalhes sobre a configuração do proxy, expanda a seção Configuração de proxy reverso nesta página.

O Elastic Beanstalk captura streams de saída padrão e erros dos processos Procfile em arquivos de log. O Elastic Beanstalk fornece nomes aos arquivos de log após o processo e os armazena no `/var/log`. Por exemplo, o processo web do exemplo anterior gera logs chamados `web-1.log` e `web-1.error.log` para `stdout` e `stderr`, respectivamente.

Hooks de plataforma

Os hooks de plataforma são criados especificamente para estender a plataforma do ambiente. Eles são scripts personalizados e outros arquivos executáveis que são implantados como parte do código-fonte da aplicação, e são executados pelo Elastic Beanstalk durante vários estágios de provisionamento da instância.

Note

Os hooks de plataforma não são compatíveis com as versões da plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2).

Hooks de plataforma de implantação de aplicações

Uma implantação de aplicação ocorre quando você fornece um novo pacote de origem para implantação ou quando faz uma alteração de configuração que exige o encerramento e a recriação de todas as instâncias do ambiente.

Para fornecer hooks de plataforma executados durante a implantação de uma aplicação, coloque os arquivos no diretório `.platform/hooks` do pacote de origem, em um dos seguintes subdiretórios.

- `prebuild`: estes arquivos serão executados depois que o mecanismo da plataforma do Elastic Beanstalk fizer download e extrair o pacote de origem do aplicação, e antes de configurar a aplicação e o servidor Web.

Os arquivos `prebuild` são executados depois da execução de comandos encontrados na seção [commands](#), de qualquer arquivo de configuração, e antes da execução de comandos `Buildfile`.

- `predeploy`: estes arquivos são executados depois que o mecanismo da plataforma do Elastic Beanstalk configura a aplicação e o servidor Web, e antes de implantá-los em seu local final de runtime.

Os arquivos `predeploy` são executados depois da execução dos comandos encontrados na seção [container_commands](#) de qualquer arquivo de configuração, e antes da execução de comandos `Procfile`.

- `postdeploy`: estes arquivos são executados depois que o mecanismo da plataforma do Elastic Beanstalk implanta a aplicação e o servidor de proxy.

Esta é a última etapa do fluxo de trabalho de implantação.

Hooks de plataforma de implantação de configuração

Uma implantação de configuração ocorre quando você faz alterações de configuração que apenas atualizam instâncias do ambiente sem recriá-las. As atualizações de opção a seguir resultam em uma atualização da configuração.

- [Propriedades do ambiente e configurações específicas da plataforma](#)
- [Arquivos estáticos](#)
- [Daemon do AWS X-Ray](#)
- [Streaming e armazenamento de logs](#)
- Porta da aplicação (para obter detalhes, expanda a seção Configuração do proxy reverso nesta página)

Para fornecer hooks que são executados durante a implantação de uma configuração, coloque-os no diretório `.platform/confighooks` do pacote de origem. São aplicados os mesmos três subdiretórios que aqueles aplicados para os hooks de implantação de aplicações.

Mais sobre hooks de plataforma

Os arquivos de hook podem ser arquivos binários ou arquivos de script que começam com uma linha `#!` contendo seu caminho de interpretador, como `#!/bin/bash`. Todos os arquivos devem ter permissão de execução. Use `chmod +x` para definir a permissão de execução em seus arquivos de hook. Em todas as versões de plataforma baseadas no Amazon Linux 2023 e no Amazon Linux 2 lançadas a partir de 29 de abril de 2022, o Elastic Beanstalk concede automaticamente permissões de execução para todos os scripts de hook da plataforma. Nesse caso, não é necessário conceder

manualmente as permissões de execução. Para obter uma lista dessas versões da plataforma, consulte as notas de versão do Linux de [29 de abril de 2022](#) no Guia de notas de versão do AWS Elastic Beanstalk.

O Elastic Beanstalk executa arquivos em cada um desses diretórios em ordem lexicográfica de nomes de arquivos. Todos os arquivos são executados como o usuário `root`. O diretório de trabalho atual (`cwd`) para hooks de plataforma é o diretório raiz do aplicativo. Para os arquivos `prebuild` e `predeploy` é o diretório de preparação do aplicativo, e para os arquivos `postdeploy` é o diretório atual do aplicativo. Se um dos arquivos falhar (terminar com um código de saída diferente de zero), a implantação será interrompida e falhará.

Um script de texto de hooks da plataforma pode falhar se contiver caracteres de quebra de linha retorno de carro/alimentação de linha (CRLF). Se um arquivo foi salvo em um host Windows e depois transferido para um servidor Linux, ele pode conter quebras de linha CRLF do Windows. Para plataformas lançadas a partir de [29 de dezembro de 2022](#), o Elastic Beanstalk converte automaticamente os caracteres CRLF do Windows em caracteres de quebra de linha alimentação de linha (LF) do Linux em arquivos de texto de hooks de plataforma. Se a aplicação for executada em qualquer plataforma Amazon Linux 2 lançada antes dessa data, você precisará converter os caracteres CRLF do Windows em caracteres LF do Linux. Uma forma de fazer isso é criar e salvar o arquivo de script em um host Linux. Ferramentas que convertem esses caracteres também estão disponíveis na internet.

Os arquivos de hook têm acesso a todas as propriedades de ambiente definidas nas opções da aplicação e às variáveis de ambiente do sistema `HOME`, `PATH` e `PORT`.

Para obter valores de variáveis de ambiente e outras opções de configuração nos scripts de hook de plataforma, é possível usar o utilitário `get-config` que o Elastic Beanstalk fornece em instâncias do ambiente. Para obter mais detalhes, consulte [the section called “Ferramentas de script de plataforma”](#).

Arquivos de configuração

Você pode adicionar [arquivos de configuração](#) ao diretório `.ebextensions` do código-fonte da sua aplicação para configurar vários aspectos do ambiente do Elastic Beanstalk. Entre outras coisas, os arquivos de configuração permitem personalizar software e outros arquivos nas instâncias do ambiente e executar comandos de inicialização nas instâncias. Para obter mais informações, consulte [the section called “Servidor do Linux”](#).

Você também pode definir [opções de configuração](#) usando arquivos de configuração. Muitas das opções controlam o comportamento da plataforma, e algumas dessas opções são [específicas da plataforma](#).

Nas plataformas do Amazon Linux 2 e Amazon Linux 2023, recomendamos o uso de hooks de plataforma, Buildfile e Procfile para configurar e executar código personalizado em suas instâncias de ambiente durante o provisionamento de instâncias. Esses mecanismos são descritos nas seções anteriores desta página. Você ainda pode usar comandos e os comandos de contêiner em arquivos de configuração `.ebextensions`, apesar de não ser fácil trabalhar com eles. Por exemplo, escrever scripts de comando dentro de um arquivo YAML pode ser um desafio do ponto de vista da sintaxe. Você ainda precisa usar arquivos de configuração `.ebextensions` para qualquer script que precise de uma referência a um recurso do AWS CloudFormation.

Configuração de proxy reverso

Todas as versões de plataformas Amazon Linux 2 e Amazon Linux 2023 usam o nginx como servidor de proxy reverso padrão. As plataformas Tomcat, Node.js, PHP e Python também oferecem suporte ao Apache HTTPD como uma alternativa. Para selecionar o Apache nessas plataformas, defina a opção `ProxyServer` no namespace `aws:elasticbeanstalk:environment:proxy` como `apache`. Todas as plataformas habilitam a configuração do servidor de proxy de forma uniforme, conforme descrito nesta seção.

Note

Nas versões da plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2), talvez seja necessário configurar servidores de proxy de forma diferente. É possível encontrar esses detalhes legados nos [respectivos tópicos de plataforma](#) neste guia.

O Elastic Beanstalk configura o servidor de proxy nas instâncias do ambiente para encaminhar o tráfego da Web para a aplicação Web principal no URL raiz do ambiente; por exemplo, `http://my-env.elasticbeanstalk.com`.

Por padrão, o Elastic Beanstalk configura o proxy para encaminhar solicitações recebidas na porta 80 para a aplicação Web principal na porta 5000. Você pode configurar esse número de porta definindo a propriedade `PORT` do ambiente usando o namespace [aws:elasticbeanstalk:application:environment](#) em um arquivo de configuração, conforme mostrado no exemplo a seguir.

```
option_settings:
  - namespace: aws:elasticbeanstalk:application:environment
    option_name: PORT
    value: <main_port_number>
```

Para obter mais informações sobre como configurar variáveis de ambiente para o aplicativo, consulte [the section called “Configurações de opção”](#).

Seu aplicativo deve ter como base a porta que está configurada para ele no proxy. Se você alterar a porta padrão usando a propriedade PORT do ambiente, seu código poderá acessá-la ao ler o valor da variável PORT do ambiente. Por exemplo, chame os `.Getenv("PORT")` em Go ou `System.getenv("PORT")` em Java. Se você configurar o proxy para enviar tráfego para vários processos de aplicativo, será possível configurar várias propriedades de ambiente e usar seus valores na configuração de proxy e no código do aplicativo. Outra opção é passar o valor da porta para o processo como um argumento de comando no `Procfile`. Para obter detalhes sobre isso, expanda a seção `Buildfile` e `Procfile` nesta página.

Configurar o nginx

O Elastic Beanstalk usa o nginx como proxy reverso padrão para mapear sua aplicação para o balanceador de carga do Elastic Load Balancing. O Elastic Beanstalk oferece uma configuração de nginx padrão que pode ser estendida ou substituída completamente por sua própria configuração.

Note

Ao adicionar ou editar um arquivo de configuração `.conf` do nginx, codifique-o como UTF-8.

Para estender a configuração nginx padrão do Elastic Beanstalk, adicione arquivos de configuração `.conf` a uma pasta denominada `.platform/nginx/conf.d/` no pacote de origem da sua aplicação. A configuração nginx do Elastic Beanstalk inclui arquivos `.conf` nesta pasta automaticamente.

```
~/workspace/my-app/
|-- .platform
|   |-- nginx
|       |-- conf.d
|           |-- myconf.conf
|-- other source files
```


Para substituir completamente a configuração nginx padrão do Elastic Beanstalk, inclua uma configuração em seu pacote de origem em `.platform/nginx/nginx.conf`.

```
~/workspace/my-app/  
|-- .platform  
|   |-- nginx  
|       |-- nginx.conf  
|-- other source files
```

Se você substituir a configuração nginx do Elastic Beanstalk, adicione a linha a seguir ao seu `nginx.conf` para extrair as configurações do Elastic Beanstalk para [Monitoramento e relatório de integridade aprimorada](#), mapeamentos automáticos de aplicações e arquivos estáticos.

```
include conf.d/elasticbeanstalk/*.conf;
```

Configurar o Apache HTTPD

As plataformas Tomcat, Node.js, PHP e Python permitem escolher o servidor de proxy Apache HTTPD como uma alternativa ao nginx. Esse não é o padrão. O exemplo a seguir configura o Elastic Beanstalk para usar o Apache HTTPD.

Example `.ebextensions/httpd-proxy.config`

```
option_settings:  
  aws:elasticbeanstalk:environment:proxy:  
    ProxyServer: apache
```

É possível estender a configuração padrão do Apache do Elastic Beanstalk com seus arquivos de configuração adicionais. Também há opção de substituir completamente a configuração padrão do Apache do Elastic Beanstalk.

Para estender a configuração padrão do Apache do Elastic Beanstalk, adicione arquivos de configuração `.conf` a uma pasta chamada `.platform/httpd/conf.d` no pacote de origem da aplicação. A configuração do Apache do Elastic Beanstalk inclui os arquivos `.conf` nessa pasta automaticamente.

```
~/workspace/my-app/  
|-- .ebextensions  
|   -- httpd-proxy.config
```

```
|-- .platform
|  -- httpd
|    -- conf.d
|      -- port5000.conf
|      -- ssl.conf
-- index.jsp
```

Por exemplo, a configuração do Apache 2.4 a seguir adiciona um ouvinte na porta 5000.

Example `.platform/httpd/conf.d/port5000.conf`

```
listen 5000
<VirtualHost *:5000>
  <Proxy *>
    Require all granted
  </Proxy>
  ProxyPass / http://localhost:8080/ retry=0
  ProxyPassReverse / http://localhost:8080/
  ProxyPreserveHost on

  ErrorLog /var/log/httpd/elasticbeanstalk-error_log
</VirtualHost>
```

Para substituir completamente a configuração padrão do Apache do Elastic Beanstalk, inclua uma configuração no pacote de origem em `.platform/httpd/conf/httpd.conf`.

```
~/workspace/my-app/
|-- .ebextensions
|  -- httpd-proxy.config
|-- .platform
|  `-- httpd
|     `-- conf
|        `-- httpd.conf
`-- index.jsp
```

Note

Se você substituir a configuração do Apache do Elastic Beanstalk, adicione as linhas a seguir ao `httpd.conf` para extrair as configurações do Elastic Beanstalk para [Monitoramento e relatório de integridade aprimorada](#), mapeamentos automáticos de aplicações e arquivos estáticos.

```
IncludeOptional conf.d/elasticbeanstalk/*.conf
```

Note

Se você estiver migrando a aplicação do Elastic Beanstalk para uma plataforma Amazon Linux 2 ou Amazon Linux 2023, leia também as informações em [the section called “Migrar para AL2023/AL2”](#).

Tópicos

- [Exemplo de aplicativo com extensões](#)
- [Fluxo de trabalho de implantação da instância](#)
- [Fluxo de trabalho de implantação de instâncias para o ECS em execução no Amazon Linux 2 e versões posteriores](#)
- [Ferramentas de script de plataforma](#)

Exemplo de aplicativo com extensões

O exemplo a seguir demonstra um pacote de origem de aplicação com vários recursos de capacidade de extensão compatíveis com as plataformas Amazon Linux 2 e Amazon Linux 2023 do Elastic Beanstalk: um Procfile e arquivos de configuração `.ebextensions`, hooks personalizados e arquivos de configuração de proxy.

```
~/my-app/  
|-- web.jar  
|-- Procfile  
|-- readme.md  
|-- .ebextensions/  
|   |-- options.config           # Option settings  
|   `-- cloudwatch.config       # Other .ebextensions sections, for example files and  
    container commands  
`-- .platform/  
    |-- nginx/                   # Proxy configuration  
    |   |-- nginx.conf  
    |   `-- conf.d/
```

```
|      `-- custom.conf
|-- hooks/          # Application deployment hooks
|  |-- prebuild/
|  |  |-- 01_set_secrets.sh
|  |  `-- 12_update_permissions.sh
|  |-- predeploy/
|  |  `-- 01_some_service_stop.sh
|  `-- postdeploy/
|      |-- 01_set_tmp_file_permissions.sh
|      |-- 50_run_something_after_app_deployment.sh
|      `-- 99_some_service_start.sh
`-- confighooks/   # Configuration deployment hooks
    |-- prebuild/
    |  `-- 01_set_secrets.sh
    |-- predeploy/
    |  `-- 01_some_service_stop.sh
    `-- postdeploy/
        |-- 01_run_something_after_config_deployment.sh
        `-- 99_some_service_start.sh
```

Note

Algumas dessas extensões não são compatíveis com as versões da plataforma de AMI do Amazon Linux (anteriores ao Amazon Linux 2).

Fluxo de trabalho de implantação da instância

Note

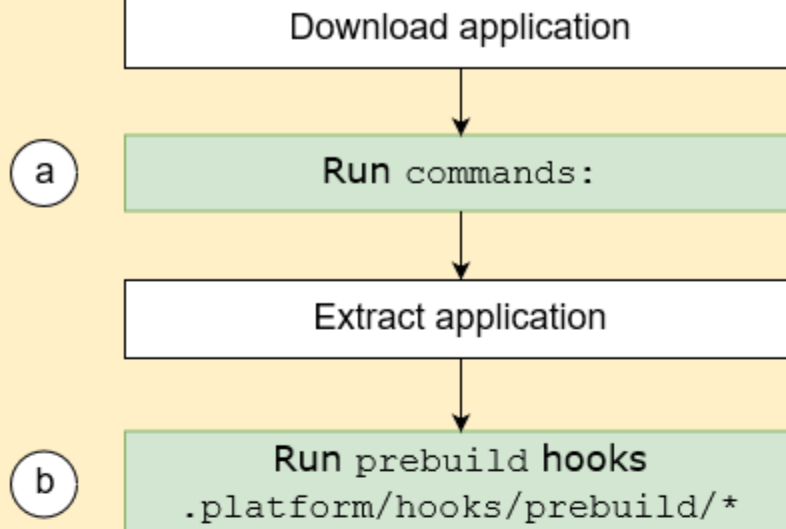
As informações nesta seção não se aplicam à ramificação da plataforma ECS em execução no Amazon Linux 2 e no Amazon Linux 2023. Para obter mais informações, consulte a próxima seção, [Fluxo de trabalho de implantação de instâncias para o ECS em execução no Amazon Linux 2 e versões posteriores](#).

Como existem muitas maneiras de estender a plataforma do ambiente, é útil saber o que acontece sempre que o Elastic Beanstalk provisiona ou implementa uma instância. O diagrama a seguir mostra todo esse fluxo de trabalho de implantação. Ele representa as diferentes fases de uma implantação e as etapas que o Elastic Beanstalk realiza em cada fase.

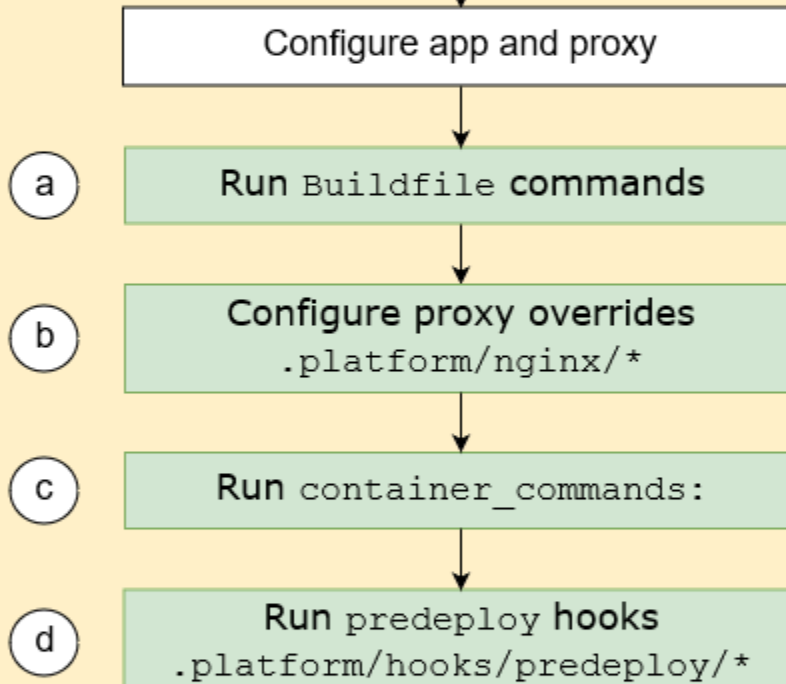
Observações

- O diagrama não representa o conjunto completo de etapas que o Elastic Beanstalk realiza em instâncias do ambiente durante a implantação. Nós fornecemos este diagrama como ilustração, para fornecer a você a ordem e o contexto para a execução de suas personalizações.
- Para simplificar, o diagrama menciona apenas os subdiretórios de hook `.platform/hooks/*` (para implantações de aplicações), e não os subdiretórios de hook `.platform/confighooks/*` (para implantações de configurações). Os hooks nos últimos subdiretórios são executados durante exatamente as mesmas etapas que os hooks nos subdiretórios correspondentes mostrados no diagrama.

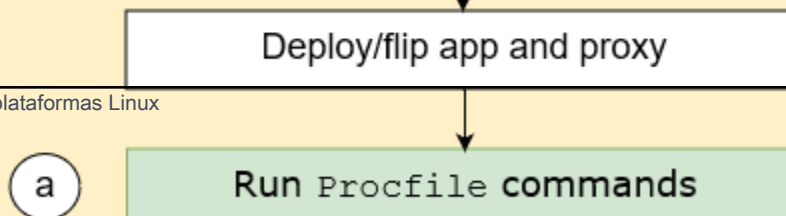
1. Initial steps



2. Configure



3. Deploy



A lista a seguir detalha as fases e etapas de implantação.

1. Passos iniciais

O Elastic Beanstalk baixa e extrai a aplicação. Após cada uma dessas etapas, o Elastic Beanstalk executa uma das etapas de extensibilidade.

- a. Executa comandos encontrados na seção [commands](#): de qualquer arquivo de configuração.
- b. Executa todos os arquivos executáveis encontrados no diretório `.platform/hooks/prebuild` do pacote de origem (`.platform/confighooks/prebuild` para uma implantação de configuração).

2. Configure

O Elastic Beanstalk configura a aplicação e o servidor de proxy.

- a. Executa os comandos encontrados no `Buildfile` no pacote de origem.
- b. Copia seus arquivos de configuração de proxy personalizados, se você tiver algum no diretório `.platform/nginx` do pacote de origem, para seu local de runtime.
- c. Executa comandos encontrados na seção [container_commands](#): de qualquer arquivo de configuração.
- d. Executa todos os arquivos executáveis encontrados no diretório `.platform/hooks/predeploy` do pacote de origem (`.platform/confighooks/predeploy` para uma implantação de configuração).

3. Implante

O Elastic Beanstalk implanta e executa a aplicação e o servidor de proxy.

- a. Executa o comando encontrado no arquivo `Procfile` em seu pacote de origem.
- b. Executa ou executa novamente o servidor de proxy com seus arquivos de configuração de proxy personalizados, se você tiver algum.
- c. Executa todos os arquivos executáveis encontrados no diretório `.platform/hooks/postdeploy` do pacote de origem (`.platform/confighooks/postdeploy` para uma implantação de configuração).

Fluxo de trabalho de implantação de instâncias para o ECS em execução no Amazon Linux 2 e versões posteriores

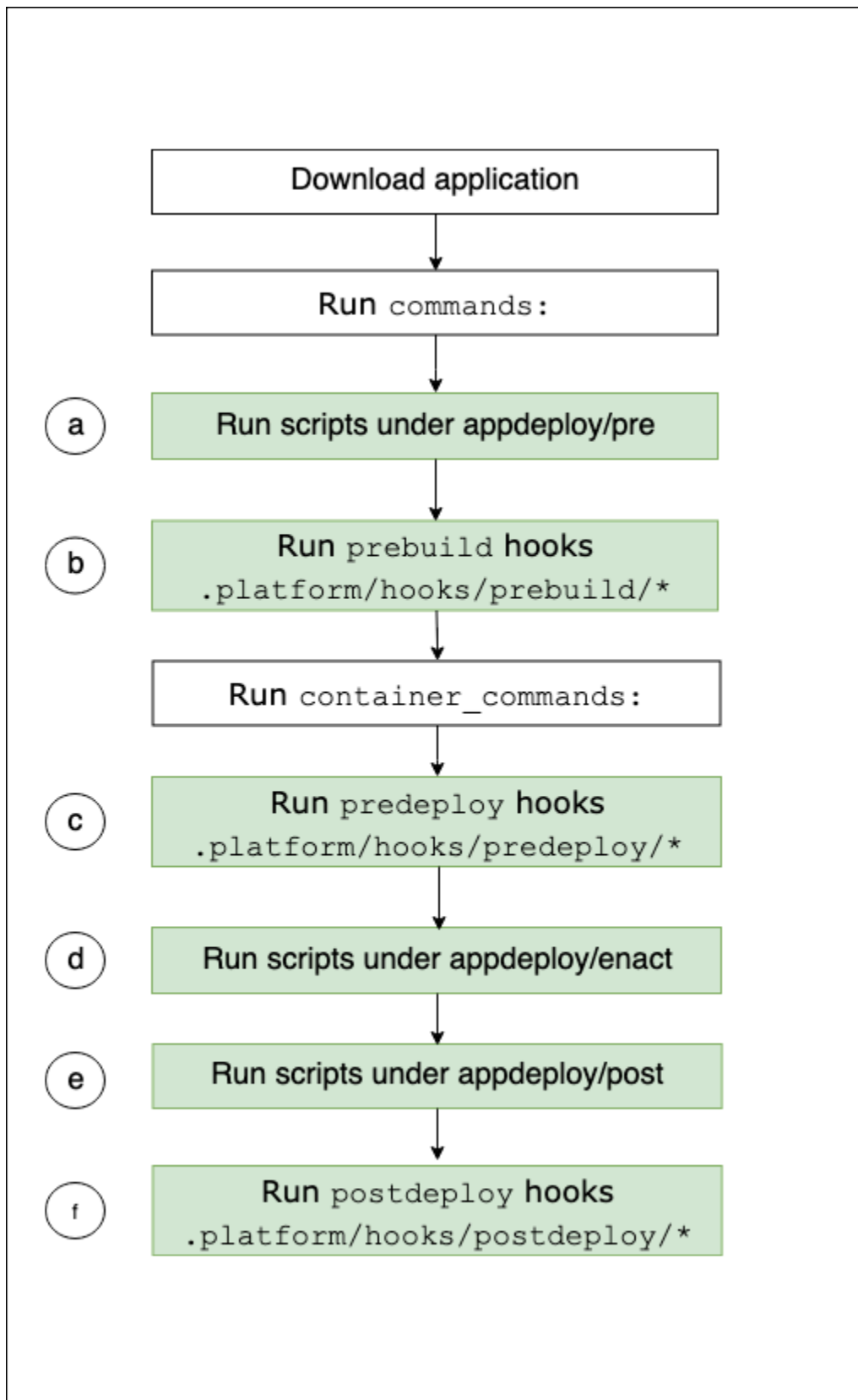
A seção anterior descreve os recursos de extensibilidade compatíveis ao longo das fases do fluxo de trabalho de implantação da aplicação. Há algumas diferenças para ramificação da plataforma Docker no [ECS em execução no Amazon Linux 2 e versões posteriores](#). Esta seção explica como esses conceitos se aplicam a essa ramificação específica da plataforma.

Como existem muitas maneiras de estender a plataforma do ambiente, é útil saber o que acontece sempre que o Elastic Beanstalk provisiona ou implementa uma instância. O diagrama a seguir mostra todo esse fluxo de trabalho de implantação para um ambiente baseado nas ramificações de plataforma ECS em execução no Amazon Linux 2 e ECS em execução no Amazon Linux 2023. Ele representa as diferentes fases de uma implantação e as etapas que o Elastic Beanstalk realiza em cada fase.

Ao contrário do fluxo de trabalho descrito na seção anterior, a fase de configuração da implantação não é compatível com os seguintes recursos de extensibilidade: comandos `Buildfile`, comandos `Procfile`, configuração de proxy reverso.

Observações

- O diagrama não representa o conjunto completo de etapas que o Elastic Beanstalk realiza em instâncias do ambiente durante a implantação. Nós fornecemos este diagrama como ilustração, para fornecer a você a ordem e o contexto para a execução de suas personalizações.
- Para simplificar, o diagrama menciona apenas os subdiretórios de hook `.platform/hooks/*` (para implantações de aplicações), e não os subdiretórios de hook `.platform/confighooks/*` (para implantações de configurações). Os hooks nos últimos subdiretórios são executados durante exatamente as mesmas etapas que os hooks nos subdiretórios correspondentes mostrados no diagrama.



A lista a seguir detalha as etapas do fluxo de trabalho de implantação.

- a. Executa todos os arquivos executáveis encontrados no diretório `appdeploy/pre` em `EBhooksDir`.
- b. Executa todos os arquivos executáveis encontrados no diretório `.platform/hooks/prebuild` do pacote de origem (`.platform/confighooks/prebuild` para uma implantação de configuração).
- c. Executa todos os arquivos executáveis encontrados no diretório `.platform/hooks/predeploy` do pacote de origem (`.platform/confighooks/predeploy` para uma implantação de configuração).
- d. Executa todos os arquivos executáveis encontrados no diretório `appdeploy/enact` em `EBhooksDir`.
- e. Executa todos os arquivos executáveis encontrados no diretório `appdeploy/post` em `EBhooksDir`.
- f. Executa todos os arquivos executáveis encontrados no diretório `.platform/hooks/postdeploy` do pacote de origem (`.platform/confighooks/postdeploy` para uma implantação de configuração).

A referência a `EBhooksDir` representa o caminho do diretório de hooks da plataforma. Para recuperar o nome do caminho do diretório, use a ferramenta de script [get-config](#) na linha de comando da instância do ambiente, como mostrado:

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig -k EBhooksDir
```

Ferramentas de script de plataforma

Este tópico descreve as ferramentas que AWS Elastic Beanstalk fornece para ambientes que usam plataformas do Amazon Linux. As ferramentas estão localizadas nas instâncias do Amazon EC2 dos ambientes do Elastic Beanstalk.

get-config

Use a ferramenta `get-config` para recuperar valores de variáveis de ambiente e outras informações da plataforma e da instância. A ferramenta está disponível em `/opt/elasticbeanstalk/bin/get-config`.

comandos get-config

Cada comando da ferramenta `get-config` retorna um tipo específico de informação. Use a sintaxe a seguir para executar os comandos de qualquer uma das ferramentas.

```
$ /opt/elasticbeanstalk/bin/get-config command [ options ]
```

O exemplo a seguir executa o comando `environment`:

```
$ /opt/elasticbeanstalk/bin/get-config environment -k PORT
```

Dependendo do comando e das opções que você escolher, a ferramenta retorna um objeto (JSON ou YAML) com pares de chave-valor ou um único valor.

É possível testar `get-config` usando SSH para se conectar a uma instância do EC2 no ambiente do Elastic Beanstalk.

Note

Quando você executa `get-config` para teste, alguns comandos podem exigir privilégios de usuário raiz para acessar as informações subjacentes. Se você receber um erro de permissão de acesso, execute o comando novamente em `sudo`.

Você não precisa adicionar `sudo` ao usar a ferramenta nos scripts implantados no ambiente. O Elastic Beanstalk executa todos os seus scripts como o usuário raiz.

As seções a seguir descrevem os comandos da ferramenta.

`optionsettings`: opções de configuração

O comando `get-config optionsettings` retorna um objeto listando as opções de configuração definidas no ambiente e usadas pela plataforma em instâncias do ambiente. Elas são organizadas por namespace.

```
$ /opt/elasticbeanstalk/bin/get-config optionsettings
{"aws:elasticbeanstalk:application:environment":
{"JDBC_CONNECTION_STRING":"","aws:elasticbeanstalk:container:tomcat:jvmoptions":{"JVM
Options":"","Xms":"256m","Xmx":"256m"},"aws:elasticbeanstalk:environment:proxy":
{"ProxyServer":"nginx","StaticFiles":
[""]},"aws:elasticbeanstalk:healthreporting:system":
```

```
{"SystemType":"enhanced"},"aws:elasticbeanstalk:hostmanager":  
{"LogPublicationControl":"false"}}
```

Para retornar o valor de uma opção de configuração específica, use a opção `--namespace (-n)` para especificar um namespace, e a opção `--option-name (-o)` para especificar o nome de uma opção.

```
$ /opt/elasticbeanstalk/bin/get-config optionsettings -  
n aws:elasticbeanstalk:container:php:phpini -o memory_limit  
256M
```

`environment` : propriedades do ambiente

O comando `get-config environment` retorna um objeto que contém uma lista de propriedades do ambiente. Elas incluem propriedades configuradas pelo usuário e aquelas fornecidas pelo Elastic Beanstalk.

```
$ /opt/elasticbeanstalk/bin/get-config environment  
{"JDBC_CONNECTION_STRING":"","RDS_PORT":"3306","RDS_HOSTNAME":"anj9aw1b0tbj6b.cijbpanmxz5u.us-  
west-2.rds.amazonaws.com","RDS_USERNAME":"testusername","RDS_DB_NAME":"ebdb","RDS_PASSWORD":"te
```

Por exemplo, o Elastic Beanstalk fornece propriedades de ambiente para conexão a uma instância de banco de dados integrada do Amazon RDS (por exemplo, `RDS_HOSTNAME`). Essas propriedades de conexão do RDS aparecem na saída do `get-config environment`. No entanto, elas não aparecem na saída do `get-config optionsettings`. Isso ocorre porque elas não foram definidas nas opções de configuração.

Para retornar uma propriedade específica do ambiente, use a opção `--key (-k)` para especificar uma chave de propriedade.

```
$ /opt/elasticbeanstalk/bin/get-config environment -k TESTPROPERTY  
testvalue
```

`container`: valores de configuração na instância

O comando `get-config container` retorna um objeto que lista valores de configuração de ambiente e plataforma para instâncias de ambiente.

O exemplo a seguir mostra a saída do comando em um ambiente Tomcat do Amazon Linux 2.

```
$ /opt/elasticbeanstalk/bin/get-config container
```

```

{"common_log_list":["/var/log/eb-engine.log", "/var/log/eb-hooks.log"], "default_log_list":["/var/log/nginx/access.log", "/var/log/nginx/error.log"], "environment_name": "myenv-1da84946", "instance_port": "80", "log_group_name_prefix": "aws/elasticbeanstalk", "proxy_server": "nginx", "static_files": [""], "xray_enabled": "false"}

```

Para retornar o valor de uma chave específica, use a opção `--key (-k)` para especificar a chave.

```

$ /opt/elasticbeanstalk/bin/get-config container -k environment_name
myenv-1da84946

```

`addons`: valores de configuração de complementos

O comando `get-config addons` retorna um objeto que contém informações de configuração adicionadas ao complemento do ambiente. Use-o para recuperar a configuração de um banco de dados do Amazon RDS associado ao ambiente.

```

$ /opt/elasticbeanstalk/bin/get-config addons
{"rds":{"Description":"RDS Environment variables","env":{"RDS_DB_NAME":"ebdb", "RDS_HOSTNAME":"ea13k2wimu1dh8i.c18mnpu5rwwg.us-east-2.rds.amazonaws.com", "RDS_PASSWORD":"password", "RDS_PORT":"3306", "RDS_USERNAME":"user"}}}

```

É possível restringir o resultado de duas maneiras. Para recuperar valores de um complemento específico, use a opção `--add-on (-a)` para especificar o nome do complemento.

```

$ /opt/elasticbeanstalk/bin/get-config addons -a rds
{"Description":"RDS Environment variables","env":{"RDS_DB_NAME":"ebdb", "RDS_HOSTNAME":"ea13k2wimu1dh8i.c18mnpu5rwwg.us-east-2.rds.amazonaws.com", "RDS_PASSWORD":"password", "RDS_PORT":"3306", "RDS_USERNAME":"user"}}

```

Para retornar o valor de uma chave específica dentro de um complemento, adicione a opção `--key (-k)` para especificar a chave.

```

$ /opt/elasticbeanstalk/bin/get-config addons -a rds -k RDS_DB_NAME
ebdb

```

`platformconfig`: valores de configuração constantes

O comando `get-config platformconfig` retorna um objeto que contém informações de configuração da plataforma que são constantes para a versão da plataforma. A saída é a mesma em

todos os ambientes que executam a mesma versão da plataforma. O objeto de saída do comando tem dois objetos incorporados:

- **GeneralConfig**: contém informações constantes nas versões mais recentes de todas as ramificações de plataformas Amazon Linux 2 e Amazon Linux 2023.
- **PlatformSpecificConfig**: contém informações que são constantes para a versão da plataforma e que são específicas para ela.

O exemplo a seguir mostra a saída do comando em um ambiente que usa o Tomcat 8.5 executando a ramificação da plataforma Corretto 11.

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig
{"GeneralConfig":{"AppUser":"webapp","AppDeployDir":"/var/app/
current/","AppStagingDir":"/var/app/
staging/","ProxyServer":"nginx","DefaultInstancePort":"80"},"PlatformSpecificConfig":
{"ApplicationPort":"8080","JavaVersion":"11","TomcatVersion":"8.5"}}
```

Para retornar o valor de uma chave específica, use a opção `--key (-k)` para especificar a chave. Essas chaves são exclusivas entre os dois objetos incorporados. Não é necessário especificar o objeto que contém a chave.

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig -k AppStagingDir
/var/app/staging/
```

Opções de saída get-config

Use a opção `--output` para especificar o formato do objeto de saída. Os valores válidos são JSON (padrão) e YAML. Essa é uma opção global. Você deve especificá-la antes do nome do comando.

O exemplo a seguir retorna valores de opção de configuração no formato YAML.

```
$ /opt/elasticbeanstalk/bin/get-config --output YAML optionsettings
aws:elasticbeanstalk:application:environment:
  JDBC_CONNECTION_STRING: ""
aws:elasticbeanstalk:container:tomcat:jvmoptions:
  JVM Options: ""
  Xms: 256m
  Xmx: 256m
aws:elasticbeanstalk:environment:proxy:
  ProxyServer: nginx
```

```
StaticFiles:
  - ""
aws:elasticbeanstalk:healthreporting:system:
  SystemType: enhanced
aws:elasticbeanstalk:hostmanager:
  LogPublicationControl: "false"
```

pkg-repo

Note

A ferramenta `pkg-repo` não está disponível para ambientes baseados em plataformas Amazon Linux 2023. No entanto, você pode aplicar atualizações de pacote e sistema operacional manualmente a uma instância AL2023. Para obter mais informações, consulte [Gerenciamento de pacotes e atualizações do sistema operacional](#) no Guia do usuário do Amazon Linux 2023

Em algumas circunstâncias urgentes, talvez seja necessário atualizar suas instâncias do Amazon EC2 com um patch de segurança do Amazon Linux 2 que ainda não foi lançado com as versões necessárias da plataforma do Elastic Beanstalk. Você não pode executar uma atualização manual em seus ambientes do Elastic Beanstalk por padrão. Isso ocorre porque as versões da plataforma estão bloqueadas em uma versão específica do repositório do Amazon Linux 2. Esse bloqueio garante que as instâncias executem versões de software compatíveis e consistentes. Para casos urgentes, a ferramenta `pkg-repo` permite uma solução alternativa para atualizar manualmente os pacotes yum no Amazon Linux 2 se você precisar instalá-los em um ambiente antes de serem lançados em uma nova versão da plataforma do Elastic Beanstalk.

A ferramenta `pkg-repo` das plataformas Amazon Linux 2 permitem desbloquear os repositórios dos pacotes yum. Em seguida, você pode executar manualmente um yum update para um patch de segurança. Por outro lado, você pode seguir a atualização usando a ferramenta para bloquear os repositórios de pacotes yum para evitar atualizações adicionais. A ferramenta `pkg-repo` está disponível no diretório `/opt/elasticbeanstalk/bin/pkg-repo` de todas as instâncias do EC2 no seu ambiente do Elastic Beanstalk.

Alterações usando a ferramenta `pkg-repo` são feitas somente na instância do EC2 em que a ferramenta é usada. Elas não afetam outras instâncias nem impedem atualizações futuras do ambiente. Os exemplos fornecidos posteriormente neste tópico explicam como aplicar as alterações em todas as instâncias chamando os comandos `pkg-repo` de scripts e arquivos de configuração.

⚠ Warning

Nós não recomendamos essa ferramenta para a maioria dos usuários. Quaisquer alterações manuais aplicadas a uma versão desbloqueada da plataforma é considerada fora de banda. Essa opção só é viável para os usuários em circunstâncias urgentes que podem aceitar os seguintes riscos:

- Não é possível garantir que as versões dos pacotes sejam consistentes entre todas as instâncias de seus ambientes.
- Não há garantias de que ambientes modificados usando-se a ferramenta `pkg-repo` funcionem corretamente. Eles não foram testados e verificados em plataformas compatíveis com o Elastic Beanstalk.

Recomendamos a aplicação de práticas recomendadas que incluem planos de teste e backout. Para ajudar a aplicar as práticas recomendadas, você pode usar o console do Elastic Beanstalk e a CLI do EB para clonar um ambiente e trocar URLs do ambiente. Para obter mais informações sobre como usar essas operações, consulte [Implantações azuis/verdes](#) no capítulo Gerenciar ambientes deste guia.

Se você planeja editar manualmente os arquivos de configuração do repositório yum, execute a ferramenta `pkg-repo` primeiro. A ferramenta `pkg-repo` talvez não funcione como o esperado em um ambiente Amazon Linux 2 com arquivos de configuração do repositório yum editados manualmente. Isso ocorre porque a ferramenta pode não reconhecer as alterações de configuração.

Para obter mais informações sobre o repositório de pacotes do Amazon Linux, consulte o tópico sobre o [Repositório de pacotes](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

comandos `pkg-repo`

Use a sintaxe a seguir para executar os comandos da ferramenta `pkg-repo`.

```
$ /opt/elasticbeanstalk/bin/pkg-repo command [options]
```

Os comandos de `pkg-repo` são os seguintes:

- `lock`: bloqueia os repositórios de pacotes yum em uma versão específica
- `unlock`: desbloqueia os repositórios de pacotes yum de uma versão específica

- **status**: lista todos os repositórios de pacotes yum e o status de bloqueio atual deles
- **help**: mostra ajuda geral ou ajuda para um comando

As opções a seguir se aplicam aos seguintes comandos.

- **lock**, **unlock** e **status** - opções: **-h**, **--help**, ou nenhum (padrão).
- **help** - opções: **lock**, **unlock**, **status** ou nenhum (padrão).

O exemplo a seguir executa o comando **unlock**:

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo unlock
Amazon Linux 2 core package repo successfully unlocked
Amazon Linux 2 extras package repo successfully unlocked
```

O exemplo a seguir executa o comando **lock**:

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo lock
Amazon Linux 2 core package repo successfully locked
Amazon Linux 2 extras package repo successfully locked
```

O exemplo a seguir executa o comando **status**:

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo status
Amazon Linux 2 core package repo is currently UNLOCKED
Amazon Linux 2 extras package repo is currently UNLOCKED
```

O exemplo a seguir executa o comando **help** para o comando **lock**.

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo help lock
```

O exemplo a seguir executa o comando **help** para a ferramenta **pkg-repo**.

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo help
```

É possível testar **pkg-repo** usando SSH para se conectar a uma instância no ambiente do Elastic Beanstalk. Uma opção de SSH é o comando CLI do EB [eb ssh](#).

Note

A ferramenta `pkg-repo` requer privilégios de usuário `root` para execução. Se você receber um erro de permissão de acesso, execute o comando novamente em `sudo`.

Você não precisa adicionar `sudo` ao usar a ferramenta nos scripts ou arquivos de configuração que você implanta no seu ambiente. O Elastic Beanstalk executa todos os seus scripts como o usuário raiz.

exemplos de pkg-repo

A seção anterior fornece exemplos de linha de comando para testes em uma instância do EC2 individual de um ambiente do Elastic Beanstalk. Essa abordagem pode ser útil para testes. No entanto, ela atualiza apenas uma instância por vez, portanto, não é prático aplicar alterações em todas as instâncias em um ambiente.

Uma abordagem mais pragmática é usar scripts de [hook de plataforma](#) ou um arquivo de configuração [.ebextensions](#) para aplicar as alterações em todas as instâncias de forma consistente.

O exemplo a seguir chama o `pkg-repo` de um arquivo de configuração da pasta [.ebextensions](#). O Elastic Beanstalk executa os comandos no arquivo `update_package.config` quando você implanta seu pacote-fonte de aplicação.

```
.ebextensions
### update_package.config
```

Para receber a versão mais recente do pacote de `docker`, essa configuração especifica o pacote de `docker` no comando `yum update`.

```
### update_package.config ###

commands:
  update_package:
    command: |
      /opt/elasticbeanstalk/bin/pkg-repo unlock
      yum update docker -y
      /opt/elasticbeanstalk/bin/pkg-repo lock
      yum clean all -y
      rm -rf /var/cache/yum
```

Essa configuração não especifica nenhum pacote no comando `yum update`. A consequência é que todas as atualizações disponíveis são aplicadas.

```
### update_package.config ###

commands:
  update_package:
    command: |
      /opt/elasticbeanstalk/bin/pkg-repo unlock
      yum update -y
      /opt/elasticbeanstalk/bin/pkg-repo lock
      yum clean all -y
      rm -rf /var/cache/yum
```

O exemplo a seguir chama o `pkg-repo` de um script bash como um [hook de plataforma](#). O Elastic Beanstalk executa o arquivo de script `update_package.sh` localizado no subdiretório `prebuild`.

```
.platform
### hooks
  ### prebuild
    ### update_package.sh
```

Para receber a versão mais recente do pacote de `docker`, esse script especifica o pacote do `docker` no comando `yum update`. Se o nome do pacote for omitido, todas as atualizações disponíveis serão aplicadas. O exemplo do arquivo de configuração anterior demonstra isso.

```
### update_package.sh ###

#!/bin/bash

/opt/elasticbeanstalk/bin/pkg-repo unlock
yum update docker -y
/opt/elasticbeanstalk/bin/pkg-repo lock
yum clean all -y
rm -rf /var/cache/yum
```

`download-source-bundle` (somente AMI do Amazon Linux)

Nas ramificações de plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2), o Elastic Beanstalk fornece uma ferramenta adicional, que é `download-source-bundle`. Use essa

ferramenta para baixar o código-fonte da sua aplicação durante a implantação da plataforma. A ferramenta está disponível em `/opt/elasticbeanstalk/bin/download-source-bundle`.

O script de exemplo `00-unzip.sh` está localizado na pasta `appdeploy/pre` em instâncias do ambiente. Ele demonstra como usar `download-source-bundle` para baixar o código-fonte da aplicação para a pasta `/opt/elasticbeanstalk/deploy/appsource` durante a implantação.

Implantar aplicações do Elastic Beanstalk a partir de contêineres do Docker

O Elastic Beanstalk é compatível com a implantação de aplicações Web a partir de contêineres do Docker. Com os contêineres do Docker, você pode definir seu próprio ambiente de runtime. Você também pode escolher sua própria plataforma, linguagem de programação e dependências da aplicação, como gerenciadores de pacotes ou ferramentas, que normalmente não são compatíveis com outras plataformas. Os contêineres do Docker são independentes e incluem todas as informações de configuração e software necessários à execução da aplicação Web. Todas as variáveis de ambiente definidas no console do Elastic Beanstalk são transmitidas para os contêineres.

Os tópicos neste capítulo pressupõem que você tem um nível básico de conhecimento sobre ambientes do Elastic Beanstalk. Se você não tiver usado o Elastic Beanstalk antes, tente o [tutorial de conceitos básicos](#) para saber os princípios. Este capítulo também pressupõe que você tenha uma compreensão básica do Docker e de como ele funciona. Para obter mais informações, consulte [Docker overview](#) (Visão geral do Docker) no site da Docker.

As ramificações da plataforma Docker

A plataforma Docker do Elastic Beanstalk oferece suporte às seguintes ramificações da plataforma:

Docker executando Amazon Linux 2 e Docker executando AL2023

O Elastic Beanstalk implanta contêineres e código-fonte do Docker nas instâncias do EC2 e as gerencia. Essas ramificações da plataforma oferecem suporte a vários contêineres. Você pode usar a ferramenta Docker Compose para simplificar a configuração, o teste e a implantação do seu aplicativo. Para obter mais informações sobre essa plataforma, consulte [the section called “Ramificação da plataforma Docker”](#).

ECS em execução no Amazon Linux 2 e ECS em execução no AL2023

Fornecemos esta ramificação para clientes que precisam de um caminho de migração para o AL2023/AL2 da ramificação da plataforma descontinuada Docker de vários contêineres em execução no (Amazon Linux AMI). As ramificações de plataforma mais recentes oferecem suporte a todos os recursos da ramificação da plataforma descontinuada. Não é necessária nenhuma alteração no código-fonte. Para obter mais informações, consulte [Migrar o Docker de vários contêineres em execução no Amazon Linux para o ECS no Amazon Linux 2023](#). Se você não tiver um ambiente Elastic Beanstalk em execução em uma ramificação da plataforma baseada em ECS, recomendamos usar a ramificação da plataforma, Docker Running no AL2023 de 64 bits. Isso oferece uma abordagem mais simples e requer menos recursos.

Essa ramificação da plataforma usa o Amazon ECS para coordenar uma implantação de vários contêineres do Docker em um cluster do Amazon ECS em um ambiente do Elastic Beanstalk. O cluster ECS contém instâncias do EC2 que hospedam contêineres Docker. Para realizar essa orquestração de implantação, o Elastic Beanstalk cria uma definição de tarefa do ECS que também inclui uma definição de contêiner. Cada uma das instâncias do ambiente executa o mesmo conjunto de contêineres que são definidos em um arquivo `DockerTaskDefinition.json`. Para obter mais informações, consulte [the section called “Ramificação da plataforma gerenciada pelo ECS”](#).

Ramificações da plataforma desativadas em execução no Amazon Linux AMI (AL1)

Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Expanda cada seção a seguir para ler mais sobre cada ramificação da plataforma descontinuada e seu caminho de migração para a ramificação da plataforma mais recente em execução no Amazon Linux 2 ou Amazon Linux 2023 (recomendado).

Docker (AMI do Amazon Linux)

Esta ramificação da plataforma pode implantar uma imagem do Docker, descrita em um `Dockerfile` ou em uma definição do `DockerTaskDefinition.json` v1. Essa ramificação da plataforma executa apenas um contêiner para cada instância. Suas ramificações da plataforma sucessivas, Docker rodando em AL2023 de 64 bits e Docker rodando em Amazon Linux 2 de 64 bits, suportam vários contêineres Docker por instância.

Recomendamos que você crie os ambientes com a ramificação da plataforma compatível mais recente Docker em execução AL2023 de 64 bits. Você pode então migrar seu aplicativo para o ambiente recém-criado. Para obter mais informações sobre esses ambientes, consulte [the section called “Ramificação da plataforma Docker”](#). Para obter mais informações sobre a migração, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Docker de vários contêineres (Amazon Linux AMI)

Essa ramificação da plataforma usa o Amazon ECS para coordenar uma implantação de vários contêineres do Docker em um cluster do Amazon ECS em um ambiente do Elastic Beanstalk. Se você estiver usando esta ramificação da plataforma descontinuada, recomendamos migrar para a ramificação da plataforma ECS em execução no Amazon Linux 2023 mais recente. A ramificação mais recente da plataforma é compatível com todos os recursos da ramificação da plataforma que foi retirada. Não é necessária nenhuma alteração no código-fonte. Para obter mais informações, consulte [Migrar o Docker de vários contêineres em execução no Amazon Linux para o ECS no Amazon Linux 2023](#).

Contêineres do Docker pré-configurados

Além das plataformas Docker mencionadas anteriormente, há também a ramificação da plataforma Docker GlassFish pré-configurada que roda no sistema operacional Amazon Linux AMI (AL1).

Esta ramificação da plataforma foi substituída pelas ramificações da plataforma Docker rodando em AL2023 de 64 bits e Docker rodando em Amazon Linux 2 de 64 bits. Para obter mais informações, consulte [Deploying a GlassFish application to the Docker platform](#) (Implantar uma aplicação GlassFish na plataforma Docker).

Usar a plataforma Docker

O AWS Elastic Beanstalk pode iniciar ambientes do Docker compilando uma imagem descrita em `Dockerfile` ou acionando uma imagem do Docker remota. Se você estiver implantando uma imagem do Docker remota, você não precisará incluir um `Dockerfile`. Em vez disso, se você também estiver usando o Docker Compose, use um arquivo `docker-compose.yml`, que especifica uma imagem a ser usada e opções de configuração adicionais. Se você não estiver usando o Docker Compose com seus ambientes Docker, use um arquivo `Dockerrun.aws.json` em vez disso.

Tópicos

- [Pré-requisitos](#)
- [Colocar uma aplicação do Elastic Beanstalk em contêineres](#)
- [Testar um contêiner localmente](#)
- [Implantar um contêiner com um Dockerfile](#)
- [Testar uma imagem de Docker remota](#)

- [Implantar uma imagem do Docker remota no Elastic Beanstalk](#)
- [Limpar](#)
- [Executar localmente um ambiente do Docker com a EB CLI](#)
- [Configuração do Docker](#)

Pré-requisitos

Este tutorial pressupõe que você tenha algum conhecimento das operações básicas do Elastic Beanstalk, da Elastic Beanstalk Command Line Interface (EB CLI) e do Docker. Caso ainda não tenha, siga as instruções em [Conceitos básicos do Elastic Beanstalk](#) para iniciar seu primeiro ambiente do Elastic Beanstalk. Este tutorial usa a [EB CLI](#), mas também é possível criar ambientes e fazer upload de aplicações usando o console do Elastic Beanstalk.

Para concluir este tutorial, você precisará fazer o seguinte:

- Uma instalação local de trabalho do Docker. Para obter mais informações, consulte [Obter o Docker](#) no site de documentação do Docker.
- Acesso ao Docker Hub. Você precisará criar um ID do Docker para acessar o Docker Hub. Para obter mais informações, consulte [Compartilhar a aplicação](#) no site de documentação do Docker.

Para saber mais sobre como configurar ambientes do Docker em plataformas do Elastic Beanstalk, consulte [Configuração do Docker](#) neste mesmo capítulo.

Colocar uma aplicação do Elastic Beanstalk em contêineres

Para este exemplo, criamos uma imagem do Docker da aplicação demonstrativa Flask a partir do [Implantar uma aplicação Flask no Elastic Beanstalk](#). O aplicativo consiste em um arquivo principal, `application.py`. Também precisa de um `Dockerfile`. Coloque os dois arquivos na raiz de um diretório.

```
~/eb-docker-flask/  
|-- Dockerfile  
|-- application.py
```

Example `~/eb-docker-flask/application.py`

```
from flask import Flask
```

```
# Print a nice greeting
def say_hello(username = "World"):
    return '<p>Hello %s!</p>\n' % username

# Some bits of text for the page
header_text = '''
    <html>\n<head> <title>EB Flask Test</title> </head>\n<body>'''
instructions = '''
    <p><em>Hint</em>: This is a RESTful web service! Append a username
    to the URL (for example: <code>/Thelonious</code>) to say hello to
    someone specific.</p>\n'''
home_link = '<p><a href="/">Back</a></p>\n'
footer_text = '</body>\n</html>'

# Elastic Beanstalk looks for an 'application' that is callable by default
application = Flask(__name__)

# Add a rule for the index page
application.add_url_rule('/', 'index', (lambda: header_text +
    say_hello() + instructions + footer_text))

# Add a rule when the page is accessed with a name appended to the site
# URL
application.add_url_rule('/<username>', 'hello', (lambda username:
    header_text + say_hello(username) + home_link + footer_text))

# Run the application
if __name__ == "__main__":
    # Setting debug to True enables debug output. This line should be
    # removed before deploying a production application.
    application.debug = True
    application.run(host="0.0.0.0")
```

Example ~/eb-docker-flask/Dockerfile

```
FROM python:3.6
COPY . /app
WORKDIR /app
RUN pip install Flask==1.0.2
EXPOSE 5000
CMD ["python", "application.py"]
```


Testar um contêiner localmente

Use a CLI do Elastic Beanstalk (CLI do EB) para configurar seu repositório local para implantação no Elastic Beanstalk. Defina o `Dockerfile` do seu aplicativo na raiz do diretório.

```
~/eb-docker-flask$ eb init -p docker application-name
```

(Opcional) Use o comando `eb local run` para criar e executar seu contêiner localmente.

```
~/eb-docker-flask$ eb local run --port 5000
```

Note

Para saber mais sobre o comando `eb local`, consulte [the section called “eb local”](#). O comando não é compatível no Windows. Como alternativa, você pode criar e executar seu contêiner com os comandos `docker build` e `docker run`. Para obter mais informações, consulte a [Documentação do Docker](#).

(Opcional) Embora o contêiner esteja em execução, use o comando `eb local open` para visualizar o aplicativo em um navegador da web. Opcionalmente, abra <http://localhost:5000/> em um navegador da Web.

```
~/eb-docker-flask$ eb local open
```

Implantar um contêiner com um Dockerfile

Depois de testar a aplicação localmente, implante-a em um ambiente do Elastic Beanstalk. O Elastic Beanstalk usa as instruções em seu `Dockerfile` para criar e executar a imagem.

Use o comando `eb create` para criar um ambiente e implante seu aplicativo.

```
~/eb-docker-flask$ eb create environment-name
```

Depois que o ambiente for iniciado, use o comando `eb open` para visualizá-lo em um navegador da web.

```
~/eb-docker-flask$ eb open
```

Testar uma imagem de Docker remota

Em seguida, criamos uma imagem de Docker do aplicativo Flask da seção anterior e a enviamos para o Docker Hub.

Observações

- As etapas a seguir criam uma imagem do Docker disponível publicamente.
- Você usará comandos do Docker da instalação local do Docker, juntamente com as credenciais do Docker Hub. Para obter mais informações, consulte a seção [Pré-requisitos](#) anterior.

Depois de criarmos e enviarmos nossa imagem, poderemos implantá-la no Elastic Beanstalk com um arquivo `docker-compose.yml`, se você estiver usando o Docker Compose com o ambiente Docker. Se você não estiver usando o Docker Compose com seus ambientes Docker, use um arquivo `DockerRun.aws.json` em vez disso. Para criar uma imagem de Docker do aplicativo Flask e a enviar para o Docker Hub, execute os comandos a seguir. Estamos usando o mesmo diretório do exemplo anterior, mas você poderá usar qualquer diretório com o código do seu aplicativo. Insira seu ID do Docker para `docker-id` para iniciar sessão no Docker Hub.

```
~/eb-docker-flask$ docker build -t docker-id/beanstalk-flask:latest .  
~/eb-docker-flask$ docker push docker-id/beanstalk-flask:latest
```

Note

Antes de enviar sua imagem, talvez seja necessário executar `docker login`. Suas credenciais do Docker Hub serão solicitadas se você executar o comando sem parâmetros.

Se você estiver usando a ferramenta Docker Compose para gerenciar seu ambiente do Docker, agora poderá implantar a aplicação usando apenas um arquivo `docker-compose.yml`. Para saber mais sobre os arquivos `docker-compose.yml`, consulte [Configuração do Docker](#).

Se você não estiver usando o Docker Compose, use um arquivo `DockerRun.aws.json`. Para obter mais informações, consulte [Implantar usando o DockerRun.aws.json v1 \(sem o Docker Compose\)](#).

Faça um novo diretório e crie um arquivo `docker-compose.yml`.

Example `~/remote-docker/docker-compose.yml`

```
version: '3.8'
services:
  beanstalk-flask:
    image: "username/beanstalk-flask"
    ports:
      - "80:5000"
```

Implantar usando o `Dockerrun.aws.json` v1 (sem o Docker Compose)

Se você não estiver usando a ferramenta Docker Compose para gerenciar seu ambiente do Docker, agora poderá implantar a aplicação usando apenas um arquivo `Dockerrun.aws.json`. Para saber mais sobre os arquivos `Dockerrun.aws.json`, consulte [Configuração para plataformas Docker \(sem Docker Compose\)](#).

Faça um novo diretório e crie um arquivo `Dockerrun.aws.json`.

Example `~/remote-docker/Dockerrun.aws.json`

```
{
  "AWSEBDockerrunVersion": "1",
  "Image": {
    "Name": "username/beanstalk-flask",
    "Update": "true"
  },
  "Ports": [
    {
      "ContainerPort": "5000"
    }
  ]
}
```

Use a CLI do EB para configurar o repositório local para implantação no Elastic Beanstalk.

```
~/remote-docker$ eb init -p docker application-name
```

(Opcional) Use `eb local run` para compilar e executar seu contêiner localmente. Para saber mais sobre o comando `eb local`, consulte [eb local](#).

```
~/remote-docker$ eb local run --port 5000
```

(Opcional) Embora o contêiner esteja em execução, use o comando `eb local open` para visualizar o aplicativo em um navegador da web. Opcionalmente, abra <http://localhost:5000/> em um navegador da Web.

```
~/remote-docker$ eb local open
```

Implantar uma imagem do Docker remota no Elastic Beanstalk

Depois de testar o contêiner localmente, implante-o em um ambiente do Elastic Beanstalk. O Elastic Beanstalk usa o arquivo `docker-compose.yml` para extrair e executar sua imagem se você estiver usando o Docker Compose. Caso contrário, o Elastic Beanstalk usa o `DockerRun.aws.json` em vez disso.

Use a EB CLI para criar um ambiente e implante sua imagem.

```
~/remote-docker$ eb create environment-name
```

Assim que seu ambiente for ativado, use `eb open` para visualizá-lo em um navegador da Web.

```
~/remote-docker$ eb open
```

Limpar

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. O Elastic Beanstalk termina todos os recursos da AWS associados ao seu ambiente, como [instâncias do Amazon EC2](#), [instâncias de banco de dados](#), [balanceadores de carga](#), grupos de segurança e [alarmes](#).

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

ou com a EB CLI:

```
~/remote-docker$ eb terminate environment-name
```

Executar localmente um ambiente do Docker com a EB CLI

É possível usar a Elastic Beanstalk Command Line Interface (EB CLI) para executar os contêineres do Docker configurados na aplicação do AWS Elastic Beanstalk localmente. A EB CLI usa o arquivo de configuração do Docker (`Dockerfile` ou `DockerRun.aws.json`) e o código-fonte no diretório do projeto para executar sua aplicação localmente no Docker.

Tópicos

- [Pré-requisitos para executar aplicativos do Docker localmente](#)
- [Preparar um aplicativo do Docker para usar com a EB CLI](#)
- [Executar localmente um aplicativo do Docker](#)
- [Limpeza após a execução local de um aplicativo do Docker](#)

Pré-requisitos para executar aplicativos do Docker localmente

- SO Linux ou Mac OS X
- [Versão 3.3 ou posterior da EB CLI](#)

Execute `eb init` no diretório do projeto para inicializar um repositório da EB CLI. Se você nunca usou a CLI do EB, consulte [Gerenciar ambientes do Elastic Beanstalk com a CLI do EB](#).

- [Versão 1.6 ou posterior do Docker](#)

Adicione-se ao grupo `docker`, faça `logout` e, em seguida, faça login novamente para garantir que você pode executar comandos do Docker sem `sudo`:

```
$ sudo usermod -a -G docker $USER
```

Execute `docker ps` para verificar se o daemon do Docker está ativo e em execução:

```
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
```

- Um aplicativo do Docker

Se você não tiver um aplicativo do Docker em uma pasta do projeto em sua máquina local, consulte [Implantar aplicações do Elastic Beanstalk a partir de contêineres do Docker](#) para obter uma introdução sobre como usar o Docker com o AWS Elastic Beanstalk.

- Perfil do Docker (opcional)

Se o seu aplicativo usa imagens de docker que estão em um repositório privado, execute `docker login` e siga as instruções para criar um perfil de autenticação.

- `w3m` (opcional)

O `w3m` é um navegador da web que você pode usar para visualizar seu aplicativo web em execução dentro de um terminal de linha de comando com `eb local run`. Se você estiver usando a linha de comando em um ambiente de desktop, o `w3m` não será necessário.

Os contêineres do Docker são executados localmente sem emular os recursos da AWS que são provisionados quando uma aplicação é implantada no Elastic Beanstalk incluindo grupos de segurança e níveis de dados ou operador.

É possível configurar os contêineres locais para se conectarem a um banco de dados passando a string de conexão necessária ou outras variáveis com a opção `envvars`, mas você deve garantir que todos os recursos na AWS sejam acessíveis a partir da máquina local [abrindo as portas apropriadas](#) em seus grupos de segurança atribuídos ou anexando um [gateway padrão](#) ou um [endereço IP elástico](#).

Preparar um aplicativo do Docker para usar com a EB CLI

Prepare o arquivo de configuração do Docker e os dados de origem como se fosse implantá-los no Elastic Beanstalk. Este tópico usa o exemplo do proxy PHP e nginx do [Tutorial do Docker de vários contêineres](#), que também está incluído neste capítulo. Você pode usar os mesmos comandos com qualquer aplicação do Docker, Docker de vários contêineres ou Docker pré-configurado.

Executar localmente um aplicativo do Docker

Execute seu aplicativo do Docker localmente com o comando `eb local run` do diretório do projeto:

```
~/project$ eb local run
Creating elasticbeanstalk_phpapp_1...
Creating elasticbeanstalk_nginxproxy_1...
Attaching to elasticbeanstalk_phpapp_1, elasticbeanstalk_nginxproxy_1
phpapp_1      | [23-Apr-2015 23:24:25] NOTICE: fpm is running, pid 1
phpapp_1      | [23-Apr-2015 23:24:25] NOTICE: ready to handle connections
```

A EB CLI lê a configuração do Docker e executa os comandos do Docker necessários para executar seu aplicativo. A primeira vez que você executa um projeto localmente, o Docker baixa as imagens de um repositório remoto e as armazena na sua máquina local. Esse processo pode levar alguns minutos.

Note

O comando `eb local run` usa dois parâmetros opcionais, `port` e `envvars`. Para substituir a porta padrão de um aplicativo do Docker, use a opção `port`:

```
$ eb local run --port 8080
```

Esse comando fala para a EB CLI usar a porta 8080 no host e mapeá-la para a porta exposta no contêiner. Se você não especificar uma porta, a EB CLI usará a porta do contêiner para o host. Esta opção só funciona com aplicativos que usam a plataforma do Docker.

Para passar variáveis do ambiente para os contêineres de aplicativo, use a opção `envvars`:

```
$ eb local run --envvars RDS_HOST=$RDS_HOST,RDS_DB=$RDS_DB,RDS_USER=$RDS_USER,RDS_PASS=$RDS_PASS
```

Use variáveis do ambiente para configurar uma conexão de banco de dados, definir opções de depuração ou passar segredos de forma segura para seu aplicativo. Para obter mais informações sobre as opções compatíveis com os subcomandos `eb local`, consulte [eb local](#).

Depois que os contêineres estiverem ativos e em execução no Docker, eles estarão prontos para receber solicitações de clientes. O processo `eb local` permanece aberto, desde que os contêineres estejam em execução. Se você precisar interromper o processo e os contêineres, pressione `Ctrl+C`.

Abra um segundo terminal para executar comandos adicionais enquanto o processo `eb local` está em execução. Use `eb local status` para visualizar o status do seu aplicativo:

```
~/project$ eb local status
Platform: 64bit Amazon Linux 2014.09 v1.2.1 running Multi-container Docker 1.3.3
(Generic)
Container name: elasticbeanstalk_nginxproxy_1
Container ip: 127.0.0.1
Container running: True
Exposed host port(s): 80
Full local URL(s): 127.0.0.1:80

Container name: elasticbeanstalk_phpapp_1
Container ip: 127.0.0.1
Container running: True
Exposed host port(s): None
Full local URL(s): None
```

Você pode usar `docker ps` para ver o status dos contêineres do ponto de vista do Docker:

```
~/project$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	
STATUS	PORTS	NAMES		
6a8e71274fed	nginx:latest	"nginx -g 'daemon of	9 minutes ago	Up 9
minutes	0.0.0.0:80->80/tcp,	elasticbeanstalk_nginxproxy_1		
82cbf620bdc1	php:fpm	"php-fpm"	9 minutes ago	Up 9
minutes	9000/tcp	elasticbeanstalk_phpapp_1		

Em seguida, visualize seu aplicativo em ação com `eb local open`:

```
~/project$ eb local open
```


Este comando abre o aplicativo no navegador da web padrão. Se você estiver executando um terminal em um ambiente de desktop, poderá ser o Firefox, o Safari ou o Google Chrome. Se você estiver executando um terminal em um ambiente sem comando ou através de uma conexão SSH, será usado um navegador de linha de comando, como o w3m, se estiver disponível.

Volte para o terminal executando o processo do aplicativo por um momento e observe a saída adicional:

```
phpapp_1 | 172.17.0.36 - 21/Apr/2015:23:46:17 +0000 "GET /index.php" 200
```

Isso mostra que o aplicativo web no contêiner do Docker recebeu uma solicitação HTTP GET para index.php que foi retornada com êxito com um status 200 (sem erro).

Execute `eb local logs` para ver onde a EB CLI grava os logs.

```
~/project$ eb local logs
Elastic Beanstalk will write logs locally to /home/user/project/.elasticbeanstalk/logs/local.
Logs were most recently created 3 minutes ago and written to /home/user/project/.elasticbeanstalk/logs/local/150420_234011665784.
```

Limpeza após a execução local de um aplicativo do Docker

Ao concluir os testes locais do seu aplicativo, você pode interromper os aplicativos e remover as imagens baixadas pelo Docker ao usar `eb local run`. A remoção das imagens é opcional. Você pode querer mantê-las para uso futuro.

Volte para o terminal que esta executando o processo `eb local` e pressione `Ctrl+C` para interromper a aplicação:

```
^CGracefully stopping... (press Ctrl+C again to force)
Stopping elasticbeanstalk_nginxproxy_1...
Stopping elasticbeanstalk_phpapp_1...

Aborting.
[1]+  Exit 5                  eb local run
```

A EB CLI tenta interromper tranquilamente cada contêiner em execução com comandos do Docker. Se você precisar interromper um processo imediatamente, pressione `Ctrl+C` novamente.

Depois que você interrompe os aplicativos, os contêineres do Docker também devem parar de executar. Verifique isso com `docker ps`:

```
$ docker ps --all
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
73d515d99d2a      nginx:latest       "nginx -g 'daemon of  21 minutes ago
Exited (0) 11 minutes ago                elasticbeanstalk_nginxproxy_1
7061c76220de      php:fpm            "php-fpm"          21 minutes ago
Exited (0) 11 minutes ago                elasticbeanstalk_phpapp_1
```

A opção `all` mostra contêineres interrompidos (se você omitiu essa opção, a saída será em branco). No exemplo acima, o Docker mostra que os dois contêineres saíram com um status 0 (sem erro).

Se tiver terminado de usar o Docker e os comandos locais da EB CLI, você poderá remover as imagens do Docker de sua máquina local para economizar espaço.

Para remover imagens do Docker de sua máquina local

1. Visualize as imagens que você baixou usando `docker images`:

```
$ docker images
REPOSITORY          TAG                 IMAGE ID           CREATED
VIRTUAL SIZE
php                 fpm                68bc5150cffc      1 hour ago
414.1 MB
nginx              latest             637d3b2f5fb5     1 hour ago
93.44 MB
```

2. Remova os dois contêineres do Docker com `docker rm`:

```
$ docker rm 73d515d99d2a 7061c76220de
73d515d99d2a
7061c76220de
```

3. Remova as imagens com `docker rmi`:

```
$ docker rmi 68bc5150cffc 637d3b2f5fb5
Untagged: php:fpm
Deleted: 68bc5150cffc0526c66b92265c3ed8f2ea50f3c71d266aa655b7a4d20c3587b0
Untagged: nginx:latest
Deleted: 637d3b2f5fb5c4f70895b77a9e76751a6e7670f4ef27a159dad49235f4fe61e0
```

Configuração do Docker

Esta seção descreve como preparar a imagem e o contêiner do Docker para implantação no Elastic Beanstalk.

Ambiente Docker com Docker Compose

Esta seção descreve como preparar a imagem e o contêiner do Docker para implantação no Elastic Beanstalk. Qualquer aplicação Web implantada no Elastic Beanstalk em um ambiente Docker deve incluir um arquivo `docker-compose.yml` se você também usar a ferramenta Docker Compose. Você pode implantar sua aplicação Web como um serviço em contêineres no Elastic Beanstalk realizando uma das seguintes ações:

- Crie um arquivo `docker-compose.yml` para implantar uma imagem do Docker a partir de um repositório hospedado do Elastic Beanstalk. Nenhum outro arquivo será necessário se todas as implantações forem originadas de imagens em repositórios públicos. (Se a implementação tiver origem em uma imagem de um repositório privado, será necessário incluir arquivos de configuração adicionais para autenticação. Para obter mais informações, consulte [Uso de imagens de um repositório privado](#).) Para obter mais informações sobre o arquivo `docker-compose.yml`, consulte [Referência de arquivo do Compose](#) no site do Docker.
- Crie um `Dockerfile` para que o Elastic Beanstalk crie e execute uma imagem personalizada. Esse arquivo é opcional, dependendo dos requisitos de implantação. Para obter mais informações sobre `Dockerfile`, consulte [Referência do arquivo do Docker](#) no site do Docker.
- Crie um arquivo `.zip` que contém os arquivos do aplicativo, todas as dependências de arquivo do aplicativo, o `Dockerfile` e o arquivo `docker-compose.yml`. Se você usar a CLI do EB para implantar a aplicação, ela criará um arquivo `.zip` para você. Os dois arquivos devem estar na raiz, ou no nível superior do arquivamento `.zip`.

Se você usar apenas um arquivo `docker-compose.yml` para implantar a aplicação, não precisará criar um arquivo `.zip`.

Este tópico é uma referência de sintaxe. Para obter procedimentos detalhados sobre como iniciar ambientes Docker usando o Elastic Beanstalk, consulte [Usar a plataforma Docker](#).

Para saber mais sobre o Docker Compose e como instalá-lo, consulte [Visão geral do Docker Compose](#) e [Instalar o Docker Compose](#) no site do Docker.

Note

Se você não usar o Docker Compose para configurar seus ambientes Docker, também não deverá usar o arquivo `docker-compose.yml`. Em vez disso, use o arquivo `Dockerrun.aws.json` ou `Dockerfile` ou ambos.

Para obter mais informações, consulte [the section called “Configuração para plataformas Docker \(sem Docker Compose\)”](#).

Usar imagens de um repositório privado

O Elastic Beanstalk deve se autenticar com o registro online que hospeda o repositório privado antes de extrair e implantar as imagens de um repositório privado. Fornecemos exemplos de duas opções para armazenar e recuperar credenciais para seu ambiente Elastic Beanstalk para autenticação em um repositório.

- A AWS Secrets Manager
- O arquivo `Dockerrun.aws.json v3`

Usar o AWS Secrets Manager

Você pode configurar o Elastic Beanstalk para fazer login no repositório privado antes que ele inicie o processo de implantação. Isso permite que o Elastic Beanstalk acesse as imagens do repositório e implante essas imagens no ambiente do Elastic Beanstalk.

Essa configuração inicia eventos na fase de pré-compilação do processo de implantação do Elastic Beanstalk. Você configurou isso no diretório de configuração [.ebextensions](#). A configuração usa scripts de [hook de plataforma](#) que chamam `docker login` para autenticar no registro online que hospeda o repositório privado. Veja a seguir detalhes das etapas de configuração.

Para configurar o Elastic Beanstalk para autenticar em seu repositório privado com o AWS Secrets Manager

Note

Permissões específicas devem ser concedidas para concluir essas etapas. Para obter mais informações, consulte as referências a seguir.

- Na Etapa 2, você precisará de permissões para criar um segredo. Para obter mais informações, consulte [Exemplo: Permissão para criar segredos](#) no AWS Secrets Manager Guia do usuário.
- Na Etapa 3, você precisará de permissões para recuperar segredos usando referências dinâmicas `secretsmanager`. Para obter mais informações, consulte [Exemplo: permissão para recuperar valores de segredos](#) no Guia do usuário do AWS Secrets Manager.

1. Crie sua estrutura de diretórios `.ebextensions` da seguinte forma.

```
### .ebextensions
#   ### env.config
### .platform
#   ### confighooks
# #   ### prebuild
# #       ### 01login.sh
#   ### hooks
#       ### prebuild
#           ### 01login.sh
### docker-compose.yml
```

2. Use o AWS Secrets Manager para salvar as credenciais do repositório privado para que o Elastic Beanstalk possa recuperar suas credenciais quando necessário. Para tanto, execute o comando [AWS CLI create-secret](#) do Secrets Manager.

```
aws secretsmanager create-secret \
    --name MyTestSecret \
    --description "My image repo credentials created with the CLI." \
    --secret-string "{\"USER\": \"EXAMPLE-USERNAME\", \"PASSWD\": \"EXAMPLE-PASSWORD\"}"
```

3. Crie o seguinte arquivo `env.config` e coloque-o no diretório `.ebextensions`, como mostrado na estrutura de diretórios anterior. Essa configuração usa o namespace [aws:elasticbeanstalk:application:environment](#) para inicializar as variáveis de ambiente `USER` e `PASSWD` Elastic Beanstalk usando referências dinâmicas ao AWS Secrets Manager. Para obter mais informações sobre referências `secretsmanager` dinâmicas, consulte [Recuperar um segredo AWS Secrets Manager em um recurso AWS CloudFormation](#) no AWS Secrets Manager Guia do usuário.

Note

USER e PASSWD no script devem corresponder às mesmas strings usadas no comando `secretsmanager create-secret` anterior.

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    USER: '{{resolve:secretsmanager:MyTestSecret:SecretString:USER}}'
    PASSWD: '{{resolve:secretsmanager:MyTestSecret:SecretString:PASSWD}}'
```

4. Crie o seguinte arquivo de script `01login.sh` e coloque-o nos seguintes diretórios (também mostrado na estrutura de diretórios anterior):

- `.platform/confighooks/prebuild`
- `.platform/hooks/prebuild`

```
### example 01login.sh
#!/bin/bash
USER=/opt/elasticbeanstalk/bin/get-config environment -k USER
/opt/elasticbeanstalk/bin/get-config environment -k PASSWD | docker login -u $USER
--password-stdin
```

O script `01login.sh` chama o script de plataforma [get-config](#) para recuperar as credenciais do repositório e, em seguida, efetuar login no repositório. Ele armazena o nome do usuário no script variável `USER`. Na próxima linha, ele recupera a senha. Em vez de armazenar a senha em um script variável, o script envia a senha diretamente para o comando `docker login` no fluxo de entrada `stdin`. A opção `--password-stdin` usa o fluxo de entrada para que você não precise armazenar a senha em uma variável. Para obter mais informações sobre autenticação com a interface de linha de comando do Docker, consulte [docker login](#) (login do docker) no site da documentação do Docker.

Observações

- Todos os arquivos de script devem ter permissão de execução. Use `chmod +x` para definir a permissão de execução em seus arquivos de hook. Em todas as versões da

plataforma baseadas no Amazon Linux 2 que foram lançadas a partir de 29 de abril de 2022, o Elastic Beanstalk concede automaticamente permissões de execução para todos os scripts de hook da plataforma. Nesse caso, não é necessário conceder manualmente as permissões de execução. Para obter uma lista dessas versões da plataforma, consulte as notas de versão [29 de abril de 2022 - Plataforma Linux](#) no Guia de notas de versão do AWS Elastic Beanstalk.

- Os arquivos de hook podem ser arquivos binários ou arquivos de script que começam com uma linha `#!/` contendo seu caminho de interpretador, como `#!/bin/bash`.
- Para obter mais informações, consulte [the section called “Hooks de plataforma”](#) em Estender as plataformas do Elastic Beanstalk Linux.

Após a autenticação do Elastic Beanstalk com o registro online que hospeda o repositório privado, você pode obter e implantar suas imagens.

Usar o arquivo `Dockerrun.aws.json v3`

Esta seção descreve outra abordagem para autenticar o Elastic Beanstalk em um repositório privado. Com essa abordagem, gere um arquivo de autenticação com o comando do Docker e, depois, carregue o arquivo de autenticação em um bucket do Amazon S3. Você também deve incluir as informações do bucket no arquivo `Dockerrun.aws.json v3`.

Como gerar e fornecer um arquivo de autenticação ao Elastic Beanstalk

1. Gere um arquivo de autenticação com o comando `docker login`. Para repositórios no Docker Hub, execute `docker login`:

```
$ docker login
```

Para outros registros, inclua o URL do servidor de registro:

```
$ docker login registry-server-url
```

Note

Se o ambiente do Elastic Beanstalk usar a versão da plataforma Docker da AMI do Amazon Linux (que precede o Amazon Linux 2), leia as informações relevantes em

[the section called “Configuração do Docker na AMI do Amazon Linux \(que precede o Amazon Linux 2\)”](#).

Para obter mais informações sobre o arquivo de autenticação, consulte [Store images on Docker Hub](#) e [docker login](#) no site do Docker.

2. Faça upload de uma cópia do arquivo de autenticação chamado `.dockercfg` em um bucket do Amazon S3 seguro.
 - O bucket do Amazon S3 deve ser hospedado na mesma Região da AWS do ambiente que o está usando. O Elastic Beanstalk não pode fazer download de arquivos de um bucket do Amazon S3 hospedado em outras regiões.
 - Conceda permissões para a operação `s3:GetObject` à função do IAM no perfil da instância. Para obter mais informações, consulte [Gerenciar perfis de instância do Elastic Beanstalk](#).
3. Inclua as informações do bucket do Amazon S3 no parâmetro `Authentication` no arquivo `Dockerrun.aws.json v3`.

Veja a seguir um exemplo de um arquivo `Dockerrun.aws.json v3`.

```
{
  "AWSEBDockerrunVersion": "3",
  "Authentication": {
    "bucket": "DOC-EXAMPLE-BUCKET",
    "key": "mydockercfg"
  }
}
```

Note

O parâmetro `AWSEBDockerrunVersion` indica a versão do arquivo `Dockerrun.aws.json`.

- A plataforma Docker do Amazon Linux 2 usa o arquivo `Dockerrun.aws.json v3` para ambientes que usam o Docker Compose. Ela usa o arquivo `Dockerrun.aws.json v1` para ambientes que não usam o Docker Compose.
- A plataforma Docker de vários contêineres da AMI do Amazon Linux usa o arquivo `Dockerrun.aws.json v2`.

Depois que o Elastic Beanstalk puder autenticar com o registro online que hospeda o repositório privado, as imagens poderão ser implantadas e extraídas.

Criar imagens personalizadas com um Dockerfile

Será necessário criar um Dockerfile se você ainda não tiver uma imagem existente hospedada em um repositório.

O seguinte snippet é um exemplo do Dockerfile. Quando você segue as instruções em [Usar a plataforma Docker](#), pode fazer upload deste Dockerfile como gravado. O Elastic Beanstalk executa o jogo 2048 quando você usa este Dockerfile.

Para obter mais informações sobre instruções que podem ser incluídas no Dockerfile, consulte a [referência do Dockerfile](#) no site do Docker.

```
FROM ubuntu:12.04

RUN apt-get update
RUN apt-get install -y nginx zip curl

RUN echo "daemon off;" >> /etc/nginx/nginx.conf
RUN curl -o /usr/share/nginx/www/master.zip -L https://codeload.github.com/gabrielecirulli/2048/zip/master
RUN cd /usr/share/nginx/www/ && unzip master.zip && mv 2048-master/* . && rm -rf 2048-master master.zip

EXPOSE 80

CMD ["/usr/sbin/nginx", "-c", "/etc/nginx/nginx.conf"]
```

Note

De um único Dockerfile, você pode executar compilações em vários estágios para produzir imagens menores com uma redução significativa de complexidade. Para obter mais informações, consulte [Use multi-stage builds](#) (Usar compilações em vários estágios) no site de documentação do Docker.

Configuração para plataformas Docker (sem Docker Compose)

Se o ambiente Docker do Elastic Beanstalk não usar o Docker Compose, leia as informações adicionais nas seções a seguir.

Configuração da plataforma Docker - sem Docker Compose

Qualquer aplicação Web implantada no Elastic Beanstalk em um ambiente Docker deve incluir um arquivo `Dockerfile` ou `Dockerrun.aws.json`. É possível implantar a aplicação Web de um contêiner do Docker no Elastic Beanstalk realizando uma destas ações:

- Crie um `Dockerfile` para que o Elastic Beanstalk crie e execute uma imagem personalizada.
- Crie um arquivo `Dockerrun.aws.json` para implantar uma imagem do Docker a partir de um repositório hospedado do Elastic Beanstalk.
- Crie um arquivo `.zip` que contém os arquivos do aplicativo, todas as dependências de arquivo do aplicativo, o `Dockerfile` e o arquivo `Dockerrun.aws.json`. Se você usar a CLI do EB para implantar a aplicação, ela criará um arquivo `.zip` para você.

Se você usar apenas um `Dockerfile` ou apenas um arquivo `Dockerrun.aws.json` para implantar seu aplicativo, não precisará criar um arquivo `.zip`.

Este tópico é uma referência de sintaxe. Para obter os procedimentos detalhados sobre como executar ambientes do Docker, consulte [Usar a plataforma Docker](#).

Dockerrun.aws.json v1

Um arquivo `Dockerrun.aws.json` descreve como implantar uma imagem do Docker remota como uma aplicação do Elastic Beanstalk. Este arquivo JSON é específico do Elastic Beanstalk. Se o seu aplicativo é executado em uma imagem disponível em um repositório hospedado, você pode especificar a imagem em um arquivo `Dockerrun.aws.json v1` e omitir o `Dockerfile`.

Chaves e valores válidos para o arquivo `Dockerrun.aws.json v1` incluem as seguintes operações:

AWSEBDockerrunVersion

(Obrigatório) Especifica o número da versão como o valor 1 para ambientes Docker de um único contêiner.

Autenticação

(Obrigatório apenas para repositórios privados) Especifica o objeto do Amazon S3 que armazena o arquivo `.dockercfg`.

Consulte [Usar imagens de um repositório privado](#).

Imagem

Especifica a imagem de base do Docker em um repositório Docker existente do qual você está criando um contêiner do Docker. Especifique o valor da chave `Name` no formato `<organization>/<image name>` para imagens no Docker Hub ou `<site>/<organization name>/<image name>` para outros sites.

Quando você especifica uma imagem no arquivo `Dockerrun.aws.json`, cada instância no ambiente do Elastic Beanstalk executa `docker pull` para executar a imagem. Opcionalmente, inclua a chave `Update`. O valor padrão é `true` e orienta o Elastic Beanstalk a verificar o repositório, enviar todas as atualizações para a imagem e substituir todas as imagens armazenadas em cache.

Ao usar um `Dockerfile`, não especifique a chave `Image` no arquivo `Dockerrun.aws.json`. O Elastic Beanstalk sempre cria e usa a imagem descrita no `Dockerfile` quando um está presente.

Portas

(Obrigatório quando você especifica a chave `Image`) Lista as portas para expor no contêiner do Docker. O Elastic Beanstalk usa o valor `ContainerPort` para conectar o contêiner do Docker ao proxy reverso em execução no host.

É possível especificar várias portas de contêiner, mas o Elastic Beanstalk usa apenas a primeira porta. Ele usa essa porta para conectar seu contêiner ao proxy reverso do host e rotear solicitações da Internet pública. Se você estiver usando um `Dockerfile`, o primeiro valor `ContainerPort` deverá corresponder à lista `EXPOSE` do `Dockerfile`.

Opcionalmente, você pode especificar uma lista de portas em `HostPort`. As entradas `HostPort` especificam as portas de host aos quais os valores de `ContainerPort` são mapeados. Se você não especificar um valor `HostPort`, ele assumirá como padrão o valor `ContainerPort`.

```
{
```

```
"Image": {
  "Name": "image-name"
},
"Ports": [
  {
    "ContainerPort": 8080,
    "HostPort": 8000
  }
]
}
```

Volumes

Mapeie volumes de uma instância do EC2 para o contêiner do Docker. Especifique um ou mais conjuntos de volumes para mapear.

```
{
  "Volumes": [
    {
      "HostDirectory": "/path/inside/host",
      "ContainerDirectory": "/path/inside/container"
    }
  ]
  ...
}
```

Registro em log

Especifique o diretório dentro do contêiner no qual a aplicação grava logs. O Elastic Beanstalk faz upload de todos os logs neste diretório para o Amazon S3 quando você solicitar logs finais ou de pacote. Se você alternar logs para uma pasta chamada `rotated` nesse diretório, também poderá configurar o Elastic Beanstalk para fazer upload de logs alternados para o Amazon S3 para armazenamento permanente. Para obter mais informações, consulte [Visualizar logs de instâncias do Amazon EC2 no ambiente do Elastic Beanstalk](#).

Comando

Especifique um comando a ser executado no contêiner. Se você especificar um Entrypoint, Command será adicionado como um argumento para Entrypoint. Para obter mais informações, consulte [CMD](#) na documentação do Docker.

Entrypoint

Especifique um comando padrão a ser executado quando o contêiner for iniciado. Para obter mais informações, consulte [ENTRYPOINT](#) na documentação do Docker.

O seguinte snippet é um exemplo que mostra a sintaxe do arquivo `Dockerrun.aws.json` para um único contêiner.

```
{
  "AWSEBDockerrunVersion": "1",
  "Image": {
    "Name": "janedoe/image",
    "Update": "true"
  },
  "Ports": [
    {
      "ContainerPort": "1234"
    }
  ],
  "Volumes": [
    {
      "HostDirectory": "/var/app/mydb",
      "ContainerDirectory": "/etc/mysql"
    }
  ],
  "Logging": "/var/log/nginx",
  "Entrypoint": "/app/bin/myapp",
  "Command": "--argument"
}
```

Você pode fornecer ao Elastic Beanstalk apenas com o arquivo `Dockerrun.aws.json` ou com um arquivamento `.zip` que contém os arquivos `Dockerrun.aws.json` e `Dockerfile`. Quando você fornece os dois arquivos, `Dockerfile` descreverá a imagem de Docker, e o arquivo `Dockerrun.aws.json` fornecerá informações adicionais para a implantação conforme descritas posteriormente nesta seção.

Note

Os dois arquivos devem estar na raiz, ou no nível superior do arquivamento `.zip`. Não crie o arquivamento em um diretório que contém os arquivos. Em vez disso, navegue para esse diretório e crie o arquivamento.

Quando você fornecer os dois arquivos, não especifique uma imagem no arquivo `Dockerrun.aws.json`. O Elastic Beanstalk cria e usa a imagem descrita no `Dockerfile` e ignora a imagem especificada no arquivo `Dockerrun.aws.json`.

Usar imagens de um repositório privado

Adicione as informações sobre o bucket do Amazon S3 que contém o arquivo de autenticação no parâmetro `Authentication` do arquivo `Dockerrun.aws.json v1`. Certifique-se de que o parâmetro `Authentication` contenha uma chave e um bucket válidos do Amazon S3. O bucket do Amazon S3 deve ser hospedado na mesma Região da AWS do ambiente que o está usando. O Elastic Beanstalk não faz download de arquivos de buckets do Amazon S3 hospedados em outras regiões.

Para obter informações sobre a geração e o upload do arquivo de autenticação, consulte [Usar imagens de um repositório privado](#).

O exemplo a seguir mostra o uso de um arquivo de autenticação chamado `mydockercfg` em um bucket chamado `DOC-EXAMPLE-BUCKET` para usar uma imagem privada em um registro de terceiros.

```
{
  "AWSEBDockerrunVersion": "1",
  "Authentication": {
    "Bucket": "DOC-EXAMPLE-BUCKET",
    "Key": "mydockercfg"
  },
  "Image": {
    "Name": "quay.io/johndoe/private-image",
    "Update": "true"
  },
  "Ports": [
    {
      "ContainerPort": "1234"
    }
  ],
}
```

```
"Volumes": [  
  {  
    "HostDirectory": "/var/app/mydb",  
    "ContainerDirectory": "/etc/mysql"  
  }  
],  
"Logging": "/var/log/nginx"  
}
```

Usar a ramificação da plataforma Amazon ECS

Este tópico aborda tanto a ramificação da plataforma Amazon ECS no Amazon Linux 2 quanto a ramificação da plataforma que ele substitui, Docker de vários contêineres on AL1 (também gerenciada pelo ECS). Salvo indicação em contrário, todas as informações neste tópico se aplicam a ambas as ramificações da plataforma.

Note

Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas.

Migrar do Docker de vários contêineres no AL1

Se você estiver usando atualmente o Docker de vários contêineres desativado em execução na ramificação da plataforma AL1, poderá migrar para a mais recente ramificação da plataforma ECS em execução no AL2023. A ramificação mais recente da plataforma é compatível com todos os recursos da ramificação da plataforma que foi retirada. Não é necessária nenhuma alteração no código-fonte. Para obter mais informações, consulte [Migrar o Docker de vários contêineres em execução no Amazon Linux para o ECS no Amazon Linux 2023](#).

Tópicos

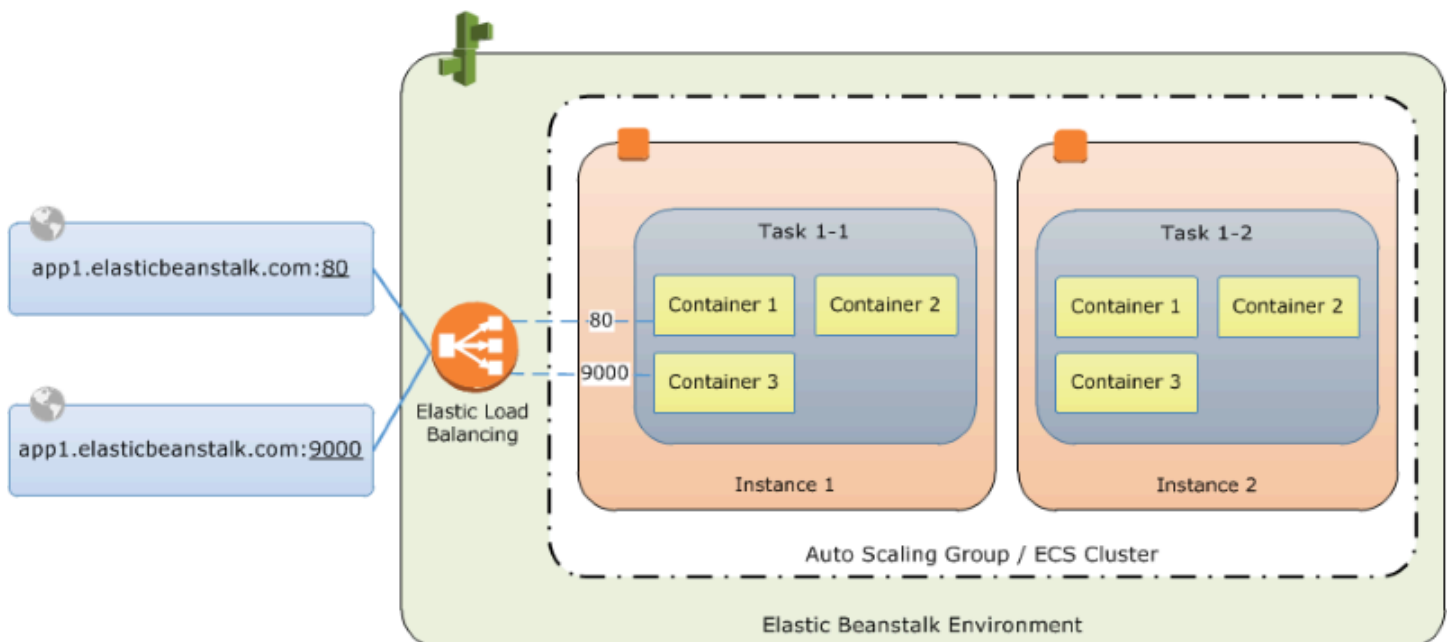
- [Plataforma Docker gerenciada pelo ECS](#)
- [Dockerrun.aws.json file](#)
- [Imagens de Docker](#)
- [Função de instância de contêiner](#)
- [Recursos do Amazon ECS criados pelo Elastic Beanstalk](#)
- [Usar vários listeners do Elastic Load Balancing](#)

- [Falha nas implantações de contêiner](#)
- [Configuração do Docker gerenciado pelo ECS](#)
- [Ambientes do Docker gerenciado pelo ECS com o console do Elastic Beanstalk](#)
- [Migrar o Docker de vários contêineres em execução no Amazon Linux para o ECS no Amazon Linux 2023](#)
- [\(Herdado\) Migrar da ramificação da plataforma Docker de vários contêineres em execução no Amazon Linux para a ramificação da plataforma Docker em execução no Amazon Linux 2](#)

Plataforma Docker gerenciada pelo ECS

O Elastic Beanstalk usa o Amazon Elastic Container Service (Amazon ECS) para coordenar implantações de contêiner em ambientes do Docker gerenciado pelo ECS. O Amazon ECS fornece ferramentas para gerenciar um cluster de instâncias executando contêineres do Docker. O Elastic Beanstalk cuida das tarefas do Amazon ECS, incluindo criação de cluster, definição de tarefas e execução. Cada uma das instâncias do ambiente executa o mesmo conjunto de contêineres que são definidos em um arquivo `DockerRun.aws.json v2`. Para tirar o máximo proveito do Docker, o Elastic Beanstalk permite criar um ambiente no qual as instâncias do Amazon EC2 executam vários contêineres do Docker, lado a lado.

O diagrama a seguir mostra um ambiente demonstrativo do Elastic Beanstalk configurado com três contêineres do Docker em execução em cada instância do Amazon EC2 em um grupo de Auto Scaling:



Note

O Elastic Beanstalk oferece recursos de extensibilidade para todas as plataformas que você pode usar para personalizar a implantação e execução das aplicações. Para a ramificação da plataforma ECS em execução no Amazon Linux 2, a implementação do fluxo de trabalho de implantação de instâncias desses recursos é diferente de outras plataformas. Para obter mais informações, consulte [Fluxo de trabalho de implantação de instâncias para o ECS em execução no Amazon Linux 2 e versões posteriores](#).

Dockerrun.aws.json file

Instâncias de contêiner: instâncias do Amazon EC2 executando o Docker gerenciado pelo ECS em um ambiente do Elastic Beanstalk exigem um arquivo de configuração denominado `Dockerrun.aws.json`. Esse arquivo é específico do Elastic Beanstalk e pode ser usado individualmente ou combinado com código-fonte e conteúdo em um [pacote de fonte](#) para criar um ambiente na plataforma Docker.

Note

A versão 1 do formato `Dockerrun.aws.json` é usada para iniciar um único contêiner do Docker em um ambiente do Elastic Beanstalk em execução no Amazon Linux AMI (a versão anterior ao Amazon Linux 2). O ambiente é baseado na ramificação de plataforma Docker em execução no Amazon Linux de 64 bits, que será retirada em 18 de junho de 2022. Para saber mais sobre o formato `Dockerrun.aws.json` v1, consulte [Configuração da plataforma Docker - sem Docker Compose](#).

O formato do `Dockerrun.aws.json` versão 2 adiciona suporte a vários contêineres por instância do Amazon EC2 e só pode ser usado com a plataforma Docker gerenciada pelo ECS. O formato difere significativamente da versão anterior.

Consulte [Dockerrun.aws.json v2](#) para detalhes sobre o formato atualizado e um exemplo de arquivo.

Imagens de Docker

A plataforma Docker gerenciada pelo ECS para o Elastic Beanstalk exige que as imagens criadas previamente e armazenadas em um repositório de imagens online público ou privado.

Note

A criação de imagens personalizadas durante a implantação com um `Dockerfile` não é compatível com a plataforma Docker gerenciada pelo ECS no Elastic Beanstalk. Crie suas imagens e implante-as em um repositório online antes de criar um ambiente do Elastic Beanstalk.

Especifique as imagens por nome em `Dockerrun.aws.json v2`. Observe as seguintes convenções:

- As imagens em repositórios oficiais no Docker Hub usam um único nome (por exemplo, `ubuntu` ou `mongo`).
- As imagens em outros repositórios no Docker Hub são qualificadas com um nome de organização (por exemplo, `amazon/amazon-ecs-agent`).
- As imagens em outros registros online são ainda mais qualificadas por um nome de domínio (por exemplo, `quay.io/assemblyline/ubuntu`).

Para configurar o Elastic Beanstalk para autenticar em um repositório privado, inclua o parâmetro `authentication` no arquivo `Dockerrun.aws.json v2`.

Função de instância de contêiner

O Elastic Beanstalk usa uma AMI otimizada para Amazon ECS com um agente de contêiner do Amazon ECS que é executado em um contêiner do Docker. O agente se comunica com o Amazon ECS para coordenar as implantações de contêiner. Para se comunicar com o Amazon ECS, cada instância do Amazon EC2 deve ter as permissões correspondentes no IAM. Estas permissões são anexadas ao [perfil da instância](#) padrão quando você cria um ambiente no Console de Gerenciamento do Elastic Beanstalk:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ECSAccess",
      "Effect": "Allow",
      "Action": [
        "ecs:Poll",
        "ecs:StartTask",
```

```
        "ecs:StopTask",
        "ecs:DiscoverPollEndpoint",
        "ecs:StartTelemetrySession",
        "ecs:RegisterContainerInstance",
        "ecs:DeregisterContainerInstance",
        "ecs:DescribeContainerInstances",
        "ecs:Submit*"
    ],
    "Resource": "*"
}
]
```

Se você criar seu próprio perfil da instância, poderá anexar a política gerenciada `AWSElasticBeanstalkMulticontainerDocker` para garantir que as permissões estejam atualizadas. Para obter instruções sobre como criar políticas e funções no IAM, consulte [Criar funções do IAM](#) no Guia do usuário do IAM.

Recursos do Amazon ECS criados pelo Elastic Beanstalk

Quando você cria um ambiente usando a plataforma Docker gerenciada pelo ECS, o Elastic Beanstalk cria e configura automaticamente vários recursos do Amazon Elastic Container Service enquanto cria o ambiente. Fazendo isso, ele cria os contêineres necessários em cada instância do Amazon EC2.

- **Cluster do Amazon ECS:** as instâncias de contêiner no Amazon ECS são organizadas em clusters. Quando é usado com o Elastic Beanstalk, sempre é criado um cluster para cada ambiente do Docker gerenciado pelo ECS.
- **Definição de tarefas do Amazon ECS:** o Elastic Beanstalk usa o arquivo `DockerRun.aws.json` v2 no projeto para gerar a definição de tarefas do Amazon ECS que é usada para configurar instâncias de contêiner no ambiente.
- **Tarefa do Amazon ECS:** o Elastic Beanstalk se comunica com o Amazon ECS para executar uma tarefa em todas as instâncias do ambiente para coordenar a implantação de contêineres. Em um ambiente escalável, o Elastic Beanstalk inicia uma nova tarefa sempre que uma instância é adicionada ao cluster. Em casos raros, pode ser necessário aumentar o espaço reservado para contêineres e imagens. Saiba mais na seção [Configurar ambientes do Docker](#).
- **Agente de contêiner do Amazon ECS:** o agente é executado em um contêiner do Docker nas instâncias do ambiente. O agente consulta o serviço do Amazon ECS e aguarda uma tarefa para executar.

- Volumes de dados do Amazon ECS: o Elastic Beanstalk insere definições de volumes (além dos volumes que você define no `DockerRun.aws.json v2`) na definição de tarefas para facilitar a coleta de logs.

O Elastic Beanstalk cria volumes de log na instância de contêiner, um para cada contêiner, em `/var/log/containers/containername`. Esses volumes são denominados `awseb-logs-containername` e são fornecidos para os contêineres que devem ser montados. Consulte [Formato de definição de contêiner](#) para detalhes sobre como montá-los.

Usar vários listeners do Elastic Load Balancing

Você pode configurar vários listeners do Elastic Load Balancing em um ambiente do Docker gerenciado pelo ECS para acomodar o tráfego de entrada para proxies ou outros serviços que não são executados na porta HTTP padrão.

Crie uma pasta `.ebextensions` no seu pacote de origem e adicione um arquivo com a extensão `.config`. O exemplo a seguir mostra um arquivo de configuração que cria um listener do Elastic Load Balancing na porta 8080.

.ebextensions/elb-listener.config

```
option_settings:  
  aws:elb:listener:8080:  
    ListenerProtocol: HTTP  
    InstanceProtocol: HTTP  
    InstancePort: 8080
```

Se o seu ambiente estiver sendo executado em uma [Amazon Virtual Private Cloud](#) (Amazon VPC) personalizada criada por você, o Elastic Beanstalk cuidará do restante. Em uma VPC padrão, é preciso configurar o grupo de segurança da instância para permitir a entrada do balanceador de carga. Adicione um segundo arquivo de configuração que inclui uma regra de entrada no security group:

.ebextensions/elb-ingress.config

```
Resources:  
  port8080SecurityGroupIngress:  
    Type: AWS::EC2::SecurityGroupIngress  
    Properties:
```

```
GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
IpProtocol: tcp
ToPort: 8080
FromPort: 8080
SourceSecurityGroupName: { "Fn::GetAtt": ["AWSEBLoadBalancer",
"SourceSecurityGroup.GroupName"] }
```

Para obter mais informações sobre o formato de arquivo de configuração, consulte [Adicionar e personalizar recursos do ambiente do Elastic Beanstalk](#) e [Configurações de opção](#).

Além de adicionar um listener à configuração do Elastic Load Balancing e abrir uma porta no grupo de segurança, é preciso mapear a porta na instância do host para uma porta no contêiner do Docker, na seção `containerDefinitions` do arquivo `Dockerrun.aws.json v2`. O trecho a seguir mostra um exemplo:

```
"portMappings": [
  {
    "hostPort": 8080,
    "containerPort": 8080
  }
]
```

Consulte [Dockerrun.aws.json v2](#) para obter detalhes sobre o formato do arquivo `Dockerrun.aws.json v2`.

Falha nas implantações de contêiner

Se uma tarefa do Amazon ECS falhar, um ou mais contêineres em seu ambiente Elastic Beanstalk não serão iniciados. O Elastic Beanstalk não reverte ambientes de vários contêineres devido a falha em uma tarefa do Amazon ECS. Se um contêiner não for iniciado no ambiente, implante novamente a versão atual ou uma versão funcional anterior do console do Elastic Beanstalk.

Para implantar uma versão existente

1. Abra o console do Elastic Beanstalk na região do seu ambiente.
2. Clique em **Actions**, à direita do nome do aplicativo, e depois clique em **View application versions**.
3. Selecione uma versão do aplicativo e clique em **Deploy**.

Configuração do Docker gerenciado pelo ECS

`Dockerrun.aws.json` é um arquivo de configuração do Elastic Beanstalk que descreve como implantar um conjunto de contêineres Docker hospedados em um cluster ECS em um ambiente Elastic Beanstalk. A plataforma Elastic Beanstalk cria uma definição de tarefa do ECS, que inclui uma definição de contêiner do ECS. Essas definições são descritas no arquivo de configuração do `Dockerrun.aws.json`.

A definição de contêiner no arquivo `Dockerrun.aws.json` descreve os contêineres a serem implantados em cada instância do Amazon EC2 no cluster ECS. Nesse caso, uma instância do Amazon EC2 também é chamada de instância de contêiner host, porque hospeda os contêineres Docker. O arquivo de configuração também descreve os volumes de dados a serem criados na instância de contêiner de host para os contêineres do Docker montarem. Para obter mais informações e um diagrama dos componentes em um ambiente Docker gerenciado pelo ECS no Elastic Beanstalk, consulte o [Plataforma Docker gerenciada pelo ECS](#) anterior neste capítulo.

Um arquivo `Dockerrun.aws.json` pode ser usado sozinho ou zipado com o código-fonte adicional em um único arquivo. O código-fonte arquivado com um `Dockerrun.aws.json` é implantado nas instâncias do contêiner do Amazon EC2 e fica acessível no diretório `/var/app/current/`.

Tópicos

- [Dockerrun.aws.json v2](#)
- [Formato do volume](#)
- [Formato de definição de contêiner](#)
- [Formato de autenticação — Usar imagens de um repositório privado](#)
- [Exemplo Dockerrun.aws.json v2](#)

Dockerrun.aws.json v2

O arquivo `Dockerrun.aws.json` inclui as seguintes seções:

AWSEBDockerrunVersion

Especifica o número da versão como o valor 2 para ambientes do Docker gerenciado pelo ECS.

volumes

Cria volumes a partir de pastas na instância de contêiner do Amazon EC2 ou a partir do seu pacote de origem (implantados em `/var/app/current`). Monte esses volumes

em caminhos dentro de seus contêineres do Docker usando `mountPoints` na seção `containerDefinitions`.

`containerDefinitions`

Uma gama de definições de contêiner.

autenticação (opcional)

A localização no Amazon S3 de um arquivo `.dockercfg` que contém dados de autenticação para um repositório privado.

As seções de definição de contêiner e de volumes do `DockerRun.aws.json` usam a mesma formatação que as seções correspondentes de um arquivo de definição de tarefas do Amazon ECS. Para obter mais informações sobre o formato de definição de tarefa e uma lista completa de parâmetros de definição de tarefa, consulte [Definições de tarefa do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Formato do volume

O parâmetro de volume cria volumes de pastas na instância de contêiner do Amazon EC2 ou de seu pacote de origem (implantado em `/var/app/current`).

Os volumes são especificados no seguinte formato:

```
"volumes": [  
  {  
    "name": "volumename",  
    "host": {  
      "sourcePath": "/path/on/host/instance"  
    }  
  }  
],
```

Monte esses volumes em caminhos dentro dos seus contêineres do Docker usando `mountPoints` na definição de contêiner.

O Elastic Beanstalk configura volumes adicionais para logs, um para cada contêiner. Eles devem ser montados pelos seus contêineres do Docker para gravar logs na instância do host.

Para obter mais detalhes, consulte o campo `mountPoints` na seção Formato de definição de contêiner a seguir.

Formato de definição de contêiner

Os exemplos a seguir mostram um subconjunto de parâmetros comumente usados na seção Definições do contêiner. Há outros parâmetros opcionais disponíveis.

A plataforma Beanstalk cria uma definição de tarefa do ECS, que inclui uma definição de contêiner do ECS. O Beanstalk suporta um subconjunto de parâmetros para a definição do contêiner ECS. Para obter mais informações, consulte [Definições de contêiner](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Um arquivo `DockerRun.aws.json` contém um ou mais objetos de definição de contêiner com os seguintes campos:

name

O nome do contêiner. Consulte [Parâmetros de definição de contêiner padrão](#) para obter informações sobre o tamanho máximo e os caracteres permitidos.

imagem

O nome de uma imagem do Docker em um repositório do Docker online a partir do qual você está criando um contêiner do Docker. Observe as seguintes convenções:

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

ambiente

Uma gama de variáveis de ambiente a serem passadas para o contêiner.

Por exemplo, a seguinte entrada define uma variável de ambiente com o nome **Container** e o valor **PHP**:

```
"environment": [  
  {  
    "name": "Container",  
    "value": "PHP"  
  }  
]
```



```
] ,
```

essential

Verdadeiro caso a tarefa deva ser interrompida se o contêiner falhar. Os contêineres não essenciais podem concluir ou falhar sem afetar o resto dos contêineres na instância.

memory

Quantidade de memória na instância de contêiner para reservar para o contêiner. Especifique um inteiro diferente de zero para um ou ambos os parâmetros `memory` ou `memoryReservation` em definições de contêiner.

memoryReservation

O limite flexível (em MiB) de memória a ser reservado para o contêiner. Especifique um inteiro diferente de zero para um ou ambos os parâmetros `memory` ou `memoryReservation` em definições de contêiner.

mountPoints

Os volumes da instância de contêiner do Amazon EC2 para montar e o local no sistema de arquivos do contêiner do Docker onde devem ser montados. Quando você montar volumes que contêm conteúdo do aplicativo, o contêiner pode ler os dados carregados em seu pacote de origem. Quando você montar volumes de log para gravar dados de log, o Elastic Beanstalk pode reunir dados de log a partir desses volumes.

O Elastic Beanstalk cria volumes de log na instância de contêiner, um para cada contêiner do Docker, em `/var/log/containers/containername`. Esses volumes são chamados `awseb-logs-containername` e devem ser montados no local dentro da estrutura de arquivos do contêiner onde os logs são gravados.

Por exemplo, o ponto de montagem a seguir mapeia a localização de log `nginx` no contêiner para o volume gerado pelo Elastic Beanstalk para o contêiner `nginx-proxy`.

```
{
  "sourceVolume": "awseb-logs-nginx-proxy",
  "containerPath": "/var/log/nginx"
}
```

portMappings

Mapeia portas de rede no contêiner para portas no host.

links

Lista de contêineres aos quais vincular. Os contêineres vinculados podem descobrir uns aos outros e se comunicar com segurança.

volumesFrom

Monte todos os volumes de um contêiner diferente. Por exemplo, para montar volumes de um contêiner chamado web:

```
"volumesFrom": [  
  {  
    "sourceContainer": "web"  
  }  
],
```

Formato de autenticação — Usar imagens de um repositório privado

A seção `authentication` contém dados de autenticação para um repositório privado. Essa entrada é opcional.

Adicione as informações sobre o bucket do Amazon S3 que contém o arquivo de autenticação no parâmetro `authentication` do arquivo `Dockerrun.aws.json`. Certifique-se de que o parâmetro `authentication` contenha uma chave e um bucket válidos do Amazon S3. O bucket do Amazon S3 deve ser hospedado na mesma região do ambiente que o está usando. O Elastic Beanstalk não fará download dos arquivos de buckets do Amazon S3 hospedados em outras regiões.

Usa o seguinte formato:

```
"authentication": {  
  "bucket": "DOC-EXAMPLE-BUCKET",  
  "key": "mydockercfg"  
},
```

Para obter informações sobre a geração e o upload do arquivo de autenticação, consulte [Usar imagens de um repositório privado](#) o tópico de configuração do ambiente deste capítulo.

Exemplo Dockerrun.aws.json v2

O snippet a seguir é um exemplo que ilustra a sintaxe do arquivo `Dockerrun.aws.json` para uma instância com dois contêineres.

```
{
  "AWSEBDockerrunVersion": 2,
  "volumes": [
    {
      "name": "php-app",
      "host": {
        "sourcePath": "/var/app/current/php-app"
      }
    },
    {
      "name": "nginx-proxy-conf",
      "host": {
        "sourcePath": "/var/app/current/proxy/conf.d"
      }
    }
  ],
  "containerDefinitions": [
    {
      "name": "php-app",
      "image": "php:fpm",
      "environment": [
        {
          "name": "Container",
          "value": "PHP"
        }
      ],
      "essential": true,
      "memory": 128,
      "mountPoints": [
        {
          "sourceVolume": "php-app",
          "containerPath": "/var/www/html",
          "readOnly": true
        }
      ]
    },
    {
      "name": "nginx-proxy",
      "image": "nginx",
      "essential": true,
      "memory": 128,
      "portMappings": [
        {
```

```
        "hostPort": 80,
        "containerPort": 80
    }
],
"links": [
    "php-app"
],
"mountPoints": [
    {
        "sourceVolume": "php-app",
        "containerPath": "/var/www/html",
        "readOnly": true
    },
    {
        "sourceVolume": "nginx-proxy-conf",
        "containerPath": "/etc/nginx/conf.d",
        "readOnly": true
    },
    {
        "sourceVolume": "awseb-logs-nginx-proxy",
        "containerPath": "/var/log/nginx"
    }
]
}
]
```

Ambientes do Docker gerenciado pelo ECS com o console do Elastic Beanstalk

Você pode iniciar um cluster de instâncias de vários contêineres em um ambiente de instância única ou em um ambiente escalável do Elastic Beanstalk usando o console do Elastic Beanstalk. Este tutorial detalha a configuração do contêiner e a preparação do código-fonte para um ambiente que usa dois contêineres.

Os contêineres, uma aplicação PHP e um proxy nginx, são executados lado a lado em cada uma das instâncias do Amazon Elastic Compute Cloud (Amazon EC2) em um ambiente do Elastic Beanstalk. Depois de criar o ambiente e verificar se os aplicativos estão funcionando, conecte-se a uma instância de contêiner para ver como tudo isso se encaixa.

Seções

- [Definir contêineres do Docker gerenciado pelo ECS](#)
- [Adicionar conteúdo](#)

- [Implantar no Elastic Beanstalk](#)
- [Conectar-se a uma instância de contêiner](#)
- [Inspeccionar o agente de contêiner do Amazon ECS](#)

Definir contêineres do Docker gerenciado pelo ECS

O primeiro passo na criação de um novo ambiente do Docker é criar um diretório para os dados do aplicativo. Essa pasta pode estar localizada em qualquer lugar da sua máquina local e ter qualquer nome que você escolher. Além de um arquivo de configuração de contêiner, essa pasta terá o conteúdo do qual o upload será feito para o Elastic Beanstalk e implantado no ambiente.

Note

Todos os códigos deste tutorial estão disponíveis no repositório awslabs no GitHub em <https://github.com/awslabs/eb-docker-nginx-proxy>

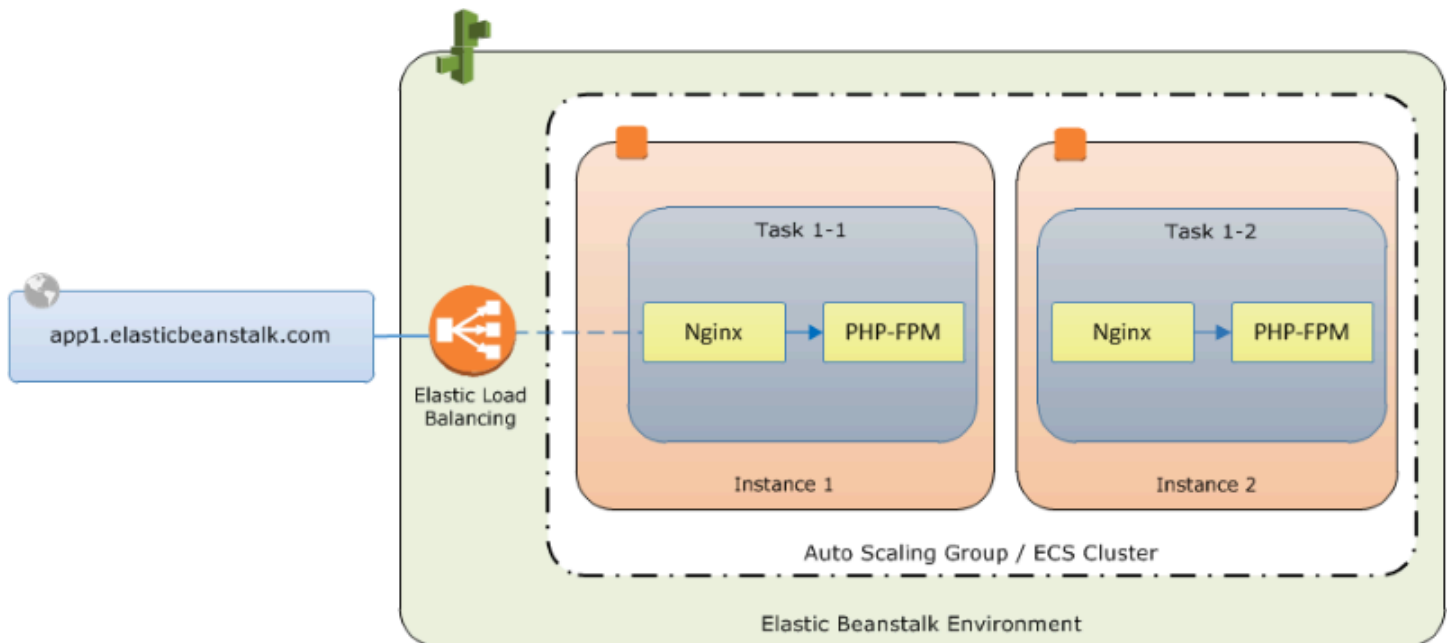
O arquivo que o Elastic Beanstalk usa para configurar os contêineres em uma instância do Amazon EC2 é um arquivo de texto formatado em JSON chamado `Dockerrun.aws.json`. Crie um arquivo de texto com esse nome na raiz do aplicativo e adicione o seguinte texto:

```
{
  "AWSEBDockerrunVersion": 2,
  "volumes": [
    {
      "name": "php-app",
      "host": {
        "sourcePath": "/var/app/current/php-app"
      }
    },
    {
      "name": "nginx-proxy-conf",
      "host": {
        "sourcePath": "/var/app/current/proxy/conf.d"
      }
    }
  ],
  "containerDefinitions": [
    {
      "name": "php-app",
```

```
"image": "php:fpm",
"essential": true,
"memory": 128,
"mountPoints": [
  {
    "sourceVolume": "php-app",
    "containerPath": "/var/www/html",
    "readOnly": true
  }
]
},
{
  "name": "nginx-proxy",
  "image": "nginx",
  "essential": true,
  "memory": 128,
  "portMappings": [
    {
      "hostPort": 80,
      "containerPort": 80
    }
  ],
  "links": [
    "php-app"
  ],
  "mountPoints": [
    {
      "sourceVolume": "php-app",
      "containerPath": "/var/www/html",
      "readOnly": true
    },
    {
      "sourceVolume": "nginx-proxy-conf",
      "containerPath": "/etc/nginx/conf.d",
      "readOnly": true
    },
    {
      "sourceVolume": "awseb-logs-nginx-proxy",
      "containerPath": "/var/log/nginx"
    }
  ]
}
]
```

}

Este exemplo de configuração define dois contêineres, um site em PHP com proxy nginx na frente dele. Esses dois contêineres serão executados lado a lado nos contêineres do Docker em cada instância do ambiente do Elastic Beanstalk, acessando conteúdo compartilhado (conteúdo do site) dos volumes na instância do host, que também são definidos nesse arquivo. Os contêineres são criados a partir de imagens hospedadas em repositórios oficiais no Docker Hub. O ambiente resultante se parece com este:



Os volumes definidos na configuração correspondem ao conteúdo que você vai criar em seguida e fazer upload como parte do pacote de origem do aplicativo. Os contêineres acessam o conteúdo no host montando volumes na seção `mountPoints` das definições de contêiner.

Para obter mais informações sobre o formato de `DockerRun.aws.json` e seus parâmetros, consulte [Formato de definição de contêiner](#).

Adicionar conteúdo

Em seguida, você vai adicionar conteúdo para seu site PHP exibir para os visitantes, e um arquivo de configuração para o proxy nginx.

php-app/index.php

```
<h1>Hello World!!!</h1>
<h3>PHP Version <pre><?= phpversion()?</pre></h3>
```

php-app/static.html

```
<h1>Hello World!</h1>
<h3>This is a static HTML page.</h3>
```

proxy/conf.d/default.conf

```
server {
    listen 80;
    server_name localhost;
    root /var/www/html;

    index index.php;

    location ~ [^/]\.php(/|$) {
        fastcgi_split_path_info ^(.+?\.php)(/.*)$;
        if (!-f $document_root$fastcgi_script_name) {
            return 404;
        }

        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
        fastcgi_param PATH_TRANSLATED $document_root$fastcgi_path_info;


        fastcgi_pass php-app:9000;
        fastcgi_index index.php;
    }
}
```

Implantar no Elastic Beanstalk

Agora, a sua pasta do aplicativo contém os seguintes arquivos:

```
### Dockerrun.aws.json
### php-app
#   ### index.php
#   ### static.html
### proxy
    ### conf.d
        ### default.conf
```


Isso é tudo de que você precisa para criar o ambiente do Elastic Beanstalk. Crie um arquivamento .zip dos arquivos e pastas acima (não incluindo a pasta do projeto de nível superior). Para criar o arquivamento no Windows Explorer, selecione o conteúdo da pasta do projeto, clique com o botão direito do mouse, selecione Enviar para e clique em Pasta compactada (zipada)

 Note

Para obter informações sobre a estrutura de arquivos e instruções necessárias para criar arquivamentos em outros ambientes, consulte [Criar um pacote de origem do aplicativo](#)

Depois, faça upload do pacote de origem para o Elastic Beanstalk e crie o ambiente. Em Platform (Plataforma), selecione Docker. Em Platform branch (Ramificação da plataforma), selecione ECS running on 64bit Amazon Linux 2 (Docker de vários contêineres em execução no Amazon Linux de 64 bits).

Para iniciar um ambiente (console)

1. Abra o console do Elastic Beanstalk com este link pré-configurado:
console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced
2. Em Plataforma, selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo aplicativo ou a plataforma do Docker para aplicativos baseados em contêiner.
3. Em Application code (Código do aplicativo), escolha Upload your code (Fazer upload do código).
4. Escolha Local file (Arquivo local) e Choose file (Escolher arquivo) e abra o pacote de origem.
5. Selecione Review and launch.
6. Revise as configurações disponíveis e, em seguida, escolha Create app.

O console do Elastic Beanstalk redireciona você para o painel de gerenciamento do novo ambiente. Essa tela mostra o status de integridade do ambiente e saída de eventos pelo serviço do Elastic Beanstalk. Quando o status estiver verde, clique no URL ao lado do nome do ambiente para ver seu novo site.

Conectar-se a uma instância de contêiner

Depois, você vai se conectar a uma instância do Amazon EC2 no ambiente do Elastic Beanstalk para ver algumas das partes móveis em ação.

A maneira mais fácil de se conectar a uma instância em seu ambiente é usar a CLI do EB. Para usá-la, [instale a CLI do EB](#), se você ainda não tiver feito isso. Também será necessário configurar o ambiente com um par de chaves SSH do Amazon EC2. Para fazer isso, use a [página de configuração de segurança](#) do console ou o comando `eb init` da CLI do EB. Use o comando `eb ssh` da CLI do EB para se conectar a uma instância do ambiente.

Agora que você se conectou à instância do Amazon EC2 que hospeda os contêineres do docker, poderá ver como as coisas são configuradas. Execute `ls` em `/var/app/current`:

```
[ec2-user@ip-10-0-0-117 ~]$ ls /var/app/current
Dockerrun.aws.json  php-app  proxy
```

Esse diretório contém os arquivos do pacote de origem do qual você fez upload para o Elastic Beanstalk durante a criação do ambiente.

```
[ec2-user@ip-10-0-0-117 ~]$ ls /var/log/containers
nginx-proxy      nginx-proxy-4ba868dbb7f3-stdouterr.log
php-app          php-app-dcc3b3c8522c-stdouterr.log      rotated
```

Este é o local em que os logs são criados na instância de contêiner e coletados pelo Elastic Beanstalk. O Elastic Beanstalk cria um volume nesse diretório para cada contêiner, que é montado no local do contêiner onde os logs são gravados.

Você também pode dar uma olhada no Docker para ver os contêineres em execução com `docker ps`.

```
[ec2-user@ip-10-0-0-117 ~]$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	NAMES	CREATED
4ba868dbb7f3	nginx	"/docker-entrypoint..."	ecs-awseb-Tutorials-env-	4 minutes ago
Up 4 minutes		0.0.0.0:80->80/tcp, :::80->80/tcp	dc2aywfjwg-1-nginx-proxy-acca84ef87c4aca15400	
dcc3b3c8522c	php:fpm	"docker-php-entrypoi..."	ecs-awseb-Tutorials-env-	4 minutes ago
Up 4 minutes		9000/tcp	dc2aywfjwg-1-php-app-b8d38ae288b7b09e8101	

```
d9367c0baad6    amazon/amazon-ecs-agent:latest    "/agent"    5 minutes ago
Up 5 minutes (healthy)    ecs-agent
```

Isso mostra os dois contêineres implantados em execução, bem como o agente de contêiner do Amazon ECS que coordenou a implantação.

Inspeccionar o agente de contêiner do Amazon ECS

As instâncias do Amazon EC2 em um ambiente do Docker gerenciado pelo ECS no Elastic Beanstalk executam um processo de agente em um contêiner do Docker. Esse agente se conecta ao Amazon ECS Service para coordenar implantações de contêiner. Essas implantações são executadas como tarefas no Amazon ECS, que são configuradas nos arquivos de definição de tarefas. O Elastic Beanstalk cria esses arquivos de definição de tarefas com base no `Dockerrun.aws.json` cujo upload será feito em um pacote de origem.

Verifique o status do agente de contêiner com uma solicitação HTTP get para `http://localhost:51678/v1/metadata`:

```
[ec2-user@ip-10-0-0-117 ~]$ curl http://localhost:51678/v1/metadata
{
  "Cluster": "awseb-Tutorials-env-dc2aywfjwg",
  "ContainerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-instance/awseb-Tutorials-env-dc2aywfjwg/db7be5215cd74658aacfcb292a6b944f",
  "Version": "Amazon ECS Agent - v1.57.1 (089b7b64)"
}
```

Essa estrutura mostra o nome do cluster do Amazon ECS e o ARN ([Nome de recurso da Amazon](#)) da instância do cluster (instância do Amazon EC2 à qual você está conectado).

Para obter mais informações, faça uma solicitação HTTP get ao `http://localhost:51678/v1/tasks`:

```
[ec2-user@ip-10-0-0-117 ~]$ curl http://localhost:51678/v1/tasks
{
  "Tasks": [
    {
      "Arn": "arn:aws:ecs:us-west-2:123456789012:task/awseb-Tutorials-env-dc2aywfjwg/bbde7ebe1d4e4537ab1336340150a6d6",
      "DesiredStatus": "RUNNING",
      "KnownStatus": "RUNNING",
      "Family": "awseb-Tutorials-env-dc2aywfjwg",
      "Version": "1",
    }
  ]
}
```

```
"Containers":[
  {
    "DockerId":"dcc3b3c8522cb9510b7359689163814c0f1453b36b237204a3fd7a0b445d2ea6",
    "DockerName":"ecs-awseb-Tutorials-env-dc2aywfjwg-1-php-app-
b8d38ae288b7b09e8101",
    "Name":"php-app",
    "Volumes":[
      {
        "Source":"/var/app/current/php-app",
        "Destination":"/var/www/html"
      }
    ]
  },
  {
    "DockerId":"4ba868dbb7f3fb3328b8afeb2cb6cf03e3cb1cdd5b109e470f767d50b2c3e303",
    "DockerName":"ecs-awseb-Tutorials-env-dc2aywfjwg-1-nginx-proxy-
acca84ef87c4aca15400",
    "Name":"nginx-proxy",
    "Ports":[
      {
        "ContainerPort":80,
        "Protocol":"tcp",
        "HostPort":80
      },
      {
        "ContainerPort":80,
        "Protocol":"tcp",
        "HostPort":80
      }
    ],
    "Volumes":[
      {
        "Source":"/var/app/current/php-app",
        "Destination":"/var/www/html"
      },
      {
        "Source":"/var/log/containers/nginx-proxy",
        "Destination":"/var/log/nginx"
      },
      {
        "Source":"/var/app/current/proxy/conf.d",
        "Destination":"/etc/nginx/conf.d"
      }
    ]
  }
]
```


Seções

- [Migrar com o console do Elastic Beanstalk](#)
- [Migrar com o AWS CLI](#)

Migrar com o console do Elastic Beanstalk

Para migrar usando o console do Elastic Beanstalk, implante o mesmo código-fonte em um novo ambiente baseado na ramificação da plataforma ECS em execução no AL2023. Não é necessária nenhuma alteração no código-fonte.

Para migrar para a ramificação da plataforma ECS em execução no Amazon Linux 2023

1. Usando o fonte da aplicação que já está implantada no ambiente antigo, crie um pacote do fonte da aplicação. Você pode usar o mesmo pacote do fonte da aplicação e o mesmo arquivo `Dockerrun.aws.json v2`.
2. Crie um novo ambiente usando a ramificação da plataforma ECS em execução no Amazon Linux 2023. Use o pacote do fonte da etapa anterior para código da aplicação. Para ter etapas mais detalhadas, consulte [Implantar no Elastic Beanstalk](#) no Tutorial do Docker gerenciado pelo ECS incluído anteriormente neste capítulo.

Migrar com o AWS CLI

Também há a opção de usar o AWS Command Line Interface (AWS CLI) para migrar o ambiente Amazon Linux do Docker de vários contêineres para a ramificação da plataforma, AL2023 do ECS mais nova. Nesse caso, não é necessário criar um novo ambiente nem reimplantar o código-fonte. Você só precisa executar o comando AWS CLI [update-environment](#). Ele executará uma atualização da plataforma para migrar o ambiente existente para a ramificação da plataforma Amazon Linux 2023 do ECS.

Use a sintaxe a seguir para migrar o ambiente para a nova ramificação da plataforma.

```
aws elasticbeanstalk update-environment \  
--environment-name my-env \  
--solution-stack-name "64bit Amazon Linux 2023 version running ECS" \  
--region my-region
```

Veja a seguir um exemplo do comando para migrar o ambiente beta-101 para a versão 3.0.0 da ramificação da plataforma Amazon Linux 2023 do ECS na região us-east-1.

```
aws elasticbeanstalk update-environment \  
--environment-name beta-101 \  
--solution-stack-name "64bit Amazon Linux 2023 v4.0.0 running ECS" \  
--region us-east-1
```

O parâmetro `solution-stack-name` fornece a ramificação da plataforma e sua versão. Use a versão mais recente da ramificação da plataforma especificando o nome da pilha de soluções adequado. A versão de cada ramificação da plataforma está incluída no nome da pilha de soluções, conforme mostrado no exemplo acima. Para obter uma lista das pilhas de soluções mais atuais para a plataforma Docker, consulte [Plataformas compatíveis](#) no guia Plataformas do AWS Elastic Beanstalk.

Note

O comando [List-available-solution-stacks](#) fornece uma lista das versões da plataforma disponíveis para sua conta em uma Região da AWS.

```
aws elasticbeanstalk list-available-solution-stacks --region us-east-1 --query  
SolutionStacks
```

Para saber mais sobre AWS CLI, consulte o [Guia do usuário da AWS Command Line Interface](#). Para obter mais informações sobre os comandos AWS CLI para o Elastic Beanstalk, consulte a [Referência de comandos da AWS CLI para o Elastic Beanstalk](#).

(Herdado) Migrar da ramificação da plataforma Docker de vários contêineres em execução no Amazon Linux para a ramificação da plataforma Docker em execução no Amazon Linux 2

Antes do lançamento da ramificação da plataforma ECS em execução no Amazon Linux 2 de 64 bits, o Elastic Beanstalk oferecia um caminho alternativo de migração para o Amazon Linux 2 para clientes com ambientes baseados no Docker de vários contêineres em execução no Amazon Linux de 64 bits. Este tópico descreve esse caminho de migração e foi mantido neste documento como referência para todos os clientes que concluíram esse caminho de migração.

Agora recomendamos que os clientes com ambientes baseados na ramificação da plataforma Docker de vários contêineres em execução no Amazon Linux de 64 bits migrem para a ramificação da plataforma ECS em execução no Amazon Linux 2 de 64 bits. Ao contrário do caminho alternativo

de migração, essa abordagem continua a usar o Amazon ECS para coordenar implantações de contêineres em ambientes do Docker gerenciado pelo ECS. Esse aspecto permite uma abordagem mais direta. Não é necessária nenhuma alteração no código-fonte e o mesmo arquivo `Dockerrun.aws.json v2` é compatível. Para obter mais informações, consulte [Migrar o Docker de vários contêineres em execução no Amazon Linux para o ECS no Amazon Linux 2023](#).

(Herdado) Migração do Docker de vários contêineres no Amazon Linux para a ramificação da plataforma Docker no Amazon Linux 2

É possível migrar as aplicações em execução na [plataforma Docker de vários contêineres na AMI do Amazon Linux](#) para a plataforma do Amazon Linux 2 Docker. A plataforma Docker de vários contêineres na AMI do Amazon Linux requer que você especifique imagens de aplicação pré-criadas para serem executadas como contêineres. Após a migração, você não terá mais essa limitação, pois a plataforma Docker do Amazon Linux 2 também permite ao Elastic Beanstalk criar imagens de contêiner durante a implantação. As aplicações continuarão sendo executadas em ambientes de vários contêineres com os benefícios adicionais da ferramenta Docker Compose.

O Docker Compose é uma ferramenta para definição e execução de aplicações do Docker de vários contêineres. Para saber mais sobre o Docker Compose e como instalá-lo, consulte [Visão geral do Docker Compose](#) e [Instalar o Docker Compose](#) no site do Docker.

O arquivo **`docker-compose.yml`**

A ferramenta Docker Compose usa o arquivo `docker-compose.yml` para configuração dos serviços de aplicações. Esse arquivo substitui o arquivo `Dockerrun.aws.json v2` no diretório de projetos de aplicações e no pacote de origem da aplicação. O arquivo `docker-compose.yml` é criado manualmente. Consulte o arquivo `Dockerrun.aws.json v2` para a maioria dos valores de parâmetro.

Veja abaixo um arquivo demonstrativo `docker-compose.yml` e o arquivo `Dockerrun.aws.json v2` correspondente para a mesma aplicação. Para obter mais informações sobre o arquivo `docker-compose.yml`, consulte [Referência de arquivo do Compose](#). Para obter mais informações sobre o arquivo `Dockerrun.aws.json v2`, consulte [Dockerrun.aws.json v2](#).

<code>docker-compose.yml</code>	<code>Dockerrun.aws.json v2</code>
<pre>version: '2.4'</pre>	<pre>{</pre>

docker-compose.yml

```

services:
  php-app:
    image: "php:fpm"
    volumes:
      - "./php-app:/var/www/html:ro"
    "
      - "${EB_LOG_BASE_DIR}/php-app:/var/log/sample-app"
    mem_limit: 128m
    environment:
      Container: PHP
  nginx-proxy:
    image: "nginx"
    ports:
      - "80:80"
    volumes:
      - "./php-app:/var/www/html:ro"
    "
      - "./proxy/conf.d:/etc/nginx/conf.d:ro"
      - "${EB_LOG_BASE_DIR}/nginx-proxy:/var/log/nginx"
    mem_limit: 128m
    links:
      - php-app

```

Dockerrun.aws.json v2

```

"AWSEBDockerrunVersion": 2,
"volumes": [
  {
    "name": "php-app",
    "host": {
      "sourcePath": "/var/app/current/php-app"
    }
  },
  {
    "name": "nginx-proxy-conf",
    "host": {
      "sourcePath": "/var/app/current/proxy/conf.d"
    }
  }
],
"containerDefinitions": [
  {
    "name": "php-app",
    "image": "php:fpm",
    "environment": [
      {
        "name": "Container",
        "value": "PHP"
      }
    ],
    "essential": true,
    "memory": 128,
    "mountPoints": [
      {
        "sourceVolume": "php-app",
        "containerPath": "/var/www/html",
        "readOnly": true
      }
    ]
  },
  {
    "name": "nginx-proxy",
    "image": "nginx",

```

docker-compose.yml**Dockerrun.aws.json v2**

```
"essential": true,
"memory": 128,
"portMappings": [
  {
    "hostPort": 80,
    "containerPort": 80
  }
],
"links": [
  "php-app"
],
"mountPoints": [
  {
    "sourceVolume": "php-app"
  },
  {
    "sourceVolume": "nginx-proxy-conf",
    "containerPath": "/var/www/html",
    "readOnly": true
  },
  {
    "sourceVolume": "nginx-proxy-conf",
    "containerPath": "/etc/nginx/conf.d",
    "readOnly": true
  },
  {
    "sourceVolume": "awseb-logs-nginx-proxy",
    "containerPath": "/var/log/nginx"
  }
]
}
```

Considerações adicionais sobre migração

A plataforma Docker Amazon Linux 2 e a plataforma Multi-container Docker Amazon Linux AMI implementam as propriedades do ambiente de forma diferente. Essas duas plataformas também têm diretórios de log diferentes que o Elastic Beanstalk cria para cada um de seus contêineres. Depois de migrar da plataforma do Amazon Linux AMI Multi-Container Docker, você precisará estar ciente dessas diferentes implementações para seu novo ambiente de plataforma do Amazon Linux 2 Docker.

Área	Plataforma Docker no Amazon Linux 2 com Docker Compose	Plataforma Docker de vários contêineres na AMI do Amazon Linux
Propriedades de ambiente	Para que seus contêineres acessem as propriedades do ambiente, é necessário adicionar uma referência ao arquivo <code>.env</code> no arquivo <code>docker-compose.yml</code> . O Elastic Beanstalk gera o arquivo <code>.env</code> , listando cada uma das propriedades como variáveis de ambiente. Para obter mais informações, consulte Fazer referência a variáveis de ambiente em contêineres .	O Elastic Beanstalk pode transmitir as propriedades do ambiente diretamente para o contêiner. Seu código em execução no contêiner pode acessar essas propriedades como variáveis de ambiente sem nenhuma configuração adicional.
Diretórios de log	Para cada contêiner, o Elastic Beanstalk cria um diretório de logs chamado <code>/var/log/eb-docker/containers/ <service name></code> (ou <code>\${EB_LOG_BASE_DIR}/<service name></code>). Para obter mais informações, consulte Registro em log personalizado do contêiner do Docker (Docker Compose) .	Para cada contêiner, o Elastic Beanstalk cria um diretório de logs chamado <code>/var/log/containers/ <containername></code> . Para obter mais informações, consulte o campo <code>mountPoints</code> no Formato de definição de contêiner .

Etapas da migração

Como migrar para a plataforma Docker do Amazon Linux 2

1. Crie o arquivo `docker-compose.yml` para a aplicação, com base no arquivo `Dockerrun.aws.json v2` existente. Para obter mais informações, consulte a seção acima [arquivo `docker-compose.yml`](#).
2. No diretório raiz da pasta do projeto da aplicação, substitua o arquivo `Dockerrun.aws.json v2` pelo `docker-compose.yml` que você acabou de criar.

A estrutura do diretório deve ser a seguinte.

```
~/myApplication
|-- docker-compose.yml
|-- .ebextensions
|-- php-app
|-- proxy
```

3. Use o comando `eb init` para configurar o diretório local para implantação no Elastic Beanstalk.

```
~/myApplication$ eb init -p docker application-name
```

4. Use o comando `eb create` para criar um ambiente e implante sua imagem do Docker.

```
~/myApplication$ eb create environment-name
```

5. Se for uma aplicação Web, depois que o ambiente for iniciado, use o comando `eb open` para exibi-la em um navegador da Web.

```
~/myApplication$ eb open environment-name
```

6. Você pode exibir o status do ambiente recém-criado usando o comando `eb status`.

```
~/myApplication$ eb status environment-name
```

Contêineres pré-configurados do Docker (AMI do Amazon Linux)

Note

Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

A ramificação de plataforma Docker GlassFish pré-configurada que é executada no Amazon Linux AMI (AL1) não é mais compatível. Para migrar a aplicação GlassFish para uma plataforma Amazon Linux 2023 com suporte, implante o GlassFish e o código da sua aplicação em uma imagem do Docker do Amazon Linux 2023. Para obter mais informações, consulte o tópico a seguir, [the section called “Tutorial do GlassFish no Docker: caminho para o Amazon Linux 2023”](#).

Primeiros passos com contêineres do Docker pré-configurados no Amazon Linux AMI (anterior ao Amazon Linux 2)

Esta seção mostra como desenvolver uma aplicação demonstrativa localmente e implantá-la no Elastic Beanstalk com um contêiner pré-configurado do Docker.

Configurar seu ambiente de desenvolvimento local

Para esta demonstração, usamos um exemplo de aplicativo GlassFish.

Para configurar seu ambiente

1. Crie uma pasta para o aplicativo de exemplo.

```
~$ mkdir eb-preconf-example
~$ cd eb-preconf-example
```

2. Faça o download do código de aplicativo de exemplo para a nova pasta.

```
~$ wget https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/samples/docker-glassfish-v1.zip
~$ unzip docker-glassfish-v1.zip
~$ rm docker-glassfish-v1.zip
```

Desenvolver e testar localmente

Como desenvolver um exemplo de aplicativo GlassFish

1. Adicione um `Dockerfile` à pasta raiz do seu aplicativo. No arquivo, especifique a imagem de base do Docker do AWS Elastic Beanstalk que será usada para executar seu contêiner Docker local pré-configurado. Mais tarde, você implantará a aplicação em uma versão da plataforma GlassFish do Docker pré-configurada do Elastic Beanstalk. Escolha a imagem de base do Docker que essa versão da plataforma usa. Para descobrir a imagem do Docker atual da versão da plataforma, consulte a seção [Docker pré-configurado](#) na página Plataformas compatíveis com o AWS Elastic Beanstalk no guia de Plataformas do AWS Elastic Beanstalk.

Example `~/Eb-preconf-example/Dockerfile`

```
# For Glassfish 5.0 Java 8
FROM amazon/aws-eb-glassfish:5.0-al-onbuild-2.11.1
```

Para obter mais informações sobre como usar um `Dockerfile`, consulte [Configuração do Docker](#).

2. Crie a imagem do Docker.

```
~/eb-preconf-example$ docker build -t my-app-image .
```

3. Execute o contêiner Docker da imagem.

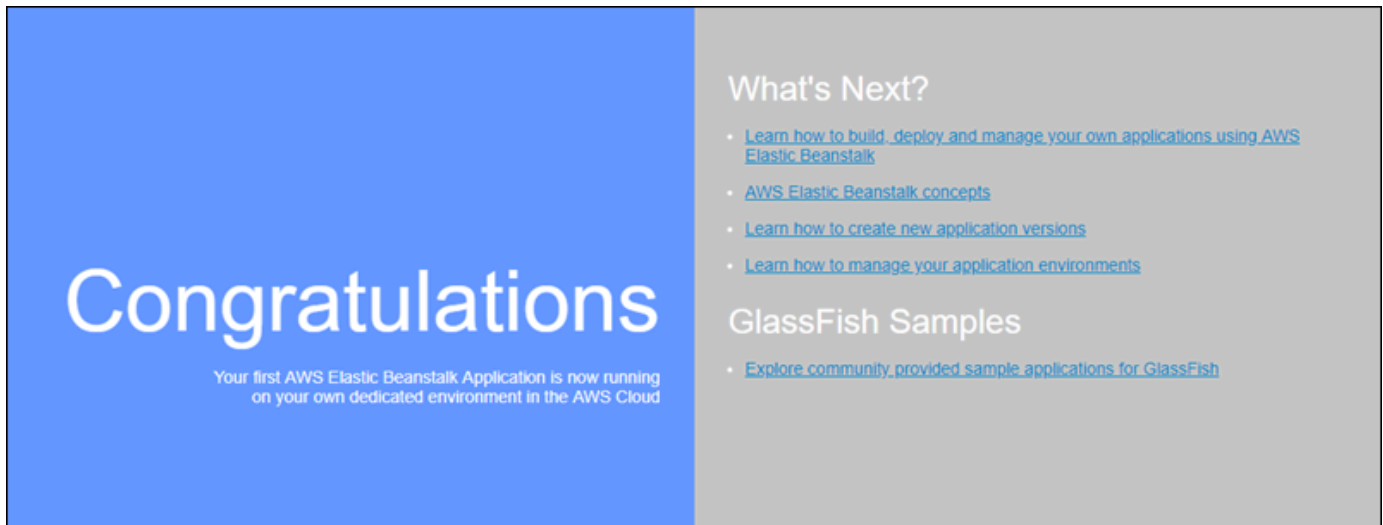
Note

Você deve incluir o sinalizador `-p` para mapear a porta 8080 no contêiner para a porta 3000 do localhost. Os contêineres do Docker do Elastic Beanstalk sempre expõem a aplicação na porta 8080 no contêiner. O sinalizador `-it` executa a imagem como processo interativo. O sinalizador `--rm` limpa o sistema de arquivos do contêiner quando o contêiner é encerrado. Você tem a opção de incluir o sinalizador `-d` para executar a imagem como um daemon.

```
$ docker run -it --rm -p 3000:8080 my-app-image
```

4. Para visualizar o aplicativo de exemplo, digite o seguinte URL no seu navegador da web.

```
http://localhost:3000
```



Implantar no Elastic Beanstalk

Depois de testar a aplicação, você estará pronto para implantá-la no Elastic Beanstalk.

Como implantar a aplicação no Elastic Beanstalk

1. Na pasta raiz do seu aplicativo, renomeie o `Dockerfile` como `Dockerfile.local`. Essa etapa é necessária para que o Elastic Beanstalk use o `Dockerfile` que contém as instruções corretas para que o Elastic Beanstalk crie uma imagem do Docker personalizada em cada instância do Amazon EC2 no ambiente do Elastic Beanstalk.

Note

Não será necessário executar essa etapa se o `Dockerfile` incluir instruções que modifiquem a imagem de base do Docker da versão da plataforma. Não será necessário usar um `Dockerfile` se o seu `Dockerfile` incluir apenas uma linha `FROM` para especificar a imagem de base da qual criar o contêiner. Neste caso, o `Dockerfile` é redundante.

2. Crie um pacote de origem do aplicativo.

```
~/eb-preconf-example$ zip myapp.zip -r *
```

3. Abra o console do Elastic Beanstalk com este link pré-configurado:
console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced
4. Em Platform (Plataforma), em Preconfigured – Docker (Pré-configurado - Docker), escolha Glassfish.
5. Para Application code, escolha Upload your code e Upload.
6. Escolha Local file (Arquivo local), selecione Browse (Procurar) e abra o pacote de origem do aplicativo que você acabou de criar.
7. Escolha Carregar.
8. Selecione Review and launch.
9. Revise as configurações disponíveis e, em seguida, escolha Create app.
10. Quando o ambiente é criado, você pode visualizar o aplicativo implantado. Escolha o URL do ambiente exibido na parte superior do painel do console.

Implantar uma aplicação GlassFish na plataforma Docker: um caminho de migração para o Amazon Linux 2023

O objetivo deste tutorial é fornecer aos clientes que usam a plataforma pré-configurada do GlassFish no Docker (com base no Amazon Linux AML) com um caminho de migração para o Amazon Linux 2023. É possível migrar a aplicação GlassFish para o Amazon Linux 2023 implantando o GlassFish e o código da aplicação para uma imagem do Docker do Amazon Linux 2023.

O tutorial explica o uso da plataforma do Docker do AWS Elastic Beanstalk para implantar uma aplicação baseada no [servidor de aplicações Java EE GlassFish](#) em um ambiente do Elastic Beanstalk.

Demonstramos duas abordagens para criar uma imagem do Docker:

- **Simples:** forneça o código-fonte da aplicação GlassFish e deixe o Elastic Beanstalk criar e executar uma imagem do Docker como parte do provisionamento do ambiente. Isso é fácil de configurar e tem um custo de maior tempo de provisionamento de instâncias.
- **Avançado:** crie uma imagem do Docker personalizada que tenha o código da aplicação e as dependências e forneça-a para o Elastic Beanstalk para uso no ambiente. Essa abordagem é um pouco mais envolvida e diminui o tempo de provisionamento de instâncias em seu ambiente.

Pré-requisitos

Este tutorial pressupõe que você tenha algum conhecimento das operações básicas do Elastic Beanstalk, da Elastic Beanstalk Command Line Interface (EB CLI) e do Docker. Caso ainda não tenha, siga as instruções em [Conceitos básicos do Elastic Beanstalk](#) para iniciar seu primeiro ambiente do Elastic Beanstalk. Este tutorial usa a [EB CLI](#), mas também é possível criar ambientes e fazer upload de aplicações usando o console do Elastic Beanstalk.

Para concluir este tutorial, você precisará fazer o seguinte:

- Uma instalação local de trabalho do Docker. Para obter mais informações, consulte [Obter o Docker](#) no site de documentação do Docker.
- Acesso ao Docker Hub. Você precisará criar um ID do Docker para acessar o Docker Hub. Para obter mais informações, consulte [Compartilhar a aplicação](#) no site de documentação do Docker.

Para saber mais sobre como configurar ambientes do Docker em plataformas do Elastic Beanstalk, consulte [Configuração do Docker](#) neste mesmo capítulo.

Exemplo simples: fornecer o código do aplicativo

Esta é uma maneira fácil de implantar o aplicativo GlassFish. Forneça o código-fonte da aplicação juntamente com o `Dockerfile` incluído neste tutorial. O Elastic Beanstalk cria uma imagem do Docker que inclui a aplicação e a pilha de software GlassFish. Depois, o Elastic Beanstalk executa a imagem nas instâncias do ambiente.

Um problema com essa abordagem é que o Elastic Beanstalk cria a imagem do Docker localmente sempre que ela cria uma instância para o ambiente. A criação da imagem aumenta o tempo de provisionamento de instâncias. Esse impacto não se limita à criação inicial do ambiente. Ele também ocorre durante ações de expansão.

Como iniciar um ambiente com um exemplo de aplicativo GlassFish

1. Faça download do exemplo `docker-glassfish-al2-v1.zip` e expanda o arquivo `.zip` para um diretório em seu ambiente de desenvolvimento.

```
~$ curl https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/samples/docker-glassfish-al2-v1.zip --output docker-glassfish-al2-v1.zip
~$ mkdir glassfish-example
~$ cd glassfish-example
```

```
~/glassfish-example$ unzip ../docker-glassfish-a12-v1.zip
```

A estrutura do diretório deve ser a seguinte.

```
~/glassfish-example
|-- Dockerfile
|-- Dockerrun.aws.json
|-- glassfish-start.sh
|-- index.jsp
|-- META-INF
|   |-- LICENSE.txt
|   |-- MANIFEST.MF
|   `-- NOTICE.txt
|-- robots.txt
`-- WEB-INF
    `-- web.xml
```

Os arquivos a seguir são fundamentais para criar e executar um contêiner do Docker em seu ambiente:

- `Dockerfile`: fornece instruções que o Docker usa para criar uma imagem com a aplicação e as dependências necessárias.
 - `glassfish-start.sh`: um script shell que a imagem do Docker executa para iniciar a aplicação.
 - `Dockerrun.aws.json`: fornece uma chave de registro em log, para incluir o log do servidor de aplicações GlassFish nas [solicitações do arquivo de log](#). Se você não estiver interessado nos logs do GlassFish, é possível omitir esse arquivo.
2. Configure o diretório local para implantação no Elastic Beanstalk.

```
~/glassfish-example$ eb init -p docker glassfish-example
```

3. (Opcional) Use o comando `eb local run` para criar e executar seu contêiner localmente.

```
~/glassfish-example$ eb local run --port 8080
```

Note

Para saber mais sobre o comando `eb local`, consulte [the section called “eb local”](#). O comando não é compatível no Windows. Como alternativa, você pode criar e executar

seu contêiner com os comandos `docker build` e `docker run`. Para obter mais informações, consulte a [Documentação do Docker](#).

- (Opcional) Embora o contêiner esteja em execução, use o comando `eb local open` para visualizar o aplicativo em um navegador da web. Opcionalmente, abra <http://localhost:8080/> em um navegador da Web.

```
~/glassfish-example$ eb local open
```

- Use o comando `eb create` para criar um ambiente e implante seu aplicativo.

```
~/glassfish-example$ eb create glassfish-example-env
```

- Depois que o ambiente for iniciado, use o comando `eb open` para visualizá-lo em um navegador da web.

```
~/glassfish-example$ eb open
```

Quando terminar de trabalhar com o exemplo, encerre o ambiente e exclua os recursos relacionados.

```
~/glassfish-example$ eb terminate --all
```

Exemplo avançado: fornecer uma imagem do Docker pré-compilada

Esta é uma maneira mais avançada de implantar o aplicativo GlassFish. Com base no primeiro exemplo, crie uma imagem do Docker que tenha o código do aplicativo e a pilha de software do GlassFish e envie-a para o Docker Hub. Depois de realizar essa etapa única, é possível iniciar ambientes do Elastic Beanstalk com base na imagem personalizada.

Ao iniciar um ambiente e fornecer sua imagem do Docker, as instâncias no seu ambiente fazem download e usam essa imagem diretamente e não precisam criar uma imagem do Docker. Portanto, o tempo de provisionamento de instâncias é reduzido.

Observações

- As etapas a seguir criam uma imagem do Docker disponível publicamente.

- Você usará comandos do Docker da instalação local do Docker, juntamente com as credenciais do Docker Hub. Para obter mais informações, consulte a seção [Pré-requisitos](#) anterior.

Como iniciar um ambiente com uma imagem do Docker pré-compilada do aplicativo GlassFish

1. Faça download e expanda o exemplo `docker-glassfish-a12-v1.zip` como no [exemplo simples](#) anterior. Se concluiu esse exemplo, é possível usar o diretório que você já tem.
2. Crie uma imagem do Docker e envie-a para o Docker Hub. Insira seu ID do Docker para *docker-id* para iniciar sessão no Docker Hub.

```
~/glassfish-example$ docker build -t docker-id/beanstalk-glassfish-example:latest .
~/glassfish-example$ docker push docker-id/beanstalk-glassfish-example:latest
```

Note

Antes de enviar sua imagem, talvez seja necessário executar `docker login`. Suas credenciais do Docker Hub serão solicitadas se você executar o comando sem parâmetros.

3. Crie um diretório adicional.

```
~$ mkdir glassfish-prebuilt
~$ cd glassfish-prebuilt
```

4. Copie o exemplo a seguir em um arquivo chamado `Dockerrun.aws.json`.

Example `~/glassfish-prebuilt/Dockerrun.aws.json`

```
{
  "AWSEBDockerrunVersion": "1",
  "Image": {
    "Name": "docker-username/beanstalk-glassfish-example"
  },
  "Ports": [
    {
      "ContainerPort": 8080,
      "HostPort": 8080
    }
  ]
}
```

```
    }  
  ],  
  "Logging": "/usr/local/glassfish5/glassfish/domains/domain1/logs"  
}
```

5. Configure o diretório local para implantação no Elastic Beanstalk.

```
~/glassfish-prebuilt$ eb init -p docker glassfish-prebuilt$
```

6. (Opcional) Use o comando `eb local run` para executar o contêiner localmente.

```
~/glassfish-prebuilt$ eb local run --port 8080
```

7. (Opcional) Embora o contêiner esteja em execução, use o comando `eb local open` para visualizar o aplicativo em um navegador da web. Opcionalmente, abra <http://localhost:8080/> em um navegador da Web.

```
~/glassfish-prebuilt$ eb local open
```

8. Use o comando `eb create` para criar um ambiente e implante sua imagem do Docker.

```
~/glassfish-prebuilt$ eb create glassfish-prebuilt-env
```

9. Depois que o ambiente for iniciado, use o comando `eb open` para visualizá-lo em um navegador da web.

```
~/glassfish-prebuilt$ eb open
```

Quando terminar de trabalhar com o exemplo, encerre o ambiente e exclua os recursos relacionados.

```
~/glassfish-prebuilt$ eb terminate --all
```

Configurar ambientes do Docker

Existem várias maneiras de configurar o comportamento do ambiente do Docker do Elastic Beanstalk.

Note

Se o ambiente do Elastic Beanstalk usar uma versão da plataforma do Amazon Linux AMI Docker (que precede o Amazon Linux 2), leia as informações adicionais no [the section called “Configuração do Docker na AMI do Amazon Linux \(que precede o Amazon Linux 2\)”](#).

Seções

- [Configurar o software em ambientes do Docker](#)
- [Fazer referência a variáveis de ambiente em contêineres](#)
- [Usando o recurso de interpolação para variáveis de ambiente \(Docker Compose\)](#)
- [Gerar logs para a geração de relatórios de integridade aprimorados \(Docker Compose\)](#)
- [Registro em log personalizado do contêiner do Docker \(Docker Compose\)](#)
- [Imagens de Docker](#)
- [Configurar atualizações gerenciadas para ambientes do Docker](#)
- [Namespaces de configuração do Docker](#)
- [Configuração do Docker na AMI do Amazon Linux \(que precede o Amazon Linux 2\)](#)

Configurar o software em ambientes do Docker

É possível usar o console do Elastic Beanstalk para configurar o software em execução nas instâncias do ambiente.

Como configurar o ambiente do Docker no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

5. Faça as alterações necessárias na configuração.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Para obter informações sobre como definir configurações de software em qualquer ambiente, consulte [the section called “Propriedades de ambientes e configurações de software”](#). As seções a seguir abordam informações específicas do Docker.

Opções de contêiner

A seção Container options (Opções de contêiner) tem opções específicas da plataforma. Para ambientes do Docker, isso permite que você escolha se o ambiente incluirá ou não o servidor de proxy Nginx.

Ambientes com Docker Compose

Se gerenciar seu ambiente do Docker com o Docker Compose, o Elastic Beanstalk assumirá que você executa um servidor de proxy como um contêiner. Portanto, ele usa como padrão None (Nenhum) para a configuração Proxy server (Servidor de proxy) e o Elastic Beanstalk não fornece uma configuração NGINX.

Note

Mesmo que você selecione NGINX como servidor de proxy, essa configuração será ignorada em um ambiente com Docker Compose. A configuração Proxy server (Servidor de proxy) ainda é padrão para None (Nenhum).

Como o proxy do servidor Web NGINX está desabilitado para a plataforma Docker no Amazon Linux 2 com o Docker Compose, é necessário seguir as instruções para gerar logs para relatórios de integridade aprimorada. Para obter mais informações, consulte [Gerar logs para a geração de relatórios de integridade aprimorados \(Docker Compose\)](#).

Propriedades de ambiente e variáveis de ambiente

A seção Environment Properties (Propriedades de ambiente) permite que você especifique definições de configuração do ambiente nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) que estão executando a aplicação. As propriedades de ambiente são passadas para o aplicativo como pares de chave-valor. Em um ambiente do Docker, o Elastic Beanstalk transmite propriedades de ambiente para contêineres como variáveis de ambiente.

O código do aplicativo em execução em um contêiner pode se referir à variável de um ambiente pelo nome e ler seu valor. O código-fonte que lê essas variáveis de ambiente varia de acordo com a linguagem de programação. É possível encontrar instruções para ler valores de variáveis de ambiente nas linguagens de programação compatíveis com as plataformas gerenciadas do Elastic Beanstalk no respectivo tópico de plataforma. Para obter uma lista de links para esses tópicos, consulte [the section called “Propriedades de ambientes e configurações de software”](#).

Ambientes com Docker Compose

Se você gerenciar seu ambiente do Docker com o Docker Compose, deverá fazer alguma configuração adicional para recuperar as variáveis de ambiente nos contêineres. Para que os executáveis em execução no contêiner acessem essas variáveis de ambiente, será necessário fazer referência a eles no `docker-compose.yml`. Para mais informações, consulte [Fazer referência a variáveis de ambiente em contêineres](#).

Fazer referência a variáveis de ambiente em contêineres

Se você estiver usando a ferramenta Docker Compose na plataforma Docker do Amazon Linux 2, o Elastic Beanstalk gerará um arquivo de ambiente do Docker Compose chamado `.env` no diretório raiz do projeto de aplicação. Esse arquivo armazena as variáveis de ambiente configuradas para Elastic Beanstalk.

Note

Se você incluir um arquivo `.env` no pacote de aplicações, o Elastic Beanstalk não gerará um arquivo `.env`.

Para que um contêiner faça referência às variáveis de ambiente definidas no Elastic Beanstalk, é necessário seguir uma ou ambas as abordagens de configuração.

- Adicione o arquivo `.env` gerado pelo Elastic Beanstalk à opção de configuração `env_file` no arquivo `docker-compose.yml`.
- Defina as variáveis de ambiente diretamente no arquivo `docker-compose.yml`.

Os arquivos a seguir fornecem um exemplo. O arquivo demonstrativo `docker-compose.yml` mostra as duas abordagens.

- Se você definir as propriedades de ambiente `DEBUG_LEVEL=1` e `LOG_LEVEL=error`, o Elastic beanstalk gerará o seguinte arquivo `.env`:

```
DEBUG_LEVEL=1
LOG_LEVEL=error
```

- Neste arquivo `docker-compose.yml`, a opção de configuração `env_file` aponta para o arquivo `.env` e também define a variável de ambiente `DEBUG=1` diretamente no arquivo `docker-compose.yml`.

```
services:
  web:
    build: .
    environment:
      - DEBUG=1
    env_file:
      - .env
```

Observações

- Se você definir a mesma variável de ambiente nos dois arquivos, a variável definida no arquivo `docker-compose.yml` terá precedência maior do que a variável definida no arquivo `.env`.
- Tenha cuidado para não deixar espaços entre o sinal de igual (=) e o valor atribuído à sua variável para evitar que espaços sejam adicionados à string.

Para saber mais sobre variáveis de ambiente no Docker Compose, consulte [Variáveis de ambiente no Compose](#)

Usando o recurso de interpolação para variáveis de ambiente (Docker Compose)

A partir de [28 de julho de 2023](#), data de lançamento da plataforma, a ramificação da plataforma Docker Amazon Linux 2 oferece o recurso de interpolação Docker Compose. Com esse recurso, os valores em um arquivo do Compose podem ser definidos por variáveis e interpolados em tempo de execução. Para obter mais informações sobre esse recurso, consulte [Interpolação](#) no site de documentação do Docker.

⚠ Important

Se desejar usar esse atributo, com suas aplicações esteja ciente de que precisará implementar uma abordagem que use hooks de plataforma.

Isso é necessário devido a uma mitigação que implementamos no mecanismo da plataforma. Essa mitigação garante compatibilidade com as versões anteriores para clientes que não conhecem o novo atributo de interpolação e têm aplicações existentes que usam variáveis de ambiente com o caractere \$. O mecanismo de plataforma atualizado escapa da interpolação por padrão substituindo o caractere \$ por caracteres \$\$.

Veja a seguir um exemplo de um script de hook de plataforma que é possível configurar para permitir o uso do recurso de interpolação.

```
#!/bin/bash

: '
example data format in .env file
key1=value1
key2=value2
'

envfile="/var/app/staging/.env"
tempfile=$(mktemp)

while IFS= read -r line; do
    # split each env var string at '='
    split_str=${line//=/ }
    if [ ${#split_str[@]} -eq 2 ]; then
        # replace '$$' with '$'
        replaced_str=${split_str[1]//\$\$/\$}
        # update the value of env var using ${replaced_str}
        line="${split_str[0]}=${replaced_str}"
    fi
    # append the updated env var to the tempfile
    echo "${line}" #"${tempfile}"
done < "${envfile}"
# replace the original .env file with the tempfile
mv "${tempfile}" "${envfile}"
```

Coloque os hooks da plataforma em ambos os diretórios:

- `.platform/confighooks/predeploy/`
- `.platform/hooks/predeploy/`

Para obter mais informações, consulte [Hooks de plataforma](#) no tópico Estender plataformas Linux deste guia.

Gerar logs para a geração de relatórios de integridade aprimorados (Docker Compose)

O [agente de integridade do Elastic Beanstalk](#) fornece métricas de integridade do sistema operacional e da aplicação para ambientes do Elastic Beanstalk. Ele se baseia em formatos de log do servidor da Web que retransmitem informações em um formato específico.

O Elastic Beanstalk pressupõe que você execute um proxy de servidor da Web como um contêiner. Como resultado, o proxy do servidor da Web NGINX é desativado para ambientes do Docker que executam o Docker Compose. É necessário configurar o servidor para gravar logs no local e no formato usados pelo agente de integridade do Elastic Beanstalk. Isso permite que você faça uso completo dos relatórios de integridade aprimorados, mesmo que o proxy do servidor da Web esteja desabilitado.

Para obter instruções sobre como fazer isso, consulte [Configuração do log de servidor web](#)

Registro em log personalizado do contêiner do Docker (Docker Compose)

Para solucionar problemas de forma eficiente e monitorar os serviços em contêineres, é possível [solicitar logs de instância](#) do Elastic Beanstalk por meio do console de gerenciamento de ambiente ou da CLI do EB. Os logs de instância são compostos de logs de pacote e logs de cauda, combinados e empacotados para permitir que você visualize logs e eventos recentes de maneira eficiente e direta.

O Elastic Beanstalk cria diretórios de logs na instância de contêiner, um para cada serviço definido no arquivo `docker-compose.yml`, em `/var/log/eb-docker/containers/<service name>`. Se você estiver usando o recurso Docker Compose na plataforma Docker do Amazon Linux 2, poderá montar esses diretórios no local dentro da estrutura do arquivo de contêiner onde os logs são gravados. Quando você monta diretórios de logs para gravar dados de log, o Elastic Beanstalk pode coletar dados de log desses diretórios.

Se as aplicações estiverem em uma plataforma Docker que não esteja usando o Docker Compose, você poderá seguir o procedimento padrão descrito em [Registro em log personalizado do contêiner do Docker \(Docker Compose\)](#).

Como configurar os arquivos de logs do seu serviço para serem arquivos de cauda e logs de pacote recuperáveis

1. Edite o arquivo `docker-compose.yml`.
2. Na chave `volumes` do seu serviço, adicione uma montagem de ligação deste modo:

```
"${EB_LOG_BASE_DIR}/<service name>:<log directory inside container>
```

No arquivo demonstrativo `docker-compose.yml` abaixo:

- `nginx-proxy` é *<nome do serviço>*
- `/var/log/nginx` é *<diretório de logs dentro do contêiner>*

```
services:
  nginx-proxy:
    image: "nginx"
    volumes:
      - "${EB_LOG_BASE_DIR}/nginx-proxy:/var/log/nginx"
```

- O diretório `var/log/nginx` contém os logs do serviço `nginx-proxy` no contêiner e será mapeado para o diretório `/var/log/eb-docker/containers/nginx-proxy` no host.
- Todos os logs nesse diretório agora são recuperáveis como logs finais e de pacote por meio da funcionalidade de [solicitação de logs de instância](#) do Elastic Beanstalk.

Observações

- `${EB_LOG_BASE_DIR}` é uma variável de ambiente definida pelo Elastic Beanstalk com o valor `/var/log/eb-docker/containers`.
- O Elastic Beanstalk cria automaticamente o diretório `/var/log/eb-docker/containers/<service name>` para cada serviço no arquivo `docker-compose.yml`.

Imagens de Docker

As ramificações da plataforma Docker e Docker gerenciado pelo ECS para o Elastic Beanstalk são compatíveis com o uso de imagens do Docker armazenadas em um repositório de imagens online público ou privado.

Especifique as imagens por nome em `DockerRun.aws.json`. Observe as seguintes convenções:

- As imagens em repositórios oficiais no Docker Hub usam um único nome (por exemplo, `ubuntu` ou `mongo`).
- As imagens em outros repositórios no Docker Hub são qualificadas com um nome de organização (por exemplo, `amazon/amazon-ecs-agent`).
- As imagens em outros repositórios online são ainda mais qualificadas por um nome de domínio (por exemplo, `quay.io/assemblyline/ubuntu` ou `account-id.dkr.ecr.us-east-2.amazonaws.com/ubuntu:trusty`).

Para ambientes que usam apenas a plataforma Docker, também é possível criar sua própria imagem durante a criação do ambiente com um `Dockerfile`. Para mais detalhes, consulte [Criar imagens personalizadas com um Dockerfile](#). A plataforma Docker de vários contêineres não é compatível com essa funcionalidade.

Usar imagens de um repositório do Amazon ECR

É possível armazenar suas imagens de Docker personalizadas na AWS com o [Amazon Elastic Container Registry](#) (Amazon ECR). Quando você armazena suas imagens de Docker no Amazon ECR, o Elastic Beanstalk autentica automaticamente o registro do Amazon ECR com o [perfil da instância](#) de seu ambiente para que não seja necessário [gerenciar um arquivo de autenticação](#) e fazer upload dele no Amazon Simple Storage Service (Amazon S3).

No entanto, é necessário fornecer suas instâncias com permissão para acessar as imagens em seu repositório do Amazon ECR adicionando permissões ao perfil da instância de seu ambiente. É possível anexar a política gerenciada [AmazonEC2ContainerRegistryReadOnly](#) ao perfil da instância para dar acesso somente de leitura a todos os repositórios do Amazon ECR em sua conta. Alternativamente, você pode conceder acesso ao repositório único usando o modelo a seguir para criar uma política personalizada:

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

```

    {
      "Sid": "AllowEbAuth",
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "AllowPull",
      "Effect": "Allow",
      "Resource": [
        "arn:aws:ecr:us-east-2:account-id:repository/repository-name"
      ],
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:GetRepositoryPolicy",
        "ecr:DescribeRepositories",
        "ecr:ListImages",
        "ecr:BatchGetImage"
      ]
    }
  ]
}

```

Substitua o nome de recurso da Amazon (ARN) na política acima pelo ARN de seu repositório.

Em seu arquivo `Dockerrun.aws.json`, faça referência à imagem pelo URL. Para a [plataforma Docker](#), o URL deve ser inserido na definição de Image:

```

"Image": {
  "Name": "account-id.dkr.ecr.us-east-2.amazonaws.com/repository-name:latest",
  "Update": "true"
},

```

Para a [plataforma Docker de vários contêineres](#), use a chave `image` em um objeto de definição do contêiner:

```

"containerDefinitions": [

```

```
{  
  "name": "my-image",  
  "image": "account-id.dkr.ecr.us-east-2.amazonaws.com/repository-name:latest",
```

Usar imagens de um repositório privado

Para usar uma imagem de Docker em um repositório privado hospedado por um registro online, você deve fornecer um arquivo de autenticação que contenha as informações necessárias para autenticar no registro.

Gere um arquivo de autenticação com o comando `docker login`. Para repositórios no Docker Hub, execute `docker login`:

```
$ docker login
```

Para outros registros, inclua o URL do servidor de registro:

```
$ docker login registry-server-url
```

Note

Se o ambiente do Elastic Beanstalk usar uma versão da plataforma Docker da AMI do Amazon Linux (que precede o Amazon Linux 2), leia as informações adicionais em [the section called “Configuração do Docker na AMI do Amazon Linux \(que precede o Amazon Linux 2\)”](#).

Faça upload de uma cópia chamada `.dockercfg` do arquivo de autenticação em um bucket seguro do Amazon S3. O bucket do Amazon S3 deve ser hospedado na mesma região da AWS do ambiente que o está usando. O Elastic Beanstalk não pode fazer download de arquivos de um bucket do Amazon S3 hospedado em outras regiões. Conceda permissões para a operação `s3:GetObject` à função do IAM no perfil da instância. Para obter mais detalhes, consulte [Gerenciar perfis de instância do Elastic Beanstalk](#).

Inclua as informações do bucket do Amazon S3 no parâmetro `Authentication (v1)` ou `authentication (v2)` no arquivo `Dockerrun.aws.json`.

Para obter mais informações sobre o formato de `DockerRun.aws.json` para ambientes do Docker, consulte [Configuração do Docker](#). Para informações sobre ambientes de vários contêineres, consulte [Configuração do Docker gerenciado pelo ECS](#).

Para obter mais informações sobre o arquivo de autenticação, consulte [Store images on Docker Hub](#) e [docker login](#) no site do Docker.

Configurar atualizações gerenciadas para ambientes do Docker

Com as [atualizações gerenciadas de plataforma](#), é possível configurar o ambiente para ser atualizado automaticamente para a versão mais recente de uma plataforma em um agendamento.

No caso de ambientes do Docker, pode ser conveniente decidir se uma atualização automática de plataforma deverá acontecer entre versões do Docker, quando a nova versão da configuração da plataforma inclui uma nova versão do Docker. O Elastic Beanstalk é compatível com atualizações de plataforma gerenciada nas versões do Docker ao atualizar a partir de um ambiente que execute uma versão da plataforma Docker mais recente do que a 2.9.0. Quando uma nova versão de plataforma inclui uma nova versão do Docker, o Elastic Beanstalk incrementa o número da versão de atualização secundária. Portanto, para permitir as atualizações gerenciadas de plataforma entre versões do Docker, habilite essas atualizações para atualizações de versão secundária e de patch. Para evitar as atualizações gerenciadas de plataforma entre versões do Docker, habilite essas atualizações para aplicar somente atualizações de versão de patch.

Por exemplo, o seguinte [arquivo de configuração](#) habilita as atualizações gerenciadas de Plataforma às 9h UTC toda terça-feira para atualizações de versão secundária e de patch, permitindo as atualizações gerenciadas entre versões do Docker:

Example `.ebextensions/managed-platform-update.config`

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: minor
```

Para ambientes que executam versões de plataforma do Docker 2.9.0 ou anterior, o Elastic Beanstalk nunca executará as atualizações gerenciadas de plataforma se a nova versão da configuração da plataforma incluir uma nova versão do Docker.

Namespaces de configuração do Docker

Você pode usar um [arquivo de configuração](#) para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. As opções de configuração podem ser definidas pelo serviço Elastic Beanstalk ou pela plataforma utilizada, e são organizadas em namespaces.

Note

Essas informações só se aplicam ao ambiente do Docker que não esteja executando o Docker Compose. Essa opção tem um comportamento diferente com ambientes do Docker que executam o Docker Compose. Para obter mais informações sobre serviços proxy com o Docker Compose, consulte [Opções de contêiner](#).

A plataforma Docker é compatível com as opções nos namespaces a seguir, além das [opções compatíveis com todos os ambientes do Elastic Beanstalk](#):

- `aws:elasticbeanstalk:environment:proxy`: escolha o servidor de proxy para o ambiente. O Docker oferece suporte à execução com o servidor de proxy Nginx ou nenhum servidor.

O arquivo de configuração de exemplo a seguir configura um ambiente do Docker para não executar nenhum servidor de proxy.

Example `.ebextensions/docker-settings.config`

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: none
```

Configuração do Docker na AMI do Amazon Linux (que precede o Amazon Linux 2)

Se o ambiente Docker do Elastic Beanstalk usar uma versão da plataforma AMI do Amazon Linux (que precede o Amazon Linux 2), leia as informações adicionais nesta seção.

Usar um arquivo de autenticação para um repositório privado

Essas informações são relevantes se você estiver [usando imagens de um repositório privado](#). A partir do Docker versão 1.7, o comando `docker login` alterou o nome do arquivo de autenticação e

o formato do arquivo. As versões da plataforma Docker da AMI do Amazon Linux (que precede o Amazon Linux 2) exigem o arquivo de configuração de formato `~/.dockercfg` mais antigo.

Com o Docker versão 1.7 e posterior, o comando `docker login` cria o arquivo de autenticação em `~/.docker/config.json` no formato a seguir.

```
{
  "auths": {
    "server": {
      "auth": "key"
    }
  }
}
```

Com o Docker versão 1.6.2 e anterior, o comando `docker login` cria o arquivo de autenticação em `~/.dockercfg` no formato a seguir.

```
{
  "server" :
  {
    "auth" : "auth_token",
    "email" : "email"
  }
}
```

Para converter um arquivo `config.json`, remova a chave `auths` externa, adicione uma chave `email` e simplifique o documento JSON para que corresponda ao formato antigo.

Nas versões da plataforma Docker do Amazon Linux 2, o Elastic Beanstalk usa o nome e o formato mais recentes do arquivo de autenticação. Se você estiver usando uma versão da plataforma Docker do Amazon Linux 2, será possível usar o arquivo de autenticação criado pelo comando `docker login` sem nenhuma conversão.

Configurar volumes de armazenamento adicionais

Para melhorar a performance na AMI do Amazon Linux, o Elastic Beanstalk configura dois volumes de armazenamento do Amazon EBS para as instâncias do Amazon EC2 do ambiente Docker. Além do volume raiz provisionado para todos os ambientes do Elastic Beanstalk, um segundo volume de 12 GB chamado `xvdcz` é provisionado para armazenamento de imagens em ambientes do Docker.

Se você precisar de mais espaço de armazenamento ou de mais IOPS para imagens de Docker, pode personalizar o volume de armazenamento de imagens usando a opção de configuração `BlockDeviceMapping` no namespace [aws:autoscaling:launchconfiguration](#).

Por exemplo, o seguinte [arquivo de configuração](#) aumenta o tamanho do volume de armazenamento para 100 GB com 500 IOPS provisionadas:

Example `.ebextensions/blockdevice-xvdcz.config`

```
option_settings:
  aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: /dev/xvdcz=:100::io1:500
```

Se você usar a opção `BlockDeviceMappings` para configurar volumes adicionais para seu aplicativo, inclua um mapeamento para `xvdcz` a fim de garantir que ele seja criado. O exemplo a seguir configura dois volumes, o volume de armazenamento de imagens `xvdcz` com as configurações padrão e um volume de aplicativo adicional de 24 GB chamado `sdh`:

Example `.ebextensions/blockdevice-sdh.config`

```
option_settings:
  aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: /dev/xvdcz=:12:true:gp2,/dev/sdh=:24
```

Note

Quando você altera as configurações nesse namespace, o Elastic Beanstalk substitui todas as instâncias no ambiente por instâncias que executam a nova configuração. Para mais detalhes, consulte [Alterações de configuração](#).

Criar e implantar aplicações Go no Elastic Beanstalk

AWS Elastic Beanstalk for Go facilita a implantação, o gerenciamento e a escalabilidade de seus aplicativos web Go usando a Amazon Web Services. O Elastic Beanstalk para Go está disponível para qualquer pessoa que esteja desenvolvendo ou hospedando uma aplicação Web usando Go. Este capítulo fornece step-by-step instruções para implantar sua aplicação web no Elastic Beanstalk.

Depois de implantar a aplicação do Elastic Beanstalk, é possível continuar usando a EB CLI para gerenciar a aplicação e o ambiente ou usar as APIs, a AWS CLI ou o console do Elastic Beanstalk.

Os tópicos neste capítulo pressupõem que você tem um nível básico de conhecimento sobre ambientes do Elastic Beanstalk. Se você não tiver usado o Elastic Beanstalk antes, tente o [tutorial de conceitos básicos](#) para saber os princípios.

Tópicos

- [Conceitos básicos do Go no Elastic Beanstalk](#)
- [Configurar seu ambiente de desenvolvimento Go](#)
- [Usar a plataforma Go do Elastic Beanstalk](#)
- [Implantar uma aplicação Go no Elastic Beanstalk](#)

Conceitos básicos do Go no Elastic Beanstalk

Para iniciar com as aplicações Go no AWS Elastic Beanstalk, tudo o que você precisa é de um [pacote de fonte](#) de aplicação para fazer upload como a primeira versão da aplicação e implantá-lo em um ambiente. Quando você cria um ambiente, o Elastic Beanstalk aloca todos os recursos da AWS necessários para executar uma aplicação Web altamente escalável.

Iniciar um ambiente com um aplicativo Go de exemplo

O Elastic Beanstalk fornece aplicações demonstrativas de página única para cada plataforma. O Elastic Beanstalk também fornece aplicações demonstrativas mais complexas que mostram o uso de recursos adicionais da AWS, como o Amazon RDS e linguagem ou recursos e APIs específicos da plataforma.

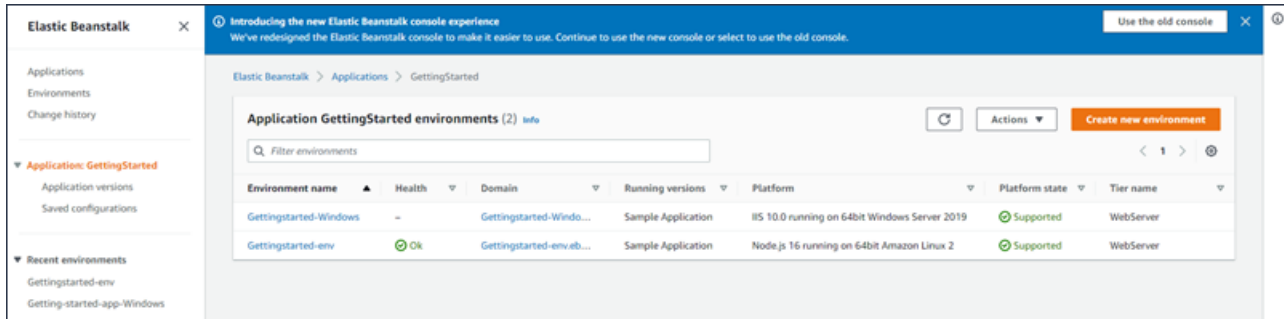
Amostras

Configurações compatíveis	Tipo de ambiente	Pacote de origem	Descrição
Go	Servidor web	go.zip	Aplicativo de página única.

Faça download da aplicação demonstrativa e implante-a no Elastic Beanstalk seguindo estas etapas.

Para iniciar um ambiente com um aplicativo de exemplo (console)

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, escolha Aplicativos e o nome de um aplicativo existente na lista ou [crie um](#).
3. Na página de visão geral da aplicação, escolha Create a new environment (Criar um novo ambiente).



Isso inicia o assistente Create environment (Criar ambiente). O assistente fornece um conjunto de etapas para a criação de um novo ambiente.

Step 1
Configure environment

Step 2
Configure service access

Step 3 - optional
Configure instance traffic and scaling

Step 4 - optional
Set up networking, database, and tags

Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

Configure environment [Info](#)

Environment tier [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- Web server environment**
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
- Worker environment**
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information [Info](#)

Application name
GettingStarted
Maximum length of 100 characters.

▶ **Application tags (optional)**

Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name
GettingStarted-env
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain name
 .us-east-1.elasticbeanstalk.com

Environment description

Platform [Info](#)

Platform type

- Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)
- Custom platform**
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform


Platform branch

Platform version

Application code [Info](#)


- Sample application**
- Existing version**
Application versions that you have uploaded.
- Upload your code**
Upload a source bundle from your computer or copy one from Amazon S3.

4. Para nível de ambiente, escolha o [nível de ambiente](#) Web server environment (Ambiente do servidor Web) ou Worker environment (Ambiente do operador). Você não pode alterar o nível de um ambiente após a criação.

 Note

O [.NET na Plataforma de servidor Windows](#) não tem suporte para a camada de ambiente do operador.

5. Em Plataforma, selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.

 Note

O Elastic Beanstalk é compatível com várias [versões](#) para a maioria das plataformas listadas. Por padrão, o console seleciona a versão recomendada da plataforma e a ramificação da plataforma escolhida. Se a aplicação exigir outra versão, você poderá selecioná-la aqui. Para obter informações sobre versões de plataforma compatíveis, consulte [the section called “Plataformas compatíveis”](#).

6. Para Application code, escolha Sample application.
7. Em Configuration presets (Predefinições de configuração), escolha Single instance (Instância única).
8. Escolha Next (Próximo).
9. A página Configurar acesso ao serviço é exibida.

Configure service access [Info](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role
 Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

[View permission details](#)

Cancel Skip to review Previous **Next**

10. Escolha Usar um perfil de serviço existente em Perfil de serviço.

11. Em seguida, vamos nos concentrar na lista suspensa Perfil de instância do EC2. Os valores exibidos nessa lista suspensa podem variar, dependendo de a conta ter criado anteriormente um novo ambiente.

Escolha uma das opções a seguir com base nos valores exibidos na lista.

- Se `aws-elasticbeanstalk-ec2-role` for exibido na lista suspensa, selecione-o na lista suspensa Perfil de instância do EC2.
- Se outro valor for exibido na lista e for o perfil padrão de instância do EC2 para seus ambientes, selecione-o na lista suspensa Perfil de instância do EC2.
- Se a lista suspensa Perfil de instância do EC2 não mostrar nenhum valor para seleção, expanda o procedimento que se segue, Criar um perfil do IAM para ser o perfil de instância do EC2.

Conclua as etapas em Criar perfil do IAM para ser o perfil de instância do EC2 para criar um perfil do IAM que você possa selecionar posteriormente para ser o perfil de instância do EC2. Depois, retorne para esta etapa.

Agora que você criou um perfil do IAM e atualizou a lista, ele será exibido como uma opção na lista suspensa. Selecione o perfil do IAM que você acabou de criar na lista suspensa Perfil de instância do EC2.

12. Escolha Skip to Review (Pular para revisão) na página Configure service access (Configurar acesso a serviço).

Isso selecionará os valores padrão para essa etapa e pulará as etapas opcionais.

13. A página Review (Revisão) exibe um resumo de todas as suas escolhas.

Para personalizar ainda mais o ambiente, escolha Edit (Editar) ao lado da etapa que inclui os itens que você deseja configurar. Você pode configurar as opções a seguir somente durante a criação do ambiente:

- Nome do ambiente
- Nome de domínio
- Versão da plataforma
- Processor
- VPC
- Nível

Você pode alterar as configurações a seguir após a criação do ambiente, mas elas exigem que novas instâncias ou outros recursos sejam provisionados, e pode levar muito tempo para que elas sejam aplicadas:

- Tipo de instância, volume raiz, par de chaves e função do AWS Identity and Access Management (IAM)
- Banco de dados interno do Amazon RDS
- Load balancer

Para obter detalhes sobre todas as configurações disponíveis, consulte [O assistente de criação de novo ambiente](#).

14. Escolha Submit (enviar) na parte inferior da página para inicializar a criação do novo ambiente.

Criar um perfil do IAM para ser o perfil de instância do EC2

The screenshot shows the 'Configure service access' dialog box in the AWS IAM console. The title is 'Configure service access' with an 'Info' link. Below the title is a 'Service access' section with a brief explanation and a 'Learn more' link. The 'Service role' section has two radio buttons: 'Create and use new service role' (unselected) and 'Use an existing service role' (selected). Under 'Existing service roles', there is a text input field containing 'aws-elasticbeanstalk-service-role' and a refresh button. The 'EC2 key pair' section has a text input field with 'Choose a key pair' and a refresh button. The 'EC2 instance profile' section has a text input field with 'aws-elasticbeanstalk-ec2-role' and a refresh button. Below these fields is a 'View permission details' button. At the bottom of the dialog are four buttons: 'Cancel', 'Skip to review', 'Previous', and 'Next' (highlighted in orange).

Para criar um perfil do IAM para ser o perfil de instância do EC2


1. Escolha Visualizar detalhes de permissão. Essa opção exibe a lista suspensa Perfil de instância do EC2.

Uma janela modal intitulada Visualizar permissões de perfil de instância é exibida. Essa janela lista os perfis gerenciados que deverão ser anexados ao novo perfil de instância do EC2 que você criar. Ela também fornece um link para iniciar o console do IAM.

2. Escolha o link Console do IAM exibido na parte superior da janela.
3. No painel de navegação do console do IAM, escolha Roles (Perfis).
4. Selecione Create role (Criar função).
5. Em Tipo de entidade confiável, selecione Serviço da AWS.
6. Em Use case (Caso de uso), selecione EC2.
7. Escolha Next (Próximo).
8. Anexe as políticas gerenciadas apropriadas. Role a janela modal Visualizar permissões de perfil de instância para ver as políticas gerenciadas. As políticas também estão listadas aqui:

- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

9. Escolha Next (Próximo).
10. Insira um nome para a função.
11. (Opcional) Adicione tags à função.
12. Selecione Create role (Criar função).
13. Retorne à janela do console do Elastic Beanstalk que está aberta.
14. Feche a janela modal Visualizar permissões de perfil de instância.

 Important

Não feche a página do navegador que exibe o console do Elastic Beanstalk.

15. Escolha



(atualizar), ao lado da lista suspensa Perfil de instância do EC2.

Essa opção atualiza a lista suspensa para que o perfil que você acabou de criar seja exibido na lista suspensa.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo diferente a qualquer momento. A implantação de uma nova versão do aplicativo é muito rápida, pois não é necessário provisionar ou reiniciar as instâncias do EC2.

Depois de implantar uma ou duas aplicações demonstrativas, quando você estiver pronto para começar a desenvolver e executar aplicações Go localmente, consulte [Configurar seu ambiente de desenvolvimento Go](#).

Configurar seu ambiente de desenvolvimento Go

Configure um ambiente de desenvolvimento Go para testar sua aplicação localmente antes de implantá-lo no AWS Elastic Beanstalk. Este tópico descreve as etapas de configuração para seu ambiente de desenvolvimento e fornece links para páginas de instalação de ferramentas úteis.

Para ver as etapas de configuração comuns e as ferramentas que se aplicam a todas as linguagens, consulte [Configurar sua máquina de desenvolvimento para uso com o Elastic Beanstalk](#).

Instalação do Go

Para executar aplicativos Go localmente, instale o Go. Se você não precisa de uma versão específica, obtenha a versão mais recente compatível com o Elastic Beanstalk. Para obter uma lista das versões compatíveis, consulte [Go](#) no documento Plataformas do AWS Elastic Beanstalk.

Baixe o Go em <https://golang.org/doc/install>.

Instalar o AWS SDK for Go

Se você precisa gerenciar os recursos da AWS em sua aplicação, instale o AWS SDK for Go usando o comando a seguir.

```
$ go get github.com/aws/aws-sdk-go
```

Para obter mais informações, consulte [AWS SDK for Go](#).

Usar a plataforma Go do Elastic Beanstalk

Use o AWS Elastic Beanstalk para executar, construir e configurar aplicações baseados em Go. Quanto aos aplicativos simples em Go, há duas formas de implantá-los:

- Forneça um pacote de origem com um arquivo de origem na raiz chamado `application.go`, que contém o pacote principal do seu aplicativo. O Elastic Beanstalk cria o binário usando o seguinte comando:

```
go build -o bin/application application.go
```

Depois que a aplicação é criada, o Elastic Beanstalk a inicia na porta 5000.

- Forneça um pacote de origem com um arquivo binário chamado `application`. O arquivo binário pode ficar localizado na raiz do pacote de origem ou no diretório `bin/` do pacote de origem. Se você colocar o arquivo binário `application` nos dois locais, o Elastic Beanstalk usará o arquivo do diretório `bin/`.

O Elastic Beanstalk inicia essa aplicação na porta 5000.

Em ambos os casos, com Go 1.11 ou posterior, também é possível fornecer requisitos de módulo em um arquivo chamado `go.mod`. Para obter mais informações, consulte [Migrar para módulos Go](#) no blog do Go.

Quanto aos aplicativos mais complexos em Go, há duas formas de implantá-los:

- Forneça um pacote de origem que inclua os arquivos de origem do aplicativo, juntamente com um [Buildfile](#) e um [Procfile](#). O Buildfile inclui um comando para criar o aplicativo e o Procfile inclui instruções para executar o aplicativo.
- Forneça um pacote de origem que inclua os arquivos binários do aplicativo, juntamente com um Procfile. O Procfile inclui instruções para executar o aplicativo.

A plataforma Go inclui um servidor de proxy para atender a ativos estáticos e encaminhar o tráfego para sua aplicação. Você pode [estender ou substituir a configuração de proxy padrão](#) para cenários avançados.

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte [the section called “Estender plataformas Linux”](#).

Configurar o ambiente Go

As configurações da plataforma Go permitem que você ajuste o comportamento das instâncias do Amazon EC2. É possível editar a configuração da instância do Amazon EC2 do ambiente do Elastic Beanstalk usando o console do Elastic Beanstalk.

Você pode usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3 e configurar variáveis que sua aplicação pode ler no ambiente.

Como configurar o ambiente Go no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).

4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Opções de log

A seção Log Options tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Enable log file rotation to Amazon S3 (Habilitar a rotação de arquivos de log para o Amazon S3): especifica se os arquivos de log das instâncias do Amazon EC2 da aplicação serão copiados para o bucket do Amazon S3 associado a ela.

Arquivos estáticos

Para melhorar a performance, você pode usar a seção Static files (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte [the section called “Arquivos estáticos”](#).

Propriedades de ambiente

A seção Environment Properties permite que você especifique definições de configuração do ambiente nas instâncias do Amazon EC2 que estão executando seu aplicativo. As propriedades de ambiente são passadas para o aplicativo como pares de chave-valor.

No ambiente Go executado no Elastic Beanstalk, as variáveis de ambiente podem ser acessadas usando a função `os.Getenv`. Por exemplo, você pode ler uma propriedade denominada `API_ENDPOINT` em uma variável com o seguinte código:

```
endpoint := os.Getenv("API_ENDPOINT")
```

Consulte [Propriedades de ambientes e outras configurações de software](#) para obter mais informações.

Namespace de configuração do Go

Você pode usar um [arquivo de configuração](#) para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. As opções de configuração podem ser definidas pelo serviço Elastic Beanstalk ou pela plataforma utilizada, e são organizadas em namespaces.

A plataforma Go não define nenhum namespace específico da plataforma. É possível configurar o proxy para servir arquivos estáticos usando o namespace `aws:elasticbeanstalk:environment:proxy:staticfiles`. Para obter detalhes e um exemplo, consulte [the section called “Arquivos estáticos”](#).

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte [Opções de configuração](#) para obter mais informações.

A plataforma Go da AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o ambiente Go do Elastic Beanstalk usar uma versão da plataforma AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

Observações

- As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). As ramificações de plataforma AL2023/AL2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem definições de configuração diferentes.
- Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Namespaces de configurações em Go: Amazon Linux AMI (AL1)

Você pode usar um [arquivo de configuração](#) para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. As opções de configuração podem

ser definidas pelo serviço Elastic Beanstalk ou pela plataforma utilizada, e são organizadas em namespaces.

Note

As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). As ramificações de plataforma AL2023/AL2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem definições de configuração diferentes.

A plataforma Go da AMI do Amazon Linux é compatível com um namespace de configuração específico da plataforma, além dos [namespaces compatíveis com todas as plataformas](#). O namespace `aws:elasticbeanstalk:container:golang:staticfiles` permite que você defina opções que mapeiam os caminhos em seu aplicativo Web para as pastas em seu pacote de origem do aplicativo que incluem o conteúdo estático.

Por exemplo, este [arquivo de configuração](#) informa ao servidor de proxy que deve servir arquivos da pasta `staticimages` no caminho `/images`:

Example `.ebextensions/go-settings.config`

```
option_settings:
  aws:elasticbeanstalk:container:golang:staticfiles:
    /html: statichtml
    /images: staticimages
```

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte [Opções de configuração](#) para obter mais informações.

Configurar o processo do aplicativo com um Procfile

Para especificar comandos personalizados para iniciar um aplicativo em Go, inclua um arquivo chamado `Procfile` na raiz do pacote de origem.

Para obter detalhes sobre como escrever e usar um `Procfile`, expanda a seção `Buildfile` e `Procfile` em [the section called “Estender plataformas Linux”](#).

Example Procfile

```
web: bin/server
queue_process: bin/queue_processor
foo: bin/fooapp
```

Você deve chamar a aplicação principal web e listá-la como o primeiro comando no seu Procfile. O Elastic Beanstalk expõe a aplicação principal web no URL raiz do ambiente, por exemplo, `http://my-go-env.elasticbeanstalk.com`.

O Elastic Beanstalk também executa quaisquer aplicações cujo nome não tenha o prefixo `web_`, mas essas aplicações estão disponíveis somente dentro da instância.

O Elastic Beanstalk espera que os processos sejam executados a partir do Procfile para serem executados continuamente. O Elastic Beanstalk monitora essas aplicações e reinicia qualquer processo que seja encerrado. Para processos de curta execução, use um comando do [Buildfile](#).

Usar um Procfile na AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o ambiente Go do Elastic Beanstalk usar uma versão da plataforma AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

Observações

- As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). As ramificações de plataforma AL2023/AL2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem definições de configuração diferentes.
- Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Transferência de portas: Amazon Linux AMI (AL1)

Note

As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). As ramificações de plataforma AL2023/AL2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem definições de configuração diferentes.

O Elastic Beanstalk configura o proxy nginx para encaminhar solicitações à aplicação no número de porta especificado na [propriedade de ambiente](#) `PORT` da aplicação. O aplicativo sempre deve escutar nessa porta. Você pode acessar essa variável em seu aplicativo chamando o método `os.Getenv("PORT")`.

O Elastic Beanstalk usa o número de porta especificado na propriedade de ambiente `PORT` para a porta da primeira aplicação no `Procfile` e, depois, incrementa o número da porta em 100 para cada aplicação subsequente no `Procfile`. Se a propriedade de ambiente `PORT` não estiver definida, o Elastic Beanstalk usará 5000 para a porta inicial.

No exemplo anterior, a propriedade de ambiente `PORT` para a aplicação web é 5000, para a aplicação `queue_process` é 5100 e para a aplicação `foo` é 5200.

Você pode especificar a porta inicial definindo a opção `PORT` com o namespace [aws:elasticbeanstalk:application:environment](#), como mostra o exemplo a seguir.

```
option_settings:  
  - namespace: aws:elasticbeanstalk:application:environment  
    option_name: PORT  
    value: <first_port_number>
```

Para obter mais informações sobre como configurar variáveis de ambiente para a aplicação, consulte [Configurações de opção](#).

Criar executável no servidor com um Buildfile

Para especificar um comando de criação e configuração personalizado para o aplicativo em Go, inclua um arquivo chamado `Buildfile` na raiz do pacote de origem. O nome do arquivo diferencia maiúsculas de minúsculas. Use o seguinte formato para o `Buildfile`:

```
<process_name>: <command>
```

O comando no Buildfile deve corresponder à expressão regular: `^[A-Za-z0-9_]+:\s*.\+$`.

O Elastic Beanstalk não monitora a aplicação executada com um Buildfile. Use um Buildfile para comandos que são executados por breves períodos e são encerrados após a conclusão das tarefas. Para processos de aplicativo de longa execução que não devem ser encerrados, use o [Procfile](#).

No seguinte exemplo de Buildfile, `build.sh` é um script de shell localizado na raiz do pacote de origem:

```
make: ./build.sh
```

Todos os caminhos no Buildfile são relativos à raiz do pacote de origem. Se você já sabe onde os arquivos residem na instância, inclua os caminhos absolutos no Buildfile.

Configurar o proxy reverso

O Elastic Beanstalk usa o nginx como proxy reverso para mapear sua aplicação para o balanceador de carga do Elastic Load Balancing na porta 80. O Elastic Beanstalk oferece uma configuração nginx padrão que você pode estender ou substituir completamente por sua própria configuração.

Por padrão, o Elastic Beanstalk configura o proxy nginx para encaminhar solicitações à sua aplicação na porta 5000. Você pode substituir a porta padrão definindo a [propriedade do ambiente](#) `PORT` para a porta de escuta do seu aplicativo principal.

Note

A porta que seu aplicativo escuta não afeta a porta na qual o servidor nginx escuta para receber solicitações do balanceador de carga.

Configurar o servidor proxy na sua versão da plataforma

Todas as plataformas AL2023/AL2 oferecem suporte a um recurso de configuração de proxy uniforme. Para obter mais informações sobre a configuração do servidor proxy nas versões de plataforma que executam o AL2023/AL2, expanda a seção Configuração do proxy reverso em [the section called “Estender plataformas Linux”](#).

Configurar o proxy da AMI do Amazon Linux (anterior ao Amazon Linux 2)

Observações

- As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). As ramificações de plataforma AL2023/AL2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem definições de configuração diferentes.
- Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Se o ambiente Go do Elastic Beanstalk usar uma versão da plataforma AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações nesta seção.

Estender e substituir a configuração padrão do proxy: Amazon Linux AMI (AL1)

O Elastic Beanstalk usa nginx como proxy reverso para mapear a aplicação para o balanceador de carga na porta 80. Para fornecer a sua própria configuração de nginx, substitua a configuração padrão fornecida pelo Elastic Beanstalk, incluindo o arquivo `.ebextensions/nginx/nginx.conf` no pacote de origem. Se esse arquivo estiver presente, o Elastic Beanstalk o usará no lugar do arquivo de configuração padrão de nginx.

Para incluir diretivas além daquelas no bloco `nginx.conf http`, forneça também os arquivos de configuração adicionais no diretório `.ebextensions/nginx/conf.d/` do pacote de origem. Todos os arquivos desse diretório devem ter a extensão `.conf`.

Para aproveitar a funcionalidade fornecida pelo Elastic Beanstalk, como [Monitoramento e relatório de integridade aprimorada](#), mapeamentos de aplicações automáticos e arquivos estáticos, inclua a linha a seguir no bloco `server` do arquivo de configuração de nginx:

```
include conf.d/elasticbeanstalk/*.conf;
```

Implantar uma aplicação Go no Elastic Beanstalk

Este tutorial mostra o processo de criação de um aplicativo Go e sua implantação em um AWS Elastic Beanstalk ambiente.

Seções

- [Pré-requisitos](#)
- [Criar um aplicativo Go](#)
- [Implantar seu aplicativo Go com a EB CLI](#)
- [Limpar](#)

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em [Conceitos básicos do Elastic Beanstalk](#) para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt (\$) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command  
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows 10, você pode [instalar o Subsistema Windows para Linux](#) para obter uma versão do Ubuntu integrada com o Windows e o Bash.

Este tutorial também usa a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB). Para obter detalhes sobre a instalação e configuração da CLI do EB, consulte [Instalar a EB CLI](#) e [Configurar a EB CLI](#).

Criar um aplicativo Go

Crie um diretório do projeto.

```
~$ mkdir eb-go  
~$ cd eb-go
```

Depois, crie uma aplicação a ser implantada usando o Elastic Beanstalk. Vamos criar um serviço Web RESTful "Hello World".

Este exemplo imprime uma saudação personalizada que varia de acordo com o caminho usado para acessar o serviço.

Crie um arquivo de texto nesse diretório chamado `application.go` com o seguinte conteúdo:

Example `~/eb-go/application.go`

```
package main

import (
    "fmt"
    "net/http"
)

func handler(w http.ResponseWriter, r *http.Request) {
    if r.URL.Path == "/" {
        fmt.Fprintf(w, "Hello World! Append a name to the URL to say hello. For example, use %s/Mary to say hello to Mary.", r.Host)
    } else {
        fmt.Fprintf(w, "Hello, %s!", r.URL.Path[1:])
    }
}

func main() {
    http.HandleFunc("/", handler)
    http.ListenAndServe(":5000", nil)
}
```

Implantar seu aplicativo Go com a EB CLI

Depois, você criará o ambiente da aplicação e implantará a aplicação configurada com o Elastic Beanstalk.

Para criar um ambiente e implantar o aplicativo Go

1. Inicialize o repositório da EB CLI com o comando `eb init`:

```
~/eb-go$ eb init -p go go-tutorial --region us-east-2
```

```
Application go-tutorial has been created.
```

Esse comando cria um aplicativo denominado `go-tutorial` e configura seu repositório local para criar ambientes com a versão da plataforma Go mais recente.

2. (Opcional) Execute `eb init` novamente para configurar um par de chaves padrão para que você possa usar SSH para se conectar à instância do EC2 que está executando seu aplicativo:

```
~/eb-go$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

Selecione um par de chaves se você já tiver um, ou siga as instruções na tela para criar um. Se nenhum prompt aparecer ou se for necessário alterar as configurações posteriormente, execute `eb init -i`.

3. Crie um ambiente e implante o aplicativo nele com `eb create`. O Elastic Beanstalk cria automaticamente um arquivo binário para a aplicação e o inicia na porta 5000.

```
~/eb-go$ eb create go-env
```

A criação de ambiente leva cerca de cinco minutos e cria os seguintes recursos:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.

- **Balancedador de carga:** um balancedador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- **Grupo de segurança do balancedador de carga:** um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- **Grupo de Auto Scaling:** grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- **Bucket do Amazon S3:** um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- **CloudWatch Alarmes da Amazon** — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- **AWS CloudFormation stack** — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- **Nome de domínio:** um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

O Elastic Beanstalk gerencia todos esses recursos. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Note

O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para ter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon S3](#).

Quando o processo de criação do ambiente estiver concluído, abra o site com `eb open`:

```
~/eb-go$ eb open
```

Isso abre uma janela do navegador usando o nome de domínio criado para o seu aplicativo.

Se o aplicativo não aparecer em execução ou se houver uma mensagem de erro, consulte [Solução de erros de implantação](#) para obter ajuda sobre como determinar a causa do erro.

Se a aplicação aparecer em execução, você implantou sua primeira aplicação Go com o Elastic Beanstalk, parabéns!

Limpar

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. [O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente, como instâncias do Amazon EC2, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.](#)

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

Ou, com a EB CLI, faça o seguinte.

```
~/eb-go$ eb terminate
```

Criar e implantar aplicações Java no Elastic Beanstalk

AWS Elastic Beanstalk oferece suporte a duas plataformas para aplicações Java.

- **Tomcat:** uma plataforma baseada no Apache Tomcat, um contêiner da Web de código aberto para aplicações que usam servlets Java e JavaServer Pages (JSPs) para atender a solicitações HTTP. O Tomcat facilita o desenvolvimento de aplicativos web, fornecendo multithreading, configuração de segurança declarativa e ampla personalização. O Elastic Beanstalk tem ramificações de plataforma para cada uma das principais versões atuais do Tomcat. Para obter mais informações, consulte [A Plataforma Tomcat](#).
- **Java SE:** uma plataforma para aplicações que não usam um contêiner da Web ou usam um que não seja Tomcat, como Jetty ou GlassFish. É possível incluir quaisquer Java Archives (JARs) de biblioteca usados pela sua aplicação no pacote de origem implantado no Elastic Beanstalk. Para obter mais informações, consulte [Plataforma Java SE](#).

As ramificações recentes das plataformas Tomcat e Java SE são baseadas no Amazon Linux 2 e versões posteriores e usam o Corretto: a distribuição da AWS do Java SE. Os nomes dessas ramificações nas listas de plataformas incluem a palavra Corretto em vez de Java, por exemplo, Corretto 11 with Tomcat 8.5.

Para obter uma lista das versões atuais da plataforma, consulte [Tomcat](#) e [Java SE](#) no guia Plataformas AWS Elastic Beanstalk.

AWS fornece várias ferramentas para trabalhar com Java e o Elastic Beanstalk.

Independentemente da ramificação da plataforma que você escolher, é possível usar o [AWS SDK for Java](#) para usar outros serviços da AWS a partir do aplicativo Java. O AWS SDK for Java é um conjunto de bibliotecas que permitem que você use as APIs da AWS a partir do seu código de aplicativo sem gravar as chamadas HTTP brutas do zero.

Se você usar um ambiente de desenvolvimento integrado do Eclipse (IDE) para desenvolver o aplicativo Java, também poderá obter o [AWS Toolkit for Eclipse](#). O AWS Toolkit for Eclipse é um plugin de código aberto que permite a você gerenciar os recursos da AWS, incluindo aplicações e ambientes do Elastic Beanstalk, a partir do Eclipse IDE.

Se você preferir usar a linha de comando, instale a [Interface de Linhas de Comando do Elastic Beanstalk](#) (CLI da EB) e use-a para criar, monitorar e gerenciar seus ambientes do Elastic Beanstalk a partir da linha de comando. Se você executar vários ambientes para o seu aplicativo, a EB CLI se integrará com o Git para permitir que você associe cada um de seus ambientes a diferentes ramificações Git.

Os tópicos neste capítulo pressupõem que você tem um nível básico de conhecimento sobre ambientes do Elastic Beanstalk. Se você não tiver usado o Elastic Beanstalk antes, tente o [tutorial de conceitos básicos](#) para saber os princípios.

Tópicos

- [Conceitos básicos do Java no Elastic Beanstalk](#)
- [Configurar seu ambiente de desenvolvimento Java](#)
- [Usar a plataforma Tomcat do Elastic Beanstalk](#)
- [Usar a plataforma Java SE do Elastic Beanstalk](#)
- [Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente de aplicações Java](#)
- [Uso do AWS Toolkit for Eclipse](#)
- [Recursos](#)

Conceitos básicos do Java no Elastic Beanstalk

Para começar a usar os aplicativos Java no AWS Elastic Beanstalk, tudo o que você precisa é de um [pacote de origem](#) de aplicativo para fazer upload como a primeira versão do aplicativo e implantar em um ambiente. Quando você cria um ambiente, o Elastic Beanstalk aloca todos os recursos da AWS necessários para executar uma aplicação Web escalável.

Iniciar um ambiente com um aplicativo Java de exemplo

O Elastic Beanstalk fornece aplicações demonstrativas de página única para cada plataforma e também exemplos mais complexos que mostram o uso de recursos adicionais da AWS, como o Amazon RDS e linguagem ou recursos e APIs específicos da plataforma.

Os exemplos de página única são o mesmo código que você obtém ao criar um ambiente sem fornecer seu próprio código-fonte. Os exemplos mais complexos são hospedados no GitHub e podem ser compilados ou criados antes da implantação em um ambiente do Elastic Beanstalk.

Amostras

Nome (Página única)	Versões com suporte	Tipo de ambiente	Origem	Descrição
Tomcat	Todas as ramificações da plataforma Tomcat com Corretto	Serviço Web Operado	tomcat.apache.org	<p>Aplicativo web Tomcat com uma única página (<code>index.jsp</code>) configurada para ser exibida na raiz do site.</p> <p>Para ambientes de operador, este exemplo inclui um arquivo <code>cron.yaml</code> que configura uma tarefa programada que chama <code>scheduled.jsp</code> uma vez por minuto. Quando <code>scheduled.jsp</code> é chamado, ele grava em um arquivo de log em <code>/tmp/sample-app.log</code>. Por fim, um arquivo de configuração é incluído em <code>.ebextensions</code> que copia os logs de <code>/tmp/</code> para os locais lidos pelo Elastic Beanstalk quando você solicita logs de ambiente.</p> <p>Se você habilitar a integração com o X-Ray em um ambiente que executa este exemplo, a aplicação mostrará conteúdo adicional relativo ao X-Ray e fornecerá uma opção para gerar informações de depuração que você pode visualizar no console do X-Ray.</p>
Corretto (Página única)	Corretto 11 Corretto 8	Serviço Web Operado	corretto.aws.com	<p>Aplicação Corretto com arquivos de configuração <code>Buildfile</code> e <code>Procfile</code>.</p> <p>Se você habilitar a integração com o X-Ray em um ambiente que executa este exemplo, a aplicação mostrará conteúdo adicional relativo ao X-Ray e fornecerá</p>

Nome (Nome)	Versões com suporte	Tipo de ambiente	Origem	Descrição
				uma opção para gerar informações de depuração que você pode visualizar no console do X-Ray.

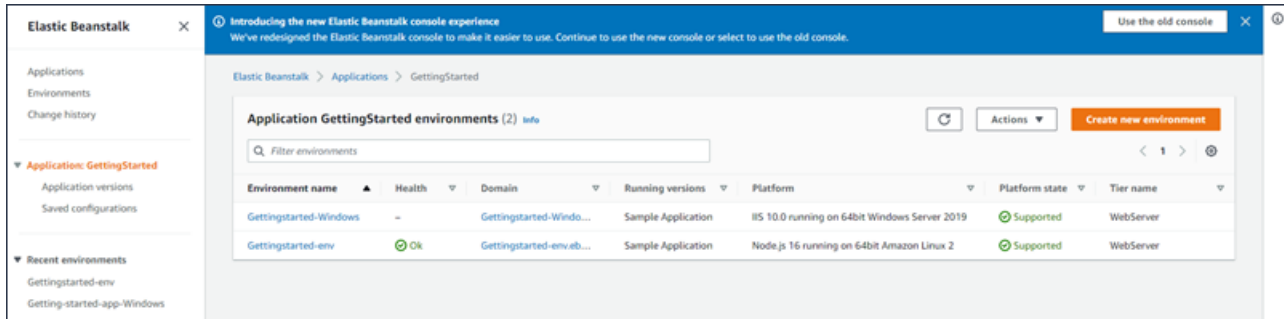
Nome (Nome)	Versões com suporte	Tipo de ambiente	Origem	Descrição
Scorekeep	Java 8	Serviço Web	Clonar o repo em GitHub.com	<p>O Scorekeep é uma API web RESTful que usa o Spring framework para fornecer uma interface para a criar e gerenciar usuários, sessões e jogos. A API faz parte do pacote de um aplicativo Web Angular 1.5 que a consome via HTTP.</p> <p>O aplicativo usa recursos da plataforma a Java SE para fazer download de dependências e criar na instância, minimizando o tamanho do pacote de origem. O aplicativo também inclui arquivos de configuração nginx que substituem a configuração padrão para servir o aplicativo o web de front-end estaticamente na porta 80 através do proxy e encaminhar as solicitações para atalhos em <code>/api</code> para a API em execução em <code>localhost:5000</code>.</p> <p>O Scorekeep também inclui uma ramificação <code>xray</code> que mostra como instrumentar uma aplicação Java para uso com o AWS X-Ray. Ele mostra a instrumentação de solicitações HTTP de entrada com um filtro de servlet, instrumentação de cliente AWS SDK automática e manual, configuração de gravador e instrumentação de solicitações HTTP de saída e clientes SQL.</p> <p>Consulte o arquivo readme para obter instruções ou use o tutorial sobre conceitos básicos do AWS X-Ray para experimentar a aplicação com o X-Ray.</p>

Nome (Nome)	Versões com suporte	Tipo de ambiente	Origem	Descrição
Does it Have Snakes?	Tomcat 8 com Java 8	Serviço Web	Clonar o repo em GitHub.com	<p>Does it Have Snakes? é uma aplicação Web Tomcat que mostra o uso de arquivos de configuração do Elastic Beanstalk, Amazon RDS, JDBC, PostgreSQL, servlets, JSPs, Simple Tag Support, arquivos de tag, Log4J, Bootstrap e Jackson.</p> <p>O código-fonte deste projeto inclui um script de compilação mínimo que compila os servlets e os modelos em arquivos de classe e empacota os arquivos necessários em um arquivo web que você pode implantar em um ambiente do Elastic Beanstalk. Consulte o arquivo readme no repositório do projeto para obter instruções completas.</p>
Locust Load Generator	Java 8	Serviço Web	Clonar o repo em GitHub.com	<p>Aplicação Web que pode ser usada para fazer um teste de carga em outra aplicação Web em execução em um ambiente diferente do Elastic Beanstalk. Mostra o uso de arquivos Buildfile e Procfile, DynamoDB e Locust, uma ferramenta de teste de carga de código aberto.</p>

Faça download de qualquer uma das aplicações demonstrativas e implante-a no Elastic Beanstalk seguindo estas etapas:

Para iniciar um ambiente com um aplicativo de exemplo (console)

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, escolha Aplicativos e o nome de um aplicativo existente na lista ou [crie um](#).
3. Na página de visão geral da aplicação, escolha Create a new environment (Criar um novo ambiente).



Isso inicia o assistente Create environment (Criar ambiente). O assistente fornece um conjunto de etapas para a criação de um novo ambiente.

Step 1
Configure environment

Step 2
Configure service access

Step 3 - optional
Configure instance traffic and scaling

Step 4 - optional
Set up networking, database, and tags

Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

Configure environment [Info](#)

Environment tier [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- Web server environment**
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
- Worker environment**
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information [Info](#)

Application name
GettingStarted
Maximum length of 100 characters.

▶ **Application tags (optional)**

Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name
GettingStarted-env
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain name
Leave blank for autogenerated value .us-east-1.elasticbeanstalk.com [Check availability](#)

Environment description

Platform [Info](#)

Platform type

- Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)
- Custom platform**
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform
Choose a platform ▼


Platform branch
Choose a platform branch ▼

Platform version
Choose a platform version ▼

Application code [Info](#)


- Sample application**
- Existing version**
Application versions that you have uploaded.
Sample Application ▼
- Upload your code**
Upload a source bundle from your computer or copy one from Amazon S3.

4. Para nível de ambiente, escolha o [nível de ambiente](#) Web server environment (Ambiente do servidor Web) ou Worker environment (Ambiente do operador). Você não pode alterar o nível de um ambiente após a criação.

 Note

O [.NET na Plataforma de servidor Windows](#) não tem suporte para a camada de ambiente do operador.

5. Em Plataforma, selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.

 Note

O Elastic Beanstalk é compatível com várias [versões](#) para a maioria das plataformas listadas. Por padrão, o console seleciona a versão recomendada da plataforma e a ramificação da plataforma escolhida. Se a aplicação exigir outra versão, você poderá selecioná-la aqui. Para obter informações sobre versões de plataforma compatíveis, consulte [the section called “Plataformas compatíveis”](#).

6. Para Application code, escolha Sample application.
7. Em Configuration presets (Predefinições de configuração), escolha Single instance (Instância única).
8. Escolha Next (Próximo).
9. A página Configurar acesso ao serviço é exibida.

Configure service access [Info](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role
 Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

[View permission details](#)

Cancel Skip to review Previous **Next**

10. Escolha Usar um perfil de serviço existente em Perfil de serviço.

11. Em seguida, vamos nos concentrar na lista suspensa Perfil de instância do EC2. Os valores exibidos nessa lista suspensa podem variar, dependendo de a conta ter criado anteriormente um novo ambiente.

Escolha uma das opções a seguir com base nos valores exibidos na lista.

- Se `aws-elasticbeanstalk-ec2-role` for exibido na lista suspensa, selecione-o na lista suspensa Perfil de instância do EC2.
- Se outro valor for exibido na lista e for o perfil padrão de instância do EC2 para seus ambientes, selecione-o na lista suspensa Perfil de instância do EC2.
- Se a lista suspensa Perfil de instância do EC2 não mostrar nenhum valor para seleção, expanda o procedimento que se segue, Criar um perfil do IAM para ser o perfil de instância do EC2.

Conclua as etapas em Criar perfil do IAM para ser o perfil de instância do EC2 para criar um perfil do IAM que você possa selecionar posteriormente para ser o perfil de instância do EC2. Depois, retorne para esta etapa.

Agora que você criou um perfil do IAM e atualizou a lista, ele será exibido como uma opção na lista suspensa. Selecione o perfil do IAM que você acabou de criar na lista suspensa Perfil de instância do EC2.

12. Escolha Skip to Review (Pular para revisão) na página Configure service access (Configurar acesso a serviço).

Isso selecionará os valores padrão para essa etapa e pulará as etapas opcionais.

13. A página Review (Revisão) exibe um resumo de todas as suas escolhas.

Para personalizar ainda mais o ambiente, escolha Edit (Editar) ao lado da etapa que inclui os itens que você deseja configurar. Você pode configurar as opções a seguir somente durante a criação do ambiente:

- Nome do ambiente
- Nome de domínio
- Versão da plataforma
- Processor
- VPC
- Nível

Você pode alterar as configurações a seguir após a criação do ambiente, mas elas exigem que novas instâncias ou outros recursos sejam provisionados, e pode levar muito tempo para que elas sejam aplicadas:

- Tipo de instância, volume raiz, par de chaves e função do AWS Identity and Access Management (IAM)
- Banco de dados interno do Amazon RDS
- Load balancer

Para obter detalhes sobre todas as configurações disponíveis, consulte [O assistente de criação de novo ambiente](#).

14. Escolha Submit (enviar) na parte inferior da página para inicializar a criação do novo ambiente.

Criar um perfil do IAM para ser o perfil de instância do EC2

The screenshot shows the 'Configure service access' dialog box in the AWS IAM console. The title is 'Configure service access' with an 'Info' link. Below the title is a 'Service access' section with a brief explanation and a 'Learn more' link. The 'Service role' section has two radio buttons: 'Create and use new service role' (unselected) and 'Use an existing service role' (selected). Under 'Existing service roles', there is a text input field containing 'aws-elasticbeanstalk-service-role' and a refresh button. The 'EC2 key pair' section has a text input field with 'Choose a key pair' and a refresh button. The 'EC2 instance profile' section has a text input field containing 'aws-elasticbeanstalk-ec2-role' and a refresh button. Below these fields is a 'View permission details' button. At the bottom of the dialog are four buttons: 'Cancel', 'Skip to review', 'Previous', and 'Next' (highlighted in orange).

Para criar um perfil do IAM para ser o perfil de instância do EC2


1. Escolha Visualizar detalhes de permissão. Essa opção exibe a lista suspensa Perfil de instância do EC2.

Uma janela modal intitulada Visualizar permissões de perfil de instância é exibida. Essa janela lista os perfis gerenciados que deverão ser anexados ao novo perfil de instância do EC2 que você criar. Ela também fornece um link para iniciar o console do IAM.

2. Escolha o link Console do IAM exibido na parte superior da janela.
3. No painel de navegação do console do IAM, escolha Roles (Perfis).
4. Selecione Create role (Criar função).
5. Em Tipo de entidade confiável, selecione Serviço da AWS.
6. Em Use case (Caso de uso), selecione EC2.
7. Escolha Next (Próximo).
8. Anexe as políticas gerenciadas apropriadas. Role a janela modal Visualizar permissões de perfil de instância para ver as políticas gerenciadas. As políticas também estão listadas aqui:

- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

9. Escolha Next (Próximo).
10. Insira um nome para a função.
11. (Opcional) Adicione tags à função.
12. Selecione Create role (Criar função).
13. Retorne à janela do console do Elastic Beanstalk que está aberta.
14. Feche a janela modal Visualizar permissões de perfil de instância.

 Important

Não feche a página do navegador que exibe o console do Elastic Beanstalk.

15. Escolha



(atualizar), ao lado da lista suspensa Perfil de instância do EC2.

Essa opção atualiza a lista suspensa para que o perfil que você acabou de criar seja exibido na lista suspensa.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá [implantar uma nova versão](#) do aplicativo ou um aplicativo completamente diferente a qualquer momento. A implantação de uma nova versão do aplicativo é muito rápida, pois não é necessário provisionar ou reiniciar as instâncias do EC2.

Depois de implantar um ou dois aplicativos de exemplo e estiver pronto para começar a desenvolver e executar aplicativos Java localmente, consulte [a próxima seção](#) para configurar um ambiente de desenvolvimento Java com todas as ferramentas e bibliotecas necessárias.

Configurar seu ambiente de desenvolvimento Java

Configure um ambiente de desenvolvimento Java para testar sua aplicação localmente antes de implantá-lo no AWS Elastic Beanstalk. Este tópico descreve as etapas de configuração do ambiente de desenvolvimento e inclui links para páginas de instalação de ferramentas úteis.

Para ver as etapas de configuração comuns e as ferramentas que se aplicam a todas as linguagens, consulte [Configurar a máquina de desenvolvimento](#).

Seções

- [Instalar o Java Development Kit](#)
- [Instalar um contêiner da web](#)
- [Download de bibliotecas](#)
- [Instalação do AWS SDK for Java](#)
- [Instalar um editor de texto ou IDE](#)
- [Instalação do AWS Toolkit for Eclipse](#)

Instalar o Java Development Kit

Instale o Java Development Kit (JDK). Se você não tiver uma preferência, obtenha a versão mais recente. Faça download do JDK em oracle.com

O JDK inclui o compilador Java, que você pode usar para criar arquivos de origem em arquivos de classe que podem ser executados em um servidor Web Elastic Beanstalk.

Instalar um contêiner da web

Se você ainda não tem outro contêiner da Web ou framework, instale a versão apropriada do Tomcat:

- [Fazer download do Tomcat 8 \(requer Java 7 ou posterior\)](#)
- [Fazer download do Tomcat 7 \(requer Java 6 ou posterior\)](#)

Download de bibliotecas

As plataformas Elastic Beanstalk incluem poucas bibliotecas por padrão. Faça download das bibliotecas que seu aplicativo usará e salve-as na pasta do projeto para implantar no pacote de origem do aplicativo.

Se você instalou o Tomcat localmente, pode copiar a API do servlet e as bibliotecas de API JavaServer Pages (JSP) da pasta de instalação. Se você implantar em uma versão de plataforma Tomcat, não precisará incluir esses arquivos em seu pacote de origem, mas deverá tê-los no classpath para compilar qualquer classe que os utilizem.

JUnit, Google Guava e Apache Commons oferecem várias bibliotecas úteis. Visite a página inicial deles para saber mais:

- [Download do JUnit](#)
- [Download do Google Guava](#)
- [Download do Apache Commons](#)

Instalação do AWS SDK for Java

Se você precisa gerenciar os recursos da AWS usando sua aplicação, instale o AWS SDK for Java. Por exemplo, com o AWS SDK for Java, é possível usar o Amazon DynamoDB (DynamoDB) para compartilhar estados de sessão de aplicações Apache Tomcat em vários servidores da Web. Para obter mais informações, consulte [Gerenciar o estado de sessão do Tomcat com o Amazon DynamoDB](#) na documentação do AWS SDK for Java.

Visite a página inicial do [AWS SDK for Java](#) para obter mais informações e instruções de instalação.

Instalar um editor de texto ou IDE

Os Integrated development environments (IDEs – Ambientes de desenvolvimento integrados) oferecem uma ampla variedade de recursos que facilitam o desenvolvimento de aplicativos. Se você não usou um IDE para desenvolvimento Java, tente o Eclipse e o IntelliJ e veja qual é melhor para você.

- [Instalação do IDE Eclipse para Java EE Developers](#)
- [Instalação do IntelliJ](#)

Note

Um IDE pode adicionar arquivos à pasta do seu projeto, que talvez você não queira confirmar no controle de código-fonte. Para evitar a confirmação desses arquivos no controle de código-fonte, use `.gitignore` ou a ferramenta de controle de código-fonte equivalente.

Se você deseja apenas começar a codificação e não precisa de todos os recursos de um IDE, considere [instalar o Sublime Text](#).

Instalação do AWS Toolkit for Eclipse

O [AWS Toolkit for Eclipse](#) é um plugin de código aberto para o IDE Java Eclipse que facilita o desenvolvimento, a depuração e a implantação de aplicações Java para os desenvolvedores que usam a AWS. Visite a página inicial do [AWS Toolkit for Eclipse](#) para obter instruções de instalação.

Usar a plataforma Tomcat do Elastic Beanstalk

Important

O AWS Elastic Beanstalk instala o Log4j dos repositórios de pacotes padrão do Amazon Linux em suas plataformas Tomcat para Amazon Linux 1 e Amazon Linux 2. As versões do Log4j disponíveis nos repositórios Amazon Linux 1 e Amazon Linux 2 não são afetadas pela [CVE-2021-44228](#) ou pela [CVE-2021-45046](#) em sua configuração padrão.

Se você fez alterações de configuração no uso do log4j pela aplicação ou instalou versões mais recentes do log4j, recomendamos que tome medidas para atualizar o código da aplicação para atenuar esse problema.

Por precaução, o Elastic Beanstalk lançou novas versões de plataforma que usam os repositórios de pacotes padrão do Amazon Linux mais recentes, que incluem o [JDK do Log4j com hot patch](#), em nossa [versão da plataforma Amazon Linux de 21 de dezembro de 2021](#). Se você personalizou a instalação do log4j como dependência da aplicação, recomendamos que atualize para a versão mais recente da plataforma Elastic Beanstalk para atenuar a CVE-2021-44228 ou a CVE-2021-45046. Você também pode habilitar atualizações gerenciadas automatizadas como parte das práticas normais de atualização.

Para obter mais informações sobre atualizações de software relacionadas à segurança do Amazon Linux, consulte a [Central de Segurança do Amazon Linux](#).

A plataforma Tomcat do AWS Elastic Beanstalk é um conjunto de [versões de plataforma](#) para aplicativos web em Java que podem ser executados em um contêiner da web Tomcat. Tomcat é executado atrás de um servidor de proxy nginx. Cada ramificação de plataforma corresponde a uma versão principal do Tomcat, como Java 8 com Tomcat 8.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para [modificar a configuração de um ambiente em execução](#). Para evitar a perda da configuração do ambiente

quando você terminá-la, você pode usar [configurações salvas](#) para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir [arquivos de configuração](#). As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

A plataforma Tomcat do Elastic Beanstalk inclui um proxy reverso que encaminha solicitações para a aplicação. É possível usar [opções de configuração](#) para configurar o servidor de proxy para servir os ativos estáticos de uma pasta em seu código-fonte para reduzir a carga em seu aplicativo. Em cenários avançados, é possível [incluir seus próprios arquivos .conf](#) em seu pacote de origem para estender a configuração de proxy do Elastic Beanstalk; ou substituí-la completamente.

Note

O Elastic Beanstalk é compatível com [nginx](#) (o padrão) e ao [Apache HTTP Server](#) como servidores de proxy na plataforma Tomcat. Se o seu ambiente do Elastic Beanstalk Tomcat usar uma ramificação de plataforma Amazon Linux AMI (anterior ao Amazon Linux 2), você também terá a opção de usar o [Apache HTTP Server Versão 2.2](#). O Apache (mais recente) é o padrão nessas ramificações de plataforma mais antigas.

Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Você deve empacotar os aplicativos Java em um arquivo WAR (Web application archive) com uma estrutura específica. Para obter informações sobre a estrutura necessária e como ela está relacionada à estrutura do diretório de seu projeto, consulte [Estruturar a pasta do projeto](#).

Para executar vários aplicativos no mesmo servidor Web, você pode [empacotar vários arquivos WAR](#) em um único pacote de origem. Cada aplicativo em um pacote de origem WAR múltiplo é executado no caminho raiz (ROOT.war é executado em `myapp.elasticbeanstalk.com/`) ou em um caminho diretamente abaixo dele (`app2.war` é executado no `myapp.elasticbeanstalk.com/app2/`), conforme determinado pelo nome do WAR. Em um pacote de origem de WAR único, o aplicativo sempre é executado no caminho raiz.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console. Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte [Opções de configuração](#).

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte [the section called “Estender plataformas Linux”](#).

Tópicos

- [Configurar o grupo do seu ambiente Tomcat](#)
- [Namespaces de configuração do Tomcat](#)
- [Empacotamento de vários arquivos WAR para ambientes Tomcat](#)
- [Estruturar a pasta do projeto](#)
- [Configurar servidor de proxy do seu ambiente Tomcat](#)

Configurar o grupo do seu ambiente Tomcat

A plataforma Tomcat do Elastic Beanstalk fornece algumas opções específicas da plataforma, além de opções padrão que todas as plataformas têm. Essas opções permitem que você configure a máquina virtual Java (JVM), que é executada nos servidores web do ambiente, e defina propriedades do sistema que fornecem strings de configuração de informações ao aplicativo.

É possível usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3 e configurar variáveis que sua aplicação pode ler no ambiente.

Como configurar seu ambiente Tomcat no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).

4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Opções de contêiner

É possível definir estas opções específicas da plataforma:

- Proxy server (Servidor de proxy): o servidor de proxy a ser usado nas instâncias de ambiente. Por padrão, é usado nginx.

Opções de contêiner JVM

O tamanho do heap na máquina virtual Java (JVM) determina quantos objetos seu aplicativo pode criar na memória antes da execução da [coleta de lixo](#). É possível modificar o Tamanho do heap JVM inicial (`-Xms` option) e um Tamanho do heap JVM máximo (opção `-Xmx`). Um tamanho de heap inicial maior permite que mais objetos sejam criados antes da execução da coleta de lixo, mas também significa que o coletor de lixo levará mais tempo para compactar o heap. O tamanho máximo do heap especifica a quantidade máxima de memória que a JVM pode alocar ao expandir o heap durante uma atividade intensa.

Note

A memória disponível depende do tipo de instância do Amazon EC2. Para obter mais informações sobre os tipos de instância do EC2 disponíveis para o ambiente do Elastic Beanstalk, consulte [Tipos de instância](#) no Guia do usuário do Amazon Elastic Compute Cloud.

A geração permanente é uma seção do heap da JVM que armazena definições de classe e metadados associados. Para modificar o tamanho da geração permanente, digite o novo tamanho na opção Tamanho do PermGen JVM máximo (`-XX:MaxPermSize`). Essa configuração se aplica apenas ao Java 7 e versões anteriores. Essa opção foi descontinuada no JDK 8 e substituída pela opção Tamanho do MaxMetaspace (`-XX:MaxMetaspaceSize`).

Important

O JDK 17 removeu a compatibilidade com a opção `-XX:MaxPermSize` do Java. O uso dessa opção com um ambiente em execução em uma ramificação da plataforma Elastic

Beanstalk com o Corretto 17 resultará em um erro. O Elastic Beanstalk lançou sua primeira ramificação da plataforma executando o Tomcat com o Corretto 17 em [13 de julho de 2023](#). Para obter mais informações, consulte os recursos a seguir.

- Site de documentação do Oracle Java: [Opções do Java removidas](#)
- Site de documentação do Oracle Java: seção Metadados da classe em [Outras considerações](#)

Para obter mais informações sobre as plataformas do Elastic Beanstalk e seus componentes, consulte [Plataformas compatíveis](#) no Guia de plataformas do AWS Elastic Beanstalk.

Opções de log

A seção Log Options (Opções de log) tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Enable log file rotation to Amazon S3 (Habilitar a rotação de arquivos de log para o Amazon S3): especifica se os arquivos de log das instâncias do Amazon EC2 da aplicação serão copiados para o bucket do Amazon S3 associado a ela.

Arquivos estáticos

Para melhorar a performance, você pode usar a seção Static files (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte [the section called “Arquivos estáticos”](#).

Propriedades de ambiente

Na seção Environment Properties (Propriedades do ambiente), é possível especificar definições de configuração do ambiente nas instâncias do Amazon EC2 que estão executando a aplicação. As propriedades de ambiente são passadas para o aplicativo como pares de chave-valor.

A plataforma Tomcat define uma propriedade de espaço reservado denominada `JDBC_CONNECTION_STRING` para ambientes Tomcat, para passar uma string de conexão a um banco de dados externo.

Note

Se você anexar uma instância de banco de dados RDS ao seu ambiente, construa a string de conexão JDBC dinamicamente a partir das propriedades do ambiente do Amazon Relational Database Service (Amazon RDS) fornecidas pelo Elastic Beanstalk. Use `JDBC_CONNECTION_STRING` apenas para instâncias de banco de dados que não são provisionadas usando o Elastic Beanstalk.

Para obter mais informações sobre o uso do Amazon RDS com a aplicação Java, consulte [Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente de aplicações Java](#).

No ambiente Tomcat executado no Elastic Beanstalk, as variáveis de ambiente podem ser acessadas usando `System.getProperty()`. Por exemplo, você pode ler uma propriedade denominada `API_ENDPOINT` em uma variável com o seguinte código.

```
String endpoint = System.getProperty("API_ENDPOINT");
```

Consulte [Propriedades de ambientes e outras configurações de software](#) para obter mais informações.

Namespaces de configuração do Tomcat

Você pode usar um [arquivo de configuração](#) para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. As opções de configuração podem ser definidas pelo serviço Elastic Beanstalk ou pela plataforma utilizada, e são organizadas em namespaces.

A Plataforma Tomcat é compatível com as opções nos namespaces a seguir, além das [opções compatíveis com todos os ambientes do Elastic Beanstalk](#):

- `aws:elasticbeanstalk:container:tomcat:jvmoptions`: modificar as configurações do JVM. As opções desse namespace correspondem às do console de gerenciamento, da seguinte maneira:
 - `Xms` – JVM command line options (Opções de linha de comando da JVM)

- JVM Options – JVM command line options (Opções de linha de comando da JVM)
- `aws:elasticbeanstalk:environment:proxy`: escolha o servidor de proxy do ambiente.

O seguinte arquivo de configuração de exemplo mostra o uso das opções de configuração específicas do Tomcat.

Example `.ebextensions/tomcat-settings.config`

```
option_settings:
  aws:elasticbeanstalk:container:tomcat:jvmoptions:
    Xms: 512m
    JVM Options: '-Xmn128m'
  aws:elasticbeanstalk:application:environment:
    API_ENDPOINT: mywebapi.zkpxsjtmd.us-west-2.elasticbeanstalk.com
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
```

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte [Opções de configuração](#) para obter mais informações.

A plataforma Tomcat da AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o seu ambiente Tomcat do Elastic Beanstalk usar uma versão da plataforma AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

Observações

- As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). As ramificações de plataforma AL2023/AL2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem definições de configuração diferentes.
- Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Namespaces de configurações do Tomcat: Amazon Linux AMI (AL1)

A plataforma Tomcat da AMI do Amazon Linux é compatível com as opções adicionais nos seguintes namespaces:

- `aws:elasticbeanstalk:container:tomcat:jvmoptions`: além das opções mencionadas anteriormente nesta página para esse namespace, versões mais antigas da plataforma da AMI do Amazon Linux também são compatíveis com:
 - `XX:MaxPermSize` – Maximum JVM permanent generation size (Tamanho máximo de geração permanente da JVM)
- `aws:elasticbeanstalk:environment:proxy`: além de escolher o servidor de proxy, configure também a compactação de resposta.

O exemplo de arquivo de configuração a seguir mostra o uso das opções de configuração de namespace do proxy.

Example `.ebextensions/tomcat-settings.config`

```
option_settings:  
  aws:elasticbeanstalk:environment:proxy:  
    GzipCompression: 'true'  
    ProxyServer: nginx
```

Incluir arquivos de configurações do Elastic Beanstalk: Amazon Linux AMI (AL1)

Para implantar arquivos de configuração `.ebextensions`, inclua-os na origem do aplicativo. Para um único aplicativo, adicione `.ebextensions` em um arquivo WAR compactado executando o seguinte comando:

Example

```
zip -ur your_application.war .ebextensions
```

Para uma aplicação que exige vários arquivos WAR, consulte [Empacotamento de vários arquivos WAR para ambientes Tomcat](#) para obter mais instruções.

Empacotamento de vários arquivos WAR para ambientes Tomcat

Se o aplicativo Web consiste em vários componentes do aplicativo da Web, você pode simplificar as implantações e reduzir os custos operacionais, ao executar componentes em um único ambiente,

em vez de executar um ambiente separado para cada componente. Essa estratégia é eficaz para aplicativos leves e econômicos que não exigem uma grande quantidade de recursos e para ambientes de desenvolvimento e teste.

Para implantar vários aplicativos web no ambiente, reúna os arquivos WAR (web application archive) de cada componente em um único [pacote de origem](#).

Para criar um pacote de origem do aplicativo que contenha vários arquivos WAR, organize os arquivos WAR usando a seguinte estrutura.

```
MyApplication.zip
### .ebextensions
### .platform
### foo.war
### bar.war
### ROOT.war
```

Ao implantar um pacote de origem que contém vários arquivos WAR em um ambiente AWS Elastic Beanstalk, cada aplicativo poderá ser acessado a partir de um caminho diferente do nome do domínio raiz. O exemplo anterior inclui três aplicações: foo, bar e ROOT. ROOT.war é um nome de arquivo especial que instrui o Elastic Beanstalk a executar essa aplicação no domínio raiz, de forma que as três aplicações estejam disponíveis em <http://MyApplication.elasticbeanstalk.com/foo>, <http://MyApplication.elasticbeanstalk.com/bar> e <http://MyApplication.elasticbeanstalk.com>.

O pacote de origem pode incluir arquivos WAR, uma pasta `.ebextensions` opcional e uma pasta `.platform` opcional. Para obter detalhes sobre essas pastas de configuração opcionais, consulte [the section called “Estender plataformas Linux”](#).

Para iniciar um ambiente (console)

1. Abra o console do Elastic Beanstalk com este link pré-configurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced
2. Em Plataforma, selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo aplicativo ou a plataforma do Docker para aplicativos baseados em contêiner.
3. Em Application code (Código do aplicativo), escolha Upload your code (Fazer upload do código).

4. Escolha Local file (Arquivo local) e Choose file (Escolher arquivo) e abra o pacote de origem.
5. Selecione Review and launch.
6. Revise as configurações disponíveis e, em seguida, escolha Create app.

Para obter informações sobre como criar pacotes de origem, consulte [Criar um pacote de origem do aplicativo](#).

Estruturar a pasta do projeto

Para funcionarem quando implantados em um servidor Tomcat, os arquivos de aplicativo web (arquivos WAR) do Java Platform Enterprise Edition (Java EE) precisam ser estruturados de acordo com determinadas [diretrizes](#). O diretório de seu projeto não precisa atender aos mesmos padrões, mas é uma boa ideia estruturá-lo da mesma forma para simplificar a compilação e o empacotamento. A estruturação da pasta do projeto de modo similar ao conteúdo do arquivo WAR também ajuda você a entender como os arquivos estão relacionados e como eles se comportam em um servidor da web.

Na hierarquia recomendada a seguir, o código-fonte do aplicativo web é colocado em um diretório `src`, a fim de isolá-lo do script de compilação e do arquivo WAR gerado.

```
~/workspace/my-app/  
|-- build.sh           - Build script that compiles classes and creates a WAR  
|-- README.MD         - Readme file with information about your project, notes  
|-- ROOT.war          - Source bundle artifact created by build.sh  
`-- src               - Source code folder  
    |-- WEB-INF       - Folder for private supporting files  
    |   |-- classes   - Compiled classes  
    |   |-- lib        - JAR libraries  
    |   |-- tags      - Tag files  
    |   |-- tlds       - Tag Library Descriptor files  
    |   |-- web.xml    - Deployment Descriptor  
    |-- com            - Uncompiled classes  
    |-- css            - Style sheets  
    |-- images        - Image files  
    |-- js             - JavaScript files  
    `-- default.jsp    - JSP (JavaServer Pages) webpage
```

O conteúdo da pasta `src` corresponde ao que você empacotará e implantará no servidor, com exceção da pasta `com`. A pasta `com` contém as classes não compiladas (arquivos `.java`). Elas precisam ser compiladas e colocadas no diretório `WEB-INF/classes` para ficarem acessíveis do código do aplicativo.

O diretório WEB-INF contém código e configurações que não são atendidos publicamente no servidor web. As outras pastas na raiz do diretório de origem (css, images e js) estão disponíveis publicamente no caminho correspondente no servidor web.

O exemplo a seguir é idêntico ao diretório do projeto anterior, exceto pelo fato de conter mais arquivos e subdiretórios. Esse exemplo de projeto inclui tags, modelos e classes de suporte simples e um arquivo Java Server Pages (JSP) para um recurso record. Também inclui uma folha de estilo e o JavaScript do [Bootstrap](#), um arquivo JSP padrão e uma página de erro para erros 404.

WEB-INF/lib inclui um arquivo Java Archive (JAR) contendo o driver Java Database Connectivity (JDBC) para PostgreSQL. WEB-INF/classes está vazio porque os arquivos de classe não foram compilados ainda.

```
~/workspace/my-app/  
|-- build.sh  
|-- README.MD  
|-- ROOT.war  
`-- src  
    |-- WEB-INF  
    |   |-- classes  
    |   |-- lib  
    |   |   `-- postgresql-9.4-1201.jdbc4.jar  
    |   |-- tags  
    |   |   `-- header.tag  
    |   |-- tlds  
    |   |   `-- records.tld  
    |   `-- web.xml  
    |-- com  
    |   `-- myapp  
    |       |-- model  
    |       |   `-- Record.java  
    |       `-- web  
    |           `-- ListRecords.java  
    |-- css  
    |   |-- bootstrap.min.css  
    |   `-- myapp.css  
    |-- images  
    |   `-- myapp.png  
    |-- js  
    |   `-- bootstrap.min.js  
    |-- 404.jsp  
    |-- default.jsp
```

```
`-- records.jsp
```

Criar um arquivo WAR com um script Shell

`build.sh` é um script de shell muito simples que compila classes Java, constrói um arquivo WAR e o copia no diretório `webapps` do Tomcat para testes locais.

```
cd src
javac -d WEB-INF/classes com/myapp/model/Record.java
javac -classpath WEB-INF/lib/*:WEB-INF/classes -d WEB-INF/classes com/myapp/model/
Record.java
javac -classpath WEB-INF/lib/*:WEB-INF/classes -d WEB-INF/classes com/myapp/web/
ListRecords.java

jar -cvf ROOT.war *.jsp images css js WEB-INF
cp ROOT.war /Library/Tomcat/webapps
mv ROOT.war ../
```

Dentro do arquivo WAR, você encontra a mesma estrutura existente no diretório `src` no exemplo anterior, com exceção da pasta `src/com`. O comando `jar` cria automaticamente o arquivo `META-INF/MANIFEST.MF`.

```
~/workspace/my-app/ROOT.war
|-- META-INF
|   |-- MANIFEST.MF
|-- WEB-INF
|   |-- classes
|   |   |-- com
|   |       |-- myapp
|   |           |-- model
|   |               |-- Records.class
|   |           |-- web
|   |               |-- ListRecords.class
|   |-- lib
|   |   |-- postgresql-9.4-1201.jdbc4.jar
|   |-- tags
|   |   |-- header.tag
|   |-- tlds
|   |   |-- records.tld
|   |-- web.xml
|-- css
|   |-- bootstrap.min.css
```

```
|  `-- myapp.css
|-- images
|  `-- myapp.png
|-- js
|  `-- bootstrap.min.js
|-- 404.jsp
|-- default.jsp
`-- records.jsp
```

Usar o **.gitignore**

Para evitar a confirmação de arquivos de classe compilados e de arquivos WAR em seu repositório Git ou a visualização de mensagens sobre a exibição deles quando você executa comandos do Git, adicione os tipos de arquivo relevantes a um arquivo denominado `.gitignore` na pasta do projeto.

```
~/workspace/myapp/.gitignore
```

```
*.zip
*.class
```

Configurar servidor de proxy do seu ambiente Tomcat

A plataforma Tomcat usa [nginx](#) (o padrão) ou [Apache HTTP Server](#) como o proxy reverso para retransmitir solicitações da porta 80 na instância para o contêiner da Web Tomcat escutando na porta 8080. O Elastic Beanstalk oferece uma configuração de proxy padrão que pode ser estendida ou substituída completamente por sua própria configuração.

Configurar o servidor proxy na sua versão da plataforma

Todas as plataformas AL2023/AL2 oferecem suporte a um recurso de configuração de proxy uniforme. Para obter mais informações sobre a configuração do servidor proxy nas versões de plataforma que executam o AL2023/AL2, expanda a seção Configuração do proxy reverso em [the section called “Estender plataformas Linux”](#).

Configurar o proxy na plataforma Tomcat da AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o seu ambiente Tomcat do Elastic Beanstalk usar uma versão da plataforma AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

Observações

- As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). As ramificações de plataforma AL2023/AL2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem definições de configuração diferentes.
- Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Escolher um servidor de proxy para seu ambiente Tomcat: Amazon Linux AMI (AL1)

As versões da plataforma Tomcat baseadas na AMI do Amazon Linux (anterior ao Amazon Linux 2) usam o [Apache 2.4](#) para o proxy por padrão. Você pode optar por usar o [Apache 2.2](#) ou o [nginx](#) incluindo um [arquivo de configuração](#) no código-fonte. O exemplo a seguir configura o Elastic Beanstalk para usar nginx.

Example `.ebextensions/nginx-proxy.config`

```
option_settings:  
  aws:elasticbeanstalk:environment:proxy:  
    ProxyServer: nginx
```

Migrar do Apache 2.2 para o Apache 2.4: Amazon Linux AMI (AL1)

Se seu aplicativo foi desenvolvido para o [Apache 2.2](#), leia esta seção para saber mais sobre como migrar para o [Apache 2.4](#).

A partir das configurações da versão 3.0.0 da plataforma Tomcat, que foram lançadas com a [atualização da plataforma Java com Tomcat em 24 de maio de 2018](#), o Apache 2.4 é o proxy padrão da plataforma Tomcat. Os arquivos `.conf` do Apache 2.4 são, em sua maioria, mas não inteiramente, compatíveis com os do Apache 2.2. O Elastic Beanstalk inclui arquivos `.conf` padrão que funcionam corretamente com cada versão do Apache. Se seu aplicativo não personalizar a configuração do Apache, conforme explicado em [Estender e substituir a configuração padrão do Apache: Amazon Linux AMI \(AL1\)](#), ele deve migrar para o Apache 2.4 sem problemas.

Se o seu aplicativo ampliar ou substituir a configuração do Apache, talvez você precise fazer algumas alterações para migrar para o Apache 2.4. Para obter mais informações, consulte [Atualizar da versão 2.2 para 2.4](#) no site The Apache Software Foundation. Como uma medida temporária, até você migrar com êxito para o Apache 2.4, você pode optar por usar o Apache 2.2 com seu aplicativo incluindo o seguinte [arquivo de configuração](#) em seu código-fonte.


Example `.ebextensions/apache-legacy-proxy.config`

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache/2.2
```

Para uma correção rápida, também é possível selecionar o servidor de proxy no console do Elastic Beanstalk.

Como selecionar o proxy no ambiente Tomcat no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

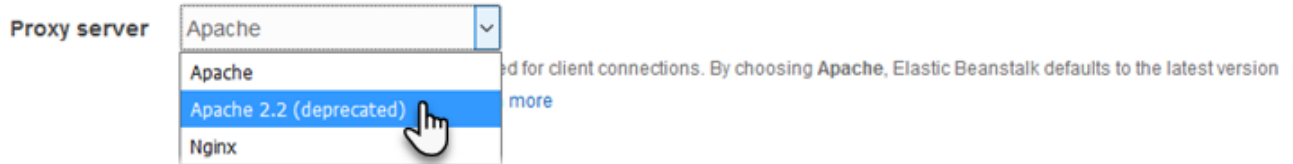
Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Para Proxy server (Servidor de proxy), escolha Apache 2.2 (deprecated).
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Modify software

Container Options

The following settings control container behavior and let you pass key-value pairs in as OS environment variables. [Learn more](#)



Estender e substituir a configuração padrão do Apache: Amazon Linux AMI (AL1)

É possível estender a configuração padrão do Apache do Elastic Beanstalk com seus arquivos de configuração adicionais. Também há opção de substituir completamente a configuração padrão do Apache do Elastic Beanstalk.

Note

- Todas as plataformas do Amazon Linux 2 são compatíveis com um recurso de configuração de proxy uniforme. Para obter detalhes sobre como configurar o servidor de proxy nas novas versões da plataforma que executam o Amazon Linux 2, expanda a seção Configuração de proxy reverso em [the section called “Estender plataformas Linux”](#).
- Se você estiver migrando a aplicação do Elastic Beanstalk para uma plataforma Amazon Linux 2, leia também as informações em [the section called “Migrar para AL2023/AL2”](#).

Para estender a configuração padrão do Apache do Elastic Beanstalk, adicione arquivos de configuração `.conf` a uma pasta chamada `.ebextensions/httpd/conf.d` no pacote de origem da aplicação. A configuração do Apache do Elastic Beanstalk inclui os arquivos `.conf` nessa pasta automaticamente.

```
~/workspace/my-app/  
|-- .ebextensions  
|   -- httpd  
|       -- conf.d  
|           -- myconf.conf  
|           -- ssl.conf  
-- index.jsp
```


Por exemplo, a configuração do Apache 2.4 a seguir adiciona um ouvinte na porta 5000.

Example `.ebextensions/httpd/conf.d/port5000.conf`

```
listen 5000
<VirtualHost *:5000>
  <Proxy *>
    Require all granted
  </Proxy>
  ProxyPass / http://localhost:8080/ retry=0
  ProxyPassReverse / http://localhost:8080/
  ProxyPreserveHost on

  ErrorLog /var/log/httpd/elasticbeanstalk-error_log
</VirtualHost>
```

Para substituir completamente a configuração padrão do Apache do Elastic Beanstalk, inclua uma configuração no pacote de origem em `.ebextensions/httpd/conf/httpd.conf`.

```
~/workspace/my-app/
|-- .ebextensions
|   |-- httpd
|       |-- conf
|           |-- httpd.conf
`-- index.jsp
```

Se você substituir a configuração do Apache do Elastic Beanstalk, adicione as linhas a seguir a `httpd.conf` para extrair as configurações do Elastic Beanstalk para [Monitoramento e relatório de integridade aprimorada](#), compactação de resposta e arquivos estáticos.

```
IncludeOptional conf.d/*.conf
IncludeOptional conf.d/elasticbeanstalk/*.conf
```

Se seu ambiente usar o Apache 2.2 como o proxy, substitua as diretivas `IncludeOptional` por `Include`. Para obter detalhes sobre o comportamento dessas duas diretivas nas duas versões do Apache, consulte [Include no Apache 2.4](#), [IncludeOptional no Apache 2.4](#) e [Include no Apache 2.2](#).

Note

Para substituir o listener padrão na porta 80, inclua um arquivo denominado `00_application.conf` em `.ebextensions/httpd/conf.d/elasticbeanstalk/` para substituir a configuração do Elastic Beanstalk.

Para ver um exemplo prático, examine o arquivo de configuração padrão do Elastic Beanstalk em `/etc/httpd/conf/httpd.conf` em uma instância no ambiente. Todos os arquivos na pasta `.ebextensions/httpd` em seu pacote de origem são copiados para `/etc/httpd` durante as implantações.

Estender a configuração nginx padrão: Amazon Linux AMI (AL1)

Para estender a configuração padrão do nginx do Elastic Beanstalk, adicione arquivos de configuração `.conf` a uma pasta denominada `.ebextensions/nginx/conf.d/` no pacote de origem da sua aplicação. A configuração nginx do Elastic Beanstalk inclui arquivos `.conf` nesta pasta automaticamente.

```
~/workspace/my-app/  
|-- .ebextensions  
|   |-- nginx  
|       |-- conf.d  
|           |-- elasticbeanstalk  
|           |   |-- my-server-conf.conf  
|           |   |-- my-http-conf.conf  
|-- index.jsp
```

Os arquivos com a extensão `.conf` na pasta `conf.d` são incluídos no bloco `http` da configuração padrão. Os arquivos na pasta `conf.d/elasticbeanstalk` são incluídos no bloco `server` dentro do bloco `http`.

Para substituir completamente a configuração nginx padrão do Elastic Beanstalk, inclua uma configuração em seu pacote de origem em `.ebextensions/nginx/nginx.conf`.

```
~/workspace/my-app/  
|-- .ebextensions  
|   |-- nginx  
|       |-- nginx.conf  
|-- index.jsp
```

Observações

- Se você substituir a configuração nginx do Elastic Beanstalk, adicione a linha a seguir ao bloco `server` da configuração para extrair as configurações do Elastic Beanstalk para o listener da porta 80, a compactação de resposta e os arquivos estáticos.

```
include conf.d/elasticbeanstalk/*.conf;
```

- Para substituir o listener padrão na porta 80, inclua um arquivo denominado `00_application.conf` em `.ebextensions/nginx/conf.d/elasticbeanstalk/` para substituir a configuração do Elastic Beanstalk.
- Inclua também a linha a seguir no bloco `http` da configuração para extrair as configurações do Elastic Beanstalk para [Monitoramento e relatório de integridade aprimorada](#) e registro em log.

```
include conf.d/*.conf;
```

Para ver um exemplo prático, examine o arquivo de configuração padrão do Elastic Beanstalk em `/etc/nginx/nginx.conf` em uma instância no ambiente. Todos os arquivos na pasta `.ebextensions/nginx` em seu pacote de origem são copiados para `/etc/nginx` durante as implantações.

Usar a plataforma Java SE do Elastic Beanstalk

A plataforma Java SE do AWS Elastic Beanstalk é um conjunto de [versões de plataforma](#) para aplicativos web em Java que podem ser executados por conta própria em um arquivo JAR compilado. Você pode compilar seu aplicativo localmente ou fazer o upload do código-fonte com um script de compilação para compilá-lo na instância. As versões da plataforma Java SE são agrupadas em ramificações da plataforma, cada uma das quais corresponde a uma versão principal do Java, por exemplo Java 8 e Java 7.

Note

O Elastic Beanstalk não analisa o arquivo JAR da sua aplicação. Mantenha os arquivos de que o Elastic Beanstalk precisa fora do arquivo JAR. Por exemplo, inclua o arquivo

`cron.yaml` de um [ambiente de trabalho](#) na raiz do pacote de origem do seu aplicativo, ao lado do arquivo JAR.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para [modificar a configuração de um ambiente em execução](#). Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar [configurações salvas](#) para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir [arquivos de configuração](#). As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

A plataforma Java SE do Elastic Beanstalk inclui um servidor [nginx](#) que atua como um proxy reverso, suprimindo conteúdo estático em cache e enviando solicitações à aplicação. A plataforma também oferece opções de configuração para definir o servidor de proxy para suprir os ativos estáticos de uma pasta em seu código-fonte para reduzir a carga em seu aplicativo. Em cenários avançados, você pode [incluir seus próprios arquivos .conf](#) em seu pacote de origem para estender a configuração de proxy do Elastic Beanstalk; ou substituí-la completamente.

Se você fornecer somente um único arquivo JAR para a origem da aplicação (por conta própria, não dentro de um pacote de origem), o Elastic Beanstalk altera o nome do arquivo JAR para `application.jar` e o executa usando `java -jar application.jar`. Para configurar os processos que são executados em instâncias do servidor em seu ambiente, inclua um [Procfile](#) opcional em seu pacote de origem. Um `Procfile` será necessário se você tiver mais de um JAR na raiz do pacote de origem, ou se desejar personalizar o comando Java para definir opções de JVM.

Recomendamos sempre fornecer um `Procfile` no pacote de origem juntamente com o aplicativo. Dessa forma, você controla com precisão quais processos são executados pelo Elastic Beanstalk para o aplicativo e quais argumentos são recebidos por esses processos.

Para compilar classes Java e executar outros comandos de build nas instâncias do EC2 em seu ambiente no momento da implantação, inclua um [Buildfile](#) no pacote de origem do aplicativo. O `Buildfile` permite que você implante o código-fonte no estado em que se encontra e crie no servidor, em vez de compilar os JARs localmente. A plataforma Java SE inclui ferramentas de build comuns para permitir que você crie no servidor.

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte [the section called “Estender plataformas Linux”](#).

Configurar o ambiente Java SE

As configurações da plataforma Java SE permitem que você ajuste o comportamento das instâncias do Amazon EC2. É possível editar a configuração da instância do Amazon EC2 do ambiente do Elastic Beanstalk usando o console do Elastic Beanstalk.

Você pode usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3 e configurar variáveis que sua aplicação pode ler no ambiente.

Como configurar seu ambiente Java SE no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Opções de log

A seção Log Options tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Enable log file rotation to Amazon S3 (Habilitar a rotação de arquivos de log para o Amazon S3): especifica se os arquivos de log das instâncias do Amazon EC2 da aplicação serão copiados para o bucket do Amazon S3 associado a ela.

Arquivos estáticos

Para melhorar a performance, você pode usar a seção `Static files` (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte [the section called “Arquivos estáticos”](#).

Propriedades de ambiente

A seção `Environment Properties` permite que você especifique definições de configuração do ambiente nas instâncias do Amazon EC2 que estão executando seu aplicativo. As propriedades de ambiente são passadas para o aplicativo como pares de chave-valor.

No ambiente Java SE executado no Elastic Beanstalk, as variáveis de ambiente podem ser acessadas usando `System.getenv()`. Por exemplo, você pode ler uma propriedade denominada `API_ENDPOINT` em uma variável com o seguinte código:

```
String endpoint = System.getenv("API_ENDPOINT");
```

Consulte [Propriedades de ambientes e outras configurações de software](#) para obter mais informações.

Namespace de configuração do Java SE

Você pode usar um [arquivo de configuração](#) para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. As opções de configuração podem ser definidas pelo serviço Elastic Beanstalk ou pela plataforma utilizada, e são organizadas em namespaces.

A plataforma Java SE não define nenhum namespace específico da plataforma.

É possível configurar o proxy para servir arquivos estáticos usando o namespace `aws:elasticbeanstalk:environment:proxy:staticfiles`. Para obter detalhes e um exemplo, consulte [the section called “Arquivos estáticos”](#).

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console,

configurações salvas, a EB CLI ou a AWS CLI. Consulte [Opções de configuração](#) para obter mais informações.

A plataforma Java SE da AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o seu ambiente Java SE do Elastic Beanstalk usa uma versão da plataforma da AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

Observações

- As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). As ramificações de plataforma AL2023/AL2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem definições de configuração diferentes.
- Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Namespaces de configurações em Java SE: Amazon Linux AMI (AL1)

Você pode usar um [arquivo de configuração](#) para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. As opções de configuração podem ser definidas pelo serviço Elastic Beanstalk ou pela plataforma utilizada, e são organizadas em namespaces.

A Plataforma Java SE oferece suporte a um namespace de configuração específico da Plataforma, além dos [namespaces compatíveis com todas as Plataformas](#). O namespace `aws:elasticbeanstalk:container:java:staticfiles` permite que você defina opções que mapeiam os caminhos em seu aplicativo Web para as pastas em seu pacote de origem do aplicativo que incluem o conteúdo estático.

Por exemplo, este trecho de [option_settings](#) define duas opções no namespace de arquivos estáticos. A primeira mapeia o caminho `/public` para uma pasta denominada `public`, e a segunda mapeia o caminho `/images` para uma pasta denominada `img`:

```
option_settings:
```

```
aws:elasticbeanstalk:container:java:staticfiles:
  /html: statichtml
  /images: staticimages
```

As pastas que você mapeia usando esse namespace devem ser pastas reais na raiz do seu pacote de origem. Você não pode mapear um caminho para uma pasta em um arquivo JAR.

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte [Opções de configuração](#) para obter mais informações.

Criar JARs no servidor com um Buildfile

Você pode criar arquivos de classe do aplicativo e JAR(s) nas instâncias do EC2 em seu ambiente chamando um comando de build de um arquivo Buildfile em seu pacote de origem.

Comandos no arquivo Buildfile são executados apenas uma vez e devem ser encerrados no momento da conclusão, enquanto os comandos no arquivo [Procfile](#) devem ser executados por toda a vida útil do aplicativo e serão reiniciados se forem encerrados. Para executar os JARs em seu aplicativo, use um Procfile.

Para obter detalhes sobre o posicionamento e a sintaxe de um Buildfile, expanda a seção Buildfile e Procfile em [the section called “Estender plataformas Linux”](#).

O exemplo acima de Buildfile executa o Apache Maven para criar um aplicativo web do código-fonte. Para ver um aplicativo de exemplo que usa esse recurso, consulte [Exemplos de aplicativos web Java](#).

Example Buildfile

```
build: mvn assembly:assembly -DdescriptorId=jar-with-dependencies
```

A plataforma Java SE inclui as seguintes ferramentas de build, que você pode invocar do script de build:

- javac: compilador Java
- ant: Apache Ant
- mvn: Apache Maven
- gradle: Gradle

Configurar o processo do aplicativo com um Procfile

Se você tem mais de um arquivo JAR na raiz do seu pacote de origem da aplicação, deve incluir um arquivo `Procfile` que informa o Elastic Beanstalk quais JARs devem ser executados. Você também pode incluir um arquivo `Procfile` para um único aplicativo JAR para configurar a Java virtual machine (JVM – máquina virtual Java) que executa seu aplicativo.

Recomendamos sempre fornecer um `Procfile` no pacote de origem juntamente com o aplicativo. Dessa forma, você controla com precisão quais processos são executados pelo Elastic Beanstalk para o aplicativo e quais argumentos são recebidos por esses processos.

Para obter detalhes sobre como escrever e usar um `Procfile`, expanda a seção `Buildfile` e `Procfile` em [the section called “Estender plataformas Linux”](#).

Example Procfile

```
web: java -Xms256m -jar server.jar
cache: java -jar mycache.jar
web_foo: java -jar other.jar
```

O comando que executa o JAR principal em seu aplicativo deve ser denominado `web` e deve ser o primeiro comando listado no `Procfile`. O servidor `nginx` encaminha todas as solicitações HTTP que ele recebe do balanceador de carga do seu ambiente para essa aplicação.

O Elastic Beanstalk pressupõe que todas as entradas no `Procfile` devam ser sempre executadas e reinicia automaticamente qualquer aplicação encerrada definida no `Procfile`. Para executar comandos que serão encerrados e não deverão ser reiniciados, use um [Buildfile](#).

Usar um Procfile na AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o seu ambiente Java SE do Elastic Beanstalk usa uma versão da plataforma da AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

Observações

- As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). As ramificações de plataforma AL2023/AL2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem definições de configuração diferentes.

- Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Transferência de portas: Amazon Linux AMI (AL1)

Por padrão, o Elastic Beanstalk configura o proxy nginx para encaminhar solicitações à sua aplicação na porta 5000. Você pode substituir a porta padrão definindo a [propriedade do ambiente](#) PORT para a porta de escuta do seu aplicativo principal.

Se você usar o Procfile para executar várias aplicações, as versões da plataforma da AMI do Amazon Linux do Elastic Beanstalk esperam que cada aplicação adicional escute em uma porta 100 maior que a anterior. O Elastic Beanstalk define a variável PORT acessível de cada aplicação como a porta na qual ele espera que a aplicação seja executada. Você pode acessar essa variável no código do seu aplicativo chamando `System.getenv("PORT")`.

No exemplo acima de Procfile, o aplicativo web escuta na porta 5.000, cache escuta na porta 5.100 e web_foo escuta na porta 5.200. web configura a porta de escuta lendo a variável PORT e acrescenta 100 a esse número para determinar em qual porta cache está escutando para que ele possa enviar solicitações a ela.

Configurar o proxy reverso

O Elastic Beanstalk usa o [nginx](#) como proxy reverso para mapear sua aplicação para o balanceador de carga do Elastic Load Balancing na porta 80. O Elastic Beanstalk oferece uma configuração nginx padrão que você pode estender ou substituir completamente por sua própria configuração.

Por padrão, o Elastic Beanstalk configura o proxy nginx para encaminhar solicitações à sua aplicação na porta 5000. Você pode substituir a porta padrão definindo a [propriedade do ambiente](#) PORT para a porta de escuta do seu aplicativo principal.

Note

A porta que seu aplicativo escuta não afeta a porta na qual o servidor nginx escuta para receber solicitações do balanceador de carga.

Configurar o servidor proxy na sua versão da plataforma

Todas as plataformas AL2023/AL2 oferecem suporte a um recurso de configuração de proxy uniforme. Para obter mais informações sobre a configuração do servidor proxy nas versões de plataforma que executam o AL2023/AL2, expanda a seção Configuração do proxy reverso em [the section called “Estender plataformas Linux”](#).

Configurar o proxy da AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o seu ambiente Java SE do Elastic Beanstalk usa uma versão da plataforma da AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

Observações

- As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). As ramificações de plataforma AL2023/AL2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem definições de configuração diferentes.
- Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Estender e substituir a configuração padrão do proxy: Amazon Linux AMI (AL1)

Para estender a configuração padrão do nginx do Elastic Beanstalk, adicione arquivos de configuração `.conf` a uma pasta denominada `.ebextensions/nginx/conf.d/` no pacote de origem da sua aplicação. A configuração nginx do Elastic Beanstalk inclui os arquivos `.conf` nessa pasta automaticamente.

```
~/workspace/my-app/  
|-- .ebextensions  
|   |-- nginx  
|       |-- conf.d  
|           |-- myconf.conf  
|-- web.jar
```

Para substituir completamente a configuração nginx padrão do Elastic Beanstalk, inclua uma configuração em seu pacote de origem em `.ebextensions/nginx/nginx.conf`:

```
~/workspace/my-app/  
|-- .ebextensions  
|   |-- nginx  
|       |-- nginx.conf  
|-- web.jar
```

Se você substituir a configuração nginx do Elastic Beanstalk, adicione a linha a seguir ao arquivo `nginx.conf` para extrair as configurações do Elastic Beanstalk para [Monitoramento e relatório de integridade aprimorada](#), mapeamentos automáticos de aplicações e arquivos estáticos.

```
include conf.d/elasticbeanstalk/*.conf;
```

A configuração demonstrativa a seguir da [aplicação de exemplo Scorekeep](#) substitui a configuração padrão do Elastic Beanstalk para atender a uma aplicação Web estática do subdiretório `public` de `/var/app/current`, em que a plataforma Java SE copia o código-fonte da aplicação. O local `/api` encaminha o tráfego para rotas sob `/api/` para o aplicativo Spring que escuta na porta 5000. O restante do tráfego é atendido pelo aplicativo web no caminho raiz.

Example

```
user                nginx;  
error_log           /var/log/nginx/error.log warn;  
pid                 /var/run/nginx.pid;  
worker_processes   auto;  
worker_rlimit_nofile 33282;  
  
events {  
    worker_connections 1024;  
}  
  
http {  
    include          /etc/nginx/mime.types;  
    default_type     application/octet-stream;  
  
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '  
                   '$status $body_bytes_sent "$http_referer" '  
                   '"$http_user_agent" "$http_x_forwarded_for";
```

```
include      conf.d/*.conf;

map $http_upgrade $connection_upgrade {
    default    "upgrade";
}

server {
    listen      80 default_server;
    root /var/app/current/public;

    location / {
    }git pull

    location /api {
        proxy_pass      http://127.0.0.1:5000;
        proxy_http_version 1.1;

        proxy_set_header    Connection      $connection_upgrade;
        proxy_set_header    Upgrade         $http_upgrade;
        proxy_set_header    Host            $host;
        proxy_set_header    X-Real-IP       $remote_addr;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    access_log    /var/log/nginx/access.log main;

    client_header_timeout 60;
    client_body_timeout   60;
    keepalive_timeout    60;
    gzip                  off;
    gzip_comp_level       4;

    # Include the Elastic Beanstalk generated locations
    include conf.d/elasticbeanstalk/01_static.conf;
    include conf.d/elasticbeanstalk/healthd.conf;
}
}
```

Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente de aplicações Java

É possível usar uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS) para armazenar dados coletados e modificados pela aplicação. O banco de dados pode ser anexado ao seu ambiente e gerenciado pelo Elastic Beanstalk ou criado e gerenciado externamente.

Se você estiver usando o Amazon RDS pela primeira vez, adicione uma instância de banco de dados a um ambiente de teste utilizando o console do Elastic Beanstalk e verifique se a aplicação pode se conectar a ele.

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização do ambiente for concluída, o nome de host da instância de banco de dados e outras informações de conexão estarão disponíveis para o seu aplicativo por meio das seguintes propriedades de ambiente:

Nome da propriedade	Descrição	Valor da propriedade
RDS_HOSTNAME	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.

Nome da propriedade	Descrição	Valor da propriedade
RDS_PORT	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
RDS_DB_NAME	O nome do banco de dado, ebdb .	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
RDS_USERNAME	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Para obter mais informações sobre como configurar uma instância de banco de dados interna, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#). Para obter instruções sobre como configurar um banco de dados externo para uso com o Elastic Beanstalk, consulte [Usar o Elastic Beanstalk com o Amazon RDS](#).

Para se conectar ao banco de dados, adicione o arquivo JAR do driver adequado à sua aplicação, carregue a classe do driver no seu código e crie um objeto de conexão com as propriedades de ambiente fornecidas pelo Elastic Beanstalk.

Seções

- [Fazer download do driver do JDBC](#)
- [Conectar a um banco de dados \(plataformas Java SE\)](#)
- [Conectar a um banco de dados \(plataformas Tomcat\)](#)
- [Solução de problemas de conexões do banco de dados](#)

Fazer download do driver do JDBC

Você precisará do arquivo JAR do driver JDBC para o mecanismo de banco de dados escolhido. Salve o arquivo JAR em seu código-fonte e o inclua em seu classpath quando você compilar a classe que cria conexões com o banco de dados.

Você pode encontrar o driver mais recente para o seu mecanismo de banco de dados nos seguintes locais:

- MySQL: [MySQL Connector/J](#)
- Oracle SE-1: [Oracle JDBC Driver](#)
- Postgres: [PostgreSQL JDBC Driver](#)
- SQL Server: [Microsoft JDBC Driver](#)

Para usar o driver JDBC, chame `Class.forName()` para carregá-lo antes de criar a conexão com `DriverManager.getConnection()` em seu código.

O JDBC usa uma string de conexão no seguinte formato:

```
jdbc:driver://hostname:port/dbName?user=userName&password=password
```

É possível recuperar o nome do host, a porta, o nome do banco de dados, o nome de usuário e a senha das variáveis de ambiente que o Elastic Beanstalk fornece à aplicação. O nome do driver é específico do tipo de banco de dados e da versão do driver. Os exemplos a seguir são de nomes de driver:

- `mysql` para MySQL
- `postgresql` para PostgreSQL
- `oracle:thin` para Oracle Thin
- `oracle:oci` para Oracle OCI
- `oracle:oci8` para Oracle OCI 8
- `oracle:kprb` para Oracle KPRB
- `sqlserver` para SQL Server

Conectar a um banco de dados (plataformas Java SE)

Em um ambiente Java SE, use `System.getenv()` para ler as variáveis de conexão do ambiente. O exemplo de código a seguir mostra uma classe que cria uma conexão com um banco de dados PostgreSQL.

```
private static Connection getRemoteConnection() {
    if (System.getenv("RDS_HOSTNAME") != null) {
        try {
            Class.forName("org.postgresql.Driver");
            String dbName = System.getenv("RDS_DB_NAME");
            String userName = System.getenv("RDS_USERNAME");
            String password = System.getenv("RDS_PASSWORD");
            String hostname = System.getenv("RDS_HOSTNAME");
            String port = System.getenv("RDS_PORT");
            String jdbcUrl = "jdbc:postgresql://" + hostname + ":" + port + "/" + dbName + "?
user=" + userName + "&password=" + password;
            logger.trace("Getting remote connection with connection string from environment
variables.");
            Connection con = DriverManager.getConnection(jdbcUrl);
            logger.info("Remote connection successful.");
            return con;
        }
        catch (ClassNotFoundException e) { logger.warn(e.toString());}
        catch (SQLException e) { logger.warn(e.toString());}
    }
    return null;
}
```

Conectar a um banco de dados (plataformas Tomcat)

Em um ambiente Tomcat, as propriedades do ambiente são fornecidas como propriedades do sistema que podem ser acessadas com `System.getProperty()`.

O exemplo de código a seguir mostra uma classe que cria uma conexão com um banco de dados PostgreSQL.

```
private static Connection getRemoteConnection() {
    if (System.getProperty("RDS_HOSTNAME") != null) {
        try {
            Class.forName("org.postgresql.Driver");
            String dbName = System.getProperty("RDS_DB_NAME");
```

```
String userName = System.getProperty("RDS_USERNAME");
String password = System.getProperty("RDS_PASSWORD");
String hostname = System.getProperty("RDS_HOSTNAME");
String port = System.getProperty("RDS_PORT");
String jdbcUrl = "jdbc:postgresql://" + hostname + ":" + port + "/" + dbName + "?
user=" + userName + "&password=" + password;
    logger.trace("Getting remote connection with connection string from environment
variables.");
    Connection con = DriverManager.getConnection(jdbcUrl);
    logger.info("Remote connection successful.");
    return con;
}
catch (ClassNotFoundException e) { logger.warn(e.toString());}
catch (SQLException e) { logger.warn(e.toString());}
}
return null;
}
```

Se você tiver problemas para obter uma conexão ou executar instruções SQL, coloque o código a seguir em um arquivo JSP. Esse código conecta a uma instância de banco de dados, cria uma tabela e grava nela.

```
<%@ page import="java.sql.*" %>
<%
// Read RDS connection information from the environment
String dbName = System.getProperty("RDS_DB_NAME");
String userName = System.getProperty("RDS_USERNAME");
String password = System.getProperty("RDS_PASSWORD");
String hostname = System.getProperty("RDS_HOSTNAME");
String port = System.getProperty("RDS_PORT");
String jdbcUrl = "jdbc:mysql://" + hostname + ":" +
    port + "/" + dbName + "?user=" + userName + "&password=" + password;

// Load the JDBC driver
try {
    System.out.println("Loading driver...");
    Class.forName("com.mysql.jdbc.Driver");
    System.out.println("Driver loaded!");
} catch (ClassNotFoundException e) {
    throw new RuntimeException("Cannot find the driver in the classpath!", e);
}

Connection conn = null;
```

```
Statement setupStatement = null;
Statement readStatement = null;
ResultSet resultSet = null;
String results = "";
int numresults = 0;
String statement = null;

try {
    // Create connection to RDS DB instance
    conn = DriverManager.getConnection(jdbcUrl);

    // Create a table and write two rows
    setupStatement = conn.createStatement();
    String createTable = "CREATE TABLE Beanstalk (Resource char(50));";
    String insertRow1 = "INSERT INTO Beanstalk (Resource) VALUES ('EC2 Instance');";
    String insertRow2 = "INSERT INTO Beanstalk (Resource) VALUES ('RDS Instance');";

    setupStatement.addBatch(createTable);
    setupStatement.addBatch(insertRow1);
    setupStatement.addBatch(insertRow2);
    setupStatement.executeBatch();
    setupStatement.close();

} catch (SQLException ex) {
    // Handle any errors
    System.out.println("SQLException: " + ex.getMessage());
    System.out.println("SQLState: " + ex.getSQLState());
    System.out.println("VendorError: " + ex.getErrorCode());
} finally {
    System.out.println("Closing the connection.");
    if (conn != null) try { conn.close(); } catch (SQLException ignore) {}
}

try {
    conn = DriverManager.getConnection(jdbcUrl);

    readStatement = conn.createStatement();
    resultSet = readStatement.executeQuery("SELECT Resource FROM Beanstalk;");

    resultSet.first();
    results = resultSet.getString("Resource");
    resultSet.next();
    results += ", " + resultSet.getString("Resource");
```

```
resultSet.close();
readStatement.close();
conn.close();

} catch (SQLException ex) {
    // Handle any errors
    System.out.println("SQLException: " + ex.getMessage());
    System.out.println("SQLState: " + ex.getSQLState());
    System.out.println("VendorError: " + ex.getErrorCode());
} finally {
    System.out.println("Closing the connection.");
    if (conn != null) try { conn.close(); } catch (SQLException ignore) {}
}
%>
```

Para exibir os resultados, coloque o seguinte código no corpo da parte HTML do arquivo JSP.

```
<p>Established connection to RDS. Read first two rows: <%= results %></p>
```

Solução de problemas de conexões do banco de dados

Se você tiver problemas para se conectar a um banco de dados a partir do seu aplicativo, revise o banco de dados e o log do contêiner da web.

Revisar logs

É possível visualizar todos os logs do seu ambiente do Elastic Beanstalk a partir do Eclipse. Se você não tiver a visualização Explorer da AWS aberta, escolha a seta ao lado do ícone laranja da AWS na barra de ferramentas e, em seguida, escolha Mostrar visualização Explorer da AWS. Expanda AWS Elastic Beanstalk e o nome do seu ambiente, em seguida, abra o menu de contexto (clique com o botão direito do mouse) do servidor. Escolha Open in WTP Server Editor.

Escolha a guia Log da visualização Server para ver os logs agregados do ambiente. Para abrir os logs mais recentes, escolha o botão Refresh no canto superior direito da página.

Role para baixo para localizar os logs do Tomcat em `/var/log/tomcat7/catalina.out`. Se você carregou a página da web do exemplo anterior várias vezes, talvez veja o seguinte.

```
-----
/var/log/tomcat7/catalina.out
```

```
-----  
INFO: Server startup in 9285 ms  
Loading driver...  
Driver loaded!  
SQLException: Table 'Beanstalk' already exists  
SQLState: 42S01  
VendorError: 1050  
Closing the connection.  
Closing the connection.
```

Todas as informações que o aplicativo web envia para a saída padrão aparecem no log do contêiner da web. No exemplo anterior, o aplicativo tenta criar a tabela sempre que a página carrega. Isso resulta na detecção de uma exceção SQL em cada carregamento de página após o primeiro.

Como exemplo, o antecedente é aceitável. Mas em aplicativos reais, mantenha suas definições de banco de dados em objetos de esquema, execute transações de dentro de classes do modelo e coordene solicitações com servlets de controlador.

Conectar a uma instância de banco de dados do RDS

É possível se conectar diretamente à instância de banco de dados do RDS no seu ambiente do Elastic Beanstalk usando a aplicação cliente MySQL.

Primeiro, abra o security group de sua instância de banco de dados do RDS para permitir o tráfego de seu computador.

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Ao lado de Endpoint, escolha o link do console do Amazon RDS.
6. Na página de detalhes da instância RDS Dashboard, em Security and Network, escolha o security group que começa com rds- ao lado de Security Groups.

Note

O banco de dados pode ter várias entradas com o rótulo Security Groups. Use o primeiro, que começa com awseb, somente se você tiver uma conta antiga que não tenha uma [Amazon Virtual Private Cloud](#) (Amazon VPC) padrão.

7. Em Security group details, escolha a guia Inbound e, em seguida, Edit.
8. Adicione uma regra para MySQL (porta 3306) que permita o tráfego de seu endereço IP, especificado no formato CIDR.
9. Escolha Save (Salvar). As alterações entram em vigor imediatamente.

Retorne aos detalhes de configuração do Elastic Beanstalk do seu ambiente e observe o endpoint. Você usará o nome de domínio para se conectar à instância de banco de dados do RDS.

Instale o cliente MySQL e inicie uma conexão com o banco de dados na porta 3306. No Windows, instale o MySQL Workbench a partir da página inicial do MySQL e siga as instruções.

No Linux, instale o cliente MySQL usando o gerenciador de pacote para sua distribuição. O exemplo a seguir funciona no Ubuntu e em outros derivados de Debian.

```
// Install MySQL client
$ sudo apt-get install mysql-client-5.5
...
// Connect to database
$ mysql -h aas839jo2vwhwb.cnubrrfwfka8.us-west-2.rds.amazonaws.com -u username -  
ppassword ebdb
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 117
Server version: 5.5.40-log Source distribution
...
```

Depois de conectado, execute comandos SQL para ver o status do banco de dados, verificar se suas tabelas e linhas foram criadas e outras informações.

```
mysql> SELECT Resource from Beanstalk;
```

```
+-----+
| Resource |
+-----+
| EC2 Instance |
| RDS Instance |
+-----+
2 rows in set (0.01 sec)
```

Uso do AWS Toolkit for Eclipse

O AWS Toolkit for Eclipse integra os recursos de gerenciamento do AWS Elastic Beanstalk ao ambiente de desenvolvimento Tomcat para facilitar a criação, a configuração e implantação de código do ambiente. O toolkit inclui suporte a várias contas da AWS, gerenciamento de ambientes existentes e conexão direta com as instâncias do ambiente para solucionar problemas.

Note

O AWS Toolkit for Eclipse só suporta projetos que usam a plataforma Java com Tomcat, não a plataforma Java SE.

Para obter mais informações sobre pré-requisitos e instalação do AWS Toolkit for Eclipse, acesse <https://aws.amazon.com/eclipse>. Confira também o vídeo [Uso do AWS Elastic Beanstalk com o AWS Toolkit for Eclipse](#). Esse tópico também fornece informações úteis sobre ferramentas, tópicos de instruções e recursos adicionais para desenvolvedores Java.

Importar ambientes existentes no Eclipse

Importe ambientes existentes que você criou no Console de Gerenciamento da AWS para o Eclipse.

Para importar ambientes existentes, expanda o nó do AWS Elastic Beanstalk e clique duas vezes em um ambiente no AWS Explorer dentro do Eclipse. Agora é possível implantar suas aplicações Elastic Beanstalk neste ambiente.

Gerenciar ambientes de aplicações Elastic Beanstalk

Tópicos

- [Alterar definições de configuração do ambiente](#)
- [Alterar tipo de ambiente](#)

- [Configurar instâncias de servidor EC2 com uso do AWS Toolkit for Eclipse](#)
- [Configurar o Elastic Load Balancing com uso do AWS Toolkit for Eclipse](#)
- [Configuração de Auto Scaling com uso do AWS Toolkit for Eclipse](#)
- [Configurar notificações com uso do AWS Toolkit for Eclipse](#)
- [Configurar contêineres Java com uso do AWS Toolkit for Eclipse](#)
- [Configurar propriedades do sistema com o AWS Toolkit for Eclipse](#)

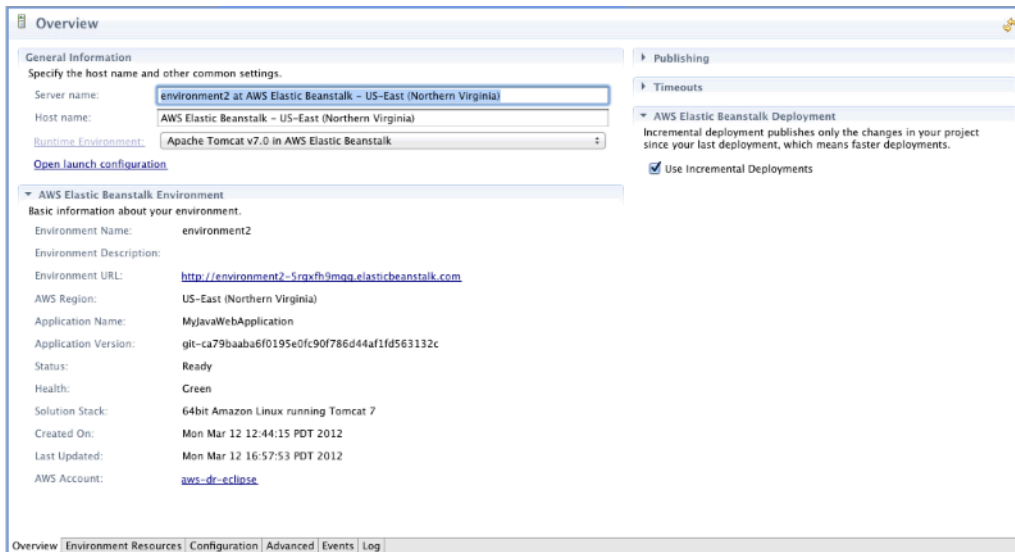
Com o AWS Toolkit for Eclipse, você pode alterar o provisionamento e a configuração dos recursos da AWS usados por seus ambientes de aplicações. Para obter informações sobre como gerenciar seus ambientes de aplicações usando o Console de Gerenciamento da AWS, consulte [Gerenciar ambientes](#). Esta seção discute as configurações de serviço específicas que você pode editar no AWS Toolkit for Eclipse como parte da configuração do ambiente de aplicativos. Para obter mais informações sobre AWS Toolkit for Eclipse, consulte o [Guia de conceitos básicos do AWS Toolkit for Eclipse](#).

Alterar definições de configuração do ambiente

Quando você implanta a aplicação, o Elastic Beanstalk configura uma série de serviços de computação na nuvem AWS. É possível controlar como esses serviços individuais são configurados usando o AWS Toolkit for Eclipse.

Para editar as configurações de ambiente de um aplicativo

1. Se o Eclipse não estiver exibindo o AWS Explorer, no menu, escolha Window (Janela), Show View (Mostrar visualização), AWS Explorer. Expanda o nó do Elastic Beanstalk e o nó da aplicação.
2. No AWS Explorer, clique duas vezes no ambiente do Elastic Beanstalk.
3. Na parte inferior do painel, clique na guia Configuration.

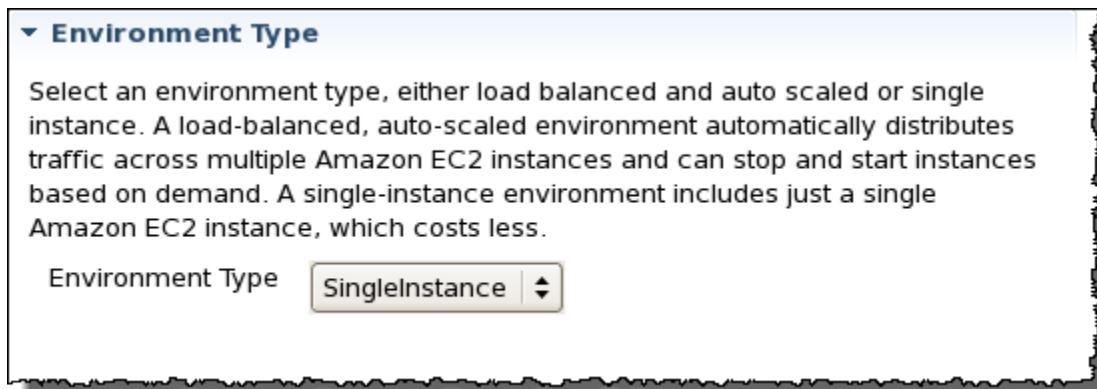


Agora, você pode definir as seguintes configurações:

- Instâncias de servidor EC2
- Load balancer
- Auto Scaling
- Notificações
- Tipos de ambiente
- Propriedades de ambiente

Alterar tipo de ambiente

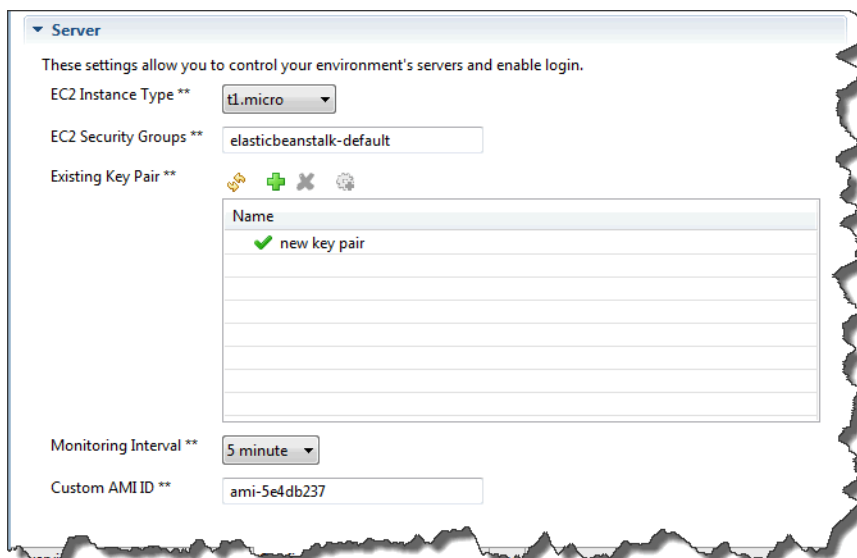
No AWS Toolkit for Eclipse, a seção Environment Type (Tipo do ambiente) da guia Configuration (Configuração) do ambiente permite selecionar Load balanced, auto scaled (Carga balanceada, escala automática) ou um ambiente de Single instance (Instância única) dependendo dos requisitos da aplicação implantada. Para um aplicativo que requer escalabilidade, selecione Load balanced, auto scaled. Para um aplicativo simples de baixo tráfego, selecione Single instance. Para obter mais informações, consulte [Tipos de ambiente](#).



Configurar instâncias de servidor EC2 com uso do AWS Toolkit for Eclipse

O Amazon Elastic Compute Cloud (EC2) é um serviço da Web para executar e gerenciar instâncias de servidor nos datacenters da Amazon. Você pode usar instâncias de servidor Amazon EC2 a qualquer momento, pelo tempo que precisar e por quaisquer motivos legais. As instâncias estão disponíveis em diferentes tamanhos e configurações. Para obter mais informações, consulte a [página do produto do Amazon EC2](#).

Em Server (Servidor), na guia Configuration (Configuração) do seu ambiente dentro do Toolkit for Eclipse, é possível editar a configuração da instância do Amazon EC2 do ambiente do Elastic Beanstalk.



Tipos de instância do Amazon EC2

Instance type (Tipo de instância) exibe os tipos de instância disponíveis para a aplicação do Elastic Beanstalk. Altere o tipo de instância para selecionar um servidor com as características (inclusive

o tamanho da memória e a capacidade de CPU) mais adequadas à aplicação. Por exemplo, os aplicativos com operações muito intensas e demoradas podem exigir mais CPU ou memória.

Para obter mais informações sobre os tipos de instância do Amazon EC2 disponíveis para a aplicação do Elastic Beanstalk, consulte [Instance Types \(Tipos de instância\)](#) no Guia do usuário do Amazon Elastic Compute Cloud.

Grupos de segurança do Amazon EC2

É possível controlar o acesso à sua aplicação do Elastic Beanstalk usando um grupo de segurança do Amazon EC2. Um grupo de segurança define regras de firewall para suas instâncias. Essas regras especificam o tráfego de rede de entrada que será entregue à sua instância. Qualquer outro tráfego de entrada será descartado. Você pode modificar as regras para um grupo a qualquer momento. As novas regras são aplicadas automaticamente para todas as instâncias em execução e as iniciadas no futuro.

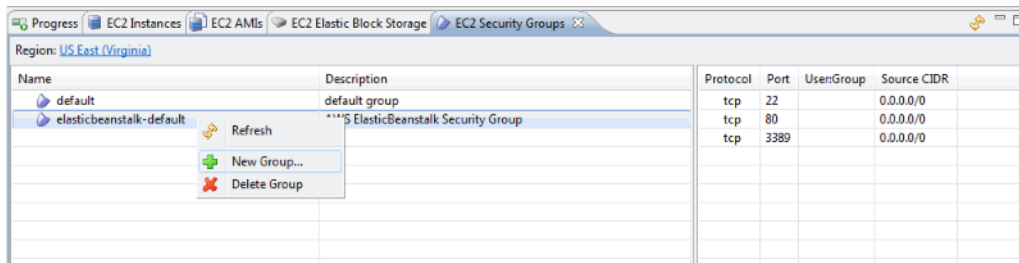
Configure os grupos de segurança do Amazon EC2 usando o Console de Gerenciamento da AWS ou o AWS Toolkit for Eclipse. Você pode especificar quais grupos de segurança do Amazon EC2 controlam o acesso à sua aplicação do Elastic Beanstalk inserindo os nomes de um ou mais grupos de segurança do Amazon EC2 (delimitados por vírgulas) na caixa EC2 Security Groups (Grupos de segurança do EC2).

Note

Se você estiver executando o aplicativo com um tipo de contêiner herdado, certifique-se de que a porta 80 (HTTP) possa ser acessada de 0.0.0.0/0 como o intervalo de CIDR de origem, caso você queira habilitar as verificações de integridade para o aplicativo. Para obter mais informações sobre as verificações de integridade, consulte [Verificações de integridade](#). Para verificar se você está usando um tipo de contêiner legado, consulte [the section called “Por que algumas versões de plataforma são marcadas como legadas?”](#)

Para criar um grupo de segurança usando o AWS Toolkit for Eclipse

1. No AWS Toolkit for Eclipse, clique na guia AWS Explorer. Expanda o nó Amazon EC2 e clique duas vezes em Security Groups (Grupos de segurança).
2. Clique com o botão direito do mouse em qualquer lugar da tabela à esquerda e clique em New Group.



3. Na caixa de diálogo Security Group, digite o nome e a descrição do security group e clique em OK.

Para obter mais informações sobre os grupos de segurança do Amazon EC2, consulte [Usar grupos de segurança](#) no Guia do usuário do Amazon Elastic Compute Cloud.

Pares de chave do Amazon EC2

É possível fazer login com segurança nas instâncias do Amazon EC2 provisionadas para a sua aplicação do Elastic Beanstalk com um par de chaves do Amazon EC2.

Important

É necessário criar um par de chaves do Amazon EC2 e configurar as instâncias do Amazon EC2 provisionadas pelo Elastic Beanstalk para usar o par de chaves do Amazon EC2 para poder acessar essas instâncias do Amazon EC2 provisionadas pelo Elastic Beanstalk. É possível criar o par de chaves usando o assistente Publish to Beanstalk (Publicar no Beanstalk) dentro do AWS Toolkit for Eclipse durante a implantação da aplicação no Elastic Beanstalk. Também há a opção de configurar os pares de chave do Amazon EC2 usando o [Console de Gerenciamento da AWS](#). Para obter instruções sobre como criar um par de chaves do Amazon EC2, consulte o [Guia de conceitos básicos do Amazon Elastic Compute Cloud](#).

Para obter mais informações sobre pares de chaves do Amazon EC2, acesse [Usar credenciais do Amazon EC2](#) no Guia do usuário do Amazon Elastic Compute Cloud. Para obter mais informações sobre a conexão com instâncias do Amazon EC2, consulte [Conectar a instâncias](#) e [Conectar a uma instância do Linux/UNIX do Windows usando PuTTY](#) no Guia do usuário do Amazon Elastic Compute Cloud.

Métricas do CloudWatch

Por padrão, somente as métricas básicas do Amazon CloudWatch estão habilitadas. Elas retornam dados em intervalos de cinco minutos. Você pode habilitar métricas mais detalhadas do CloudWatch, em intervalos de um minuto, selecionando 1 minute (1 minuto) para o Monitoring Interval (Intervalo de monitoramento) na seção Server (Servidor) da guia Configuration (Configuração) para o ambiente no AWS Toolkit for Eclipse.

Note

As cobranças de serviço do Amazon CloudWatch podem se aplicar a métricas de intervalo de um minuto. Consulte [Amazon CloudWatch](#) para obter mais informações.

ID da AMI personalizado

Você pode modificar a AMI padrão usada nas instâncias do Amazon EC2 pela própria AMI personalizada inserindo o identificador da AMI personalizada na caixa Custom AMI ID (ID da AMI personalizada) na seção Server (Servidor) da guia Configuration (Configuração) para o ambiente no AWS Toolkit for Eclipse.

Important

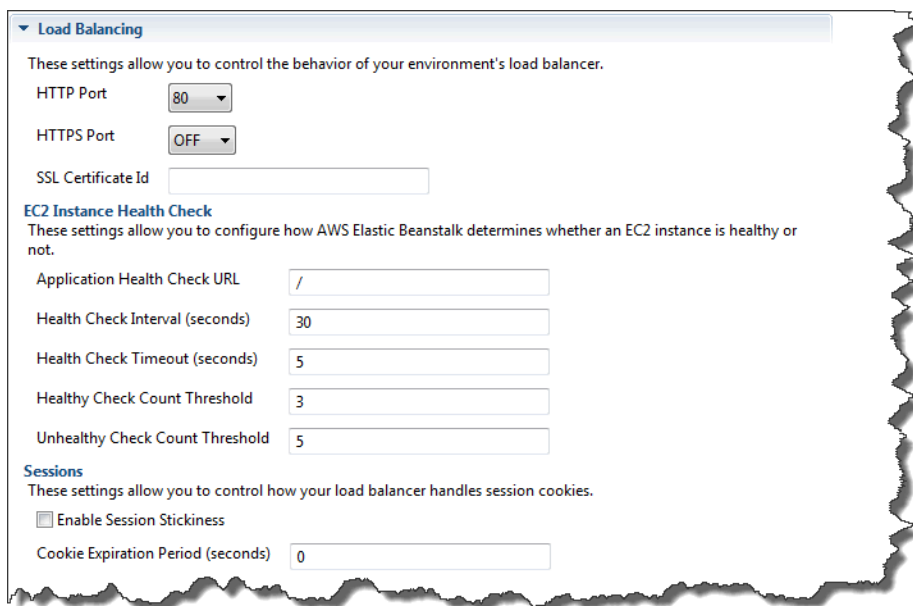
O uso da AMI própria é uma tarefa avançada que deve ser feita com cuidado. Se você precisar de uma AMI personalizada, recomendamos começar com a AMI do Elastic Beanstalk padrão e depois modificá-la. Para ser considerado íntegro, o Elastic Beanstalk espera que as instâncias do Amazon EC2 atendam a um conjunto de requisitos, inclusive ter um gerenciador de host em execução. Se esses requisitos não forem atendidos, talvez o ambiente não funcione corretamente.

Configurar o Elastic Load Balancing com uso do AWS Toolkit for Eclipse

O Elastic Load Balancing é um serviço Web da Amazon que melhora a disponibilidade e a escalabilidade de seu aplicativo. Com o Elastic Load Balancing, é possível distribuir cargas de aplicação entre duas ou mais instâncias do Amazon EC2. O Elastic Load Balancing melhora a disponibilidade por meio de redundância e acomoda o crescimento de tráfego do aplicativo.

O Elastic Load Balancing distribui e balanceia automaticamente o tráfego de entrada de aplicativo entre todas as instâncias de servidor EC2 que você está executando. O serviço também facilita a adição de novas instâncias, caso você precise aumentar a capacidade do aplicativo.

O Elastic Beanstalk provisiona automaticamente o Elastic Load Balancing ao implantar uma aplicação. Em Load Balancing (Balanceamento de carga), na guia Configuration (Configuração) do seu ambiente dentro do Toolkit for Eclipse, é possível editar a configuração de balanceamento de carga do ambiente do Elastic Beanstalk.



Load Balancing

These settings allow you to control the behavior of your environment's load balancer.

HTTP Port: 80

HTTPS Port: OFF

SSL Certificate Id: [Empty]

EC2 Instance Health Check

These settings allow you to configure how AWS Elastic Beanstalk determines whether an EC2 instance is healthy or not.

Application Health Check URL: /

Health Check Interval (seconds): 30

Health Check Timeout (seconds): 5

Healthy Check Count Threshold: 3

Unhealthy Check Count Threshold: 5

Sessions

These settings allow you to control how your load balancer handles session cookies.

Enable Session Stickiness

Cookie Expiration Period (seconds): 0

As seções a seguir descrevem os parâmetros do Elastic Load Balancing que você pode configurar para o seu aplicativo.

Portas

O balanceador de carga provisionado para lidar com as solicitações da aplicação do Elastic Beanstalk envia solicitações para as instâncias do Amazon EC2 que estão executando a aplicação. O balanceador de carga provisionado pode receber solicitações em portas HTTP e HTTPS e encaminhá-las para as instâncias do Amazon EC2 em sua aplicação do AWS Elastic Beanstalk. Por padrão, o load balancer lida com solicitações na porta HTTP. Pelo menos uma das portas (HTTP ou HTTPS) deve estar ativada.



These settings allow you to control the behavior of your environment's load balancer.

HTTP Port: 80

HTTPS Port: 443

SSL Certificate Id: arn:aws:iam::123456789012:/server-ce

EC2 Instance Health Check

⚠ Important

Certifique-se de que a porta especificada não esteja bloqueada, caso contrário, os usuários não poderão se conectar à aplicação do Elastic Beanstalk.

Controle da porta HTTP

Para desativar a porta HTTP, selecione OFF para HTTP Listener Port. Para ativar a porta HTTP, selecione uma porta HTTP (por exemplo, 80).

ℹ Note

Para acessar o ambiente usando uma porta diferente da porta 80 padrão, como a porta 8080, adicione uma escuta ao load balancer existente e configure a nova escuta nessa porta. Por exemplo, usando a [AWS CLI for Classic Load Balancers](#), digite o comando a seguir, substituindo *LOAD_BALANCER_NAME* pelo nome do balanceador de carga do Elastic Beanstalk.

```
aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
--listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
InstancePort=80"
```

Por exemplo, usando a [AWS CLI para Application Load Balancers](#), digite o comando a seguir, substituindo *LOAD_BALANCER_ARN* pelo ARN do balanceador de carga do Elastic Beanstalk.

```
aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP
--port 8080
```

Se você quiser que o Elastic Beanstalk monitore o ambiente, não remova a escuta na porta 80.

Controle da porta HTTPS

O Elastic Load Balancing oferece suporte ao protocolo HTTPS/TLS para permitir a criptografia do tráfego para conexões de clientes com o load balancer. As conexões do load balancer com as instâncias do EC2 são feitas usando texto simples. Por padrão, a porta HTTPS é desativada.

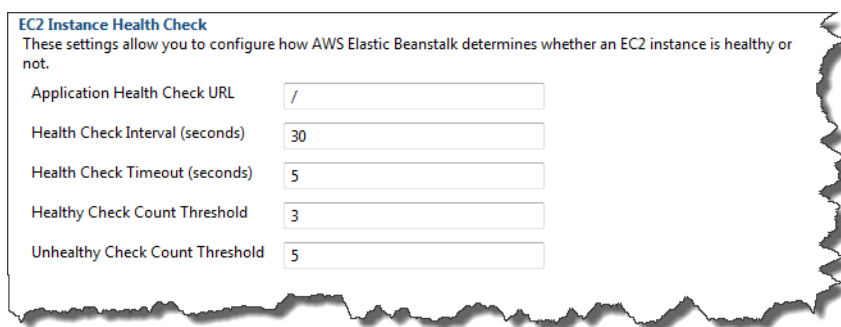
Para ativar a porta HTTPS

1. Crie um novo certificado usando o AWS Certificate Manager (ACM), ou faça upload de um certificado e uma chave no AWS Identity and Access Management (IAM). Para obter mais informações sobre como solicitar um certificado do ACM, consulte [Solicitação de um certificado](#) no Manual do usuário do AWS Certificate Manager. Para obter mais informações sobre a importação de certificados de terceiros para o ACM, consulte [Importação de certificados](#) no Manual do usuário do AWS Certificate Manager. Se o ACM [não estiver disponível em sua região da AWS](#), use o AWS Identity and Access Management (IAM) para fazer upload de um certificado de terceiros. Os serviços ACM e IAM armazenarão o certificado e fornecerão um nome de recurso da Amazon (ARN) para o certificado SSL. Para obter mais informações sobre a criação e o upload de certificados para o IAM, consulte [Trabalhar com certificados de servidor](#) no Guia do usuário do IAM.
2. Especifique a porta HTTPS selecionando uma porta na lista suspensa HTTPS Listener Port.
3. Na caixa de texto SSL Certificate ID (ID de certificado SSL), insira o Nome de recurso da Amazon (ARN) do certificado SSL. Por exemplo, o `arn:aws:iam::123456789012:server-certificate/abc/certs/build` ou o `arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`. Use o certificado SSL que você criou e carregou na etapa 1.

Para desativar a porta HTTPS, selecione OFF em HTTPS Listener Port.

Verificações de integridade

Você pode controlar as configurações da verificação de integridade usando a seção EC2 Instance Health Check do painel Load Balancing.



EC2 Instance Health Check
These settings allow you to configure how AWS Elastic Beanstalk determines whether an EC2 instance is healthy or not.

Application Health Check URL	<input type="text" value="/"/>
Health Check Interval (seconds)	<input type="text" value="30"/>
Health Check Timeout (seconds)	<input type="text" value="5"/>
Healthy Check Count Threshold	<input type="text" value="3"/>
Unhealthy Check Count Threshold	<input type="text" value="5"/>

A lista a seguir descreve os parâmetros de verificação de integridade que você pode definir para o aplicativo.

- Para determinar a integridade da instância, o Elastic Beanstalk procura um código de resposta 200 em um URL consultado por ele. Por padrão, o Elastic Beanstalk verifica na TCP: 80 os contêineres não herdados e na HTTP: 80 os contêineres herdados. Você pode substituir para corresponder a um recurso existente no seu aplicativo (por exemplo, `/myapp/index.jsp`), inserindo-o na caixa Application Health Check URL. Se você substituir o URL padrão, o Elastic Beanstalk usará HTTP para consultar o recurso. Para verificar se você está usando um tipo de contêiner legado, consulte [the section called “Por que algumas versões de plataforma são marcadas como legadas?”](#)
- Para Health Check Interval (seconds) (Intervalo de verificação de integridade (segundos)), insira o número de segundos entre verificações de integridade das instâncias do Amazon EC2 da aplicação.
- Em Health Check Timeout, especifique o número de segundos que o Elastic Load Balancing deve aguardar por uma resposta antes de considerar que a instância não está respondendo.
- Use as caixas Healthy Check Count Threshold e Unhealthy Check Count Threshold, especifique o número de testes consecutivos de URL bem-sucedidos ou malsucedidos antes de o Elastic Load Balancing alterar o status de integridade da instância. Por exemplo, especificar 5 na caixa de texto Unhealthy Check Count Threshold significa que o URL deve retornar uma mensagem de erro ou exceder o tempo limite cinco vezes consecutivas para o Elastic Load Balancing considerar que a verificação de integridade "falhou".

Sessões

Por padrão, um load balancer roteia cada solicitação de forma independente para a instância de servidor com a menor carga. Por comparação, uma sticky session vincula a sessão do usuário a uma determinada instância do servidor, para que todas as solicitações desse usuário durante a sessão sejam enviadas para a mesma instância do servidor.

O Elastic Beanstalk usa cookies HTTP gerados pelo balanceador de carga quando as sticky sessions são permitidas em uma aplicação. O load balancer usa um cookie especial gerado pelo load balancer para rastrear cada solicitação na instância do aplicativo. Quando o load balancer receber uma solicitação, ele primeiro verificará se esse cookie está presente na solicitação. Se estiver, a solicitação será enviada para a instância do aplicativo especificada no cookie. Se não houver um cookie, o load balancer selecionará uma instância do aplicativo com base no algoritmo de balanceamento de carga existente. Um cookie será inserido na resposta para vincular as solicitações subsequentes do mesmo usuário para essa instância do aplicativo. A configuração da política define a expiração do cookie, que estabelece a validade de cada cookie.

Em Load Balancer da seção Sessions, especifique se deseja ou não que o load balancer do aplicativo permita a perdurabilidade da sessão e a duração de cada cookie.



Para obter mais informações sobre Elastic Load Balancing, consulte o [Guia do desenvolvedor de Elastic Load Balancing](#).

Configuração de Auto Scaling com uso do AWS Toolkit for Eclipse

O Amazon EC2 Auto Scaling é um serviço da Web da Amazon projetado para executar ou encerrar automaticamente instâncias do Amazon EC2 com base em triggers definidos pelo usuário. Os usuários podem configurar grupos de Auto Scaling e associar triggers a esses grupos para escalar automaticamente recursos de computação com base em métricas como uso de largura de banda ou utilização da CPU. O Amazon EC2 Auto Scaling funciona com o Amazon CloudWatch para recuperar métricas das instâncias do servidor que executam a sua aplicação.

O Amazon EC2 Auto Scaling permite que você pegue um grupo de instâncias do Amazon EC2 e defina vários parâmetros para que esse grupo aumente ou diminua automaticamente em número. O Amazon EC2 Auto Scaling pode adicionar ou remover instâncias do Amazon EC2 desse grupo para ajudar você a lidar com as alterações de tráfego em sua aplicação sem problemas.

O Amazon EC2 Auto Scaling também monitora a integridade de cada instância do Amazon EC2 que ele executa. Se alguma instância for encerrada inesperadamente, o Amazon EC2 Auto Scaling detectará o encerramento e executará uma instância de substituição. Esse recurso permite que você mantenha automaticamente um número desejado fixo de instâncias do Amazon EC2.

O Elastic Beanstalk provisiona o Amazon EC2 Auto Scaling para sua aplicação. Em Auto Scaling, na guia Configuration (Configuração) do seu ambiente dentro do Toolkit for Eclipse, é possível editar a configuração do Auto Scaling do ambiente do Elastic Beanstalk.

Auto Scaling

Auto-scaling automatically launches or terminates EC2 instances based on defined metrics and thresholds called triggers. Auto-scaling will also launch a new EC2 instance in the event of a failure. These settings allow you to control auto-scaling behavior.

Minimum Instance Count

Maximum Instance Count

Availability Zones

Scaling Cooldown Time (seconds)

Scaling Trigger

Trigger Measurement

Trigger Statistic

Unit of Measurement

Measurement Period (seconds)

Breach Duration (seconds)

Upper Threshold

Scale-up Increment

Lower Threshold

Scale-down Increment

As seções a seguir discutem como configurar os parâmetros de Auto Scaling do aplicativo.

Configuração de ativação

É possível editar a configuração de execução para controlar como sua aplicação do Elastic Beanstalk provisiona os recursos do Amazon EC2 Auto Scaling.

Use as configurações Minimum Instance Count (Contagem de instância mínima) e Maximum Instance Count (Contagem de instância máxima) para especificar o tamanho mínimo e máximo do grupo de Auto Scaling usado pela aplicação Elastic Beanstalk.

Minimum Instance Count

Maximum Instance Count

Availability Zones

Scaling Cooldown Time (seconds)

Note

Para manter um número fixo de instâncias do Amazon EC2, defina as caixas de texto Minimum Instance Count (Contagem de instância mínima) e Maximum Instance Count (Contagem de instância máxima) com o mesmo valor.

Em Availability Zones (Zonas de disponibilidade), especifique o número de zonas de disponibilidade nas quais suas instâncias do Amazon EC2 devem estar. É importante definir esse número se você deseja criar aplicativos tolerantes a falhas: se uma zona de disponibilidade ficar inativa, suas instâncias continuarão sendo executadas nas outras.

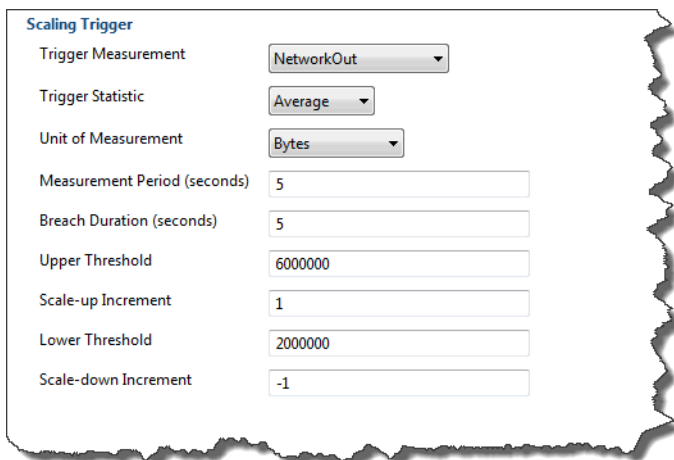
Note

No momento, não é possível especificar a zona de disponibilidade na qual sua instância estará.

Acionadores

Um trigger é um mecanismo de Amazon EC2 Auto Scaling definido para especificar ao sistema quando aumentar (expandir) e quando diminuir (reduzir) o número de instâncias. É possível configurar gatilhos para acionar quaisquer métricas publicadas no Amazon CloudWatch, como utilização de CPU, e determinar se as condições especificadas foram atendidas. Quando o limite superior ou inferior da métrica for excedido no período especificado, o gatilho iniciará um processo de longa duração chamado ação de escalabilidade.

É possível definir um acionador de escalabilidade para sua aplicação do Elastic Beanstalk usando o AWS Toolkit for Eclipse.



Field	Value
Trigger Measurement	NetworkOut
Trigger Statistic	Average
Unit of Measurement	Bytes
Measurement Period (seconds)	5
Breach Duration (seconds)	5
Upper Threshold	6000000
Scale-up Increment	1
Lower Threshold	2000000
Scale-down Increment	-1

Você pode configurar a lista de parâmetros do trigger a seguir na seção Scaling Trigger (Gatilho de escalabilidade) da guia Configuration (Configuração) para o seu ambiente dentro do Toolkit for Eclipse.

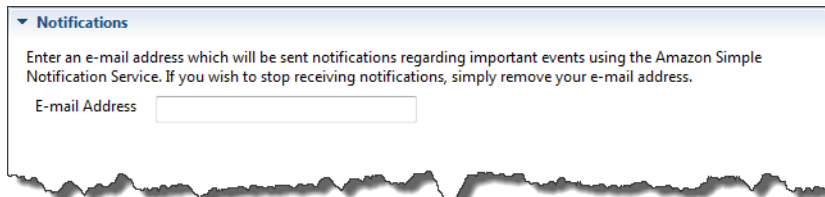
- Em Trigger Measurement, especifique a métrica para o gatilho.

- Em Trigger Statistic (Estatística do gatilho), especifique a estatística que o gatilho usará: **Minimum**, **Maximum**, **Sum** ou **Average**.
- Em Unit of Measurement, especifique as unidades de medida do gatilho.
- Em Measurement Period, especifique com que frequência o Amazon CloudWatch avalia as métricas do gatilho. Em Breach Duration, especifique o tempo que uma métrica pode ficar além do limite definido (conforme especificado em Upper Threshold e Lower Threshold) antes de o gatilho ser acionado.
- Em Scale-up Increment (Incremento de aumento) e Scale-down Increment (Incremento de redução), especifique quantas instâncias do Amazon EC2 devem ser adicionadas ou removidas ao realizar uma ação de escalabilidade.

Para obter mais informações sobre o Amazon EC2 Auto Scaling, consulte a seção Amazon EC2 Auto Scaling na [documentação do Amazon Elastic Compute Cloud](#).

Configurar notificações com uso do AWS Toolkit for Eclipse

O Elastic Beanstalk usa o Amazon Simple Notification Service (Amazon SNS) para notificar você de eventos importantes que afetam sua aplicação. Para habilitar as notificações do Amazon SNS, basta inserir seu endereço de e-mail na caixa de texto Email Address (Endereço de e-mail) em Notifications (Notificações) da guia Configuration (Configuração) do ambiente, dentro do Toolkit for Eclipse. Para desabilitar as notificações do Amazon SNS, remova seu endereço de e-mail da caixa de texto.



Configurar contêineres Java com uso do AWS Toolkit for Eclipse

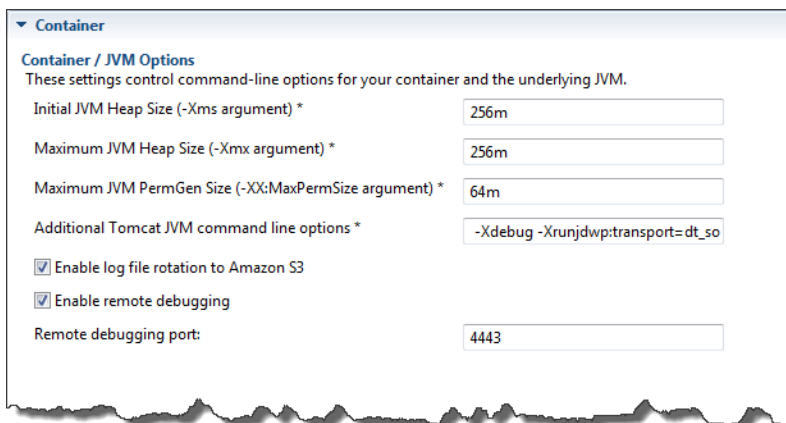
O painel Container/JVM Options (Opções de contêiner/JVM) permite ajustar o comportamento da máquina virtual Java nas instâncias do Amazon EC2 e habilitar ou desabilitar a alternância de logs do Amazon S3. Use o AWS Toolkit for Eclipse para configurar as informações do contêiner. Para obter mais informações sobre as opções disponíveis para ambientes Tomcat, consulte [the section called "Configurar o grupo do seu ambiente Tomcat"](#).

Note

É possível modificar as definições de configuração com tempo de inatividade zero, basta trocar o CNAME dos ambientes. Para obter mais informações, consulte [Implantações azuis/verdes com o Elastic Beanstalk](#).

Como acessar o painel de opções de contêiner/JVM para a aplicação do Elastic Beanstalk

1. Se o Eclipse não estiver exibindo o AWS Explorer, no menu, escolha Window (Janela), Show View (Mostrar visualização), AWS Explorer. Expanda o nó do Elastic Beanstalk e o nó da aplicação.
2. No AWS Explorer, clique duas vezes no ambiente do Elastic Beanstalk.
3. Na parte inferior do painel, clique na guia Configuration.
4. Em Container, configure as opções de contêiner.



Depuração remota

Para testar seu aplicativo remotamente, execute o aplicativo no modo de depuração.

Para habilitar a depuração remota

1. Selecione Enable remote debugging.
2. Em Remote debugging port, especifique o número da porta usada para a depuração remota.

A configuração Additional Tomcat JVM command line options é preenchida automaticamente.

Para iniciar a depuração remota

1. No menu do AWS Toolkit for Eclipse, escolha Window > Show View (Exibir visualização) > Other (Outros).
2. Expanda a pasta Server (Servidor) e escolha Servers (Servidores). Escolha OK.
3. No painel Servers, clique com o botão direito do mouse no servidor que executa o seu aplicativo e clique em Restart in Debug.

Configurar propriedades do sistema com o AWS Toolkit for Eclipse

O exemplo a seguir define a propriedade do sistema JDBC_CONNECTION_STRING no AWS Toolkit for Eclipse. Depois de definidas, essas propriedades ficam disponíveis para a sua aplicação do Elastic Beanstalk como propriedades do sistema chamadas JDBC_CONNECTION_STRING.

Note

O AWS Toolkit for Eclipse ainda não oferece suporte à modificação de configuração de ambientes, inclusive de propriedades do sistema, para ambientes em uma VPC. A menos que você tenha uma conta mais antiga usando o EC2 Classic, use o Console de Gerenciamento da AWS (descrito na próxima seção) ou a [EB CLI](#).

Note

As definições de configuração do ambiente podem incluir qualquer caractere ASCII imprimível, exceto o acento grave (` , ASCII 96), e não pode exceder 200 caracteres.

Como definir as propriedades do sistema para a sua aplicação do Elastic Beanstalk

1. Se o Eclipse não exibir o AWS Explorer, escolha Window (Janela), Show View (Mostrar visualização), Other (Outra). Expanda AWS Toolkit e escolha AWS Explorer.
2. No painel do AWS Explorer, expanda Elastic Beanstalk, expanda o nó da aplicação e clique duas vezes no ambiente do Elastic Beanstalk.
3. Na parte inferior do painel do seu ambiente, clique na guia Advanced.
4. Em aws:elasticbeanstalk:application:environment, clique em JDBC_CONNECTION_STRING e digite uma string de conexão. Por exemplo, a seguinte string de conexão JDBC se conecta a

uma instância de banco de dados MySQL na porta 3306 de localhost, com o nome de usuário me e a senha mypassword:

```
jdbc:mysql://localhost:3306/mydatabase?user=me&password=mypassword
```

Isso estará acessível à sua aplicação do Elastic Beanstalk como uma propriedade do sistema chamada JDBC_CONNECTION_STRING.

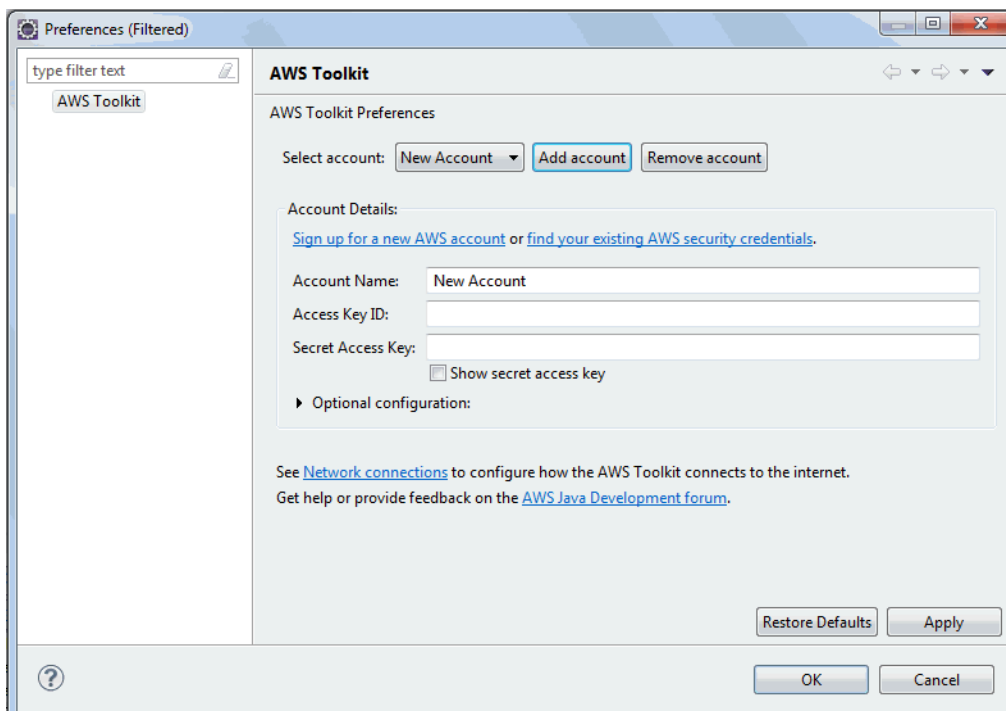
5. Pressione Ctrl+C no teclado ou selecione File (Arquivo), Save (Salvar) para salvar as alterações feitas na configuração do ambiente. As alterações são refletidas em aproximadamente um minuto.

Gerenciar várias contas da AWS

Você pode configurar diferentes contas da AWS para executar tarefas diferentes, como testes, preparação e produção. Use o AWS Toolkit for Eclipse para adicionar, editar e excluir contas facilmente.

Para adicionar uma conta AWS com o AWS Toolkit for Eclipse

1. No Eclipse, certifique-se de que a barra de ferramentas está visível. Na barra de ferramentas, clique na seta ao lado do ícone da AWS e selecione Preferences (Preferências).
2. Clique em Add account.



3. Na caixa de texto Nome da conta, digite o nome de exibição da conta.
4. Na caixa de texto Access Key ID (ID da chave de acesso), digite seu ID de chave de acesso da AWS.
5. Na caixa de texto Secret Access Key (Chave de acesso secreta), digite sua chave secreta da AWS.

Para acesso à API, é necessário ter um ID de chave de acesso e uma chave de acesso secreta. Use as chaves de acesso do usuário do IAM em vez das chaves de acesso da Usuário raiz da conta da AWS. Para obter mais informações sobre a criação de chaves de acesso, consulte [Gerenciar chaves de acesso para usuários do IAM](#) no Guia do usuário do IAM.

6. Clique em OK.

Como usar uma conta diferente para implantar uma aplicação no Elastic Beanstalk

1. Na barra de ferramentas do Eclipse, clique na seta ao lado do ícone da AWS e selecione Preferences (Preferências).
2. Para Default Account (Conta padrão), selecione a conta que você quer usar para implantar aplicações no Elastic Beanstalk.
3. Clique em OK.
4. No painel Project Explorer, clique com o botão direito do mouse na aplicação a ser implantada e selecione Amazon Web Services > Deploy to Elastic Beanstalk (Implantar no Elastic Beanstalk).

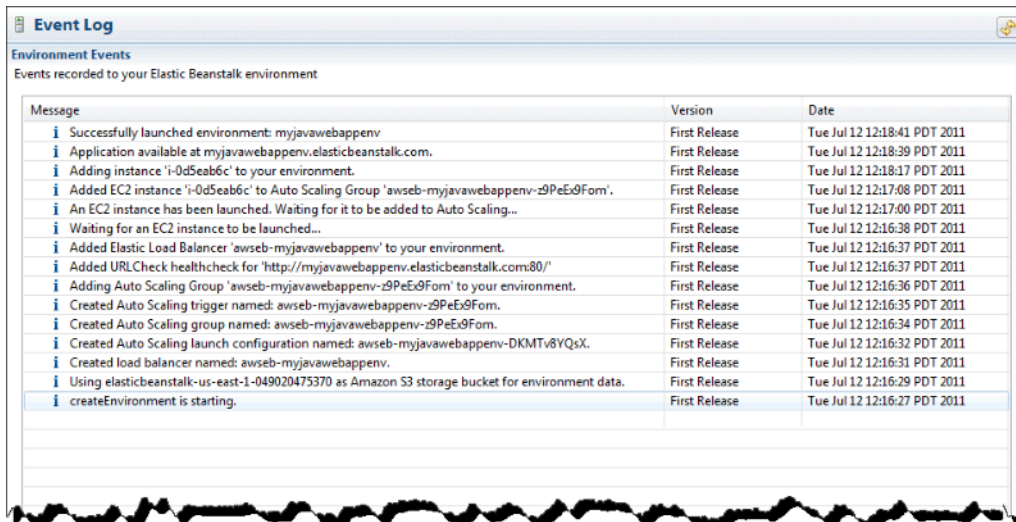
Visualizar eventos do

É possível usar o AWS Toolkit for Eclipse para acessar eventos e notificações associados a sua aplicação.

Para visualizar eventos do aplicativo

1. Se o Eclipse não estiver exibindo o AWS Explorer, no menu, clique em Window (Janela) > Show View (Mostrar visualização) > AWS Explorer. Expanda o nó do Elastic Beanstalk e o nó da aplicação.
2. No AWS Explorer, clique duas vezes no ambiente do Elastic Beanstalk.
3. Na parte inferior do painel, clique na guia Eventos.

É exibida uma lista de eventos de todos os ambientes do seu aplicativo.



Event Log
Environment Events
Events recorded to your Elastic Beanstalk environment

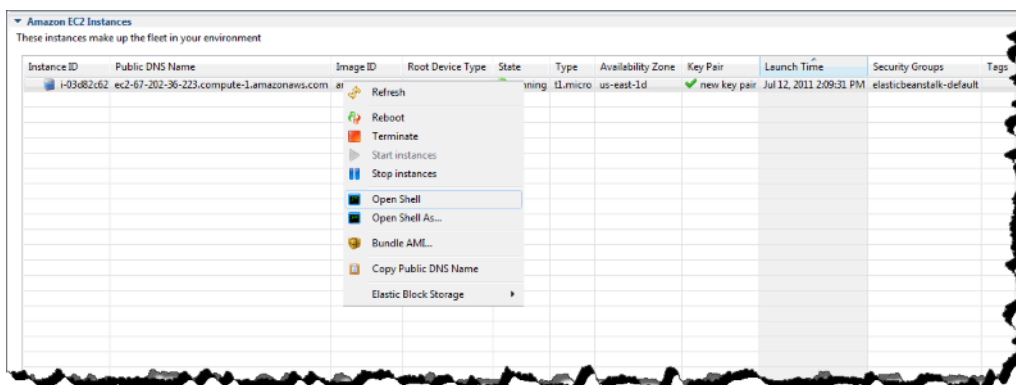
Message	Version	Date
Successfully launched environment: myjavawebappenv	First Release	Tue Jul 12 12:18:41 PDT 2011
Application available at myjavawebappenv.elasticbeanstalk.com.	First Release	Tue Jul 12 12:18:39 PDT 2011
Adding instance 'i-0d5eab6c' to your environment.	First Release	Tue Jul 12 12:18:17 PDT 2011
Added EC2 instance 'i-0d5eab6c' to Auto Scaling Group 'awsseb-myjavawebappenv-z9PeE9Fom'.	First Release	Tue Jul 12 12:17:08 PDT 2011
An EC2 instance has been launched. Waiting for it to be added to Auto Scaling...	First Release	Tue Jul 12 12:17:00 PDT 2011
Waiting for an EC2 instance to be launched...	First Release	Tue Jul 12 12:16:38 PDT 2011
Added Elastic Load Balancer 'awsseb-myjavawebappenv' to your environment.	First Release	Tue Jul 12 12:16:37 PDT 2011
Added URLCheck healthcheck for 'http://myjavawebappenv.elasticbeanstalk.com:80/'	First Release	Tue Jul 12 12:16:37 PDT 2011
Adding Auto Scaling Group 'awsseb-myjavawebappenv-z9PeE9Fom' to your environment.	First Release	Tue Jul 12 12:16:36 PDT 2011
Created Auto Scaling trigger named: awsseb-myjavawebappenv-z9PeE9Fom.	First Release	Tue Jul 12 12:16:35 PDT 2011
Created Auto Scaling group named: awsseb-myjavawebappenv-z9PeE9Fom.	First Release	Tue Jul 12 12:16:34 PDT 2011
Created Auto Scaling launch configuration named: awsseb-myjavawebappenv-DKMTv8YQsX.	First Release	Tue Jul 12 12:16:32 PDT 2011
Created load balancer named: awsseb-myjavawebappenv.	First Release	Tue Jul 12 12:16:31 PDT 2011
Using elasticbeanstalk-us-east-1-049020475370 as Amazon S3 storage bucket for environment data.	First Release	Tue Jul 12 12:16:29 PDT 2011
createEnvironment is starting.	First Release	Tue Jul 12 12:16:27 PDT 2011

Listar e conectar com instâncias de servidor

É possível visualizar uma lista de instâncias do Amazon EC2 executando o ambiente da aplicação do Elastic Beanstalk por meio da AWS Toolkit for Eclipse ou do Console de Gerenciamento da AWS. Você pode se conectar a essas instâncias usando o Secure Shell (SSH). Para obter informações sobre como listar e conectar-se a instâncias de servidor usando o Console de Gerenciamento da AWS, consulte [Listar e conectar com instâncias de servidor](#). A seção a seguir apresenta as etapas para você visualizar e se conectar às instâncias do servidor usando o AWS Toolkit for Eclipse.

Para visualizar e conectar-se a instâncias do Amazon EC2 em um ambiente

1. No AWS Toolkit for Eclipse, clique em AWSExplorer. Expanda o nó Amazon EC2 e clique duas vezes em Instances.
2. Na janela Amazon EC2 Instances, na coluna Instance ID, clique com o botão direito do mouse no ID da instância do Amazon EC2 em execução no load balancer do aplicativo. Em seguida, clique em Open Shell.



Amazon EC2 Instances
These instances make up the fleet in your environment

Instance ID	Public DNS Name	Image ID	Root Device Type	State	Type	Availability Zone	Key Pair	Launch Time	Security Groups	Tags
i-03d82c62	ec2-67-202-36-223.compute-1.amazonaws.com			running	t1.micro	us-east-1d	new key pair	Jul 12, 2011 2:09:31 PM	elasticbeanstalk-default	

Context menu actions: Refresh, Reboot, Terminate, Start instances, Stop instances, Open Shell, Open Shell As..., Bundle AMI..., Copy Public DNS Name, Elastic Block Storage

O Eclipse abre automaticamente o cliente SSH e faz a conexão com a instância do EC2.

Para obter mais informações sobre como conectar-se a uma instância do Amazon EC2, consulte o [Guia de conceitos básicos do Amazon Elastic Compute Cloud](#).

Encerrar um ambiente

Para evitar a cobrança de recursos da AWS não utilizados, termine um ambiente em execução usando o AWS Toolkit for Eclipse. Para obter detalhes sobre o encerramento do ambiente, consulte [Encerrar um ambiente do Elastic Beanstalk](#).

Para encerrar um ambiente


1. No AWS Toolkit for Eclipse, clique no painel AWS Explorer. Expanda o nó Elastic Beanstalk.
2. Expanda a aplicação do Elastic Beanstalk e clique com o botão direito do mouse no ambiente do Elastic Beanstalk.
3. Clique em Terminate Environment. Alguns minutos serão necessários para o Elastic Beanstalk terminar os recursos da AWS em execução no ambiente.

Recursos

Há vários lugares que você pode ir para obter ajuda adicional durante o desenvolvimento de seus aplicativos Java.

Recurso	Descrição
O fórum de desenvolvimento Java da AWS	Publique suas perguntas e receba comentários.
Centro do desenvolvedor Java	Um local central para acessar código de exemplo, documentação, ferramentas e outros recursos.

Trabalhar com .NET Core no Linux

 Confira o.NET no AWS Developer Center

Você já visitou nosso Net Developer Center? É nosso balcão único para tudo o que está disponível em.NET AWS.

Para obter mais informações, consulte [o.NET no AWS Developer Center](#).

Esta seção fornece informações sobre a implantação de aplicações .NET Core no Linux usando o AWS Elastic Beanstalk.

Os tópicos neste capítulo pressupõem que você tem um nível básico de conhecimento sobre ambientes do Elastic Beanstalk. Se você não tiver usado o Elastic Beanstalk antes, tente o [tutorial de conceitos básicos](#) para saber os princípios.

Tópicos

- [Conceitos básicos do .NET Core no Linux](#)
- [Configurar seu ambiente de desenvolvimento .NET Core no Linux](#)
- [Usar a plataforma .NET Core no Linux](#)
- [Tutorial: Implantar uma aplicação ASP.NET Core no Linux usando o Elastic Beanstalk](#)
- [O AWS Toolkit for Visual Studio: trabalhar com o .NET Core](#)
- [Migrar de plataformas .NET no Windows Server para a plataforma .NET Core no Linux](#)

Conceitos básicos do .NET Core no Linux

Para começar a usar as aplicações .NET Core do Linux no AWS Elastic Beanstalk, tudo que você precisa é um [pacote de origem](#) de aplicação para fazer upload como a primeira versão da aplicação e implantá-lo em um ambiente. Quando você cria um ambiente, o Elastic Beanstalk aloca todos os recursos da AWS necessários para executar uma aplicação Web altamente escalável.

Iniciar um ambiente com uma aplicação demonstrativa .NET Core no Linux

O Elastic Beanstalk fornece aplicações demonstrativas de página única para cada plataforma.

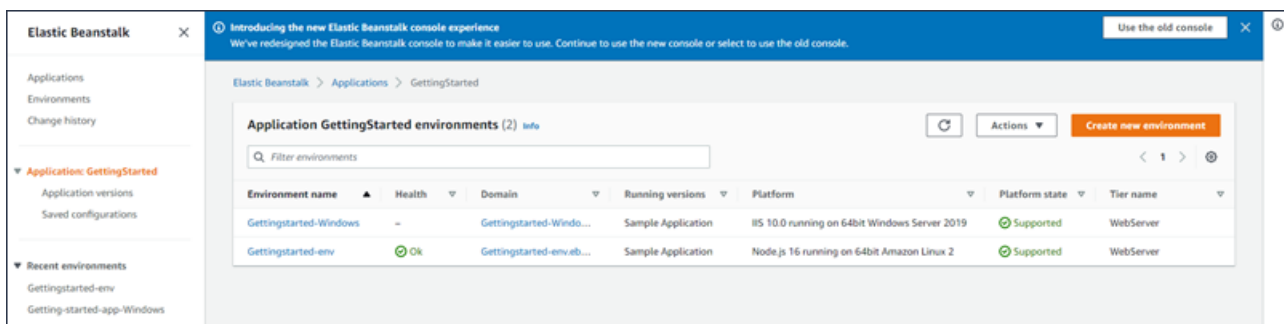
Amostras

Configurações compatíveis	Tipo de ambiente	Pacote de origem	Descrição
.NET Core no Linux	Servidor web	dotnet-core-linux.zip	Aplicativo de página única.
.NET Core no Linux	Servidor web	dotnet-core-linux-multiple-apps.zip	Dois aplicativos Web que são executados no mesmo servidor Web.

Faça download da aplicação demonstrativa e implante-a no Elastic Beanstalk seguindo estas etapas.

Para iniciar um ambiente com um aplicativo de exemplo (console)

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, escolha Aplicativos e o nome de um aplicativo existente na lista ou [crie um](#).
3. Na página de visão geral da aplicação, escolha Create a new environment (Criar um novo ambiente).



Isso inicia o assistente Create environment (Criar ambiente). O assistente fornece um conjunto de etapas para a criação de um novo ambiente.

Step 1
Configure environment

Step 2
Configure service access

Step 3 - optional
Configure instance traffic and scaling

Step 4 - optional
Set up networking, database, and tags

Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

Configure environment [Info](#)

Environment tier [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- Web server environment**
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
- Worker environment**
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information [Info](#)

Application name
GettingStarted
Maximum length of 100 characters.

▶ **Application tags (optional)**

Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name
GettingStarted-env
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain name
 .us-east-1.elasticbeanstalk.com

Environment description

Platform [Info](#)

Platform type

- Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)
- Custom platform**
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform


Platform branch

Platform version

Application code [Info](#)


- Sample application**
- Existing version**
Application versions that you have uploaded.
- Upload your code**
Upload a source bundle from your computer or copy one from Amazon S3.

4. Para nível de ambiente, escolha o [nível de ambiente](#) Web server environment (Ambiente do servidor Web) ou Worker environment (Ambiente do operador). Você não pode alterar o nível de um ambiente após a criação.

 Note

O [.NET na Plataforma de servidor Windows](#) não tem suporte para a camada de ambiente do operador.

5. Em Plataforma, selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.

 Note

O Elastic Beanstalk é compatível com várias [versões](#) para a maioria das plataformas listadas. Por padrão, o console seleciona a versão recomendada da plataforma e a ramificação da plataforma escolhida. Se a aplicação exigir outra versão, você poderá selecioná-la aqui. Para obter informações sobre versões de plataforma compatíveis, consulte [the section called “Plataformas compatíveis”](#).

6. Para Application code, escolha Sample application.
7. Em Configuration presets (Predefinições de configuração), escolha Single instance (Instância única).
8. Escolha Next (Próximo).
9. A página Configurar acesso ao serviço é exibida.

Configure service access [Info](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role
 Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

[View permission details](#)

Cancel Skip to review Previous **Next**

10. Escolha Usar um perfil de serviço existente em Perfil de serviço.

11. Em seguida, vamos nos concentrar na lista suspensa Perfil de instância do EC2. Os valores exibidos nessa lista suspensa podem variar, dependendo de a conta ter criado anteriormente um novo ambiente.

Escolha uma das opções a seguir com base nos valores exibidos na lista.

- Se `aws-elasticbeanstalk-ec2-role` for exibido na lista suspensa, selecione-o na lista suspensa Perfil de instância do EC2.
- Se outro valor for exibido na lista e for o perfil padrão de instância do EC2 para seus ambientes, selecione-o na lista suspensa Perfil de instância do EC2.
- Se a lista suspensa Perfil de instância do EC2 não mostrar nenhum valor para seleção, expanda o procedimento que se segue, Criar um perfil do IAM para ser o perfil de instância do EC2.

Conclua as etapas em Criar perfil do IAM para ser o perfil de instância do EC2 para criar um perfil do IAM que você possa selecionar posteriormente para ser o perfil de instância do EC2. Depois, retorne para esta etapa.

Agora que você criou um perfil do IAM e atualizou a lista, ele será exibido como uma opção na lista suspensa. Selecione o perfil do IAM que você acabou de criar na lista suspensa Perfil de instância do EC2.

12. Escolha Skip to Review (Pular para revisão) na página Configure service access (Configurar acesso a serviço).

Isso selecionará os valores padrão para essa etapa e pulará as etapas opcionais.

13. A página Review (Revisão) exibe um resumo de todas as suas escolhas.

Para personalizar ainda mais o ambiente, escolha Edit (Editar) ao lado da etapa que inclui os itens que você deseja configurar. Você pode configurar as opções a seguir somente durante a criação do ambiente:

- Nome do ambiente
- Nome de domínio
- Versão da plataforma
- Processor
- VPC
- Nível

Você pode alterar as configurações a seguir após a criação do ambiente, mas elas exigem que novas instâncias ou outros recursos sejam provisionados, e pode levar muito tempo para que elas sejam aplicadas:

- Tipo de instância, volume raiz, par de chaves e função do AWS Identity and Access Management (IAM)
- Banco de dados interno do Amazon RDS
- Load balancer

Para obter detalhes sobre todas as configurações disponíveis, consulte [O assistente de criação de novo ambiente](#).

14. Escolha Submit (enviar) na parte inferior da página para inicializar a criação do novo ambiente.

Criar um perfil do IAM para ser o perfil de instância do EC2

The screenshot shows the 'Configure service access' dialog box in the AWS IAM console. The title is 'Configure service access' with an 'Info' link. Below the title is a 'Service access' section with a brief description and a 'Learn more' link. The 'Service role' section has two radio buttons: 'Create and use new service role' (unselected) and 'Use an existing service role' (selected). Under 'Existing service roles', there is a text input field containing 'aws-elasticbeanstalk-service-role' and a refresh button. The 'EC2 key pair' section has a text input field with 'Choose a key pair' and a refresh button. The 'EC2 instance profile' section has a text input field with 'aws-elasticbeanstalk-ec2-role' and a refresh button. Below these fields is a 'View permission details' button. At the bottom of the dialog are 'Cancel', 'Skip to review', 'Previous', and 'Next' buttons.

Para criar um perfil do IAM para ser o perfil de instância do EC2


1. Escolha Visualizar detalhes de permissão. Essa opção exibe a lista suspensa Perfil de instância do EC2.

Uma janela modal intitulada Visualizar permissões de perfil de instância é exibida. Essa janela lista os perfis gerenciados que deverão ser anexados ao novo perfil de instância do EC2 que você criar. Ela também fornece um link para iniciar o console do IAM.

2. Escolha o link Console do IAM exibido na parte superior da janela.
3. No painel de navegação do console do IAM, escolha Roles (Perfis).
4. Selecione Create role (Criar função).
5. Em Tipo de entidade confiável, selecione Serviço da AWS.
6. Em Use case (Caso de uso), selecione EC2.
7. Escolha Next (Próximo).
8. Anexe as políticas gerenciadas apropriadas. Role a janela modal Visualizar permissões de perfil de instância para ver as políticas gerenciadas. As políticas também estão listadas aqui:

- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

9. Escolha Next (Próximo).
10. Insira um nome para a função.
11. (Opcional) Adicione tags à função.
12. Selecione Create role (Criar função).
13. Retorne à janela do console do Elastic Beanstalk que está aberta.
14. Feche a janela modal Visualizar permissões de perfil de instância.

 Important

Não feche a página do navegador que exibe o console do Elastic Beanstalk.

15. Escolha



(atualizar), ao lado da lista suspensa Perfil de instância do EC2.

Essa opção atualiza a lista suspensa para que o perfil que você acabou de criar seja exibido na lista suspensa.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo diferente a qualquer momento. A implantação de uma nova versão da aplicação é muito rápida, pois não é necessário provisionar nem reiniciar as instâncias do Amazon EC2.

Depois de implantar uma ou duas aplicações demonstrativas, quando você estiver pronto para começar a desenvolver e executar aplicações .NET Core localmente, consulte [Configurar seu ambiente de desenvolvimento .NET Core no Linux](#).

Configurar seu ambiente de desenvolvimento .NET Core no Linux

Configure um ambiente de desenvolvimento .NET Core para testar sua aplicação localmente antes de implantá-lo no AWS Elastic Beanstalk. Este tópico descreve as etapas de configuração do ambiente de desenvolvimento e inclui links para páginas de instalação de ferramentas úteis.

Para ver as etapas de configuração comuns e as ferramentas que se aplicam a todas as linguagens, consulte [Configurar sua máquina de desenvolvimento para uso com o Elastic Beanstalk](#).

Seções

- [Instalar o .NET Core SDK](#)
- [Instalar um IDE](#)
- [Instalar o AWS Toolkit for Visual Studio](#)

Instalar o .NET Core SDK

É possível usar o .NET Core SDK para desenvolver aplicativos executados no Linux.

Consulte a [página de downloads do .NET](#) para fazer download e instalar o .NET Core SDK.

Instalar um IDE

Os Integrated development environments (IDEs – Ambientes de desenvolvimento integrados) oferecem uma variedade de recursos que facilitam o desenvolvimento de aplicativos. Se você nunca usou um IDE para desenvolvimento .NET, experimente o Visual Studio Community para começar.

Visite a página [Visual Studio Community](#) para fazer download e instalar o Visual Studio Community.

Instalar o AWS Toolkit for Visual Studio

O [AWS Toolkit for Visual Studio](#) é um plugin de código aberto para o IDE do Visual Studio que facilita para os desenvolvedores desenvolverem, depurarem e implantarem aplicações .NET usando a AWS. Consulte a [página inicial do Toolkit for Visual Studio](#) para obter instruções de instalação.

Usar a plataforma .NET Core no Linux

A plataforma .NET no Linux do AWS Elastic Beanstalk é um conjunto de [versões de plataforma](#) para aplicações .NET Core que são executadas no sistema operacional Linux.

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte [the section called “Estender plataformas Linux”](#). Veja a seguir algumas considerações específicas da plataforma.

Introdução à plataforma .NET Core no Linux

Servidor de proxy

A plataforma .NET Core no Linux do Elastic Beanstalk inclui um proxy reverso que encaminha solicitações para a aplicação. Por padrão, o Elastic Beanstalk usa [nginx](#) como servidor de proxy. É possível optar por não usar nenhum servidor de proxy e configurar o [Kestrel](#) como o servidor Web. O Kestrel é incluído por padrão em modelos de projeto ASP.NET Core.

Estrutura do aplicativo

É possível publicar aplicações dependentes do runtime que usam o runtime do .NET Core fornecido pelo Elastic Beanstalk. Você também pode publicar aplicativos independentes que incluem o runtime do .NET Core e as dependências do aplicativo no pacote de origem. Para saber mais, consulte [the section called “Empacotar aplicativos”](#).

Configuração da plataforma

Para configurar os processos que são executados em instâncias do servidor em seu ambiente, inclua um [Procfile](#) opcional em seu pacote de origem. Um `Procfile` será necessário se você tiver mais de um aplicativo no pacote de origem.

Recomendamos sempre fornecer um `Procfile` no pacote de origem juntamente com o aplicativo. Dessa forma, você controla com precisão quais processos o Elastic Beanstalk executa para a aplicação.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para [modificar a configuração de um ambiente em execução](#). Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar [configurações salvas](#) para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir [arquivos de configuração](#). As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console. Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte [Opções de configuração](#).

Configurar o ambiente .NET Core no Linux

As configurações da plataforma .NET Core no Linux permitem ajustar o comportamento das instâncias do Amazon EC2. É possível editar a configuração da instância do Amazon EC2 do ambiente do Elastic Beanstalk usando o console do Elastic Beanstalk.

Você pode usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3 e configurar variáveis que sua aplicação pode ler no ambiente.

Como configurar o ambiente .NET Core no Linux usando o console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Opções de log

A seção Log Options (Opções de log) tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Enable log file rotation to Amazon S3 (Habilitar a rotação de arquivos de log para o Amazon S3): especifica se os arquivos de log das instâncias do Amazon EC2 da aplicação serão copiados para o bucket do Amazon S3 associado a ela.

Propriedades de ambiente

A seção Environment Properties (Propriedades do ambiente) permite especificar definições de configuração do ambiente nas instâncias do Amazon EC2 que estão executando a aplicação. As propriedades de ambiente são passadas para o aplicativo como pares de chave-valor.

No ambiente .NET Core no Linux em execução no Elastic Beanstalk, as variáveis de ambiente podem ser acessadas usando `Environment.GetEnvironmentVariable("variable-name")`. Por exemplo, você pode ler uma propriedade denominada `API_ENDPOINT` em uma variável com o seguinte código.

```
string endpoint = Environment.GetEnvironmentVariable("API_ENDPOINT");
```

Consulte [Propriedades de ambientes e outras configurações de software](#) para obter mais informações.

Namespace de configuração do .NET Core no Linux

Você pode usar um [arquivo de configuração](#) para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. As opções de configuração podem ser definidas pelo serviço Elastic Beanstalk ou pela plataforma utilizada, e são organizadas em namespaces.

A plataforma .NET Core no Linux é compatível com opções no namespace a seguir, além das [opções compatíveis com todos os ambientes do Elastic Beanstalk](#):

- `aws:elasticbeanstalk:environment:proxy`: escolha usar nginx ou nenhum servidor de proxy. Os valores válidos são `nginx` ou `none`.

O arquivo de configuração demonstrativo a seguir mostra o uso das opções de configuração específicas do .NET Core no Linux.

Example `.ebextensions/proxy-settings.config`

```
option_settings:  
  aws:elasticbeanstalk:environment:proxy:  
    ProxyServer: none
```

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console,

configurações salvas, a EB CLI ou a AWS CLI. Consulte [Opções de configuração](#) para obter mais informações.

Empacotar aplicações para a plataforma .NET Core no Linux

É possível executar aplicações .NET Core dependentes do runtime e independentes no AWS Elastic Beanstalk.

Uma aplicação dependente de runtime usa um runtime do .NET Core que o Elastic Beanstalk fornece para executar sua aplicação. O Elastic Beanstalk usa o arquivo `runtimeconfig.json` no pacote de origem para determinar o runtime a ser usado para sua aplicação. O Elastic Beanstalk escolhe o runtime compatível mais recente disponível para a aplicação.

Um aplicativo independente inclui o runtime do .NET Core, o aplicativo e suas dependências. Para usar uma versão do runtime do .NET Core não incluída pelo Elastic Beanstalk em suas plataformas, forneça uma aplicação independente.

Exemplos

É possível compilar aplicativos independentes e dependentes do runtime com o comando `dotnet publish`. Para saber mais sobre a publicação de aplicativos .NET Core, consulte [Visão geral da publicação de aplicativos .NET Core](#) na documentação do .NET Core.

A estrutura de arquivos demonstrativa a seguir define uma única aplicação que usa um runtime do .NET Core fornecido pelo Elastic Beanstalk.

```
### appsettings.Development.json
### appsettings.json
### dotnetcoreapp.deps.json
### dotnetcoreapp.dll
### dotnetcoreapp.pdb
### dotnetcoreapp.runtimeconfig.json
### web.config
### Procfile
### .ebextensions
### .platform
```

É possível incluir vários aplicativos no pacote de origem. O exemplo a seguir define dois aplicativos a serem executados no mesmo servidor Web. Para executar vários aplicativos, é necessário incluir um [Procfile](#) no pacote de origem. Para obter um exemplo completo de aplicativo, consulte [dotnet-core-linux-multiple-apps.zip](#).


```
### DotnetMultipleApp1
#   ### Amazon.Extensions.Configuration.SystemsManager.dll
#   ### appsettings.Development.json
#   ### appsettings.json
#   ### AWSSDK.Core.dll
#   ### AWSSDK.Extensions.NETCore.Setup.dll
#   ### AWSSDK.SimpleSystemsManagement.dll
#   ### DotnetMultipleApp1.deps.json
#   ### DotnetMultipleApp1.dll
#   ### DotnetMultipleApp1.pdb
#   ### DotnetMultipleApp1.runtimeconfig.json
#   ### Microsoft.Extensions.PlatformAbstractions.dll
#   ### Newtonsoft.Json.dll
#   ### web.config
### DotnetMultipleApp2
#   ### Amazon.Extensions.Configuration.SystemsManager.dll
#   ### appsettings.Development.json
#   ### appsettings.json
#   ### AWSSDK.Core.dll
#   ### AWSSDK.Extensions.NETCore.Setup.dll
#   ### AWSSDK.SimpleSystemsManagement.dll
#   ### DotnetMultipleApp2.deps.json
#   ### DotnetMultipleApp2.dll
#   ### DotnetMultipleApp2.pdb
#   ### DotnetMultipleApp2.runtimeconfig.json
#   ### Microsoft.Extensions.PlatformAbstractions.dll
#   ### Newtonsoft.Json.dll
#   ### web.config
### Procfile
### .ebextensions
### .platform
```

Usar um Procfile para configurar seu ambiente .NET Core no Linux

Para executar várias aplicações no mesmo servidor Web, é necessário incluir um Procfile no pacote de origem que informa ao Elastic Beanstalk quais aplicações devem ser executadas.

Recomendamos sempre fornecer um Procfile no pacote de origem juntamente com o aplicativo. Dessa forma, você controla com precisão quais processos são executados pelo Elastic Beanstalk para o aplicativo e quais argumentos são recebidos por esses processos.

O exemplo a seguir usa um `Procfile` para especificar duas aplicações a serem executadas pelo Elastic Beanstalk no mesmo servidor Web.

Example Procfile

```
web: dotnet ./dotnet-core-app1/dotnetcoreapp1.dll
web2: dotnet ./dotnet-core-app2/dotnetcoreapp2.dll
```

Para obter detalhes sobre como escrever e usar um `Procfile`, expanda a seção `Buildfile e Procfile` em [the section called “Estender plataformas Linux”](#).

Configurar o servidor de proxy para o seu ambiente .NET Core no Linux

O AWS Elastic Beanstalk usa [nginx](#) como proxy reverso para retransmitir solicitações para a aplicação. O Elastic Beanstalk oferece uma configuração `nginx` padrão que você pode estender ou substituir completamente por sua própria configuração.

Por padrão, o Elastic Beanstalk configura o proxy `nginx` para encaminhar solicitações à sua aplicação na porta 5000. Você pode substituir a porta padrão definindo a [propriedade do ambiente](#) `PORT` para a porta de escuta do seu aplicativo principal.

Note

A porta que seu aplicativo escuta não afeta a porta na qual o servidor `nginx` escuta para receber solicitações do balanceador de carga.

Configurar o servidor proxy na sua versão da plataforma

Todas as plataformas AL2023/AL2 oferecem suporte a um recurso de configuração de proxy uniforme. Para obter mais informações sobre a configuração do servidor proxy nas versões de plataforma que executam o AL2023/AL2, expanda a seção `Configuração do proxy reverso` em [the section called “Estender plataformas Linux”](#).

O arquivo de configuração de exemplo a seguir estende a configuração de `nginx` do ambiente. A configuração direciona solicitações para `/api` para um segundo aplicativo Web que escuta na porta 5200 no servidor Web. Por padrão, o Elastic Beanstalk encaminha solicitações para uma única aplicação que escuta na porta 5000.

Example 01_custom.conf

```
location /api {
    proxy_pass          http://127.0.0.1:5200;
    proxy_http_version 1.1;

    proxy_set_header    Upgrade $http_upgrade;
    proxy_set_header    Connection $http_connection;
    proxy_set_header    Host $host;
    proxy_cache_bypass  $http_upgrade;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto $scheme;
}
```

Tutorial: Implantar uma aplicação ASP.NET Core no Linux usando o Elastic Beanstalk

Este tutorial descreve o processo de criação de uma nova aplicação ASP.NET Core e implantação em um ambiente do Amazon Linux 2 com o Elastic Beanstalk.

Neste tutorial, primeiro use a ferramenta da linha de comando dotnet do .NET Core SDK para fazer o seguinte:

- Gere uma aplicação que atenda a solicitações HTTP com o ASP.NET.
- Instale dependências de runtime.
- Compile e execute a aplicação Web localmente.
- Publique os artefatos da aplicação em um diretório de saída. Os artefatos incluem o código-fonte compilado, as dependências de runtime e os arquivos de configuração.

Depois, faça o seguinte com a aplicação recém-criada:

- Crie um pacote de origem de aplicação que contenha os artefatos publicados.
- Crie um ambiente do Amazon Linux 2 e implante a aplicação nele com o Elastic Beanstalk.
- Abra o URL do site criado pelo Elastic Beanstalk para executar a aplicação.

O código-fonte do aplicativo está disponível aqui: [dotnet-core-linux-tutorial-source.zip](#).

[O pacote de origem implantável está disponível aqui: dotnet-core-linux-tutorial -bundle.](#)

Seções

- [Pré-requisitos](#)
- [Gerar um projeto .NET Core como uma aplicação Web](#)
- [Iniciar um ambiente do Elastic Beanstalk e implantar a aplicação](#)
- [Limpeza](#)
- [Próximas etapas](#)

Pré-requisitos

Este tutorial usa o .NET Core SDK para gerar uma aplicação Web .NET Core básica, executá-la localmente e criar um pacote implantável.

Requisitos

- .NET Core 3.1 ou posterior

Como instalar o .NET Core SDK

1. Faça download do instalador em microsoft.com/net/core. Escolha a plataforma de desenvolvimento. Selecione Download .NET Core SDK (Fazer download do .NET Core SDK).
2. Execute o instalador e siga as instruções.

Note

Embora os exemplos neste tutorial sejam listagens da linha de comando do Windows o .NET Core SDK oferece suporte a plataformas de desenvolvimento em vários sistemas operacionais. Os comandos dotnet mostrados neste tutorial são consistentes em diferentes plataformas de desenvolvimento.

Este tutorial usa um utilitário ZIP de linha de comando para criar um pacote de origem que você pode implantar no Elastic Beanstalk. Para usar o comando zip no Windows, você pode instalar UnxUtils. (UnxUtils é uma coleção leve de utilitários de linha de comando úteis, como zip ls e.) Se preferir, você pode [usar o Windows Explorer](#) ou qualquer outro utilitário ZIP para criar arquivos de pacote de origem.

Para instalar UnxUtils

1. Baixe [UnxUtils](#).
2. Extraia o arquivo em um diretório local. Por exemplo, C:\Program Files (x86).
3. Adicione o caminho aos binários para sua variável de usuário PATH do Windows. Por exemplo, C:\Program Files (x86)\UnxUtils\usr\local\wbin.
 - a. Pressione a tecla Windows e insira **environment variables**.
 - b. Escolha Edit environment variables for your account.
 - c. Selecione PATH e, em seguida, Edit (Editar).
 - d. Adicione caminhos ao campo Variable value, separados por ponto e vírgula. Por exemplo:
C:\item1\path;C:\item2\path
 - e. Selecione OK duas vezes para aplicar as novas configurações.
 - f. Feche todas as janelas do prompt de comando em execução e abra novamente uma janela do prompt de comando.
4. Abra uma nova janela de solicitação de comando e execute o comando `zip` para verificar se ele funciona.

```
> zip -h
Copyright (C) 1990-1999 Info-ZIP
Type 'zip "-L"' for software license.
...
```

Gerar um projeto .NET Core como uma aplicação Web

Use a ferramenta da linha de comando `dotnet` para gerar um projeto de aplicação Web .NET Core em C# e executá-lo localmente. A aplicação Web predefinida NET Core exibe Hello World!.

Como gerar um novo projeto .NET Core

1. Abra uma nova janela de prompt de comando e navegue até a pasta do usuário.

```
> cd %USERPROFILE%
```

2. Use o comando `dotnet new` para gerar um novo projeto .NET Core.

```
C:\Users\username> dotnet new web -o dotnet-core-tutorial
```

```
The template "ASP.NET Core Empty" was created successfully.
```

```
Processing post-creation actions...
```

```
Running 'dotnet restore' on dotnet-core-tutorial\dotnet-core-tutorial.csproj...
```

```
  Determining projects to restore...
```

```
    Restored C:\Users\username\dotnet-core-tutorial\dotnet-core-tutorial.csproj (in 154 ms).
```

```
Restore succeeded.
```

Para executar o site localmente

1. Use o comando `dotnet restore` para instalar dependências.

```
C:\Users\username> cd dotnet-core-tutorial
C:\Users\username\dotnet-core-tutorial> dotnet restore
  Determining projects to restore...
  All projects are up-to-date for restore.
```

2. Use o comando `dotnet run` para criar e iniciar a aplicação localmente.

```
C:\Users\username\dotnet-core-tutorial> dotnet run
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\username\dotnet-core-tutorial
```

3. Abra localhost:5000 para visualizar o site pelo navegador da Web padrão.

A aplicação retorna `Hello World!`, que é exibido no navegador da Web.

Para executar a aplicação em um servidor Web, é necessário empacotar o código-fonte compilado com um arquivo de configuração `web.config` e as dependências de runtime. A ferramenta `dotnet` oferece um comando `publish` que reúne esses arquivos em um diretório de acordo com a configuração em `dotnet-core-tutorial.csproj`.

Para criar seu site

- Use o comando `dotnet publish` para gerar o código compilado e as dependências em uma pasta denominada `site`.

```
C:\users\username\dotnet-core-tutorial> dotnet publish -o site
Microsoft (R) Build Engine version 16.7.0-preview-20360-03+188921e2f for .NET
Copyright (C) Microsoft Corporation. All rights reserved.

Determining projects to restore...
All projects are up-to-date for restore.
dotnet-core-tutorial -> C:\Users\username\dotnet-core-tutorial\bin\Debug
\netcoreapp3.1\dotnet-core-tutorial.dll
dotnet-core-tutorial -> C:\Users\username\dotnet-core-tutorial\site\
```

Para criar um pacote de origem

- Use o comando `zip` para criar um pacote de origem denominado `dotnet-core-tutorial.zip`.

O pacote de origem contém todos os arquivos publicados na pasta do site.

Note

Se você usar um utilitário ZIP diferente, certifique-se de adicionar todos os arquivos à pasta raiz do arquivo ZIP resultante. Isso é necessário para uma implantação bem-sucedida da aplicação no ambiente do Elastic Beanstalk.

```
C:\users\username\dotnet-core-tutorial> cd site
C:\users\username\dotnet-core-tutorial\site> zip -r ../dotnet-core-tutorial.zip .
  adding: appsettings.Development.json (164 bytes security) (deflated 38%)
  adding: appsettings.json (164 bytes security) (deflated 39%)
```

```
adding: dotnet-core-tutorial.deps.json (164 bytes security) (deflated 93%)
adding: dotnet-core-tutorial.dll (164 bytes security) (deflated 58%)
adding: dotnet-core-tutorial.exe (164 bytes security) (deflated 57%)
adding: dotnet-core-tutorial.pdb (164 bytes security) (deflated 48%)
adding: dotnet-core-tutorial.runtimeconfig.json (164 bytes security) (deflated
33%)
adding: web.config (164 bytes security) (deflated 41%)
```

Note

Neste tutorial, você está executando apenas uma aplicação no servidor Web, portanto, não é necessário um Procfile no pacote de origem. No entanto, para implantar várias aplicações no mesmo servidor Web, é necessário incluir um Procfile. Para ter mais informações, consulte [Usar um Procfile para configurar seu ambiente .NET Core no Linux](#).

Iniciar um ambiente do Elastic Beanstalk e implantar a aplicação

Use o console do Elastic Beanstalk para iniciar um ambiente do Elastic Beanstalk e implantar o pacote de origem.

[Você pode baixar o pacote de origem aqui: dotnet-core-linux-tutorial -bundle](#)

Como executar um ambiente e implantar o código (console)

1. [Abra o console do Elastic Beanstalk com este link pré-configurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials &environmentType=LoadBalanced](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced)
2. Para Platform (Plataforma), selecione .NET Core on Linux (.NET Core no Linux).
3. Escolha Local file (Arquivo local) e Choose file (Escolher arquivo) e abra o pacote de origem.
4. Selecione Review and launch.
5. Revise as configurações disponíveis e, em seguida, escolha Create app. A aplicação grava Hello World! na resposta e retorna.

A criação do ambiente e a implantação do seu código exigem aproximadamente 10 minutos.

Ao iniciar um ambiente, os seguintes recursos são criados:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região*.elasticbeanstalk.com.

Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

O Elastic Beanstalk gerencia todos esses recursos. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Note

O bucket do Amazon S3 criado pelo Elastic Beanstalk é compartilhado entre ambientes e não é excluído quando o ambiente é encerrado. Para ter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon S3](#).

Limpeza

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. [O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente, como instâncias do Amazon EC2, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.](#)

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).

4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

Próximas etapas

À medida que você continuar a desenvolver a aplicação, é recomendável gerenciar os ambientes e implantar a aplicação sem criar um arquivo .zip manualmente nem fazer upload dele para o console do Elastic Beanstalk. A interface de [linha de comando do Elastic Beanstalk](#) (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da interface da linha de comando.

Se você usa o Visual Studio para desenvolver seu aplicativo, você também pode usar o AWS Toolkit for Visual Studio para implantar alterações em seu código, gerenciar seus ambientes do Elastic Beanstalk e gerenciar outros recursos. AWS Consulte [A AWS Toolkit for Visual Studio](#) para obter mais informações.

Para fins de desenvolvimento e teste, considere utilizar a funcionalidade de implantação do Elastic Beanstalk para adicionar uma instância de banco de dados gerenciada diretamente ao ambiente. Para obter informações sobre como configurar um banco de dados no ambiente, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

Por fim, se você planeja usar a aplicação em um ambiente de produção, recomendamos [configurar um nome de domínio personalizado](#) para o ambiente e [habilitar HTTPS](#) para conexões seguras.

O AWS Toolkit for Visual Studio: trabalhar com o .NET Core

O AWS Toolkit for Visual Studio é um plugin para o Visual Studio IDE. Com o toolkit, é possível implantar e gerenciar aplicações no Elastic Beanstalk enquanto trabalha no ambiente do Visual Studio.

Este tópico mostra como você pode executar as seguintes tarefas usando o AWS Toolkit for Visual Studio:

- Crie uma aplicação Web ASP.NET Core usando um modelo do Visual Studio.
- Crie um ambiente Amazon Linux do Elastic Beanstalk.
- Implante a aplicação Web ASP.NET Core no novo ambiente Amazon Linux.

Este tópico também explora como você pode usar o AWS Toolkit for Visual Studio para gerenciar os ambientes de aplicação do Elastic Beanstalk e monitorar a integridade da aplicação.

Seções

- [Pré-requisitos](#)
- [Criar um projeto de aplicação](#)
- [Criar um ambiente do Elastic Beanstalk e implantar a aplicação](#)
- [Encerrar um ambiente](#)
- [Gerenciar os ambientes de aplicações do Elastic Beanstalk](#)
- [Monitorar a integridade do aplicativo](#)

Pré-requisitos

Antes de iniciar este tutorial, é necessário instalar o AWS Toolkit for Visual Studio. Para obter instruções, consulte [Configurar o AWS Toolkit for Visual Studio](#).

Se nunca usou o toolkit, a primeira coisa que você precisará fazer após instalá-lo será inscrever-se com as credenciais da AWS com o toolkit. Para obter mais informações sobre isso, consulte [Fornecer credenciais da AWS](#).

Criar um projeto de aplicação

Se você não tiver um projeto de aplicação .NET Core no Visual Studio, poderá facilmente criar um usando um dos modelos de projeto do Visual Studio.

Como criar um novo projeto de aplicação Web ASP.NET Core

1. No Visual Studio, no menu File (Arquivo), escolha New (Novo) e, depois, escolha Project (Projeto).
2. Na caixa de diálogo Create a new project (Criar um projeto), selecione C#, Linux e Cloud (Nuvem).
3. Na lista de modelos de projeto exibida, selecione ASP.NET Core Web Application (Aplicação Web ASP.NET Core) e Next (Próximo).

 Note

Se a ASP.NET Core Web Application (Aplicação Web ASP.NET Core) não estiver listada nos modelos de projeto, você poderá instalá-la no Visual Studio.

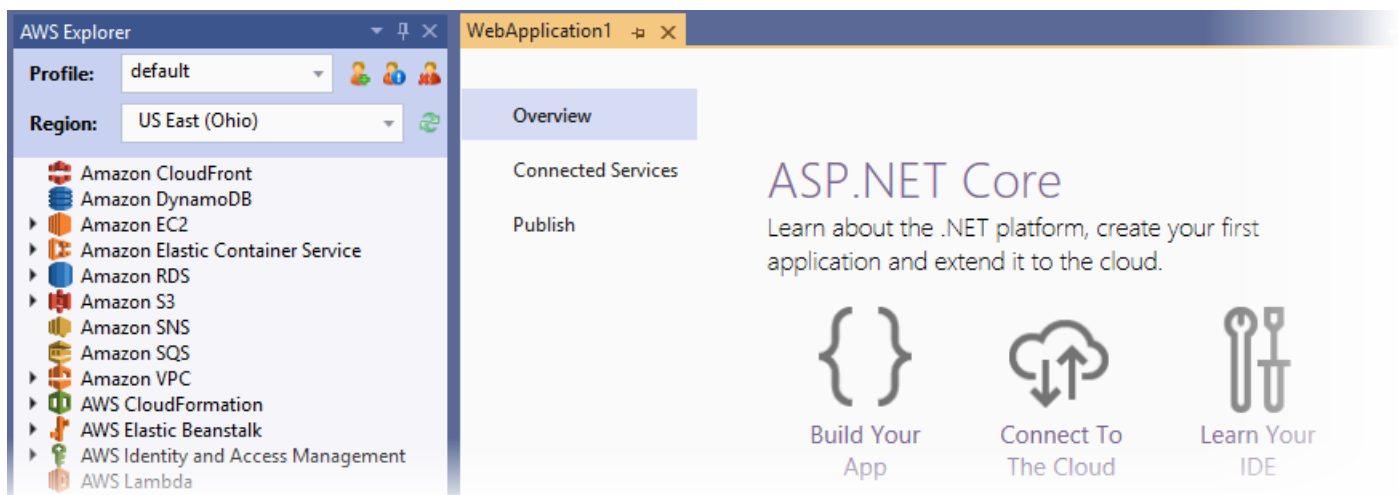
1. Role até a parte inferior da lista de modelos e selecione o link Install more tools and features (Instalar mais ferramentas e recursos) que está localizado na lista de modelos.
 2. Se você for solicitado a permitir que a aplicação do Visual Studio, faça alterações no dispositivo, selecione Yes (Sim).
 3. Selecione a guia Workloads (Cargas de trabalho) e a opção ASP.NET and web development (ASP.NET e desenvolvimento Web).
 4. Selecione o botão Modify (Modificar). O Visual Studio Installer (Instalador do Visual Studio) instala o modelo de projeto.
 5. Depois que o instalador for concluído, saia do painel para retornar ao ponto em que você parou no Visual Studio.
4. Na caixa de diálogo Configure your new project (Configurar o novo projeto), insira um Project name (Nome do projeto). O Solution name (Nome da solução) é padrão para o nome do projeto. Depois, selecione Create (Criar).
 5. Na caixa de diálogo Create a new ASP.NET Core web application (Criar uma aplicação Web ASP.NET Core), selecione .NET Core e ASP.NET Core 3.1. Na lista de tipos de aplicação exibida, selecione Web Application (Aplicação Web) e selecione o botão Create (Criar).

Create a new ASP.NET Core web application

.NET Core ASP.NET Core 3.1

- Empty**
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.
- API**
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.
- Web Application**
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.
- Web Application (Model-View-Controller)**
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

O Visual Studio exibe a caixa de diálogo Creating Project (Criando projeto) ao criar a aplicação. Depois que o Visual Studio concluir a geração da aplicação, será exibido um painel com o nome da aplicação.



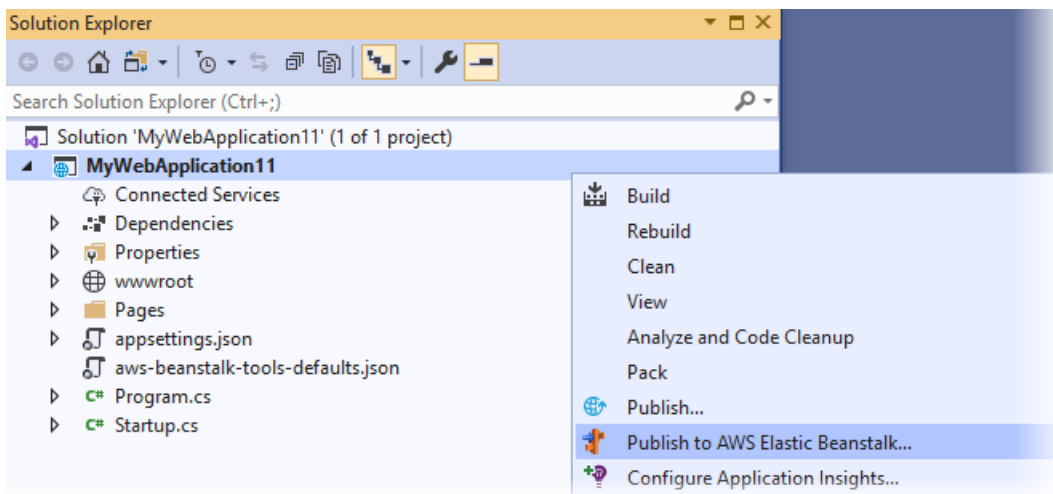
The screenshot shows the Visual Studio interface. On the left, the AWS Explorer sidebar is open, displaying a list of AWS services including Amazon CloudFront, Amazon DynamoDB, Amazon EC2, Amazon Elastic Container Service, Amazon RDS, Amazon S3, Amazon SNS, Amazon SQS, Amazon VPC, AWS CloudFormation, AWS Elastic Beanstalk, AWS Identity and Access Management, and AWS Lambda. The main window shows the 'WebApplication1' project. The 'Overview' tab is selected, displaying the 'ASP.NET Core' section. Below the title, it says 'Learn about the .NET platform, create your first application and extend it to the cloud.' There are three icons with text below them: a curly brace icon for 'Build Your App', a cloud with arrows icon for 'Connect To The Cloud', and a wrench and screwdriver icon for 'Learn Your IDE'.

Criar um ambiente do Elastic Beanstalk e implantar a aplicação

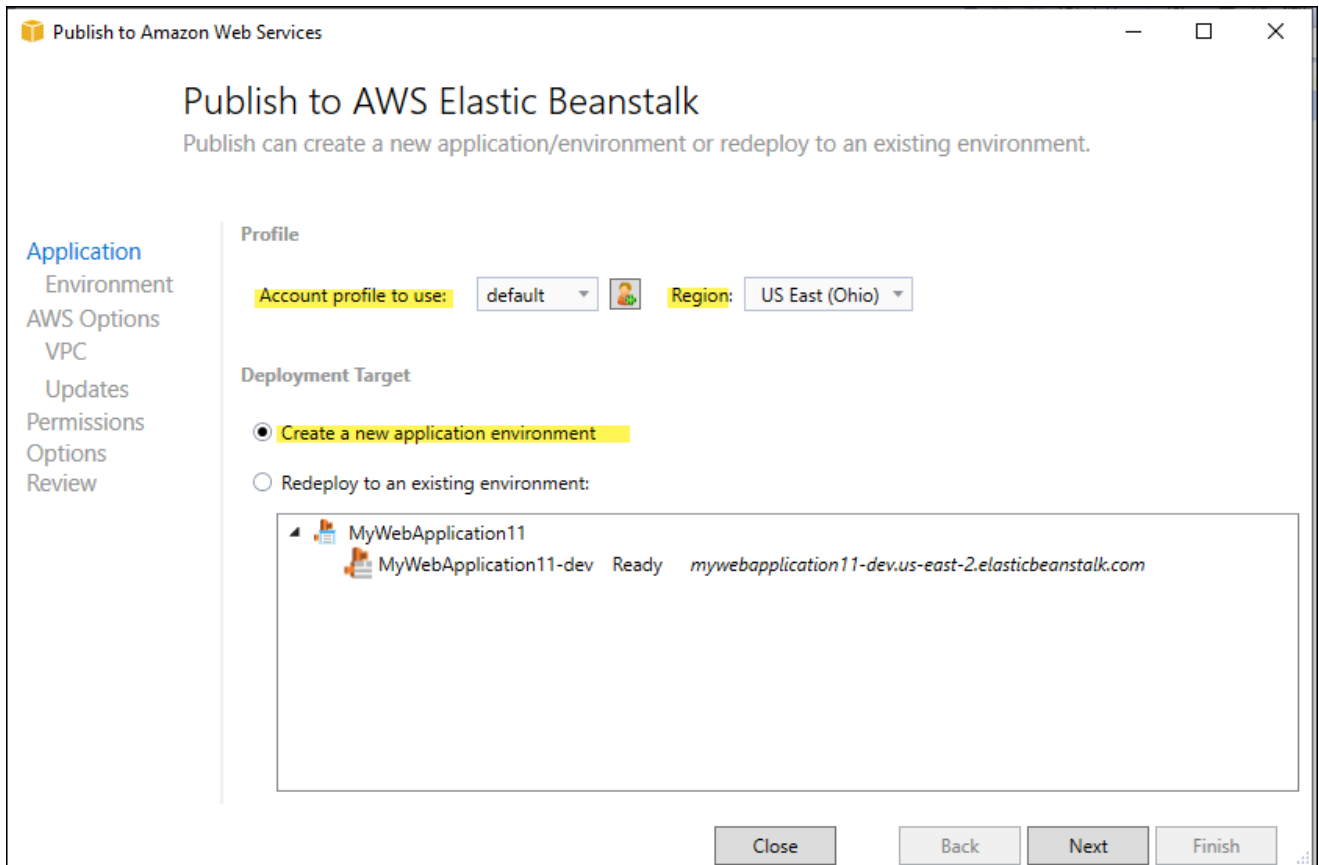
Esta seção descreve como criar um ambiente do Elastic Beanstalk para a aplicação e implantá-la nesse ambiente.

Como criar um ambiente e implantar a aplicação

1. No Visual Studio, selecione View (Visualizar) e, depois, Solution Explorer.
2. Em Solution Explorer, abra o menu de contexto (clique com o botão direito do mouse) da aplicação e selecione Publish to AWS Elastic Beanstalk (Publicar no).



3. No assistente Publicar na AWS Elastic Beanstalk, insira as informações da sua conta.
 - a. Em Account profile to use (Perfil de conta a ser usado), selecione a conta default (padrão) ou selecione o ícone Add another account (Adicionar outra conta) para inserir as informações da novas conta.
 - b. Em Region (Região), selecione a região em que você deseja implantar a aplicação. Para obter informações sobre regiões da AWS disponíveis, consulte [Endpoints e cotas do AWS Elastic Beanstalk](#) no Referência geral da AWS. Se você selecionar uma região não compatível com o Elastic Beanstalk, a opção para implantar no Elastic Beanstalk não estará disponível.
 - c. Selecione Create a new application environment (Criar um ambiente de aplicação) e Next (Próximo).



4. Na caixa de diálogo Application Environment (Ambiente de aplicação), insira os detalhes do novo ambiente de aplicação.
5. Na próxima caixa de diálogo de opções da AWS, defina as opções do Amazon EC2 e outras opções relacionadas à AWS para a aplicação implantada.
 - a. Em Container type (Tipo de contêiner), selecione 64bit Amazon Linux 2 v<n.n.n> running .NET Core (Amazon Linux 2 v<n.n.n> de 64 bits executando o .NET Core).

Note

Recomendamos que você selecione a versão atual da plataforma do Linux. Essa versão contém as correções de erros e segurança mais recentes que estão incluídas na imagem de máquina da Amazon (AMI) mais recente.

- b. Em Instance Type (Tipo de instância), selecione t2.micro. (Escolher um tipo de microinstância minimizará o custo associado à execução da instância.)
- c. Para Par de chaves, selecione Criar novo par de chaves. Insira um nome para o novo par de chaves e selecione OK. (Neste exemplo, usamos **myuseastkeypair**.) Um par de

chaves permite o acesso do desktop remoto em suas instâncias do Amazon EC2. Para obter mais informações sobre pares de chaves do Amazon EC2, consulte [Usar credenciais](#) no Guia do usuário da Amazon Elastic Compute Cloud.

- d. Para uma aplicação simples e de baixo tráfego, selecione Single instance environment (Ambiente de instância única). Para obter mais informações, consulte [Tipos de ambiente](#).
- e. Selecione Próximo.

Publish to Amazon Web Services

AWS

Set Amazon EC2 and other AWS-related options for the deployed application.

Application
Environment
AWS Options
VPC
Updates
Permissions
Options
Review

Amazon EC2 Launch Configuration

Container type *: 64bit Amazon Linux 2 v2.0.2 running .NET Core

Instance type *: t2.micro Key pair *: myuseastkeypair

Use custom AMI:

Use non-default VPC Single instance environment Enable Rolling Deployments

Load balancer type

Choose the type of load balancer to use when not creating a single instance environment.

Application
An Application Load Balancer makes routing decisions at the application layer (HTTP/HTTPS).

Relational Database Access

Select the Amazon RDS security groups to be modified to permit access from the EC2 instance(s) hosting your application.

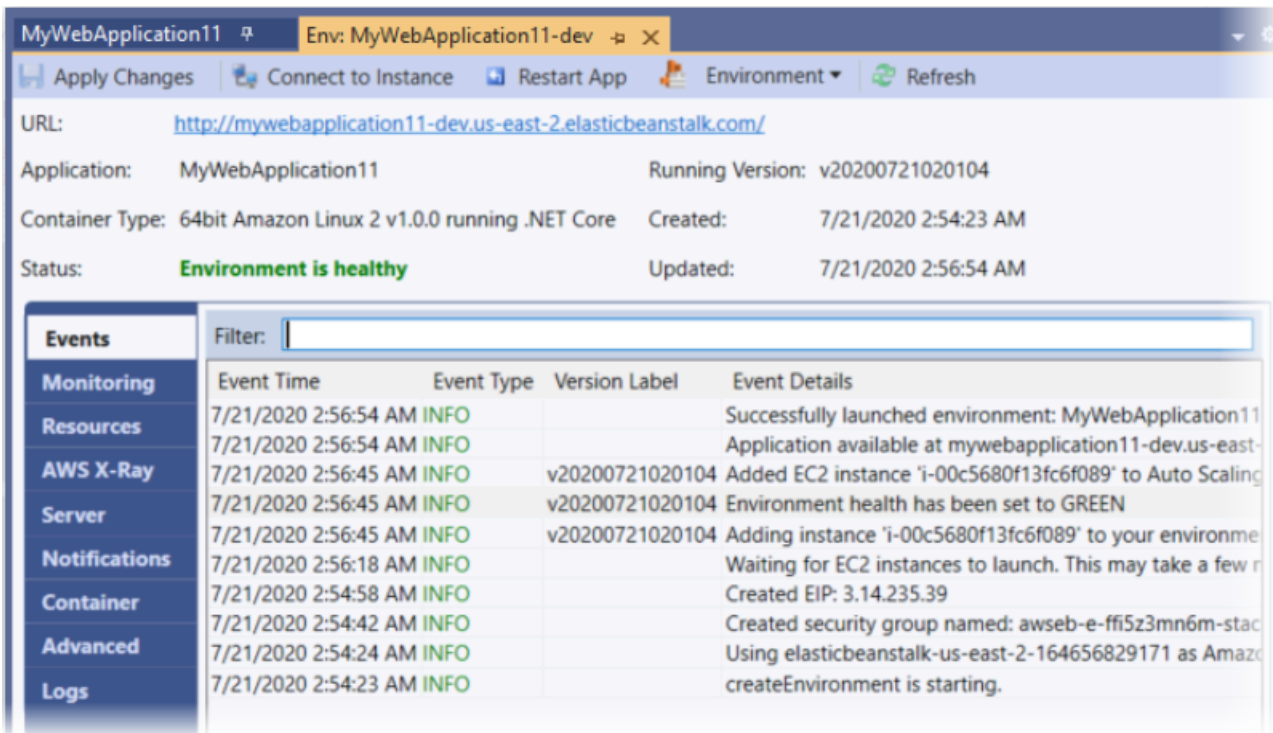
Close Back Next Finish

Para obter mais informações sobre as opções da AWS que não são usadas neste exemplo, considere as seguintes páginas:

- Em Use custom AMI (Usar AMI personalizada), consulte [Usar uma imagem de máquina da Amazon \(AMI\) personalizada](#).
- Se você não selecionar Single instance environment (Ambiente de instância única), será necessário escolher um Load balance type (Tipo de balanceamento de carga). Consulte [Balanceador de carga do ambiente do Elastic Beanstalk](#) para obter mais informações.

- O Elastic Beanstalk usará a configuração padrão da [Amazon VPC](#) (Amazon Virtual Private Cloud) se você não escolheu Use non-default VPC (Usar VPC não padrão). Para obter mais informações, consulte [Usar o Elastic Beanstalk com Amazon VPC](#).
 - Escolher a opção Enable Rolling Deployments (Habilitar implantações contínuas) divide uma implantação em lotes para evitar possíveis períodos de inatividade durante as implantações. Para obter mais informações, consulte [Implantar aplicações em ambientes do Elastic Beanstalk](#).
 - Escolher a opção Relational Database Access (Acesso ao banco de dados relacional) conecta o ambiente do Elastic Beanstalk a um banco de dados do Amazon RDS criado anteriormente com grupos de segurança de banco de dados do Amazon RDS. Para obter mais informações, consulte [Controlar o acesso com grupos de segurança](#) no Guia do usuário do Amazon RDS.
6. Selecione Next (Próximo) na caixa de diálogo Permissions (Permissões).
 7. Selecione Next (Próximo) na caixa de diálogo Applications Options (Opções de aplicações).
 8. Revise as opções de implantação. Depois de verificar se as configurações estão corretas, selecione Deploy (Implantar).

A aplicação Web ASP.NET Core é exportada como um arquivo de implantação da Web. Esse arquivo é, então, carregado para o Amazon S3 e registrado como uma nova versão da aplicação com o Elastic Beanstalk. O recurso de implantação do Elastic Beanstalk monitora o ambiente existente até ele se tornar disponível com o código recém-implantado. O Status do ambiente será exibido na guia Env:<nome do ambiente>. Depois que o status for atualizado para Environment is healthy (O ambiente está íntegro), você poderá selecionar o endereço URL para executar a aplicação Web.



Event Time	Event Type	Version Label	Event Details
7/21/2020 2:56:54 AM	INFO		Successfully launched environment: MyWebApplication11
7/21/2020 2:56:54 AM	INFO		Application available at mywebapplication11-dev.us-east-
7/21/2020 2:56:45 AM	INFO	v20200721020104	Added EC2 instance 'i-00c5680f13fc6f089' to Auto Scaling
7/21/2020 2:56:45 AM	INFO	v20200721020104	Environment health has been set to GREEN
7/21/2020 2:56:45 AM	INFO	v20200721020104	Adding instance 'i-00c5680f13fc6f089' to your environme
7/21/2020 2:56:18 AM	INFO		Waiting for EC2 instances to launch. This may take a few r
7/21/2020 2:54:58 AM	INFO		Created EIP: 3.14.235.39
7/21/2020 2:54:42 AM	INFO		Created security group named: awseb-e-ffi5z3mn6m-stac
7/21/2020 2:54:24 AM	INFO		Using elasticbeanstalk-us-east-2-164656829171 as Amaz
7/21/2020 2:54:23 AM	INFO		createEnvironment is starting.

Encerrar um ambiente

Para evitar cobranças de recursos da AWS não utilizados, use o AWS Toolkit for Visual Studio para terminar um ambiente em execução.

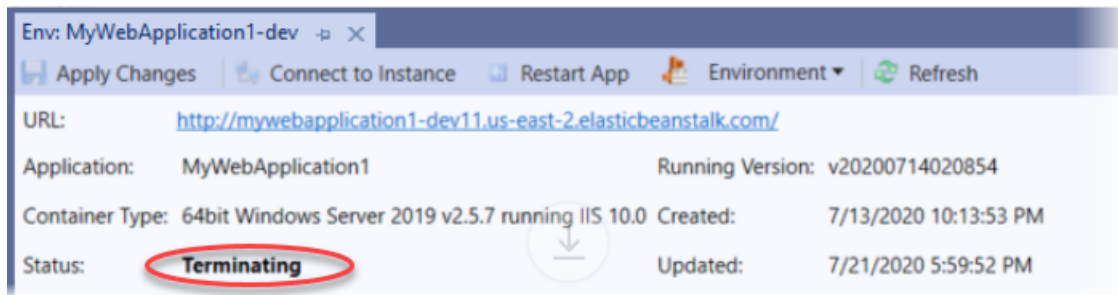
Note

Você sempre pode iniciar um novo ambiente usando a mesma versão mais tarde.

Para encerrar um ambiente

1. Expanda o nó Elastic Beanstalk e o nó da aplicação. No AWS Explorer, abra o menu de contexto (clique com o botão direito do mouse) do ambiente de aplicação e selecione Terminate Environment (Terminar ambiente).
2. Quando solicitado, selecione Yes (Sim) para confirmar que você deseja encerrar o ambiente. São necessários alguns minutos para o Elastic Beanstalk terminar os recursos da AWS em execução no ambiente.

O Status do ambiente na guia Env:<nome do ambiente> é alterado para Terminating (Encerrando) e, por fim, para Terminated (Encerrado).



Note

Quando você encerra o ambiente, o CNAME associado ao ambiente encerrado torna-se disponível para qualquer pessoa utilizar.

Gerenciar os ambientes de aplicações do Elastic Beanstalk

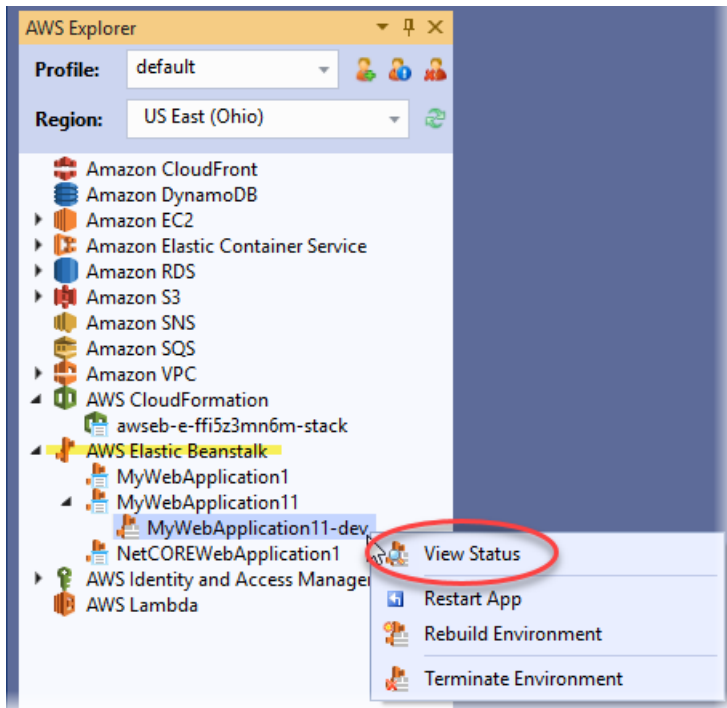
Com o AWS Toolkit for Visual Studio e o Console de Gerenciamento da AWS, você pode alterar o provisionamento e a configuração dos recursos da AWS usados por seus ambientes de aplicações. Para obter informações sobre como gerenciar seus ambientes de aplicações usando o Console de Gerenciamento da AWS, consulte [Gerenciar ambientes](#). Esta seção descreve as configurações de serviço específicas que você pode editar no AWS Toolkit for Visual Studio como parte da configuração de ambientes de aplicações.

Alterar os ajustes de configuração do ambiente

Quando a aplicação é implantada, o Elastic Beanstalk configura vários serviços de computação em nuvem conectados na AWS. É possível controlar como esses serviços individuais são configurados usando o AWS Toolkit for Visual Studio.

Para editar as configurações de ambiente de um aplicativo

1. No Visual Studio, no menu Arquivo, escolha AWS Explorer.
2. Expanda o nó do Elastic Beanstalk e o nó da aplicação. Abra o menu de contexto (clique com o botão direito do mouse) do ambiente de aplicação e selecione View Status (Visualizar status).



Agora, você pode definir as seguintes configurações:

- AWS X-Ray
- de aplicativos
- Load balancer (aplica-se apenas a ambientes de várias instâncias)
- Auto Scaling (aplica-se apenas a ambientes de várias instâncias)
- Notificações
- Contêiner
- Opções de configuração avançadas

Configurando o AWS X-Ray usando o AWS Toolkit for Visual Studio

AWS X-Ray fornece recursos de rastreamento de solicitações, coleta de exceções e criação de capacidades. Com o painel do AWS X-Ray, é possível habilitar ou desabilitar o X-Ray para sua aplicação. Para obter mais informações sobre o X-Ray, consulte o [Guia do desenvolvedor do AWS X-Ray](#).

Events

Monitoring

Resources

AWS X-Ray

Server

Notifications

Container

Advanced

Logs

AWS X-Ray is a service that collects data about requests that your application serves, and provides tools you can use to view, filter, and gain insights into that data to identify issues and opportunities for optimization. For any traced request to your application, you can see detailed information not only about the request and response, but also about calls that your application makes to downstream AWS resources, microservices, databases and HTTP web APIs.

Enable AWS X-Ray true

To see your application's service map and traces visit the [AWS X-Ray Console](#).

To learn how to instrument your .NET application visit the [AWS X-Ray SDK for .NET GitHub repository](#).

Configurar instâncias do EC2 usando o AWS Toolkit for Visual Studio

É possível usar o Amazon Elastic Compute Cloud (Amazon EC2) para executar e gerenciar instâncias de servidor nos datacenters da Amazon. Você pode usar instâncias de servidor Amazon EC2 a qualquer momento, pelo tempo que precisar e por quaisquer motivos legais. As instâncias estão disponíveis em diferentes tamanhos e configurações. Para obter mais informações, consulte [Amazon EC2](#).

É possível editar a configuração da instância do Amazon EC2 com a guia Server (Servidor) dentro da guia do ambiente da aplicação no AWS Toolkit for Visual Studio.

Events

Monitoring

Resources

AWS X-Ray

Server

Notifications

Container

Advanced

Logs

These settings allow you to control your environment's servers and enable login.

*EC2 Instance Type

*EC2 Security Group

*Existing Key Pair

*Monitoring Interval

*AMI ID

Note: *It may take a few minutes to see changes to these options take effect in your environment.

Tipos de instância do Amazon EC2

Instance type (Tipo de instância) exibe os tipos de instância disponíveis para a aplicação do Elastic Beanstalk. Altere o tipo de instância para selecionar um servidor com as características (inclusive o tamanho da memória e a capacidade de CPU) mais adequadas à aplicação. Por exemplo, os aplicativos com operações muito intensas e demoradas podem exigir mais CPU ou memória.

Para obter mais informações sobre os tipos de instância do Amazon EC2 disponíveis para a aplicação do Elastic Beanstalk, consulte [Instance Types \(Tipos de instância\)](#) no Guia do usuário do Amazon Elastic Compute Cloud.

Grupos de segurança do Amazon EC2

É possível controlar o acesso à sua aplicação do Elastic Beanstalk usando um grupo de segurança do Amazon EC2. Um grupo de segurança define regras de firewall para suas instâncias. Essas regras especificam qual tráfego de rede de entrada deve ser fornecido para a instância. Todo o outro tráfego de entrada é descartado. Você pode modificar as regras para um grupo a qualquer momento. As novas regras são aplicadas automaticamente para todas as instâncias em execução e as iniciadas no futuro.

É possível especificar quais grupos de segurança do Amazon EC2 controlam o acesso à aplicação do Elastic Beanstalk. Para fazer isso, insira os nomes de grupos de segurança específicos do Amazon EC2 (separando vários grupos de segurança com vírgulas) na caixa de texto EC2 Security Groups (Grupos de segurança do EC2). É possível fazer isso usando o Console de Gerenciamento da AWS ou o AWS Toolkit for Visual Studio.

Como criar um grupo de segurança usando o AWS Toolkit for Visual Studio

1. No Visual Studio, no AWS Explorer, expanda o nó Amazon EC2 e selecione Security Groups (Grupos de segurança).
2. Selecione Create Security Group (Criar grupo de segurança) e insira um nome e uma descrição para o grupo de segurança.
3. Selecione OK.

Para obter mais informações sobre os grupos de segurança do Amazon EC2, consulte [Usar grupos de segurança](#) no Guia do usuário do Amazon Elastic Compute Cloud.

Pares de chave do Amazon EC2

É possível fazer login com segurança nas instâncias do Amazon EC2 provisionadas para a sua aplicação do Elastic Beanstalk com um par de chaves do Amazon EC2.

Important

É necessário criar um par de chaves do Amazon EC2 e configurar as instâncias do Amazon EC2 provisionadas pelo Elastic Beanstalk para poder acessar essas instâncias. É necessário

criar seu par de chaves usando o assistente Publicar na AWS dentro do Toolkit for Visual Studio da AWS ao implantar a aplicação no Elastic Beanstalk. Se você quiser criar pares de chaves adicionais usando o Toolkit, siga as etapas descritas aqui. Também há a opção de configurar os pares de chave do Amazon EC2 usando o [Console de Gerenciamento da AWS](#). Para obter instruções sobre como criar um par de chaves do Amazon EC2, consulte o [Guia de conceitos básicos do Amazon Elastic Compute Cloud](#).

A caixa de texto Existing Key Pair (Par de chaves existente) permite que você especifique o nome de um par de chaves do Amazon EC2 que pode ser usado para fazer login com segurança nas instâncias do Amazon EC2 que executam a aplicação do Elastic Beanstalk.

Para especificar o nome de um par de chaves do Amazon EC2

1. Expanda o nó Amazon EC2 e selecione Key Pairs (Pares de chaves).
2. Selecione Create Key Pair (Criar par de chaves) e insira o nome do par de chaves.
3. Selecione OK.

Para obter mais informações sobre pares de chaves do Amazon EC2, acesse [Usar credenciais do Amazon EC2](#) no Guia do usuário do Amazon Elastic Compute Cloud. Para obter mais informações sobre como se conectar às instâncias do Amazon EC2, consulte

Intervalo de monitoramento

Por padrão, somente as métricas básicas do Amazon CloudWatch estão habilitadas. Elas retornam dados em intervalos de cinco minutos. Você pode habilitar métricas mais detalhadas do CloudWatch, em intervalos de um minuto, selecionando 1 minute (1 minuto) para o Monitoring Interval (Intervalo de monitoramento) na seção Server (Servidor) da guia Configuration (Configuração) para o ambiente no AWS Toolkit for Eclipse.

Note

As cobranças de serviço do Amazon CloudWatch podem se aplicar a métricas de intervalo de um minuto. Consulte [Amazon CloudWatch](#) para obter mais informações.

ID da AMI personalizado

Você pode modificar a AMI padrão usada nas instâncias do Amazon EC2 pela própria AMI personalizada inserindo o identificador da AMI personalizada na caixa Custom AMI ID (ID da AMI personalizada) na seção Server (Servidor) da guia Configuration (Configuração) para o ambiente no AWS Toolkit for Eclipse.

Important

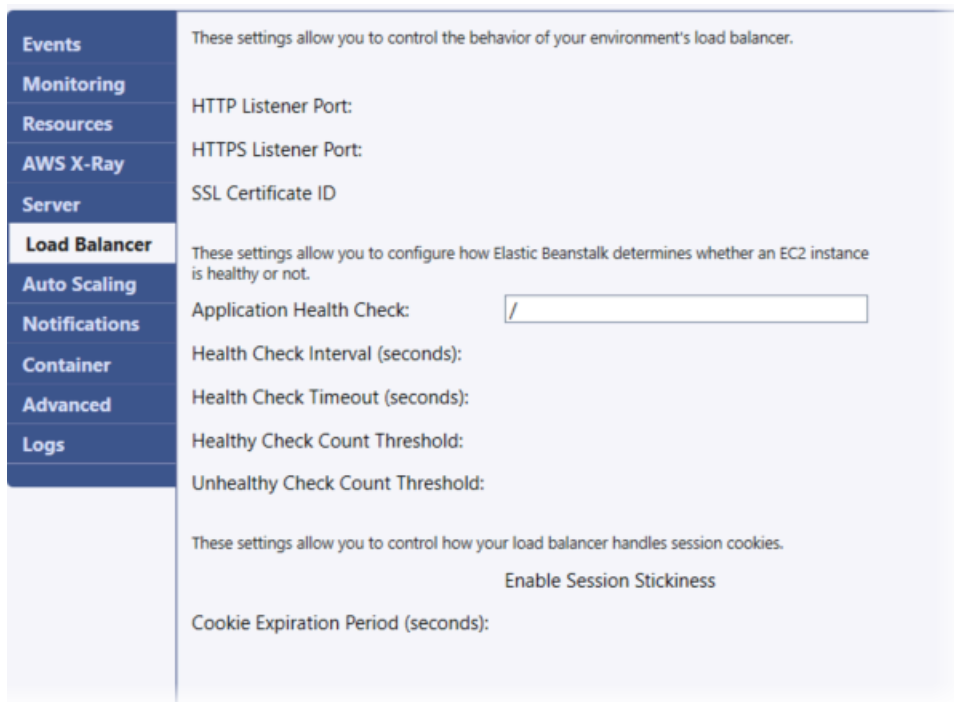
O uso da AMI própria é uma tarefa avançada que deve ser feita com cuidado. Se você precisar de uma AMI personalizada, recomendamos começar com a AMI do Elastic Beanstalk padrão e depois modificá-la. Para ser considerado íntegro, o Elastic Beanstalk espera que as instâncias do Amazon EC2 atendam a um conjunto de requisitos, inclusive ter um gerenciador de host em execução. Se esses requisitos não forem atendidos, talvez o ambiente não funcione corretamente.

Configuração do Elastic Load Balancing usando o AWS Toolkit for Visual Studio

O Elastic Load Balancing é um serviço web da Amazon que ajuda você a aprimorar a disponibilidade e a escalabilidade de seu aplicativo. Este serviço facilita a distribuição de cargas de aplicativo entre duas ou mais instâncias do Amazon EC2. O Elastic Load Balancing melhora a disponibilidade ao fornecer mais redundância e acomoda o crescimento de tráfego da aplicação.

Com o Elastic Load Balancing, é possível distribuir e equilibrar automaticamente o tráfego de entrada de aplicações entre todas as instâncias em execução. Também é possível adicionar facilmente novas instâncias quando for necessário aumentar a capacidade da aplicação.

O Elastic Beanstalk provisiona automaticamente o Elastic Load Balancing ao implantar uma aplicação. É possível editar a configuração da instância do Amazon EC2 do ambiente do Elastic Beanstalk na guia Load Balancer (Balanceador de carga) na guia do ambiente da aplicação no AWS Toolkit for Visual Studio.



As seções a seguir descrevem os parâmetros do Elastic Load Balancing que você pode configurar para o seu aplicativo.

Portas

O balanceador de carga provisionado para lidar com as solicitações da aplicação do Elastic Beanstalk envia solicitações para as instâncias do Amazon EC2 que estão executando a aplicação. O balanceador de carga provisionado pode receber solicitações em portas HTTP e HTTPS e encaminhá-las para as instâncias do Amazon EC2 em sua aplicação do AWS Elastic Beanstalk. Por padrão, o load balancer lida com solicitações na porta HTTP. Para realizar esse trabalho, pelo menos uma das portas (HTTP ou HTTPS) deve estar ativada.



⚠ Important

A porta especificada não deve estar bloqueada. Caso contrário, você não conseguirá se conectar à aplicação do Elastic Beanstalk.

Controle da porta HTTP

Para desativar a porta HTTP, selecione OFF em HTTP Listener Port. Para ativar a porta HTTP, você deve selecionar uma porta HTTP (por exemplo, a 80) na lista.

Note

Para acessar o ambiente usando uma porta diferente da porta 80 padrão, como a porta 8080, adicione uma escuta ao load balancer existente e configure a nova escuta nessa porta. Por exemplo, usando a [AWS CLI for Classic Load Balancers](#), digite o comando a seguir, substituindo *LOAD_BALANCER_NAME* pelo nome do balanceador de carga do Elastic Beanstalk.

```
aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
--listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
InstancePort=80"
```

Por exemplo, usando a [AWS CLI para Application Load Balancers](#), digite o comando a seguir, substituindo *LOAD_BALANCER_ARN* pelo ARN do balanceador de carga do Elastic Beanstalk.

```
aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP
--port 8080
```

Se você quiser que o Elastic Beanstalk monitore o ambiente, não remova a escuta na porta 80.

Controle da porta HTTPS

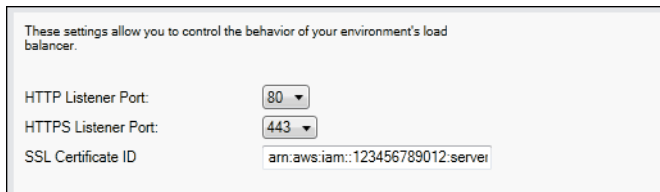
O Elastic Load Balancing oferece suporte ao protocolo HTTPS/TLS para permitir a criptografia do tráfego para conexões de clientes com o load balancer. As conexões do load balancer para as instâncias do EC2 usam criptografia de texto simples. Por padrão, a porta HTTPS é desativada.

Para ativar a porta HTTPS

1. Crie um novo certificado usando o AWS Certificate Manager (ACM), ou faça upload de um certificado e uma chave no AWS Identity and Access Management (IAM). Para obter mais informações sobre como solicitar um certificado do ACM, consulte [Solicitação de um certificado](#)

no Manual do usuário do AWS Certificate Manager. Para obter mais informações sobre a importação de certificados de terceiros para o ACM, consulte [Importação de certificados](#) no Manual do usuário do AWS Certificate Manager. Se o ACM [não estiver disponível](#) em sua região, use o AWS Identity and Access Management (IAM) para fazer upload de um certificado de terceiros. Os serviços ACM e IAM armazenarão o certificado e fornecerão um nome de recurso da Amazon (ARN) para o certificado SSL. Para obter mais informações sobre a criação e o upload de certificados para o IAM, consulte [Trabalhar com certificados de servidor](#) no Guia do usuário do IAM.

2. Especifique a porta HTTPS selecionando uma porta em HTTPS Listener Port.



These settings allow you to control the behavior of your environment's load balancer.

HTTP Listener Port:	80
HTTPS Listener Port:	443
SSL Certificate ID	arn:aws:iam::123456789012:server-certificate/abc/certs/build

3. Para SSL Certificate ID (ID de certificado SSL), insira o Nome de recurso da Amazon (ARN) do seu certificado SSL. Por exemplo, o **arn:aws:iam::123456789012:server-certificate/abc/certs/build** ou o **arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678**. Use o certificado SSL que você criou ou do qual fez upload na etapa 1.

Para desativar a porta HTTPS, selecione OFF em HTTPS Listener Port.

Verificações de integridade

A definição de verificação de integridade inclui um URL a ser consultado para saber a integridade da instância. Por padrão, o Elastic Beanstalk usa TCP: 80 para contêineres não legados e HTTP: 80 para contêineres legados. É possível substituir o URL padrão para corresponder a um recurso existente na aplicação (por exemplo, `/myapp/default.aspx`), inserindo-o na caixa Application Health Check URL (URL de verificação de integridade da aplicação). Se você substituir o URL padrão, o Elastic Beanstalk usará HTTP para consultar o recurso. Para verificar se você está usando um tipo de contêiner legado, consulte [the section called "Por que algumas versões de plataforma são marcadas como legadas?"](#)

Você pode controlar as configurações da verificação de integridade usando a seção EC2 Instance Health Check do painel Load Balancing.

These settings allow you to configure how Elastic Beanstalk determines whether an EC2 instance is healthy or not.

Application Health Check:	<input type="text" value="/"/>	
Health Check Interval (seconds):	<input type="text" value="30"/>	(5 - 300)
Health Check Timeout (seconds):	<input type="text" value="5"/>	(2 - 60)
Healthy Check Count Threshold:	<input type="text" value="3"/>	(2 - 10)
Unhealthy Check Count Threshold:	<input type="text" value="5"/>	(2 - 10)

A definição de verificação de integridade inclui um URL a ser consultado para saber a integridade da instância. Substitua o URL padrão para corresponder a um recurso existente na aplicação (por exemplo, `/myapp/index.jsp`), inserindo-o na caixa Application Health Check URL (URL de verificação de integridade da aplicação).

A lista a seguir descreve os parâmetros de verificação de integridade que você pode definir para o aplicativo.

- Em Health Check Interval (seconds), insira o número de segundos que o Elastic Load Balancing aguarda entre as verificações de integridade das instâncias do Amazon EC2 do seu aplicativo.
- Em Health Check Timeout (seconds), especifique o número de segundos que o Elastic Load Balancing aguarda por uma resposta antes que ele considere que a instância não está respondendo.
- Em Healthy Check Count Threshold e Unhealthy Check Count Threshold, especifique o número de testes consecutivos de URL bem-sucedidos ou sem êxito antes que o Elastic Load Balancing altere o status de integridade da instância. Por exemplo, especificar **5** em Unhealthy Check Count Threshold (Limite de contagem de verificações de falta de integridade) significa que o URL deve retornar uma mensagem de erro ou de tempo esgotado cinco vezes consecutivas antes que o Elastic Load Balancing considere que a verificação de integridade falhou.

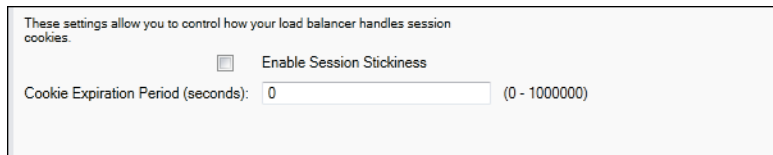
Sessões

Por padrão, um load balancer roteia cada solicitação de forma independente para a instância de servidor com a menor carga. Por comparação, uma sticky session vincula a sessão do usuário a uma determinada instância do servidor, para que todas as solicitações desse usuário durante a sessão sejam enviadas para a mesma instância do servidor.

O Elastic Beanstalk usa cookies HTTP gerados pelo balanceador de carga quando as sticky sessions são permitidas em uma aplicação. O load balancer usa um cookie especial gerado pelo load balancer para rastrear cada solicitação na instância do aplicativo. Quando o load balancer receber uma solicitação, ele primeiro verificará se esse cookie está presente na solicitação. Se estiver presente, a solicitação será enviada para a instância da aplicação especificada no cookie. Se não

houver um cookie, o load balancer selecionará uma instância de aplicativo com base no algoritmo de balanceamento de carga existente. Um cookie será inserido na resposta para vincular as solicitações subsequentes do mesmo usuário para essa instância do aplicativo. A configuração da política define a expiração do cookie, que estabelece a validade de cada cookie.

É possível usar a seção Sessions (Sessões) na guia Load Balancer para especificar se deseja que o load balancer da aplicação permita a perdurabilidade da sessão.



The screenshot shows a configuration box for session stickiness. At the top, it says "These settings allow you to control how your load balancer handles session cookies." Below this, there is a checkbox labeled "Enable Session Stickiness" which is currently unchecked. Underneath the checkbox, there is a text input field for "Cookie Expiration Period (seconds)" with the value "0" entered. To the right of the input field, the range "(0 - 1000000)" is displayed.

Para obter mais informações sobre Elastic Load Balancing, consulte o [Guia do desenvolvedor de Elastic Load Balancing](#).

Configurar o Auto Scaling usando o AWS Toolkit for Visual Studio

O Amazon EC2 Auto Scaling é um serviço da Web da Amazon projetado para executar ou encerrar automaticamente instâncias do Amazon EC2 com base em triggers definidos pelo usuário. É possível configurar grupos de Auto Scaling e associar triggers a esses grupos para escalar automaticamente recursos de computação com base em métricas como uso de largura de banda ou utilização da CPU. O Amazon EC2 Auto Scaling funciona com o Amazon CloudWatch para recuperar métricas das instâncias do servidor que executam a sua aplicação.

O Amazon EC2 Auto Scaling permite que você pegue um grupo de instâncias do Amazon EC2 e defina vários parâmetros para que esse grupo aumente ou diminua automaticamente em número. O Amazon EC2 Auto Scaling pode adicionar ou remover instâncias do Amazon EC2 desse grupo para ajudar você a lidar com as alterações de tráfego em sua aplicação sem problemas.

O Amazon EC2 Auto Scaling também monitora a integridade de cada instância do Amazon EC2 que ele executa. Se alguma instância for encerrada inesperadamente, o Amazon EC2 Auto Scaling detectará o encerramento e executará uma instância de substituição. Esse recurso permite que você mantenha automaticamente um número desejado fixo de instâncias do Amazon EC2.

O Elastic Beanstalk provisiona o Amazon EC2 Auto Scaling para sua aplicação. É possível editar a configuração da instância do Amazon EC2 do ambiente do Elastic Beanstalk na guia Auto Scaling, na guia do ambiente da aplicação no AWS Toolkit for Visual Studio.

Events	Auto-scaling automatically launches or terminates EC2 instances based on defined metrics and thresholds called triggers. Auto-scaling will also launch a new EC2 instance in the event of a failure. These settings allow you to control auto-scaling behavior.	
Monitoring		
Resources		
AWS X-Ray		
Server		
Load Balancer		
Auto Scaling		
Notifications		
Container		
Advanced		
Logs		
	Minimum Instance Count:	<input type="text" value="1"/> (0 - 10000)
	Maximum Instance Count:	<input type="text" value="4"/> (0 - 10000)
	Availability Zones:	<input type="text" value="Any"/>
	Scaling Cooldown Time (seconds):	<input type="text" value="360"/> (0 - 10000)
	Trigger Measurement:	<input type="text" value="NetworkOut"/>
	Trigger Statistic:	<input type="text" value="Average"/>
	Unit of Measurement:	<input type="text" value="Bytes"/>
	Measurement Period (minutes):	<input type="text" value="5"/> (1 - 600)
	Breach Duration (minutes):	<input type="text" value="5"/> (1 - 600)
	Upper Threshold:	<input type="text" value="6000000"/>
	Upper Breach Scalement Increment:	<input type="text" value="1"/>
	Lower Threshold:	<input type="text" value="2000000"/>
	Lower Breach Scalement Increment:	<input type="text" value="-1"/>

A seção a seguir discute como configurar os parâmetros de Auto Scaling para seu aplicativo.

Iniciar a configuração

É possível editar a configuração de execução para controlar como sua aplicação do Elastic Beanstalk provisiona os recursos do Amazon EC2 Auto Scaling.

As caixas Minimum Instance Count (Contagem de instância mínima) e Maximum Instance Count (Contagem de instância máxima) permitem que você especifique os tamanhos mínimo e máximo do grupo de Auto Scaling que a aplicação do Elastic Beanstalk usa.

Auto-scaling automatically launches or terminates EC2 instances based on defined metrics and thresholds called triggers. Auto-scaling will also launch a new EC2 instance in the event of a failure. These settings allow you to control auto-scaling behavior.		
Minimum Instance Count:	<input type="text" value="1"/>	(1 - 10000)
Maximum Instance Count:	<input type="text" value="4"/>	(1 - 10000)
Availability Zones:	<input type="text" value="Any"/>	
Scaling Cooldown Time (seconds):	<input type="text" value="360"/>	(0 - 10000)

Note

Para manter um número fixo de instâncias do Amazon EC2, defina Minimum Instance Count e Maximum Instance Count com o mesmo valor.

A caixa Availability Zones permite que você especifique o número de zonas de disponibilidade em que deseja que as suas instâncias do Amazon EC2 estejam. É importante definir esse número se quiser criar aplicativos tolerantes a falhas. Se uma zona de disponibilidade ficar inativa, as instâncias continuarão sendo executadas em nas outras zonas de disponibilidade.

Note

No momento, não é possível especificar a zona de disponibilidade na qual sua instância estará.

Acionadores

Um trigger é um mecanismo do Amazon EC2 Auto Scaling que é definido para informar ao sistema quando você deseja aumentar (expandir) e diminuir (reduzir) o número de instâncias. É possível configurar os triggers para acionar quaisquer métricas publicadas no Amazon CloudWatch, como utilização de CPU, e determinar se as condições especificadas foram atendidas. Quando o limite superior ou inferior das condições que você especificou para a métrica forem violados para o período especificado, o trigger iniciará um processo de longa duração chamado de ação de escalabilidade.

É possível definir um acionador de escalabilidade para a aplicação do Elastic Beanstalk usando o AWS Toolkit for Visual Studio.

Trigger Measurement:	<input type="text" value="NetworkOut"/>	
Trigger Statistic:	<input type="text" value="Average"/>	
Unit of Measurement:	<input type="text" value="Bytes"/>	
Measurement Period (minutes):	<input type="text" value="5"/>	(1 - 600)
Breach Duration (minutes):	<input type="text" value="5"/>	(1 - 600)
Upper Threshold:	<input type="text" value="6000000"/>	(0 - 20000000)
Upper Breach Scalement Increment:	<input type="text" value="1"/>	
Lower Threshold:	<input type="text" value="2000000"/>	(0 - 20000000)
Lower Breach Scalement Increment:	<input type="text" value="-1"/>	

Os triggers do Amazon EC2 Auto Scaling funcionam monitorando uma métrica específica do Amazon CloudWatch de uma instância específica. As métricas incluem utilização de CPU, tráfego de rede e

a atividade do disco. Use a configuração Trigger Measurement para selecionar uma métrica para o trigger.

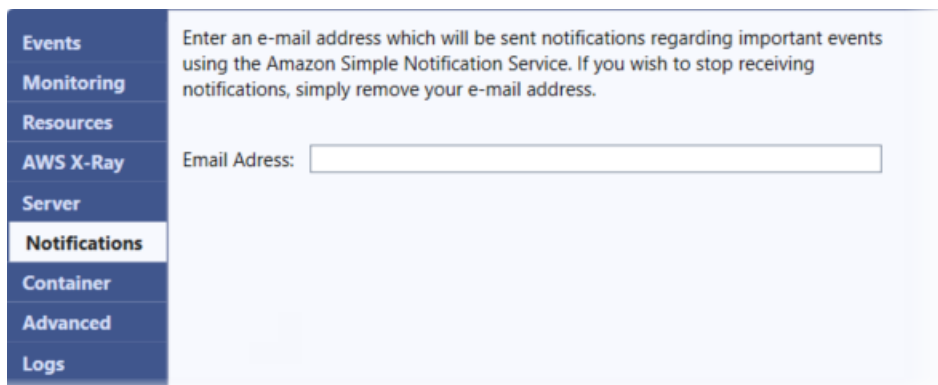
A lista a seguir descreve os parâmetros do acionador que você pode configurar usando o Console de Gerenciamento da AWS.

- Você pode especificar quais estatísticas o trigger deve usar. Você pode selecionar Minimum, Maximum, Sum ou Average em Trigger Statistic.
- Em Unit of Measurement, especifique a unidade de medida do trigger.
- O valor na caixa Measurement Period (Período de medição) especifica a frequência com que o Amazon CloudWatch mede as métricas do trigger. A Breach Duration (Duração da ruptura) é a quantidade de tempo que uma métrica pode durar além do limite definido (conforme especificado em Upper Threshold (Limite superior) e Lower Threshold (Limite inferior)) antes que o trigger seja acionado.
- Em Upper Breach Scale Increment e Lower Breach Scale Increment, especifique quantas instâncias do Amazon EC2 devem ser adicionadas ou removidas ao realizar uma ação de escalabilidade.

Para obter mais informações sobre o Amazon EC2 Auto Scaling, consulte a seção Amazon EC2 Auto Scaling na [documentação do Amazon Elastic Compute Cloud](#).

Configurar as notificações usando o AWS Toolkit for Visual Studio

O Elastic Beanstalk usa o Amazon Simple Notification Service (Amazon SNS) para notificar você de eventos importantes que afetam sua aplicação. Para permitir notificações do Amazon SNS, insira seu endereço de e-mail na caixa Email Address (Endereço de e-mail). Para desativar essas notificações, remova seu endereço de e-mail da caixa.

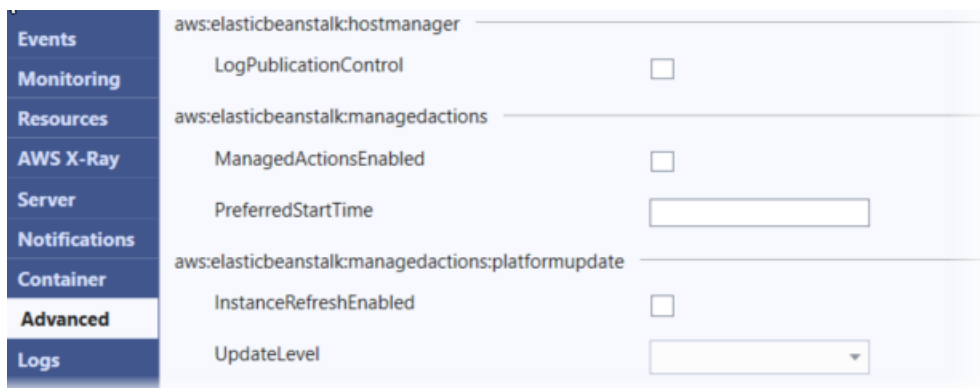


The screenshot shows a configuration window for notifications. On the left is a vertical navigation menu with the following items: Events, Monitoring, Resources, AWS X-Ray, Server, Notifications (highlighted), Container, Advanced, and Logs. The main area contains the following text: "Enter an e-mail address which will be sent notifications regarding important events using the Amazon Simple Notification Service. If you wish to stop receiving notifications, simply remove your e-mail address." Below this text is a label "Email Address:" followed by an empty text input field.

Configurar opções de ambiente adicionais usando o AWS Toolkit for Visual Studio

O Elastic Beanstalk define um grande número de opções de configuração que podem ser usadas para configurar o comportamento de seu ambiente e os recursos contidos neles. As opções de configuração são organizadas em namespaces como `aws:autoscaling:asg`. Cada namespace define opções para o grupo de Auto Scaling de um ambiente. O painel Advanced (Avançado) lista os namespaces da opção de configuração em ordem alfabética que você pode atualizar após a criação do ambiente.

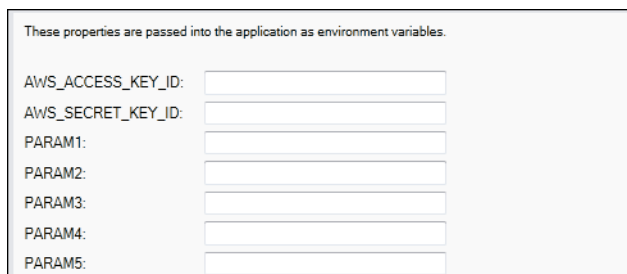
Para obter uma lista completa de namespaces e opções, incluindo os valores padrão e compatíveis de cada um deles, consulte [Opções gerais para todos os ambientes](#) e [Opções da plataforma .NET Core no Linux](#).



Events	aws:elasticbeanstalk:hostmanager
Monitoring	LogPublicationControl <input type="checkbox"/>
Resources	aws:elasticbeanstalk:managedactions
AWS X-Ray	ManagedActionsEnabled <input type="checkbox"/>
Server	PreferredStartTime <input type="text"/>
Notifications	aws:elasticbeanstalk:managedactions:platformupdate
Container	InstanceRefreshEnabled <input type="checkbox"/>
Advanced	UpdateLevel <input type="text"/>
Logs	

Configurar contêineres do .NET Core usando o AWS Toolkit for Visual Studio.

O painel Container (Contêiner) permite especificar variáveis de ambiente que podem ser lidas a partir do código da aplicação.



These properties are passed into the application as environment variables.

AWS_ACCESS_KEY_ID:	<input type="text"/>
AWS_SECRET_KEY_ID:	<input type="text"/>
PARAM1:	<input type="text"/>
PARAM2:	<input type="text"/>
PARAM3:	<input type="text"/>
PARAM4:	<input type="text"/>
PARAM5:	<input type="text"/>

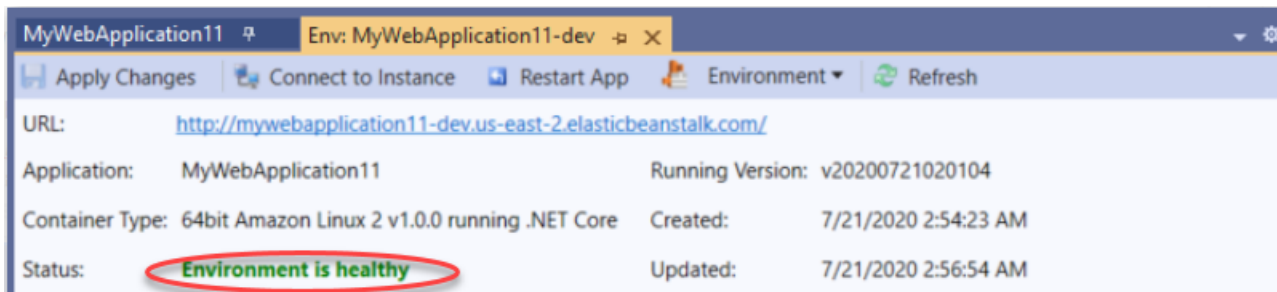
Monitorar a integridade do aplicativo

É importante saber que seu site de produção está disponível e respondendo a solicitações. O Elastic Beanstalk oferece recursos para ajudar você a monitorar a capacidade de resposta da aplicação. Ele monitora as estatísticas sobre a aplicação e alerta quando os limites são excedidos.

Para obter informações sobre o monitoramento de integridade fornecido pelo Elastic Beanstalk, consulte [Relatórios de integridade básica](#).

É possível acessar as informações operacionais sobre sua aplicação usando o AWS Toolkit for Visual Studio ou o Console de Gerenciamento da AWS.

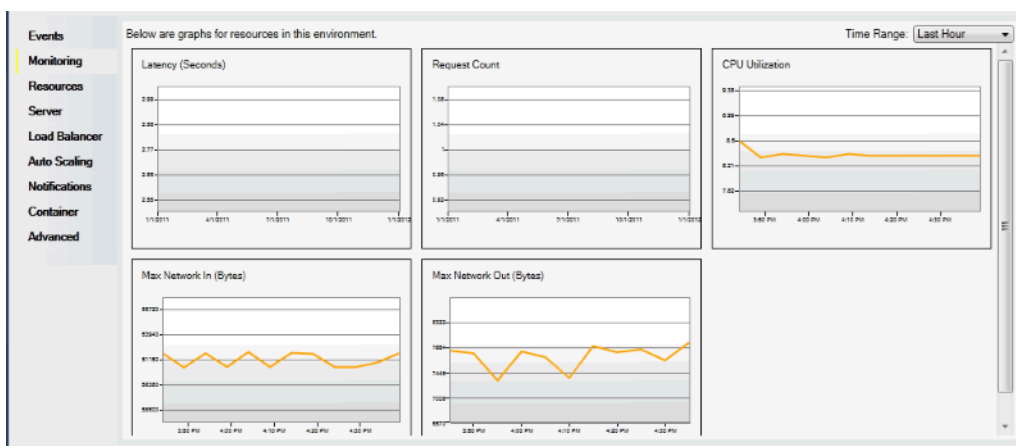
O toolkit exibe o status do ambiente e a integridade do aplicativo no campo Status.



Para monitorar a integridade do aplicativo

1. No AWS Toolkit for Visual Studio, em AWS Explorer, expanda o nó Elastic Beanstalk e, depois, o nó da aplicação.
2. Abra o menu de contexto (clique com o botão direito do mouse) do ambiente de aplicação e selecione View Status (Visualizar status).
3. Na guia do ambiente de aplicação, selecione Monitoring (Monitoramento).

O painel Monitoramento inclui um conjunto de gráficos que mostra o uso de recursos do seu ambiente de aplicativos específico.



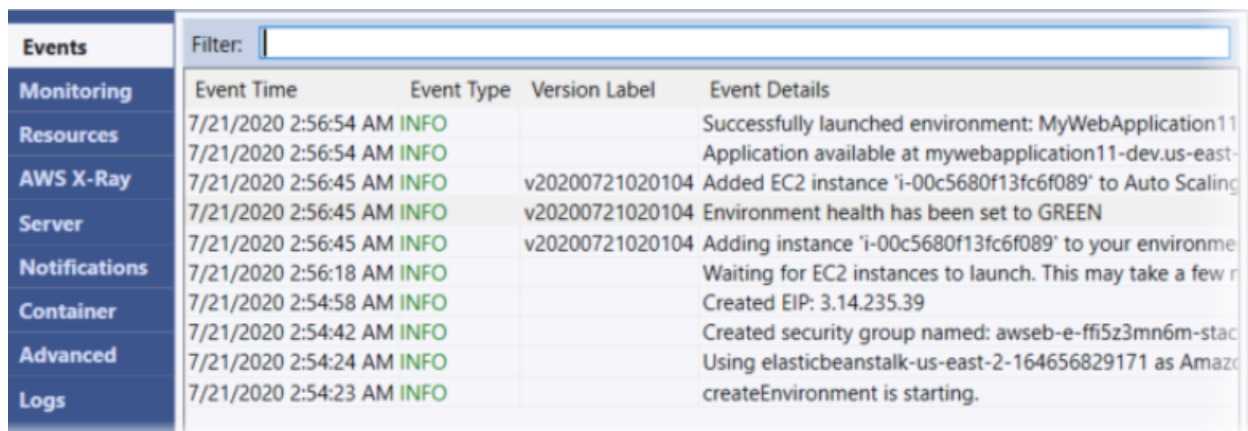
Note

Por padrão, o intervalo de tempo é definido como a última hora. Para modificar essa configuração, na lista Time Range (Período), selecione um período diferente.

Use o AWS Toolkit for Visual Studio ou o Console de Gerenciamento da AWS para visualizar eventos associados a sua aplicação.

Para visualizar eventos do aplicativo

1. No AWS Toolkit for Visual Studio, em AWS Explorer, expanda o nó Elastic Beanstalk e o nó da aplicação.
2. Abra o menu de contexto (clique com o botão direito do mouse) do ambiente de aplicação e selecione View Status (Visualizar status).
3. Na guia do ambiente de aplicação, selecione Events (Eventos).



Events	Filter:
Monitoring	Event Time Event Type Version Label Event Details
Resources	7/21/2020 2:56:54 AM INFO Successfully launched environment: MyWebApplication11
AWS X-Ray	7/21/2020 2:56:54 AM INFO Application available at mywebapplication11-dev.us-east-
Server	7/21/2020 2:56:45 AM INFO v20200721020104 Added EC2 instance 'i-00c5680f13fc6f089' to Auto Scaling
Notifications	7/21/2020 2:56:45 AM INFO v20200721020104 Environment health has been set to GREEN
Container	7/21/2020 2:56:45 AM INFO v20200721020104 Adding instance 'i-00c5680f13fc6f089' to your environme
Advanced	7/21/2020 2:56:18 AM INFO Waiting for EC2 instances to launch. This may take a few m
Logs	7/21/2020 2:54:58 AM INFO Created EIP: 3.14.235.39
	7/21/2020 2:54:42 AM INFO Created security group named: awseb-e-ffi5z3mn6m-stac
	7/21/2020 2:54:24 AM INFO Using elasticbeanstalk-us-east-2-164656829171 as Amaz
	7/21/2020 2:54:23 AM INFO createEnvironment is starting.

Migrar de plataformas .NET no Windows Server para a plataforma .NET Core no Linux

É possível migrar aplicações executadas em plataformas [.NET no Windows Server](#) para as plataformas .NET no Linux. Veja a seguir algumas considerações ao migrar de plataformas Windows para Linux.

Considerações sobre a migração para a plataforma .NET Core no Linux

Área	Alterações e informações
Configuração do aplicativo	Em plataformas Windows, use um manifesto de implantação para especificar os aplicativos executados em seu ambiente. As plataformas .NET Core no Linux usam um Procfile para especificar as aplicações executadas nas instâncias do seu ambiente. Para obter detalhes sobre o empacotamento de aplicativos, consulte the section called “Empacotar aplicativos” .
Servidor de proxy	Nas plataformas Windows, use o IIS como servidor de proxy do aplicativo. As plataformas .NET Core no Linux incluem nginx como um proxy reverso por padrão. É possível escolher não usar nenhum servidor de proxy e usar o Kestrel como servidor Web do aplicativo. Para saber mais, consulte the section called “Servidor de proxy” .
Roteamento	Em plataformas Windows, use o IIS no código do aplicativo e inclua um manifesto de implantação para configurar o caminho do IIS. Para a plataforma .NET Core no Linux, use o roteamento ASP .NET Core no código da aplicação e atualize a configuração nginx do ambiente. Para saber mais, consulte the section called “Servidor de proxy” .
Logs	As plataformas Linux e Windows fazem streaming de logs diferentes. Para obter mais detalhes, consulte the section called “Como o Elastic Beanstalk configura o CloudWatch Logs” .

Criar e implantar aplicações .NET no Elastic Beanstalk

Confira o.NET no AWS Developer Center

Você já visitou nosso Net Developer Center? É o nosso balcão único para todas as coisas que estão disponíveis no domínio.NET AWS.

Para obter mais informações, consulte [o.NET no AWS Developer Center](#).

AWS Elastic Beanstalk para o.NET facilita a implantação, o gerenciamento e a escalabilidade de seus aplicativos web ASP.NET que usam a Amazon Web Services. O Elastic Beanstalk para .NET

está disponível para qualquer pessoa que esteja desenvolvendo ou hospedando uma aplicação Web que usa o IIS.

Get started now (Comece a usar agora): para começar com um tutorial, você pode ir diretamente para [Tutorial: Implantação de um aplicativo ASP.NET Core com o Elastic Beanstalk](#). Neste tutorial, você implantará um exemplo de aplicativo Web ASP.NET em um AWS Elastic Beanstalk aplicativo.

O restante desta seção apresenta instruções para criar, testar, implantar e reimplantar a aplicação Web ASP.NET no Elastic Beanstalk. Alguns exemplos demonstram o uso do AWS Toolkit for Visual Studio, [the section called “A AWS Toolkit for Visual Studio”](#) e a subseção explica como gerenciar e configurar seus aplicativos e ambientes usando o kit de ferramentas. Para obter mais informações sobre pré-requisitos, instruções de instalação e execução de códigos demonstrativos, acesse o [AWS Toolkit for Microsoft Visual Studio](#). Esse site também fornece informações úteis sobre ferramentas, tópicos de como agir e recursos adicionais para desenvolvedores de ASP.NET.

Note

Esta plataforma não é compatível com os seguintes recursos do Elastic Beanstalk:

- Ambientes de operador. Para obter mais detalhes, consulte [Ambientes de operador do Elastic Beanstalk](#).
- Logs em pacote. Para obter mais detalhes, consulte [Visualizar logs da instância](#).

Além disso, versões de plataforma anteriores à v2.0.0 não oferecem suporte a relatórios de integridade aprimorados, atualizações de plataforma gerenciada, atualizações imutáveis, implantações imutáveis e implantações contínuas com um lote adicional.

Os tópicos neste capítulo pressupõem que você tem um nível básico de conhecimento sobre ambientes do Elastic Beanstalk. Se você não tiver usado o Elastic Beanstalk antes, tente o [tutorial de conceitos básicos](#) para saber os princípios.

IMPORTANTE: Compatibilidade com TLS 1.2

Important

Compatibilidade com TLS 1.2

Em 31 de dezembro de 2023, AWS começou a aplicar totalmente o TLS 1.2 em todos os endpoints AWS da API. Isso removeu a capacidade de usar as versões 1.0 e 1.1 do TLS com todas as AWS APIs. Este procedimento foi originalmente comunicado em [28 de junho de 2022](#). Para evitar o risco de impacto na disponibilidade, atualize suas versões da plataforma para uma versão mais nova o mais rápido possível.

Impacto potencial

As versões das plataformas Elastic Beanstalk que executam o TLS v1.1 ou anterior serão afetadas. Essa alteração afetará ações do ambiente que incluem, entre outras: implantações de configuração, implantações de aplicações, ajuste de escala automático, lançamento de novo ambiente, alternância de logs, relatórios de integridade aprimorados e publicação de logs de aplicações no bucket do Amazon S3 associado a suas aplicações.

Versões da plataforma Windows afetadas

Os clientes com ambientes Elastic Beanstalk na seguinte versão de plataforma são aconselhados a atualizar cada um de seus ambientes correspondentes para a plataforma Windows versão 2.8.3 ou posterior, lançada em [18 de fevereiro de 2022](#).

- Windows Server 2019 - versão da plataforma 2.8.2 ou anteriores

Os clientes com ambientes Elastic Beanstalk nas seguintes versões de plataforma são aconselhados a atualizar cada um de seus ambientes correspondentes para a plataforma Windows versão 2.10.7 ou posterior, lançada em [28 de dezembro de 2022](#).

- Windows Server 2016 - versão da plataforma 2.10.6 ou versões anteriores
- Windows Server 2012 — todas as versões da plataforma; essa plataforma foi retirada em [4 de dezembro de 2023](#)
- Windows Server 2008 — todas as versões da plataforma; essa plataforma foi desativada em [28 de outubro de 2019](#)

Para obter a lista das versões da plataforma Windows Server mais recentes e compatíveis, consulte [Plataformas compatíveis](#) no guia Plataformas do AWS Elastic Beanstalk .

Para obter detalhes e melhores práticas sobre como atualizar seu ambiente, consulte [Atualizar a versão de plataforma do ambiente Elastic Beanstalk](#).

Tópicos

- [Introdução ao Windos.NET no Elastic Beanstalk](#)
- [Configurar seu ambiente de desenvolvimento .NET](#)
- [Usar a plataforma .NET do Elastic Beanstalk](#)
- [Tutorial: Implantação de um aplicativo ASP.NET Core com o Elastic Beanstalk](#)
- [Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicações .NET](#)
- [A AWS Toolkit for Visual Studio](#)
- [Migrar a aplicação .NET no local para o Elastic Beanstalk](#)
- [Recursos](#)

Introdução ao Windos.NET no Elastic Beanstalk

Para começar a usar aplicativos.NET AWS Elastic Beanstalk, você só precisa de um [pacote de origem](#) de aplicativo para carregar como sua primeira versão do aplicativo e implantá-lo em um ambiente. Quando você cria um ambiente, o Elastic Beanstalk aloca todos os recursos necessários para executar uma aplicação AWS web altamente escalável.

Iniciando um ambiente com um aplicativo Windows.NET de amostra

O Elastic Beanstalk fornece aplicativos de amostra de página única para cada plataforma.

Amostras

Nome	Configurações compatíveis	Tipo de ambiente	Origem	Descrição
.NET padrão	ERA 2022 Núcleo do servidor WS 2022	Serviço Web	dotnet-asp-windows.zip	Aplicativo de amostra do ASP.NET Core com uma única página configurada para ser exibida na raiz do site.

Nome	Configurações compatíveis	Tipo de ambiente	Origem	Descrição
	FOI 2019			
	Núcleo do servidor WS 2019			
	FOI EM 2016			
	Núcleo do servidor WS 2016			

Aposentadoria das filiais da plataforma Windows Server 2012 R2

As seguintes ramificações da plataforma Elastic Beanstalk foram descontinuadas:

- Windows Server 2012 R2 executando o IIS 8.5
- Windows Server Core 2012 R2 executando o IIS 8.5

Se você atualmente usa essas ramificações de plataforma desativadas, é altamente recomendável que comece a planejar sua migração para uma das seguintes ramificações da plataforma Windows Server, que são atuais e totalmente suportadas:

- Windows Server 2022 com IIS 10.0 versão 2.x
- Windows Server 2019 com IIS 10.0 versão 2.x

Veja considerações completas sobre migração em [Migrar de versões principais anteriores da plataforma do Windows Server](#).

Para obter mais informações sobre a substituição da plataforma, consulte [Política de suporte às plataformas do Elastic Beanstalk](#).

Note

Após a descontinuação, as ramificações da plataforma Windows Server 2012 não estão mais disponíveis no console do Elastic Beanstalk. Você pode continuar operando seus ambientes

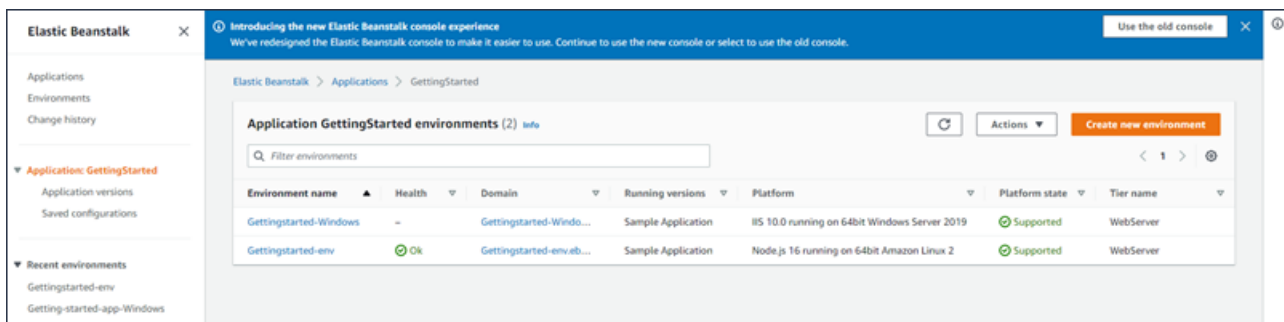
existentes baseados nessas ramificações desativadas da plataforma até 4 de março de 2024, ou seja, 90 dias após a data de desativação de 4 de dezembro.

O Elastic Beanstalk tornará privadas as AMIs do Windows 2012 do Beanstalk após 4 de março de 2024. Essa ação impedirá a inicialização de instâncias nos ambientes do Windows 2012 que usam a AMI padrão do Beanstalk. Para reter o acesso às AMIs, você pode copiá-las em suas contas para usá-las nos ambientes do Beanstalk. Para obter instruções detalhadas, consulte [Preservar o acesso a uma imagem de máquina da Amazon \(AMI\) para uma plataforma descontinuada](#).

Faça download de qualquer uma das aplicações demonstrativas e implante-a no Elastic Beanstalk usando o procedimento a seguir.

Para iniciar um ambiente com um aplicativo de exemplo (console)

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, escolha Aplicativos e o nome de um aplicativo existente na lista ou [crie um](#).
3. Na página de visão geral da aplicação, escolha Create a new environment (Criar um novo ambiente).



Isso inicia o assistente Create environment (Criar ambiente). O assistente fornece um conjunto de etapas para a criação de um novo ambiente.

Step 1
Configure environment

Step 2
Configure service access

Step 3 - optional
Configure instance traffic and scaling

Step 4 - optional
Set up networking, database, and tags

Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

Configure environment [Info](#)

Environment tier [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- Web server environment**
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
- Worker environment**
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information [Info](#)

Application name
GettingStarted
Maximum length of 100 characters.

▶ **Application tags (optional)**

Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name
GettingStarted-env
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain name
 .us-east-1.elasticbeanstalk.com

Environment description

Platform [Info](#)

Platform type

- Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)
- Custom platform**
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform


Platform branch

Platform version

Application code [Info](#)


- Sample application**
- Existing version**
Application versions that you have uploaded.
- Upload your code**
Upload a source bundle from your computer or copy one from Amazon S3.

4. Para nível de ambiente, escolha o [nível de ambiente](#) Web server environment (Ambiente do servidor Web) ou Worker environment (Ambiente do operador). Você não pode alterar o nível de um ambiente após a criação.

 Note

O [.NET na Plataforma de servidor Windows](#) não tem suporte para a camada de ambiente do operador.

5. Em Plataforma, selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.

 Note

O Elastic Beanstalk é compatível com várias [versões](#) para a maioria das plataformas listadas. Por padrão, o console seleciona a versão recomendada da plataforma e a ramificação da plataforma escolhida. Se a aplicação exigir outra versão, você poderá selecioná-la aqui. Para obter informações sobre versões de plataforma compatíveis, consulte [the section called “Plataformas compatíveis”](#).

6. Para Application code, escolha Sample application.
7. Em Configuration presets (Predefinições de configuração), escolha Single instance (Instância única).
8. Escolha Próximo.
9. A página Configurar acesso ao serviço é exibida.

Configure service access [Info](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role
 Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

[View permission details](#)

Cancel Skip to review Previous **Next**

10. Escolha Usar um perfil de serviço existente em Perfil de serviço.

11. Em seguida, vamos nos concentrar na lista suspensa Perfil de instância do EC2. Os valores exibidos nessa lista suspensa podem variar, dependendo de a conta ter criado anteriormente um novo ambiente.

Escolha uma das opções a seguir com base nos valores exibidos na lista.

- Se `aws-elasticbeanstalk-ec2-role` for exibido na lista suspensa, selecione-o na lista suspensa Perfil de instância do EC2.
- Se outro valor for exibido na lista e for o perfil padrão de instância do EC2 para seus ambientes, selecione-o na lista suspensa Perfil de instância do EC2.
- Se a lista suspensa Perfil de instância do EC2 não mostrar nenhum valor para seleção, expanda o procedimento que se segue, Criar um perfil do IAM para ser o perfil de instância do EC2.

Conclua as etapas em Criar perfil do IAM para ser o perfil de instância do EC2 para criar um perfil do IAM que você possa selecionar posteriormente para ser o perfil de instância do EC2. Depois, retorne para esta etapa.

Agora que você criou um perfil do IAM e atualizou a lista, ele será exibido como uma opção na lista suspensa. Selecione o perfil do IAM que você acabou de criar na lista suspensa Perfil de instância do EC2.

12. Escolha Skip to Review (Pular para revisão) na página Configure service access (Configurar acesso a serviço).

Isso selecionará os valores padrão para essa etapa e pulará as etapas opcionais.

13. A página Review (Revisão) exibe um resumo de todas as suas escolhas.

Para personalizar ainda mais o ambiente, escolha Edit (Editar) ao lado da etapa que inclui os itens que você deseja configurar. Você pode configurar as opções a seguir somente durante a criação do ambiente:

- Nome do ambiente
- Nome de domínio
- Versão da plataforma
- Processador
- VPC
- Nível

Você pode alterar as configurações a seguir após a criação do ambiente, mas elas exigem que novas instâncias ou outros recursos sejam provisionados, e pode levar muito tempo para que elas sejam aplicadas:

- Tipo de instância, volume raiz, par de chaves e função AWS Identity and Access Management (IAM)
- Banco de dados interno do Amazon RDS
- Load balancer

Para obter detalhes sobre todas as configurações disponíveis, consulte [O assistente de criação de novo ambiente](#).

14. Escolha Submit (enviar) na parte inferior da página para inicializar a criação do novo ambiente.

Criar um perfil do IAM para ser o perfil de instância do EC2

Configure service access [Info](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role

Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

Para criar um perfil do IAM para ser o perfil de instância do EC2

1. Escolha Visualizar detalhes de permissão. Essa opção exibe a lista suspensa Perfil de instância do EC2.

Uma janela modal intitulada Visualizar permissões de perfil de instância é exibida. Essa janela lista os perfis gerenciados que deverão ser anexados ao novo perfil de instância do EC2 que você criar. Ela também fornece um link para iniciar o console do IAM.

2. Escolha o link Console do IAM exibido na parte superior da janela.
3. No painel de navegação do console do IAM, escolha Roles (Perfis).
4. Selecione Criar função.
5. Em Tipo de entidade confiável, selecione Serviço da AWS .
6. Em Use case (Caso de uso), selecione EC2.
7. Escolha Próximo.
8. Anexe as políticas gerenciadas apropriadas. Role a janela modal Visualizar permissões de perfil de instância para ver as políticas gerenciadas. As políticas também estão listadas aqui:

- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

9. Escolha Próximo.
10. Insira um nome para a função.
11. (Opcional) Adicione tags à função.
12. Selecione Criar função.
13. Retorne à janela do console do Elastic Beanstalk que está aberta.
14. Feche a janela modal Visualizar permissões de perfil de instância.

 Important

Não feche a página do navegador que exibe o console do Elastic Beanstalk.

15. Escolha



(atualizar), ao lado da lista suspensa Perfil de instância do EC2.

Essa opção atualiza a lista suspensa para que o perfil que você acabou de criar seja exibido na lista suspensa.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá [implantar uma nova versão](#) do aplicativo ou um aplicativo completamente diferente a qualquer momento. A implantação de uma nova versão da aplicação é rápida, pois não é necessário provisionar ou reiniciar as instâncias do EC2.

Depois que você tiver implantado uma ou duas aplicações de demonstração e estiver pronto para começar a desenvolver localmente, consulte as instruções na [próxima seção](#) para configurar um ambiente de desenvolvimento .NET.

Configurar seu ambiente de desenvolvimento .NET

Configure um ambiente de desenvolvimento .NET para testar sua aplicação localmente, antes de implantá-lo no AWS Elastic Beanstalk. Este tópico descreve as etapas de configuração do ambiente de desenvolvimento e inclui links para páginas de instalação de ferramentas úteis.

Para ver as etapas de configuração comuns e as ferramentas que se aplicam a todas as linguagens, consulte [Configurar sua máquina de desenvolvimento para uso com o Elastic Beanstalk](#).

Seções

- [Instalar um IDE](#)
- [Instalar o AWS Toolkit for Visual Studio](#)

Se você precisa gerenciar os recursos da AWS a partir da sua aplicação, instale o AWS SDK for .NET. Por exemplo, é possível usar o Amazon S3 para armazenar e recuperar dados.

Com o AWS SDK for .NET, você pode começar em minutos com um único pacote completo para fazer download que inclui modelos de projeto do Visual Studio, a biblioteca AWS .NET, exemplos de código em C# e documentação. São fornecidos exemplos práticos em C# de como usar a biblioteca para criar aplicativos. São fornecidos tutoriais de vídeo online e documentação de referência para ajudar você a aprender a usar as bibliotecas e os exemplos de códigos.

Visite a página inicial do [AWS SDK for .NET](#) para obter mais informações e instruções de instalação.

Instalar um IDE

Os Integrated development environments (IDEs – Ambientes de desenvolvimento integrados) oferecem uma ampla variedade de recursos que facilitam o desenvolvimento de aplicativos. Se você nunca usou um IDE para desenvolvimento .NET, experimente o Visual Studio Community para começar.

Visite a página do [Visual Studio Community](#) para fazer download e instalar o Visual Studio Community.

Instalar o AWS Toolkit for Visual Studio

O [AWS Toolkit for Visual Studio](#) é um plugin de código aberto para o Visual Studio IDE que torna mais fácil para os desenvolvedores desenvolverem, depurarem e implantarem aplicações .NET

usando a AWS. Acesse a [página inicial do Toolkit for Visual Studio](#) para obter instruções de instalação.

Usar a plataforma .NET do Elastic Beanstalk

AWS Elastic Beanstalk oferece suporte a várias plataformas para diferentes versões da estrutura de programação .NET e do Windows Server. Consulte [.NET no Windows Server com IIS](#) no documento Plataformas do AWS Elastic Beanstalk para obter uma lista completa.

O Elastic Beanstalk fornece [opções de configuração](#) que podem ser usadas para personalizar o software que é executado nas instâncias do EC2 no ambiente do Elastic Beanstalk. Você também pode configurar as variáveis do ambiente exigidas pela aplicação, permitir a alternância de logs para o Amazon S3 e definir configurações de framework .NET.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para [modificar a configuração de um ambiente em execução](#). Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar [configurações salvas](#) para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir [arquivos de configuração](#). As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console. Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte [Opções de configuração](#).

Configurar o ambiente .NET no console do Elastic Beanstalk

É possível usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3, configurar variáveis que a aplicação pode ler no ambiente e alterar as configurações de framework .NET.

Como configurar seu ambiente .NET no console do Elastic Beanstalk

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Opções de contêiner

- Target .NET runtime (Tempo de execução de .NET de destino): defina como 2.0 para executar CLR v2.
- Enable 32-bit applications (Habilitar aplicações 32 bits): defina como True para executar aplicações de 32 bits.

Opções de log

A seção Log Options tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Enable log file rotation to Amazon S3 (Habilitar a rotação de arquivos de log para o Amazon S3): especifica se os arquivos de log das instâncias do Amazon EC2 da aplicação serão copiados para o bucket do Amazon S3 associado a ela.

Propriedades de ambiente

A seção Environment Properties permite que você especifique definições de configuração do ambiente nas instâncias do Amazon EC2 que estão executando seu aplicativo. Essas definições são passadas para o aplicativo como pares de chave-valor. Use `System.EnvironmentVariable` para leitura. Chaves idênticas podem existir em `web.config` e como propriedades do ambiente. Use o namespace `System.Configuration` para ler valores de `web.config`.

```
NameValueCollection appConfig = ConfigurationManager.AppSettings;  
string endpoint = appConfig["API_ENDPOINT"];
```

Consulte [Propriedades de ambientes e outras configurações de software](#) para obter mais informações.

O namespace `aws:elasticbeanstalk:container:dotnet:appool`

Você pode usar um [arquivo de configuração](#) para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. As opções de configuração podem ser definidas pelo serviço Elastic Beanstalk ou pela plataforma utilizada, e são organizadas em namespaces.

A plataforma .NET define opções no namespace

`aws:elasticbeanstalk:container:dotnet:appool` que você pode usar para configurar o tempo de execução do .NET.

O arquivo de configuração de exemplo a seguir mostra as configurações de cada uma das opções disponíveis neste namespace:

Example `.ebextensions/dotnet-settings.config`

```
option_settings:
  aws:elasticbeanstalk:container:dotnet:appool:
    Target Runtime: 2.0
    Enable 32-bit Applications: True
```

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte [Opções de configuração](#) Para mais informações.

Migrar entre versões principais da plataforma do Windows Server do Elastic Beanstalk

AWS Elastic BeanstalkO tem várias versões principais da plataforma do Windows Server. Esta página aborda as principais melhorias para cada versão principal e o que deve ser considerado antes de migrar para uma versão mais recente.

A plataforma do Windows Server está na versão 2 (v2) no momento. Se o aplicativo usar qualquer versão de plataforma do Windows Server anterior à v2, recomendamos migrar para a v2.

Novidades das versões principais da plataforma do Windows Server

Plataforma do Windows Server V2

A versão 2 (v2) da plataforma Windows Server do Elastic Beanstalk foi [lançada em fevereiro de 2019](#). A V2 aproxima o comportamento da plataforma Windows Server das plataformas Linux do Elastic Beanstalk de várias maneiras importantes. A V2 é totalmente compatível com a v1, o que facilita a migração da v1.

Agora, a plataforma do Windows Server oferece suporte a:

- Versionamento: cada versão obtém um novo número de versão, e você pode fazer referência às versões anteriores (que ainda estão disponíveis para você) ao criar e gerenciar ambientes.
- Integridade aprimorada: para obter detalhes, consulte [Monitoramento e relatório de integridade aprimorada](#).
- Implantações imutáveis e contínuas com um lote adicional: para obter detalhes sobre as políticas de implantação, consulte [Implantar aplicações em ambientes do Elastic Beanstalk](#).
- Atualizações imutáveis: para obter detalhes sobre os tipos de atualização, consulte [Alterações de configuração](#).
- Atualizações gerenciadas de plataforma: para obter detalhes, consulte [Atualizações gerenciadas de plataforma](#).

Note

Os novos recursos de implantação e atualização dependem da integridade aprimorada. Habilite a integridade aprimorada para usá-los. Para obter mais detalhes, consulte [Habilitar relatórios de integridade aprimorada do Elastic Beanstalk](#).

Plataforma do Windows Server V1

A versão 1.0.0 (v1) da plataforma Windows Server do Elastic Beanstalk foi lançada em outubro de 2015. Essa versão altera a ordem na qual o Elastic Beanstalk processa comandos em [arquivos de configuração](#) durante a criação e as atualizações do ambiente.

As versões de plataformas anteriores não têm um número de versão no nome da pilha de soluções:

- Windows Server 2012 R2 de 64 bits executando IIS 8.5

- Windows Server Core 2012 R2 de 64 bits executando IIS 8.5
- Windows Server 2012 de 64 bits executando IIS 8
- Windows Server 2008 R2 de 64 bits executando IIS 7.5

Em versões anteriores, a ordem de processamento dos arquivos de configuração é inconsistente. Durante a criação do ambiente, `Container Commands` executam depois que o código-fonte do aplicativo é implantado no IIS. Durante uma implantação em um ambiente em execução, os comandos de contêiner executam antes da implantação da nova versão. Durante uma expansão, os arquivos de configuração não são processados.

Além disso, o IIS inicia antes da execução dos comandos de contêiner. Esse comportamento levou alguns clientes a implementar soluções em comandos de contêiner, pausando o servidor IIS antes da execução dos comandos e reiniciando-o após a conclusão.

A versão 1 corrige a inconsistência e aproxima o comportamento da plataforma Windows Server ao das plataformas baseadas em Linux do Elastic Beanstalk. Na plataforma v1, o Elastic Beanstalk sempre executa comandos de contêiner antes de iniciar o servidor IIS.

As pilhas de soluções da plataforma v1 têm um v1 após a versão do Windows Server:

- Windows Server 2012 R2 v1.1.0 de 64 bits executando IIS 8.5
- Windows Server Core 2012 R2 v1.1.0 de 64 bits executando IIS 8.5
- Windows Server 2012 v1.1.0 de 64 bits executando IIS 8
- Windows Server 2008 R2 v1.1.0 de 64 bits executando IIS 7.5

Além disso, a plataforma v1 extrai o conteúdo do pacote de origem do aplicativo em `C:\staging` antes da execução de comandos de contêiner. Após a conclusão dos comandos de contêiner, o conteúdo dessa pasta é compactado em um arquivo `.zip` e implantado no IIS. Esse fluxo de trabalho permite que você modifique o conteúdo do pacote de origem do aplicativo com comandos ou com um script antes da implantação.

Migrar de versões principais anteriores da plataforma do Windows Server

Leia esta seção sobre considerações de migração antes de atualizar seu ambiente. Para atualizar a plataforma de seu ambiente para uma versão mais recente, visite [Atualizar a versão de plataforma do ambiente Elastic Beanstalk](#).

De V1 para V2

A plataforma v2 do Windows Server não oferece suporte a .NET Core 1.x e 2.0. Se você está migrando o aplicativo do Windows Server v1 para v2 e o aplicativo usa uma dessas versões do .NET Core, atualize o aplicativo para uma versão do .NET Core que ofereça suporte a v2. Para obter uma lista de versões compatíveis, consulte [.NET no Windows Server com IIS](#) nas Plataformas do AWS Elastic Beanstalk.

Se o aplicativo usar uma imagem de máquina da Amazon (AMI), crie uma nova AMI personalizada com base em uma AMI da plataforma Windows Server v2. Para saber mais, consulte [Usar uma imagem de máquina da Amazon \(AMI\) personalizada](#).

Note

Os recursos de implantação e atualização que são novos no Windows Server v2 dependem da integridade aprimorada. Ao migrar um ambiente para v2, a integridade aprimorada é desabilitada. Habilite-a para usar esses recursos. Para obter mais detalhes, consulte [Habilitar relatórios de integridade aprimorada do Elastic Beanstalk](#).

Da pré-V1

Além das considerações sobre a migração de v1, se você estiver migrando o aplicativo de uma pilha de soluções do Windows Server anterior à v1 e estiver usando comandos de contêiner, remova todos os comandos adicionados para resolver as inconsistências de processamento ao migrar para uma versão mais recente. A partir da v1, é garantido que os comandos de contêiner sejam executados completamente antes da origem do aplicativo implantado e antes do início do IIS. Isso permite que você faça qualquer alteração na origem em C:\staging e modifique os arquivos de configuração do IIS durante esta etapa sem problemas.

Por exemplo, você pode usar a AWS CLI para fazer download de um arquivo DLL na fonte da aplicação do Amazon S3:

```
.ebextensions\copy-dll.config
```

```
container_commands:
  copy-dll:
    command: aws s3 cp s3://DOC-EXAMPLE-BUCKET/dlls/large-dll.dll .\lib\
```

Para obter mais informações sobre o uso de arquivos de configuração, consulte [Personalização avançada de ambiente com arquivos de configuração \(.ebextensions\)](#).

Executar vários aplicativos e aplicativos ASP.NET Core com um manifesto de implantação

É possível usar um manifesto de implantação para informar o Elastic Beanstalk como implantar a aplicação. Ao usar esse método, você não precisa usar MSDeploy para gerar um pacote de origem para uma única aplicação ASP.NET que é executada no caminho raiz do seu site. Em vez disso, você pode usar um arquivo de manifesto para executar várias aplicações em caminhos diferentes. Ou, como alternativa, você pode dizer ao Elastic Beanstalk para implantar e executar a aplicação com o ASP.NET Core. Também é possível usar um manifesto de implantação para configurar um grupo de aplicações para executar suas aplicações.

Os manifestos de implantação adicionam suporte para [aplicações .NET Core](#) ao Elastic Beanstalk. Você pode implantar uma aplicação .NET Framework sem um manifesto de implantação. No entanto, as aplicações .NET Core exigem um manifesto de implantação para execução no Elastic Beanstalk. Ao usar um manifesto de implantação, você cria um arquivo do site para cada aplicação e, em seguida, empacota os arquivos do site em um segundo arquivo ZIP que contém o manifesto de implantação.

Os manifestos de implantação também adicionam a capacidade de [executar vários aplicativos em diferentes caminhos](#). Um manifesto de implantação define um conjunto de alvos de implantação, cada um com um arquivamento do site e um caminho em que o IIS deve executá-lo. Por exemplo, você pode executar uma API da Web no caminho /api para atender a solicitações assíncronas e um aplicativo web no caminho raiz que consome a API.

Você também pode usar um manifesto de implantação para [executar várias aplicações usando grupos de aplicações no IIS ou no Kestrel](#). Você pode configurar um grupo de aplicativos para reiniciar seus aplicativos periodicamente, executar aplicativos de 32 bits ou usar uma versão específica do tempo de execução do .NET Framework.

Para uma personalização completa, você pode [escrever seus próprios scripts de implantação](#) no Windows PowerShell e informar ao Elastic Beanstalk quais scripts executar para instalar, desinstalar e reiniciar seu aplicativo.

Os manifestos de implantação e os recursos relacionados exigem uma plataforma do Windows Server [versão 1.2.0 ou posterior](#).

Seções

- [Aplicativos .NET Core](#)
- [Executar vários aplicativos](#)
- [Configurar grupos de aplicativos](#)
- [Definir implantações personalizadas](#)

Aplicativos .NET Core

É possível usar um manifesto de implantação para executar aplicações .NET Core no Elastic Beanstalk. .NET Core é uma versão multiplataforma do .NET que é fornecida com uma ferramenta de linha de comando (`dotnet`). Você pode usá-lo para gerar uma aplicação, executá-lo localmente e prepará-lo para publicação.

Note

Consulte [Tutorial: Implantação de um aplicativo ASP.NET Core com o Elastic Beanstalk](#) para ver um tutorial e uma aplicação demonstrativa que usa um manifesto de implantação para executar uma aplicação .NET Core no Elastic Beanstalk.

Para executar uma aplicação .NET Core no Elastic Beanstalk, execute `dotnet publish` e empacote a saída em um arquivo ZIP, sem incluir os diretórios. Coloque o arquivo do site em um pacote de origem com um manifesto de implantação com um destino de implantação do tipo `aspNetCoreWeb`.

O seguinte manifesto de implantação executa um aplicativo .NET Core em um arquivo de site chamado `dotnet-core-app.zip` no caminho raiz.

Example `aws-windows-deployment-manifest.json` - Núcleo.NET

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "my-dotnet-core-app",
        "parameters": {
          "archive": "dotnet-core-app.zip",
          "iisPath": "/"
        }
      }
    ]
  }
}
```

```
    }  
  }  
]  
}  
}
```

Inclua o arquivo de manifesto e de site em um arquivo ZIP para criar um pacote de origem.

Example dotnet-core-bundle.zip

```
.  
|-- aws-windows-deployment-manifest.json  
`-- dotnet-core-app.zip
```

O arquivo do site contém o código do aplicativo compilado, as dependências e o arquivo `web.config`.

Example dotnet-core-app.zip

```
.  
|-- Microsoft.AspNetCore.Hosting.Abstractions.dll  
|-- Microsoft.AspNetCore.Hosting.Server.Abstractions.dll  
|-- Microsoft.AspNetCore.Hosting.dll  
|-- Microsoft.AspNetCore.Http.Abstractions.dll  
|-- Microsoft.AspNetCore.Http.Extensions.dll  
|-- Microsoft.AspNetCore.Http.Features.dll  
|-- Microsoft.AspNetCore.Http.dll  
|-- Microsoft.AspNetCore.HttpOverrides.dll  
|-- Microsoft.AspNetCore.Server.IISIntegration.dll  
|-- Microsoft.AspNetCore.Server.Kestrel.dll  
|-- Microsoft.AspNetCore.WebUtilities.dll  
|-- Microsoft.Extensions.Configuration.Abstractions.dll  
|-- Microsoft.Extensions.Configuration.EnvironmentVariables.dll  
|-- Microsoft.Extensions.Configuration.dll  
|-- Microsoft.Extensions.DependencyInjection.Abstractions.dll  
|-- Microsoft.Extensions.DependencyInjection.dll  
|-- Microsoft.Extensions.FileProviders.Abstractions.dll  
|-- Microsoft.Extensions.FileProviders.Physical.dll  
|-- Microsoft.Extensions.FileSystemGlobbing.dll  
|-- Microsoft.Extensions.Logging.Abstractions.dll  
|-- Microsoft.Extensions.Logging.dll  
|-- Microsoft.Extensions.ObjectPool.dll
```

```
|-- Microsoft.Extensions.Options.dll
|-- Microsoft.Extensions.PlatformAbstractions.dll
|-- Microsoft.Extensions.Primitives.dll
|-- Microsoft.Net.Http.Headers.dll
|-- System.Diagnostics.Contracts.dll
|-- System.Net.WebSockets.dll
|-- System.Text.Encodings.Web.dll
|-- dotnet-core-app.deps.json
|-- dotnet-core-app.dll
|-- dotnet-core-app.pdb
|-- dotnet-core-app.runtimeconfig.json
`-- web.config
```

Consulte [o tutorial](#) para obter um exemplo completo.

Executar vários aplicativos

Você pode executar vários aplicativos com um manifesto de implantação definindo vários alvos de implantação.

O seguinte manifesto de implantação configura dois aplicativos .NET Core. O `WebApiSampleApp` aplicativo implementa uma API web simples e atende solicitações assíncronas no caminho `/api`. O aplicativo `DotNetSampleApp` é um aplicativo web que atende solicitações no caminho raiz.

Example `aws-windows-deployment-manifest.json` - vários aplicativos

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "WebAPISample",
        "parameters": {
          "appBundle": "WebApiSampleApp.zip",
          "iisPath": "/api"
        }
      },
      {
        "name": "DotNetSample",
        "parameters": {
          "appBundle": "DotNetSampleApp.zip",
          "iisPath": "/"
        }
      }
    ]
  }
}
```

```
    }  
  ]  
}  
}
```

Um aplicativo de exemplo com vários aplicativos está disponível em:

- Pacote de origem implantável - [-v2.zip dotnet-multiapp-sample-bundle](#)
- Código fonte - [dotnet-multiapp-sample-source-v2.zip](#)

Configurar grupos de aplicativos

Você pode oferecer suporte a várias aplicações no seu ambiente Windows. Duas abordagens estão disponíveis:

- Você pode usar o modelo de out-of-process hospedagem com o servidor web Kestrel. Com esse modelo, você configura várias aplicações para serem executadas em um grupo de aplicações.
- Você pode usar o modelo de hospedagem no processo. Com esse modelo, você usa vários grupos de aplicações para executar várias aplicações com apenas uma aplicação em cada grupo. Se você estiver usando o servidor IIS e precisar executar várias aplicações, deverá usar essa abordagem.

Para configurar o Kestrel para executar várias aplicações em um grupo de aplicações, adicione `hostingModel="OutOfProcess"` ao arquivo `web.config`. Considere os seguintes exemplos:

Example `web.config` - para o modelo de hospedagem Kestrel out-of-process

```
<configuration>  
<location path="." inheritInChildApplications="false">  
<system.webServer>  
<handlers>  
<add  
  name="aspNetCore"  
  path="*" verb="*"   
  modules="AspNetCoreModuleV2"  
  resourceType="Unspecified" />  
</handlers>  
<aspNetCore  
  processPath="dotnet"  
  arguments=".\CoreWebApp-5-0.dll"
```

```
    stdoutLogEnabled="false"
    stdoutLogFile=".\logs\stdout"
    hostingModel="OutOfProcess" />
</system.webServer>
</location>
</configuration>
```

Example aws-windows-deployment-manifest.json - vários aplicativos

```
{
  "manifestVersion": 1,
  "deployments": {"msDeploy": [
    {"name": "Web-app1",
      "parameters": {"archive": "site1.zip",
        "iisPath": "/"
      }
    },
    {"name": "Web-app2",
      "parameters": {"archive": "site2.zip",
        "iisPath": "/app2"
      }
    }
  ]
}
```

O IIS não suporta várias aplicações em um grupo de aplicações porque ele usa o modelo de hospedagem no processo. Portanto, você precisa configurar várias aplicações atribuindo cada aplicação a um grupo de aplicações. Em outras palavras, atribua apenas uma aplicação a um grupo de aplicações.

Você pode configurar o IIS para usar grupos de aplicações diferentes no arquivo `aws-windows-deployment-manifest.json`. Faça as seguintes atualizações conforme você se refere ao próximo arquivo de exemplo:

- Adicione uma `iisConfig` seção que inclua uma subseção chamada `appPools`.
- No bloco `appPools`, liste os grupos de aplicações.
- Na seção `deployments`, defina uma seção `parameters` para cada aplicação.
- Para cada aplicação, a seção `parameters` especifica um arquivo, um caminho para executá-la e um `appPool` no qual executá-la.

O manifesto de implantação a seguir configura dois grupos de aplicações que reiniciam sua aplicação a cada 10 minutos. Eles também anexam suas aplicações a uma aplicação web .NET Framework que é executada no caminho especificado.

Example aws-windows-deployment-manifest.json - um aplicativo por pool de aplicativos

```
{
  "manifestVersion": 1,
  "iisConfig": {"appPools": [
    {"name": "MyFirstPool",
      "recycling": {"regularTimeInterval": 10}
    },
    {"name": "MySecondPool",
      "recycling": {"regularTimeInterval": 10}
    }
  ]
},
  "deployments": {"msDeploy": [
    {"name": "Web-app1",
      "parameters": {
        "archive": "site1.zip",
        "iisPath": "/",
        "appPool": "MyFirstPool"
      }
    },
    {"name": "Web-app2",
      "parameters": {
        "archive": "site2.zip",
        "iisPath": "/app2",
        "appPool": "MySecondPool"
      }
    }
  ]
}
```

Definir implantações personalizadas

Para controle ainda maior, você pode personalizar totalmente a implantação de um aplicativo definindo uma implantação personalizada.

O manifesto de implantação a seguir informa ao Elastic Beanstalk para executar um script `install` denominado `siteInstall.ps1`. Esse script instala o site durante a execução e as implantações da

instância. Além disso, o manifesto de implantação também instrui o Elastic Beanstalk a `uninstall` executar um script antes de instalar uma nova versão durante a implantação `restart` e um script para reiniciar o aplicativo quando [você escolher Reiniciar o App Server no console de gerenciamento](#). AWS

Example `aws-windows-deployment-manifest.json` - implantação personalizada

```
{
  "manifestVersion": 1,
  "deployments": {
    "custom": [
      {
        "name": "Custom site",
        "scripts": {
          "install": {
            "file": "siteInstall.ps1"
          },
          "restart": {
            "file": "siteRestart.ps1"
          },
          "uninstall": {
            "file": "siteUninstall.ps1"
          }
        }
      }
    ]
  }
}
```

Inclua todos os artefatos necessários para executar o aplicativo em seu pacote de origem com o manifesto e os scripts.

Example `Custom-site-bundle .zip`

```
.
|-- aws-windows-deployment-manifest.json
|-- siteInstall.ps1
|-- siteRestart.ps1
|-- siteUninstall.ps1
`-- site-content.zip
```

Tutorial: Implantação de um aplicativo ASP.NET Core com o Elastic Beanstalk

Neste tutorial, você percorrerá o processo de criação de um novo aplicativo ASP.NET Core e sua implantação no AWS Elastic Beanstalk.

Primeiramente, você usará a ferramenta de linha de comando `dotnet` do .NET Core SDK para gerar um aplicativo de linha de comando do .NET Core básico, instalar dependências, compilar código e executar aplicativos localmente. Em seguida, você modificará a classe `Program.cs` padrão e adicionará uma classe `Startup.cs` do ASP.NET e arquivos de configuração para criar um aplicativo que atenda às solicitações HTTP com ASP.NET e IIS.

Por fim, o Elastic Beanstalk usará um [manifesto de implantação](#) para configurar as implantações de aplicações .NET Core, aplicações personalizadas e várias aplicações .NET Core ou MSBuild em um único servidor. Para implantar um aplicativo .NET Core em um ambiente Windows Server, adicione um arquivo do site a um pacote de origem do aplicativo com um manifesto de implantação. O comando `dotnet publish` gera classes e dependências compiladas que você pode combinar a um arquivo `web.config` para criar o arquivo do site. O manifesto de implantação informa para o Elastic Beanstalk o caminho em que o site deve ser executado e pode ser usado para configurar grupos de aplicações e executar várias aplicações em caminhos diferentes.

O código-fonte está disponível aqui: [dotnet-core-windows-tutorial.zip](#)

Seções

- [Pré-requisitos](#)
- [Gerar um projeto .NET Core](#)
- [Iniciar um ambiente do Elastic Beanstalk](#)
- [Atualizar o código-fonte](#)
- [Implantar o aplicativo](#)
- [Limpeza](#)
- [Próximas etapas](#)

Pré-requisitos

Este tutorial usa o .NET Core SDK para gerar um aplicativo .NET Core básico, executá-lo localmente e criar um pacote implantável.

Requisitos

- .NET Core (x64) 1.0.1, 2.0.0 ou mais recente

Como instalar o .NET Core SDK

1. Faça download do instalador em microsoft.com/net/core. Selecione Windows. Escolha Baixar o .NET SDK.
2. Execute o instalador e siga as instruções.

Este tutorial usa um utilitário ZIP de linha de comando para criar um pacote de origem que você pode implantar no Elastic Beanstalk. Para usar o comando `zip` no Windows, você pode instalar o `UnxUtils`, uma coleção simples de utilitários de linha de comando úteis, como `zip` e `ls`. Se preferir, você pode [usar o Windows Explorer](#) ou qualquer outro utilitário ZIP para criar arquivos de pacote de origem.

Para instalar UnxUtils

1. Baixar [UnxUtils](#).
2. Extraia o arquivo em um diretório local. Por exemplo, `C:\Program Files (x86)`.
3. Adicione o caminho aos binários para sua variável de usuário `PATH` do Windows. Por exemplo, `C:\Program Files (x86)\UnxUtils\usr\local\wbin`.
 - a. Pressione a tecla Windows e insira **environment variables**.
 - b. Escolha Edit environment variables for your account.
 - c. Selecione `PATH` e, em seguida, Edit (Editar).
 - d. Adicione caminhos ao campo Variable value, separados por ponto e vírgula. Por exemplo:
`C:\item1\path;C:\item2\path`
 - e. Selecione OK duas vezes para aplicar as novas configurações.
 - f. Feche todas as janelas do prompt de comando em execução e abra novamente uma janela do prompt de comando.
4. Abra uma nova janela de solicitação de comando e execute o comando `zip` para verificar se ele funciona.

```
> zip -h
Copyright (C) 1990-1999 Info-ZIP
```

```
Type 'zip "-L"' for software license.  
...
```

Gerar um projeto .NET Core

Use a ferramenta de linha de comando `dotnet` para gerar um novo projeto .NET Core em C# e executá-lo localmente. O aplicativo .NET Core padrão é um utilitário de linha de comando que imprime `Hello World!` e depois é encerrado.

Como gerar um novo projeto .NET Core

1. Abra uma nova janela de prompt de comando e navegue até a pasta do usuário.

```
> cd %USERPROFILE%
```

2. Use o comando `dotnet new` para gerar um novo projeto .NET Core.

```
C:\Users\username> dotnet new console -o dotnet-core-tutorial  
Content generation time: 65.0152 ms  
The template "Console Application" created successfully.  
C:\Users\username> cd dotnet-core-tutorial
```

3. Use o comando `dotnet restore` para instalar dependências.

```
C:\Users\username\dotnet-core-tutorial> dotnet restore  
Restoring packages for C:\Users\username\dotnet-core-tutorial\dotnet-core-tutorial.csproj...  
Generating MSBuild file C:\Users\username\dotnet-core-tutorial\obj\dotnet-core-tutorial.csproj.nuget.g.props.  
Generating MSBuild file C:\Users\username\dotnet-core-tutorial\obj\dotnet-core-tutorial.csproj.nuget.g.targets.  
Writing lock file to disk. Path: C:\Users\username\dotnet-core-tutorial\obj\project.assets.json  
Restore completed in 1.25 sec for C:\Users\username\dotnet-core-tutorial\dotnet-core-tutorial.csproj.  
  
NuGet Config files used:  
  C:\Users\username\AppData\Roaming\NuGet\NuGet.Config  
  C:\Program Files (x86)\NuGet\Config\Microsoft.VisualStudio.Offline.config  
Feeds used:  
  https://api.nuget.org/v3/index.json
```

```
C:\Program Files (x86)\Microsoft SDKs\NuGetPackages\
```

4. Use o comando `dotnet run` para criar e executar o aplicativo localmente.

```
C:\Users\username\dotnet-core-tutorial> dotnet run  
Hello World!
```

Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para iniciar um ambiente do Elastic Beanstalk. Para este exemplo, você o iniciará com uma plataforma .NET. Após iniciar e configurar seu ambiente, você poderá implantar um novo código-fonte a qualquer momento.

Para iniciar um ambiente (console)

1. [Abra o console do Elastic Beanstalk usando este link pré-configurado:](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação do ambiente leva cerca de 10 minutos. Durante esse período, você pode atualizar seu código-fonte.

Atualizar o código-fonte

Modifique o aplicativo padrão para um aplicativo web que usa ASP.NET e IIS.

- ASP.NET é a framework de site para .NET.
- O IIS é o servidor Web que executa a aplicação nas instâncias do Amazon EC2 no ambiente do Elastic Beanstalk.

Os exemplos de código-fonte a seguir estão disponíveis aqui: [dotnet-core-tutorial-source.zip](#)

Note

O procedimento a seguir mostra como converter o código do projeto em um aplicativo web. Para simplificar o processo, é possível gerar o projeto como um aplicativo web desde o início. Na seção anterior [Gerar um projeto .NET Core](#), modifique o comando da etapa `dotnet new` pelo comando a seguir.

```
C:\Users\username> dotnet new web -o dotnet-core-tutorial -n WindowsSampleApp
```

Para adicionar suporte para ASP.NET e IIS ao seu código

1. Copie `Program.cs` para o diretório do aplicativo para ser executado como um construtor de host da web.

Example `c:\users\username\dotnet-core-tutorial\Program.cs`

```
namespace Microsoft.AspNetCore.Hosting;
using WindowsSampleApp;

public static class Program
{
    public static void Main(string[] args)
    {
        CreateWebHostBuilder(args).Build().Run();
    }

    public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>
        WebHost.CreateDefaultBuilder(args).UseStartup<Startup>();
}
```

2. Adicione `Startup.cs` para executar um site ASP.NET.

Example `c:\users\username\dotnet-core-tutorial\Startup.cs`

```
namespace WindowsSampleApp
{
    public class Startup
    {
        public void Configure(IApplicationBuilder app)
        {

```

```
        app.UseRouting();
        app.UseEndpoints(endpoints =>
        {
            endpoints.MapGet("/", () => "Hello World from Elastic Beanstalk");
        });
    }
}
```

3. Adicione `WindowsSampleApp.csproj`, que inclui o middleware IIS e o arquivo `web.config` da saída de dotnet publish.

Note

O exemplo a seguir foi desenvolvido usando o .NET Core Runtime 2.2.1. Pode ser necessário modificar os valores de atributos `TargetFramework` ou `Version` nos elementos `PackageReference` para corresponder à versão do .NET Core Runtime que está sendo usado nos projetos personalizados.

Example `c:\users\username\dotnet-core-tutorial\WindowsSampleApp.csproj`

```
<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>net6.0</TargetFramework>
    <RollForward>LatestMajor</RollForward>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
    <RootNamespace>WindowsSampleApp</RootNamespace>
  </PropertyGroup>

</Project>
```

Em seguida, instale as novas dependências e execute o site ASP.NET localmente.

Para executar o site localmente

1. Use o comando `dotnet restore` para instalar dependências.
2. Use o comando `dotnet run` para criar e executar o aplicativo localmente.

3. Abra [localhost:5000](#) para visualizar o site.

Para executar o aplicativo em um servidor Web, você precisa empacotar o código-fonte compilado com um arquivo de configuração `web.config` e dependências de runtime. A ferramenta `dotnet` oferece um comando `publish` que reúne esses arquivos em um diretório de acordo com a configuração em `dotnet-core-tutorial.csproj`.

Para criar seu site

- Use o comando `dotnet publish` para gerar o código compilado e as dependências em uma pasta denominada `site`.

```
C:\users\username\dotnet-core-tutorial> dotnet publish -o site
```

Para implantar a aplicação no Elastic Beanstalk, combine o arquivo do site com um [manifesto de implantação](#). Isso informa o Elastic Beanstalk como executá-lo.

Para criar um pacote de origem

1. Adicione os arquivos na pasta do site a um arquivo ZIP.

Note

Se você usar um utilitário ZIP diferente, certifique-se de adicionar todos os arquivos à pasta raiz do arquivo ZIP resultante. Isso é necessário para uma implantação bem-sucedida da aplicação no ambiente do Elastic Beanstalk.

```
C:\users\username\dotnet-core-tutorial> cd site
C:\users\username\dotnet-core-tutorial\site> zip ../site.zip *
  adding: dotnet-core-tutorial.deps.json (164 bytes security) (deflated 84%)
  adding: dotnet-core-tutorial.dll (164 bytes security) (deflated 59%)
  adding: dotnet-core-tutorial.pdb (164 bytes security) (deflated 28%)
  adding: dotnet-core-tutorial.runtimeconfig.json (164 bytes security) (deflated
26%)
  adding: Microsoft.AspNetCore.Authentication.Abstractions.dll (164 bytes security)
(deflated 49%)
```

```
adding: Microsoft.AspNetCore.Authentication.Core.dll (164 bytes security)
(deflated 57%)
adding: Microsoft.AspNetCore.Connections.Abstractions.dll (164 bytes security)
(deflated 51%)
adding: Microsoft.AspNetCore.Hosting.Abstractions.dll (164 bytes security)
(deflated 49%)
adding: Microsoft.AspNetCore.Hosting.dll (164 bytes security) (deflated 60%)
adding: Microsoft.AspNetCore.Hosting.Server.Abstractions.dll (164 bytes security)
(deflated 44%)
adding: Microsoft.AspNetCore.Http.Abstractions.dll (164 bytes security) (deflated
54%)
adding: Microsoft.AspNetCore.Http.dll (164 bytes security) (deflated 55%)
adding: Microsoft.AspNetCore.Http.Extensions.dll (164 bytes security) (deflated
50%)
adding: Microsoft.AspNetCore.Http.Features.dll (164 bytes security) (deflated
50%)
adding: Microsoft.AspNetCore.HttpOverrides.dll (164 bytes security) (deflated
49%)
adding: Microsoft.AspNetCore.Server.IISIntegration.dll (164 bytes security)
(deflated 46%)
adding: Microsoft.AspNetCore.Server.Kestrel.Core.dll (164 bytes security)
(deflated 63%)
adding: Microsoft.AspNetCore.Server.Kestrel.dll (164 bytes security) (deflated
46%)
adding: Microsoft.AspNetCore.Server.Kestrel.Https.dll (164 bytes security)
(deflated 44%)
adding: Microsoft.AspNetCore.Server.Kestrel.Transport.Abstractions.dll (164 bytes
security) (deflated 56%)
adding: Microsoft.AspNetCore.Server.Kestrel.Transport.Sockets.dll (164 bytes
security) (deflated 51%)
adding: Microsoft.AspNetCore.WebUtilities.dll (164 bytes security) (deflated 55%)
adding: Microsoft.Extensions.Configuration.Abstractions.dll (164 bytes security)
(deflated 48%)
adding: Microsoft.Extensions.Configuration.Binder.dll (164 bytes security)
(deflated 47%)
adding: Microsoft.Extensions.Configuration.dll (164 bytes security) (deflated
46%)
adding: Microsoft.Extensions.Configuration.EnvironmentVariables.dll (164 bytes
security) (deflated 46%)
adding: Microsoft.Extensions.Configuration.FileExtensions.dll (164 bytes
security) (deflated 47%)
adding: Microsoft.Extensions.DependencyInjection.Abstractions.dll (164 bytes
security) (deflated 54%)
```

```
adding: Microsoft.Extensions.DependencyInjection.dll (164 bytes security)
(deflated 53%)
adding: Microsoft.Extensions.FileProviders.Abstractions.dll (164 bytes security)
(deflated 46%)
adding: Microsoft.Extensions.FileProviders.Physical.dll (164 bytes security)
(deflated 47%)
adding: Microsoft.Extensions.FileSystemGlobbing.dll (164 bytes security)
(deflated 49%)
adding: Microsoft.Extensions.Hosting.Abstractions.dll (164 bytes security)
(deflated 47%)
adding: Microsoft.Extensions.Logging.Abstractions.dll (164 bytes security)
(deflated 54%)
adding: Microsoft.Extensions.Logging.dll (164 bytes security) (deflated 48%)
adding: Microsoft.Extensions.ObjectPool.dll (164 bytes security) (deflated 45%)
adding: Microsoft.Extensions.Options.dll (164 bytes security) (deflated 53%)
adding: Microsoft.Extensions.Primitives.dll (164 bytes security) (deflated 50%)
adding: Microsoft.Net.Http.Headers.dll (164 bytes security) (deflated 53%)
adding: System.IO.Pipelines.dll (164 bytes security) (deflated 50%)
adding: System.Runtime.CompilerServices.Unsafe.dll (164 bytes security) (deflated
43%)
adding: System.Text.Encodings.Web.dll (164 bytes security) (deflated 57%)
adding: web.config (164 bytes security) (deflated 39%)
C:\users\username\dotnet-core-tutorial\site> cd ../
```

2. Adicione um manifesto de implantação que aponta para o arquivo do site.

Example c:\users\username\dotnet-core-tutorial\aws-windows-deployment-manifest.json

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "test-dotnet-core",
        "parameters": {
          "appBundle": "site.zip",
          "iisPath": "/",
          "iisWebSite": "Default Web Site"
        }
      }
    ]
  }
}
```


- Use o comando `zip` para criar um pacote de origem denominado `dotnet-core-tutorial.zip`.

```
C:\users\username\dotnet-core-tutorial> zip dotnet-core-tutorial.zip site.zip aws-  
windows-deployment-manifest.json  
  adding: site.zip (164 bytes security) (stored 0%)  
  adding: aws-windows-deployment-manifest.json (164 bytes security) (deflated 50%)
```

Implantar o aplicativo

Implante o pacote de origem no ambiente do Elastic Beanstalk criado por você.

[Você pode baixar o pacote de origem aqui: .zip dotnet-core-tutorial-bundle](#)

Para implantar um pacote de origem

- Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
- No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

- Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
- Use a caixa de diálogo na tela para carregar o pacote de origem.
- Escolha Deploy (Implantar).
- Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

O aplicativo simplesmente grava `Hello from ASP.NET Core!` na resposta e é retornado.

A screenshot of a terminal window showing the output of the application. The text "Hello from ASP.NET Core!" is displayed in a monospaced font, with "Hello" in blue, "from" in green, "ASP.NET" in red, and "Core!" in blue. The terminal window has a white background and a black border.

Ao iniciar um ambiente, os seguintes recursos são criados:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de

solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Note

O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para obter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon S3](#).

Limpeza

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. [O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente, como instâncias do Amazon EC2, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.](#)

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

Próximas etapas

À medida que você desenvolva a aplicação, provavelmente vai precisar gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de [linha de comando do Elastic Beanstalk](#) (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

Se você usa o Visual Studio para desenvolver seu aplicativo, você também pode usar o AWS Toolkit for Visual Studio para implantar alterações, gerenciar seus ambientes do Elastic Beanstalk e gerenciar outros recursos. AWS Consulte [A AWS Toolkit for Visual Studio](#) para obter mais informações.

Para desenvolvimento e teste, convém usar a funcionalidade do Elastic Beanstalk para adicionar uma instância de banco de dados gerenciada diretamente ao seu ambiente. Para obter instruções sobre como configurar um banco de dados no ambiente, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, [configure um nome de domínio personalizado](#) para o seu ambiente e [habilite HTTPS](#) para conexões seguras.

Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicações .NET

É possível usar uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS) para armazenar dados coletados e modificados pela aplicação. O banco de dados pode ser acoplado ao seu ambiente e gerenciado pelo Elastic Beanstalk ou pode ser criado como desacoplado e gerenciado externamente por outro serviço. Este tópico fornece instruções para criar o Amazon RDS usando o console do Elastic Beanstalk. O banco de dados será acoplado ao ambiente e gerenciado pelo Elastic Beanstalk. Para obter mais informações sobre como integrar um Amazon RDS com o Elastic Beanstalk, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

Seções

- [Adicionar uma instância de banco de dados ao seu ambiente](#)
- [Download de um driver](#)
- [Conectar-se a um banco de dados](#)

Adicionar uma instância de banco de dados ao seu ambiente

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização do ambiente for concluída, o nome de host da instância de banco de dados e outras informações de conexão estarão disponíveis para o seu aplicativo por meio das seguintes propriedades de ambiente:

Nome da propriedade	Descrição	Valor da propriedade
RDS_HOSTNAME	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
RDS_PORT	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
RDS_DB_NAME	O nome do banco de dado, ebdb .	Na guia Configuration (Configuração) no console

Nome da propriedade	Descrição	Valor da propriedade
		do Amazon RDS: DB Name (Nome do banco de dados).
RDS_USERNAME	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Para obter mais informações sobre como configurar uma instância de banco de dados acoplada a um ambiente Elastic Beanstalk, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

Download de um driver

Faça download e instale o pacote EntityFramework e um driver de banco de dados para o seu ambiente de desenvolvimento com NuGet.

Provedores comuns de banco de dados de framework de entidade para .NET

- SQL Server – `Microsoft.EntityFrameworkCore.SqlServer`
- MySQL – `Pomelo.EntityFrameworkCore.MySql`
- PostgreSQL – `Npgsql.EntityFrameworkCore.PostgreSQL`

Conectar-se a um banco de dados

O Elastic Beanstalk fornece informações de conexão referentes às instâncias de banco de dados associadas nas propriedades de ambiente. Use `ConfigurationManager.AppSettings` para ler as propriedades e configurar uma conexão de banco de dados.

Example Helpers.cs - método de string de conexão

```
using System;  
using System.Collections.Generic;
```

```
using System.Configuration;
using System.Linq;
using System.Web;

namespace MVC5App.Models
{
    public class Helpers
    {
        public static string GetRDSConnectionString()
        {
            var appConfig = ConfigurationManager.AppSettings;

            string dbname = appConfig["RDS_DB_NAME"];

            if (string.IsNullOrEmpty(dbname)) return null;

            string username = appConfig["RDS_USERNAME"];
            string password = appConfig["RDS_PASSWORD"];
            string hostname = appConfig["RDS_HOSTNAME"];
            string port = appConfig["RDS_PORT"];

            return "Data Source=" + hostname + ";Initial Catalog=" + dbname + ";User ID=" +
                username + ";Password=" + password + ";";
        }
    }
}
```

Use a string de conexão para inicializar o contexto do banco de dados.

Example DBContext.cs

```
using System.Data.Entity;
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;

namespace MVC5App.Models
{
    public class RDSContext : DbContext
    {
        public RDSContext()
            : base(GetRDSConnectionString())
        {

```

```
    }  
  
    public static RDSContext Create()  
    {  
        return new RDSContext();  
    }  
}  
}
```

A AWS Toolkit for Visual Studio

O Visual Studio fornece modelos para diferentes linguagens de programação e tipos de aplicativos. Você pode começar com qualquer um desses modelos. O AWS Toolkit for Visual Studio também oferece três modelos de projeto que iniciam o desenvolvimento da sua aplicação: AWS Console Project, AWS Web Project e AWS Empty Project. Para este exemplo, você vai criar um novo aplicativo web ASP.NET.

Como criar um novo projeto de aplicativo web ASP.NET

1. No Visual Studio, no menu Arquivo, clique em Novo e em Projeto.
2. Na caixa de diálogo Novo projeto, clique em Modelos instalados, clique em Visual C# e, em seguida, clique em Web. Clique em Aplicativo web ASP.NET vazio, digite um nome de projeto e, em seguida, clique em OK.

Para executar um projeto

Execute um destes procedimentos:

1. Pressione F5.
2. Selecione Iniciar depuração no menu Depurar.

Testar localmente

O Visual Studio facilita testar seu aplicativo localmente. Para testar ou executar aplicativos web em ASP.NET, você precisa de um servidor web. O Visual Studio oferece várias opções, como o Internet Information Services (IIS), IIS Express ou o Visual Studio Development Server. Para saber mais sobre cada uma dessas opções e decidir qual é a melhor para você, consulte [Servidores Web no Visual Studio para projetos Web ASP.NET](#).

Criar um ambiente do Elastic Beanstalk

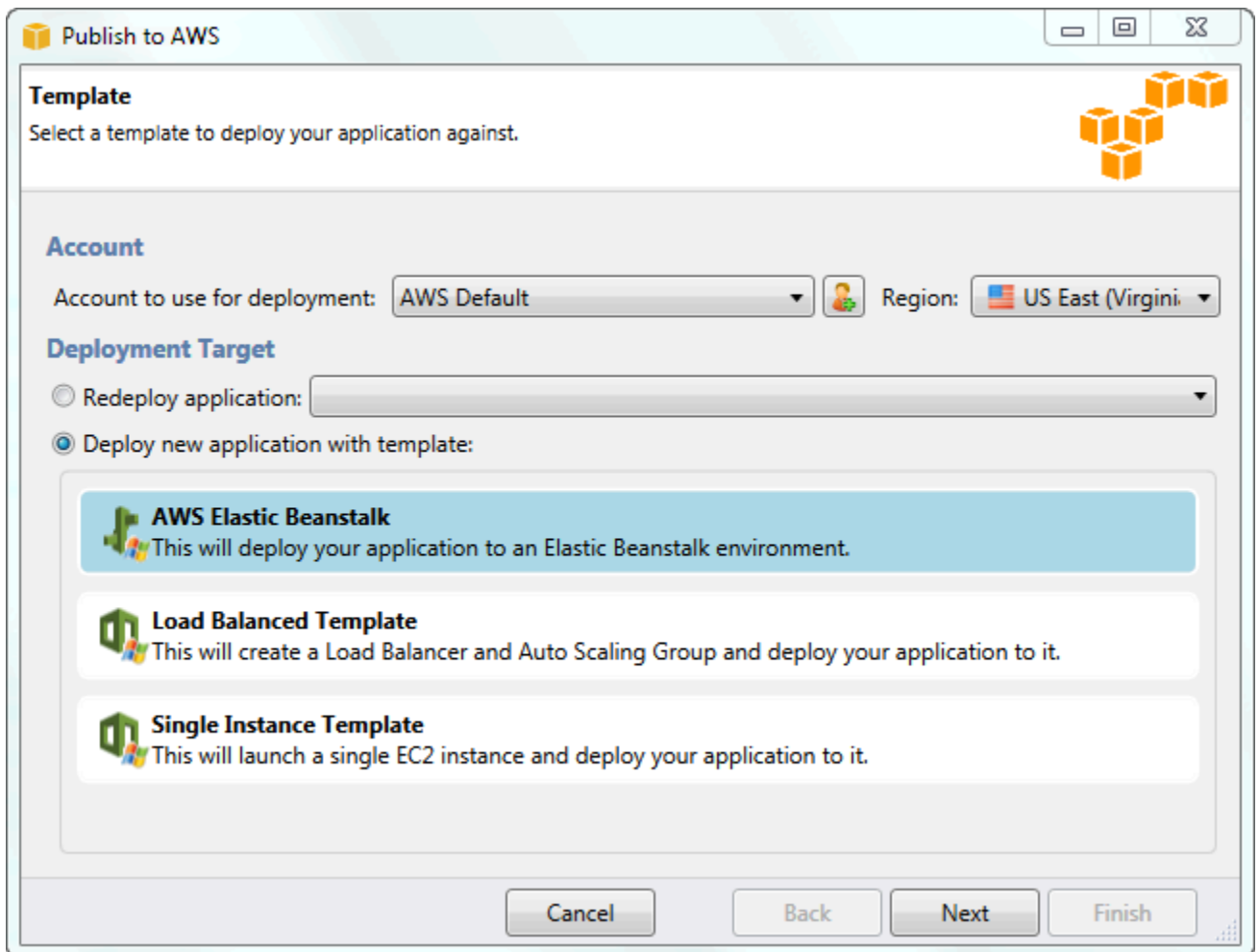
Depois de testar a aplicação, você estará pronto para implantá-la no Elastic Beanstalk.

Note

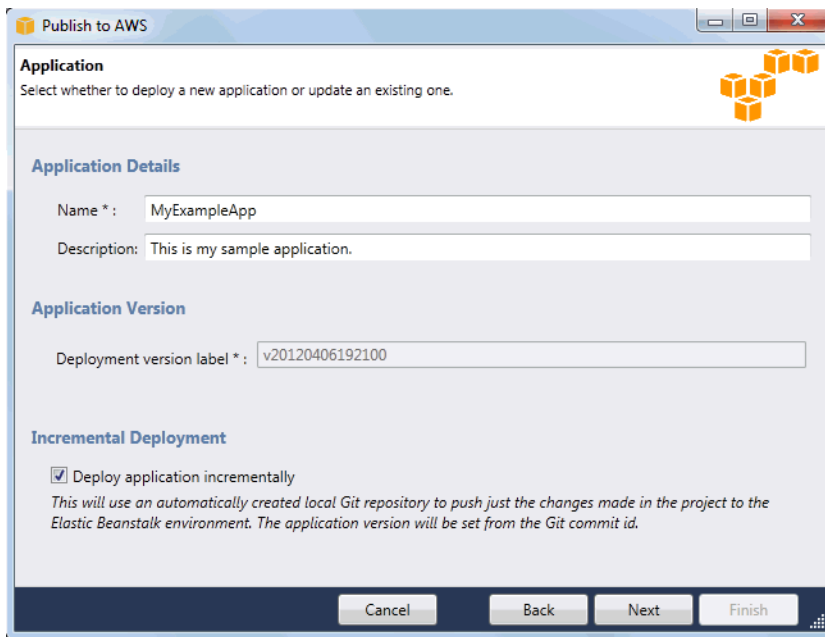
O [Arquivo de configuração](#) precisa fazer parte do projeto para ser incluído no arquivo. Se preferir, em vez de incluir os arquivos de configuração no projeto, você pode usar o Visual Studio para implantar todos os arquivos na pasta do projeto. Em Solution Explorer, clique com o botão direito do mouse no nome do projeto e, em seguida, clique em Propriedades. Clique na guia Package/Publish Web. Na seção Items to deploy, selecione All Files in the Project Folder na lista suspensa.

Para implantar aplicações no Elastic Beanstalk utilizando o AWS Toolkit for Visual Studio

1. Em Explorador de soluções, clique com o botão direito do mouse na aplicação e depois escolha Publicar na AWS.
2. No assistente Publicar na AWS, insira as informações da sua conta.
 - a. Para Conta da AWS a utilizar para implantação, selecione sua conta ou selecione Outra para inserir as informações da nova conta.
 - b. Para Região, selecione a região em que você deseja implantar o aplicativo. Para obter informações sobre regiões da AWS disponíveis, consulte [Endpoints e cotas do AWS Elastic Beanstalk](#) no Referência geral da AWS. Se você selecionar uma região não compatível com o Elastic Beanstalk, a opção para implantar no Elastic Beanstalk se tornará indisponível.
 - c. Clique em Implantar novo aplicativo com modelo e selecione Elastic Beanstalk. Em seguida, clique em Próximo.




3. Na página Aplicativo, insira os detalhes de seu aplicativo.
 - a. Em Nome, digite o nome do aplicativo.
 - b. Em Descrição, digite uma descrição do aplicativo. Esta etapa é opcional.
 - c. O rótulo da versão do aplicativo é exibido automaticamente em Deployment version label (Rótulo da versão de implantação)
 - d. Selecione Implantar aplicativo incrementalmente para implantar somente os arquivos alterados. Uma implantação incremental é mais rápida porque você está atualizando somente os arquivos alterados em vez de todos os arquivos. Se você escolher essa opção, uma versão do aplicativo será definida a partir do ID de confirmação do Git. Se você optar por não implantar seu aplicativo de forma incremental, poderá atualizar o rótulo da versão na caixa Rótulo da versão de implantação.



- e. Clique em Next.
4. Na página Ambiente, descreva os detalhes de seu ambiente.
 - a. Selecione Criar um novo ambiente para este aplicativo.
 - b. Em Nome, digite um nome para o ambiente.
 - c. Em Descrição, caracterize seu ambiente. Esta etapa é opcional.
 - d. Em Tipo, selecione o tipo de ambiente que deseja.

Você pode selecionar um ambiente Carga balanceada, escalabilidade automática ou Única instância. Para obter mais informações, consulte [Tipos de ambiente](#).

 Note

Para ambientes de uma única instância, as configurações de balanceamento de carga, autoescalabilidade e URL de verificação de integridade não se aplicam

- e. O URL do ambiente é exibido automaticamente em URL do ambiente assim que você move o cursor para essa caixa.
- f. Clique em Verificar disponibilidade para garantir que o URL do ambiente esteja disponível.

Publish to AWS

Environment
Select or define an environment in which the application will run.

Create a new environment for the application:

Name * :

Description:

Type:

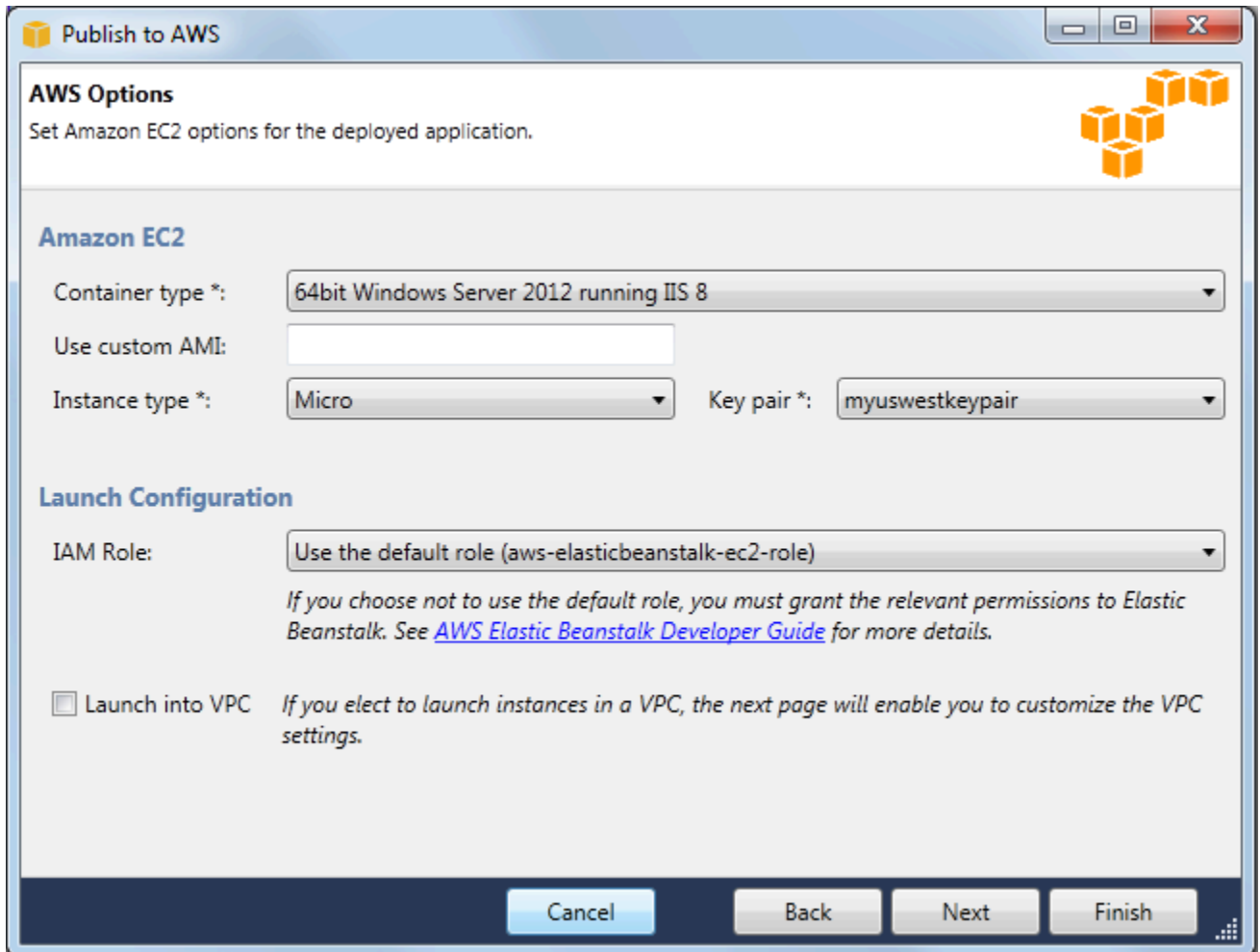
Environment URL:
 .elasticbeanstalk.com

Use an existing environment:

- g. Clique em Next.
5. Na página Opções da AWS, configure as opções adicionais e as informações de segurança de sua implantação.
- Em Tipo de contêiner, selecione Windows Server 2012 de 64 bits executando IIS 8 ou Windows Server 2008 de 64 bits executando IIS 7.5.
 - Para Tipo de instância, selecione Micro.
 - Para Par de chaves, selecione Criar novo par de chaves. Digite um nome para o novo par de chaves: neste exemplo, usamos **myuswestkeypair** e, depois, clique em OK. Um par de chaves permite o acesso do desktop remoto em suas instâncias do Amazon EC2. Para obter mais informações sobre pares de chaves do Amazon EC2, consulte [Usar credenciais](#) no Guia do usuário do Amazon Elastic Compute Cloud.
 - Selecione um perfil da instância.

Se você não tiver um perfil da instância, selecione Criar um perfil da instância padrão. Para obter informações sobre como usar perfis de instância com o Elastic Beanstalk, consulte [Gerenciar perfis de instância do Elastic Beanstalk](#).

- e. Se você tiver uma VPC personalizada que gostaria de usar com seu ambiente, clique em Iniciar no VPC. Você pode configurar as informações da VPC na próxima página. Para obter mais informações sobre a Amazon VPC, consulte [Amazon Virtual Private Cloud \(Amazon VPC\)](#). Para obter uma lista de tipos de contêiner não legados compatíveis, consulte [the section called “Por que algumas versões de plataforma são marcadas como legadas?”](#)

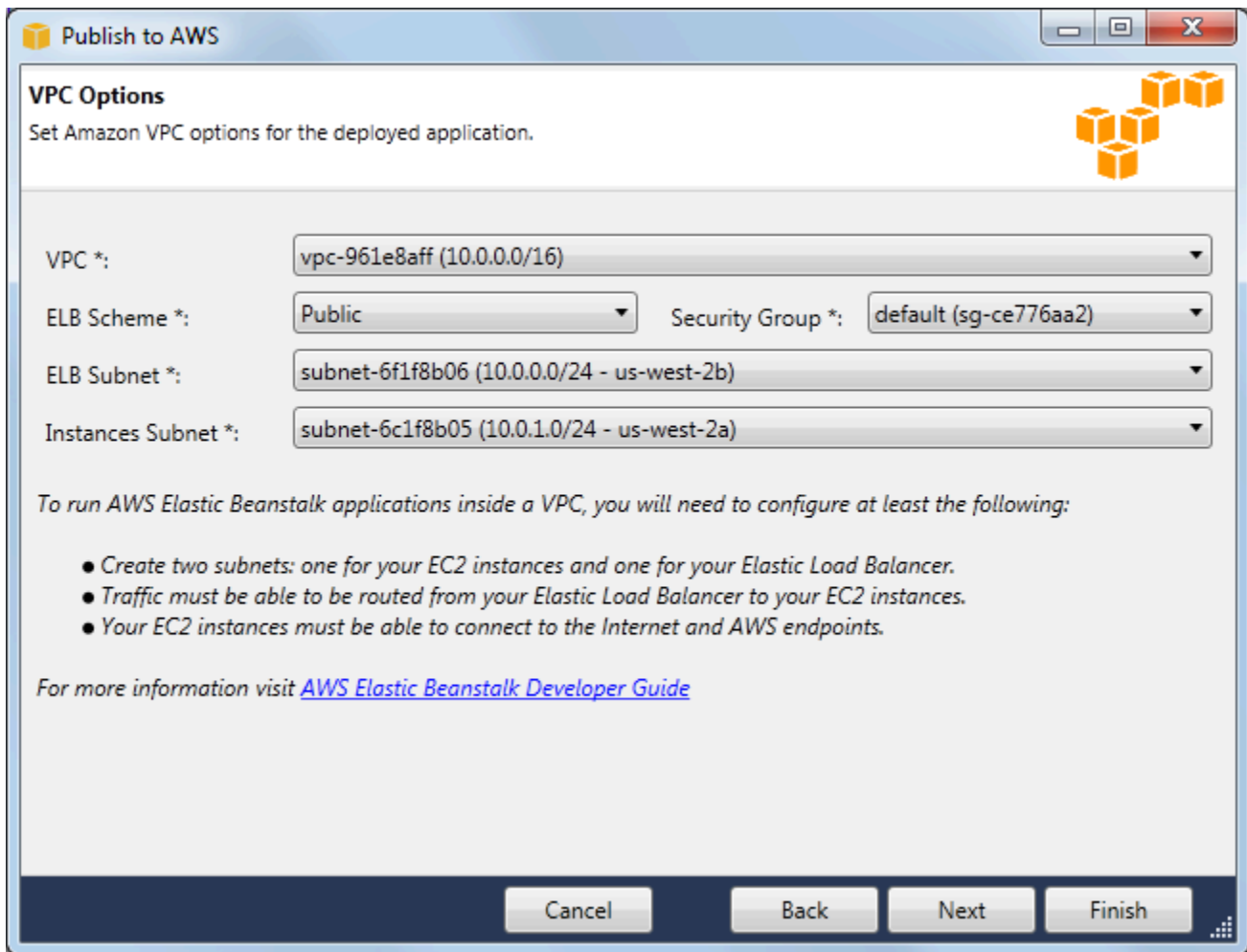


The screenshot shows the 'Publish to AWS' dialog box with the following configuration:

- AWS Options**
 - Set Amazon EC2 options for the deployed application.
- Amazon EC2**
 - Container type *: 64bit Windows Server 2012 running IIS 8
 - Use custom AMI: (empty text box)
 - Instance type *: Micro
 - Key pair *: myuswestkeypair
- Launch Configuration**
 - IAM Role: Use the default role (aws-elasticbeanstalk-ec2-role)
 - If you choose not to use the default role, you must grant the relevant permissions to Elastic Beanstalk. See [AWS Elastic Beanstalk Developer Guide](#) for more details.*
 - Launch into VPC *If you elect to launch instances in a VPC, the next page will enable you to customize the VPC settings.*

Buttons at the bottom: Cancel, Back, Next, Finish.

- f. Clique em Next.
6. Se você optou por iniciar seu ambiente dentro de uma VPC, a Opções da VPC será exibida; caso contrário, a página Opções adicionais será exibida. Aqui você configurará as opções de sua VPC.



Publish to AWS

VPC Options
Set Amazon VPC options for the deployed application.

VPC *: vpc-961e8aff (10.0.0.0/16)

ELB Scheme *: Public Security Group *: default (sg-ce776aa2)

ELB Subnet *: subnet-6f1f8b06 (10.0.0.0/24 - us-west-2b)

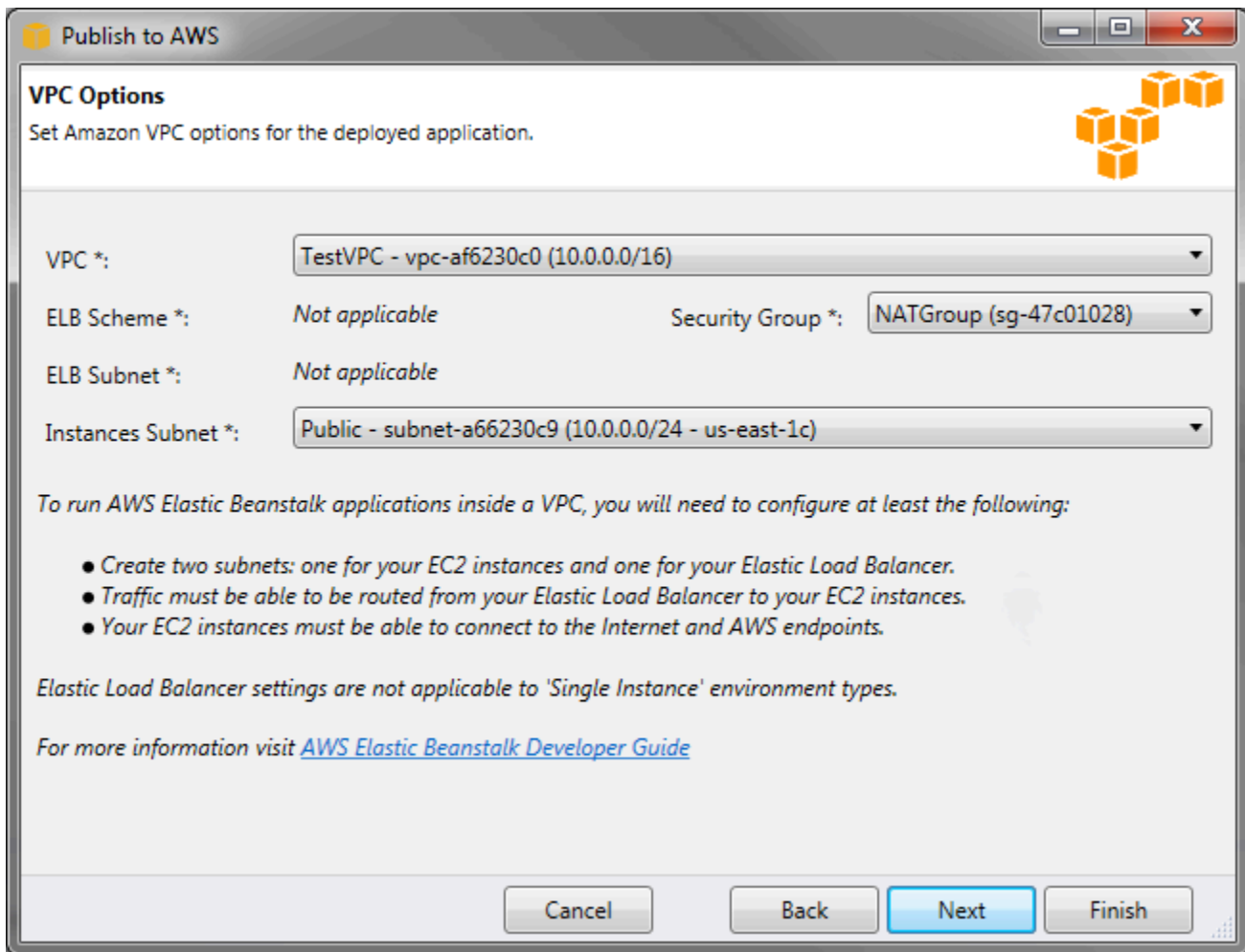
Instances Subnet *: subnet-6c1f8b05 (10.0.1.0/24 - us-west-2a)

To run AWS Elastic Beanstalk applications inside a VPC, you will need to configure at least the following:

- Create two subnets: one for your EC2 instances and one for your Elastic Load Balancer.
- Traffic must be able to be routed from your Elastic Load Balancer to your EC2 instances.
- Your EC2 instances must be able to connect to the Internet and AWS endpoints.

For more information visit [AWS Elastic Beanstalk Developer Guide](#)

Cancel Back Next Finish



Publish to AWS

VPC Options
Set Amazon VPC options for the deployed application.

VPC *: TestVPC - vpc-af6230c0 (10.0.0.0/16)

ELB Scheme *: Not applicable Security Group *: NATGroup (sg-47c01028)

ELB Subnet *: Not applicable

Instances Subnet *: Public - subnet-a66230c9 (10.0.0.0/24 - us-east-1c)

To run AWS Elastic Beanstalk applications inside a VPC, you will need to configure at least the following:

- Create two subnets: one for your EC2 instances and one for your Elastic Load Balancer.
- Traffic must be able to be routed from your Elastic Load Balancer to your EC2 instances.
- Your EC2 instances must be able to connect to the Internet and AWS endpoints.

Elastic Load Balancer settings are not applicable to 'Single Instance' environment types.

For more information visit [AWS Elastic Beanstalk Developer Guide](#)

Cancel Back **Next** Finish

- Selecione o ID da VPC em que gostaria de iniciar seu ambiente.
- Para um ambiente escalável e com balanceamento de carga, selecione private (privado) em ELB Scheme (Esquema ELB) se não quiser que o Elastic Load Balancer fique disponível na Internet.

Para um ambiente de única instância, essa opção não é aplicável porque o ambiente não tem um load balancer. Para obter mais informações, consulte [Tipos de ambiente](#).

- Para um ambiente escalável e com balanceamento de carga, selecione as sub-redes para o Elastic Load Balancer e as instâncias do EC2. Se você criou sub-redes públicas e privadas, o Elastic Load Balancer e as instâncias do EC2 precisam estar associados à sub-rede correta. Por padrão, a Amazon VPC cria uma sub-rede pública padrão usando 10.0.0.0/24 e uma sub-rede privada usando 10.0.1.0/24. É possível visualizar as sub-redes existentes no console da Amazon VPC em <https://console.aws.amazon.com/vpc/>.

Para um ambiente de única instância, a VPC precisa apenas de uma sub-rede pública para a instância. A seleção de uma sub-rede para o load balancer não é aplicável porque o ambiente não tem um load balancer. Para obter mais informações, consulte [Tipos de ambiente](#).

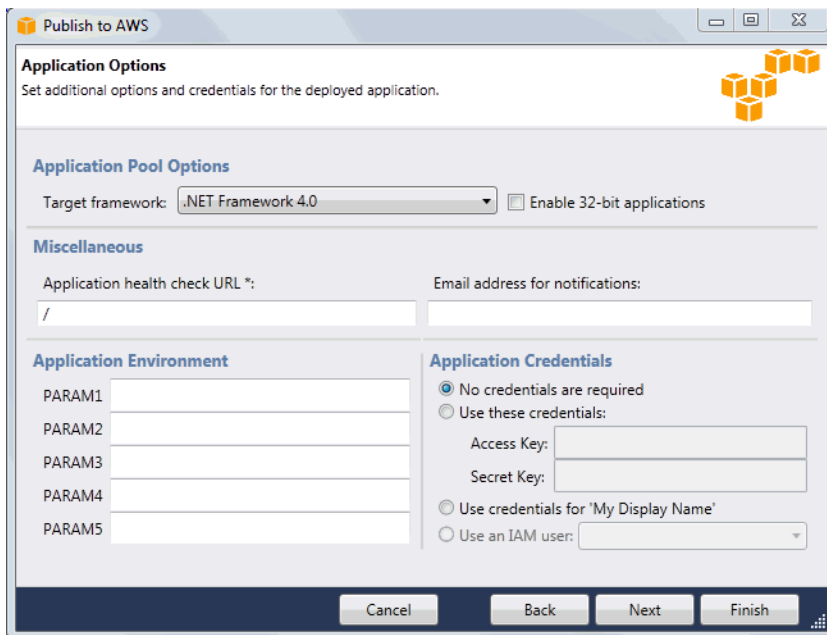
- d. Para um ambiente escalável e com balanceamento de carga, selecione o grupo de segurança criado para as instâncias, se aplicável.

Para um ambiente de única instância, você não precisa de um dispositivo NAT. Selecione o security group padrão. O Elastic Beanstalk atribui um endereço IP elástico à instância, permitindo que ela acesse a Internet.

- e. Clique em Next.

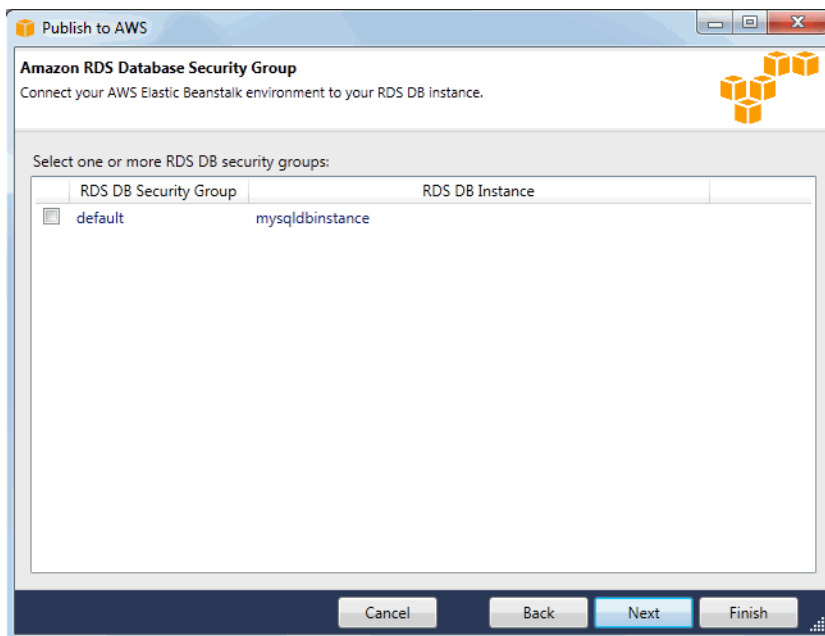
7. Na página Opções do aplicativo, configure as opções de seu aplicativo.

- a. Para o framework de destino, selecione .NET Framework 4.0.
- b. O Elastic Load Balancing usa uma verificação de integridade para determinar se as instâncias do Amazon EC2 que executam seu aplicativo estão íntegras. A verificação de integridade determina um status de integridade da instância sondando um URL especificado em um intervalo definido. Você pode substituir o URL padrão para corresponder a um recurso existente no seu aplicativo (por exemplo, `/myapp/index.aspx`), inserindo-o na caixa URL da verificação de integridade do aplicativo. Para obter mais informações sobre como configurar verificações de integridade do aplicativo, consulte [Verificação de integridade](#).
- c. Digite um endereço de e-mail para receber notificações do Amazon Simple Notification Service (Amazon SNS) sobre eventos importantes que afetam o seu aplicativo.
- d. A seção Ambiente do aplicativo permite que você especifique variáveis de ambiente nas instâncias do Amazon EC2 que estão executando seu aplicativo. Essa configuração proporciona maior portabilidade, pois acaba com a necessidade de recompilar o código-fonte quando você move entre ambientes.
- e. Selecione a opção de credenciais do aplicativo que deseja usar para implantar seu aplicativo.



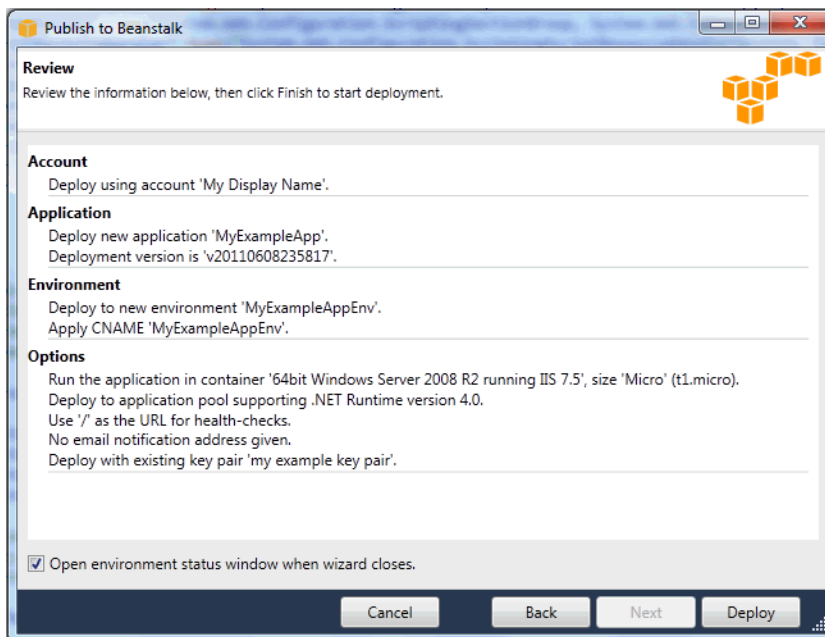
The screenshot shows the 'Publish to AWS' wizard window. The title bar says 'Publish to AWS'. The main heading is 'Application Options' with the subtitle 'Set additional options and credentials for the deployed application.' Below this, there are several sections: 'Application Pool Options' with a 'Target framework' dropdown set to '.NET Framework 4.0' and an 'Enable 32-bit applications' checkbox; 'Miscellaneous' with 'Application health check URL' and 'Email address for notifications' text boxes; 'Application Environment' with five 'PARAM' text boxes; and 'Application Credentials' with radio buttons for 'No credentials are required', 'Use these credentials:' (with 'Access Key' and 'Secret Key' text boxes), 'Use credentials for 'My Display Name'', and 'Use an IAM user:' (with a dropdown menu). At the bottom are 'Cancel', 'Back', 'Next', and 'Finish' buttons.

- f. Clique em Next.
8. Se você já tiver configurado um banco de dados do Amazon RDS, a página Security group do banco de dados do Amazon RDS será exibida. Se você deseja conectar seu ambiente do Elastic Beanstalk com sua instância de banco de dados do Amazon RDS, selecione um ou mais security groups. Caso contrário, vá para a próxima etapa. Quando estiver pronto, clique em Próximo.

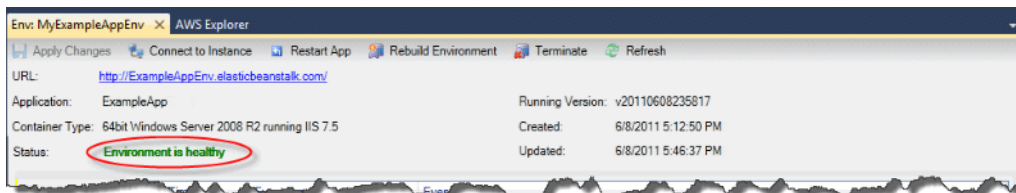


The screenshot shows the 'Publish to AWS' wizard window at the 'Amazon RDS Database Security Group' step. The title bar says 'Publish to AWS'. The main heading is 'Amazon RDS Database Security Group' with the subtitle 'Connect your AWS Elastic Beanstalk environment to your RDS DB instance.' Below this, there is a section 'Select one or more RDS DB security groups:' containing a table with two columns: 'RDS DB Security Group' and 'RDS DB Instance'. The table has one row with 'default' in the first column and 'mysqldbinstance' in the second column, with a checkbox selected next to 'default'. At the bottom are 'Cancel', 'Back', 'Next', and 'Finish' buttons.

9. Revise as opções de implantação. Se tudo estiver como você deseja, clique em Implantar.



Seu projeto ASP.NET atualizado será exportado como um arquivo de implantação da web, enviado para o Amazon S3 e registrado como uma nova versão do aplicativo no Elastic Beanstalk. O recurso de implantação do Elastic Beanstalk irá monitorar seu ambiente até ele se tornar disponível com o código recém-implantado. Na guia env:<nome do ambiente>, você verá o status do seu ambiente.



Encerrar um ambiente

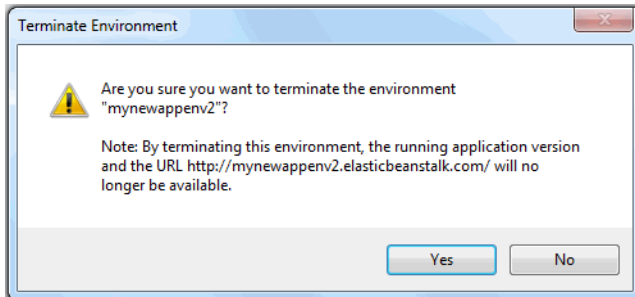
Com o AWS Toolkit for Visual Studio, você pode terminar um ambiente em execução para evitar a cobrança de recursos da AWS não utilizados.

Note

Você sempre pode iniciar um novo ambiente usando a mesma versão mais tarde.

Para encerrar um ambiente

1. Expanda o nó Elastic Beanstalk e o nó da aplicação no AWS Explorer. Clique com o botão direito do mouse no ambiente de seu aplicativo e selecione Encerrar ambiente.
2. Quando solicitado, clique em Sim para confirmar que você deseja encerrar o ambiente. Alguns minutos serão necessários para o Elastic Beanstalk terminar os recursos da AWS em execução no ambiente.



Note

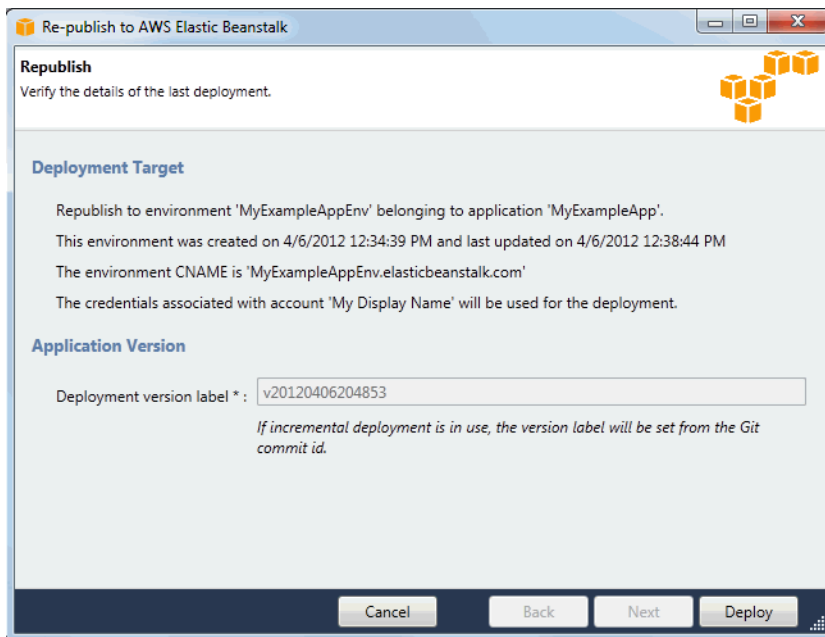
Quando você encerra o ambiente, o CNAME associado ao ambiente encerrado torna-se disponível para qualquer pessoa utilizar.

Implantar em seu ambiente

Agora que você testou o seu aplicativo, é fácil editá-lo, reimplantá-lo e ver os resultados em instantes.

Para editar e reimplantar seu aplicativo web ASP.NET

1. Em Solution Explorer, clique com o botão direito em seu aplicativo e, em seguida, clique em Republish to Environment **nome do seu ambiente**>. O assistente Republicar no AWS Elastic Beanstalk é aberto.



2. Revise os detalhes de implantação e clique em Deploy.

Note

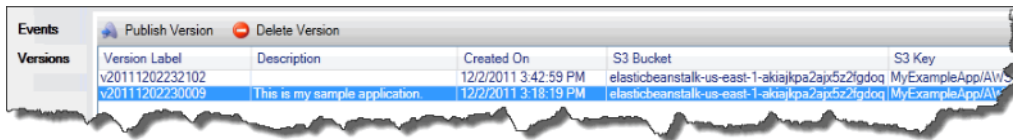
Se você quiser alterar as configurações, clique em Cancelar e use, em vez disso, o assistente Publicar na AWS. Para obter instruções, consulte [Criar um ambiente do Elastic Beanstalk](#).

Seu projeto web ASP.NET atualizado será exportado como um arquivo de implantação da web com o rótulo da nova versão, enviado para o Amazon S3 e registrado como uma nova versão do aplicativo no Elastic Beanstalk. O recurso de implantação do Elastic Beanstalk monitora seu ambiente existente até ele se tornar disponível com o código recém-implantado. Na guia env:<nome do ambiente>, você verá o status do seu ambiente.

Você também pode implantar um aplicativo existente em um ambiente existente se, por exemplo, você precisar reverter para uma versão anterior do aplicativo.

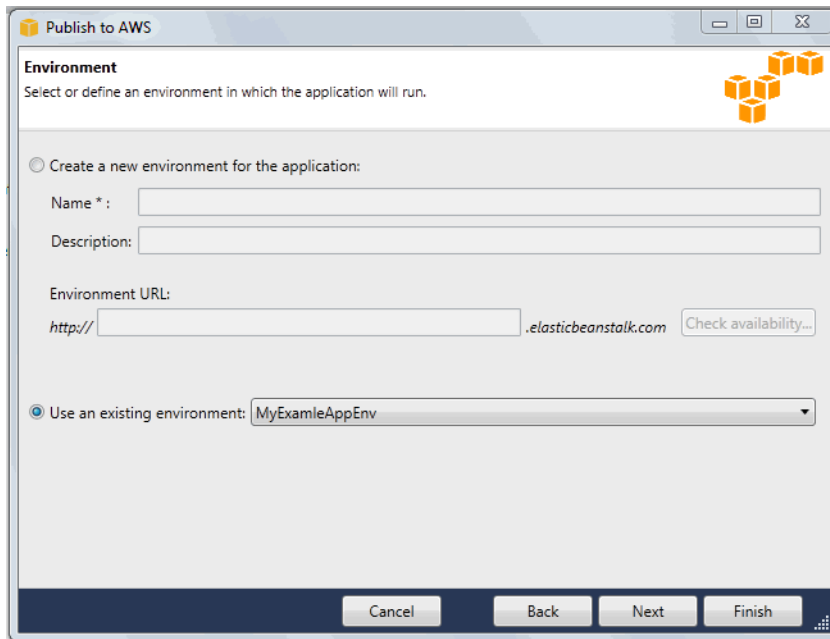
Para implantar uma versão do aplicativo em um ambiente existente

1. Clique com o botão direito do mouse na aplicação do Elastic Beanstalk expandindo o nó Elastic Beanstalk no AWS Explorer. Selecione View Status.
2. Na guia App: <nome do aplicativo>, clique em Versions.



Version Label	Description	Created On	S3 Bucket	S3 Key
v20111202232102		12/2/2011 3:42:59 PM	elasticbeanstalk-us-east-1-akiajka2ap5z2fdoq	MyExampleApp/AW
v20111202233009	This is my sample application.	12/2/2011 3:18:19 PM	elasticbeanstalk-us-east-1-akiajka2ap5z2fdoq	MyExampleApp/AW

3. Clique na versão do aplicativo que você deseja implantar e clique em Publish Version.
4. No assistente Publish Application Version, clique em Next.



Publish to AWS

Environment
Select or define an environment in which the application will run.

Create a new environment for the application:

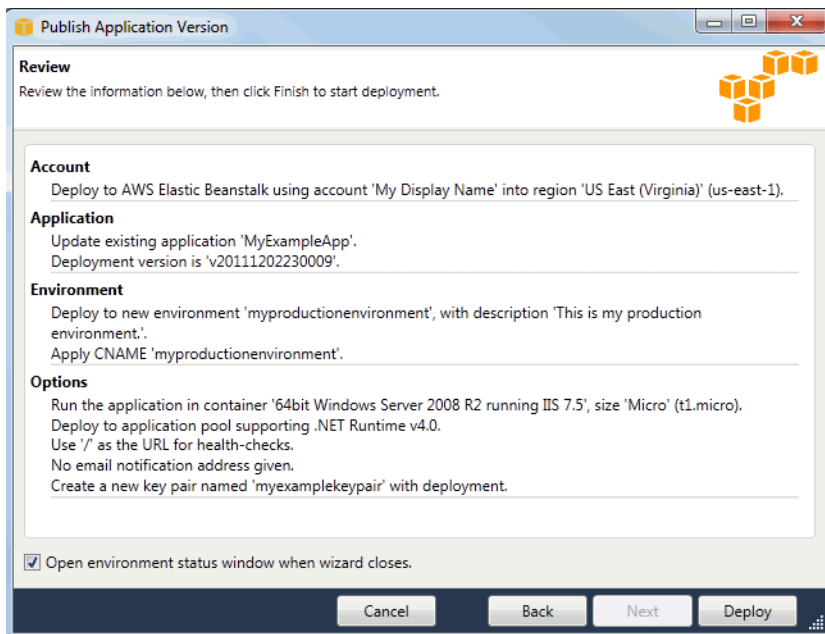
Name * :

Description:

Environment URL:
http:// .elasticbeanstalk.com

Use an existing environment:

5. Analise as opções de implantação e clique em Deploy.



Publish Application Version

Review
Review the information below, then click Finish to start deployment.

Account
Deploy to AWS Elastic Beanstalk using account 'My Display Name' into region 'US East (Virginia)' (us-east-1).

Application
Update existing application 'MyExampleApp'.
Deployment version is 'v20111202233009'.

Environment
Deploy to new environment 'myproductionenvironment', with description 'This is my production environment'.
Apply CNAME 'myproductionenvironment'.

Options
Run the application in container '64bit Windows Server 2008 R2 running IIS 7.5', size 'Micro' (t1.micro).
Deploy to application pool supporting .NET Runtime v4.0.
Use '/' as the URL for health-checks.
No email notification address given.
Create a new key pair named 'myexamplekeypair' with deployment.

Open environment status window when wizard closes.

Seu projeto ASP.NET será exportado como um arquivo de implantação da web e enviado para o Amazon S3. O recurso de implantação do Elastic Beanstalk irá monitorar seu ambiente até ele se tornar disponível com o código recém-implantado. Na guia env:<**nome do ambiente**>, você verá o status do seu ambiente.

Gerenciar os ambientes de aplicações do Elastic Beanstalk

Com o AWS Toolkit for Visual Studio e o Console de Gerenciamento da AWS, você pode alterar o provisionamento e a configuração dos recursos da AWS usados por seus ambientes de aplicações. Para obter informações sobre como gerenciar seus ambientes de aplicações usando o Console de Gerenciamento da AWS, consulte [Gerenciar ambientes](#). Esta seção discute as configurações de serviço específicas que você pode editar no AWS Toolkit for Visual Studio como parte da configuração de ambientes de aplicações.

Alterar os ajustes de configuração do ambiente

Quando você implanta a aplicação, o Elastic Beanstalk configura uma série de serviços de computação na nuvem AWS. Você pode controlar como esses serviços individuais são configurados usando o AWS Toolkit for Visual Studio.

Para editar as configurações de ambiente de um aplicativo

- Expanda o nó do Elastic Beanstalk e o nó da aplicação. Depois, clique com o botão direito no ambiente do Elastic Beanstalk no AWS Explorer. Selecione View Status.

Agora, você pode definir as seguintes configurações:

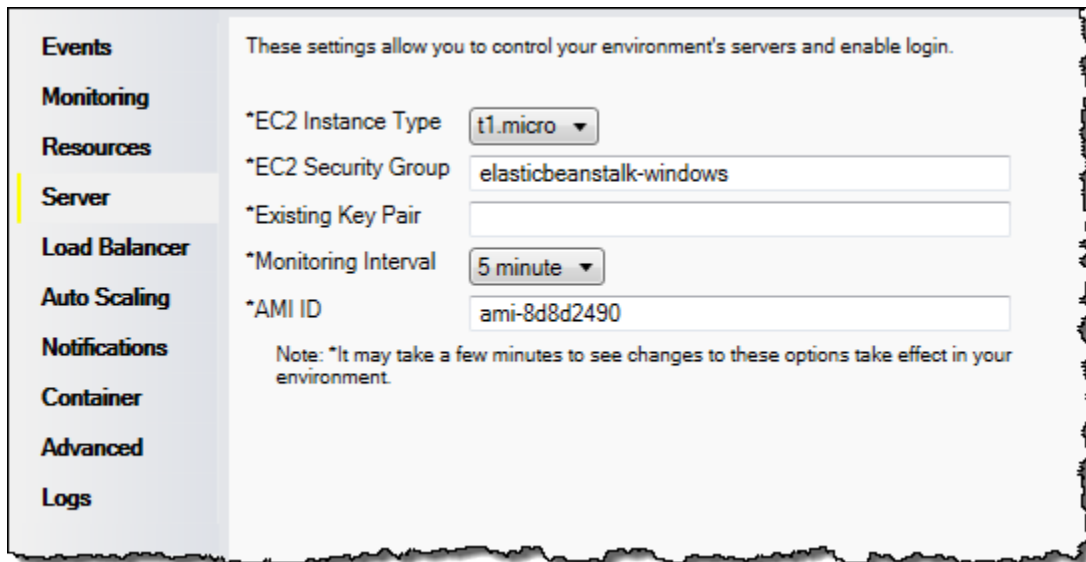
- de aplicativos
- Balanceamento de carga
- Auto Scaling
- Notificações
- Propriedades de ambiente

Configuração de instâncias de servidor EC2 usando o AWS Toolkit for Visual Studio

O Amazon Elastic Compute Cloud (Amazon EC2) é um serviço web que você usa para executar e gerenciar instâncias de servidor nos datacenters da Amazon. Você pode usar instâncias de servidor

Amazon EC2 a qualquer momento, pelo tempo que precisar e por quaisquer motivos legais. As instâncias estão disponíveis em diferentes tamanhos e configurações. Para obter mais informações, acesse [Amazon EC2](#).

É possível editar a configuração da instância do Amazon EC2 do ambiente do Elastic Beanstalk na guia Server (Servidor) do ambiente da aplicação no AWS Toolkit for Visual Studio.



Tipos de instância do Amazon EC2

Instance type (Tipo de instância) exibe os tipos de instância disponíveis para a aplicação do Elastic Beanstalk. Altere o tipo de instância para selecionar um servidor com as características (inclusive o tamanho da memória e a capacidade de CPU) mais adequadas à aplicação. Por exemplo, os aplicativos com operações muito intensas e demoradas podem exigir mais CPU ou memória.

Para obter mais informações sobre os tipos de instância do Amazon EC2 disponíveis para a aplicação do Elastic Beanstalk, consulte [Instance Types \(Tipos de instância\)](#) no Guia do usuário do Amazon Elastic Compute Cloud.

Grupos de segurança do Amazon EC2

É possível controlar o acesso à sua aplicação do Elastic Beanstalk usando um grupo de segurança do Amazon EC2. Um grupo de segurança define regras de firewall para suas instâncias. Essas regras especificam o tráfego de rede de entrada que será entregue à sua instância. Qualquer outro tráfego de entrada será descartado. Você pode modificar as regras para um grupo a qualquer momento. As novas regras são aplicadas automaticamente para todas as instâncias em execução e as iniciadas no futuro.

Você pode configurar seus grupos de segurança do Amazon EC2 usando o Console de Gerenciamento da AWS ou o AWS Toolkit for Visual Studio. Você pode especificar quais grupos de segurança do Amazon EC2 controlam o acesso à sua aplicação do Elastic Beanstalk inserindo os nomes de um ou mais grupos de segurança do Amazon EC2 (delimitados por vírgulas) na caixa de texto EC2 Security Groups (Grupos de segurança do EC2).

Note

Certifique-se de que a porta 80 (HTTP) esteja acessível a partir de 0.0.0.0/0 como o intervalo CIDR de origem, caso deseje permitir verificações de integridade do seu aplicativo. Para obter mais informações sobre as verificações de integridade, consulte [Verificações de integridade](#).

Como criar um grupo de segurança usando o AWS Toolkit for Visual Studio

1. No Visual Studio, no AWS Explorer, expanda o nó Amazon EC2 e, em seguida, clique duas vezes em Security Groups (Grupos de Segurança).
2. Clique em Create Security Group e insira um nome e uma descrição para o seu security group.
3. Clique em OK.

Para obter mais informações sobre os grupos de segurança do Amazon EC2, consulte [Usar grupos de segurança](#) no Guia do usuário do Amazon Elastic Compute Cloud.

Pares de chave do Amazon EC2

É possível fazer login com segurança nas instâncias do Amazon EC2 provisionadas para a sua aplicação do Elastic Beanstalk com um par de chaves do Amazon EC2.

Important

É necessário criar um par de chaves do Amazon EC2 e configurar as instâncias do Amazon EC2 provisionadas pelo Elastic Beanstalk para usar o par de chaves do Amazon EC2 para poder acessar essas instâncias do Amazon EC2 provisionadas pelo Elastic Beanstalk. É necessário criar seu par de chaves usando o assistente Publicar na AWS dentro do Toolkit for Visual Studio da AWS ao implantar a aplicação no Elastic Beanstalk. Se você quiser criar pares de chaves adicionais usando o Toolkit, siga as etapas abaixo. Também há a opção de configurar os pares de chave do Amazon EC2 usando o [Console de Gerenciamento da AWS](#).

Para obter instruções sobre como criar um par de chaves do Amazon EC2, consulte o [Guia de conceitos básicos do Amazon Elastic Compute Cloud](#).

A caixa de texto Existing Key Pair (Par de chaves existente) permite que você especifique o nome de um par de chaves do Amazon EC2 que pode ser usado para fazer login com segurança nas instâncias do Amazon EC2 que executam a aplicação do Elastic Beanstalk.

Para especificar o nome de um par de chaves do Amazon EC2

1. Expanda o nó Amazon EC2 e clique duas vezes em Key Pairs.
2. Clique em Create Key Pair e insira o nome do par de chaves.
3. Clique em OK.

Para obter mais informações sobre pares de chaves do Amazon EC2, acesse [Usar credenciais do Amazon EC2](#) no Guia do usuário do Amazon Elastic Compute Cloud. Para obter mais informações sobre como se conectar às instâncias do Amazon EC2, consulte [Listar e conectar com instâncias de servidor](#).

Intervalo de monitoramento

Por padrão, somente as métricas básicas do Amazon CloudWatch estão habilitadas. Elas retornam dados em intervalos de cinco minutos. Você pode habilitar métricas mais detalhadas do CloudWatch, em intervalos de um minuto, selecionando 1 minute (1 minuto) para o Monitoring Interval (Intervalo de monitoramento) na seção Server (Servidor) da guia Configuration (Configuração) para o ambiente no AWS Toolkit for Eclipse.


Note

As cobranças de serviço do Amazon CloudWatch podem se aplicar a métricas de intervalo de um minuto. Consulte [Amazon CloudWatch](#) para obter mais informações.

ID da AMI personalizado

Você pode modificar a AMI padrão usada nas instâncias do Amazon EC2 pela própria AMI personalizada inserindo o identificador da AMI personalizada na caixa Custom AMI ID (ID da AMI

personalizada) na seção Server (Servidor) da guia Configuration (Configuração) para o ambiente no AWS Toolkit for Eclipse.

 Important

O uso da AMI própria é uma tarefa avançada que deve ser feita com cuidado. Se você precisar de uma AMI personalizada, recomendamos começar com a AMI do Elastic Beanstalk padrão e depois modificá-la. Para ser considerado íntegro, o Elastic Beanstalk espera que as instâncias do Amazon EC2 atendam a um conjunto de requisitos, inclusive ter um gerenciador de host em execução. Se esses requisitos não forem atendidos, talvez o ambiente não funcione corretamente.

Configuração do Elastic Load Balancing usando o AWS Toolkit for Visual Studio

O Elastic Load Balancing é um serviço web da Amazon que ajuda você a aprimorar a disponibilidade e a escalabilidade de seu aplicativo. Este serviço facilita a distribuição de cargas de aplicativo entre duas ou mais instâncias do Amazon EC2. O Elastic Load Balancing permite disponibilidade através de redundância e é compatível com aumento de tráfego do seu aplicativo.

O Elastic Load Balancing permite que você distribua e balanceie automaticamente o tráfego de entrada dos aplicativos em todas as instâncias que você está executando. O serviço também facilita a adição de novas instâncias, caso você precise aumentar a capacidade do aplicativo.

O Elastic Beanstalk provisiona automaticamente o Elastic Load Balancing ao implantar uma aplicação. É possível editar a configuração da instância do Amazon EC2 do ambiente do Elastic Beanstalk na guia Load Balancer (Balanceador de carga) na guia do ambiente da aplicação no AWS Toolkit for Visual Studio.

These settings allow you to control the behavior of your environment's load balancer.

HTTP Listener Port: 80

HTTPS Listener Port: OFF

SSL Certificate ID:

These settings allow you to configure how Elastic Beanstalk determines whether an EC2 instance is healthy or not.

Application Health Check: /

Health Check Interval (seconds): 30 (5 - 300)

Health Check Timeout (seconds): 5 (2 - 60)

Healthy Check Count Threshold: 3 (2 - 10)

Unhealthy Check Count Threshold: 5 (2 - 10)

These settings allow you to control how your load balancer handles session cookies.

Enable Session Stickiness

Cookie Expiration Period (seconds): 0 (0 - 1000000)

As seções a seguir descrevem os parâmetros do Elastic Load Balancing que você pode configurar para o seu aplicativo.

Portas

O balanceador de carga provisionado para lidar com as solicitações da aplicação do Elastic Beanstalk envia solicitações para as instâncias do Amazon EC2 que estão executando a aplicação. O balanceador de carga provisionado pode receber solicitações em portas HTTP e HTTPS e encaminhá-las para as instâncias do Amazon EC2 em sua aplicação do AWS Elastic Beanstalk. Por padrão, o load balancer lida com solicitações na porta HTTP. Pelo menos uma das portas (HTTP ou HTTPS) deve estar ativada.

These settings allow you to control the behavior of your environment's load balancer.

HTTP Listener Port: 80

HTTPS Listener Port: OFF

SSL Certificate ID:

Important

Certifique-se de que a porta especificada não esteja bloqueada, caso contrário, os usuários não poderão se conectar à aplicação do Elastic Beanstalk.

Controle da porta HTTP

Para desativar a porta HTTP, selecione OFF em HTTP Listener Port. Para ativar a porta HTTP, você deve selecionar uma porta HTTP (por exemplo, a 80) na lista.

Note

Para acessar o ambiente usando uma porta diferente da porta 80 padrão, como a porta 8080, adicione uma escuta ao load balancer existente e configure a nova escuta nessa porta. Por exemplo, usando a [AWS CLI for Classic Load Balancers](#), digite o comando a seguir, substituindo *LOAD_BALANCER_NAME* pelo nome do balanceador de carga do Elastic Beanstalk.

```
aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
--listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
InstancePort=80"
```

Por exemplo, usando a [AWS CLI para Application Load Balancers](#), digite o comando a seguir, substituindo *LOAD_BALANCER_ARN* pelo ARN do balanceador de carga do Elastic Beanstalk.

```
aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP
--port 8080
```

Se você quiser que o Elastic Beanstalk monitore o ambiente, não remova a escuta na porta 80.

Controle da porta HTTPS

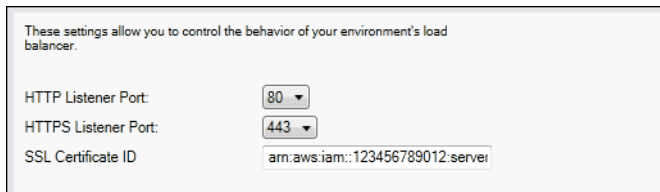
O Elastic Load Balancing oferece suporte ao protocolo HTTPS/TLS para permitir a criptografia do tráfego para conexões de clientes com o load balancer. As conexões do load balancer para as instâncias do EC2 usam criptografia de texto simples. Por padrão, a porta HTTPS é desativada.

Para ativar a porta HTTPS

1. Crie um novo certificado usando o AWS Certificate Manager (ACM), ou faça upload de um certificado e uma chave no AWS Identity and Access Management (IAM). Para obter mais informações sobre como solicitar um certificado do ACM, consulte [Solicitação de um certificado](#)

no Manual do usuário do AWS Certificate Manager. Para obter mais informações sobre a importação de certificados de terceiros para o ACM, consulte [Importação de certificados](#) no Manual do usuário do AWS Certificate Manager. Se o ACM [não estiver disponível](#) em sua região, use o AWS Identity and Access Management (IAM) para fazer upload de um certificado de terceiros. Os serviços ACM e IAM armazenarão o certificado e fornecerão um nome de recurso da Amazon (ARN) para o certificado SSL. Para obter mais informações sobre a criação e o upload de certificados para o IAM, consulte [Trabalhar com certificados de servidor](#) no Guia do usuário do IAM.

2. Especifique a porta HTTPS selecionando uma porta em HTTPS Listener Port.



These settings allow you to control the behavior of your environment's load balancer.

HTTP Listener Port:	80
HTTPS Listener Port:	443
SSL Certificate ID	arn:aws:iam::123456789012:server-certificate/abc/certs/build

3. Para SSL Certificate ID (ID de certificado SSL), insira o Nome de recurso da Amazon (ARN) do seu certificado SSL. Por exemplo, o **arn:aws:iam::123456789012:server-certificate/abc/certs/build** ou o **arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678**. Use o certificado SSL que você criou ou do qual fez upload na etapa 1.

Para desativar a porta HTTPS, selecione OFF em HTTPS Listener Port.

Verificações de integridade

A definição de verificação de integridade inclui um URL a ser consultado para saber a integridade da instância. Por padrão, o Elastic Beanstalk usa TCP: 80 para contêineres não legados e HTTP: 80 para contêineres legados. Você pode substituir o URL padrão para corresponder a um recurso existente no seu aplicativo (por exemplo, `/myapp/default.aspx`), inserindo-o na caixa Application Health Check URL. Se você substituir o URL padrão, o Elastic Beanstalk usará HTTP para consultar o recurso. Para verificar se você está usando um tipo de contêiner legado, consulte [the section called "Por que algumas versões de plataforma são marcadas como legadas?"](#)

Você pode controlar as configurações da verificação de integridade usando a seção EC2 Instance Health Check do painel Load Balancing.

These settings allow you to configure how Elastic Beanstalk determines whether an EC2 instance is healthy or not.

Application Health Check:	<input type="text" value="/"/>	
Health Check Interval (seconds):	<input type="text" value="30"/>	(5 - 300)
Health Check Timeout (seconds):	<input type="text" value="5"/>	(2 - 60)
Healthy Check Count Threshold:	<input type="text" value="3"/>	(2 - 10)
Unhealthy Check Count Threshold:	<input type="text" value="5"/>	(2 - 10)

A definição de verificação de integridade inclui um URL a ser consultado para saber a integridade da instância. Substitua o URL padrão para corresponder a um recurso existente no seu aplicativo (por exemplo, `/myapp/index.jsp`), inserindo-o na caixa Application Health Check URL.

A lista a seguir descreve os parâmetros de verificação de integridade que você pode definir para o aplicativo.

- Em Health Check Interval (seconds), insira o número de segundos que o Elastic Load Balancing aguarda entre as verificações de integridade das instâncias do Amazon EC2 do seu aplicativo.
- Em Health Check Timeout (seconds), especifique o número de segundos que o Elastic Load Balancing aguarda por uma resposta antes que ele considere que a instância não está respondendo.
- Em Healthy Check Count Threshold e Unhealthy Check Count Threshold, especifique o número de testes consecutivos de URL bem-sucedidos ou sem êxito antes que o Elastic Load Balancing altere o status de integridade da instância. Por exemplo, especificar **5** em Unhealthy Check Count Threshold significa que o URL deve retornar uma mensagem de erro ou tempo esgotado cinco vezes consecutivas antes que o Elastic Load Balancing considere que a verificação de integridade falhou.

Sessões

Por padrão, um load balancer roteia cada solicitação de forma independente para a instância de servidor com a menor carga. Por comparação, uma sticky session vincula a sessão do usuário a uma determinada instância do servidor, para que todas as solicitações desse usuário durante a sessão sejam enviadas para a mesma instância do servidor.

O Elastic Beanstalk usa cookies HTTP gerados pelo balanceador de carga quando as sticky sessions são permitidas em uma aplicação. O load balancer usa um cookie especial gerado pelo load balancer para rastrear cada solicitação na instância do aplicativo. Quando o load balancer receber uma solicitação, ele primeiro verificará se esse cookie está presente na solicitação. Se estiver, a solicitação será enviada para a instância do aplicativo especificada no cookie. Se não houver um cookie, o load balancer selecionará uma instância de aplicativo com base no algoritmo de

balanceamento de carga existente. Um cookie será inserido na resposta para vincular as solicitações subsequentes do mesmo usuário para essa instância do aplicativo. A configuração da política define a expiração do cookie, que estabelece a validade de cada cookie.

Você pode usar a seção Sessions na guia Load Balancer para especificar se deseja ou não que o load balancer do seu aplicativo permita a perdurabilidade da sessão.



The screenshot shows a configuration panel for session cookies. At the top, it reads: "These settings allow you to control how your load balancer handles session cookies." Below this, there is a checkbox labeled "Enable Session Stickiness" which is currently unchecked. Underneath, there is a text input field for "Cookie Expiration Period (seconds)" with the value "0" entered. To the right of the input field, the range "(0 - 1000000)" is displayed.

Para obter mais informações sobre o Elastic Load Balancing, consulte o [Guia do desenvolvedor do Elastic Load Balancing](#).

Configurar o Auto Scaling usando o AWS Toolkit for Visual Studio

O Amazon EC2 Auto Scaling é um serviço da Web da Amazon projetado para executar ou encerrar automaticamente instâncias do Amazon EC2 com base em triggers definidos pelo usuário. Os usuários podem configurar grupos de Auto Scaling e associar triggers a esses grupos para escalar automaticamente recursos de computação com base em métricas como uso de largura de banda ou utilização da CPU. O Amazon EC2 Auto Scaling funciona com o Amazon CloudWatch para recuperar métricas das instâncias do servidor que executam a sua aplicação.

O Amazon EC2 Auto Scaling permite que você pegue um grupo de instâncias do Amazon EC2 e defina vários parâmetros para que esse grupo aumente ou diminua automaticamente em número. O Amazon EC2 Auto Scaling pode adicionar ou remover instâncias do Amazon EC2 desse grupo para ajudar você a lidar com as alterações de tráfego em sua aplicação sem problemas.

O Amazon EC2 Auto Scaling também monitora a integridade de cada instância do Amazon EC2 que ele executa. Se alguma instância for encerrada inesperadamente, o Amazon EC2 Auto Scaling detectará o encerramento e executará uma instância de substituição. Esse recurso permite que você mantenha automaticamente um número desejado fixo de instâncias do Amazon EC2.

O Elastic Beanstalk provisiona o Amazon EC2 Auto Scaling para sua aplicação. É possível editar a configuração da instância do Amazon EC2 do ambiente do Elastic Beanstalk na guia Auto Scaling, na guia do ambiente da aplicação no AWS Toolkit for Visual Studio.

Events

Monitoring

Resources

Server

Load Balancer

Auto Scaling

Notifications

Container

Advanced

Auto-scaling automatically launches or terminates EC2 instances based on defined metrics and thresholds called triggers. Auto-scaling will also launch a new EC2 instance in the event of a failure. These settings allow you to control auto-scaling behavior.

Minimum Instance Count: (1 - 10000)

Maximum Instance Count: (1 - 10000)

Availability Zones:

Scaling Cooldown Time (seconds): (0 - 10000)

Trigger Measurement:

Trigger Statistic:

Unit of Measurement:

Measurement Period (minutes): (1 - 600)

Breach Duration (minutes): (1 - 600)

Upper Threshold: (0 - 20000000)

Upper Breach Scalement Increment:

Lower Threshold: (0 - 20000000)

Lower Breach Scalement Increment:

A seção a seguir discute como configurar os parâmetros de Auto Scaling para seu aplicativo.

Iniciar a configuração

É possível editar a configuração de execução para controlar como sua aplicação do Elastic Beanstalk provisiona os recursos do Amazon EC2 Auto Scaling.

As caixas Minimum Instance Count (Contagem de instância mínima) e Maximum Instance Count (Contagem de instância máxima) permitem que você especifique os tamanhos mínimo e máximo do grupo de Auto Scaling que a aplicação do Elastic Beanstalk usa.

Auto-scaling automatically launches or terminates EC2 instances based on defined metrics and thresholds called triggers. Auto-scaling will also launch a new EC2 instance in the event of a failure. These settings allow you to control auto-scaling behavior.

Minimum Instance Count: (1 - 10000)

Maximum Instance Count: (1 - 10000)

Availability Zones:

Scaling Cooldown Time (seconds): (0 - 10000)

Note

Para manter um número fixo de instâncias do Amazon EC2, defina Minimum Instance Count e Maximum Instance Count com o mesmo valor.

A caixa Availability Zones permite que você especifique o número de zonas de disponibilidade em que deseja que as suas instâncias do Amazon EC2 estejam. É importante definir esse número se quiser criar aplicativos tolerantes a falhas. Se uma Zona de disponibilidade ficar inativa, suas instâncias continuarão sendo executadas em suas outras Zonas de disponibilidade.

Note

No momento, não é possível especificar a zona de disponibilidade na qual sua instância estará.

Acionadores

Um trigger é um mecanismo do Amazon EC2 Auto Scaling que você define para dizer ao sistema quando você deseja aumentar (expandir) o número de instâncias e quando você deseja diminuir (reduzir) o número de instâncias. Você pode configurar os triggers para acionar quaisquer métricas publicadas no Amazon CloudWatch, como utilização de CPU, e determinar se as condições que você especificou foram atendidas. Quando o limite superior ou inferior das condições que você especificou para a métrica forem violados para o período especificado, o trigger iniciará um processo de longa duração chamado de ação de escalabilidade.

É possível definir um acionador de escalabilidade para a aplicação do Elastic Beanstalk usando o AWS Toolkit for Visual Studio.

Trigger Measurement:	<input type="text" value="NetworkOut"/>
Trigger Statistic:	<input type="text" value="Average"/>
Unit of Measurement:	<input type="text" value="Bytes"/>
Measurement Period (minutes):	<input type="text" value="5"/> (1 - 600)
Breach Duration (minutes):	<input type="text" value="5"/> (1 - 600)
Upper Threshold:	<input type="text" value="6000000"/> (0 - 20000000)
Upper Breach Scalement Increment:	<input type="text" value="1"/>
Lower Threshold:	<input type="text" value="2000000"/> (0 - 20000000)
Lower Breach Scalement Increment:	<input type="text" value="-1"/>

Os triggers do Amazon EC2 Auto Scaling funcionam observando uma determinada métrica do Amazon CloudWatch de uma instância. Os triggers incluem utilização de CPU, tráfego de rede e a atividade do disco. Use a configuração Trigger Measurement para selecionar uma métrica para o trigger.

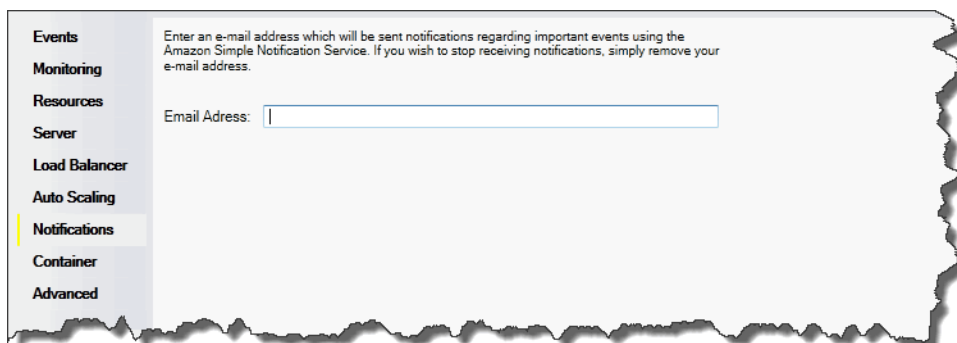
A lista a seguir descreve os parâmetros do acionador que você pode configurar usando o Console de Gerenciamento da AWS.

- Você pode especificar quais estatísticas o trigger deve usar. Você pode selecionar Minimum, Maximum, Sum ou Average em Trigger Statistic.
- Em Unit of Measurement, especifique a unidade de medida do trigger.
- O valor na caixa Measurement Period (Período de medição) especifica a frequência com que o Amazon CloudWatch mede as métricas do trigger. A Breach Duration é a quantidade de tempo que uma métrica pode ficar além do limite definido (conforme especificado em Upper Threshold e Lower Threshold) antes que o trigger seja acionado.
- Em Upper Breach Scale Increment e Lower Breach Scale Increment, especifique quantas instâncias do Amazon EC2 devem ser adicionadas ou removidas ao realizar uma ação de escalabilidade.

Para obter mais informações sobre o Amazon EC2 Auto Scaling, consulte a seção Amazon EC2 Auto Scaling na [documentação do Amazon Elastic Compute Cloud](#).

Configurar as notificações usando o AWS Toolkit for Visual Studio

O Elastic Beanstalk usa o Amazon Simple Notification Service (Amazon SNS) para notificar você de eventos importantes que afetam sua aplicação. Para permitir notificações do Amazon SNS, basta inserir seu endereço de e-mail na caixa Email Address. Para desativar essas notificações, remova seu endereço de e-mail da caixa.



Configurar contêineres .NET usando o AWS Toolkit for Visual Studio

O painel Container/.NET Options permite ajustar o comportamento de suas instâncias do Amazon EC2 e habilitar ou desabilitar a rotação de logs do Amazon S3. Você pode usar o AWS Toolkit for Visual Studio para configurar as informações de contêiner.

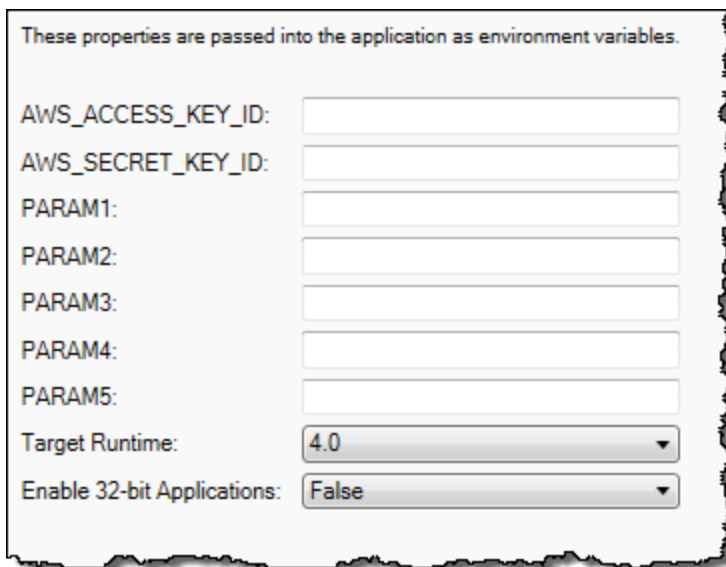
Note

É possível modificar as definições de configuração com tempo de inatividade zero, basta trocar o CNAME dos ambientes. Para obter mais informações, consulte [Implantações azuis/verdes com o Elastic Beanstalk](#).

Se desejar, você pode estender o número de parâmetros. Para obter mais informações sobre como estender os parâmetros, consulte [Configurações de opção](#).

Como acessar o painel de opções de contêiner/.NET para a aplicação do Elastic Beanstalk

1. No AWS Toolkit for Visual Studio, expanda o nó Elastic Beanstalk e o nó da aplicação.
2. No AWS Explorer, clique duas vezes no ambiente do Elastic Beanstalk.
3. Na parte inferior do painel Overview, clique na guia Configuration.
4. Em Container, configure as opções de contêiner.



These properties are passed into the application as environment variables.

AWS_ACCESS_KEY_ID:	<input type="text"/>
AWS_SECRET_KEY_ID:	<input type="text"/>
PARAM1:	<input type="text"/>
PARAM2:	<input type="text"/>
PARAM3:	<input type="text"/>
PARAM4:	<input type="text"/>
PARAM5:	<input type="text"/>
Target Runtime:	<input type="text" value="4.0"/>
Enable 32-bit Applications:	<input type="text" value="False"/>

Opções de contêiner .NET

Você pode escolher a versão do .NET Framework para o seu aplicativo. Escolha 2.0 ou 4.0 para o Target runtime. Selecione Enable 32-bit Applications se quiser permitir aplicativos de 32 bits.

Configurações do aplicativo

A seção Application Settings permite que você defina as variáveis de ambiente que você pode ler do código do seu aplicativo.

These properties are passed into the application as environment variables.

AWS_ACCESS_KEY_ID:	<input type="text"/>
AWS_SECRET_KEY_ID:	<input type="text"/>
PARAM1:	<input type="text"/>
PARAM2:	<input type="text"/>
PARAM3:	<input type="text"/>
PARAM4:	<input type="text"/>
PARAM5:	<input type="text"/>

Gerenciar contas

Se quiser configurar diferentes contas da AWS para executar tarefas diversas, como testes, preparação e produção, você poderá adicionar, editar e excluir contas usando o AWS Toolkit for Visual Studio.

Para gerenciar várias contas

1. No Visual Studio, no menu View (Exibir), clique em AWS Explorer.
2. Ao lado da lista Conta, clique no botão Adicionar Conta.



A caixa de diálogo Adicionar Conta é exibida.

Add Account

Display Name:

Access Key:

Secret Key:
Account Number*:

This account information can be found at: <http://aws.amazon.com/security-credentials>
* Account Number is an optional field used for constructing amazon resource names (ARN).

OK Cancel

3. Preencha as informações solicitadas.

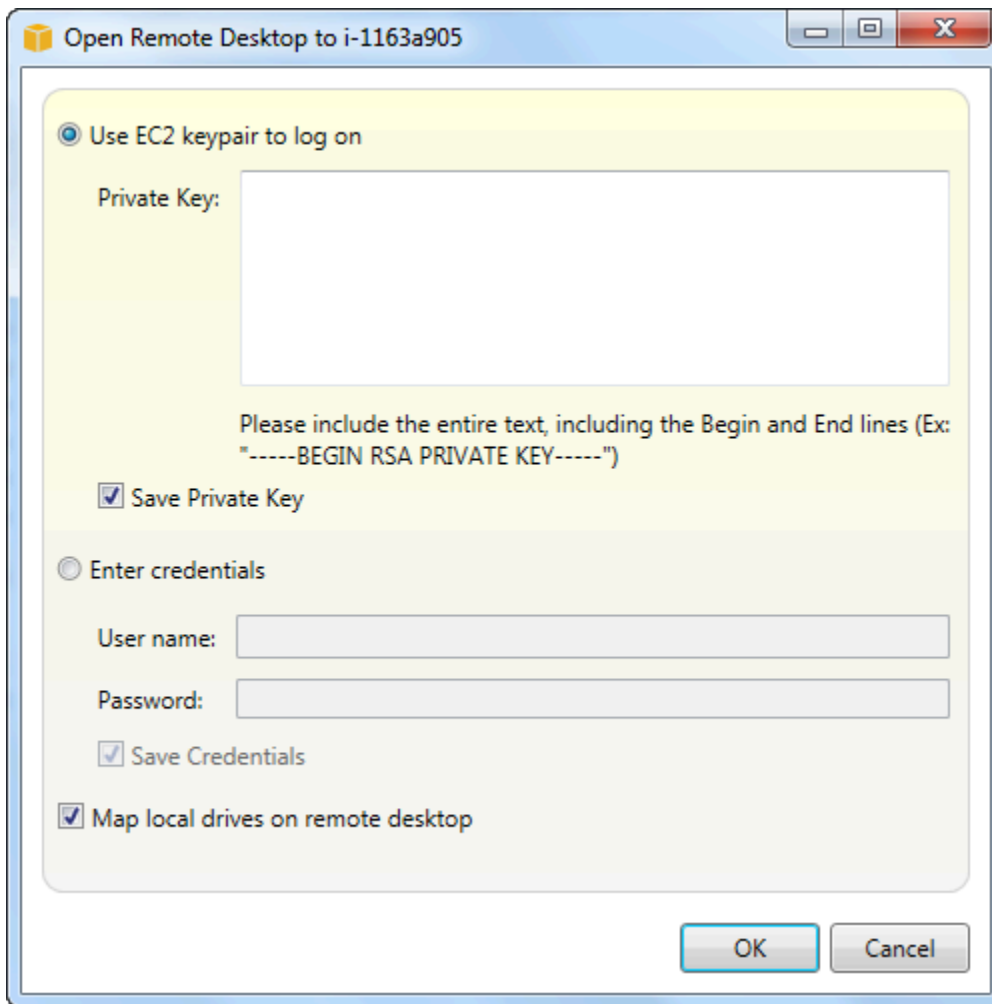
4. As informações da sua conta aparecem na guia AWS Explorer. Quando você publicar no Elastic Beanstalk, poderá selecionar a conta que deseja usar.

Listar e conectar com instâncias de servidor

É possível visualizar uma lista de instâncias do Amazon EC2 executando o ambiente da aplicação do Elastic Beanstalk por meio do AWS Toolkit for Visual Studio ou do Console de Gerenciamento da AWS. Você pode se conectar a essas instâncias usando a conexão de desktop remoto. Para obter informações sobre como listar e conectar-se a instâncias de servidor usando o Console de Gerenciamento da AWS, consulte [Listar e conectar com instâncias de servidor](#). A seção a seguir apresenta as etapas para você visualizar e se conectar às instâncias do servidor usando o AWS Toolkit for Visual Studio.

Para visualizar e conectar-se a instâncias do Amazon EC2 em um ambiente

1. No Visual Studio, no AWS Explorer, expanda o nó Amazon EC2 e clique duas vezes em Instances (Instâncias).
2. Clique com o botão direito do mouse no ID da instância do Amazon EC2 em execução no balanceador de carga da aplicação na coluna Instance (Instância) e selecione Open Remote Desktop (Abrir desktop remoto) no menu de contexto.



3. Selecione Use EC2 keypair to log on e cole o conteúdo do arquivo de chave privada que você usou para implantar o seu aplicativo na caixa Private key. Como alternativa, insira seu nome de usuário e senha nas caixas de texto Nome de usuário e Senha.

Note

Se o par de chaves estiver armazenado no Toolkit, a caixa de texto não será exibida.

4. Clique em OK.

Monitorar a integridade do aplicativo

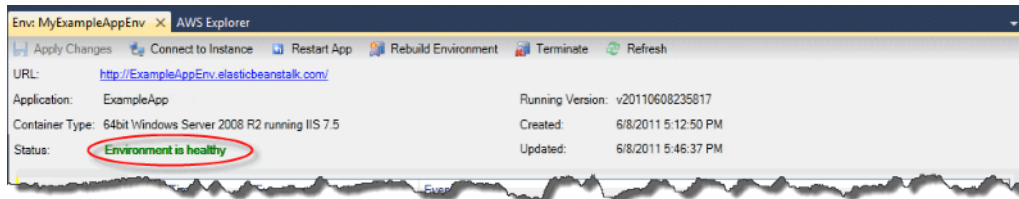
Quando você executa um site de produção, é importante saber que seu aplicativo está disponível e respondendo às solicitações. Para ajudar a monitorar a capacidade de resposta da aplicação, o

Elastic Beanstalk dispõe de recursos que monitoram as estatísticas sobre a aplicação e criam alertas que são acionados quando os limites são excedidos.

Para obter informações sobre o monitoramento de integridade fornecido pelo Elastic Beanstalk, consulte [Relatórios de integridade básica](#).

É possível acessar as informações operacionais sobre sua aplicação usando o AWS Toolkit for Visual Studio ou o Console de Gerenciamento da AWS.

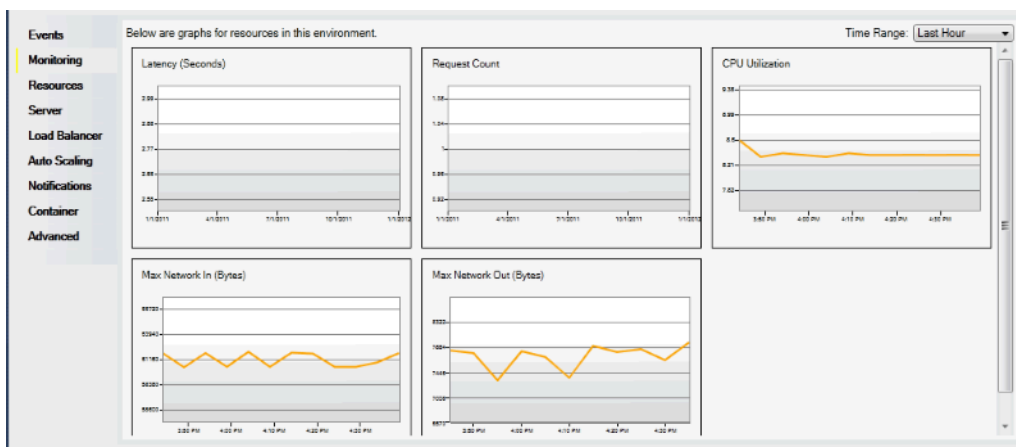
O toolkit exibe o status do ambiente e a integridade do aplicativo no campo Status.



Para monitorar a integridade do aplicativo

1. No AWS Toolkit for Visual Studio, em AWS Explorer, expanda o nó Elastic Beanstalk e, depois, o nó da aplicação.
2. Clique com o botão direito do mouse no ambiente do Elastic Beanstalk e, depois, clique em View Status (Visualizar status).
3. Na guia do seu ambiente de aplicativos, clique em Monitoramento.

O painel Monitoramento inclui um conjunto de gráficos que mostra o uso de recursos do seu ambiente de aplicativos específico.



Note

Por padrão, o intervalo de tempo é definido como a última hora. Para modificar essa configuração, na lista Período, clique em um período diferente.

Use o AWS Toolkit for Visual Studio ou o Console de Gerenciamento da AWS para visualizar eventos associados a sua aplicação.

Para visualizar eventos do aplicativo

1. No AWS Toolkit for Visual Studio, em AWS Explorer, expanda o nó Elastic Beanstalk e o nó da aplicação.
2. Clique com o botão direito do mouse no ambiente do Elastic Beanstalk em AWS Explorer e, depois, clique em View Status (Visualizar status).
3. Na guia do seu ambiente de aplicativos, clique em Eventos.



Event Time	Event Type	Version Label	Event Details
12/2/2011 3:43:19 PM	INFO	v20111202232102	Environment update completed successfully.
12/2/2011 3:43:19 PM	INFO	v20111202232102	New application version was deployed to running EC2 instances.
12/2/2011 3:43:06 PM	INFO	v20111202232102	Waiting for 2 seconds while EC2 instances download the updated application version.
12/2/2011 3:43:04 PM	INFO	v20111202232102	Deploying version v20111202232102 to 1 instance(s).
12/2/2011 3:42:59 PM	INFO	v20111202230009	Environment update is starting.
12/2/2011 3:30:33 PM	INFO	v20111202230009	Environment health has transitioned from RED to GREEN.
12/2/2011 3:29:37 PM	WARN	v20111202230009	Environment health has been set to RED.
12/2/2011 3:28:39 PM	INFO	v20111202230009	Launched environment: MyExampleAppEnv. However, there were issues during launch. See event log for details.
12/2/2011 3:28:35 PM	INFO	v20111202230009	Exceeded maximum amount time to wait for the application to become available. Setting environment Ready.
12/2/2011 3:19:13 PM	INFO	v20111202230009	Adding instance i-93d6e6f0 to your environment.
12/2/2011 3:18:50 PM	INFO	v20111202230009	Added EC2 instance i-93d6e6f0 to Auto Scaling Group 'awseb-MyExampleAppEnv-5y33OGVvOm'.
12/2/2011 3:18:47 PM	INFO	v20111202230009	An EC2 instance has been launched. Waiting for it to be added to Auto Scaling...
12/2/2011 3:18:34 PM	INFO	v20111202230009	Waiting for an EC2 instance to be launched...
12/2/2011 3:18:33 PM	INFO	v20111202230009	Adding Auto Scaling Group 'awseb-MyExampleAppEnv-5y33OGVvOm' to your environment.
12/2/2011 3:18:33 PM	INFO	v20111202230009	Added URLCheck healthcheck for 'http://MyExampleAppEnv.elasticbeanstalk.com:80/
12/2/2011 3:18:31 PM	INFO	v20111202230009	Created Auto Scaling trigger named: awseb-MyExampleAppEnv-5y33OGVvOm.
12/2/2011 3:18:30 PM	INFO	v20111202230009	Created Auto Scaling group named: awseb-MyExampleAppEnv-5y33OGVvOm.
12/2/2011 3:18:30 PM	INFO	v20111202230009	Created Auto Scaling launch configuration named: awseb-MyExampleAppEnv-JDwosDtlJA.
12/2/2011 3:18:29 PM	INFO	v20111202230009	Created load balancer named: awseb-MyExampleAppEnv.
12/2/2011 3:18:28 PM	INFO	v20111202230009	Created security group named: elasticbeanstalk-windows.
12/2/2011 3:18:28 PM	INFO	v20111202230009	Using elasticbeanstalk-us-east-1-049020475370 as Amazon S3 storage bucket for environment data.

Implantar aplicações do Elastic Beanstalk em .NET usando a ferramenta de implantação

O AWS Toolkit for Visual Studio inclui uma ferramenta de implantação, uma ferramenta da linha de comando com a mesma funcionalidade do assistente de implantação no AWS Toolkit. Você pode usar a ferramenta de implantação em seu pipeline de compilações ou em outros scripts para automatizar as implantações no Elastic Beanstalk.

A ferramenta de implantação oferece suporte a implantações iniciais e reimplementações. Se você já implantou seu aplicativo usando a ferramenta de implantação, pode reimplantá-lo usando

o assistente de implantação no Visual Studio. Da mesma forma, se você implantou usando o assistente, pode reimplantar usando a ferramenta de implantação.

Note

A ferramenta de implantação não aplica os [valores recomendados](#) às opções de configuração, como o console ou a EB CLI. Use os [arquivos de configuração](#) para garantir que as configurações necessárias sejam configuradas quando você iniciar seu ambiente.

Este capítulo orienta você pelo processo de implantação de uma aplicação .NET demonstrativa no Elastic Beanstalk usando a ferramenta de implantação e, depois, pelo processo de reimplantação dela usando uma implantação incremental. Para ver mais detalhes sobre a ferramenta de implantação, incluindo as opções de parâmetros, consulte [Ferramenta de implantação](#).

Pré-requisitos

Para usar a ferramenta de implantação, é necessário instalar o AWS Toolkit for Visual Studio. Para obter informações sobre os pré-requisitos e as instruções de instalação, consulte [AWS Toolkit for Microsoft Visual Studio](#).

A ferramenta de implantação costuma ser instalada em um dos seguintes diretórios no Windows:

32 bits	64 bits
C:\Program Files\AWS Tools\Deployment Tool\awsdeploy.exe	C:\Program Files (x86)\AWS Tools\Deployment Tool\awsdeploy.exe

Implantar no Elastic Beanstalk

Para implantar a aplicação demonstrativa no Elastic Beanstalk usando a ferramenta de implantação, você precisa primeiro modificar o arquivo de configuração `ElasticBeanstalkDeploymentSample.txt`, que é fornecido no diretório `Samples`. Esse arquivo de configuração contém as informações necessárias para implantar a aplicação, incluindo nome e versão da aplicação, nome do ambiente e as credenciais de acesso da AWS. Após modificar o arquivo de configuração, você poderá usar a linha de comando para implantar o aplicativo de

exemplo. O arquivo de implantação Web é carregado para o Amazon S3 e registrado como uma nova versão da aplicação com o Elastic Beanstalk. A implantação do aplicativo leva alguns minutos. Quando o ambiente estiver íntegro, a ferramenta de implantação emitirá um URL para o aplicativo em execução.

Como implantar uma aplicação .NET no Elastic Beanstalk

1. No subdiretório Samples onde a ferramenta de implantação está instalada, abra `ElasticBeanstalkDeploymentSample.txt` e insira a chave de acesso da AWS e a chave secreta da AWS, conforme mostrado no exemplo a seguir.

```
### AWS Access Key and Secret Key used to create and deploy the application
instance
AWSAccessKey = AKIAIOSFODNN7EXAMPLE
AWSSecretKey = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

Note

Para acesso à API, é necessário ter um ID de chave de acesso e uma chave de acesso secreta. Use as chaves de acesso do usuário do IAM em vez das chaves de acesso da Usuário raiz da conta da AWS. Para obter mais informações sobre a criação de chaves de acesso, consulte [Gerenciar chaves de acesso para usuários do IAM](#) no Guia do usuário do IAM.

2. No prompt de linha de comando, digite o seguinte:

```
C:\Program Files (x86)\AWS Tools\Deployment Tool>awsdeploy.exe /w Samples
\ElasticBeanstalkDeploymentSample.txt
```

A implantação do aplicativo leva alguns minutos. Se a implantação for bem-sucedida, você verá a mensagem, `Application deployment completed; environment health is Green.`

Note

Se você receber o seguinte erro, o CNAME já existe.

```
[Error]: Deployment to AWS Elastic Beanstalk failed with exception: DNS name
(MyAppEnv.elasticbeanstalk.com) is not available.
```

Como o CNAME deve ser exclusivo, você precisa alterar `Environment.CNAME` em `ElasticBeanstalkDeploymentSample.txt`.

3. No navegador da Web, acesse o URL do seu aplicativo em execução. O formato do URL é `<CNAME.elasticbeanstalk.com>` (e., `MyAppEnv.elasticbeanstalk.com`).

Migrar a aplicação .NET no local para o Elastic Beanstalk

Se você está pensando em migrar sua aplicação .NET de servidores no local para a Amazon Web Services (AWS), o .NET Migration Assistant para o AWS Elastic Beanstalk pode ser útil. O assistente é um utilitário interativo do PowerShell que migra uma aplicação .NET do servidor Windows com o IIS em execução on-premises para o AWS Elastic Beanstalk. O assistente pode migrar um site inteiro para o Elastic Beanstalk com nenhuma ou poucas alterações.

Para obter mais informações sobre o .NET Migration Assistant para AWS Elastic Beanstalk e sobre como fazer download dele, consulte o repositório <https://github.com/aws-labs/windows-web-app-migration-assistant> no GitHub.

Se o aplicativo incluir bancos de dados do Microsoft SQL Server, a documentação do assistente no GitHub incluirá várias opções para migrá-los.

Recursos

Você tem à sua disposição diversas fontes de ajuda adicional durante o desenvolvimento de seus aplicativos .NET:

Recurso	Descrição
.Fórum de desenvolvimento .NET	Publique suas perguntas e receba comentários.
.Centro do desenvolvedor de .NET	Um local central para acessar código de exemplo, documentação, ferramentas e outros recursos.
AWS Documentação do SDK for .NET	Saiba mais sobre a configuração do SDK e a execução de exemplos de código, recursos do SDK e informações detalhadas sobre as operações de API do SDK.

Implantar aplicações Node.js no Elastic Beanstalk

Com o AWS Elastic Beanstalk para Node.js, é fácil implantar, gerenciar e escalar suas aplicações web Node.js usando a Amazon Web Services. O Elastic Beanstalk para Node.js está disponível para qualquer pessoa que desenvolva ou hospede uma aplicação Web usando Node.js. Este capítulo fornece instruções passo a passo para implantar a aplicação Web Node.js no Elastic Beanstalk usando o console de gerenciamento do Elastic Beanstalk e fornece demonstrações para tarefas comuns, como a integração do banco de dados e como trabalhar com a framework do Express.

Depois de implantar a aplicação do Elastic Beanstalk, é possível continuar a usar a EB CLI para gerenciar a aplicação e o ambiente, ou usar o console do Elastic Beanstalk, AWS CLI ou APIs.

Tópicos

- [Conceitos básicos do Node.js no Elastic Beanstalk](#)
- [Configurar seu ambiente de desenvolvimento Node.js](#)
- [Usar a plataforma Node.js do Elastic Beanstalk](#)
- [Implantar uma aplicação Express no Elastic Beanstalk](#)
- [Implantar uma aplicação Express com clustering no Elastic Beanstalk](#)
- [Implantar uma aplicação Node.js com o DynamoDB no Elastic Beanstalk](#)
- [Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicação Node.js](#)
- [Recursos](#)

Os tópicos neste capítulo pressupõem que você tem um nível básico de conhecimento sobre ambientes do Elastic Beanstalk. Se você não tiver usado o Elastic Beanstalk antes, tente o [tutorial de conceitos básicos](#) para saber os princípios.

Conceitos básicos do Node.js no Elastic Beanstalk

Para começar a usar as aplicações Node.js no AWS Elastic Beanstalk, tudo o que você precisa é de um [pacote de fonte](#) de aplicação para fazer upload como a primeira versão da aplicação e implantar em um ambiente. Quando você cria um ambiente, o Elastic Beanstalk aloca todos os recursos da AWS necessários para executar uma aplicação Web altamente escalável.

Iniciar um ambiente com um aplicativo Node.js de exemplo

O Elastic Beanstalk fornece aplicações demonstrativas de página única para cada plataforma e também exemplos mais complexos que mostram o uso de recursos adicionais da AWS, como o Amazon RDS e linguagem ou recursos e APIs específicos da plataforma.

Note

Siga as etapas no arquivo README .md do pacote de origem para implantá-lo.

Amostras

Tipo de ambiente	Pacote de origem	Descrição
Servidor Web	nodejs.zip	Aplicativo de página única. Use o procedimento em Criar um aplicativo de exemplo para iniciar este exemplo.
Servidor Web com o Amazon RDS	nodejs-ex-ample-express-rds.zip	Aplicativo de log de caminhada que usa a framework Express e um Amazon Relational Database Service (RDS). Tutorial
Servidor Web com o Amazon ElastiCache	nodejs-ex-ample-express-elasticache.zip	Aplicação Web Express que usa o Amazon ElastiCache para clustering. O clustering aprimora a alta disponibilidade, a performance e a segurança do seu aplicativo web. Tutorial
Servidor Web com DynamoDB,	nodejs-ex-ample-	Site Express que coleta informações de contato dos usuários para uma nova campanha de marketing da empresa. Use a AWS SDK for JavaScript in Node.js

Tipo de ambiente	Pacote de origem	Descrição
Amazon SNS e Amazon SQS	dyn amo.zip	para gravar entradas em uma tabela do DynamoDB e arquivos de configuração do Elastic Beanstalk para criar recursos no DynamoDB, Amazon SNS e Amazon SQS. Tutorial

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo completamente diferente a qualquer momento. A implantação de uma nova versão do aplicativo é muito rápida, pois não é necessário provisionar ou reiniciar as instâncias do EC2. Para obter detalhes sobre a implantação de aplicativo, consulte [Implantar uma nova versão do seu aplicativo](#).

Depois de implantar um ou dois aplicativos de exemplo e quando estiver pronto para começar a desenvolver e executar aplicativos Node.js localmente, consulte [a próxima seção](#) para configurar um ambiente de desenvolvimento Node.js com todas as ferramentas e bibliotecas necessárias.

Configurar seu ambiente de desenvolvimento Node.js

Configure um ambiente de desenvolvimento Node.js para testar sua aplicação localmente antes de implantá-lo no AWS Elastic Beanstalk. Este tópico descreve as etapas de configuração do ambiente de desenvolvimento e inclui links para páginas de instalação de ferramentas úteis.

Para ver as etapas de configuração comuns e as ferramentas que se aplicam a todas as linguagens, consulte [Configurar a máquina de desenvolvimento](#).

Tópicos

- [Instale o Node.js](#).
- [Confirmar instalação do NPM](#)
- [Instale a AWS SDK para Node.js](#)
- [Instale o gerador Express](#)
- [Configurar uma estrutura e um servidor Express](#)

Instale o Node.js.

Instale Node.js para executar aplicativos Node.js localmente. Se você não tiver uma preferência, obtenha a versão mais recente compatível com o Elastic Beanstalk. Consulte [Node.js](#) no documento Plataformas do AWS Elastic Beanstalk para obter uma lista das versões suportadas.

Faça download do Node.js em nodejs.org.

Confirmar instalação do NPM

O Node.js usa um gerenciador de pacotes do NPM que ajuda a instalar ferramentas e frameworks de trabalho para uso em seu aplicativo. Como o NPM é distribuído com Node.js, você o instalará automaticamente quando baixar e instalar o Node.js. Para confirmar que você tem o NPM instalado, você pode executar o seguinte comando:

```
$ npm -v
```

Para obter mais informações sobre o NPM, visite o site [npmjs](https://npmjs.com).

Instale a AWS SDK para Node.js

Se você precisa gerenciar os recursos da AWS a partir da sua aplicação, instale a AWS SDK for JavaScript in Node.js. Instale o SDK com o NPM:

```
$ npm install aws-sdk
```

Visite a página inicial [AWS SDK for JavaScript in Node.js](#) para obter mais informações.

Instale o gerador Express

O Express é um framework de aplicativo web que é executado no Node.js. Para usá-lo, primeiro instale o aplicativo de linha de comando do gerador Express. Depois que o gerador Express estiver instalado, você poderá executar o comando `express` para gerar uma estrutura de projeto base para sua aplicação web. Depois que o projeto base, os arquivos e as dependências estiverem instalados, você poderá iniciar um servidor Express local em sua máquina de desenvolvimento.

Note

- Estas etapas orientam você na instalação do gerador Express em um sistema operacional Linux.

- Para Linux, dependendo do seu nível de permissão para diretórios do sistema, pode ser necessário prefixar alguns desses comandos com `sudo`.

Para instalar o gerador Express em seu ambiente de desenvolvimento

1. Crie um diretório de trabalho para sua estrutura e servidor Express.

```
~$ mkdir node-express  
~$ cd node-express
```

2. Instale o Express globalmente para ter acesso ao comando `express`.

```
~/node-express$ npm install -g express-generator
```

3. Dependendo de seu sistema operacional, pode ser necessário configurar seu caminho para executar o comando `express`. A saída da etapa anterior fornece informações se você precisar definir sua variável de caminho. O seguinte é um exemplo para Linux.

```
~/node-express$ export PATH=$PATH:/usr/local/share/npm/bin/express
```

Ao seguir os tutoriais deste capítulo, você precisará executar o comando `express` de diferentes diretórios. Cada tutorial configura uma estrutura básica do projeto Express em seu próprio diretório.

Agora você instalou o gerador de linha de comando Express. Você pode usá-lo para criar um diretório de estrutura para seu aplicativo da web, configurar dependências e iniciar o servidor de aplicativos da web. Em seguida, seguiremos as etapas para fazer isso no diretório `node-express` que criamos.

Configurar uma estrutura e um servidor Express

Siga estas etapas para criar os diretórios e conteúdos básicos da estrutura do Express. Os tutoriais neste capítulo também incluem estas etapas para configurar a estrutura básica do Express em cada um dos diretórios de aplicativos do tutorial.

Configurar uma estrutura e um servidor Express

1. Execute o comando `express`. Isso gera `package.json`, `app.js` e alguns diretórios.


```
~/node-express$ express
```

Quando solicitado, digite **y** se você deseja continuar.

2. Configure as dependências locais.

```
~/node-express$ npm install
```

3. Verifique se o servidor de aplicativos da web é inicializado.

```
~/node-express$ npm start
```

Você deve ver saída semelhante a:

```
> nodejs@0.0.0 start /home/local/user/node-express  
> node ./bin/www
```

Por padrão, o servidor é executado na porta 3000. Para testá-lo, execute `curl http://localhost:3000` em outro terminal ou abra um navegador no computador local e insira o endereço URL `http://localhost:3000`.

Pressione Ctrl+C para interromper o servidor.

Usar a plataforma Node.js do Elastic Beanstalk

A plataforma Node.js do AWS Elastic Beanstalk é um conjunto de [versões da plataforma](#) para aplicações Web Node.js que são executadas por trás de um servidor proxy NGINX.

O Elastic Beanstalk fornece [opções de configuração](#) que podem ser usadas para personalizar o software que é executado nas instâncias do EC2 no ambiente Elastic Beanstalk. É possível [configurar as variáveis de ambiente](#) necessárias à aplicação, habilitar a alternância de logs para o Amazon S3 e mapear pastas da origem da aplicação que contenham arquivos estáticos para caminhos atendidos pelo servidor de proxy.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para [modificar a configuração de um ambiente em execução](#). Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar [configurações salvas](#) para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir [arquivos de configuração](#). As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

É possível [incluir um arquivo Package.json](#) no pacote de fontes para instalar pacotes durante a implantação, fornecer um comando iniciar e especificar a versão do Node.js que você deseja que a aplicação use. Você pode incluir um [arquivo npm-shrinkwrap.json](#) para bloquear versões de dependência.

A plataforma Node.js inclui um servidor proxy para servir ativos estáticos, encaminhar tráfego para a aplicação e compactar respostas. Você pode [estender ou substituir a configuração de proxy padrão](#) para cenários avançados.

Existem várias opções para iniciar seu aplicativo. Você pode adicionar um [Procfile](#) ao pacote de fontes para especificar o comando que inicia a aplicação. Quando você não fornece um `Procfile`, mas fornece um arquivo `package.json`, o Elastic Beanstalk executa `npm start`. Se você não fornecer nenhum dos dois, o Elastic Beanstalk procurará o arquivo `app.js` ou `server.js`, nessa ordem, e o executará.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console. Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte [Opções de configuração](#).

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte [the section called “Estender plataformas Linux”](#).


Configurar o ambiente Node.js

Você pode usar as configurações da plataforma Node.js para aprimorar ainda mais o comportamento das instâncias do Amazon EC2. É possível editar a configuração da instância do Amazon EC2 do ambiente do Elastic Beanstalk usando o console do Elastic Beanstalk.

Você pode usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3 e configurar variáveis que sua aplicação pode ler no ambiente.

Para configurar o ambiente Node.js no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Opções de contêiner

É possível definir estas opções específicas da plataforma:

- Proxy server (Servidor de proxy): o servidor de proxy a ser usado nas instâncias de ambiente. Por padrão, o NGNIX é usado.

Opções de log

A seção Log Options (Opções de log) tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Enable log file rotation to Amazon S3 (Habilitar a rotação de arquivos de log para o Amazon S3): especifica se os arquivos de log das instâncias do Amazon EC2 da aplicação serão copiados para o bucket do Amazon S3 associado a ela.

Arquivos estáticos

Para melhorar a performance, você pode usar a seção Static files (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo

no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte [the section called “Arquivos estáticos”](#).

Propriedades de ambiente

Use a seção Propriedades do Ambiente para especificar definições de configuração de ambientes nas instâncias do Amazon EC2 que estão executando sua aplicação. Essas definições são passadas para o aplicativo como pares de chave-valor.

Dentro do ambiente Node.js que é executado no AWS Elastic Beanstalk, você pode acessar as variáveis de ambiente executando `process.env.ENV_VARIABLE`.

```
var endpoint = process.env.API_ENDPOINT
```

A plataforma Node.js define a variável de ambiente `PORT` como a porta para a qual o servidor proxy direciona o tráfego. Para obter mais informações, consulte [Configurar o servidor de proxy](#).

Consulte [Propriedades de ambientes e outras configurações de software](#) para obter mais informações.

Configurar um ambiente Node.js do Amazon Linux AMI (anterior ao Amazon Linux 2)

As categorias a seguir de configuração de software do console só são compatíveis com um ambiente Node.js do Elastic Beanstalk que use uma versão da plataforma Amazon Linux AMI (anterior ao Amazon Linux 2).

Observações

- As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). As ramificações de plataforma AL2023/AL2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem definições de configuração diferentes.
- Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023

atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Opções de contêiner: Amazon Linux AMI (AL1)

Na página de configuração, especifique o seguinte:

- Proxy server (Servidor proxy): especifica o servidor Web a ser usado para conexões do proxy com o Node.js. Por padrão, o NGINX é usado. Se você selecionar none (nenhum), os mapeamentos de arquivos não terão efeito e a compactação GZIP será desabilitada.
- Node.js version (Versão do): especifica a versão do Node.js. Para obter uma lista das versões compatíveis do Node.js, consulte [Node.js](#), no guia Plataformas AWS Elastic Beanstalk.
- GZIP compression (Compactação): especifica se a compactação GZIP está habilitada. Por padrão, a compactação GZIP está habilitada.
- Node command (Comando de nó): permite inserir o comando usado para iniciar a aplicação Node.js. Uma string vazia (o padrão) significa que o Elastic Beanstalk usa `app.js`, em seguida `server.js` e depois `npm start`.

Namespace de configuração do Node.js

Você pode usar um [arquivo de configuração](#) para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. As opções de configuração podem ser definidas pelo serviço Elastic Beanstalk ou pela plataforma utilizada, e são organizadas em namespaces.

É possível escolher o proxy a ser usado nas instâncias do seu ambiente usando o namespace `aws:elasticbeanstalk:environment:proxy`. O exemplo a seguir configura o ambiente para usar o servidor proxy Apache HTTPD.

Example `.ebextensions/nodejs-settings.config`

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
```

É possível configurar o proxy para servir arquivos estáticos usando o namespace `aws:elasticbeanstalk:environment:proxy:staticfiles`. Para obter mais informações e um exemplo, consulte [the section called “Arquivos estáticos”](#).

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte [Opções de configuração](#) para obter mais informações.

A plataforma Node.js do Amazon Linux AMI (anterior ao Amazon Linux 2)

Se o ambiente Node.js do Elastic Beanstalk usar uma versão da plataforma Amazon Linux AMI (anterior ao Amazon Linux 2), considere as configurações e recomendações específicas nesta seção.

Observações

- As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). As ramificações de plataforma AL2023/AL2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem definições de configuração diferentes.
- Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Opções de configuração específicas da plataforma Node.js: Amazon Linux AMI (AL1)

O Elastic Beanstalk é compatível com algumas opções de configurações específicas da plataforma para versões da plataforma Node.js do Amazon Linux AMI. Você pode escolher o servidor proxy para ser executado na frente da aplicação, uma versão específica do Node.js para ser executada e o comando usado para executar a aplicação.

Como servidor proxy, você pode usar um servidor proxy NGINX ou Apache. Você pode definir o valor `none` para a opção `ProxyServer`. Nesse caso, o Elastic Beanstalk executa a aplicação como autônoma, não por trás de algum servidor de proxy. Se o ambiente executar uma aplicação autônoma, atualize o código para escutar a porta para a qual o NGINX direciona o tráfego.

```
var port = process.env.PORT || 8080;

app.listen(port, function() {
  console.log('Server running at http://127.0.0.1:%s', port);
});
```

Versões da linguagem Node.js: Amazon Linux AMI (AL1)

Em termos de versão de linguagem compatível, a plataforma Node.js do Amazon Linux AMI é diferente de outras plataformas gerenciadas pelo Elastic Beanstalk. Isso porque cada versão da plataforma Node.js só é compatível com algumas versões da linguagem Node.js. Para obter uma lista das versões compatíveis do Node.js, consulte [Node.js](#), no guia Plataformas AWS Elastic Beanstalk.

Você pode usar uma opção de configuração específica da plataforma para definir a versão da linguagem. Para obter instruções, consulte [the section called “Configurar o ambiente Node.js”](#). Ou então, use o console do Elastic Beanstalk para atualizar a versão do Node.js que o ambiente usa, como parte da atualização da versão da plataforma.

Note

Quando o suporte oferecido para a versão do Node.js que você estiver usando for removido da plataforma, será necessário alterar ou remover a configuração da versão antes de fazer a [atualização da plataforma](#). Isso pode ocorrer quando uma vulnerabilidade de segurança é identificada para uma ou mais versões do Node.js.

Quando isso ocorrer, tentar atualizar para uma nova versão da plataforma não compatível com a [NodeVersion](#) configurada falhará. Para evitar a necessidade de criar um novo ambiente, altere a opção de configuração NodeVersion para uma versão de Node.js que tenha suporte na versão de plataforma antiga e na nova, ou [remova a configuração de opção](#) e execute a atualização da plataforma.

Para configurar a versão do Node.js do ambiente no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, em Platform (Plataforma), escolha Change (Alterar).
4. Na caixa de diálogo Update platform version (Atualizar versão da plataforma), selecione uma versão do Node.js.

Update platform version [X]

Availability warning
This operation replaces your instances; your application is unavailable during the update. To keep at least one instance in service during the update, enable rolling updates. Another option is to clone the current environment, which creates a newer version of the platform, and then swap the CNAME of the environments when you are ready to deploy the clone. Learn more at [Updating AWS Elastic Beanstalk Environments with Rolling Updates](#) and [Deploying Version with Zero Downtime](#).

Platform branch
Node.js running on 64bit Amazon Linux

Current Node.js version
12.14.0

Current platform version
4.13.0

New Node.js version
12.14.1

New platform version
4.13.0 (Recommended)

Cancel Save

5. Escolha Save (Salvar).

Namespaces da configuração do Node.js: Amazon Linux AMI (AL1)

A plataforma Node.js do Amazon Linux AMI define opções adicionais nos namespaces `aws:elasticbeanstalk:container:nodejs:staticfiles` e `aws:elasticbeanstalk:container:nodejs`.

O arquivo de configuração a seguir diz ao Elastic Beanstalk para usar `npm start` para executar a aplicação. Ele também define o tipo de proxy como Apache e permite compactação. Por fim, ele configura o proxy para servir arquivos estáticos de dois diretórios de origem. Uma origem são os

arquivos HTML no caminho `html` na raiz do site a partir do diretório de origem `statichtml`. A outra fonte são arquivos de imagem no `images` caminho sob a raiz do site a partir do diretório de origem `staticimages`.

Example `.ebextensions/node-settings.config`

```
option_settings:  
  aws:elasticbeanstalk:container:nodejs:  
    NodeCommand: "npm start"  
    ProxyServer: apache  
    GzipCompression: true  
  aws:elasticbeanstalk:container:nodejs:staticfiles:  
    /html: statichtml  
    /images: staticimages
```

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte [Opções de configuração](#) para obter mais informações.

Configurar o processo da aplicação com um Procfile

Você pode incluir um arquivo denominado `Procfile` na raiz do pacote de fontes para especificar o comando que inicia a aplicação.

Example Procfile

```
web: node index.js
```

Para obter informações sobre o uso do `Procfile`, expanda a seção `Buildfile` e `Procfile` em [the section called “Estender plataformas Linux”](#).

Note

Esse recurso substitui a opção `NodeCommand` herdada no namespace `aws:elasticbeanstalk:container:nodejs`.

Configurar as dependências do aplicativo

A aplicação pode ter dependências em alguns módulos Node.js, como os que você especifica nas instruções `require()`. Esses módulos são armazenados em um diretório `node_modules`. Quando a aplicação é executada, o Node.js carrega os módulos desse diretório. Para obter mais informações, consulte [Loading from node_modules folders](#) (Carregar das pastas `node_modules`) na documentação do Node.js.

Você pode especificar essas dependências usando um arquivo `package.json`. Se o Elastic Beanstalk detectar esse arquivo e não houver um diretório `node_modules` presente, o Elastic Beanstalk executa `npm install` como usuário de aplicativo da web. O comando `npm install` instala as dependências no diretório `node_modules`, que o Elastic Beanstalk cria de antemão. O comando `npm install` acessa os pacotes listados no arquivo `package.json` do registro npm público ou outros locais. Para obter mais informações, consulte o site [npm Docs](#).

Se o Elastic Beanstalk detectar o diretório `node_modules`, não executa o `npm install`, mesmo que exista um arquivo `package.json`. O Elastic Beanstalk pressupõe que os pacotes de dependências estejam disponíveis no diretório `node_modules` para o Node.js acessar e carregar.

As seções a seguir fornecem mais informações sobre como estabelecer as dependências do módulo Node.js para a aplicação.

Note

Se você tiver algum problema de implantação quando o Elastic Beanstalk estiver executando o `npm install`, considere a adoção de uma abordagem alternativa. Inclua o diretório `node_modules` com os módulos de dependências no pacote de fontes da aplicação. Dessa forma você contorna os problemas de instalação das dependências do registro npm público enquanto você investiga o problema. Como a origem dos módulos de dependências é um diretório local, fazer isso também pode ajudar a reduzir o tempo de implantação. Para obter mais informações, consulte [Incluir dependências do Node.js em um diretório `node_modules`](#).

Especificar as dependências do Node.js com um arquivo `package.json`

Use um arquivo `package.json` na raiz do código-fonte do projeto para especificar pacotes de dependência e para fornecer um comando para iniciar. Quando existe um arquivo `package.json` presente e não existe um diretório `node_modules` na raiz da origem do projeto, o Elastic Beanstalk executa `npm install` como usuário do aplicativo da web para instalar dependências do registro

npm público. O Elastic Beanstalk também usa o comando `start` para iniciar a aplicação. Para obter mais informações sobre o arquivo `package.json`, consulte [Specifying dependencies in a package.json file](#) (Especificar dependências em um arquivo) no site npm Docs.

Use a palavra-chave `scripts` para fornecer um comando para iniciar. Atualmente, a palavra-chave `scripts` é usada em vez da opção herdada `NodeCommand` no namespace `aws:elasticbeanstalk:container:nodejs`.

Example `package.json`: Express

```
{
  "name": "my-app",
  "version": "0.0.1",
  "private": true,
  "dependencies": {
    "ejs": "latest",
    "aws-sdk": "latest",
    "express": "latest",
    "body-parser": "latest"
  },
  "scripts": {
    "start": "node app.js"
  }
}
```

Modo de produção e dependências de desenvolvimento

Para especificar suas dependências no arquivo `package.json`, use os atributos `dependencies` e `devDependencies`. O atributo `dependencies` designa os pacotes exigidos pelo seu aplicativo em produção. O atributo `devDependencies` designa pacotes que são necessários apenas para desenvolvimento e testes locais.

O Elastic Beanstalk executa `npm install` como usuário do aplicativo da web com os seguintes comandos. As opções de comando variam dependendo da versão npm incluída na ramificação da plataforma em que seu aplicativo é executado.

- npm v6 — O Elastic Beanstalk instala dependências no modo de produção por padrão. Ele usa o comando `npm install --production`.
- npm v7 ou superior — O Elastic Beanstalk omite as `devDependencies`. Ele usa o comando `npm install --omit=dev`.

Ambos os comandos listados acima não instalam os pacotes que são devDependencies.

Se você precisar instalar os pacotes devDependencies, defina a propriedade de ambiente NPM_USE_PRODUCTION como false. Com esta configuração, não usaremos as opções acima ao executar o npm install. Isso resultará na instalação dos pacotes devDependencies.

SSH e HTTPS

A partir do lançamento da plataforma Amazon Linux 2 de 7 de março de 2023, você também pode usar os protocolos SSH e HTTPS para recuperar pacotes de um repositório Git. A ramificação da plataforma Node.js 16 suporta os protocolos SSH e HTTPS. Node.js 14 suporta somente o protocolo HTTPS.

Example package.json —Node.js 16 suporta HTTPS e SSH

```
...
"dependencies": {
  "aws-sdk": "https://github.com/aws/aws-sdk-js.git",
  "aws-chime": "git+ssh://git@github.com:aws/amazon-chime-sdk-js.git"
}
```

Versões e intervalos de versões

Important

O recurso para especificar intervalos de versões não está disponível para ramificações da plataforma Node.js em execução no AL2023. Oferecemos suporte apenas a uma versão do Node.js em uma ramificação específica do Node.js no AL2023. Se o arquivo package.json especificar um intervalo de versões, ele será ignorado e a versão da ramificação da plataforma do Node.js será usada por padrão.

Use a palavra-chave engines no arquivo package.json para especificar a versão do Node.js que você deseja que a aplicação use. Você também pode especificar um intervalo de versões usando a notação npm. Para obter mais informações sobre a sintaxe para intervalos de versões, consulte [Semantic Versioning using npm](#) (Versionamento semântico usando o npm) no site Node.js. A palavra-chave engines no arquivo package.json do Node.js substitui a opção herdada NodeVersion no namespace aws:elasticbeanstalk:container:nodejs.

Example `package.json`: versão única do Node.js

```
{
  ...
  "engines": { "node" : "14.16.0" }
}
```

Example `package.json`: intervalo de versões do Node.js

```
{
  ...
  "engines": { "node" : ">=10 <11" }
}
```

Quando um intervalo de versões é indicado, o Elastic Beanstalk instala a versão do Node.js mais recente que a plataforma tem disponível dentro do intervalo. Neste exemplo, o intervalo indica que a versão deve ser maior ou igual à versão 10, mas menor que a versão 11. Como resultado, o Elastic Beanstalk instala a versão mais recente do Node.js, a 10.x.y, disponível na [plataforma compatível](#).

Lembre-se de que você só pode especificar uma versão do Node.js que corresponda à ramificação da plataforma. Por exemplo, se você estiver usando a ramificação da plataforma Node.js 16, só poderá especificar uma versão 16.x.y do Node.js. Você pode usar as opções de intervalo de versões suportadas pelo npm para permitir mais flexibilidade. Para saber quais são as versões válidas do Node.js para cada ramificação da plataforma, consulte [Node.js](#) no guia Plataformas do AWS Elastic Beanstalk.

Note

Quando o suporte oferecido para a versão do Node.js que você estiver usando for removido da plataforma, será necessário alterar ou remover a configuração da versão do Node.js antes de fazer a [atualização da plataforma](#). Isso pode ocorrer quando uma vulnerabilidade de segurança é identificada para uma ou mais versões do Node.js.

Nesse caso, ocorre uma falha na tentativa de atualizar para uma nova versão da plataforma não compatível com a versão do Node.js configurada. Para evitar a necessidade de criar um novo ambiente, altere a configuração da versão Node.js `package.json` para uma versão Node.js suportada pela versão antiga da plataforma e a nova. Você tem a opção de especificar um intervalo de versão Node.js que inclua uma versão suportada, conforme

descrito anteriormente neste tópico. Você também tem a opção de remover a configuração e, em seguida, implantar o novo pacote de fonte.

Incluir dependências do Node.js em um diretório `node_modules`

Para implantar pacotes de dependência em instâncias de ambiente juntamente com o código da aplicação, inclua-os em um diretório chamado `node_modules` na raiz da fonte de seu projeto. Para obter mais informações, consulte [Downloading and installing packages locally](#) (Baixar e instalar pacotes localmente) no site npm Docs.

Ao implantar um diretório `node_modules` em uma versão da plataforma Node.js do Amazon Linux 2, o Elastic Beanstalk pressupõe que você está fornecendo seus próprios pacotes de dependências e evita instalar as dependências especificadas em um arquivo [package.json](#). O Node.js procura as dependências no diretório `node_modules`. Para obter mais informações, consulte [Loading from node_modules Folders](#) (Carregar das pastas `node_modules`) na documentação do Node.js.

Note

Se você tiver algum problema de implantação enquanto o Elastic Beanstalk estiver executando `npm install`, considere a abordagem descrita neste tópico como uma solução alternativa enquanto investiga o problema.

Bloquear dependências com `npm shrinkwrap`

A plataforma Node.js executa o `npm install` como usuário do aplicativo da web sempre que você faz uma implantação. Quando há novas versões das dependências disponíveis, elas são instaladas quando você implanta a aplicação, o que pode causar demora na implantação.

Você pode evitar a atualização de dependências criando um arquivo `npm-shrinkwrap.json` que bloqueia as dependências do aplicativo para a versão atual.

```
$ npm install
$ npm shrinkwrap
wrote npm-shrinkwrap.json
```

Inclua esse arquivo no seu pacote de origem para garantir que as dependências sejam instaladas somente uma vez.

Configurar o servidor de proxy

O Elastic Beanstalk usa o NGINX ou o Apache HTTPD como proxy reverso para mapear a aplicação para o balanceador de carga do Elastic Load Balancing na porta 80. O padrão é NGINX. O Elastic Beanstalk oferece uma configuração de proxy padrão que pode ser estendida ou modificada completamente por sua própria configuração.

Por padrão, o Elastic Beanstalk configura o proxy para encaminhar solicitações à sua aplicação na porta 5000. É possível modificar a porta padrão definindo a PORT [propriedade de ambiente](#) para a porta de escuta da sua aplicação principal.

Note

A porta na qual a aplicação escuta não afeta a porta à qual o servidor NGINX escuta para receber solicitações do balanceador de carga.

Configurar o servidor proxy na sua versão da plataforma

Todas as plataformas AL2023/AL2 oferecem suporte a um recurso de configuração de proxy uniforme. Para obter mais informações sobre a configuração do servidor proxy nas versões de plataforma que executam o AL2023/AL2, expanda a seção Configuração do proxy reverso em [the section called “Estender plataformas Linux”](#).

Configurar o proxy da AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o ambiente Node.js do Elastic Beanstalk usar uma versão da plataforma Amazon Linux AMI (anterior ao Amazon Linux 2), leia as informações desta seção.

Observações

- As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). As ramificações de plataforma AL2023/AL2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem definições de configuração diferentes.
- Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023

atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

Estender e substituir a configuração padrão do proxy: Amazon Linux AMI (AL1)

A plataforma Node.js usa um proxy reverso para retransmitir solicitações da porta 80 na instância para a aplicação que está escutando na porta 8081. O Elastic Beanstalk oferece uma configuração de proxy padrão que pode ser estendida ou modificada completamente por sua própria configuração.

Para estender a configuração padrão, adicione arquivos `.conf` a `/etc/nginx/conf.d` com um arquivo de configuração. Para um exemplo específico, consulte [Encerrar o HTTPS em instâncias do EC2 que executam Node.js](#).

A plataforma Node.js define a variável de ambiente `PORT` como a porta para a qual o servidor proxy direciona o tráfego. Leia essa variável no seu código para configurar a porta da sua aplicação.

```
var port = process.env.PORT || 3000;

var server = app.listen(port, function () {
  console.log('Server running at http://127.0.0.1:' + port + '/');
});
```

A configuração padrão do NGINX encaminha o tráfego para um servidor upstream denominado `nodejs` em `127.0.0.1:8081`. É possível remover a configuração padrão e fornecer a sua própria em um [arquivo de configuração](#).

Example `.ebextensions/proxy.config`

O exemplo a seguir remove a configuração padrão e adiciona uma configuração personalizada que encaminha o tráfego para a porta 5000, em vez de 8081.

```
files:
  /etc/nginx/conf.d/proxy.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      upstream nodejs {
        server 127.0.0.1:5000;
        keepalive 256;
      }
```



```
server {
    listen 8080;

    if ($time_iso8601 ~ "^(\\d{4})-(\\d{2})-(\\d{2})T(\\d{2})") {
        set $year $1;
        set $month $2;
        set $day $3;
        set $hour $4;
    }
    access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour
healthd;
    access_log /var/log/nginx/access.log main;

    location / {
        proxy_pass http://nodejs;
        proxy_set_header    Connection "";
        proxy_http_version 1.1;
        proxy_set_header     Host          $host;
        proxy_set_header     X-Real-IP     $remote_addr;
        proxy_set_header     X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    gzip on;
    gzip_comp_level 4;
    gzip_types text/html text/plain text/css application/json application/x-
javascript text/xml application/xml application/xml+rss text/javascript;

    location /static {
        alias /var/app/current/static;
    }

}

/opt/elasticbeanstalk/hooks/configdeploy/post/99_kill_default_nginx.sh:
mode: "000755"
owner: root
group: root
content: |
#!/bin/bash -xe
rm -f /etc/nginx/conf.d/00_elastic_beanstalk_proxy.conf
service nginx stop
service nginx start
```

```
container_commands:
  removeconfig:
    command: "rm -f /tmp/deployment/config/
#etc#nginx#conf.d#00_elastic_beanstalk_proxy.conf /etc/nginx/
conf.d/00_elastic_beanstalk_proxy.conf"
```

A configuração de exemplo (`/etc/nginx/conf.d/proxy.conf`), usa a configuração padrão em `/etc/nginx/conf.d/00_elastic_beanstalk_proxy.conf` como base para incluir o bloco de servidor padrão com configurações de compactação e log, bem como um mapeamento de arquivo estático.

O comando `removeconfig` remove a configuração padrão do contêiner para garantir que o servidor proxy use a configuração personalizada. O Elastic Beanstalk recria a configuração padrão quando cada configuração é implantada. Para explicar isso, no exemplo a seguir, um hook de implantação pós-configuração (`/opt/elasticbeanstalk/hooks/configdeploy/post/99_kill_default_nginx.sh`) é adicionado. Isso remove a configuração padrão e reinicia o servidor de proxy.

Note

A configuração padrão pode ser alterada em versões futuras da plataforma Node.js. Use a versão mais recente da configuração como base para suas personalizações para garantir a compatibilidade.

Se você substituir a configuração padrão, deverá definir os mapeamento de arquivos estáticos e a compactação GZIP. Isso ocorre porque a plataforma não pode aplicar as [configurações padrão](#).

Implantar uma aplicação Express no Elastic Beanstalk

Esta seção apresenta a implantação de uma aplicação demonstrativa no Elastic Beanstalk usando a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB) e a atualização da aplicação para usar a framework [Express](#).

Pré-requisitos

Este tutorial requer os seguintes pré-requisitos:

- Os tempos de execução do Node.js

- O software gerenciador de pacotes Node.js padrão, npm
- O gerador de linha de comando Express
- A Elastic Beanstalk Command Line Interface (EB CLI)

Para obter detalhes sobre como instalar os três primeiros componentes listados e configurar seu ambiente de desenvolvimento local, consulte [Configurar seu ambiente de desenvolvimento Node.js](#). Para este tutorial, você não precisa instalar o AWS SDK para Node.js, que também é mencionado no tópico referenciado.

Para obter detalhes sobre como instalar e configurar a EB CLI, consulte [Instalar a EB CLI](#) e [Configurar a EB CLI](#).

Criar um ambiente do Elastic Beanstalk

Seu diretório de aplicativos

Este tutorial usa um diretório chamado `nodejs-example-express-rds` para o pacote de origem do aplicativo. Crie o diretório `nodejs-example-express-rds` para este tutorial.

```
~$ mkdir nodejs-example-express-rds
```

Note

Cada tutorial neste capítulo usa seu próprio diretório para o pacote de origem do aplicativo. O nome do diretório corresponde ao nome do aplicativo de amostra usado pelo tutorial.

Altere seu diretório de trabalho atual para `nodejs-example-express-rds`.

```
~$ cd nodejs-example-express-rds
```

Agora, vamos configurar um ambiente do Elastic Beanstalk executando a plataforma Node.js e o aplicativo de amostra. Usaremos a interface de linha de comando do Elastic Beanstalk (EB CLI).

Para configurar um repositório EB CLI para seu aplicativo e criar um ambiente Elastic Beanstalk executando a plataforma Node.js

1. Crie um repositório com o comando [eb init](#).

```
~/nodejs-example-express-rds$ eb init --platform node.js --region <region>
```

Esse comando cria um arquivo de configuração em uma pasta chamada `.elasticbeanstalk` que especifica as configurações para a criação de ambientes para a aplicação e cria uma aplicação do Elastic Beanstalk com nome baseado na pasta atual.

2. Crie um ambiente executando um aplicativo de exemplo com o comando [eb create](#).

```
~/nodejs-example-express-rds$ eb create --sample nodejs-example-express-rds
```

Esse comando cria um ambiente com balanceamento de carga com as configurações padrão da plataforma do Node.js e os seguintes recursos:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.

- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

3. Quando a criação do ambiente for concluída, use o comando [eb open](#) para abrir o URL do ambiente no navegador padrão.

```
~/nodejs-example-express-rds$ eb open
```

Agora você criou um ambiente Node.js Elastic Beanstalk com um aplicativo de amostra. Você pode atualizá-lo com seu próprio aplicativo. Em seguida, atualizamos o aplicativo de exemplo para usar a framework do Express.

Atualize o aplicativo para usar o Express

Depois de criar um ambiente com um aplicativo de amostra, você pode atualizá-lo com seu próprio aplicativo. Neste procedimento, primeiro executamos os comandos `npm install` e `express` para configurar a estrutura Express no diretório do seu aplicativo. Em seguida, usamos a EB CLI para atualizar seu ambiente Elastic Beanstalk com o aplicativo atualizado.

Para atualizar seu aplicativo para usar o Express

1. Execute o comando `express`. Isso gera `package.json`, `app.js` e alguns diretórios.

```
~/nodejs-example-express-rds$ express
```

Quando solicitado, digite **y** se você deseja continuar.

Note

Se o comando `express` não funcionar, talvez você não tenha instalado o gerador de linha de comando Express conforme descrito na seção Pré-requisitos anterior. Ou talvez seja necessário configurar a configuração do caminho do diretório para sua máquina local para executar o comando `express`. Consulte a seção Pré-requisitos para obter etapas detalhadas sobre como configurar seu ambiente de desenvolvimento, para que você possa continuar com este tutorial.

2. Configure as dependências locais.

```
~/nodejs-example-express-rds$ npm install
```

3. (Opcional) Verifique a inicialização do servidor do aplicativo da web.

```
~/nodejs-example-express-rds$ npm start
```

Você deve ver saída semelhante a:

```
> nodejs@0.0.0 start /home/local/user/node-express
> node ./bin/www
```

Por padrão, o servidor é executado na porta 3000. Para testá-lo, execute `curl http://localhost:3000` em outro terminal ou abra um navegador no computador local e insira o endereço URL `http://localhost:3000`.

Pressione `Ctrl+C` para interromper o servidor.

4. Implante as alterações em seu ambiente Elastic Beanstalk com o comando [eb deploy](#).

```
~/nodejs-example-express-rds$ eb deploy
```

5. Assim que o ambiente estiver verde e pronto, atualize o URL para verificar se funcionou. Você deve ver uma página da web que diz Welcome to Express.

Em seguida, vamos atualizar o aplicativo Express para servir arquivos estáticos e adicionar uma nova página.

Para configurar arquivos estáticos e adicionar uma nova página ao seu aplicativo Express

1. Adicione um segundo arquivo de configuração na pasta `.ebextensions` com o seguinte conteúdo:

`nodejs-example-express-rds/.ebextensions/staticfiles.config`

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /stylesheets: public/stylesheets
```

Isso configura o servidor de proxy para servir arquivos na pasta `public` no caminho `/public` do aplicativo. Servir arquivos estaticamente do servidor proxy reduz a carga em seu aplicativo. Para obter mais informações, consulte [Arquivos estáticos](#) anterior deste capítulo.

2. (Opcional) Para confirmar se os mapeamentos estáticos estão configurados corretamente, comente a configuração do mapeamento estático em `nodejs-example-express-rds/app.js`. Isso remove o mapeamento do aplicativo de nó.

```
// app.use(express.static(path.join(__dirname, 'public')));
```

Os mapeamentos de arquivo estático no arquivo `staticfiles.config` da etapa anterior ainda devem carregar a folha de estilo com êxito, mesmo depois de comentar essa linha. Para verificar se os mapeamentos de arquivo estático são carregados por meio da configuração de arquivo estático do proxy, em vez do aplicativo expresso, remova os valores a seguir `option_settings:`. Depois de ter sido removido da configuração do arquivo estático e do aplicativo de nó, a folha de estilo falhará ao carregar.

Lembre-se de redefinir o conteúdo do `nodejs-example-express-rds/app.js` e `staticfiles.config` quando terminar o teste.

3. Adicionar `nodejs-example-express-rds/routes/hike.js`. Digite o seguinte:

```
exports.index = function(req, res) {
```

```
res.render('hike', {title: 'My Hiking Log'});
};

exports.add_hike = function(req, res) {
};
```

4. Atualize `nodejs-example-express-rds/app.js` para incluir três novas linhas.

Primeiro, adicione a seguinte linha para adicionar um `require` a essa rota:

```
var hike = require('./routes/hike');
```

O seu arquivo deve ser semelhante ao seguinte snippet:

```
var express = require('express');
var path = require('path');
var hike = require('./routes/hike');
```

Em seguida, adicione as seguintes duas linhas ao `nodejs-example-express-rds/app.js` depois de `var app = express();`

```
app.get('/hikes', hike.index);
app.post('/add_hike', hike.add_hike);
```

O seu arquivo deve ser semelhante ao seguinte snippet:

```
var app = express();
app.get('/hikes', hike.index);
app.post('/add_hike', hike.add_hike);
```

5. Copie `nodejs-example-express-rds/views/index.jade` para `nodejs-example-express-rds/views/hike.jade`.

```
~/nodejs-example-express-rds$ cp views/index.jade views/hike.jade
```

6. Implante as alterações com o comando [eb deploy](#).

```
~/nodejs-example-express-rds$ eb deploy
```


7. O ambiente será atualizado após alguns minutos. Depois que seu ambiente estiver verde e pronto, verifique se funcionou atualizando seu navegador e anexando **hikes** no final do URL (por exemplo, `http://node-express-env-syypntcz2q.elasticbeanstalk.com/hikes`).

Você deve ver uma página da web intitulada My Hiking Log.

Agora, você criou um aplicativo da Web que usa a estrutura Express. Na próxima seção, modificaremos o aplicativo para usar um Amazon Relational Database Service (RDS) para armazenar um log de caminhada.

Atualize o aplicativo para usar o Amazon RDS

Na próxima etapa, atualizaremos o aplicativo para usar o Amazon RDS para MySQL.

Para atualizar seu aplicativo para usar o RDS for MySQL

1. Para criar um banco de dados RDS para MySQL acoplado ao seu ambiente Elastic Beanstalk, siga as instruções no tópico [Adicionar um banco de dados](#) incluído posteriormente neste capítulo. Adicionar uma instância de banco de dados leva cerca de 10 minutos.
2. Atualize a seção de dependências no `package.json` com o seguinte conteúdo:

```
"dependencies": {
  "async": "^3.2.4",
  "express": "4.18.2",
  "jade": "1.11.0",
  "mysql": "2.18.1",
  "node-uuid": "^1.4.8",
  "body-parser": "^1.20.1",
  "method-override": "^3.0.0",
  "morgan": "^1.10.0",
  "errorhandler": "^1.5.1"
}
```

3. Executar `npm install`.

```
~/nodejs-example-express-rds$ npm install
```

4. Atualize `app.js` para se conectar ao banco de dados, criar uma tabela e inserir um único log de caminhada padrão. Sempre que este aplicativo for implantado, ele descartará a tabela de caminhadas anterior e a recriará.

```
/**
 * Module dependencies.
 */

const express = require('express')
, routes = require('./routes')
, hike = require('./routes/hike')
, http = require('http')
, path = require('path')
, mysql = require('mysql')
, async = require('async')
, bodyParser = require('body-parser')
, methodOverride = require('method-override')
, morgan = require('morgan')
, errorHandler = require('errorhandler');

const { connect } = require('http2');

const app = express()

app.set('views', __dirname + '/views')
app.set('view engine', 'jade')
app.use(methodOverride())
app.use(bodyParser.json())
app.use(bodyParser.urlencoded({ extended: true }))
app.use(express.static(path.join(__dirname, 'public')))

app.set('connection', mysql.createConnection({
  host: process.env.RDS_HOSTNAME,
  user: process.env.RDS_USERNAME,
  password: process.env.RDS_PASSWORD,
  port: process.env.RDS_PORT}));

function init() {
  app.get('/', routes.index);
  app.get('/hikes', hike.index);
  app.post('/add_hike', hike.add_hike);
}
```

```
const client = app.get('connection');
async.series([
  function connect(callback) {
    client.connect(callback);
    console.log('Connected!');
  },
  function clear(callback) {
    client.query('DROP DATABASE IF EXISTS mynode_db', callback);
  },
  function create_db(callback) {
    client.query('CREATE DATABASE mynode_db', callback);
  },
  function use_db(callback) {
    client.query('USE mynode_db', callback);
  },
  function create_table(callback) {
    client.query('CREATE TABLE HIKES (' +
      'ID VARCHAR(40), ' +
      'HIKE_DATE DATE, ' +
      'NAME VARCHAR(40), ' +
      'DISTANCE VARCHAR(40), ' +
      'LOCATION VARCHAR(40), ' +
      'WEATHER VARCHAR(40), ' +
      'PRIMARY KEY(ID))', callback);
  },
  function insert_default(callback) {
    const hike = {HIKE_DATE: new Date(), NAME: 'Hazard Stevens',
      LOCATION: 'Mt Rainier', DISTANCE: '4,027m vertical', WEATHER: 'Bad', ID:
      '12345'};
    client.query('INSERT INTO HIKES set ?', hike, callback);
  }
], function (err, results) {
  if (err) {
    console.log('Exception initializing database.');
```

```
    throw err;
  } else {
    console.log('Database initialization complete.');
```

```
    init();
  }
});

module.exports = app
```

5. Adicione o seguinte conteúdo ao arquivo `routes/hike.js`. Isso permitirá que as rotas insiram novos logs de caminhada no banco de dados HIKES.

```
const uuid = require('node-uuid');
exports.index = function(req, res) {
  res.app.get('connection').query( 'SELECT * FROM HIKES', function(err,
rows) {
    if (err) {
      res.send(err);
    } else {
      console.log(JSON.stringify(rows));
      res.render('hike', {title: 'My Hiking Log', hikes: rows});
    }
  });
};
exports.add_hike = function(req, res){
  const input = req.body.hike;
  const hike = { HIKE_DATE: new Date(), ID: uuid.v4(), NAME: input.NAME,
LOCATION: input.LOCATION, DISTANCE: input.DISTANCE, WEATHER: input.WEATHER};
  console.log('Request to log hike:' + JSON.stringify(hike));
  req.app.get('connection').query('INSERT INTO HIKES set ?', hike, function(err) {
    if (err) {
      res.send(err);
    } else {
      res.redirect('/hikes');
    }
  });
};
```

6. Substitua o conteúdo de `routes/index.js` pelo seguinte:

```
/*
 * GET home page.
 */

exports.index = function(req, res){
  res.render('index', { title: 'Express' });
};
```

7. Adicione o seguinte modelo de jade ao `views/hike.jade` para fornecer a interface do usuário para adicionar logs de caminhada.

```
extends layout
```

```
block content
  h1= title
  p Welcome to #{title}

  form(action="/add_hike", method="post")
    table(border="1")
      tr
        td Your Name
        td
          input(name="hike[NAME]", type="textbox")
      tr
        td Location
        td
          input(name="hike[LOCATION]", type="textbox")
      tr
        td Distance
        td
          input(name="hike[DISTANCE]", type="textbox")
      tr
        td Weather
        td
          input(name="hike[WEATHER]", type="radio", value="Good")
          | Good
          input(name="hike[WEATHER]", type="radio", value="Bad")
          | Bad
          input(name="hike[WEATHER]", type="radio", value="Seattle", checked)
          | Seattle
      tr
        td(colspan="2")
          input(type="submit", value="Record Hike")

  div
    h3 Hikes
    table(border="1")
      tr
        td Date
        td Name
        td Location
        td Distance
        td Weather
      each hike in hikes
        tr
          td #{hike.HIKE_DATE.toDateString()}
          td #{hike.NAME}
```

```
td #{hike.LOCATION}
td #{hike.DISTANCE}
td #{hike.WEATHER}
```

8. Implante as alterações com o comando [eb deploy](#).

```
~/nodejs-example-express-rds$ eb deploy
```

Limpeza

Se você terminar de trabalhar com o Elastic Beanstalk, poderá encerrar seu ambiente.

Use o comando `eb terminate` para encerrar seu ambiente e todos os recursos que ele contém.

```
~/nodejs-example-express-rds$ eb terminate
The environment "nodejs-example-express-rds-env" and all associated instances will be
terminated.
To confirm, type the environment name: nodejs-example-express-rds-env
INFO: terminateEnvironment is starting.
...
```

Implantar uma aplicação Express com clustering no Elastic Beanstalk

[Este tutorial mostra como implantar um aplicativo de amostra no Elastic Beanstalk usando a interface de linha de comando \(EB CLI\) do Elastic Beanstalk e, em seguida, atualizando o aplicativo para usar a estrutura Express, a Amazon e o clustering. ElastiCache](#) O clustering aprimora a alta disponibilidade, a performance e a segurança do seu aplicativo web. Para saber mais sobre a Amazon ElastiCache, acesse [O que é a Amazon ElastiCache para Memcached?](#) no Guia do usuário do Amazon ElastiCache for Memcached.

Note

Este exemplo cria AWS recursos, pelos quais você pode ser cobrado. Para obter mais informações sobre AWS preços, consulte <https://aws.amazon.com/pricing/>. Alguns serviços fazem parte do nível de uso AWS gratuito. Se for um cliente novo, você pode testar esses serviços gratuitamente. Consulte <https://aws.amazon.com/free/> para obter mais informações.

Pré-requisitos

Este tutorial requer os seguintes pré-requisitos:

- Os tempos de execução do Node.js
- O software gerenciador de pacotes Node.js padrão, npm
- O gerador de linha de comando Express
- A Elastic Beanstalk Command Line Interface (EB CLI)

Para obter detalhes sobre como instalar os três primeiros componentes listados e configurar seu ambiente de desenvolvimento local, consulte [Configurar seu ambiente de desenvolvimento Node.js](#). Para este tutorial, você não precisa instalar o AWS SDK para Node.js, que também é mencionado no tópico referenciado.

Para obter detalhes sobre como instalar e configurar a EB CLI, consulte [Instalar a EB CLI](#) e [Configurar a EB CLI](#).

Criar um ambiente do Elastic Beanstalk

Seu diretório de aplicativos

Este tutorial usa um diretório chamado `nodejs-example-express-elasticache` para o pacote de origem do aplicativo. Crie o diretório `nodejs-example-express-elasticache` para este tutorial.

```
~$ mkdir nodejs-example-express-elasticache
```

Note

Cada tutorial neste capítulo usa seu próprio diretório para o pacote de origem do aplicativo. O nome do diretório corresponde ao nome do aplicativo de amostra usado pelo tutorial.

Altere seu diretório de trabalho atual para `nodejs-example-express-elasticache`.

```
~$ cd nodejs-example-express-elasticache
```

Agora, vamos configurar um ambiente do Elastic Beanstalk executando a plataforma Node.js e o aplicativo de amostra. Usaremos a interface de linha de comando do Elastic Beanstalk (EB CLI).

Para configurar um repositório EB CLI para seu aplicativo e criar um ambiente Elastic Beanstalk executando a plataforma Node.js

1. Crie um repositório com o comando [eb init](#).

```
~/nodejs-example-express-elasticache$ eb init --platform node.js --region <region>
```

Esse comando cria um arquivo de configuração em uma pasta chamada `.elasticbeanstalk` que especifica as configurações para a criação de ambientes para a aplicação e cria uma aplicação do Elastic Beanstalk com nome baseado na pasta atual.

2. Crie um ambiente executando um aplicativo de exemplo com o comando [eb create](#).

```
~/nodejs-example-express-elasticache$ eb create --sample nodejs-example-express-elasticache
```

Esse comando cria um ambiente com balanceamento de carga com as configurações padrão da plataforma do Node.js e os seguintes recursos:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.

- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

3. Quando a criação do ambiente for concluída, use o comando [eb open](#) para abrir o URL do ambiente no navegador padrão.

```
~/nodejs-example-express-elasticache$ eb open
```

Agora você criou um ambiente Node.js Elastic Beanstalk com um aplicativo de amostra. Você pode atualizá-lo com seu próprio aplicativo. Em seguida, atualizamos o aplicativo de exemplo para usar a framework do Express.

Atualize o aplicativo para usar o Express

Atualize a aplicação demonstrativa no ambiente do Elastic Beanstalk para usar a framework do Express.

Você pode baixar o código-fonte final em [nodejs-example-express-elasticache.zip](#).

Para atualizar seu aplicativo para usar o Express

Depois de criar um ambiente com um aplicativo de amostra, você pode atualizá-lo com seu próprio aplicativo. Neste procedimento, primeiro executamos os comandos `npm install eexpress` para configurar a estrutura Express no diretório do seu aplicativo.

1. Execute o comando `express`. Isso gera `package.json`, `app.js` e alguns diretórios.

```
~/nodejs-example-express-elasticache$ express
```

Quando solicitado, digite **y** se você deseja continuar.

Note

Se o comando `express` não funcionar, talvez você não tenha instalado o gerador de linha de comando Express conforme descrito na seção Pré-requisitos anterior. Ou talvez seja necessário configurar a configuração do caminho do diretório para sua máquina local para executar o comando `express`. Consulte a seção Pré-requisitos para obter etapas detalhadas sobre como configurar seu ambiente de desenvolvimento, para que você possa continuar com este tutorial.

2. Configure as dependências locais.

```
~/nodejs-example-express-elasticache$ npm install
```

3. (Opcional) Verifique a inicialização do servidor do aplicativo da web.

```
~/nodejs-example-express-elasticache$ npm start
```

Você deve ver saída semelhante a:

```
> nodejs@0.0.0 start /home/local/user/node-express  
> node ./bin/www
```

Por padrão, o servidor é executado na porta 3000. Para testá-lo, execute `curl http://localhost:3000` em outro terminal ou abra um navegador no computador local e insira o endereço URL `http://localhost:3000`.

Pressione Ctrl+C para interromper o servidor.

4. Renomeie `nodejs-example-express-elasticache/app.js` para `nodejs-example-express-elasticache/express-app.js`.

```
~/nodejs-example-express-elasticache$ mv app.js express-app.js
```

5. Atualize a linha `var app = express();` no `nodejs-example-express-elasticache/express-app.js` da seguinte forma:

```
var app = module.exports = express();
```

6. No seu computador local, crie um arquivo chamado `nodejs-example-express-elasticache/app.js` com o seguinte código.

```
/**
 * Module dependencies.
 */

const express = require('express'),
    session = require('express-session'),
    bodyParser = require('body-parser'),
    methodOverride = require('method-override'),
    cookieParser = require('cookie-parser'),
    fs = require('fs'),
    filename = '/var/nodelist',
    app = express();

let MemcachedStore = require('connect-memcached')(session);

function setup(cacheNodes) {
  app.use(bodyParser.raw());
  app.use(methodOverride());
  if (cacheNodes.length > 0) {
    app.use(cookieParser());

    console.log('Using memcached store nodes:');
    console.log(cacheNodes);

    app.use(session({
      secret: 'your secret here',
      resave: false,
```

```
        saveUninitialized: false,
        store: new MemcachedStore({ 'hosts': cacheNodes })
    }));
} else {
    console.log('Not using memcached store.');
```

```
    app.use(session({
        resave: false,
        saveUninitialized: false, secret: 'your secret here'
    }));
}

app.get('/', function (req, resp) {
    if (req.session.views) {
        req.session.views++
        resp.setHeader('Content-Type', 'text/html')
        resp.send(`You are session: ${req.session.id}. Views: ${req.session.views}`)
    } else {
        req.session.views = 1
        resp.send(`You are session: ${req.session.id}. No views yet, refresh the page!`
    )
    }
});

if (!module.parent) {
    console.log('Running express without cluster. Listening on port %d',
process.env.PORT || 5000)
    app.listen(process.env.PORT || 5000)
}
}

console.log("Reading elastic cache configuration")
// Load elasticache configuration.
fs.readFile(filename, 'UTF8', function (err, data) {
    if (err) throw err;

    let cacheNodes = []
    if (data) {
        let lines = data.split('\n');
        for (let i = 0; i < lines.length; i++) {
            if (lines[i].length > 0) {
                cacheNodes.push(lines[i])
            }
        }
    }
}
```

```
    setup(cacheNodes)
  });

module.exports = app;
```

7. Substitua o conteúdo do arquivo `nodejs-example-express-elasticache/bin/www` com o seguinte:

```
#!/usr/bin/env node

/**
 * Module dependencies.
 */

const app = require('../app');
const cluster = require('cluster');
const debug = require('debug')('nodejs-example-express-elasticache:server');
const http = require('http');
const workers = {},
    count = require('os').cpus().length;

function spawn() {
    const worker = cluster.fork();
    workers[worker.pid] = worker;
    return worker;
}

/**
 * Get port from environment and store in Express.
 */

const port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

if (cluster.isMaster) {
    for (let i = 0; i < count; i++) {
        spawn();
    }

    // If a worker dies, log it to the console and start another worker.
    cluster.on('exit', function (worker, code, signal) {
```

```
    console.log('Worker ' + worker.process.pid + ' died.');
```

```
    cluster.fork();
  });

  // Log when a worker starts listening
  cluster.on('listening', function (worker, address) {
    console.log('Worker started with PID ' + worker.process.pid + '.');
  });
} else {
  /**
   * Create HTTP server.
   */

  let server = http.createServer(app);

  /**
   * Event listener for HTTP server "error" event.
   */

  function onError(error) {
    if (error.syscall !== 'listen') {
      throw error;
    }

    const bind = typeof port === 'string'
      ? 'Pipe ' + port
      : 'Port ' + port;

    // handle specific listen errors with friendly messages
    switch (error.code) {
      case 'EACCES':
        console.error(bind + ' requires elevated privileges');
        process.exit(1);
        break;
      case 'EADDRINUSE':
        console.error(bind + ' is already in use');
        process.exit(1);
        break;
      default:
        throw error;
    }
  }
}
```

```
/**
 * Event listener for HTTP server "listening" event.
 */

function onListening() {
  const addr = server.address();
  const bind = typeof addr === 'string'
    ? 'pipe ' + addr
    : 'port ' + addr.port;
  debug('Listening on ' + bind);
}

/**
 * Listen on provided port, on all network interfaces.
 */

server.listen(port);
server.on('error', onError);
server.on('listening', onListening);
}

/**
 * Normalize a port into a number, string, or false.
 */

function normalizePort(val) {
  const port = parseInt(val, 10);

  if (isNaN(port)) {
    // named pipe
    return val;
  }

  if (port >= 0) {
    // port number
    return port;
  }

  return false;
}
```

8. Implante as alterações em seu ambiente Elastic Beanstalk com o comando [eb deploy](#).

```
~/nodejs-example-express-elasticache$ eb deploy
```

9. O ambiente será atualizado após alguns minutos. Assim que o ambiente estiver verde e pronto, atualize o URL para verificar se funcionou. Você deverá ver uma página da web dizendo: "Welcome to Express".

Você pode acessar os logs das instâncias EC2 que executam a aplicação. Para obter instruções sobre como acessar os logs, consulte [Visualizar logs de instâncias do Amazon EC2 no ambiente do Elastic Beanstalk](#).

Em seguida, vamos atualizar o aplicativo Express para usar a Amazon ElastiCache.

Para atualizar seu aplicativo Express para usar a Amazon ElastiCache

1. No computador local, crie um diretório `.ebextensions` no diretório de nível superior do seu pacote de origem. Neste exemplo, usamos `nodejs-example-express-elasticache/.ebextensions`.
2. Crie um arquivo de configuração `nodejs-example-express-elasticache/.ebextensions/elasticache-iam-with-script.config` com o seguinte snippet. Para obter mais informações sobre o arquivo de configuração, consulte [Namespace de configuração do Node.js](#). Isso cria um usuário do IAM com as permissões necessárias para descobrir os nós do ElastiCache e grava em um arquivo sempre que o cache muda. Você também pode copiar o arquivo do [nodejs-example-express-elasticache/arquivo.zip](#). Para obter mais informações sobre as ElastiCache propriedades, consulte [Exemplo: ElastiCache](#).

Note

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Resources:

```
MyCacheSecurityGroup:
  Type: 'AWS::EC2::SecurityGroup'
  Properties:
    GroupDescription: "Lock cache down to webserver access only"
```



```
SecurityGroupIngress:
  - IpProtocol: tcp
    FromPort:
      Fn::GetOptionSetting:
        OptionName: CachePort
        DefaultValue: 11211
    ToPort:
      Fn::GetOptionSetting:
        OptionName: CachePort
        DefaultValue: 11211
    SourceSecurityGroupName:
      Ref: AWSEBSecurityGroup
MyElastiCache:
  Type: 'AWS::ElastiCache::CacheCluster'
  Properties:
    CacheNodeType:
      Fn::GetOptionSetting:
        OptionName: CacheNodeType
        DefaultValue: cache.t2.micro
    NumCacheNodes:
      Fn::GetOptionSetting:
        OptionName: NumCacheNodes
        DefaultValue: 1
    Engine:
      Fn::GetOptionSetting:
        OptionName: Engine
        DefaultValue: redis
    VpcSecurityGroupIds:
      -
        Fn::GetAtt:
          - MyCacheSecurityGroup
          - GroupId
AWSEBAutoScalingGroup :
  Metadata :
    ElastiCacheConfig :
      CacheName :
        Ref : MyElastiCache
      CacheSize :
        Fn::GetOptionSetting:
          OptionName : NumCacheNodes
          DefaultValue: 1
WebServerUser :
  Type : AWS::IAM::User
  Properties :
```

```
    Path : "/"
    Policies:
      -
        PolicyName: root
        PolicyDocument :
          Statement :
            -
              Effect : Allow
              Action :
                - cloudformation:DescribeStackResource
                - cloudformation:ListStackResources
                - elasticache:DescribeCacheClusters
              Resource : "*"
    WebServerKeys :
      Type : AWS::IAM::AccessKey
      Properties :
        UserName :
          Ref: WebServerUser

Outputs:
  WebsiteURL:
    Description: sample output only here to show inline string function parsing
    Value: |
      http://`{ "Fn::GetAtt" : [ "AWSEBLoadBalancer", "DNSName" ] }`
  MyElastiCacheName:
    Description: Name of the elasticache
    Value:
      Ref : MyElastiCache
  NumCacheNodes:
    Description: Number of cache nodes in MyElastiCache
    Value:
      Fn::GetOptionSetting:
        OptionName : NumCacheNodes
        DefaultValue: 1

files:
  "/etc/cfn/cfn-credentials" :
    content : |
      AWSAccessKeyId=`{ "Ref" : "WebServerKeys" }`
      AWSSecretKey=`{ "Fn::GetAtt" : ["WebServerKeys", "SecretAccessKey"] }`
    mode : "000400"
    owner : root
    group : root
```

```

"/etc/cfn/get-cache-nodes" :
  content : |
    # Define environment variables for command line tools
    export AWS_ELASTICACHE_HOME="/home/ec2-user/elasticache/$(ls /home/ec2-user/
elasticache/)"
    export AWS_CLOUDFORMATION_HOME=/opt/aws/apitools/cfn
    export PATH=$AWS_CLOUDFORMATION_HOME/bin:$AWS_ELASTICACHE_HOME/bin:$PATH
    export AWS_CREDENTIAL_FILE=/etc/cfn/cfn-credentials
    export JAVA_HOME=/usr/lib/jvm/jre

    # Grab the Cache node names and configure the PHP page
    aws cloudformation list-stack-resources --stack `{ "Ref" :
"AWS::StackName" }` --region `{ "Ref" : "AWS::Region" }` --output text | grep
MyElasticache | awk '{print $4}' | xargs -I {} aws elasticache describe-cache-
clusters --cache-cluster-id {} --region `{ "Ref" : "AWS::Region" }` --show-
cache-node-info --output text | grep '^ENDPOINT' | awk '{print $2 ":" $3}' >
`{ "Fn::GetOptionSetting" : { "OptionName" : "NodeListPath", "DefaultValue" : "/
var/www/html/nodelist" } }`
    mode : "000500"
    owner : root
    group : root

"/etc/cfn/hooks.d/cfn-cache-change.conf" :
  "content": |
    [cfn-cache-size-change]
    triggers=post.update
    path=Resources.AWSEBAutoScalingGroup.Metadata.ElasticacheConfig
    action=/etc/cfn/get-cache-nodes
    runas=root

sources :
  "/home/ec2-user/elasticache" : "https://elasticache-downloads.s3.amazonaws.com/
AmazonElasticacheCli-latest.zip"

commands:
  make-elasticache-executable:
    command: chmod -R ugo+x /home/ec2-user/elasticache/*/bin/*

packages :
  "yum" :
    "aws-apitools-cfn" : []

container_commands:
  initial_cache_nodes:

```

```
command: /etc/cfn/get-cache-nodes
```

3. No seu computador local, crie um arquivo de configuração `nodejs-example-express-elasticache/.ebextensions/elasticache_settings.config` com o seguinte trecho para configurar. ElastiCache

```
option_settings:
  "aws:elasticbeanstalk:customoption":
    CacheNodeType: cache.t2.micro
    NumCacheNodes: 1
    Engine: memcached
    NodeListPath: /var/nodelist
```

4. No seu computador local, substitua `nodejs-example-express-elasticache/express-app.js` pelo seguinte snippet. Esse arquivo lê a lista de nós do disco (`/var/nodelist`) e configura o Express para usar `memcached` como armazenamento de sessão se os nós estiverem presentes. Seu arquivo deve ter uma aparência semelhante à seguinte:

```
/**
 * Module dependencies.
 */

var express = require('express'),
    session = require('express-session'),
    bodyParser = require('body-parser'),
    methodOverride = require('method-override'),
    cookieParser = require('cookie-parser'),
    fs = require('fs'),
    filename = '/var/nodelist',
    app = module.exports = express();

var MemcachedStore = require('connect-memcached')(session);

function setup(cacheNodes) {
  app.use(bodyParser.raw());
  app.use(methodOverride());
  if (cacheNodes) {
    app.use(cookieParser());

    console.log('Using memcached store nodes:');
    console.log(cacheNodes);
  }
}
```

```
    app.use(session({
      secret: 'your secret here',
      resave: false,
      saveUninitialized: false,
      store: new MemcachedStore({'hosts': cacheNodes})
    }));
  } else {
    console.log('Not using memcached store.');
```

```
    app.use(cookieParser('your secret here'));
    app.use(session());
  }

  app.get('/', function(req, resp){
    if (req.session.views) {
      req.session.views++
      resp.setHeader('Content-Type', 'text/html')
      resp.write('Views: ' + req.session.views)
      resp.end()
    } else {
      req.session.views = 1
      resp.end('Refresh the page!')
    }
  });

  if (!module.parent) {
    console.log('Running express without cluster.');
```

```
    app.listen(process.env.PORT || 5000);
  }
}

// Load elasticache configuration.
fs.readFile(filename, 'UTF8', function(err, data) {
  if (err) throw err;

  var cacheNodes = [];
  if (data) {
    var lines = data.split('\n');
    for (var i = 0 ; i < lines.length ; i++) {
      if (lines[i].length > 0) {
        cacheNodes.push(lines[i]);
      }
    }
  }
  setup(cacheNodes);
}
```

```
});
```

5. No computador local, atualize `package.json` com o seguinte conteúdo:

```
"dependencies": {
  "cookie-parser": "~1.4.4",
  "debug": "~2.6.9",
  "express": "~4.16.1",
  "http-errors": "~1.6.3",
  "jade": "~1.11.0",
  "morgan": "~1.9.1",
  "connect-memcached": "*",
  "express-session": "*",
  "body-parser": "*",
  "method-override": "*"
}
```

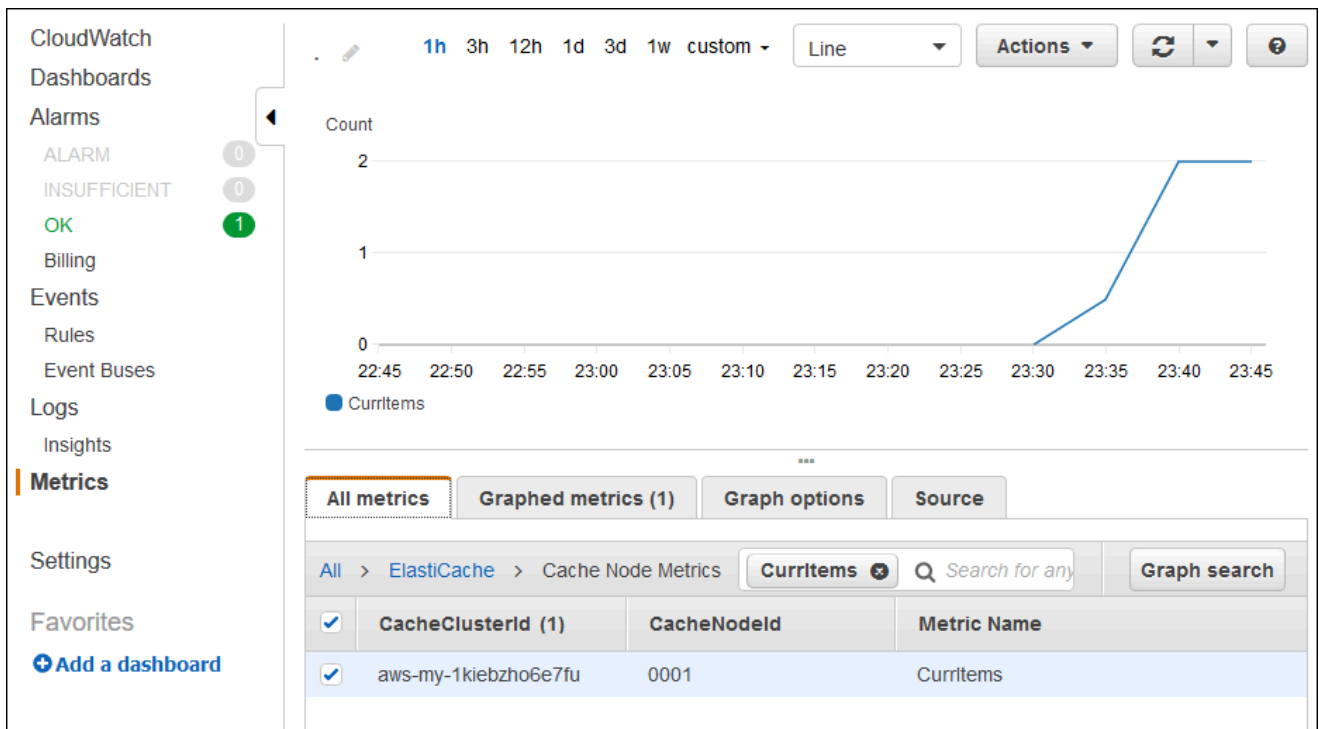
6. Executar `npm install`.

```
~/nodejs-example-express-elasticache$ npm install
```

7. Implante o aplicativo atualizado.

```
~/nodejs-example-express-elasticache$ eb deploy
```

8. O ambiente será atualizado após alguns minutos. Depois que seu ambiente estiver verde e pronto, verifique se o código funcionou.
 - a. Verifique o [CloudWatch console da Amazon](#) para ver suas ElastiCache métricas. Para visualizar suas ElastiCache métricas, selecione Métricas no painel esquerdo e, em seguida, pesquise por `Currltems`. Selecione ElastiCache > Métricas do nó de cache e, em seguida, selecione seu nó de cache para visualizar o número de itens no cache.



Note

Verifique se você está visualizando a mesma região na qual implantou seu aplicativo.

Se você copiar e colar o URL do aplicativo em outro navegador da Web e atualizar a página, verá sua CurrItem contagem aumentar após 5 minutos.

- Faça um snapshot dos logs. Para obter mais informações sobre recuperação de logs, consulte [Visualizar logs de instâncias do Amazon EC2 no ambiente do Elastic Beanstalk](#).
- Verifique o arquivo `/var/log/nodejs/nodejs.log` no pacote de logs. Você deverá ver algo semelhante a:

```
Using memcached store nodes:
[ 'aws-my-1oys9co8zt1uo.1iwtrn.0001.use1.cache.amazonaws.com:11211' ]
```

Limpeza

Se não quiser mais executar seu aplicativo, você pode limpar encerrando seu ambiente e excluindo o aplicativo.

Use o comando `eb terminate` para encerrar o seu ambiente e o comando `eb delete` para excluir o seu aplicativo.

Para encerrar seu ambiente

No diretório onde você criou o repositório local, execute `eb terminate`.

```
$ eb terminate
```

Esse processo pode levar alguns minutos. O Elastic Beanstalk exibirá uma mensagem quando o ambiente foi encerrado com êxito.

Implantar uma aplicação Node.js com o DynamoDB no Elastic Beanstalk

Este tutorial e seu aplicativo de exemplo [nodejs-example-dynamo.zip orientam](#) você pelo processo de implantação de um aplicativo Node.js que usa o AWS SDK JavaScript em Node.js para interagir com o serviço Amazon DynamoDB. Você criará uma tabela do DynamoDB em um banco de dados desacoplado ou externo do ambiente. AWS Elastic Beanstalk Você também configurará o aplicativo para usar um banco de dados desacoplado. Em um ambiente de produção, é uma prática recomendada usar um banco de dados desacoplado do ambiente Elastic Beanstalk para que seja independente do ciclo de vida do ambiente. Essa prática também permite que você execute [implantações azul/verde](#).

O aplicativo de exemplo ilustra o seguinte:

- Uma tabela do DynamoDB que armazena dados de texto fornecidos pelo usuário.
- Os [arquivos de configuração](#) para criar a tabela.
- Um tópico do Amazon Simple Notification Service.
- O uso de um [arquivo package.json](#) para instalar pacotes durante a implantação.

Seções

- [Pré-requisitos](#)
- [Criar um ambiente do Elastic Beanstalk](#)
- [Adicionar permissões às instâncias do seu ambiente](#)
- [Implante o aplicativo de exemplo](#)
- [Criar uma tabela do DynamoDB](#)
- [Atualizar os arquivos de configuração do aplicativo](#)

- [Configurar seu ambiente para alta disponibilidade](#)
- [Limpeza](#)
- [Próximas etapas](#)

Pré-requisitos

Este tutorial requer os seguintes pré-requisitos:

- Os tempos de execução do Node.js
- O software gerenciador de pacotes Node.js padrão, npm
- O gerador de linha de comando Express
- A Elastic Beanstalk Command Line Interface (EB CLI)

Para obter detalhes sobre como instalar os três primeiros componentes listados e configurar seu ambiente de desenvolvimento local, consulte [Configurar seu ambiente de desenvolvimento Node.js](#). Para este tutorial, você não precisa instalar o AWS SDK para Node.js, que também é mencionado no tópico referenciado.

Para obter detalhes sobre como instalar e configurar a EB CLI, consulte [Instalar a EB CLI](#) e [Configurar a EB CLI](#).

Criar um ambiente do Elastic Beanstalk

Seu diretório de aplicativos

Este tutorial usa um diretório chamado `nodejs-example-dynamo` para o pacote de origem do aplicativo. Crie o diretório `nodejs-example-dynamo` para este tutorial.

```
~$ mkdir nodejs-example-dynamo
```

Note

Cada tutorial neste capítulo usa seu próprio diretório para o pacote de origem do aplicativo. O nome do diretório corresponde ao nome do aplicativo de amostra usado pelo tutorial.

Altere seu diretório de trabalho atual para `nodejs-example-dynamo`.

```
~$ cd nodejs-example-dynamo
```

Agora, vamos configurar um ambiente do Elastic Beanstalk executando a plataforma Node.js e o aplicativo de amostra. Usaremos a interface de linha de comando do Elastic Beanstalk (EB CLI).

Para configurar um repositório EB CLI para seu aplicativo e criar um ambiente Elastic Beanstalk executando a plataforma Node.js

1. Crie um repositório com o comando [eb init](#).

```
~/nodejs-example-dynamo$ eb init --platform node.js --region <region>
```

Esse comando cria um arquivo de configuração em uma pasta chamada `.elasticbeanstalk` que especifica as configurações para a criação de ambientes para a aplicação e cria uma aplicação do Elastic Beanstalk com nome baseado na pasta atual.

2. Crie um ambiente executando um aplicativo de exemplo com o comando [eb create](#).

```
~/nodejs-example-dynamo$ eb create --sample nodejs-example-dynamo
```

Esse comando cria um ambiente com balanceamento de carga com as configurações padrão da plataforma do Node.js e os seguintes recursos:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.

- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

3. Quando a criação do ambiente for concluída, use o comando [eb open](#) para abrir o URL do ambiente no navegador padrão.

```
~/nodejs-example-dynamo$ eb open
```

Agora você criou um ambiente Node.js Elastic Beanstalk com um aplicativo de amostra. Você pode atualizá-lo com seu próprio aplicativo. Em seguida, atualizamos o aplicativo de exemplo para usar a framework do Express.

Adicionar permissões às instâncias do seu ambiente

Seu aplicativo é executado em uma ou mais instâncias do EC2 por trás de um load balancer, atendendo a solicitações HTTP da Internet. Quando recebe uma solicitação que exige o uso de AWS serviços, o aplicativo usa as permissões da instância em que é executado para acessar esses serviços.

O aplicativo de amostra usa permissões de instância para gravar dados em uma tabela do DynamoDB e enviar notificações para um tópico do Amazon SNS com o SDK em Node.js. JavaScript Adicione as políticas gerenciadas a seguir ao [perfil da instância](#) padrão para conceder às instâncias do EC2 no ambiente permissão para acessar o DynamoDB e o Amazon SNS:

- AmazonDynamoDB FullAccess
- Amazon SNS FullAccess

Para adicionar políticas ao perfil da instância padrão

1. Abra a página [Roles \(Funções\)](#) no console do IAM.
2. Escolha `aws-elasticbeanstalk-ec2` funções.
3. Na guia Permissions (Permissões), escolha Attach policies (Anexar políticas).
4. Selecione a política gerenciada para os serviços adicionais que o seu aplicativo utiliza. Para este tutorial, selecione `AmazonSNSFullAccess` e `AmazonDynamoDBFullAccess`.
5. Escolha Anexar política.

Consulte [Gerenciar perfis de instância do Elastic Beanstalk](#) para saber mais sobre o gerenciamento de perfis de instância.

Implante o aplicativo de exemplo

Agora seu ambiente está pronto para você implantar e executar o aplicativo de exemplo deste tutorial: [nodejs-example-dynamo.zip](#).

Para implantar e executar o aplicativo de exemplo do tutorial

1. Altere seu diretório de trabalho atual para o diretório do aplicativo `nodejs-example-dynamo`.

```
~$ cd nodejs-example-dynamo
```

2. Baixe e extraia o conteúdo do pacote de origem do aplicativo de exemplo [nodejs-example-dynamo.zip](#) para o diretório do aplicativo. `nodejs-example-dynamo`
3. Implante o aplicativo de exemplo em seu ambiente Elastic Beanstalk com o comando [eb deploy](#).

```
~/nodejs-example-dynamo$ eb deploy
```

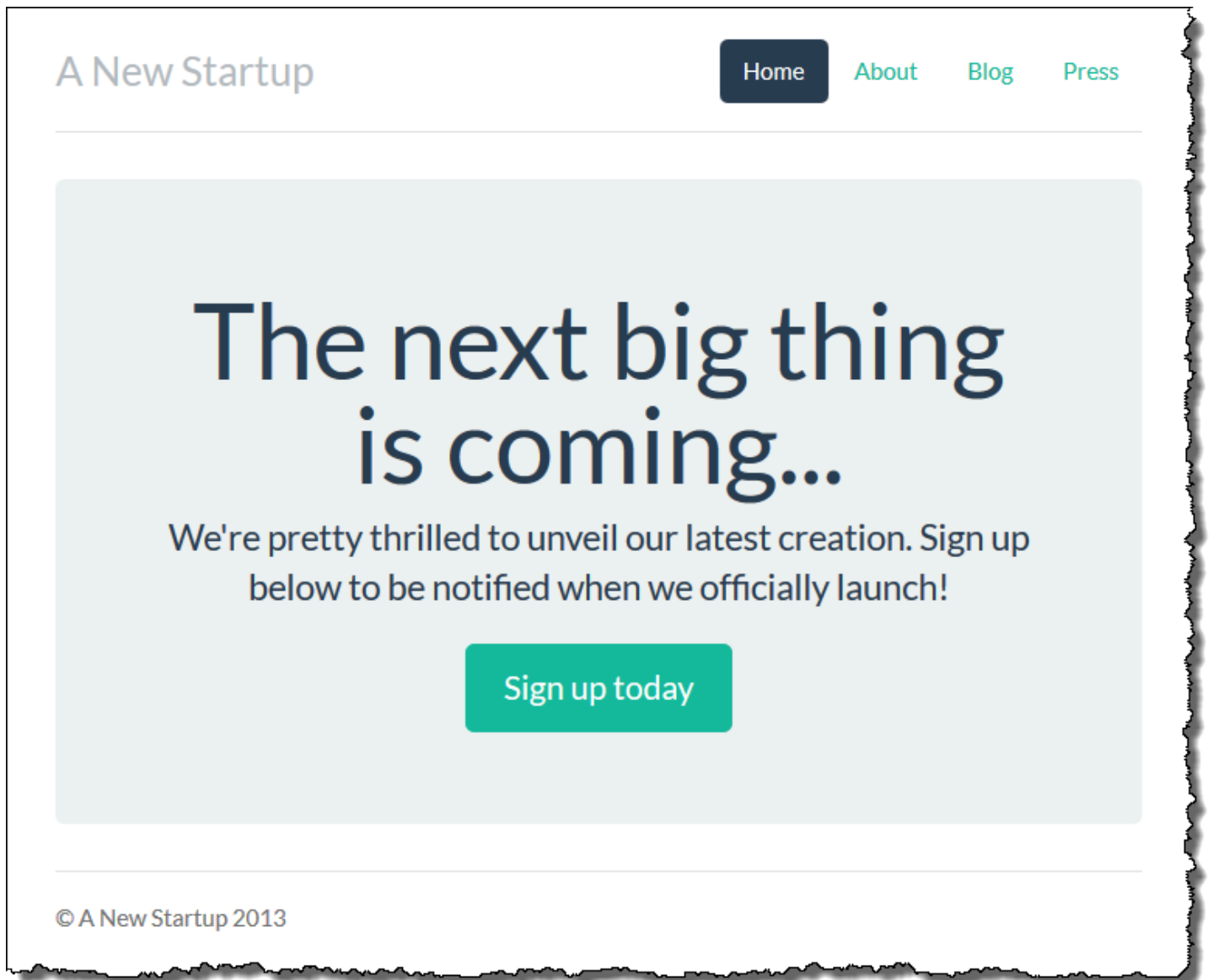
Note

Por padrão, o `eb deploy` comando cria um arquivo ZIP da pasta do seu projeto. Você pode configurar a EB CLI para implantar um artefato do seu processo de compilação, em vez de criar um arquivo ZIP da pasta do projeto. Para ter mais informações, consulte [Implantar um artefato, em vez da pasta do projeto](#).

4. Quando a criação do ambiente for concluída, use o comando [eb open](#) para abrir o URL do ambiente no navegador padrão.

```
~/nodejs-example-dynamo$ eb open
```

O site coleta informações de contato do usuário e usa uma tabela do DynamoDB para armazenar os dados. Para adicionar uma entrada, escolha Sign up today, insira um nome e um endereço de e-mail e, em seguida, selecione Sign Up!. A aplicação Web grava o conteúdo do formulário na tabela e aciona uma notificação por e-mail do Amazon SNS.



No momento, o tópico do Amazon SNS está configurado com um espaço reservado para notificações de e-mail. Você atualizará a configuração em breve, mas enquanto isso, pode verificar a tabela do DynamoDB e o tópico do Amazon SNS no Console de Gerenciamento da AWS Management Console.

Para visualizar a tabela

1. Abra a [página Tables \(Tabelas\)](#) no console do DynamoDB.
2. Encontre a tabela que o aplicativo criou. O nome começa com `awseb` e contém `StartupSignupsTable`
3. Selecione a tabela, escolha `Items` e `Start search` para visualizar todos os itens na tabela.

A tabela contém uma entrada para cada endereço de e-mail enviado no site de inscrição. Além de gravar na tabela, a aplicação envia uma mensagem para um tópico do Amazon SNS que tem duas assinaturas, uma para notificações por e-mail para você e outra para uma fila do Amazon Simple Queue Service que uma aplicação de operador pode ler para processar solicitações e enviar e-mails para os clientes interessados.

Para visualizar o tópico

1. Abra a [página Topics \(Tópicos\)](#) no console do Amazon SNS.
2. Encontre o tópico que o aplicativo criou. O nome começa com `awseb` e contém `NewSignupTopic`.
3. Escolha o tópico para visualizar as assinaturas.

O aplicativo ([app.js](#)) define duas rotas. O caminho raiz (`/`) retorna uma página da Web renderizada a partir de um modelo Embedded JavaScript (EJS) com um formulário que o usuário preenche para registrar seu nome e endereço de e-mail. A submissão do formulário envia uma solicitação POST com os dados do formulário para a rota `/signup`, que grava uma entrada na tabela do DynamoDB e publica uma mensagem para o tópico do Amazon SNS para notificar o proprietário da inscrição.

A aplicação demonstrativa inclui os [arquivos de configuração](#) que criam a tabela do DynamoDB, o tópico do Amazon SNS e a fila do Amazon SQS usados pela aplicação. Isso permite que você crie um novo ambiente e teste a funcionalidade imediatamente, mas tem a desvantagem de vincular a tabela do DynamoDB ao ambiente. Para um ambiente de produção, você deve criar a tabela do DynamoDB fora do seu ambiente para evitar perdê-lo quando encerrar o ambiente ou atualizar sua configuração.

Criar uma tabela do DynamoDB

Para usar uma tabela do DynamoDB externa com uma aplicação em execução no Elastic Beanstalk primeiro crie uma tabela no DynamoDB. Quando uma instância é criada fora do Elastic Beanstalk, ela é completamente independente do Elastic Beanstalk e dos ambientes do Elastic Beanstalk e não será encerrada pelo Elastic Beanstalk.

Crie uma tabela com as seguintes configurações:

- Table name (Nome da tabela): **nodejs-tutorial**
- chave primária: **email**
- Tipo de chave primária: string

Como criar uma tabela do DynamoDB

1. Abra a [página Tables \(Tabelas\)](#) no console de gerenciamento do DynamoDB.
2. Escolha Create table.
3. Digite um nome de tabela e chave primária.
4. Escolha o tipo de chave primária.
5. Escolha Criar.

Atualizar os arquivos de configuração do aplicativo

Atualize os [arquivos de configuração](#) na origem do aplicativo para usar a tabela nodejs-tutorial em vez de criar uma nova.

Para atualizar o aplicativo de exemplo para uso em produção

1. Altere seu diretório de trabalho atual para o diretório do aplicativo nodejs-example-dynamo.

```
~$ cd nodejs-example-dynamo
```

2. Abra `.ebextensions/options.config` e altere os valores das configurações a seguir:
 - NewSignupEmail— Seu endereço de e-mail.
 - STARTUP_SIGNUP_TABLE: `nodejs-tutorial`

Example `.ebextensions/options.config`

```
option_settings:
  aws:elasticbeanstalk:customoption:
    NewSignupEmail: you@example.com
  aws:elasticbeanstalk:application:environment:
    THEME: "flatly"
    AWS_REGION: '`{"Ref" : "AWS::Region"}``'
    STARTUP_SIGNUP_TABLE: nodejs-tutorial
    NEW_SIGNUP_TOPIC: '`{"Ref" : "NewSignupTopic"}``'
  aws:elasticbeanstalk:container:nodejs:
    ProxyServer: nginx
  aws:elasticbeanstalk:container:nodejs:staticfiles:
    /static: /static
```



```
aws:autoscaling:asg:
  Cooldown: "120"
aws:autoscaling:trigger:
  Unit: "Percent"
  Period: "1"
  BreachDuration: "2"
  UpperThreshold: "75"
  LowerThreshold: "30"
  MeasureName: "CPUUtilization"
```

Isso aplica as seguintes configurações para o aplicativo:

- O endereço de e-mail que o tópico do Amazon SNS usa para notificações é definido como seu endereço ou aquele que você insere no arquivo `options.config`.
- A tabela `nodejs-tutorial` será usada em vez da criada por `.ebextensions/create-dynamodb-table.config`.

3. Remover `.ebextensions/create-dynamodb-table.config`.

```
~/nodejs-tutorial$ rm .ebextensions/create-dynamodb-table.config
```

Na próxima vez que você implantar o aplicativo, a tabela criada por esse arquivo de configuração será excluída.

4. Implante o aplicativo atualizado em seu ambiente Elastic Beanstalk com o comando [eb deploy](#).

```
~/nodejs-example-dynamo$ eb deploy
```

5. Quando a criação do ambiente for concluída, use o comando [eb open](#) para abrir o URL do ambiente no navegador padrão.

```
~/nodejs-example-dynamo$ eb open
```

Quando você implanta, o Elastic Beanstalk atualiza a configuração do tópico do Amazon SNS e exclui a tabela do DynamoDB que ele criou quando a primeira versão da aplicação foi implantada.

Agora, quando você encerrar o ambiente, a tabela `nodejs-tutorial` não será excluída. Isso permite que você realize implantações azul/verde, modifique os arquivos de configuração ou desative seu site sem o risco de perda de dados.

Abra seu site em um navegador e verifique se o formulário funciona como esperado. Crie algumas entradas e, depois, verifique o console do DynamoDB para verificar a tabela.

Para visualizar a tabela

1. Abra a [página Tables \(Tabelas\)](#) no console do DynamoDB.
2. Encontre a tabela nodejs-tutorial.
3. Selecione a tabela, escolha Items e Start search para visualizar todos os itens na tabela.

Você também pode ver que o Elastic Beanstalk excluiu a tabela que foi criada anteriormente.

Configurar seu ambiente para alta disponibilidade

Por fim, configure o grupo de Auto Scaling do seu ambiente com uma contagem mínima de instâncias mais alta. Execute pelo menos duas instâncias o tempo todo para impedir que os servidores Web em seu ambiente sejam um ponto único de falha e para permitir que você implante alterações sem colocar seu site fora do serviço.

Para configurar o grupo de Auto Scaling de seu ambiente para alta disponibilidade

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Na seção Auto Scaling group (Grupo de Auto Scaling) defina Min instances (Mínimo de instâncias) como **2**.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Limpeza

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. [O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente, como instâncias do Amazon EC2, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.](#)

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

Você também pode excluir as tabelas externas do DynamoDB que você criou.

Como excluir uma tabela do DynamoDB

1. Abra a [página Tables \(Tabelas\)](#) no console do DynamoDB.
2. Selecione uma tabela.
3. Escolha Actions e, em seguida, escolha Delete table.
4. Escolha Excluir.

Próximas etapas

O aplicativo de exemplo usa arquivos de configuração para definir as configurações do software e criar AWS recursos como parte do seu ambiente. Consulte [Personalização avançada de ambiente com arquivos de configuração \(.ebextensions\)](#) para obter mais informações sobre os arquivos de configuração e seu uso.

O aplicativo de exemplo para este tutorial usa a framework da Web Express para Node.js. Para obter mais informações sobre o Express, consulte a documentação oficial em expressjs.com.

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, [configure um nome de domínio personalizado](#) para seu ambiente e [habilite o HTTPS](#) para conexões seguras.

Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicação Node.js

É possível usar uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS) para armazenar dados coletados e modificados pela aplicação. O banco de dados pode ser acoplado ao seu ambiente e gerenciado pelo Elastic Beanstalk ou pode ser criado como desacoplado e gerenciado externamente por outro serviço. Este tópico fornece instruções para criar o Amazon RDS usando o console do Elastic Beanstalk. O banco de dados será acoplado ao ambiente e gerenciado pelo Elastic Beanstalk. Para obter mais informações sobre como integrar um Amazon RDS com o Elastic Beanstalk, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

Seções

- [Adicionar uma instância de banco de dados ao seu ambiente](#)
- [Download de um driver](#)
- [Conectar-se a um banco de dados](#)

Adicionar uma instância de banco de dados ao seu ambiente

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).

5. Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização do ambiente for concluída, o nome de host da instância de banco de dados e outras informações de conexão estarão disponíveis para o seu aplicativo por meio das seguintes propriedades de ambiente:

Nome da propriedade	Descrição	Valor da propriedade
RDS_HOSTNAME	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
RDS_PORT	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
RDS_DB_NAME	O nome do banco de dado, ebdb .	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
RDS_USERNAME	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Para obter mais informações sobre como configurar uma instância de banco de dados acoplada a um ambiente Elastic Beanstalk, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

Download de um driver

Adicione o driver de banco de dados ao arquivo [package.json do projeto](#) em dependencies.

Example **package.json**: Express com MySQL

```
{
  "name": "my-app",
  "version": "0.0.1",
  "private": true,
  "dependencies": {
    "ejs": "latest",
    "aws-sdk": "latest",
    "express": "latest",
    "body-parser": "latest",
    "mysql": "latest"
  },
  "scripts": {
    "start": "node app.js"
  }
}
```

Pacotes de drivers comuns para Node.js

- MySQL – [mysql](#)
- PostgreSQL – [node-postgres](#)
- SQL Server – [node-mssql](#)
- Oracle – [node-oracledb](#)

Conectar-se a um banco de dados

O Elastic Beanstalk fornece informações de conexão referentes às instâncias de banco de dados associadas nas propriedades de ambiente. Use `process.env.VARIABLE` para ler as propriedades e configurar uma conexão de banco de dados.

Example **app.js**: conexão de banco de dados MySQL

```
var mysql = require('mysql');

var connection = mysql.createConnection({
  host      : process.env.RDS_HOSTNAME,
```

```
user      : process.env.RDS_USERNAME,
password  : process.env.RDS_PASSWORD,
port      : process.env.RDS_PORT
});

connection.connect(function(err) {
  if (err) {
    console.error('Database connection failed: ' + err.stack);
    return;
  }

  console.log('Connected to database.');
```

```
connection.end();
```

Para obter mais informações sobre a criação de uma string de conexão usando node-mysql, consulte npmjs.org/package/mysql.

Recursos

Há vários lugares que você pode acessar para obter ajuda adicional durante o desenvolvimento de seus aplicativos Node.js:

Recurso	Descrição
GitHub	Instale a AWS SDK for Node.js usando o GitHub.
Fórum de desenvolvimento do Node.js	Publique suas perguntas e receba comentários.
AWS SDK para Node.js (pré-visualização para desenvolvedores)	Um local central para acessar código de exemplo, documentação, ferramentas e outros recursos.

Criar e implantar aplicações PHP no Elastic Beanstalk

Com o AWS Elastic Beanstalk para PHP, é fácil implantar, gerenciar e escalar suas aplicações web PHP usando a Amazon Web Services. O Elastic Beanstalk para PHP está disponível para qualquer pessoa que esteja desenvolvendo ou hospedando uma aplicação Web usando PHP. Esta seção fornece instruções para implantar a aplicação Web PHP para o Elastic Beanstalk. É possível

implantar a aplicação em apenas alguns minutos usando a Interface de Linhas de Comando (CLI do EB) do Elastic Beanstalk ou o Console de Gerenciamento do Elastic Beanstalk. Ela também fornece um demonstração de frameworks comuns, como CakePHP e Symfony.

Os tópicos neste capítulo pressupõem que você tem um nível básico de conhecimento sobre ambientes do Elastic Beanstalk. Se você não tiver usado o Elastic Beanstalk antes, tente o [tutorial de conceitos básicos](#) para saber os princípios.

Se você precisar de ajuda com desenvolvimento do aplicativo PHP, há vários lugares que você pode acessar:

Recurso	Descrição
GitHub	Instalar o AWS SDK for PHP usando o GitHub.
Fórum de desenvolvimento do PHP	Publique suas perguntas e receba comentários.
Centro do desenvolvedor para PHP	Um local central para acessar código de exemplo, documentação, ferramentas e outros recursos.
Perguntas frequentes sobre o AWS SDK for PHP	Obtenha respostas para perguntas frequentes.

Tópicos

- [Configurar seu ambiente de desenvolvimento de PHP](#)
- [Usar a plataforma PHP do Elastic Beanstalk](#)
- [Implantar uma aplicação Laravel no Elastic Beanstalk](#)
- [Implantar uma aplicação CakePHP no Elastic Beanstalk](#)
- [Implantar uma aplicação Symfony no Elastic Beanstalk](#)
- [Implantar uma aplicação PHP de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk](#)
- [Implantação de um WordPress site de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk](#)
- [Implantar um site Drupal de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk](#)
- [Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicações PHP](#)

Configurar seu ambiente de desenvolvimento de PHP

Configure um ambiente de desenvolvimento PHP para testar sua aplicação localmente, antes de implantá-lo no AWS Elastic Beanstalk. Este tópico descreve as etapas de configuração do ambiente de desenvolvimento e inclui links para páginas de instalação de ferramentas úteis.

Para ver as etapas de configuração comuns e as ferramentas que se aplicam a todas as linguagens, consulte [Configurar a máquina de desenvolvimento](#).

Seções

- [Instalar PHP](#)
- [Instalar o Composer](#)
- [Instalação do AWS SDK for PHP](#)
- [Instalar um editor de texto ou IDE](#)

Instalar PHP

Instalas PHP e algumas extensões comuns. Se você não tiver uma preferência, obtenha a versão mais recente. As etapas variam de acordo com sua plataforma e o gerenciador de pacotes disponível.

No Amazon Linux, use yum:

```
$ sudo yum install php
$ sudo yum install php-mbstring
$ sudo yum install php-intl
```

Note

Para obter versões de pacote PHP específicas que correspondam à versão na [versão da plataforma PHP](#) do Elastic Beanstalk, use o comando `yum search php` para encontrar versões de pacote disponíveis, como `php72`, `php72-mbstring` e `php72-intl`. Depois, use `sudo yum install package` para instalá-las.

No Ubuntu, use apt:

```
$ sudo apt install php-all-dev
```

```
$ sudo apt install php-intl
$ sudo apt install php-mbstring
```

No OS X, use brew:

```
$ brew install php
$ brew install php-intl
```

Note

Para obter versões de pacote PHP específicas que correspondam à versão na [versão da plataforma PHP](#) do Elastic Beanstalk, consulte [Homebrew Formulae](#) para obter as versões PHP disponíveis, como `php@7.2`. Depois, use `brew install package` para instalá-las. Dependendo da versão, `php-intl` podem ser incluídos no pacote PHP principal e não existir como pacote separado.

No Windows 10, [instale o Windows Subsystem para Linux](#) para obter o Ubuntu e instalar PHP com `apt`. Para versões mais antigas, acesse a página de downloads em [windows.php.net](#) para obter o PHP e leia [esta página](#) para obter informações sobre extensões.

Após a instalação do PHP, abra o terminal novamente e execute `php --version` para garantir que a nova versão esteja instalada e definida como padrão.

Instalar o Composer

O Composer é um gerenciamento de dependência para PHP. Você pode usá-lo para instalar as bibliotecas, rastrear as dependências do aplicativo e gerar projetos para quadros PHP populares.

Instalar o Composer com o script do PHP em [getcomposer.org](#).

```
$ curl -s https://getcomposer.org/installer | php
```

O instalador gera um arquivo PHAR no diretório atual. Move este arquivo para um local em seu ambiente PATH para que você possa usá-lo como um executável.

```
$ mv composer.phar ~/.local/bin/composer
```

Instale bibliotecas com o comando `require`.

```
$ composer require twig/twig
```

O Composer adiciona bibliotecas que você instala localmente no arquivo [composer.json do projeto](#). Quando você implanta o código do projeto, o Elastic Beanstalk usa o Composer para instalar as bibliotecas listadas neste arquivo em instâncias da aplicação do seu ambiente.

Se você tiver problemas ao instalar o Composer, consulte a [documentação do composer](#).

Instalação do AWS SDK for PHP

Se você precisa gerenciar os recursos da AWS a partir de sua aplicação, instale o AWS SDK for PHP. Por exemplo, com o SDK for PHP, é possível usar o Amazon DynamoDB (DynamoDB) para armazenar informações de usuário e sessão sem criar um banco de dados relacional.

Instale o SDK for PHP com o Composer.

```
$ composer require aws/aws-sdk-php
```

Acesse a [página inicial do AWS SDK for PHP](#) para obter mais informações e instruções de instalação.

Instalar um editor de texto ou IDE

Os Integrated development environments (IDEs – Ambientes de desenvolvimento integrados) oferecem uma ampla variedade de recursos que facilitam o desenvolvimento de aplicativos. Se você não usou um IDE para desenvolvimento PHP, tente o Eclipse e o PhpStorm e veja qual é melhor para você.

- [Instalar o Eclipse](#)
- [Instalar o PhpStorm](#)

Note

Um IDE pode adicionar arquivos à pasta do seu projeto, que talvez você não queira confirmar no controle de código-fonte. Para evitar a confirmação desses arquivos no controle de código-fonte, use `.gitignore` ou a ferramenta de controle de código-fonte equivalente.

Se você deseja apenas começar a codificação e não precisa de todos os recursos de um IDE, considere [instalar o Sublime Text](#).

Usar a plataforma PHP do Elastic Beanstalk

AWS Elastic Beanstalk suporta várias plataformas para diferentes versões da linguagem de programação PHP. Essas plataformas oferecem suporte a aplicativos web em PHP que podem ser executados sozinhos ou sob o Composer. Saiba mais em [PHP](#) no documento Plataformas do AWS Elastic Beanstalk .

O Elastic Beanstalk fornece [opções de configuração](#) que podem ser usadas para personalizar o software que é executado nas instâncias do EC2 no ambiente do Elastic Beanstalk. É possível [configurar as variáveis de ambiente](#) necessárias à sua aplicação, habilitar a alternância de log para o Amazon S3, mapear pastas na origem da aplicação que contenham arquivos estáticos para caminhos servidos pelo servidor de proxy e definir configurações de inicialização comuns do PHP.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para [modificar a configuração de um ambiente em execução](#). Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar [configurações salvas](#) para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir [arquivos de configuração](#). As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

Se você usa o Composer, pode [incluir um arquivo composer.json](#) em seu pacote de origem para instalar pacotes durante a implantação.

Para configuração avançada de PHP e definições de PHP que não são fornecidos como opções de configuração, é possível [usar arquivos de configuração para fornecer um arquivo INI](#) que pode estender e substituir as configurações padrão aplicadas pelo Elastic Beanstalk ou instalar extensões adicionais.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console. Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte [Opções de configuração](#).

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte [the section called “Estender plataformas Linux”](#).

Considerações sobre o PHP 8.1 no Amazon Linux 2

Leia esta seção se você estiver usando a ramificação da plataforma PHP 8.1.

Considerações sobre o PHP 8.1 no Amazon Linux 2

Note

As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). Não se aplica às ramificações da plataforma PHP com base no AL2023. Também não se aplica ao PHP 8.0 Amazon Linux 2 filial da plataforma.

O Elastic Beanstalk armazena os pacotes RPM relacionados ao PHP 8.1 para o PHP 8.1 no Amazon Linux 2 ramificação da plataforma nas instâncias do EC2 em um diretório local, em vez do repositório Amazon Linux. Você pode usar `rpm -i` para instalar os pacotes necessários. A partir da [versão 3.5.0 da plataforma PHP 8.1](#), o Elastic Beanstalk armazena os pacotes RPM relacionados à PHP 8.1 no diretório local do EC2 a seguir.

```
/opt/elasticbeanstalk/RPMS
```

O exemplo a seguir instala o pacote `php-debuginfo`.

```
$rpm -i /opt/elasticbeanstalk/RPMS/php-debuginfo-8.1.8-1.amzn2.x86_64.rpm
```

A versão no nome do pacote variará de acordo com a versão real listada no diretório local do EC2 `/opt/elasticbeanstalk/RPMS`. Use a mesma sintaxe para instalar outros pacotes RPM da plataforma PHP 8.1.

Expanda a seção a seguir para exibir uma lista dos pacotes RPM que fornecemos.

Pacotes RPM

A lista a seguir fornece os pacotes RMP que a plataforma PHP 8.1 do Elastic Beanstalk fornece no Amazon Linux 2. Eles ficam localizados no diretório local `/opt/elasticbeanstalk/RPMS`.

Os números de versão 8.1.8-1 e 3.7.0-1 nos nomes dos pacotes listados são apenas um exemplo.

- php-8.1.8-1.amzn2.x86_64.rpm
- php-bcmath-8.1.8-1.amzn2.x86_64.rpm
- php-cli-8.1.8-1.amzn2.x86_64.rpm
- php-common-8.1.8-1.amzn2.x86_64.rpm
- php-dba-8.1.8-1.amzn2.x86_64.rpm
- php-debug-8.1.8-1.amzn2.x86_64.rpm
- php-debuginfo-8.1.8-1.amzn2.x86_64.rpm
- php-devel-8.1.8-1.amzn2.x86_64.rpm
- php-embedded-8.1.8-1.amzn2.x86_64.rpm
- php-enchant-8.1.8-1.amzn2.x86_64.rpm
- php-fpm-8.1.8-1.amzn2.x86_64.rpm
- php-gd-8.1.8-1.amzn2.x86_64.rpm
- php-gmp-8.1.8-1.amzn2.x86_64.rpm
- php-intl-8.1.8-1.amzn2.x86_64.rpm
- php-ldap-8.1.8-1.amzn2.x86_64.rpm
- php-mbstring-8.1.8-1.amzn2.x86_64.rpm
- php-mysqlnd-8.1.8-1.amzn2.x86_64.rpm
- php-odbc-8.1.8-1.amzn2.x86_64.rpm
- php-opcache-8.1.8-1.amzn2.x86_64.rpm
- php-pdo-8.1.8-1.amzn2.x86_64.rpm
- php-pear-1.10.13-1.amzn2.noarch.rpm
- php-pgsql-8.1.8-1.amzn2.x86_64.rpm
- php-process-8.1.8-1.amzn2.x86_64.rpm
- php-pspell-8.1.8-1.amzn2.x86_64.rpm
- php-snmp-8.1.8-1.amzn2.x86_64.rpm
- php-soap-8.1.8-1.amzn2.x86_64.rpm
- php-sodium-8.1.8-1.amzn2.x86_64.rpm
- php-xml-8.1.8-1.amzn2.x86_64.rpm
- php-pecl-imagick-3.7.0-1.amzn2.x86_64.rpm
- php-pecl-imagick-debuginfo-3.7.0-1.amzn2.x86_64.rpm

- `php-pecl-imagick-devel-3.7.0-1.amzn2.noarch.rpm`

Você pode usar os pacotes PEAR e PECL para instalar extensões comuns. Para obter mais informações sobre o PEAR, consulte o site [PEAR PHP Extension and Application Repository](#) (Repositório de extensão e aplicação do PEAR PHP). Para obter mais informações sobre o PECL, consulte o site de [extensões PECL](#).

Os exemplos de comandos a seguir instalam as extensões Memcached.

```
$pecl install memcache
```

Ou você também pode usar o seguinte:

```
$pear install pecl/memcache
```

Os exemplos de comando a seguir instalam as extensões Redis.

```
$pecl install redis
```

Ou você também pode usar o seguinte:

```
$pear install pecl/redis
```

Configurar seu ambiente PHP

É possível usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3, configurar variáveis que a aplicação pode ler no ambiente e alterar as configurações do PHP.

Como configurar seu ambiente PHP no console do Elastic Beanstalk

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).

4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Configurações de PHP

- Proxy server (Servidor de proxy): o servidor de proxy a ser usado nas instâncias de ambiente. Por padrão, é usado nginx.
- Document root (Raiz do documento): a pasta que contém a página padrão do site. Se a sua página de boas-vindas não estiver na raiz do pacote de origem, especifique a pasta que a contém relativo ao caminho raiz. Por exemplo, /public se a página de boas-vindas estiver em uma pasta chamada public.
- Memory limite (Limite de memória): a quantidade máxima de memória que um script tem permissão para alocar. Por exemplo, 512M.
- Zlib output compression (Compactação de saída Zlib): defina como 0n para compactar respostas.
- Allow URL fopen (Permitir URL fopen): defina como 0ff para impedir o download de arquivos de scripts de locais remotos.
- Display errors (Exibir erros): defina como 0n para mostrar mensagens de erro interno para depuração.
- Max execution time (Tempo de execução máximo): o tempo máximo em segundos que um script pode ser executado antes que o ambiente o encerre.

Opções de log

A seção Log Options tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Enable log file rotation to Amazon S3 (Habilitar a rotação de arquivos de log para o Amazon S3): especifica se os arquivos de log das instâncias do Amazon EC2 da aplicação serão copiados para o bucket do Amazon S3 associado a ela.

Arquivos estáticos

Para melhorar a performance, você pode usar a seção Static files (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para

mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte [the section called “Arquivos estáticos”](#).

Propriedades de ambiente

A seção Environment Properties permite que você especifique definições de configuração do ambiente nas instâncias do Amazon EC2 que estão executando seu aplicativo. Essas definições são passadas para o aplicativo como pares de chave-valor.

Seu código de aplicativo pode acessar propriedades de ambiente usando `$_SERVER` ou a função `get_cfg_var`.

```
$endpoint = $_SERVER['API_ENDPOINT'];
```

Consulte [Propriedades de ambientes e outras configurações de software](#) Para mais informações.

O namespace `aws:elasticbeanstalk:container:php:phpini`

Você pode usar um [arquivo de configuração](#) para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. As opções de configuração podem ser definidas pelo serviço Elastic Beanstalk ou pela plataforma utilizada, e são organizadas em namespaces.

Você pode usar o namespace `aws:elasticbeanstalk:environment:proxy` para escolher o servidor de proxy do ambiente.

É possível usar o namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` para configurar o proxy de ambiente para servir arquivos estáticos. Defina mapeamentos de caminhos virtuais para diretórios de aplicativos.

A plataforma PHP define opções no namespace `aws:elasticbeanstalk:container:php:phpini`, inclusive as que não estão disponíveis no console do Elastic Beanstalk. `composer_options` define opções personalizadas para usar ao instalar dependências com o Composer por meio de `composer.phar install`. Para obter mais informações, incluindo as opções disponíveis, acesse <http://getcomposer.org/doc/03-cli.md#install>.

O exemplo de [arquivo de configuração](#) a seguir especifica uma opção de arquivos estáticos que mapeia um diretório chamado `staticimages` para o caminho `/images`, e mostra as configurações para cada uma das opções disponíveis no namespace `aws:elasticbeanstalk:container:php:phpini`:

Example `.ebextensions/php-settings.config`

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /images: staticimages
  aws:elasticbeanstalk:container:php:phpini:
    document_root: /public
    memory_limit: 128M
    zlib.output_compression: "Off"
    allow_url_fopen: "On"
    display_errors: "Off"
    max_execution_time: 60
    composer_options: vendor/package
```

Note

O namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` não está definido nas ramificações da plataforma PHP de AMI do Amazon Linux (anterior ao Amazon Linux 2).

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte [Opções de configuração](#) Para mais informações.

Instalar as dependências do aplicativo

Seu aplicativo pode ter dependências em outros pacotes PHP. É possível configurar a aplicação para instalar essas dependências nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) do ambiente. Como alternativa, você pode incluir as dependências do aplicativo no pacote de origem e implantá-las com o aplicativo. A seção a seguir discute cada uma dessas maneiras.

Usar um arquivo do Composer para instalar dependências em instâncias

Use um arquivo `composer.json` na raiz da origem de seu projeto para usar o Composer para instalar os pacotes exigidos pela aplicação nas instâncias do Amazon EC2 do ambiente.

Example `composer.json`

```
{
  "require": {
    "monolog/monolog": "1.0.*"
  }
}
```

Quando um arquivo `composer.json` está presente, o Elastic Beanstalk executa `composer.phar install` para instalar dependências. Você pode adicionar opções para acrescentar ao comando definindo a opção [composer_options](#) no namespace `aws:elasticbeanstalk:container:php:phpini`.

Incluir dependências no pacote de origem

Se o seu aplicativo tiver um grande número de dependências, sua instalação poderá demorar muito tempo. Isso pode aumentar a implantação e as operações de escalabilidade, porque as dependências são instaladas em cada nova instância.

Para evitar o impacto negativo no momento da implantação, use o Composer em seu ambiente de desenvolvimento para resolver dependências e instalá-las na pasta `vendor`.

Para incluir dependências em seu pacote de origem do aplicativo

1. Execute o seguinte comando:

```
% composer install
```

2. Inclua a pasta `vendor` gerada na raiz do seu pacote de origem do aplicativo.

Quando o Elastic Beanstalk encontra uma pasta `vendor` na instância, ele ignora o arquivo `composer.json` (mesmo se ele existir). O aplicativo usa as dependências da pasta `vendor`.

Atualização do Composer

Talvez seja necessário atualizar o Composer se você ver um erro ao tentar instalar pacotes com um arquivo do Composer ou se não conseguir usar a versão mais recente da plataforma. Entre as atualizações da plataforma, você pode atualizar o Composer nas instâncias do seu ambiente por meio do uso de arquivos de configuração em sua [.ebextensions](#) pasta.

Você pode atualizar automaticamente o Composer com a seguinte configuração.

```
commands:
  01updateComposer:
    command: /usr/bin/composer.phar self-update 2.7.0
```

A [configuração de opção](#) a seguir define a variável de COMPOSER_HOME ambiente, que configura a localização do cache do Composer.

```
option_settings:
  - namespace: aws:elasticbeanstalk:application:environment
    option_name: COMPOSER_HOME
    value: /home/webapp/composer-home
```

Você pode combinar os dois no mesmo arquivo de configuração em sua `.ebextensions` pasta.

Example `.ebextensions/composer.config`

```
commands:
  01updateComposer:
    command: /usr/bin/composer.phar self-update 2.7.0

option_settings:
  - namespace: aws:elasticbeanstalk:application:environment
    option_name: COMPOSER_HOME
    value: /home/webapp/composer-home
```

Note

Devido às atualizações na instalação do Composer na versão da plataforma AL2023 [de 22 de fevereiro de 2024](#) e na versão da plataforma AL2 em [28 de fevereiro de 2024](#), a atualização automática do Composer pode falhar se for COMPOSER_HOME definida quando a atualização automática for executada.

Os seguintes comandos combinados falharão na execução: `export COMPOSER_HOME=/home/webapp/composer-home && /usr/bin/composer.phar self-update 2.7.0`. No entanto, o exemplo anterior funcionará. No exemplo anterior, a configuração da opção para não `COMPOSER_HOME` será passada para a `01updateComposer` execução e não será definida quando o comando de atualização automática for executado.

Important

Se você omitir o número da versão do comando `composer.phar self-update`, o Composer será atualizado para a versão mais recente disponível a cada vez que você implantar seu código-fonte e quando novas instâncias forem provisionadas pelo Auto Scaling. Isso pode fazer com que as operações de escalabilidade e as implantações falhem se for lançada uma versão do Composer que não seja compatível com o seu aplicativo.

Para obter mais informações sobre as Plataformas PHP do Elastic Beanstalk, incluindo a versão do Composer, consulte [Versões da plataformas PHP](#) no documento AWS Elastic Beanstalk Platforms (Plataformas do Elastic Beanstalk).

Como estender o `php.ini`

Use um arquivo de configuração com um bloco `files` para adicionar um arquivo `.ini` em `/etc/php.d/` nas instâncias em seu ambiente. O arquivo de configuração principal, `php.ini`, obtém configurações de arquivos nesta pasta em ordem alfabética. Muitas extensões são habilitadas por padrão pelos arquivos nessa pasta.

Example `.ebextensions/mongo.config`

```
files:
  "/etc/php.d/99mongo.ini":
    mode: "000755"
    owner: root
    group: root
    content: |
      extension=mongo.so
```

Implantar uma aplicação Laravel no Elastic Beanstalk

O Laravel é uma estrutura de código aberto model-view-controller (MVC) para PHP. Este tutorial mostra o processo de geração de um aplicativo Laravel, sua implantação em um AWS Elastic Beanstalk ambiente e sua configuração para se conectar a uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS).

Seções

- [Pré-requisitos](#)
- [Iniciar um ambiente do Elastic Beanstalk](#)
- [Instalar o Laravel e gerar um site](#)
- [Implantar o aplicativo](#)
- [Definir as configurações do Composer](#)
- [Adicionar um banco de dados ao seu ambiente](#)
- [Limpeza](#)
- [Próximas etapas](#)

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em [Conceitos básicos do Elastic Beanstalk](#) para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt (\$) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command  
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows 10, você pode [instalar o Subsistema Windows para Linux](#) para obter uma versão do Ubuntu integrada com o Windows e o Bash.

O Laravel 6 requer PHP 7.2 ou posterior. Ele também requer as extensões PHP listadas no tópico [requisitos do servidor](#) na documentação oficial do Laravel. Siga as instruções do tópico [Configurar seu ambiente de desenvolvimento de PHP](#) para instalar o PHP e o Composer.

Para obter informações sobre suporte e manutenção do Laravel, consulte o tópico [política de suporte](#) na documentação oficial do Laravel.

Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma PHP e aceite as configurações padrão e código de exemplo.

Para iniciar um ambiente (console)

1. [Abra o console do Elastic Beanstalk usando este link pré-configurado:](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.

- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Note

O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para ter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon S3](#).

Instalar o Laravel e gerar um site

O Composer pode instalar o Laravel e criar um projeto de trabalho com um comando:

```
~$ composer create-project --prefer-dist laravel/laravel eb-laravel
```

O Composer instala o Laravel e suas dependências, e gera um projeto padrão.

Se tiver problemas com a instalação do Laravel, consulte o tópico sobre instalação na documentação oficial: <https://laravel.com/docs/6.x>

Implantar o aplicativo

Crie um [pacote de origem](#) com os arquivos criados pelo Composer. O comando a seguir cria um pacote de origem chamado `laravel-default.zip`. Ele exclui arquivos na pasta `vendor`, que tomam muito espaço e não são necessários para implantação da aplicação no Elastic Beanstalk.

```
~/eb-laravel$ zip ../laravel-default.zip -r * .[^.]* -x "vendor/*"
```

Faça upload do pacote de origem para o Elastic Beanstalk para implantar o Laravel no ambiente.

Para implantar um pacote de origem

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

Note

Para otimizar mais o pacote de origem, inicie um repositório Git e use o comando `git archive` para criar o pacote de origem. O projeto do Laravel padrão inclui um arquivo `.gitignore` que diz para o Git excluir a pasta `vendor` e outros arquivos que não são necessários para implantação.

Definir as configurações do Composer

Quando a implantação for concluída, clique no URL para abrir o aplicativo Laravel no navegador:

Forbidden

You don't have permission to access / on this server.

O que é isso? Por padrão, o Elastic Beanstalk fornece a raiz do seu projeto no caminho raiz do site. Neste caso, contudo, a página padrão (`index.php`) está em um nível abaixo na pasta `public`. Você pode verificar isso adicionando `/public` ao URL. Por exemplo, `http://laravel.us-east-2.elasticbeanstalk.com/public`.

Para fornecer a aplicação Laravel no caminho raiz, use o console do Elastic Beanstalk para configurar a raiz do documento para o site.

Para configurar a raiz do documento do seu site

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Para Document Root (Raiz do documento), insira `/public`.

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.
7. Quando a atualização for concluída, clique no URL para reabrir seu site no navegador.



Até agora, tudo tranquilo. Em seguida, adicione um banco de dados ao ambiente e configure o Laravel para se conectar a ele.

Adicionar um banco de dados ao seu ambiente

Execute uma instância de banco de dados do RDS no ambiente do Elastic Beanstalk. Você pode usar bancos de dados MySQL, SQLServer ou PostgreSQL com o Laravel no Elastic Beanstalk. Neste exemplo, usaremos MySQL.

Como adicionar uma instância de banco de dados do RDS ao ambiente do Elastic Beanstalk

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Em Engine (Mecanismo), escolha mysql.
6. Digite um username (nome de usuário) e uma password (senha) principais. O Elastic Beanstalk fornecerá esses valores para a aplicação usando propriedades de ambiente.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A criação de uma instância de banco de dados leva cerca de 10 minutos. Para obter mais informações sobre bancos de dados acoplados a um ambiente do Elastic Beanstalk, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

Enquanto isso, atualize o código-fonte para ler as informações de conexão do ambiente. O Elastic Beanstalk fornece detalhes de conexão usando variáveis de ambiente, como `RDS_HOSTNAME`, que você pode acessar na aplicação.

A configuração de banco de dados do Laravel é armazenada em um arquivo chamado `database.php` na pasta `config` no código do projeto. Encontre a entrada `mysql` e modifique as variáveis de `host`, `database`, `username`, and `password` para ler os valores correspondentes do Elastic Beanstalk:

Example `~/Eb-laravel/config/database.php`

```
...
'connections' => [

    'sqlite' => [
        'driver' => 'sqlite',
        'database' => env('DB_DATABASE', database_path('database.sqlite')),
        'prefix' => '',
    ],

    'mysql' => [
        'driver' => 'mysql',
        'host' => env('RDS_HOSTNAME', '127.0.0.1'),
        'port' => env('RDS_PORT', '3306'),
        'database' => env('RDS_DB_NAME', 'forge'),
        'username' => env('RDS_USERNAME', 'forge'),
        'password' => env('RDS_PASSWORD', ''),
        'unix_socket' => env('DB_SOCKET', ''),
        'charset' => 'utf8mb4',
        'collation' => 'utf8mb4_unicode_ci',
        'prefix' => '',
        'strict' => true,
        'engine' => null,
    ],

    ...
]
```

Para verificar se a conexão de banco de dados está configurada corretamente, adicione código a `index.php` para conectar ao banco de dados e adicionar código à resposta padrão:

Example ~/Eb-laravel/public/index.php

```
...
if(DB::connection()->getDatabaseName())
{
    echo "Connected to database ".DB::connection()->getDatabaseName();
}
$response->send();
...
```

Quando a instância de banco de dados tiver iniciado, empacote e implante o aplicativo atualizado em seu ambiente.

Como atualizar o ambiente do Elastic Beanstalk

1. Crie um novo pacote de origem:

```
~/eb-laravel$ zip ../laravel-v2-rds.zip -r * .[^.]* -x "vendor/*"
```

2. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua Região da AWS
3. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

4. Escolha Upload and Deploy.
5. Escolha Browse e faça upload de `laravel-v2-rds.zip`.
6. Escolha Implantar.

Implantar uma nova versão do seu aplicativo leva menos de um minuto. Quando a implantação for concluída, atualize a página da web novamente para verificar se a conexão com o banco de dados foi bem-sucedida:

Connected to database ebdb

Laravel

DOCUMENTATION

LARACASTS

NEWS

FORGE

GITHUB

Limpeza

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. [O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente, como instâncias do Amazon EC2, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.](#)

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

Além disso, é possível encerrar recursos do banco de dados que você criou fora do ambiente do Elastic Beanstalk. Ao encerrar uma instância de banco de dados do Amazon RDS, você poderá criar um snapshot e restaurar os dados em outra instância posteriormente.

Para encerrar sua instância de banco de dados do RDS

1. Abra o [console do Amazon RDS](#).
2. Escolha Databases (Bancos de dados).

3. Escolha a instância de banco de dados.
4. Escolha Ações e, em seguida, escolha Excluir.
5. Escolha se deseja criar um snapshot. Depois, escolha Excluir.

Próximas etapas

Para obter mais informações sobre o Laravel, acesse o site oficial do Laravel em laravel.com.

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de [linha de comando do Elastic Beanstalk](#) (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

Neste tutorial, você usou o console do Elastic Beanstalk para configurar as opções do Composer. Para tornar essa configuração parte de origem do aplicativo, você pode usar um arquivo de configuração, como o seguinte.

Example .ebextensions/composer.config

```
option_settings:  
  aws:elasticbeanstalk:container:php:phpini:  
    document_root: /public
```

Para ter mais informações, consulte [Personalização avançada de ambiente com arquivos de configuração \(.ebextensions\)](#).

A execução de uma instância de banco de dados do Amazon RDS no ambiente do Elastic Beanstalk é ideal para desenvolvimento e teste, mas vincula o ciclo de vida do banco de dados ao ambiente. Para obter instruções sobre como se conectar a um banco de dados em execução fora de seu ambiente, consulte [Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicações PHP](#).

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, [configure um nome de domínio personalizado](#) para seu ambiente e [habilite HTTPS](#) para conexões seguras.

Implantar uma aplicação CakePHP no Elastic Beanstalk

CakePHP é uma framework MVC de código aberto para PHP. Este tutorial orienta você no processo de gerar um projeto CakePHP, implantá-lo em um ambiente do Elastic Beanstalk e configurá-lo para se conectar a uma instância de banco de dados do Amazon RDS.

Seções

- [Pré-requisitos](#)
- [Iniciar um ambiente do Elastic Beanstalk](#)
- [Instalar o CakePHP e gerar um site](#)
- [Implantar o aplicativo](#)
- [Adicionar um banco de dados ao seu ambiente](#)
- [Limpeza](#)
- [Próximas etapas](#)

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em [Conceitos básicos do Elastic Beanstalk](#) para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt (\$) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command  
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows 10, você pode [instalar o Subsistema Windows para Linux](#) para obter uma versão do Ubuntu integrada com o Windows e o Bash.

O CakePHP 4 requer PHP 7.2 ou posterior. Ele também requer as extensões PHP listadas na documentação oficial de [instalação do CakePHP](#). Siga as instruções do tópico [Configurar seu ambiente de desenvolvimento de PHP](#) para instalar o PHP e o Composer.

Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma PHP e aceite as configurações padrão e código de exemplo.

Para iniciar um ambiente (console)

1. [Abra o console do Elastic Beanstalk usando este link pré-configurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.

- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Note

O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para ter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon S3](#).

Instalar o CakePHP e gerar um site

O Composer pode instalar o CakePHP e criar um projeto de trabalho com um comando:

```
~$ composer create-project --prefer-dist cakephp/app eb-cake
```

O Composer instala o CakePHP e cerca de 20 dependências e gera um projeto padrão.

Se tiver problemas ao instalar o CakePHP, visite o tópico de instalação na documentação oficial: <http://book.cakephp.org/4.0/en/installation.html>

Implantar o aplicativo

Crie um [pacote de origem](#) com os arquivos criados pelo Composer. O comando a seguir cria um pacote de origem chamado `cake-default.zip`. Ele exclui arquivos na pasta `vendor`, que tomam muito espaço e não são necessários para implantação da aplicação no Elastic Beanstalk.

```
eb-cake zip ../cake-default.zip -r * .[^.]* -x "vendor/*"
```

Faça upload do pacote de origem para o Elastic Beanstalk para implantar o CakePHP no ambiente.

Para implantar um pacote de origem

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

Note

Para otimizar mais o pacote de origem, inicie um repositório Git e use o comando `git archive` para criar o pacote de origem. O projeto do Symfony padrão inclui um arquivo `.gitignore` que diz para o Git excluir a pasta `vendor` e outros arquivos que não são necessários para implantação.

Quando o processo for concluído, clique na URL para abrir a aplicação CakePHP no navegador:

Até agora, tudo tranquilo. Em seguida, adicione um banco de dados ao ambiente e configure o CakePHP para se conectar a ele.

Adicionar um banco de dados ao seu ambiente

Execute uma instância de banco de dados do Amazon RDS no ambiente do Elastic Beanstalk. Você pode usar bancos de dados MySQL, SQLServer ou PostgreSQL com o CakePHP no Elastic Beanstalk. Neste exemplo, usaremos o PostgreSQL.

Como adicionar uma instância de banco de dados do Amazon RDS ao ambiente do Elastic Beanstalk

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Em Database (Banco de dados), escolha Edit (Editar).
5. Em DB engine, escolha postgres.
6. Digite um username (nome de usuário) e uma password (senha) principais. O Elastic Beanstalk fornecerá esses valores para a aplicação usando propriedades de ambiente.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A criação de uma instância de banco de dados leva cerca de 10 minutos. Enquanto isso, atualize o código-fonte para ler as informações de conexão do ambiente. O Elastic Beanstalk fornece detalhes

de conexão usando variáveis de ambiente, como `RDS_HOSTNAME`, que você pode acessar na aplicação.

A configuração de banco de dados do CakePHP é armazenada em um arquivo chamado `app.php` na pasta `config` no código do projeto. Abra este arquivo e adicione um código que leia as variáveis de ambiente do `$_SERVER` e as atribua às variáveis locais. Insira as linhas realçadas no exemplo abaixo após a primeira linha (`<?php`):

Example `~/Eb-cake/config/app.php`

```
<?php
if (!defined('RDS_HOSTNAME')) {
    define('RDS_HOSTNAME', $_SERVER['RDS_HOSTNAME']);
    define('RDS_USERNAME', $_SERVER['RDS_USERNAME']);
    define('RDS_PASSWORD', $_SERVER['RDS_PASSWORD']);
    define('RDS_DB_NAME', $_SERVER['RDS_DB_NAME']);
}
return [
    ...
```

A conexão do banco de dados está configurada mais abaixo no arquivo `app.php`. Encontre a seção a seguir e modifique a configuração de fontes de dados padrão com o nome do driver que corresponde ao seu mecanismo de banco de dados (`Mysql`, `Sqlserver` ou `Postgres`) e defina as variáveis `host`, `username`, `password` e `database` para ler os valores correspondentes do Elastic Beanstalk:

Example `~/Eb-cake/config/app.php`

```
...
/**
 * Connection information used by the ORM to connect
 * to your application's datastores.
 * Drivers include Mysql Postgres Sqlite Sqlserver
 * See vendor\cakephp\cakephp\src\Database\Driver for complete list
 */
'Datasources' => [
    'default' => [
        'className' => 'Cake\Database\Connection',
        'driver' => 'Cake\Database\Driver\Postgres',
        'persistent' => false,
        'host' => RDS_HOSTNAME,
```

```
/*
 * CakePHP will use the default DB port based on the driver selected
 * MySQL on MAMP uses port 8889, MAMP users will want to uncomment
 * the following line and set the port accordingly
 */
//'port' => 'non_standard_port_number',
'username' => RDS_USERNAME,
'password' => RDS_PASSWORD,
'database' => RDS_DB_NAME,
/*
 * You do not need to set this flag to use full utf-8 encoding (internal
 default since CakePHP 3.6).
 */
//'encoding' => 'utf8mb4',
'timezone' => 'UTC',
'flags' => [],
'cacheMetadata' => true,
'log' => false,
...

```

Quando a instância de banco de dados tiver iniciado, crie um pacote e implante o aplicativo atualizado no ambiente:

Como atualizar o ambiente do Elastic Beanstalk

1. Crie um novo pacote de origem:

```
~/eb-cake$ zip ../cake-v2-rds.zip -r * .[^.]* -x "vendor/*"
```

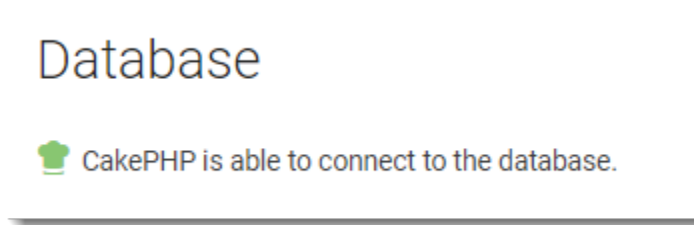
2. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
3. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

4. Escolha Upload and Deploy.
5. Escolha Browse e faça upload de `cake-v2-rds.zip`.
6. Escolha Implantar.

Implantar uma nova versão do seu aplicativo leva menos de um minuto. Quando a implantação for concluída, atualize a página da web novamente para verificar se a conexão com o banco de dados foi bem-sucedida:



Limpeza

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. [O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente, como instâncias do Amazon EC2, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.](#)

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

Além disso, é possível encerrar recursos do banco de dados que você criou fora do ambiente do Elastic Beanstalk. Ao encerrar uma instância de banco de dados do Amazon RDS, você poderá criar um snapshot e restaurar os dados em outra instância posteriormente.

Para encerrar sua instância de banco de dados do RDS

1. Abra o [console do Amazon RDS](#).
2. Escolha Databases (Bancos de dados).

3. Escolha a instância de banco de dados.
4. Escolha Ações e, em seguida, escolha Excluir.
5. Escolha se deseja criar um snapshot. Depois, escolha Excluir.

Próximas etapas

Para obter mais informações sobre o CakePHP, leia o livro em book.cakephp.org.

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de [linha de comando do Elastic Beanstalk](#) (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

A execução de uma instância de banco de dados do Amazon RDS no ambiente do Elastic Beanstalk é ideal para desenvolvimento e teste, mas vincula o ciclo de vida do banco de dados ao ambiente. Para obter instruções sobre como se conectar a um banco de dados em execução fora de seu ambiente, consulte [Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicações PHP](#).

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, [configure um nome de domínio personalizado](#) para seu ambiente e [habilite HTTPS](#) para conexões seguras.

Implantar uma aplicação Symfony no Elastic Beanstalk

O [Symfony](#) é uma framework de código aberto para o desenvolvimento de aplicações web de PHP dinâmico. Este tutorial mostra o processo de geração de um aplicativo Symfony e sua implantação em um ambiente. AWS Elastic Beanstalk

Seções

- [Pré-requisitos](#)
- [Iniciar um ambiente do Elastic Beanstalk](#)
- [Instalar o Symfony e gerar um site](#)
- [Implantar o aplicativo](#)
- [Definir as configurações do Composer](#)
- [Limpeza](#)
- [Próximas etapas](#)

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em [Conceitos básicos do Elastic Beanstalk](#) para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt (\$) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command  
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows 10, você pode [instalar o Subsistema Windows para Linux](#) para obter uma versão do Ubuntu integrada com o Windows e o Bash.

O Symfony 4.4.9 requer PHP 7.1.3 ou posterior. Ele também requer as extensões PHP listadas no tópico [requisitos técnicos](#) na documentação de instalação oficial do Symfony. Neste tutorial, usamos o PHP 7.2 e a [versão correspondente da plataforma](#) do Elastic Beanstalk. Siga as instruções do tópico [Configurar seu ambiente de desenvolvimento de PHP](#) para instalar o PHP e o Composer.

Para obter informações sobre suporte e manutenção do Symfony, consulte o tópico [lançamentos do symfony](#) no site do Symfony. Para obter mais informações sobre atualizações relacionadas ao suporte à versão do PHP para o Symfony 4.4.9, consulte o tópico [notas de release do Symfony 4.4.9](#) no site do Symfony.

Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma PHP e aceite as configurações padrão e código de exemplo.

Para iniciar um ambiente (console)

1. [Abra o console do Elastic Beanstalk usando este link pré-configurado:](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.

3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).

- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` - se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Note

O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para ter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon S3](#).

Instalar o Symfony e gerar um site

O Composer pode instalar o Symfony e criar um projeto de trabalho com um comando:

```
~$ composer create-project symfony/website-skeleton eb-symfony
```

O Composer instala o Symfony e suas dependências e gera um projeto padrão.

Se tiver problemas com a instalação do Symfony, consulte o tópico [instalação](#) na documentação oficial do Symfony.

Implantar o aplicativo

Acesse o diretório do projeto.

```
~$ cd eb-symfony
```

Crie um [pacote de origem](#) com os arquivos criados pelo Composer. O comando a seguir cria um pacote de origem chamado `symfony-default.zip`. Ele exclui arquivos na pasta `vendor`, que tomam muito espaço e não são necessários para implantação da aplicação no Elastic Beanstalk.

```
eb-symfony$ zip ../symfony-default.zip -r * .[^.]* -x "vendor/*"
```

Faça upload do pacote de origem para o Elastic Beanstalk para implantar o Symfony no ambiente.

Para implantar um pacote de origem

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

Note

Para otimizar mais o pacote de origem, inicie um repositório Git e use o comando [git archive](#) para criar o pacote de origem. O projeto do Symfony padrão inclui um arquivo `.gitignore` que diz para o Git excluir a pasta `vendor` e outros arquivos que não são necessários para implantação.

Definir as configurações do Composer

Quando a implantação for concluída, clique na URL para abrir a aplicação Symfony no navegador:

O que é isso? Por padrão, o Elastic Beanstalk fornece a raiz do seu projeto no caminho raiz do site. Neste caso, contudo, a página padrão (app.php) está em um nível abaixo na pasta web. Você pode verificar isso adicionando /public ao URL. Por exemplo, <http://symfony.us-east-2.elasticbeanstalk.com/public>.

Para fornecer a aplicação Symfony no caminho raiz, use o console do Elastic Beanstalk para configurar a raiz do documento para o site.

Para configurar a raiz do documento do seu site

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Para Document Root (Raiz do documento), insira **/public**.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.
7. Quando a atualização for concluída, clique no URL para reabrir seu site no navegador.

Limpeza

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. [O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente, como instâncias do Amazon EC2, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.](#)

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

Próximas etapas

Para obter mais informações sobre Symfony, consulte [O que é o Symfony?](#) no site symfony.com.

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de [linha de comando do Elastic Beanstalk](#) (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

Neste tutorial, você usou o console do Elastic Beanstalk para configurar as opções do Composer. Para tornar essa configuração parte de origem do aplicativo, você pode usar um arquivo de configuração, como o seguinte.

Example .ebextensions/composer.config

```
option_settings:
  aws:elasticbeanstalk:container:php:phpini:
    document_root: /public
```

Para ter mais informações, consulte [Personalização avançada de ambiente com arquivos de configuração \(.ebextensions\)](#).

O Symfony usa seus próprios arquivos de configuração para configurar as conexões do banco de dados. Para obter instruções sobre como se conectar a um banco de dados com o Symfony, consulte [Conexão a um banco de dados com Symfony](#).

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, [configure um nome de domínio personalizado](#) para seu ambiente e [habilite HTTPS](#) para conexões seguras.

Implantar uma aplicação PHP de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk

Este tutorial mostra o processo de [iniciar uma instância de banco de dados RDS](#) externa e configurar um ambiente de alta disponibilidade executando um aplicativo PHP para se conectar a ela. AWS Elastic Beanstalk executa a execução de uma instância de banco de dados externa ao Elastic Beanstalk desvincula o banco de dados do ciclo de vida do ambiente. Com isso, é possível se conectar ao mesmo banco de dados de vários ambientes, trocar um banco de dados por outro ou realizar uma implantação azul/verde sem afetar seu banco de dados.

O tutorial usa um [exemplo de aplicativo PHP](#) que usa um banco de dados MySQL para armazenar dados de texto fornecidos pelo usuário. O aplicativo de exemplo usa [arquivos de configuração](#) para definir as [configurações de PHP](#) e para criar uma tabela no banco de dados para o aplicativo usar. Ele também mostra como usar um [arquivo do Composer](#) para instalar pacotes durante a implantação.

Seções

- [Pré-requisitos](#)
- [Executar uma instância de banco de dados no Amazon RDS](#)
- [Criar um ambiente do Elastic Beanstalk](#)
- [Configurar grupos de segurança, propriedades de ambiente e escalabilidade](#)
- [Implantar o aplicativo de exemplo](#)
- [Limpeza](#)
- [Próximas etapas](#)

Pré-requisitos

[Antes de começar, baixe o pacote de origem do aplicativo de amostra em GitHub: -app-1.3.zip eb-demo-php-simple](#)

Os procedimentos neste tutorial para tarefas do Amazon Relational Database Service (Amazon RDS) assumem que você esteja iniciando recursos em uma [Amazon Virtual Private Cloud](#) (Amazon VPC) padrão. Todas as novas contas incluem uma VPC padrão em cada região. Se você não tem uma VPC padrão, os procedimentos podem variar. Consulte [Usar o Elastic Beanstalk com o Amazon RDS](#) para obter instruções para Plataformas VPC personalizadas e EC2-Classic.

Executar uma instância de banco de dados no Amazon RDS

Para usar um banco de dados externo com uma aplicação em execução no Elastic Beanstalk, primeiro execute uma instância de banco de dados com o Amazon RDS. Quando uma instância é executada com o Amazon RDS, ela é completamente independente do Elastic Beanstalk e dos seus ambientes do Elastic Beanstalk e não será encerrada nem monitorada pelo Elastic Beanstalk.

Use o console do Amazon RDS para executar uma instância de banco de dados MySQL Multi-AZ. Escolha uma implantação Multi-AZ para garantir que seu banco de dados execute o failover e continue disponível se a instância de banco de dados de origem ficar fora de serviço.

Para executar uma instância de banco de dados do RDS em uma VPC padrão

1. Abra o [console do RDS](#).
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Selecione Criar banco de dados.
4. Escolha Standard Create (Criação padrão).

Important

Não escolha Easy Create (Criação fácil). Se o escolher, você não poderá definir as configurações necessárias para iniciar esse banco de dados do RDS.

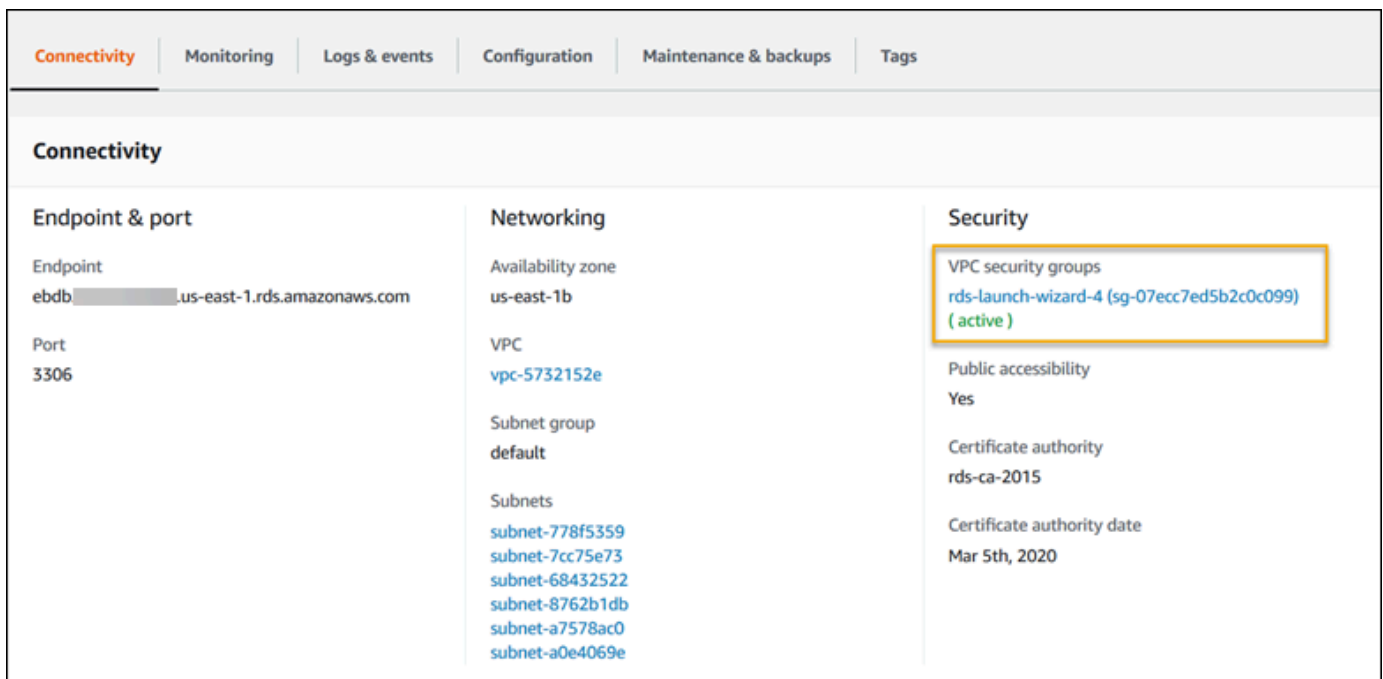
5. Em Additional configuration (Configuração adicional), para Initial database name (Nome do banco de dados inicial), digite **ebdb**.
6. Revise as configurações padrão e ajuste essas configurações de acordo com seus requisitos específicos. Preste atenção às seguintes opções:
 - DB instance class (Classe de instância de banco de dados): escolha um tamanho de instância que tem uma quantidade apropriada de memória e energia de CPU para sua workload.
 - Multi-AZ deployment (Implantação Multi-AZ): para obter alta disponibilidade, defina como Create an Aurora Replica/Reader node in a different AZ (Criar uma réplica Aurora/nó do Leitor em um AZ diferente).
 - Master username (Nome do usuário principal) e Master password (Senha principal): o nome de usuário do banco de dados e senha. Anote essas configurações porque você vai usá-las mais tarde.

7. Verifique as configurações padrão para as opções restantes e selecione Create database (Criar banco de dados).

Em seguida, modifique o security group anexado à sua instância de banco de dados para permitir o tráfego de entrada na porta apropriada. Este é o mesmo grupo de segurança a ser anexado ao ambiente do Elastic Beanstalk mais tarde, para que a regra adicionada conceda permissão de entrada a outros recursos no mesmo grupo de segurança.

Para modificar as regras de entrada no grupo de segurança anexado à instância do RDS

1. Abra o [console do Amazon RDS](#).
2. Escolha Databases (Bancos de dados).
3. Escolha o nome da sua instância de banco de dados para visualizar detalhes.
4. Na seção Connectivity (Conectividade), anote as Subnets (Sub-redes), os Security groups (Grupos de segurança) e o Endpoint que são exibidos nesta página. Faça isso para poder usar essas informações mais tarde.
5. Em Security (Segurança), você pode ver o grupo de segurança associado à instância de banco de dados. Abra o link para visualizar o grupo de segurança no console do Amazon EC2.



6. Nos detalhes do grupo de segurança, escolha a guia Inbound (Entrada).
7. Selecione a opção Editar.
8. Escolha Add Rule.

- Para Tipo, escolha o mecanismo de banco de dados usado pelo aplicativo.
- Em Source (Origem), digite **sg-** para exibir uma lista dos grupos de segurança disponíveis. Escolha o grupo de segurança associado ao grupo do Auto Scaling usado com seu ambiente do Elastic Beanstalk. Isso é para que as instâncias do Amazon EC2 no ambiente possam ter acesso ao banco de dados.

Type	Protocol	Port Range	Source	Description
MYSQL/Aurora	TCP	3306	Custom 72.21.198.67/32	e.g. SSH for Admin Desktop
MYSQL/Aurora	TCP	3306	Custom sg-	e.g. SSH for Admin Desktop

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

- Escolha Salvar.

A criação de uma instância de banco de dados leva cerca de 10 minutos. Enquanto isso, crie o ambiente do Elastic Beanstalk.

Criar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma PHP e aceite as configurações padrão e código de exemplo. Depois de iniciar o ambiente, você pode configurar o ambiente para se conectar ao banco de dados e, em seguida, implantar o aplicativo de amostra do qual você baixou GitHub.

Para iniciar um ambiente (console)

- [Abra o console do Elastic Beanstalk usando este link pré-configurado:](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
- Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
- Para Application code, escolha Sample application.
- Selecione Review and launch.

5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` - se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele. A instância de banco de dados RDS que você executou está fora do seu ambiente, portanto você é responsável por gerenciar o ciclo de vida dela.

Note


O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para ter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon S3](#).

Configurar grupos de segurança, propriedades de ambiente e escalabilidade

Adicione o security group da instância de banco de dados ao seu ambiente em execução. Esse procedimento faz com que o Elastic Beanstalk provisione novamente todas as instâncias no ambiente com o grupo de segurança adicional anexado.

Para adicionar um security group ao seu ambiente

- Execute um destes procedimentos:
 - Como adicionar um grupo de segurança usando o console do Elastic Beanstalk
 - a. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
 - b. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note


Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

- c. No painel de navegação, escolha Configuration (Configuração).
 - d. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar).
 - e. Em EC2 security groups (Grupos de segurança do EC2), escolha o grupo de segurança a ser anexado às instâncias, além do grupo de segurança da instância que o Elastic Beanstalk criará.
 - f. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.
 - g. Leia o aviso e escolha Confirm.
- Para adicionar um grupo de segurança usando um [arquivo de configuração](#), use o exemplo de arquivo [securitygroup-addexisting.config](#).

Em seguida, use as propriedades do ambiente para passar as informações de conexão para o ambiente. A aplicação demonstrativa usa um conjunto padrão de propriedades que correspondem àquelas que o Elastic Beanstalk configura quando você provisiona um banco de dados no ambiente.

Como configurar as propriedades do ambiente para uma instância de banco de dados do Amazon RDS

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Na seção Environment properties, defina as variáveis que seu aplicativo lê para construir uma string de conexão. Para compatibilidade com ambientes que têm uma instância de banco de

dados do RDS integrada, use os seguintes nomes e valores. É possível encontrar todos os valores, exceto sua senha, no [console do RDS](#).

Nome da propriedade	Descrição	Valor da propriedade
RDS_HOSTNAME	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
RDS_PORT	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
RDS_DB_NAME	O nome do banco de dado, ebdb .	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
RDS_USERNAME	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	<input type="text" value="ebdb"/> ✕
RDS_HOSTNAME	<input type="text" value="webapp-db.jxzcb5mpaniu.us-wes"/> ✕
RDS_PORT	<input type="text" value="5432"/> ✕
RDS_USERNAME	<input type="text" value="webapp-admin"/> ✕
<input type="text" value="RDS_PASSWORD"/>	<input type="text" value="kUj5uKxmWDMYc403"/> +

[Cancel](#) [Apply](#)

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Por fim, configure o grupo de Auto Scaling do seu ambiente com uma contagem mínima de instâncias mais alta. Execute pelo menos duas instâncias o tempo todo para impedir que os servidores Web em seu ambiente sejam um ponto único de falha e para permitir que você implante alterações sem colocar seu site fora do serviço.

Para configurar o grupo de Auto Scaling de seu ambiente para alta disponibilidade

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).

5. Na seção Auto Scaling group (Grupo de Auto Scaling) defina Min instances (Mínimo de instâncias) como **2**.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Implantar o aplicativo de exemplo

Agora, o ambiente está pronto para executar a aplicação demonstrativa e conectar-se ao Amazon RDS. Implante o aplicativo de exemplo em seu ambiente.

Note

[Faça o download do pacote de origem em GitHub, se ainda não o fez: eb-demo-php-simple-app-1.3.zip](#)

Para implantar um pacote de origem

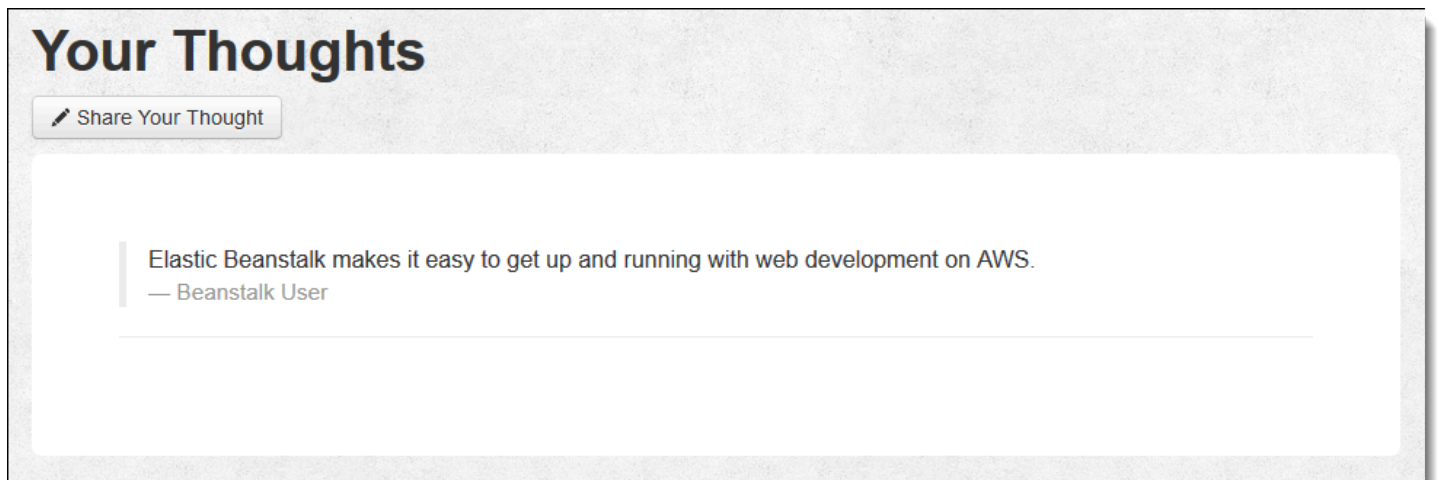
1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

O site coleta os comentários do usuário e usa um banco de dados MySQL para armazenar os dados. Para adicionar um comentário, escolha Share Your Thought (Compartilhar sua opinião), insira um comentário e escolha Submit Your Thought (Enviar sua opinião). O aplicativo Web grava o comentário no banco de dados para que qualquer instância do ambiente possa ler e ele não será perdido se as instâncias ficarem fora de serviço.



Limpeza

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. [O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente, como instâncias do Amazon EC2, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.](#)

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

Além disso, é possível encerrar recursos do banco de dados que você criou fora do ambiente do Elastic Beanstalk. Ao encerrar uma instância de banco de dados do Amazon RDS, você poderá criar um snapshot e restaurar os dados em outra instância posteriormente.

Para encerrar sua instância de banco de dados do RDS

1. Abra o [console do Amazon RDS](#).
2. Escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados.
4. Escolha Ações e, em seguida, escolha Excluir.
5. Escolha se deseja criar um snapshot. Depois, escolha Excluir.

Próximas etapas

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de [linha de comando do Elastic Beanstalk](#) (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

O aplicativo de exemplo usa os arquivos de configuração para especificar as configurações de PHP e criar uma tabela no banco de dados, se ainda não existir. Você também pode usar um arquivo de configuração para definir as configurações de security group das instâncias durante a criação do ambiente para evitar atualizações de configuração demoradas. Consulte [Personalização avançada de ambiente com arquivos de configuração \(.ebextensions\)](#) para obter mais informações.

Para desenvolvimento e teste, convém usar a funcionalidade do Elastic Beanstalk para adicionar uma instância de banco de dados gerenciada diretamente ao seu ambiente. Para obter instruções sobre como configurar um banco de dados no ambiente, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

Se você precisa de um banco de dados de alta performance, considere usar o [Amazon Aurora](#). O Amazon Aurora é um mecanismo de banco de dados compatível com MySQL que oferece recursos de banco de dados comerciais a baixo custo. Para conectar seu aplicativo a outro banco de dados, repita as etapas de [configuração de grupo de segurança](#) e [atualize as propriedades de ambiente relacionadas ao RDS](#).

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, [configure um nome de domínio personalizado](#) para seu ambiente e [habilite HTTPS](#) para conexões seguras.

Implantação de um WordPress site de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk

Este tutorial descreve como [iniciar uma instância de banco de dados Amazon RDS](#) externa e, em seguida AWS Elastic Beanstalk, como configurar um ambiente de alta disponibilidade executando um WordPress site para se conectar a ela. O site usa o Amazon Elastic File System (Amazon EFS) como o armazenamento compartilhado para arquivos carregados.

A execução de uma instância de banco de dados externa ao Elastic Beanstalk desvincula o banco de dados do ciclo de vida do ambiente. Com isso, é possível se conectar ao mesmo banco de dados de vários ambientes, trocar um banco de dados por outro ou realizar uma [implantação azul/verde](#) sem afetar seu banco de dados.

Note

Para obter informações atuais sobre a compatibilidade das versões do PHP com WordPress versões, consulte [Compatibilidade e WordPress versões do PHP](#) no WordPress site. Você deve consultar essas informações antes de atualizar para uma nova versão do PHP para suas WordPress implementações.

Tópicos

- [Pré-requisitos](#)
- [Executar uma instância de banco de dados no Amazon RDS](#)
- [Baixar WordPress](#)
- [Iniciar um ambiente do Elastic Beanstalk](#)
- [Configurar grupos de segurança e propriedades de ambiente](#)
- [Configurar e implantar seu aplicativo](#)
- [Instalar WordPress](#)
- [Atualizar chaves e salts](#)
- [Remover restrições de acesso](#)
- [Configurar o grupo de Auto Scaling](#)
- [Atualizar WordPress](#)
- [Limpar](#)

- [Próximas etapas](#)

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em [Conceitos básicos do Elastic Beanstalk](#) para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt (\$) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command  
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows 10, você pode [instalar o Subsistema Windows para Linux](#) para obter uma versão do Ubuntu integrada com o Windows e o Bash.

VPC padrão

Os procedimentos do Amazon Relational Database Service (Amazon RDS) neste tutorial assumem que você esteja iniciando recursos em uma [Amazon Virtual Private Cloud](#) (Amazon VPC) padrão. Todas as novas contas incluem uma VPC padrão em cada AWS região. Se você não tem uma VPC padrão, os procedimentos podem variar. Consulte [Usar o Elastic Beanstalk com o Amazon RDS](#) para obter instruções para Plataformas VPC personalizadas e EC2-Classik.

AWS Regiões

O aplicativo de amostra usa o Amazon EFS, que só funciona em AWS regiões que oferecem suporte ao Amazon EFS. Para saber mais sobre AWS as regiões suportadas, consulte os [endpoints e cotas do Amazon Elastic File System](#) no. Referência geral da AWS

Executar uma instância de banco de dados no Amazon RDS


Quando uma instância é executada com o Amazon RDS, ela é completamente independente do Elastic Beanstalk e dos seus ambientes do Elastic Beanstalk e não será encerrada nem monitorada pelo Elastic Beanstalk.

Nas etapas a seguir, você usará o console do Amazon RDS para:

- Iniciar um banco de dados com o mecanismo MySQL .
- Habilitar uma Multi-AZ deployment (Implantação Multi-AZ). Isso cria uma espera em uma zona de disponibilidade (AZ) diferente para fornecer redundância de dados, eliminar congelamentos de E/S e minimizar picos de latência durante backups do sistema.

Para executar uma instância de banco de dados do RDS em uma VPC padrão

1. Abra o [console do RDS](#).
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Selecione Criar banco de dados.
4. Escolha Standard Create (Criação padrão).

 Important

Não escolha Easy Create (Criação fácil). Se o escolher, você não poderá definir as configurações necessárias para iniciar esse banco de dados do RDS.

5. Em Additional configuration (Configuração adicional), para Initial database name (Nome do banco de dados inicial), digite **ebdb**.
6. Revise as configurações padrão e ajuste essas configurações de acordo com seus requisitos específicos. Preste atenção às seguintes opções:
 - DB instance class (Classe de instância de banco de dados): escolha um tamanho de instância que tem uma quantidade apropriada de memória e energia de CPU para sua workload.
 - Multi-AZ deployment (Implantação Multi-AZ): para obter alta disponibilidade, defina como Create an Aurora Replica/Reader node in a different AZ (Criar uma réplica Aurora/nó do Leitor em um AZ diferente).
 - Master username (Nome do usuário principal) e Master password (Senha principal): o nome de usuário do banco de dados e senha. Anote essas configurações porque você vai usá-las mais tarde.
7. Verifique as configurações padrão para as opções restantes e selecione Create database (Criar banco de dados).

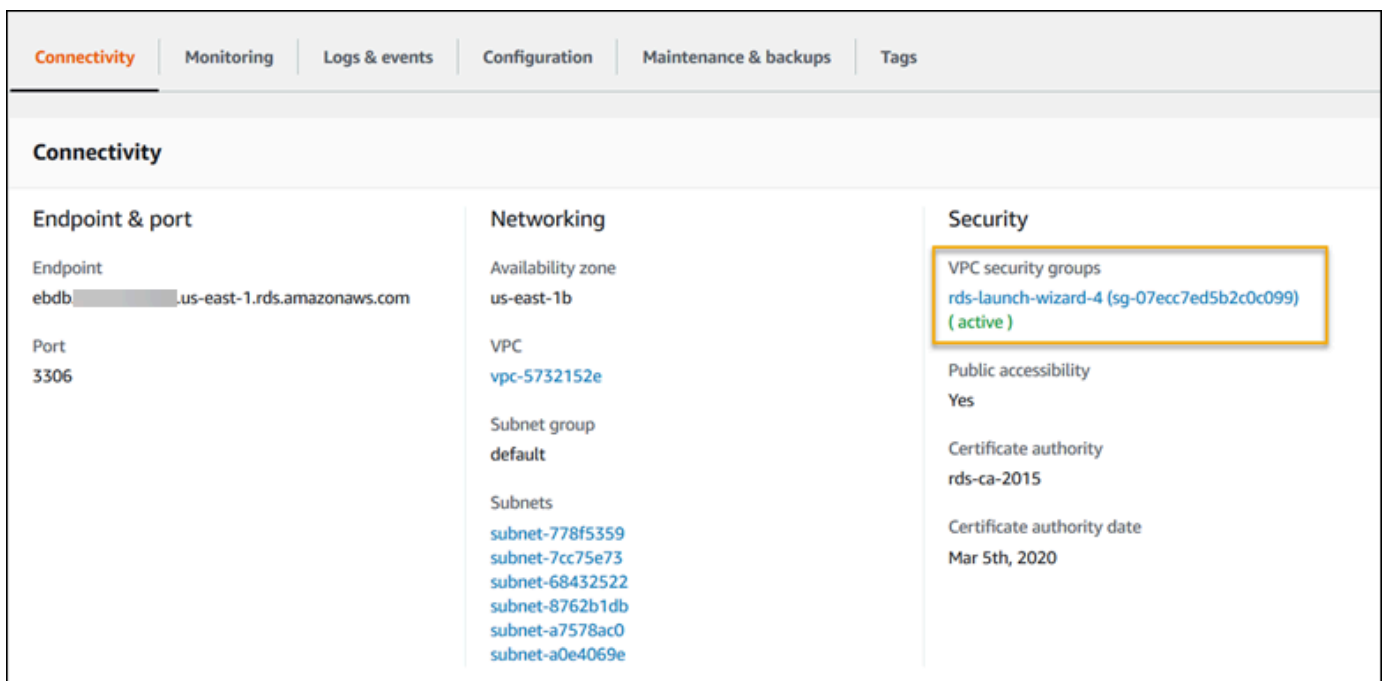
Depois que sua instância de banco de dados for criada, modifique o grupo de segurança anexado a ela para permitir o tráfego de entrada na porta apropriada.

Note

Este é o mesmo grupo de segurança a ser anexado ao ambiente do Elastic Beanstalk mais tarde, para que a regra adicionada agora conceda permissão de entrada a outros recursos no mesmo grupo de segurança.

Para modificar as regras de entrada no grupo de segurança anexado à instância do RDS

1. Abra o [console do Amazon RDS](#).
2. Escolha Databases (Bancos de dados).
3. Escolha o nome da sua instância de banco de dados para visualizar detalhes.
4. Na seção Connectivity (Conectividade), anote as Subnets (Sub-redes), os Security groups (Grupos de segurança) e o Endpoint que são exibidos nesta página. Faça isso para poder usar essas informações mais tarde.
5. Em Security (Segurança), você pode ver o grupo de segurança associado à instância de banco de dados. Abra o link para visualizar o grupo de segurança no console do Amazon EC2.



The screenshot shows the Amazon RDS console interface. At the top, there are tabs for Connectivity, Monitoring, Logs & events, Configuration, Maintenance & backups, and Tags. The 'Connectivity' tab is selected. Below the tabs, the 'Connectivity' section is displayed, divided into three columns: Endpoint & port, Networking, and Security. The 'Endpoint & port' column shows the endpoint 'ebdb-...us-east-1.rds.amazonaws.com' and port '3306'. The 'Networking' column shows 'Availability zone: us-east-1b', 'VPC: vpc-5732152e', 'Subnet group: default', and a list of subnets. The 'Security' column shows 'VPC security groups: rds-launch-wizard-4 (sg-07ecc7ed5b2c0c099) (active)', 'Public accessibility: Yes', 'Certificate authority: rds-ca-2015', and 'Certificate authority date: Mar 5th, 2020'. A yellow box highlights the 'VPC security groups' section.

6. Nos detalhes do grupo de segurança, escolha a guia Inbound (Entrada).
7. Selecione a opção Editar.
8. Escolha Add Rule.
9. Para Tipo, escolha o mecanismo de banco de dados usado pelo aplicativo.

- Em Source (Origem), digite **sg-** para exibir uma lista dos grupos de segurança disponíveis. Escolha o grupo de segurança associado ao grupo do Auto Scaling usado com seu ambiente do Elastic Beanstalk. Isso é para que as instâncias do Amazon EC2 no ambiente possam ter acesso ao banco de dados.

The screenshot shows the 'Edit inbound rules' interface in the AWS console. It features a table with columns: Type, Protocol, Port Range, Source, and Description. Two rules are listed, both for MySQL/Aurora on TCP port 3306. The 'Source' dropdown for the second rule is open, showing a search for 'sg-'. A dropdown menu is visible, listing security groups, with 'sg-07ecc7ed5b2c0c099 - rds-launch-wizard-4' selected. Below the table is an 'Add Rule' button and a note: 'NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.' At the bottom right are 'Cancel' and 'Save' buttons.

- Escolha Salvar.

A criação de uma instância de banco de dados leva cerca de 10 minutos. Enquanto isso, baixe WordPress e crie seu ambiente do Elastic Beanstalk.

Baixar WordPress

Para se preparar para a implantação WordPress usando AWS Elastic Beanstalk, você deve copiar os WordPress arquivos para o seu computador e fornecer as informações de configuração corretas.

Para criar um WordPress projeto

- Faça o download WordPress em wordpress.org.

```
~$ curl https://wordpress.org/wordpress-6.2.tar.gz -o wordpress.tar.gz
```

- Faça download dos arquivos de configuração no repositório de exemplo.

```
~$ wget https://github.com/aws-samples/eb-php-wordpress/releases/download/v1.1/eb-php-wordpress-v1.zip
```

- Extraia WordPress e altere o nome da pasta.

```
~$ tar -xvf wordpress.tar.gz
~$ mv wordpress wordpress-beanstalk
```

```
~$ cd wordpress-beanstalk
```

4. Extraia os arquivos de configuração durante a WordPress instalação.

```
~/wordpress-beanstalk$ unzip ../eb-php-wordpress-v1.zip
creating: .ebextensions/
inflating: .ebextensions/dev.config
inflating: .ebextensions/efs-create.config
inflating: .ebextensions/efs-mount.config
inflating: .ebextensions/loadbalancer-sg.config
inflating: .ebextensions/wordpress.config
inflating: LICENSE
inflating: README.md
inflating: wp-config.php
```

Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Depois de iniciar o ambiente, você pode configurá-lo para se conectar ao banco de dados e, em seguida, implantar o WordPress código no ambiente.

Nas etapas a seguir, você usará o console do Elastic Beanstalk para:

- Criar uma aplicação do Elastic Beanstalk usando a plataforma PHP gerenciada.
- Aceitar as configurações padrão e o código de exemplo.

Para iniciar um ambiente (console)

1. [Abra o console do Elastic Beanstalk usando este link pré-configurado:](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação do ambiente leva cerca de cinco minutos e cria os seguintes recursos.

Recursos criados do Elastic Beanstalk

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` - se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Como a instância de banco de dados do Amazon RDS executada está fora do ambiente, você é responsável por gerenciar o ciclo de vida dela.

Note

O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para ter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon S3](#).

Configurar grupos de segurança e propriedades de ambiente

Adicione o security group da instância de banco de dados ao seu ambiente em execução. Esse procedimento faz com que o Elastic Beanstalk provisione novamente todas as instâncias no ambiente com o grupo de segurança adicional anexado.

Para adicionar um security group ao seu ambiente

- Execute um destes procedimentos:
 - Como adicionar um grupo de segurança usando o console do Elastic Beanstalk
 - a. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
 - b. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

- c. No painel de navegação, escolha Configuration (Configuração).
 - d. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar).
 - e. Em EC2 security groups (Grupos de segurança do EC2), escolha o grupo de segurança a ser anexado às instâncias, além do grupo de segurança da instância que o Elastic Beanstalk criará.
 - f. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.
 - g. Leia o aviso e escolha Confirm.
- Para adicionar um grupo de segurança usando um [arquivo de configuração](#), use o exemplo de arquivo [securitygroup-addexisting.config](#).

Em seguida, use as propriedades do ambiente para passar as informações de conexão para o ambiente.

O WordPress aplicativo usa um conjunto padrão de propriedades que correspondem às que o Elastic Beanstalk configura quando você provisiona um banco de dados em seu ambiente.

Como configurar as propriedades do ambiente para uma instância de banco de dados do Amazon RDS

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Na seção Environment properties, defina as variáveis que seu aplicativo lê para construir uma string de conexão. Para compatibilidade com ambientes que têm uma instância de banco de

dados do RDS integrada, use os seguintes nomes e valores. É possível encontrar todos os valores, exceto sua senha, no [console do RDS](#).

Nome da propriedade	Descrição	Valor da propriedade
RDS_HOSTNAME	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
RDS_PORT	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
RDS_DB_NAME	O nome do banco de dado, ebdb .	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
RDS_USERNAME	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	<input type="text" value="ebdb"/> ✕
RDS_HOSTNAME	<input type="text" value="webapp-db.jxzc b5mpaniu.us-wes"/> ✕
RDS_PORT	<input type="text" value="5432"/> ✕
RDS_USERNAME	<input type="text" value="webapp-admin"/> ✕
<input type="text" value="RDS_PASSWORD"/>	<input type="text" value="kUj5uKxmWDMYc403"/> +

[Cancel](#) [Apply](#)

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Configurar e implantar seu aplicativo

Verifique se a estrutura da pasta `wordpress-beanstalk` está correta, conforme exibido.

```
wordpress-beanstalk$ tree -aL 1
.
### .ebextensions
### index.php
### LICENSE
### license.txt
### readme.html
### README.md
### wp-activate.php
### wp-admin
### wp-blog-header.php
### wp-comments-post.php
### wp-config.php
### wp-config-sample.php
```


```
### wp-content
### wp-cron.php
### wp-includes
### wp-links-opml.php
### wp-load.php
### wp-login.php
### wp-mail.php
### wp-settings.php
### wp-signup.php
### wp-trackback.php
### xmlrpc.php
```

O arquivo `wp-config.php` personalizado do repositório do projeto usa as variáveis de ambiente que você definiu na etapa anterior para configurar a conexão do banco de dados. A pasta `.ebextensions` contém arquivos de configuração que criam recursos adicionais no ambiente do Elastic Beanstalk.

Os arquivos de configuração devem ser modificados para funcionar com sua conta. Substitua os valores do espaço reservado nos arquivos com IDs apropriados e crie um pacote de origem.

Como atualizar os arquivos de configuração e criar um pacote de origem

1. Modifique os arquivos de configuração da seguinte forma.
 - `.ebextensions/dev.config`— Restringe o acesso ao seu ambiente para protegê-lo durante o processo WordPress de instalação. Substitua o endereço IP do espaço reservado próximo à parte superior do arquivo pelo endereço IP público do computador que você usará para acessar o site do seu ambiente e concluir a WordPress instalação.
2. Crie um [pacote de origem](#) com os arquivos em sua pasta de projeto. O comando a seguir cria um pacote de origem chamado `wordpress-beanstalk.zip`.

 Note

Dependendo da sua rede, pode ser necessário usar um bloco de endereços IP.


- `.ebextensions/efs-create.config`: cria um sistema de arquivos EFS e pontos de montagem em cada zona de disponibilidade/sub-rede na VPC. Identifique os IDs de sua VPC padrão e sub-rede no [console da Amazon VPC](#).

```
~/eb-wordpress$ zip ../wordpress-beanstalk.zip -r * .[^.]*
```

Faça o upload do pacote de origem para o Elastic WordPress Beanstalk para implantá-lo em seu ambiente.

Para implantar um pacote de origem

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note


Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

Instalar WordPress

Para concluir sua WordPress instalação

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha o URL do ambiente para abrir seu site em um navegador. Você é redirecionado para um assistente WordPress de instalação porque ainda não configurou o site.
4. Execute uma instalação padrão. O arquivo `wp-config.php` já está presente no código-fonte e configurado para ler as informações de conexão do banco de dados do ambiente, portanto você não deve ser solicitado para configurar a conexão.

A instalação leva cerca de um minuto para ser concluída.

Atualizar chaves e salts

O arquivo WordPress de configuração `wp-config.php` também lê valores de chaves e salts das propriedades do ambiente. Atualmente, essas propriedades são todas definidas como `test` pelo arquivo `wordpress.config` na pasta `.ebextensions`.

O salt de hash pode ser qualquer valor que atenda aos [requisitos de propriedade de ambiente](#), mas você não deve armazená-lo no controle de origem. Use o console do Elastic Beanstalk para definir essas propriedades diretamente no ambiente.

Para atualizar propriedades de ambiente

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuração.
4. Em Software (Software), escolha Edit (Editar).
5. Para `Environment properties`, modifique as seguintes propriedades:
 - `AUTH_KEY`: o valor escolhido para `AUTH_KEY`.
 - `SECURE_AUTH_KEY`: o valor escolhido para `SECURE_AUTH_KEY`.
 - `LOGGED_IN_KEY`: o valor escolhido para `LOGGED_IN_KEY`.
 - `NONCE_KEY`: o valor escolhido para `NONCE_KEY`.
 - `AUTH_SALT`: o valor escolhido para `AUTH_SALT`.
 - `SECURE_AUTH_SALT`: o valor escolhido para `SECURE_AUTH_SALT`.
 - `LOGGED_IN_SALT`: o valor escolhido para `LOGGED_IN_SALT`.
 - `NONCE_SALT`: o valor escolhido para `NONCE_SALT`.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Note

A definição de propriedades no ambiente substitui os valores diretamente em `wordpress.config`.

Remover restrições de acesso

O projeto de exemplo inclui o arquivo de configuração `loadbalancer-sg.config`. Ele cria um grupo de segurança e o atribui ao load balancer do ambiente, usando o endereço IP que você configurou no `dev.config`. Ele restringe o acesso HTTP na porta 80 às conexões da sua rede. Caso contrário, uma pessoa externa poderia se conectar ao seu site antes de você instalar WordPress e configurar sua conta de administrador.

Agora que você instalou WordPress, remova o arquivo de configuração para abrir o site para o mundo.

Para remover a restrição e atualizar seu ambiente

1. Exclua o arquivo `.ebextensions/loadbalancer-sg.config` de seu diretório de projeto.

```
~/wordpress-beanstalk$ rm .ebextensions/loadbalancer-sg.config
```

2. Criar um pacote de origem.

```
~/eb-wordpress$ zip ../wordpress-beanstalk-v2.zip -r * .[^.]*
```

Faça o upload do pacote de origem para o Elastic WordPress Beanstalk para implantá-lo em seu ambiente.

Para implantar um pacote de origem

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

Configurar o grupo de Auto Scaling

Por fim, configure o grupo de Auto Scaling do seu ambiente com uma contagem mínima de instâncias mais alta. Execute pelo menos duas instâncias em todos os momentos para evitar que os servidores da web no seu ambiente sejam um único ponto de falha. Isso também permite que você implante alterações sem colocar seu site fora do serviço.

Para configurar o grupo de Auto Scaling de seu ambiente para alta disponibilidade

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Na seção Auto Scaling group (Grupo de Auto Scaling) defina Min instances (Mínimo de instâncias) como **2**.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Para oferecer suporte a uploads de conteúdo em várias instâncias, o projeto demonstrativo usa o Amazon EFS para criar um sistema de arquivos compartilhados. Crie uma postagem no site e faça upload de conteúdo para armazená-lo no sistema de arquivos compartilhados. Visualize a publicação e atualize a página várias vezes para atingir ambas as instâncias e verificar se o sistema de arquivos compartilhado está funcionando.

Atualizar WordPress

Para atualizar para uma nova versão do WordPress, faça backup do seu site e implante-o em um novo ambiente.

Important

Não use a funcionalidade de atualização em seus arquivos de origem WordPress nem atualize seus arquivos de origem para usar uma nova versão. Ambas as ações podem resultar em postagens de URLs retornando erros 404, embora elas ainda estejam no banco de dados e sistema de arquivos.

Para atualizar WordPress

1. No console do WordPress administrador, use a ferramenta de exportação para exportar suas publicações para um arquivo XML.
2. Implante e instale a nova versão do WordPress Elastic Beanstalk com as mesmas etapas que você usou para instalar a versão anterior. Para evitar o tempo de inatividade, é possível criar um ambiente com a nova versão.
3. Na nova versão, instale a ferramenta WordPress Importador no console administrativo e use-a para importar o arquivo XML contendo suas postagens. Se as postagens foram criadas pelo usuário administrador na versão antiga, as atribua para o usuário administrador no novo site em vez de tentar importar o usuário administrador.
4. Se você implantou a nova versão em um ambiente separado, faça uma troca de [CNAME](#) para redirecionar os usuários de um local antigo para o novo site.

Limpar

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. [O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente, como instâncias do Amazon EC2, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.](#)

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

Além disso, é possível encerrar recursos do banco de dados que você criou fora do ambiente do Elastic Beanstalk. Ao encerrar uma instância de banco de dados do Amazon RDS, você poderá criar um snapshot e restaurar os dados em outra instância posteriormente.

Para encerrar sua instância de banco de dados do RDS

1. Abra o [console do Amazon RDS](#).
2. Escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados.
4. Escolha Ações e, em seguida, escolha Excluir.
5. Escolha se deseja criar um snapshot. Depois, escolha Excluir.

Próximas etapas

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de [linha de comando do Elastic Beanstalk](#) (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

O aplicativo de exemplo usa os arquivos de configuração para especificar as configurações de PHP e criar uma tabela no banco de dados, se ainda não existir. Você também pode usar um arquivo de configuração para definir as configurações de security group das instâncias durante a criação do ambiente para evitar atualizações de configuração demoradas. Consulte [Personalização avançada de ambiente com arquivos de configuração \(.ebextensions\)](#) para obter mais informações.

Para desenvolvimento e teste, convém usar a funcionalidade do Elastic Beanstalk para adicionar uma instância de banco de dados gerenciada diretamente ao seu ambiente. Para obter instruções sobre como configurar um banco de dados no ambiente, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

Se você precisa de um banco de dados de alta performance, considere usar o [Amazon Aurora](#). O Amazon Aurora é um mecanismo de banco de dados compatível com MySQL que oferece recursos de banco de dados comerciais a baixo custo. Para conectar seu aplicativo a outro banco de dados, repita as etapas de [configuração de grupo de segurança](#) e [atualize as propriedades de ambiente relacionadas ao RDS](#).

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, [configure um nome de domínio personalizado](#) para seu ambiente e [habilite HTTPS](#) para conexões seguras.

Implantar um site Drupal de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk

Este tutorial mostra o processo de [execução de uma instância de banco de dados RDS](#) externa a. AWS Elastic Beanstalk. Em seguida, ele descreve como configurar um ambiente de alta disponibilidade que executa um site do Drupal para se conectar a ele. O site usa o Amazon Elastic File System (Amazon EFS) como o armazenamento compartilhado para arquivos carregados. A execução de uma instância de banco de dados externa ao Elastic Beanstalk desvincula o banco de dados do ciclo de vida de seu ambiente e permite que você se conecte ao mesmo banco de dados de vários ambientes, troque de um banco de dados para outro ou realize uma implantação azul/verde sem afetar o banco de dados.

Seções

- [Pré-requisitos](#)
- [Executar uma instância de banco de dados no Amazon RDS](#)
- [Iniciar um ambiente do Elastic Beanstalk](#)
- [Definir configurações de segurança e propriedades de ambiente](#)
- [Configurar e implantar seu aplicativo](#)
- [Instalar o Drupal](#)
- [Atualizar a configuração de Drupal e remover as restrições de acesso](#)
- [Configurar o grupo de Auto Scaling](#)
- [Limpeza](#)

- [Próximas etapas](#)

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em [Conceitos básicos do Elastic Beanstalk](#) para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt (\$) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command  
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows 10, você pode [instalar o Subsistema Windows para Linux](#) para obter uma versão do Ubuntu integrada com o Windows e o Bash.

Os procedimentos neste tutorial para tarefas do Amazon Relational Database Service (Amazon RDS) assumem que você esteja iniciando recursos em uma [Amazon Virtual Private Cloud](#) (Amazon VPC) padrão. Todas as novas contas incluem uma VPC padrão em cada região. Se você não tem uma VPC padrão, os procedimentos podem variar. Consulte [Usar o Elastic Beanstalk com o Amazon RDS](#) para obter instruções para Plataformas VPC personalizadas e EC2-Classic.

A aplicação demonstrativa usa o Amazon EFS. Ele só funciona em AWS regiões que oferecem suporte ao Amazon EFS. Para saber mais sobre o suporte a AWS regiões, consulte [Amazon Elastic File System Endpoints and Quotas](#) no. Referência geral da AWS

Se a plataforma do seu ambiente do Elastic Beanstalk usa PHP 7.4 ou anterior, recomendamos que você use o Drupal versão 8.9.13 para este tutorial. Para plataformas instaladas com PHP 8.0 ou posterior, recomendamos que você use o Drupal 9.1.5.

Para obter mais informações sobre versões do Drupal e as versões do PHP compatíveis, consulte [Requisitos do PHP](#) no site do Drupal. As versões básicas que o Drupal recomenda estão listadas no site <https://www.drupal.org/project/drupal>.

Executar uma instância de banco de dados no Amazon RDS

Para usar um banco de dados externo com uma aplicação em execução no Elastic Beanstalk, primeiro execute uma instância de banco de dados com o Amazon RDS. Quando uma instância é executada com o Amazon RDS, ela é completamente independente do Elastic Beanstalk e dos seus ambientes do Elastic Beanstalk e não será encerrada nem monitorada pelo Elastic Beanstalk.

Use o console do Amazon RDS para executar uma instância de banco de dados MySQL Multi-AZ. Escolha uma implantação Multi-AZ para garantir que seu banco de dados execute o failover e continue disponível se a instância de banco de dados de origem ficar fora de serviço.

Para executar uma instância de banco de dados do RDS em uma VPC padrão

1. Abra o [console do RDS](#).
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Selecione Criar banco de dados.
4. Escolha Standard Create (Criação padrão).

Important

Não escolha Easy Create (Criação fácil). Se o escolher, você não poderá definir as configurações necessárias para iniciar esse banco de dados do RDS.

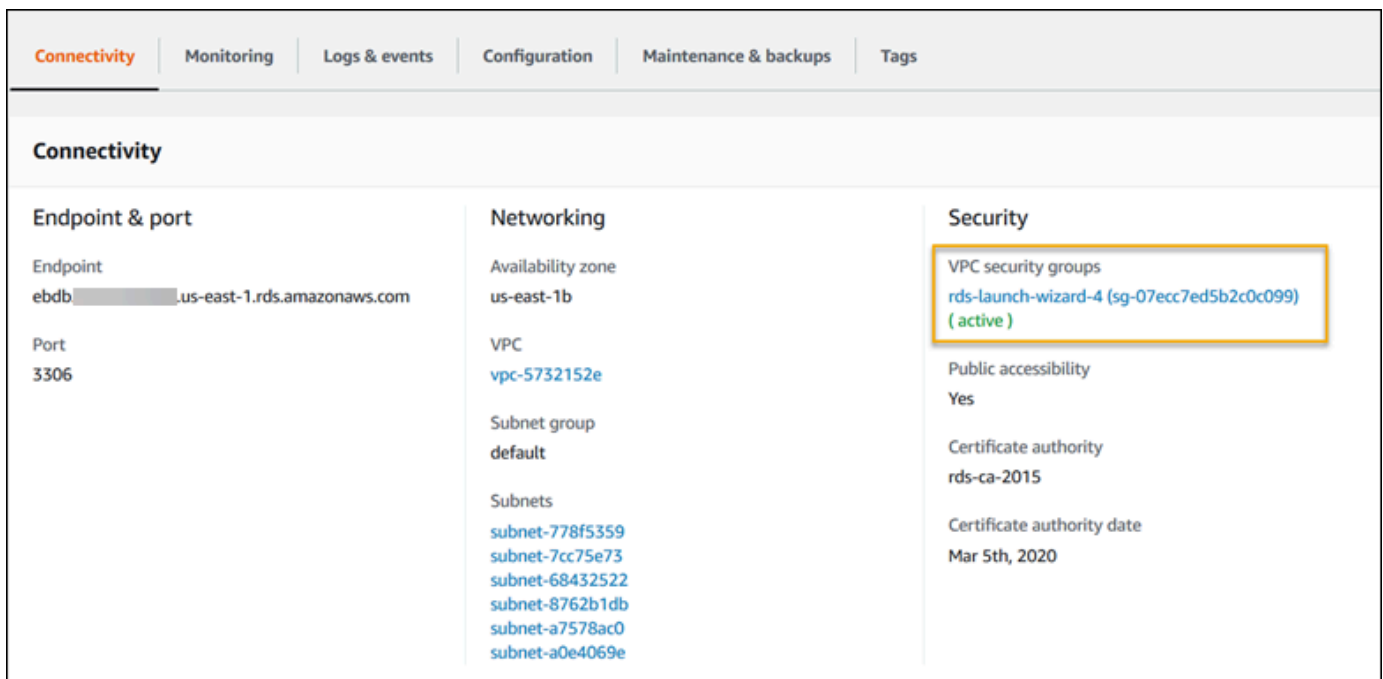
5. Em Additional configuration (Configuração adicional), para Initial database name (Nome do banco de dados inicial), digite **ebdb**.
6. Revise as configurações padrão e ajuste essas configurações de acordo com seus requisitos específicos. Preste atenção às seguintes opções:
 - DB instance class (Classe de instância de banco de dados): escolha um tamanho de instância que tem uma quantidade apropriada de memória e energia de CPU para sua workload.
 - Multi-AZ deployment (Implantação Multi-AZ): para obter alta disponibilidade, defina como Create an Aurora Replica/Reader node in a different AZ (Criar uma réplica Aurora/nó do Leitor em um AZ diferente).
 - Master username (Nome do usuário principal) e Master password (Senha principal): o nome de usuário do banco de dados e senha. Anote essas configurações porque você vai usá-las mais tarde.

7. Verifique as configurações padrão para as opções restantes e selecione Create database (Criar banco de dados).

Em seguida, modifique o security group anexado à sua instância de banco de dados para permitir o tráfego de entrada na porta apropriada. Este é o mesmo grupo de segurança a ser anexado ao ambiente do Elastic Beanstalk mais tarde, para que a regra adicionada conceda permissão de entrada a outros recursos no mesmo grupo de segurança.

Para modificar as regras de entrada no grupo de segurança anexado à instância do RDS

1. Abra o [console do Amazon RDS](#).
2. Escolha Databases (Bancos de dados).
3. Escolha o nome da sua instância de banco de dados para visualizar detalhes.
4. Na seção Connectivity (Conectividade), anote as Subnets (Sub-redes), os Security groups (Grupos de segurança) e o Endpoint que são exibidos nesta página. Faça isso para poder usar essas informações mais tarde.
5. Em Security (Segurança), você pode ver o grupo de segurança associado à instância de banco de dados. Abra o link para visualizar o grupo de segurança no console do Amazon EC2.



6. Nos detalhes do grupo de segurança, escolha a guia Inbound (Entrada).
7. Selecione a opção Editar.
8. Escolha Add Rule.

- Para Tipo, escolha o mecanismo de banco de dados usado pelo aplicativo.
- Em Source (Origem), digite **sg-** para exibir uma lista dos grupos de segurança disponíveis. Escolha o grupo de segurança associado ao grupo do Auto Scaling usado com seu ambiente do Elastic Beanstalk. Isso é para que as instâncias do Amazon EC2 no ambiente possam ter acesso ao banco de dados.

Type	Protocol	Port Range	Source	Description	
MYSQL/Aurora	TCP	3306	Custom	72.21.198.67/32	e.g. SSH for Admin Desktop
MYSQL/Aurora	TCP	3306	Custom	sg-	e.g. SSH for Admin Desktop

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

- Escolha Salvar.

A criação de uma instância de banco de dados leva cerca de 10 minutos. Enquanto isso, inicie o ambiente do Elastic Beanstalk.

Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma PHP e aceite as configurações padrão e código de exemplo. Depois de iniciar o ambiente, você poderá configurá-lo para conectar-se ao banco de dados e implantar o código Drupal no ambiente.

Para iniciar um ambiente (console)

- [Abra o console do Elastic Beanstalk usando este link pré-configurado:](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
- Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
- Para Application code, escolha Sample application.
- Selecione Review and launch.

5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` - se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele. A instância de banco de dados RDS que você executou está fora do seu ambiente, portanto você é responsável por gerenciar o ciclo de vida dela.

Note


O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para ter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon S3](#).

Definir configurações de segurança e propriedades de ambiente

Adicione o security group da instância de banco de dados ao seu ambiente em execução. Esse procedimento faz com que o Elastic Beanstalk provisione novamente todas as instâncias no ambiente com o grupo de segurança adicional anexado.

Para adicionar um security group ao seu ambiente

- Execute um destes procedimentos:
 - Como adicionar um grupo de segurança usando o console do Elastic Beanstalk
 - a. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
 - b. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note


Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

- c. No painel de navegação, escolha Configuration (Configuração).
 - d. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar).
 - e. Em EC2 security groups (Grupos de segurança do EC2), escolha o grupo de segurança a ser anexado às instâncias, além do grupo de segurança da instância que o Elastic Beanstalk criará.
 - f. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.
 - g. Leia o aviso e escolha Confirm.
- Para adicionar um grupo de segurança usando um [arquivo de configuração](#), use o exemplo de arquivo [securitygroup-addexisting.config](#).

Em seguida, use as propriedades do ambiente para passar as informações de conexão para o ambiente. A aplicação demonstrativa usa um conjunto padrão de propriedades que correspondem àquelas que o Elastic Beanstalk configura quando você provisiona um banco de dados no ambiente.

Como configurar as propriedades do ambiente para uma instância de banco de dados do Amazon RDS

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Na seção Environment properties, defina as variáveis que seu aplicativo lê para construir uma string de conexão. Para compatibilidade com ambientes que têm uma instância de banco de

dados do RDS integrada, use os seguintes nomes e valores. É possível encontrar todos os valores, exceto sua senha, no [console do RDS](#).

Nome da propriedade	Descrição	Valor da propriedade
RDS_HOSTNAME	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
RDS_PORT	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
RDS_DB_NAME	O nome do banco de dados, ebdb .	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
RDS_USERNAME	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	<input type="text" value="ebdb"/> ✕
RDS_HOSTNAME	<input type="text" value="webapp-db.jxzc b5mpaniu.us-wes"/> ✕
RDS_PORT	<input type="text" value="5432"/> ✕
RDS_USERNAME	<input type="text" value="webapp-admin"/> ✕
<input type="text" value="RDS_PASSWORD"/>	<input type="text" value="kUj5uKxmWDMYc403"/> +

[Cancel](#) [Apply](#)

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Após a instalação do Drupal, você precisa se conectar à instância com o SSH para recuperar alguns detalhes de configuração. Atribua uma chave SSH às instâncias do ambiente.

Para configurar SSH

1. Se você ainda não tiver criado um par de chaves, abra a [página de pares de chaves](#) do console do Amazon EC2 e siga as instruções para criar um.
2. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
3. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

4. No painel de navegação, escolha Configuration (Configuração).
5. Em Security (Segurança), escolha Edit (Editar).

6. Para EC2 key pair (Par de chaves EC2), escolha o par de chaves.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Configurar e implantar seu aplicativo

[Para criar um projeto do Drupal para o Elastic Beanstalk, baixe o código-fonte do Drupal e combine-o com os arquivos no repositório aws-samples/ em. eb-php-drupal](#) GitHub

Para criar um projeto Drupal

1. Execute o comando a seguir para baixar o Drupal em www.drupal.org/download. Para saber mais sobre downloads, consulte o [site do Drupal](#).

Se a plataforma do seu ambiente do Elastic Beanstalk usa PHP 7.4 ou anterior, recomendamos que você faça o download da versão 8.9.13 do Drupal para este tutorial. Execute o comando a seguir para fazer o download.

```
~$ curl https://ftp.drupal.org/files/projects/drupal-8.9.13.tar.gz -o drupal.tar.gz
```

Se sua plataforma usa PHP 8.0 ou posterior, recomendamos que você faça o download do Drupal 9.1.5. Você pode usar este comando para fazer o download.

```
~$ curl https://ftp.drupal.org/files/projects/drupal-9.1.5.tar.gz -o drupal.tar.gz
```

Para obter mais informações sobre versões do Drupal e as versões do PHP compatíveis, consulte [Requisitos do PHP](#) na documentação oficial do Drupal. As versões básicas que o Drupal recomenda estão listadas no site [do Drupal](#).

2. Use o seguinte comando para fazer o download dos arquivos de configuração do repositório de exemplo:

```
~$ wget https://github.com/aws-samples/eb-php-drupal/releases/download/v1.1/eb-php-drupal-v1.zip
```

3. Extraia o Drupal e mude o nome da pasta.

Se você baixou o Drupal 8.9.13:

```
~$ tar -xvf drupal.tar.gz
```

```
~$ mv drupal-8.9.13 drupal-beanstalk
~$ cd drupal-beanstalk
```

Se você baixou o Drupal 9.1.5:

```
~$ tar -xvf drupal.tar.gz
~$ mv drupal-9.1.5 drupal-beanstalk
~$ cd drupal-beanstalk
```

4. Extraia os arquivos de configuração durante a instalação do Drupal.

```
~/drupal-beanstalk$ unzip ../eb-php-drupal-v1.zip
creating: .ebextensions/
inflating: .ebextensions/dev.config
inflating: .ebextensions/drupal.config
inflating: .ebextensions/efs-create.config
inflating: .ebextensions/efs-filesystem.template
inflating: .ebextensions/efs-mount.config
inflating: .ebextensions/loadbalancer-sg.config
inflating: LICENSE
inflating: README.md
inflating: beanstalk-settings.php
```

Verifique se a estrutura da pasta `drupal-beanstalk` está correta, conforme exibido.

```
drupal-beanstalk$ tree -aL 1
.
### autoload.php
### beanstalk-settings.php
### composer.json
### composer.lock
### core
### .csslintrc
### .ebextensions
### .ebextensions
### .editorconfig
### .eslintignore
### .eslintrc.json
### example.gitignore
### .gitattributes
### .htaccess
```



```
### .ht.router.php
### index.php
### LICENSE
### LICENSE.txt
### modules
### profiles
### README.md
### README.txt
### robots.txt
### sites
### themes
### update.php
### vendor
### web.config
```

O arquivo `beanstalk-settings.php` do repositório do projeto usa as variáveis de ambiente que você definiu na etapa anterior para configurar a conexão do banco de dados. A pasta `.ebextensions` contém arquivos de configuração que criam recursos adicionais no ambiente do Elastic Beanstalk.

Os arquivos de configuração devem ser modificados para funcionar com sua conta. Substitua os valores do espaço reservado nos arquivos com IDs apropriados e crie um pacote de origem.

Para atualizar os arquivos de configuração e criar um pacote de origem.


1. Modifique os arquivos de configuração da seguinte forma.
 - `.ebextensions/dev.config`: restringe o acesso ao seu ambiente para o seu endereço IP a fim de protegê-lo durante o processo de instalação do Drupal. Substitua o endereço IP de espaço reservado próximo à parte superior do arquivo pelo seu endereço IP público.
 - `.ebextensions/efs-create.config`: cria um sistema de arquivos EFS e pontos de montagem em cada zona de disponibilidade/sub-rede na VPC. Identifique os IDs de sua VPC padrão e sub-rede no [console da Amazon VPC](#).
2. Crie um [pacote de origem](#) com os arquivos em sua pasta de projeto. O comando a seguir cria um pacote de origem chamado `drupal-beanstalk.zip`. Ele exclui arquivos na pasta `vendor`, que tomam muito espaço e não são necessários para implantação da aplicação no Elastic Beanstalk.

```
~/eb-drupal$ zip ../drupal-beanstalk.zip -r * .[^.]* -x "vendor/*"
```

Faça upload do pacote de origem para o Elastic Beanstalk para implantar o Drupal no ambiente.

Para implantar um pacote de origem

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note


Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

Instalar o Drupal

Para concluir a instalação do Drupal

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha o URL do ambiente para abrir seu site em um navegador. Você será redirecionado para um assistente de instalação do Drupal, pois o site ainda não foi configurado.
4. Realize a instalação padrão usando as seguintes configurações para o banco de dados:
 - Nome do banco de dados: o DB Name (Nome do banco de dados) exibido no console do Amazon RDS.
 - Nome de usuário e senha do banco de dados: os valores de Master username (Nome do usuário principal) e Master Password (Senha principal) inseridos ao criar o banco de dados.

- **Advanced Options (Opções avançadas) > Host:** o Endpoint da instância de banco de dados exibida no console do Amazon RDS.

A instalação leva cerca de um minuto para ser concluída.

Atualizar a configuração de Drupal e remover as restrições de acesso

O processo de instalação do Drupal criou um arquivo chamado `settings.php` na pasta `sites/default` na instância. Você precisa desse arquivo em seu código-fonte para evitar redefinir seu site em implantações subsequentes, mas o arquivo atualmente contém segredos os quais você não deseja confirmar a origem. Conecte-se à instância do aplicativo para recuperar informações do arquivo de configurações.

Para conectar sua instância de aplicativo com SSH

1. Abra a [página instâncias](#) do console do Amazon EC2.
2. Escolha a instância do aplicativo. É a que recebeu o nome do seu ambiente do Elastic Beanstalk.
3. Selecione Conectar.
4. Siga as instruções para conectar a instância com SSH. O comando será semelhante à seguinte.

```
$ ssh -i ~/.ssh/mykey ec2-user@ec2-00-55-33-222.us-west-2.compute.amazonaws.com
```

Obtenha o id do diretório de sincronização da última linha do arquivo de configurações.

```
[ec2-user ~]$ tail -n 1 /var/app/current/sites/default/settings.php
$config_directories['sync'] = 'sites/default/files/
config_4ccfX2sPQm79p1mk5IbUq9S_FokcEN04mxyC-L18-4g_xKj_7j9ydn31kD0Y0gnzMu071Tvc4Q/
sync';
```

O arquivo também contém os sites da chave de hash atual, mas você pode ignorar o valor atual e usar seu próprio.

Atribua o caminho do diretório de sincronização e chave de hash para as propriedades de ambiente. O arquivo de configurações personalizadas do repositório do projeto lê essas propriedades para configurar o site durante a implantação, além das propriedades de conexão de banco de dados que você definiu anteriormente.

Propriedades de configuração do Drupal

- SYNC_DIR: o caminho para o diretório de sincronização.
- HASH_SALT: qualquer valor de string que atenda aos [requisitos de propriedade de ambiente](#).

Como configurar as propriedades do ambiente no console do Elastic Beanstalk

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Role para baixo até Propriedades do ambiente.
6. Selecione Adicionar propriedade do ambiente.
7. Insira os pares Nome e Valor da propriedade.
8. Se for necessário adicionar mais variáveis, repita a Etapa 6 e a Etapa 7.
9. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Finalmente, o projeto de amostra inclui um arquivo de configuração (`loadbalancer-sg.config`) que cria um security group e o atribui ao load balancer do ambiente, usando o endereço IP configurado em `dev.config` para restringir o acesso HTTP na porta 80 às conexões da sua rede. Caso contrário, uma parte externa pode se conectar em seu sites antes de você ter instalado o Drupal e configurado sua conta de admin.

Para atualizar a configuração do Drupal e remover restrições de acesso

1. Exclua o arquivo `.ebextensions/loadbalancer-sg.config` de seu diretório de projeto.

```
~/drupal-beanstalk$ rm .ebextensions/loadbalancer-sg.config
```

2. Copie o arquivo `settings.php` personalizado na pasta dos sites.

```
~/drupal-beanstalk$ cp beanstalk-settings.php sites/default/settings.php
```

3. Criar um pacote de origem.

```
~/eb-drupal$ zip ../drupal-beanstalk-v2.zip -r * .[^.]* -x "vendor/*"
```

Faça upload do pacote de origem para o Elastic Beanstalk para implantar o Drupal no ambiente.

Para implantar um pacote de origem

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

Configurar o grupo de Auto Scaling

Por fim, configure o grupo de Auto Scaling do seu ambiente com uma contagem mínima de instâncias mais alta. Execute pelo menos duas instâncias o tempo todo para impedir que os servidores Web em seu ambiente sejam um ponto único de falha e para permitir que você implante alterações sem colocar seu site fora do serviço.

Para configurar o grupo de Auto Scaling de seu ambiente para alta disponibilidade

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Na seção Auto Scaling group (Grupo de Auto Scaling) defina Min instances (Mínimo de instâncias) como **2**.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Para oferecer suporte a uploads de conteúdo em várias instâncias, o projeto demonstrativo usa o Amazon Elastic File System para criar um sistema de arquivos compartilhados. Crie uma postagem no site e faça upload de conteúdo para armazená-lo no sistema de arquivos compartilhados. Visualize a publicação e atualize a página várias vezes para atingir ambas as instâncias e verificar se o sistema de arquivos compartilhado está funcionando.

Limpeza

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. [O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente, como instâncias do Amazon EC2, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.](#)

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

Além disso, é possível encerrar recursos do banco de dados que você criou fora do ambiente do Elastic Beanstalk. Ao encerrar uma instância de banco de dados do Amazon RDS, você poderá criar um snapshot e restaurar os dados em outra instância posteriormente.

Para encerrar sua instância de banco de dados do RDS

1. Abra o [console do Amazon RDS](#).
2. Escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados.
4. Escolha Ações e, em seguida, escolha Excluir.
5. Escolha se deseja criar um snapshot. Depois, escolha Excluir.

Próximas etapas

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de [linha de comando do Elastic Beanstalk](#) (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

O aplicativo de exemplo usa os arquivos de configuração para especificar as configurações de PHP e criar uma tabela no banco de dados, se ainda não existir. Você também pode usar um arquivo de configuração para definir as configurações de security group da instância durante a criação do ambiente para evitar atualizações de configuração demoradas. Consulte [Personalização avançada de ambiente com arquivos de configuração \(.ebextensions\)](#) para obter mais informações.

Para desenvolvimento e teste, convém usar a funcionalidade do Elastic Beanstalk para adicionar uma instância de banco de dados gerenciada diretamente ao seu ambiente. Para obter instruções sobre como configurar um banco de dados no ambiente, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

Se você precisa de um banco de dados de alta performance, considere usar o [Amazon Aurora](#). O Amazon Aurora é um mecanismo de banco de dados compatível com MySQL que oferece recursos de banco de dados comerciais a baixo custo. Para conectar seu aplicativo a outro banco de dados, repita as etapas de [configuração de grupo de segurança](#) e [atualize as propriedades de ambiente relacionadas ao RDS](#).

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, [configure um nome de domínio personalizado](#) para seu ambiente e [habilite HTTPS](#) para conexões seguras.

Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicações PHP

É possível usar uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS) para armazenar dados coletados e modificados pela aplicação. O banco de dados pode ser acoplado ao seu ambiente e gerenciado pelo Elastic Beanstalk ou pode ser criado como desacoplado e gerenciado externamente por outro serviço. Este tópico fornece instruções para criar o Amazon RDS usando o console do Elastic Beanstalk. O banco de dados será acoplado ao ambiente e gerenciado pelo Elastic Beanstalk. Para obter mais informações sobre como integrar um Amazon RDS com o Elastic Beanstalk, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

Seções

- [Adicionar uma instância de banco de dados ao seu ambiente](#)
- [Download de um driver](#)
- [Conexão com banco de dados com um PDO ou MySQLi](#)
- [Conexão a um banco de dados com Symfony](#)

Adicionar uma instância de banco de dados ao seu ambiente

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização do ambiente for concluída, o nome de host da instância de banco de dados e outras informações de conexão estarão disponíveis para o seu aplicativo por meio das seguintes propriedades de ambiente:

Nome da propriedade	Descrição	Valor da propriedade
RDS_HOSTNAME	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
RDS_PORT	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
RDS_DB_NAME	O nome do banco de dados, ebdb .	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
RDS_USERNAME	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Para obter mais informações sobre como configurar uma instância de banco de dados acoplada a um ambiente Elastic Beanstalk, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

Download de um driver

Para usar Objetos de dados PHP (PDO) para se conectar ao banco de dados, instale o driver que corresponde ao mecanismo de banco de dados que você escolheu.

- MySQL – [PDO_MYSQL](#)
- PostgreSQL – [PDO_PGSQL](#)
- Oracle: [PDO_OCI](#)
- SQL Server – [PDO_SQLSRV](#)

Para obter mais informações, consulte <http://php.net/manual/en/pdo.installation.php>.

Conexão com banco de dados com um PDO ou MySQLi

Você pode usar `$_SERVER['VARIABLE']` para ler as informações de conexão do ambiente.

Para obter um PDO, crie um Nome da Fonte de Dados (DSN) do host, porta e nome. Passe o DSN para o [construtor do PDO](#) com o nome de usuário e senha do banco de dados.

Exemplo Conexão com banco de dados RDS com PDO - MySQL

```
<?php
$dbhost = $_SERVER['RDS_HOSTNAME'];
$dbport = $_SERVER['RDS_PORT'];
$dbname = $_SERVER['RDS_DB_NAME'];
$charset = 'utf8' ;

$dsn = "mysql:host={$dbhost};port={$dbport};dbname={$dbname};charset={$charset}";
$username = $_SERVER['RDS_USERNAME'];
$password = $_SERVER['RDS_PASSWORD'];

$pdo = new PDO($dsn, $username, $password);
?>
```

Para outros drivers, substitua `mysql` pelo nome do driver: `pgsql`, `oci` ou `sqlsrv`.

Para MySQLi, passe o nome de host, nome de usuário, senha, nome do banco de dados e a porta para o construtor `mysqli`.

Exemplo Conexão com banco de dados RDS com mysqli_connect()

```
$link = new mysqli($_SERVER['RDS_HOSTNAME'], $_SERVER['RDS_USERNAME'],  
$_SERVER['RDS_PASSWORD'], $_SERVER['RDS_DB_NAME'], $_SERVER['RDS_PORT']);
```

Conexão a um banco de dados com Symfony

Para o Symfony versão 3.2 e mais recente, é possível usar `%env(PROPERTY_NAME)%` para definir os parâmetros de banco de dados em um arquivo de configuração com base nas propriedades do ambiente definidas pelo Elastic Beanstalk.

Exemplo app/config/parameters.yml

```
parameters:  
  database_driver:   pdo_mysql  
  database_host:    '%env(RDS_HOSTNAME)%'  
  database_port:    '%env(RDS_PORT)%'  
  database_name:    '%env(RDS_DB_NAME)%'  
  database_user:    '%env(RDS_USERNAME)%'  
  database_password: '%env(RDS_PASSWORD)%'
```

Consulte [Parâmetros externos \(Symfony 3.4\)](#) para obter mais informações.

Para versões anteriores do Symfony, as variáveis de ambiente estão acessíveis somente se elas começarem com `SYMFONY__`. Isso significa que as propriedades do ambiente definidas pelo Elastic Beanstalk não podem ser acessadas, e você deve definir suas próprias propriedades do ambiente para passar as informações de conexão para o Symfony.

Para se conectar a um banco de dados com o Symfony 2, [Crie uma propriedade de ambiente](#) para cada parâmetro. Em seguida, use `%property.name%` para acessar a variável transformada por Symfony em um arquivo de configuração. Por exemplo, uma propriedade de ambiente chamada `SYMFONY__DATABASE__USER` é acessível como `database.user`.

```
database_user:    "%database.user%"
```

Consulte [Parâmetros externos \(Symfony 2.8\)](#) para obter mais informações.

Como trabalhar com Python

Esta seção fornece tutoriais e informações sobre como implantar aplicações em Python usando a AWS Elastic Beanstalk Elastic Beanstalk.

Os tópicos neste capítulo pressupõem que você tem um nível básico de conhecimento sobre ambientes do Elastic Beanstalk. Se você não tiver usado o Elastic Beanstalk antes, tente o [tutorial de conceitos básicos](#) para saber os princípios.

Tópicos

- [Configurar o ambiente de desenvolvimento Python](#)
- [Usar a plataforma Python do Elastic Beanstalk](#)
- [Implantar uma aplicação Flask no Elastic Beanstalk](#)
- [Implantar uma aplicação Django no Elastic Beanstalk](#)
- [Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicações Python](#)
- [Ferramentas e recursos do Python](#)

Configurar o ambiente de desenvolvimento Python

Configure um ambiente de desenvolvimento Python para testar a aplicação localmente antes de implantá-lo no AWS Elastic Beanstalk. Este tópico descreve as etapas de configuração do ambiente de desenvolvimento e inclui links para páginas de instalação de ferramentas úteis.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt (\$) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command  
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows 10, você pode [instalar o Subsistema Windows para Linux](#) para obter uma versão do Ubuntu integrada com o Windows e o Bash.

Para ver as etapas de configuração comuns e as ferramentas que se aplicam a todas as linguagens, consulte [Configurar a máquina de desenvolvimento](#).

Seções

- [Pré-requisitos](#)
- [Usar um ambiente virtual](#)
- [Configurar um projeto Python para o Elastic Beanstalk](#)

Pré-requisitos

Para todas as aplicações Python que você implantará com o Elastic Beanstalk, estes pré-requisitos são comuns:

1. Uma versão Python correspondente à versão da plataforma Python do Elastic Beanstalk que a aplicação usará.
2. O utilitário `pip`, que corresponde à sua versão do Python. Ele é usado para instalar e listar as dependências de seu projeto, dessa forma, o Elastic Beanstalk sabe como configurar o ambiente da aplicação.
3. A Interface da Linha de Comando AWS Elastic Beanstalk (EB CLI) Ele é usado para inicializar a aplicação com os arquivos necessários para a implantação com o Elastic Beanstalk.
4. Uma instalação `ssh` de trabalho. Ela é usada para conectar-se às suas instâncias em execução quando você precisa examinar ou depurar uma implantação.
5. O pacote `virtualenv`. Ele é usado para criar um ambiente utilizado para desenvolver e testar a aplicação de forma que o ambiente possa ser replicado pelo Elastic Beanstalk sem instalar pacotes extras que não são necessários para a aplicação. Instalar esse pacote com o seguinte comando:

```
$ pip install virtualenv
```

Para obter instruções sobre como instalar o Python, o `pip` e a CLI do EB, consulte [Instalar a EB CLI](#).

Usar um ambiente virtual

Quando você tiver os pré-requisitos instalados, configure um ambiente virtual com `virtualenv` para instalar as dependências de seu aplicativo. Usando um ambiente virtual, você pode distinguir exatamente quais pacotes são necessários ao seu aplicativo para que eles sejam instalados nas instâncias do EC2 que executam seu aplicativo.

Para configurar um ambiente virtual

1. Abra uma janela de linha de comando e digite:

```
$ virtualenv /tmp/eb_python_app
```

Substitua *eb_python_app* por um nome que faça sentido para a aplicação (usar o nome da aplicação é uma boa ideia). O comando `virtualenv` cria um ambiente virtual para você no diretório especificado e imprime os resultados de suas ações:

```
Running virtualenv with interpreter /usr/bin/python
New python executable in /tmp/eb_python_app/bin/python3.7
Also creating executable in /tmp/eb_python_app/bin/python
Installing setuptools, pip...done.
```

2. Quando seu ambiente virtual estiver pronto, inicie-o executando o script `activate` localizado no diretório `bin` do ambiente. Por exemplo, para iniciar o ambiente `eb_python_app` criado na etapa anterior, você digita:

```
$ source /tmp/eb_python_app/bin/activate
```

O ambiente virtual imprime o nome dele (por exemplo: `(eb_python_app)`) no início de cada prompt de comando, lembrando que você está em um ambiente Python virtual.

3. Para parar de usar seu ambiente virtual e voltar ao interpretador Python padrão do sistema com todas as bibliotecas instaladas, execute o comando `deactivate`.

```
(eb_python_app) $ deactivate
```

Note

Após a criação, você poderá reiniciar o ambiente virtual a qualquer momento executando o script `activate` novamente.

Configurar um projeto Python para o Elastic Beanstalk

É possível usar a CLI do Elastic Beanstalk para preparar as aplicações Python para implantação com o Elastic Beanstalk.

Como configurar uma aplicação Python para implantação com o Elastic Beanstalk

1. Em seu [ambiente virtual](#), volte para a parte superior da árvore de diretórios do projeto (python_eb_app) e digite:

```
pip freeze >requirements.txt
```

Este comando copia os nomes e versões dos pacotes que estão instalados em seu ambiente virtual para `requirements.txt`. Por exemplo, se o pacote PyYAML, versão 3.11 estiver instalado em seu ambiente virtual, o arquivo conterá a linha:

```
PyYAML==3.11
```

Dessa forma, o Elastic Beanstalk pode replicar o ambiente Python da aplicação usando os mesmos pacotes e as mesmas versões que você usou para desenvolver e testar a aplicação.

2. Configure o repositório da EB CLI com o comando `eb init`. Siga os prompts para escolher a região, a plataforma e outras opções. Para obter instruções detalhadas, consulte [Gerenciar ambientes do Elastic Beanstalk com a CLI do EB](#).

Por padrão, o Elastic Beanstalk procura um arquivo chamado `application.py` para iniciar a aplicação. Se ele não existir no projeto Python que você criou, alguns ajustes no ambiente do seu aplicativo serão necessários. Você também precisará definir variáveis de ambiente para que os módulos do aplicativo possam ser carregados. Consulte [Usar a plataforma Python do Elastic Beanstalk](#) para obter mais informações.

Usar a plataforma Python do Elastic Beanstalk

A plataforma Python do AWS Elastic Beanstalk é um conjunto de [versões da plataforma](#) para aplicações web Python que podem ser executadas com um servidor de proxy com WSGI. Cada ramificação da plataforma corresponde a uma versão do Python, como Python 3.8.

Começando com as ramificações de plataforma do Amazon Linux 2, o Elastic Beanstalk fornece o [Gunicorn](#) como o servidor WSGI padrão.

Você pode adicionar um `Procfile` ao seu pacote de origem para especificar e configurar o servidor WSGI para seu aplicativo. Para obter mais detalhes, consulte [the section called “Procfile”](#).

Você pode usar os arquivos `Pipfile.lock` e `Pipfile` criados pelo Pipenv para especificar dependências de pacotes Python e outros requisitos. Para obter detalhes sobre como especificar dependências, consulte [the section called “Especificação de dependências”](#).

O Elastic Beanstalk fornece [opções de configuração](#) que podem ser usadas para personalizar o software que é executado nas instâncias do EC2 no ambiente do Elastic Beanstalk. Você pode configurar variáveis de ambiente necessárias à aplicação, habilitar a alternância de logs para o Amazon S3 e mapear pastas da origem da aplicação que contenham arquivos estáticos para caminhos atendidos pelo servidor de proxy.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para [modificar a configuração de um ambiente em execução](#). Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar [configurações salvas](#) para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir [arquivos de configuração](#). As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console. Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte [Opções de configuração](#).

Para os pacotes do Python disponíveis no `pip`, você também pode incluir um arquivo de requisitos na raiz do código-fonte da aplicação. O Elastic Beanstalk instala todos os pacotes de dependência especificados em um arquivo de requisitos durante a implantação. Para obter mais detalhes, consulte [the section called “Especificação de dependências”](#).

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte [the section called “Estender plataformas Linux”](#).

Configurar o ambiente Python

As configurações da plataforma Python permitem que você ajuste o comportamento das instâncias do Amazon EC2. É possível editar a configuração da instância do Amazon EC2 do ambiente do Elastic Beanstalk usando o console do Elastic Beanstalk.

Use o console do Elastic Beanstalk para configurar as configurações do processo Python, habilitar o AWS X-Ray, habilitar a alternância de logs para o Amazon S3 e configurar variáveis que a aplicação pode ler no ambiente.

Como configurar o ambiente Python no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Configurações do Python

- Proxy server (Servidor de proxy): o servidor de proxy a ser usado nas instâncias de ambiente. Por padrão, é usado nginx.
- WSGI Path (Caminho do WSGI): o nome ou o caminho para o arquivo de aplicação principal. Por exemplo, `application.py` ou `django/wsgi.py`.
- NumProcesses: o número de processos a serem executados em cada instância da aplicação.
- NumThreads: o número de threads a serem executados em cada processo.

Configurações do AWS X-Ray

- X-Ray daemon: execute o daemon do AWS X-Ray para processar dados de rastreamento do [AWS X-Ray SDK for Python](#).

Opções de log

A seção Log Options tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Enable log file rotation to Amazon S3 (Habilitar a rotação de arquivos de log para o Amazon S3): especifica se os arquivos de log das instâncias do Amazon EC2 da aplicação serão copiados para o bucket do Amazon S3 associado a ela.

Arquivos estáticos

Para melhorar a performance, você pode usar a seção Static files (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte [the section called “Arquivos estáticos”](#).

Por padrão, o servidor de proxy em um ambiente Python atende todos os arquivos de uma pasta chamada `static` no caminho `/static`. Por exemplo, se a origem do aplicativo contiver um arquivo chamado `logo.png` em uma pasta chamada `static`, o servidor de proxy o servirá para os usuários em `subdomain.elasticbeanstalk.com/static/logo.png`. Você pode configurar mapeamentos adicionais conforme explicado nesta seção.

Propriedades de ambiente

Você pode usar as propriedades de ambiente para fornecer informações ao aplicativo e configurar variáveis de ambiente. Por exemplo, crie uma propriedade de ambiente chamada `CONNECTION_STRING` que especifica uma string de conexão que o aplicativo pode usar para se conectar a um banco de dados.

No ambiente do Python em execução no Elastic Beanstalk, esses valores podem ser acessados usando o dicionário do Python `os.environ`. Para obter mais informações, consulte <http://docs.python.org/library/os.html>.

Você pode usar um código semelhante ao mostrado abaixo para acessar as chaves e os parâmetros:

```
import os
endpoint = os.environ['API_ENDPOINT']
```

As propriedades de ambiente também podem fornecer informações para uma framework. Por exemplo, crie uma propriedade chamada `DJANGO_SETTINGS_MODULE` para configurar o Django a usar um módulo de configurações específicas. Dependendo do ambiente, o valor pode ser `development.settings`, `production.settings` etc.

Consulte [Propriedades de ambientes e outras configurações de software](#) para obter mais informações.

Namespaces de configuração do Python

Você pode usar um [arquivo de configuração](#) para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. As opções de configuração podem ser definidas pelo serviço Elastic Beanstalk ou pela plataforma utilizada, e são organizadas em namespaces.

A plataforma Python define opções nos namespaces

```
aws:elasticbeanstalk:environment:proxy,
aws:elasticbeanstalk:environment:proxy:staticfiles e
aws:elasticbeanstalk:container:python.
```

O arquivo de configuração de exemplo a seguir especifica as definições da opção de configuração para criar uma propriedade de ambiente chamada `DJANGO_SETTINGS_MODULE`, selecionar o servidor de proxy Apache, especificar duas opções de arquivos estáticos que mapeiam um diretório chamado `statichtml` para o caminho `/html` e um diretório chamado `staticimages` para o caminho `/images` e especificar configurações adicionais no namespace [aws:elasticbeanstalk:container:python](#). Esse namespace contém opções que permitem especificar a localização do script de WSGI no código-fonte e o número de threads e processos para executar no WSGI.

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    DJANGO_SETTINGS_MODULE: production.settings
  aws:elasticbeanstalk:environment:proxy:
```

```
ProxyServer: apache
aws:elasticbeanstalk:environment:proxy:staticfiles:
  /html: statichtml
  /images: staticimages
aws:elasticbeanstalk:container:python:
  WSGIPath: ebdjango.wsgi:application
  NumProcesses: 3
  NumThreads: 20
```

Observações

- Se você estiver usando uma versão de plataforma Python da AMI do Amazon Linux (anterior ao Amazon Linux 2), substitua o valor `WSGIPath` por `ebdjango/wsgi.py`. O valor no exemplo funciona com o servidor WSGI do Gunicorn, que não é mais compatível nas versões da plataforma da AMI do Amazon Linux.
- Além disso, essas versões de plataforma mais antigas usam um namespace diferente para configurar arquivos estáticos: `aws:elasticbeanstalk:container:python:staticfiles`. Ele tem os mesmos nomes de opção e semântica que o namespace de arquivo estático padrão.

Os arquivos de configuração também são compatíveis com várias chaves para [modificar mais o software nas instâncias do ambiente](#). Este exemplo usa a chave dos [pacotes](#) para instalar o Memcached com yum e [comandos do contêiner](#) para executar comandos que configuram o servidor durante a implantação:

```
packages:
  yum:
    libmemcached-devel: '0.31'

container_commands:
  collectstatic:
    command: "django-admin.py collectstatic --noinput"
  01syncdb:
    command: "django-admin.py syncdb --noinput"
    leader_only: true
  02migrate:
    command: "django-admin.py migrate"
    leader_only: true
  03wsgipass:
```

```
command: 'echo "WSGIPassAuthorization On" >> ../wsgi.conf'
99customize:
  command: "scripts/customize.sh"
```

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte [Opções de configuração](#) para obter mais informações.

Configurar o servidor WSGI com um Procfile

Você pode adicionar um Procfile ao seu pacote de origem para especificar e configurar o servidor WSGI para seu aplicativo. O exemplo a seguir usa um Procfile para especificar uWSGI como o servidor e configurá-lo.

Example Procfile

```
web: uwsgi --http :8000 --wsgi-file application.py --master --processes 4 --threads 2
```

O exemplo a seguir usa um Procfile para configurar o Gunicorn, o servidor WSGI padrão.

Example Procfile

```
web: gunicorn --bind :8000 --workers 3 --threads 2 project.wsgi:application
```

Observações

- Se você configurar qualquer servidor WSGI que não seja Gunicorn, certifique-se de também especificá-lo como uma dependência do seu aplicativo, para que ele seja instalado em suas instâncias de ambiente. Para obter detalhes sobre a especificação de dependência, consulte [the section called “Especificação de dependências”](#).
- A porta padrão para o servidor WSGI é 8000. Se você especificar um número de porta diferente no comando Procfile, defina a [propriedade do ambiente](#) PORT para esse número de porta também.

Quando você usa um Procfile, ele substitui as opções de namespace `aws:elasticbeanstalk:container:python` definidas usando arquivos de configuração.

Para obter detalhes sobre o uso do Procfile, expanda a seção Buildfile e Procfile em [the section called “Estender plataformas Linux”](#).

Especificação de dependências usando um arquivo de requisitos

Um aplicativo em Python típico tem dependências em outros pacotes Python de terceiros. Com a plataforma Python do Elastic Beanstalk, há algumas maneiras de especificar pacotes Python dos quais depende a aplicação.

Utilizar **pip** e **requirements.txt**

A ferramenta padrão para instalar pacotes Python é `pip`. Ele tem um recurso que permite especificar todos os pacotes necessários (e suas versões) em um único arquivo de requisitos. Para obter mais informações sobre o arquivo de requisitos, consulte [Formato do arquivo de requisitos](#) no site de documentação do `pip`.

Crie um arquivo chamado `requirements.txt` e coloque-o no diretório de nível superior do pacote de origem. Veja a seguir um exemplo do arquivo `requirements.txt` para Django.

```
Django==2.2
mysqlclient==2.0.3
```

Em seu ambiente de desenvolvimento, use o comando `pip freeze` para gerar o arquivo de requisitos.

```
~/my-app$ pip freeze > requirements.txt
```

Para garantir que o arquivo de requisitos contenha apenas os pacotes realmente usados pelo aplicativo, use um [ambiente virtual](#) que tenha somente esses pacotes instalados. Fora de um ambiente virtual, a saída de `pip freeze` incluirá todos os pacotes do `pip` instalados na máquina de desenvolvimento, bem como aqueles que acompanham o sistema operacional.

Note

Nas versões da plataforma Python da AMI do Amazon Linux, o Elastic Beanstalk não é compatível com Pipenv nem Pipfiles. Se você usar Pipenv para gerenciar as dependências do aplicativo, execute o seguinte comando para gerar um arquivo `requirements.txt`.

```
~/my-app$ pipenv lock -r > requirements.txt
```

Para saber mais, consulte [Gerar um requirements.txt](#) na documentação do Pipenv.

Usar Pipenv e **Pipfile**

Pipenv é uma ferramenta moderna de empacotamento Python. Ele combina a instalação do pacote com a criação e gerenciamento de um arquivo de dependência e um virtualenv para o seu aplicativo. Para obter mais informações, consulte [Pipenv: Python Dev Workflow for Humans](#).

O Pipenv mantém dois arquivos:

- **Pipfile** — Esse arquivo contém vários tipos de dependências e requisitos.
- **Pipfile.lock** — Esse arquivo contém um instantâneo da versão que permite compilações determinísticas.

Você pode criar esses arquivos em seu ambiente de desenvolvimento e incluí-los no diretório de nível superior do pacote de origem implantado no Elastic Beanstalk. Para obter mais informações sobre esses dois arquivos, consulte [Exemplo de Pipfile e Pipfile.lock](#).

O exemplo a seguir usa o Pipenv para instalar o Django e a framework REST do Django. Esses comandos criam os arquivos **Pipfile** e **Pipfile.lock**.

```
~/my-app$ pipenv install django
~/my-app$ pipenv install djangorestframework
```

Precedência

Se você incluir mais de um dos arquivos de requisitos descritos neste tópico, o Elastic Beanstalk usará apenas um deles. A lista a seguir mostra a precedência, em ordem decrescente.

1. **requirements.txt**
2. **Pipfile.lock**
3. **Pipfile**

Note

A partir da versão da plataforma Amazon Linux 2 de 7 de março de 2023, se você fornecer mais de um desses arquivos, o Elastic Beanstalk emitirá uma mensagem do console informando qual dos arquivos de dependência foi usado durante uma implantação.

As etapas a seguir descrevem a lógica que o Elastic Beanstalk segue para instalar as dependências ao implantar uma instância.

- Se houver um arquivo `requirements.txt`, usamos o comando `pip install -r requirements.txt`.
- A partir do lançamento da plataforma Amazon Linux 2 em 7 de março de 2023, se não houver nenhum arquivo `requirements.txt`, mas houver um `Pipfile.lock`, usamos o comando `pipenv sync`. Antes desse lançamento, usamos `pipenv install --ignore-pipfile`.
- Se não houver um arquivo `requirements.txt` nem um `Pipfile.lock`, mas houver um `Pipfile`, usamos o comando `pipenv install --skip-lock`.
- Se nenhum dos três arquivos de requisitos for encontrado, não instalaremos nenhuma dependência do aplicativo.

Implantar uma aplicação Flask no Elastic Beanstalk

O Flask é uma framework de aplicativo web de código aberto para Python. Este tutorial mostra o processo de geração de um aplicativo Flask e sua implantação em um AWS Elastic Beanstalk ambiente.

Neste tutorial, você fará o seguinte:

- [Configurar um ambiente Python virtual com Flask](#)
- [Criar uma aplicação Flask](#)
- [Implantar o site com a CLI do EB](#)
- [Limpeza](#)

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em [Conceitos básicos do Elastic Beanstalk](#) para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt (\$) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command  
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows 10, você pode [instalar o Subsistema Windows para Linux](#) para obter uma versão do Ubuntu integrada com o Windows e o Bash.

O Flask requer o Python 3.7 ou posterior. Neste tutorial, usamos Python 3.7 e a versão correspondente da plataforma do Elastic Beanstalk. Instale o Python seguindo as instruções em [Configurar o ambiente de desenvolvimento Python](#).

O framework [Flask](#) será instalado como parte do tutorial.

Este tutorial também usa a interface da linha de comando do Elastic Beanstalk (CLI do EB). Para obter detalhes sobre a instalação e configuração da CLI do EB, consulte [Instalar a EB CLI](#) e [Configurar a EB CLI](#).

Configurar um ambiente Python virtual com Flask

Crie um diretório de projeto e um ambiente virtual para seu aplicativo e instale o Flask.

Para configurar o ambiente de projetos

1. Crie um diretório do projeto.

```
~$ mkdir eb-flask  
~$ cd eb-flask
```

2. Crie e ative um ambiente virtual denominado `virt`:

```
~/eb-flask$ virtualenv virt
```

```
~$ source virt/bin/activate  
(virt) ~/eb-flask$
```

Você verá `(virt)` acrescentado ao prompt de comando, indicando que você está em um ambiente virtual. Use o ambiente virtual para o restante deste tutorial.

3. Instale o Flask com `pip install`:

```
(virt)~/eb-flask$ pip install flask==2.0.3
```

4. Veja as bibliotecas instaladas com `pip freeze`:

```
(virt)~/eb-flask$ pip freeze  
click==8.1.1  
Flask==2.0.3  
itsdangerous==2.1.2  
Jinja2==3.1.1  
MarkupSafe==2.1.1  
Werkzeug==2.1.0
```

Esse comando relaciona todos os pacotes instalados em seu ambiente virtual. Como você está em um ambiente virtual, os pacotes instalados globalmente, como a EB CLI, não são mostrados.

5. Salve a saída de `pip freeze` como um arquivo chamado `requirements.txt`.

```
(virt)~/eb-flask$ pip freeze > requirements.txt
```

Esse arquivo informa o Elastic Beanstalk para instalar as bibliotecas durante a implantação. Para ter mais informações, consulte [Especificação de dependências usando um arquivo de requisitos](#).

Criar uma aplicação Flask

Depois, crie uma aplicação a ser implantada usando o Elastic Beanstalk. Vamos criar um serviço Web RESTful "Hello World".

Crie um novo arquivo de texto nesse diretório chamado `application.py` com o seguinte conteúdo:

Example `~/eb-flask/application.py`

```
from flask import Flask
```

```
# print a nice greeting.
def say_hello(username = "World"):
    return '<p>Hello %s!</p>\n' % username

# some bits of text for the page.
header_text = '''
    <html>\n<head> <title>EB Flask Test</title> </head>\n<body>'''
instructions = '''
    <p><em>Hint</em>: This is a RESTful web service! Append a username
    to the URL (for example: <code>/Thelonious</code>) to say hello to
    someone specific.</p>\n'''
home_link = '<p><a href="/">Back</a></p>\n'
footer_text = '</body>\n</html>'

# EB looks for an 'application' callable by default.
application = Flask(__name__)

# add a rule for the index page.
application.add_url_rule('/', 'index', (lambda: header_text +
    say_hello() + instructions + footer_text))

# add a rule when the page is accessed with a name appended to the site
# URL.
application.add_url_rule('/<username>', 'hello', (lambda username:
    header_text + say_hello(username) + home_link + footer_text))

# run the app.
if __name__ == "__main__":
    # Setting debug to True enables debug output. This line should be
    # removed before deploying a production app.
    application.debug = True
    application.run()
```

Este exemplo imprime uma saudação personalizada que varia de acordo com o caminho usado para acessar o serviço.

Note

Com a adição de `application.debug = True` antes de executar o aplicativo, a saída da depuração é habilitada para o caso de algo der errado. É uma boa prática de

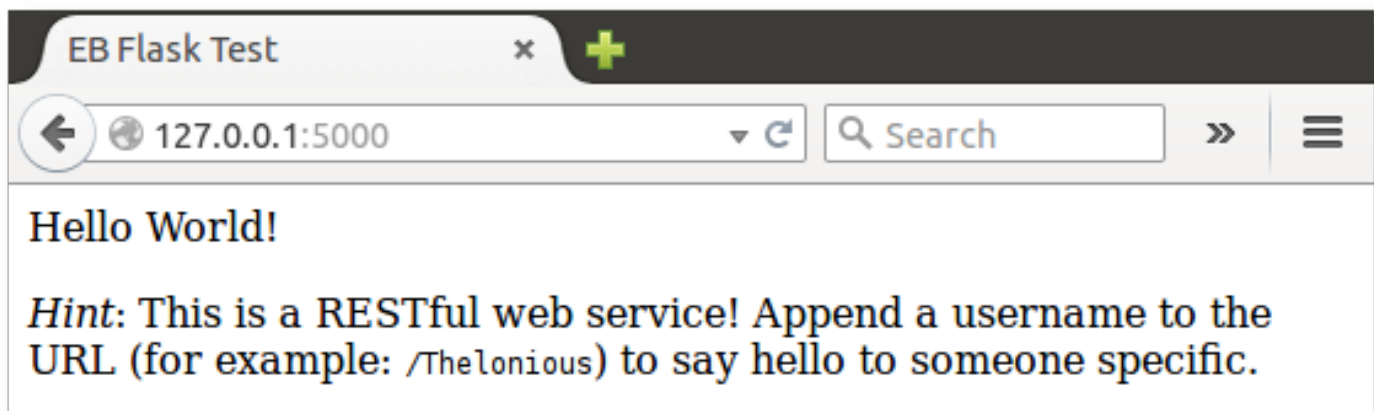
desenvolvimento, mas você deve remover instruções de depuração no código de produção, já que a saída da depuração pode revelar aspectos internos do seu aplicativo.

Use `application.py` como o nome do arquivo e forneça um objeto que pode ser chamado `application` (o objeto Flask, neste caso) para que o Elastic Beanstalk possa localizar facilmente o código da aplicação.

Execute `application.py` com Python:

```
(virt) ~/eb-flask$ python application.py
* Serving Flask app "application" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 313-155-123
```

Abra `http://127.0.0.1:5000/` em seu navegador da web. Você deve ver o aplicativo em execução, mostrando a página de índice:



Verifique o log do servidor para ver a saída da sua solicitação. Você pode interromper o servidor web e retornar ao ambiente virtual pressionando `Ctrl+C`.

Se você obteve a saída da depuração em vez disso, corrija os erros e verifique se a aplicação é executada localmente antes de configurá-la para o Elastic Beanstalk.

Implantar o site com a CLI do EB

Você adicionou tudo o que precisa para implantar a aplicação no Elastic Beanstalk. O diretório do projeto deve ter esta aparência:

```
~/eb-flask/  
|-- virt  
|-- application.py  
`-- requirements.txt
```

No entanto, a pasta `virt` não é necessária para que a aplicação seja executada no Elastic Beanstalk. Quando você faz a implantação, o Elastic Beanstalk cria um ambiente virtual nas instâncias do servidor e instala as bibliotecas listadas em `requirements.txt`. Para minimizar o tamanho do pacote de origem que você carregou durante a implantação, adicione um arquivo [.ebignore](#) que instrui a EB CLI a deixar a pasta `virt` de fora.

Example `~/eb-flask/.ebignore`

```
virt
```

Depois, você criará o ambiente da aplicação e implantará a aplicação configurada com o Elastic Beanstalk.

Para criar um ambiente e implantar o aplicativo Flask

1. Inicialize o repositório da EB CLI com o comando `eb init`:

```
~/eb-flask$ eb init -p python-3.7 flask-tutorial --region us-east-2  
Application flask-tutorial has been created.
```

Esse comando cria uma nova aplicação nomeada `flask-tutorial` e configura seu repositório local para criar ambientes com a versão da plataforma Python 3.7 mais recente.

2. (opcional) Execute `eb init` novamente para configurar um par de chaves padrão para que você possa se conectar à instância do EC2 que executa seu aplicativo com SSH:

```
~/eb-flask$ eb init  
Do you want to set up SSH for your instances?  
(y/n): y  
Select a keypair.
```

- ```
1) my-keypair
2) [Create new KeyPair]
```

Selecione um par de chaves se você já tiver um ou siga os prompts para criar um novo. Se nenhum prompt aparecer ou se for necessário alterar as configurações posteriormente, execute `eb init -i`.

3. Crie um ambiente e implante seu aplicativo nele com `eb create`:

```
~/eb-flask$ eb create flask-env
```

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.

- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

#### Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

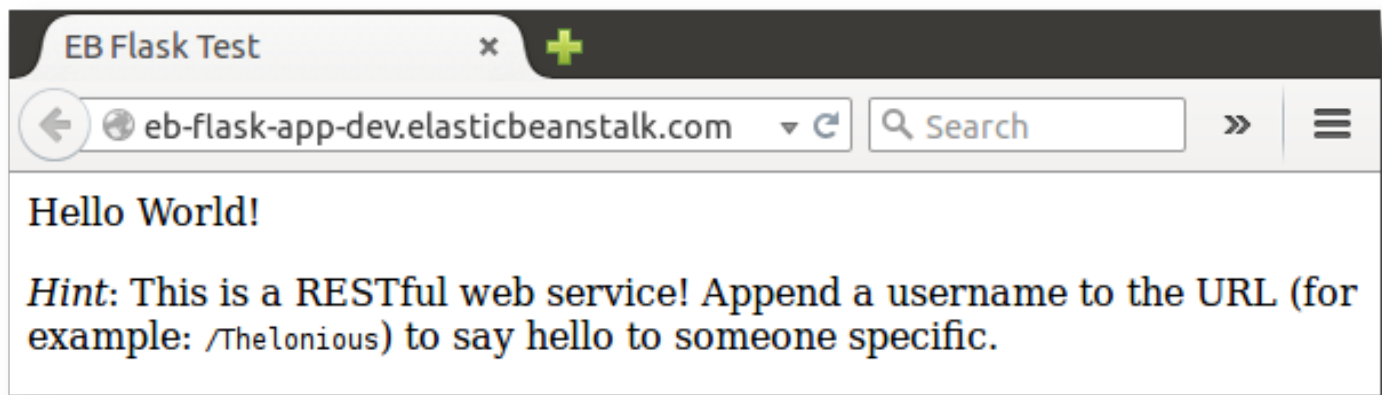
#### Note

O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para ter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon S3](#).

Quando o processo de criação do ambiente for concluído, abra o site com `eb open`:

```
~/eb-flask$ eb open
```

Isso abrirá uma janela do navegador usando o nome de domínio criado para o seu aplicativo. Você deve ver o mesmo site Flask que criou e testou localmente.



Se o aplicativo não aparecer em execução ou se houver uma mensagem de erro, consulte [Solução de erros de implantação](#) para obter ajuda sobre como determinar a causa do erro.

Caso veja a aplicação em execução, você implantou sua primeira aplicação Flask com o Elastic Beanstalk. Parabéns!

## Limpeza

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. [O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente, como instâncias do Amazon EC2, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.](#)

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

ou com a EB CLI:



```
~/eb-flask$ eb terminate flask-env
```

## Próximas etapas

Para mais informações sobre o Flask, acesse [flask.pocoo.org](http://flask.pocoo.org).

Se você deseja testar outra framework da Web para Python, consulte [Implantar uma aplicação Django no Elastic Beanstalk](#).

## Implantar uma aplicação Django no Elastic Beanstalk

Este tutorial explica o processo de implantação de um site [Django](#) padrão gerado automaticamente para ambientes do AWS Elastic Beanstalk que executam Python. Este tutorial mostra como hospedar uma aplicação Web Python na nuvem usando um ambiente do Elastic Beanstalk.

Neste tutorial, você fará o seguinte:

- [Configurar um ambiente virtual Python e instalar o Django](#)
- [Criar um projeto Django](#)
- [Configurar a aplicação Django para o Elastic Beanstalk](#)
- [Implantar o site com a CLI do EB](#)
- [Atualizar seu aplicativo](#)
- [Limpar](#)

## Pré-requisitos

Para usar qualquer serviço da AWS, incluindo o Elastic Beanstalk, você precisa ter uma conta e credenciais da AWS. Para saber mais e se cadastrar, visite <https://aws.amazon.com/>.

Para seguir este tutorial, você deve ter todos os [Pré-requisitos comuns](#) para Python instalados, incluindo os seguintes pacotes:

- Python 3.7 ou posterior
- pip
- virtualenv
- awsebcli

A framework do [Django](#) será instalada como parte do tutorial.

### Note

A criação de ambientes com a CLI do EB exige uma [função de serviço](#). É possível criar uma função de serviço criando um ambiente no console do Elastic Beanstalk. Se você não tiver uma função de serviço, a CLI do EB tentará criar uma ao executar `eb create`.

## Configurar um ambiente virtual Python e instalar o Django

Crie um ambiente virtual com `virtualenv` e use-o para instalar o Django e suas dependências. Usando um ambiente virtual, é possível saber exatamente de quais pacotes a aplicação precisa, para que eles sejam instalados nas instâncias do Amazon EC2 que executam a aplicação.

As etapas a seguir demonstram os comandos que você deve inserir para sistemas baseados em Unix e Windows, mostrados em guias separadas.

Para configurar o ambiente virtual

1. Crie um ambiente virtual denominado `eb-virt`.

Unix-based systems

```
~$ virtualenv ~/eb-virt
```

Windows

```
C:\> virtualenv %HOMEPATH%\eb-virt
```

2. Ative o ambiente virtual.

Unix-based systems

```
~$ source ~/eb-virt/bin/activate
(eb-virt) ~$
```

Windows

```
C:\>%HOMEPATH%\eb-virt\Scripts\activate
```

```
(eb-virt) C:\>
```

Você verá `(eb-virt)` acrescentado ao começo do prompt de comando, indicando que você está em um ambiente virtual.

#### Note

O restante das instruções mostra o prompt de comando do Linux no diretório inicial `~$`. No Windows, ele é `C:\Users\USERNAME>`, em que *USERNAME* é seu nome de login do Windows.

3. Use `pip` para instalar o Django.

```
(eb-virt)~$ pip install django==2.2
```

#### Note

A versão do Django que você instalar deve ser compatível com a versão do Python na configuração do Python do Elastic Beanstalk que você escolher para a implantação de seu aplicativo. Para obter informações sobre implantação, consulte [???](#) neste tópico. Para obter mais informações sobre as versões atuais da plataforma Python, consulte [Python](#) no documento Plataformas do AWS Elastic Beanstalk. Para obter informações sobre a compatibilidade de versão do Django com o Python, consulte [Qual versão do Python posso usar com o Django?](#)

4. Para verificar se o Django está instalado, insira o seguinte.

```
(eb-virt)~$ pip freeze
Django==2.2
...
```

Esse comando relaciona todos os pacotes instalados em seu ambiente virtual. Posteriormente, você usará a saída desse comando para configurar o projeto para uso com o Elastic Beanstalk.

## Criar um projeto Django

Agora você está pronto para criar um projeto Django e executá-lo em sua máquina, usando o ambiente virtual.

### Note

Este tutorial usa SQLite, que é um mecanismo de banco de dados simples incluído no Python. O banco de dados é implantado com seus arquivos de projeto. Para ambientes de produção, recomendamos que você use o Amazon Relational Database Service (Amazon RDS) e que você o separe do ambiente. Para obter mais informações, consulte . [Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicações Python.](#)

Para gerar um aplicativo Django

1. Ative seu ambiente virtual.

Unix-based systems

```
~$ source ~/eb-virt/bin/activate
(eb-virt) ~$
```

Windows

```
C:\>%HOMEPATH%\eb-virt\Scripts\activate
(eb-virt) C:\>
```

Você verá o prefixo (eb-virt) acrescentado ao prompt de comando, indicando que você está em um ambiente virtual.

### Note

O restante das instruções mostra o prompt de comando do Linux ~\$ em seu diretório inicial e no diretório inicial do Linux ~/. No Windows, eles são C:\Users\*USERNAME*>, em que *USERNAME* é seu nome de login do Windows.

2. Use o comando `django-admin startproject ebdjango` para criar um projeto Django denominado `ebdjango`:

```
(eb-virt)~$ django-admin startproject ebdjango
```

Este comando cria um site Django padrão denominado `ebdjango` com a seguinte estrutura de diretório.

```
~/ebdjango
|-- ebdjango
| |-- __init__.py
| |-- settings.py
| |-- urls.py
| `-- wsgi.py
`-- manage.py
```

3. Execute o site Django localmente com `manage.py runserver`.

```
(eb-virt) ~$ cd ebdjango
```

```
(eb-virt) ~/ebdjango$ python manage.py runserver
```

4. Em um navegador da web, abra `http://127.0.0.1:8000/` para visualizar o site.
5. Verifique o log do servidor para ver a saída da sua solicitação. Para interromper o servidor web e retornar ao seu ambiente virtual, pressione `Ctrl+C`.

```
Django version 2.2, using settings 'ebdjango.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[07/Sep/2018 20:14:09] "GET / HTTP/1.1" 200 16348
Ctrl+C
```

## Configurar a aplicação Django para o Elastic Beanstalk

Agora que você tem um site desenvolvido pelo Django em sua máquina local, você pode configurá-lo para implantação com o Elastic Beanstalk.

Por padrão, o Elastic Beanstalk procura um arquivo chamado `application.py` para iniciar sua aplicação. Como isso não existe no projeto Django que você criou, é necessário fazer alguns ajustes

no ambiente do aplicativo. Também é necessário definir variáveis de ambiente para que os módulos do aplicativo possam ser carregados.

## Como configurar seu site para o Elastic Beanstalk

1. Ative seu ambiente virtual.

### Unix-based systems

```
~/ebdjango$ source ~/eb-virt/bin/activate
```

### Windows

```
C:\Users\USERNAME\ebdjango>%HOMEPATH%\eb-virt\Scripts\activate
```

2. Execute `pip freeze` e salve a saída em um arquivo chamado `requirements.txt`.

```
(eb-virt) ~/ebdjango$ pip freeze > requirements.txt
```

O Elastic Beanstalk usa o `requirements.txt` para determinar que pacote instalar nas instâncias do EC2 que executam a aplicação.

3. Crie um diretório chamado `.ebextensions`.

```
(eb-virt) ~/ebdjango$ mkdir .ebextensions
```

4. No diretório `.ebextensions`, adicione um [arquivo de configuração](#) chamado `django.config` com o texto a seguir.

### Example `~/ebdjango/.ebextensions/django.config`

```
option_settings:
 aws:elasticbeanstalk:container:python:
 WSGIPath: ebdjango.wsgi:application
```

Essa configuração, `WSGIPath`, especifica o local do script WSGI que o Elastic Beanstalk usa para iniciar a aplicação.

**Note**

Se você estiver usando uma versão de plataforma Python da AMI do Amazon Linux (anterior ao Amazon Linux 2), substitua o valor `WSGIPath` por `ebdjango/wsgi.py`. O valor no exemplo funciona com o servidor WSGI do Gunicorn, que não é mais compatível nas versões da plataforma da AMI do Amazon Linux.

5. Use o comando `deactivate` para desativar o ambiente virtual.

```
(eb-virt) ~/ebdjango$ deactivate
```

Reative o ambiente virtual sempre que for necessário adicionar pacotes ao aplicativo ou executá-lo localmente.

## Implantar o site com a CLI do EB

Você adicionou tudo o que precisa para implantar a aplicação no Elastic Beanstalk. O diretório do projeto deve ter esta aparência:


```
~/ebdjango/
|-- .ebextensions
| |-- django.config
|-- ebdjango
| |-- __init__.py
| |-- settings.py
| |-- urls.py
| |-- wsgi.py
|-- db.sqlite3
|-- manage.py
|-- requirements.txt
```

Depois, você criará o ambiente da aplicação e implantará a aplicação configurada com o Elastic Beanstalk.

Imediatamente após a implantação, edite a configuração do Django para adicionar o nome de domínio que o Elastic Beanstalk atribuiu à aplicação para os do Django `ALLOWED_HOSTS`. Depois, reimplante o aplicativo. Este é um requisito de segurança do Django, projetado para impedir ataques

ao cabeçalho de Host HTTP. Para obter mais informações, consulte [Validação do cabeçalho do host](#).

Para criar um ambiente e implantar o aplicativo Django

 Note

O tutorial usa a CLI do EB como um mecanismo de implantação, mas também é possível usar o console do Elastic Beanstalk para implantar um arquivo zip com o conteúdo do seu projeto.

1. Inicialize o repositório da EB CLI com o comando `eb init`:

```
~/ebdjango$ eb init -p python-3.7 django-tutorial
Application django-tutorial has been created.
```

Esse comando cria um aplicativo chamado `django-tutorial`. Ele também configura o seu repositório local para criar ambientes com a versão mais recente da plataforma Python 3.7.

2. (Opcional) Execute `eb init` novamente para configurar um par de chaves padrão para que você possa usar SSH para se conectar à instância do EC2 que está executando seu aplicativo:

```
~/ebdjango$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [Create new KeyPair]
```

Selecione um par de chaves se você já tiver um, ou siga os as instruções na tela para criar um. Se nenhum prompt aparecer ou se for necessário alterar as configurações posteriormente, execute `eb init -i`.

3. Crie um ambiente e implante o aplicativo nele com `eb create`.

```
~/ebdjango$ eb create django-env
```



**Note**

Se você vir uma mensagem de erro "service role required", execute `eb create` interativamente (sem especificar um nome de ambiente) e a CLI do EB cria a função para você.

Esse comando cria um ambiente do Elastic Beanstalk com carga balanceada chamado `django-env`. A criação do ambiente leva cerca de 5 minutos. Como o Elastic Beanstalk cria os recursos necessários para executar a aplicação, ele gera mensagens informativas que a CLI do EB transmite ao terminal.

- Quando o processo de criação do ambiente for concluído, localize o nome de domínio do seu novo ambiente executando `eb status`.

```
~/ebdjango$ eb status
Environment details for: django-env
 Application name: django-tutorial
 ...
 CNAME: eb-django-app-dev.elasticbeanstalk.com
 ...
```

Seu nome de domínio do ambiente é o valor da propriedade CNAME.

- Abra o arquivo `settings.py` no diretório `ebdjango`. Localize a configuração `ALLOWED_HOSTS` e adicione o nome de domínio do aplicativo que você encontrou na etapa anterior ao valor da configuração. Se você não encontrar essa configuração no arquivo, adicione-a em uma nova linha.

```
...
ALLOWED_HOSTS = ['eb-django-app-dev.elasticbeanstalk.com']
```

- Salve o arquivo e, em seguida, implante o aplicativo executando `eb deploy`. Quando você executa `eb deploy`, a EB CLI empacota o conteúdo do diretório do projeto e implanta-o em seu ambiente.

```
~/ebdjango$ eb deploy
```

**Note**

Se você estiver usando o Git com seu projeto, consulte [Usar a EB CLI com o Git](#).

7. Quando o processo de atualização do ambiente for concluído, abra o site com `eb open`.

```
~/ebdjango$ eb open
```

Isso abre uma janela do navegador usando o nome de domínio criado para o seu aplicativo. Você deve ver o mesmo site Django que você criou e testou localmente.

Se o aplicativo não aparecer em execução ou se houver uma mensagem de erro, consulte [Solução de erros de implantação](#) para obter ajuda sobre como determinar a causa do erro.

Se a aplicação aparecer em execução, você implantou sua primeira aplicação Django com o Elastic Beanstalk. Parabéns!

## Atualizar seu aplicativo

Agora que você tem uma aplicação em execução no Elastic Beanstalk, pode atualizar e reimplantar a aplicação ou sua configuração, e o Elastic Beanstalk fará o trabalho de atualizar suas instâncias e iniciar sua nova versão da aplicação.

Para este exemplo, vamos habilitar o console de administração do Django e definir algumas outras configurações.

### Modificar as configurações do site

Por padrão, o site Django usa o fuso horário UTC para exibir a hora. Você pode alterá-lo especificando um fuso horário em `settings.py`.

### Para alterar o fuso horário do site

1. Modifique a configuração `TIME_ZONE` em `settings.py`.

Example `~/ebdjango/ebdjango/settings.py`

```
...
Internationalization
LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'US/Pacific'
USE_I18N = True
USE_L10N = True
USE_TZ = True
```

Para obter uma lista de fusos horários, visite [esta página](#).

2. Implante a aplicação no ambiente do Elastic Beanstalk.

```
~/ebdjango/$ eb deploy
```

### Criar um administrador de site

É possível criar um administrador de site para o aplicativo Django para acessar o console de administração diretamente do site. Os detalhes de login do administrador são armazenados de forma segura na imagem do banco de dados local incluída no projeto padrão gerado pelo Django.

Para criar um administrador de site

1. Inicialize o banco de dados local do aplicativo Django.

```
(eb-virt) ~/ebdjango$ python manage.py migrate
Operations to perform:
 Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
 Applying contenttypes.0001_initial... OK
 Applying auth.0001_initial... OK
 Applying admin.0001_initial... OK
 Applying admin.0002_logentry_remove_auto_add... OK
 Applying admin.0003_logentry_add_action_flag_choices... OK
 Applying contenttypes.0002_remove_content_type_name... OK
 Applying auth.0002_alter_permission_name_max_length... OK
 Applying auth.0003_alter_user_email_max_length... OK
 Applying auth.0004_alter_user_username_opts... OK
 Applying auth.0005_alter_user_last_login_null... OK
 Applying auth.0006_require_contenttypes_0002... OK
 Applying auth.0007_alter_validators_add_error_messages... OK
 Applying auth.0008_alter_user_username_max_length... OK
 Applying auth.0009_alter_user_last_name_max_length... OK
 Applying sessions.0001_initial... OK
```

2. Execute `manage.py createsuperuser` para criar um administrador.

```
(eb-virt) ~/ebdjango$ python manage.py createsuperuser
Username: admin
Email address: me@mydomain.com
Password: *****
Password (again): *****
Superuser created successfully.
```

3. Para informar ao Django onde armazenar os arquivos estáticos, defina `STATIC_ROOT` em `settings.py`.

Example `~/ebdjango/ebdjango/settings.py`

```
Static files (CSS, JavaScript, Images)
https://docs.djangoproject.com/en/2.2/howto/static-files/
STATIC_URL = '/static/'
STATIC_ROOT = 'static'
```

4. Execute `manage.py collectstatic` para preencher o diretório `static` com os ativos estáticos (JavaScript, CSS e imagens) para o site de administração.

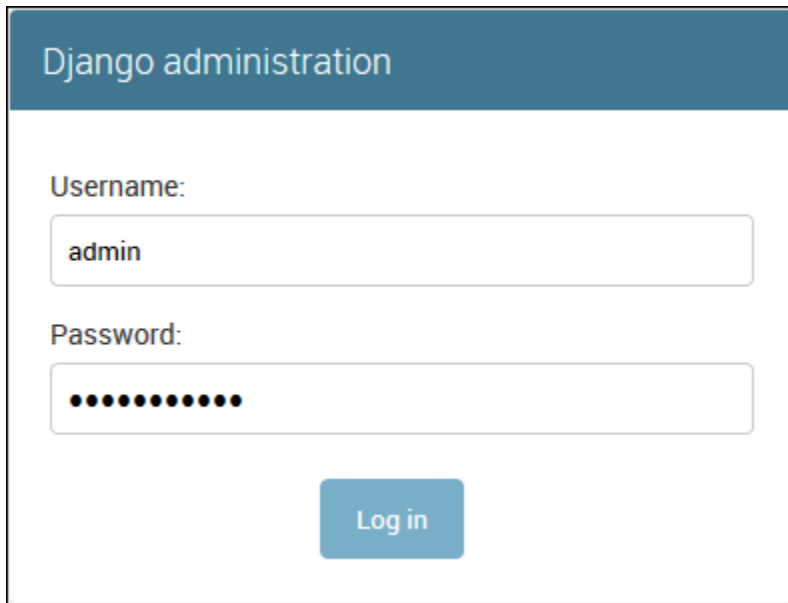
```
(eb-virt) ~/ebdjango$ python manage.py collectstatic
119 static files copied to ~/ebdjango/static
```

5. Implante o aplicativo.

```
~/ebdjango$ eb deploy
```

6. Exiba o console de administração abrindo o site em seu navegador, anexando `/admin/` ao URL do site, como o seguinte:

```
http://django-env.p33kq46sfh.us-west-2.elasticbeanstalk.com/admin/
```



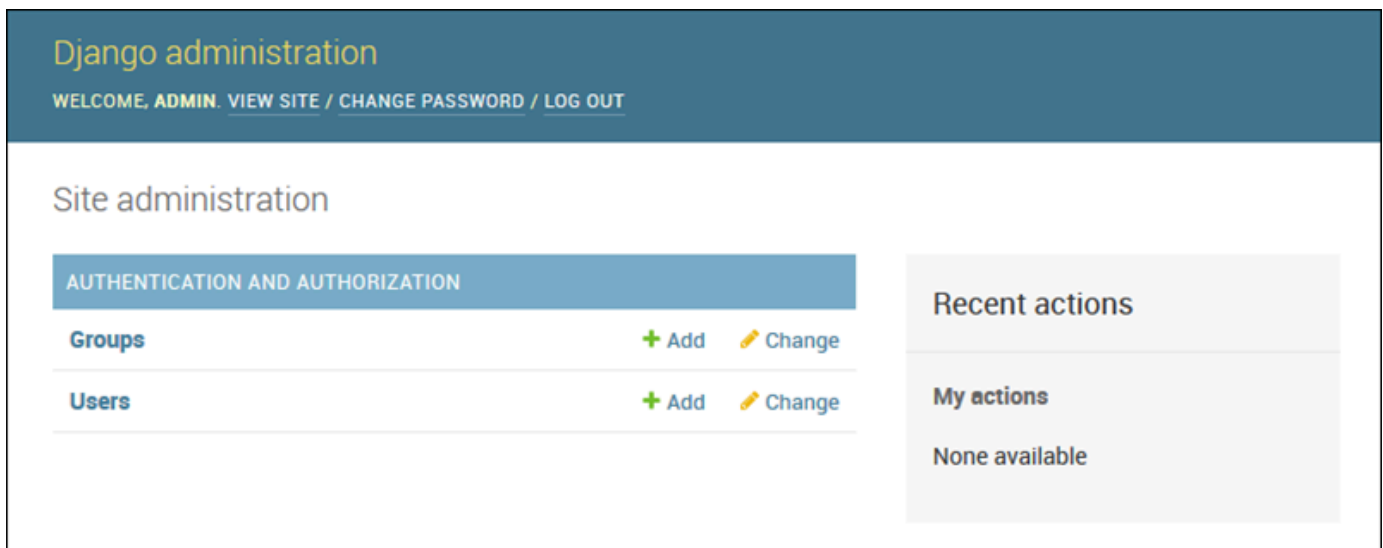
Django administration

Username:  
admin

Password:  
●●●●●●●●

Log in

7. Faça login com o nome de usuário e a senha que você configurou na etapa 2.



Django administration

WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Site administration

AUTHENTICATION AND AUTHORIZATION

|        |                       |                        |
|--------|-----------------------|------------------------|
| Groups | <a href="#">+ Add</a> | <a href="#">Change</a> |
| Users  | <a href="#">+ Add</a> | <a href="#">Change</a> |

Recent actions

My actions

None available

Você pode usar um procedimento semelhante de atualização/teste local seguido por eb deploy. O Elastic Beanstalk cuida do processo de atualizar seus servidores em tempo real para que você possa se dedicar ao desenvolvimento de aplicações, em vez da administração de servidor.

Adicionar um arquivo de configuração de migração de banco de dados

É possível adicionar comandos ao script `.ebextensions` que serão executados quando o site for atualizado. Isso permite gerar automaticamente migrações de banco de dados.

Para adicionar uma etapa de migração quando o aplicativo é implantado

1. Crie um [arquivo de configuração](#) chamado `db-migrate.config` com o conteúdo a seguir.

Example `~/ebdjango/.ebextensions/db-migrate.config`

```
container_commands:
 01_migrate:
 command: "source /var/app/venv/*/bin/activate && python3 manage.py migrate"
 leader_only: true
option_settings:
 aws:elasticbeanstalk:application:environment:
 DJANGO_SETTINGS_MODULE: ebdjango.settings
```

Esse arquivo de configuração ativa o ambiente virtual do servidor e executa o comando `manage.py migrate` durante o processo de implantação, antes de iniciar a sua aplicação. Como ele é executado antes da inicialização do aplicativo, também é necessário configurar a variável de ambiente `DJANGO_SETTINGS_MODULE` explicitamente (em geral, o `wsgi.py` cuida desse procedimento para você durante o startup). A especificação de `leader_only: true` no comando garante que ele seja executado somente uma vez quando você faz a implantação em várias instâncias.

2. Implante o aplicativo.

```
~/ebdjango$ eb deploy
```

## Limpar

Para economizar horas de instância e outros recursos da AWS entre as sessões de desenvolvimento, termine o ambiente do Elastic Beanstalk com `eb terminate`.

```
~/ebdjango$ eb terminate django-env
```

Esse comando encerra o ambiente e todos os recursos da AWS que são executados nele. Ele não exclui o aplicativo, mas você sempre pode criar mais ambientes com a mesma configuração executando `eb create` novamente. Para obter mais informações sobre comandos da CLI do EB, consulte [Gerenciar ambientes do Elastic Beanstalk com a CLI do EB](#).

Se você já concluiu o aplicativo de exemplo, também pode remover a pasta do projeto e o ambiente virtual.

```
~$ rm -rf ~/eb-virt
~$ rm -rf ~/ebdjango
```

## Próximas etapas

Para obter mais informações sobre o Django, incluindo um tutorial detalhado, consulte [a documentação oficial](#).

Se você deseja testar outra framework da Web para Python, consulte [Implantar uma aplicação Flask no Elastic Beanstalk](#).

## Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicações Python

É possível usar uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS) para armazenar dados coletados e modificados pela aplicação. O banco de dados pode ser acoplado ao seu ambiente e gerenciado pelo Elastic Beanstalk ou pode ser criado como desacoplado e gerenciado externamente por outro serviço. Este tópico fornece instruções para criar o Amazon RDS usando o console do Elastic Beanstalk. O banco de dados será acoplado ao ambiente e gerenciado pelo Elastic Beanstalk. Para obter mais informações sobre como integrar um Amazon RDS com o Elastic Beanstalk, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

### Seções


- [Adicionar uma instância de banco de dados ao seu ambiente](#)
- [Download de um driver](#)
- [Conectar-se a um banco de dados](#)

## Adicionar uma instância de banco de dados ao seu ambiente

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.

- No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

- No painel de navegação, escolha Configuration (Configuração).
- Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
- Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.
- Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização do ambiente for concluída, o nome de host da instância de banco de dados e outras informações de conexão estarão disponíveis para o seu aplicativo por meio das seguintes propriedades de ambiente:

| Nome da propriedade | Descrição                                                                                                                  | Valor da propriedade                                                                                |
|---------------------|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| RDS_HOSTNAME        | O nome do host da instância de banco de dados.                                                                             | Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.     |
| RDS_PORT            | A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados. | Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta). |
| RDS_DB_NAME         | O nome do banco de dado, <b>ebdb</b> .                                                                                     | Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).    |
| RDS_USERNAME        | O nome de usuário que você configurou para seu banco de dados.                                                             | Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.            |



| Nome da propriedade | Descrição                                            | Valor da propriedade                                     |
|---------------------|------------------------------------------------------|----------------------------------------------------------|
| RDS_PASSWORD        | A senha que você configurou para seu banco de dados. | Não disponível para referência no console do Amazon RDS. |

Para obter mais informações sobre como configurar uma instância de banco de dados acoplada a um ambiente Elastic Beanstalk, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

## Download de um driver

Adicione o driver de banco de dados ao [arquivo de requisitos](#) de seu projeto.

Example requirements.txt: Django com MySQL

```
Django==2.2
mysqlclient==2.0.3
```

Pacotes de drivers comuns para Python

- MySQL – `mysqlclient`
- PostgreSQL – `psycopg2`
- Oracle: `cx_Oracle`
- SQL Server – `adodbapi`

Para obter mais informações, consulte [Python DatabaseInterfaces](#) e [Django 2.2 - bancos de dados suportados](#).

## Conectar-se a um banco de dados

O Elastic Beanstalk fornece informações de conexão referentes às instâncias de banco de dados associadas nas propriedades de ambiente. Use os `os.environ['VARIABLE']` para ler as propriedades e configurar uma conexão de banco de dados.

Example Arquivo de configurações Django: dicionário de BANCOS DE DADOS

```
import os
```

```
if 'RDS_HOSTNAME' in os.environ:
 DATABASES = {
 'default': {
 'ENGINE': 'django.db.backends.mysql',
 'NAME': os.environ['RDS_DB_NAME'],
 'USER': os.environ['RDS_USERNAME'],
 'PASSWORD': os.environ['RDS_PASSWORD'],
 'HOST': os.environ['RDS_HOSTNAME'],
 'PORT': os.environ['RDS_PORT'],
 }
 }
```

## Ferramentas e recursos do Python

Há vários lugares que você pode acessar para obter ajuda adicional durante o desenvolvimento de seus aplicativos Python:

| Recurso                                            | Descrição                                                                                     |
|----------------------------------------------------|-----------------------------------------------------------------------------------------------|
| <a href="#">Boto (o AWS SDK for Python)</a>        | Instale o Boto usando o GitHub.                                                               |
| <a href="#">Fórum de desenvolvimento do Python</a> | Publique suas perguntas e receba comentários.                                                 |
| <a href="#">Centro de desenvolvedores Python</a>   | Um local central para acessar código de exemplo, documentação, ferramentas e outros recursos. |

## Criar e implantar aplicações Ruby no Elastic Beanstalk

Com o AWS Elastic Beanstalk para Ruby, é fácil implantar, gerenciar e escalar suas aplicações web Ruby usando a Amazon Web Services. O Elastic Beanstalk está disponível para qualquer pessoa que esteja desenvolvendo ou hospedando uma aplicação Web usando Ruby. Esta seção fornece instruções passo a passo para a implantação de uma aplicação demonstrativa no Elastic Beanstalk usando a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB) e atualizando a aplicação para usar as frameworks de trabalho de aplicação Web [Rails](#) e [Sinatra](#).

Os tópicos neste capítulo pressupõem que você tem um nível básico de conhecimento sobre ambientes do Elastic Beanstalk. Se você não tiver usado o Elastic Beanstalk antes, tente o [tutorial de conceitos básicos](#) para saber os princípios.

## Tópicos

- [Configurar ambiente de desenvolvimento Ruby](#)
- [Usar a plataforma Ruby do Elastic Beanstalk](#)
- [Implantar uma aplicação rails no Elastic Beanstalk](#)
- [Implantar uma aplicação sinatra no Elastic Beanstalk](#)
- [Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicações Ruby](#)

## Configurar ambiente de desenvolvimento Ruby

Configure um ambiente de desenvolvimento Ruby para testar a aplicação localmente antes de implantá-lo no AWS Elastic Beanstalk. Este tópico descreve as etapas de configuração do ambiente de desenvolvimento e inclui links para páginas de instalação de ferramentas úteis.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt (\$) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows 10, você pode [instalar o Subsistema Windows para Linux](#) para obter uma versão do Ubuntu integrada com o Windows e o Bash.

Para ver as etapas de configuração comuns e as ferramentas que se aplicam a todas as linguagens, consulte [Configurar sua máquina de desenvolvimento para uso com o Elastic Beanstalk](#)

## Seções

- [Instalar o Ruby](#)
- [Instalação do AWS SDK for Ruby](#)
- [Instalar um editor de texto ou IDE](#)

## Instalar o Ruby

Instale o GCC se você não tiver um compilador C. No Ubuntu, use apt.

```
~$ sudo apt install gcc
```

No Amazon Linux, use yum.

```
~$ sudo yum install gcc
```

Instale o RVM para gerenciar as instalações da linguagem Ruby em sua máquina. Use os comandos no [rvm.io](https://rvm.io) para obter as chaves do projeto e executar o script de instalação.

```
~$ gpg2 --recv-keys key1 key2
~$ curl -sSL https://get.rvm.io | bash -s stable
```

Esse script instala o RVM em uma pasta chamada `.rvm` no diretório do usuário e modifica seu perfil de shell para carregar um script de configuração sempre que você abrir um novo terminal. Carregue o script manualmente para começar.

```
~$ source ~/.rvm/scripts/rvm
```

Use `rvm get head` para obter a versão mais recente.

```
~$ rvm get head
```

Visualize as versões disponíveis do Ruby.

```
~$ rvm list known
MRI Rubies
...
[ruby-]2.6[.8]
[ruby-]2.7[.4]
[ruby-]3[.0.2]
...
```

Confira o [Ruby](#) no documento Plataformas da AWS Elastic Beanstalk para encontrar a versão mais recente do Ruby disponível em uma plataforma Elastic Beanstalk. Instale essa versão.

```
~$ rvm install 3.0.2
Searching for binary rubies, this might take some time.
Found remote file https://rubies.travis-ci.org/ubuntu/20.04/x86_64/ruby-3.0.2.tar.bz2
Checking requirements for ubuntu.
Updating system..
```

```
...
Requirements installation successful.
ruby-3.0.2 - #configure
ruby-3.0.2 - #download
...
```

Teste a instalação do Ruby.

```
~$ ruby --version
ruby 3.0.2p107 (2021-07-07 revision 0db68f0233) [x86_64-linux]
```

## Instalação do AWS SDK for Ruby

Se você precisa gerenciar os recursos da AWS a partir de sua aplicação, instale o AWS SDK for Ruby. Por exemplo, com o SDK for Ruby, é possível usar o Amazon DynamoDB (DynamoDB) para armazenar informações de usuário e sessão sem criar um banco de dados relacional.

Instale o SDK for Ruby e suas dependências com o comando `gem`.

```
$ gem install aws-sdk
```

Acesse a [página inicial do AWS SDK for Ruby](#) para obter mais informações e instruções de instalação.

## Instalar um editor de texto ou IDE

Os Integrated development environments (IDEs – Ambientes de desenvolvimento integrados) oferecem uma ampla variedade de recursos que facilitam o desenvolvimento de aplicativos. Se você não tiver usado um IDE para desenvolvimento Ruby, tente o Aptana e o RubyMine e veja qual funciona melhor para você.

- [Instalar o Aptana](#)
- [RubyMine](#)

### Note

Um IDE pode adicionar arquivos à pasta do seu projeto, que talvez você não queira confirmar no controle de código-fonte. Para evitar a confirmação desses arquivos no controle de código-fonte, use `.gitignore` ou a ferramenta de controle de código-fonte equivalente.

Se você deseja apenas começar a codificação e não precisa de todos os recursos de um IDE, considere [instalar o Sublime Text](#).

## Usar a plataforma Ruby do Elastic Beanstalk

A plataforma Ruby do AWS Elastic Beanstalk é um conjunto de [configurações de ambientes](#) para aplicações Web do Ruby que podem ser executadas por trás de um servidor de proxy NGINX sob um servidor de aplicações Puma. Cada ramificação da plataforma corresponde a uma versão do Ruby. Se você usa o RubyGems, pode [incluir um arquivo Gemfile](#) em seu pacote de origem para instalar pacotes durante a implantação.

### Configuração do servidor de aplicações

O Elastic Beanstalk instala o servidor de aplicações Puma com base na ramificação da plataforma Ruby que você escolhe ao criar o ambiente. Para obter mais informações sobre os componentes fornecidos com as versões da plataforma Ruby, consulte [Plataformas compatíveis](#) no guia Plataformas do AWS Elastic Beanstalk.

Você pode configurar a aplicação com seu próprio servidor Puma. Isso permite usar uma versão do Puma diferente da que é pré-instalada com a ramificação da plataforma Ruby. Você também pode configurar a aplicação para usar um servidor de aplicações diferente, como o Passenger. Para isso, você deve incluir e personalizar um Gemfile em sua implantação. Você também precisa configurar um Procfile para iniciar o servidor de aplicações. Para obter mais informações, consulte [Configurar o processo da aplicação com um Procfile](#).

### Outras opções de configuração

O Elastic Beanstalk fornece [opções de configuração](#) que podem ser usadas para personalizar o software que é executado nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) no ambiente do Elastic Beanstalk. Você pode configurar variáveis de ambiente necessárias à aplicação, habilitar a alternância de logs para o Amazon S3 e mapear pastas da origem da aplicação que contenham arquivos estáticos para caminhos atendidos pelo servidor de proxy. A plataforma também predefine algumas variáveis de ambiente comuns relacionadas ao Rails e ao Rack para facilitar a descoberta e o uso.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para [modificar a configuração de um ambiente em execução](#). Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar [configurações salvas](#) para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir [arquivos de configuração](#). As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console. Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte [Opções de configuração](#).

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte [the section called “Estender plataformas Linux”](#).

## Configurar o ambiente Ruby

É possível usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3 e configurar variáveis que sua aplicação pode ler no ambiente.

Para acessar as definições de configuração de software para seu ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

## Opções de log

A seção Log options (Opções de log) tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.

- Enable log file rotation to Amazon S3 (Habilitar a rotação de arquivos de log para o Amazon S3): especifica se os arquivos de log das instâncias do Amazon EC2 da aplicação serão copiados para o bucket do Amazon S3 associado a ela.

## Arquivos estáticos

Para melhorar a performance, você pode usar a seção Static files (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte [the section called “Arquivos estáticos”](#).

Por padrão, o servidor de proxy em um ambiente Ruby é configurado para atender arquivos estáticos da seguinte maneira:

- Arquivos na pasta `public` são atendidos a partir do caminho `/public` e a raiz do domínio (caminho `/`).
- Arquivos da subpasta `public/assets` são atendidos a partir do caminho `/assets`.

Os exemplos a seguir ilustram como funciona a configuração padrão:

- Por exemplo, se a origem da aplicação-fonte contiver um arquivo chamado `logo.png` em uma pasta chamada `public`, o servidor de proxy o servirá a usuários a partir de `subdomain.elasticbeanstalk.com/public/logo.png` e `subdomain.elasticbeanstalk.com/logo.png`.
- Se a origem da aplicação-fonte contiver um arquivo nomeado `logo.png` em uma pasta nomeada `assets`, dentro da pasta `public`, o servidor de proxy o servirá a partir de `subdomain.elasticbeanstalk.com/assets/logo.png`.

Você pode configurar mais mapeamentos para arquivos estáticos. Para obter mais informações, consulte [Namespaces de configuração do Ruby](#) mais adiante neste tópico.



**Note**

Para versões de plataforma anteriores a Ruby 2.7 AL2 versão 3.3.7, a configuração padrão do servidor de proxy Elastic Beanstalk nginx não é compatível com a veiculação de arquivos estáticos a partir da raiz do domínio (*subdomain*.elasticbeanstalk.com/). Esta versão da plataforma foi lançada em 21 de outubro de 2021. Para obter mais informações, consulte [Novas versões da plataforma - Ruby](#) nas Notas de Release do AWS Elastic Beanstalk.

## Propriedades de ambiente

A seção Environment Properties permite que você especifique definições de configuração do ambiente nas instâncias do Amazon EC2 que estão executando seu aplicativo. As propriedades de ambiente são passadas para o aplicativo como pares de chave-valor.

A plataforma Ruby define as seguintes propriedades de configuração do ambiente:

- `BUNDLE_WITHOUT`: uma lista separada por dois pontos de grupos a ignorar ao [instalar dependências](#) de um [Gemfile](#).
- `BUNDLER_DEPLOYMENT_MODE`: defina como `true` (padrão) para instalar as dependências no [modo de implantação](#) usando o Bundler. Defina como `false` para executar `bundle install` no modo de desenvolvimento.

**Note**

Essa propriedade de ambiente não é definida nas ramificações da plataforma Ruby da AMI do Amazon Linux (anteriores ao Amazon Linux 2).

- `RAILS_SKIP_ASSET_COMPILATION`: definido como `true` para ignorar a execução de [rake assets:precompile](#) durante a implantação.
- `RAILS_SKIP_MIGRATIONS`: definido como `true` para ignorar a execução de [rake db:migrate](#) durante a implantação.
- `RACK_ENV`: especifique o estágio do ambiente para o Rack. Por exemplo, `development`, `production` ou `test`.

No ambiente Ruby executado no Elastic Beanstalk, as variáveis de ambiente podem ser acessadas usando o objeto ENV. Por exemplo, você pode ler uma propriedade denominada API\_ENDPOINT em uma variável com o seguinte código:

```
endpoint = ENV['API_ENDPOINT']
```

Consulte [Propriedades de ambientes e outras configurações de software](#) para obter mais informações.

## Namespaces de configuração do Ruby

Você pode usar um [arquivo de configuração](#) para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. As opções de configuração podem ser definidas pelo serviço Elastic Beanstalk ou pela plataforma utilizada, e são organizadas em namespaces.

É possível usar o namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` para configurar o proxy de ambiente para servir arquivos estáticos. Defina mapeamentos de caminhos virtuais para diretórios de aplicativos.

A plataforma Ruby não define nenhum namespace específico da plataforma. Em vez disso, ela define as propriedades do ambiente para opções de Rails e Rack comuns.

O arquivo de configuração a seguir especifica uma opção de arquivos estáticos que mapeia um diretório chamado `staticimages` para o caminho `/images`, define cada uma das propriedades de ambiente definidas pela plataforma e define uma propriedade de ambiente adicional chamada `LOGGING`.

### Example `.ebextensions/ruby-settings.config`

```
option_settings:
 aws:elasticbeanstalk:environment:proxy:staticfiles:
 /images: staticimages
 aws:elasticbeanstalk:application:environment:
 BUNDLE_WITHOUT: test
 BUNDLER_DEPLOYMENT_MODE: true
 RACK_ENV: development
 RAILS_SKIP_ASSET_COMPILATION: true
 RAILS_SKIP_MIGRATIONS: true
 LOGGING: debug
```

**Note**

A propriedade de ambiente `BUNDLER_DEPLOYMENT_MODE` e o namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` não são definidos nas ramificações da plataforma Ruby de AMI do Amazon Linux (anterior ao Amazon Linux 2).

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte [Opções de configuração](#) para obter mais informações.

## Instalar pacotes com um Gemfile

Use um arquivo `Gemfile` na raiz da origem de seu projeto para usar o RubyGems para instalar os pacotes exigidos pelo seu aplicativo.

### Example Gemfile

```
source "https://rubygems.org"
gem 'sinatra'
gem 'json'
gem 'rack-parser'
```

Quando um arquivo `Gemfile` está presente, o Elastic Beanstalk executa `bundle install` para instalar dependências. Para mais informações, consulte as páginas [Gemfiles](#) e [Bundle](#) (Pacote) no site Bundler.io.

**Note**

Você pode usar uma versão diferente do Puma além da versão padrão que é pré-instalada com a plataforma Ruby. Para isso, inclua uma entrada em um `Gemfile` especificando a versão. Você também pode especificar um servidor de aplicações diferente, como o Passenger, usando um `Gemfile` personalizado.

Em ambos desses casos, você precisa configurar um `Procfile` para iniciar o servidor de aplicações.

Para obter mais informações, consulte [Configurar o processo da aplicação com um Procfile](#).

## Configurar o processo do aplicativo com um Procfile

Para especificar o comando que inicia a aplicação em Ruby, inclua um arquivo chamado `Procfile` na raiz do pacote de origem.

### Note

O Elastic Beanstalk não é compatível com esse recurso nas ramificações da plataforma Ruby da AMI do Amazon Linux (anteriores ao Amazon Linux 2). As ramificações da plataforma com os nomes que contêm Puma ou Passenger, independentemente de suas versões Ruby, precedem o Amazon Linux 2 e não são compatíveis com o recurso `Procfile`.

Para obter detalhes sobre como escrever e usar um `Procfile`, expanda a seção `Buildfile` e `Procfile` em [the section called “Estender plataformas Linux”](#).

Quando você não fornece um `Procfile`, o Elastic Beanstalk gera o arquivo padrão a seguir, que pressupõe que você esteja usando o servidor de aplicações Puma pré-instalado.

```
web: puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

Se você quiser usar seu próprio servidor Puma fornecido, é possível instalá-lo usando um [Gemfile](#). O seguinte exemplo de `Procfile` mostra como iniciá-lo.

### Example Procfile

```
web: bundle exec puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

Se você quiser usar o servidor de aplicativos Passenger, use os seguintes exemplos de arquivos para configurar seu ambiente Ruby a ser instalado e usar o Passenger.

1. Use este exemplo de arquivo para instalar o Passenger.

### Example Gemfile

```
source 'https://rubygems.org'
gem 'passenger'
```

2. Use este arquivo demonstrativo para instruir o Elastic Beanstalk a iniciar o Passenger.

## Example Procfile

```
web: bundle exec passenger start /var/app/current --socket /var/run/puma/my_app.sock
```

### Note

Não é necessário alterar nada na configuração do servidor de proxy nginx para usar o Passenger. Para usar outros servidores de aplicativos, talvez seja necessário personalizar a configuração do nginx para encaminhar solicitações adequadamente para seu aplicativo.

## Implantar uma aplicação rails no Elastic Beanstalk

Rails é uma estrutura de código aberto model-view-controller (MVC) para Ruby. Este tutorial mostra o processo de geração de um aplicativo Rails e sua implantação em um AWS Elastic Beanstalk ambiente.

### Seções

- [Pré-requisitos](#)
- [Iniciar um ambiente do Elastic Beanstalk](#)
- [Instalar o rails e gerar um site](#)
- [Definir as configurações do rails](#)
- [Implantar o aplicativo](#)
- [Limpeza](#)
- [Próximas etapas](#)

### Pré-requisitos

#### Conhecimento básico do Elastic Beanstalk

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em [Conceitos básicos do Elastic Beanstalk](#) para iniciar seu primeiro ambiente do Elastic Beanstalk.

## Linha de comando

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt (\$) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows 10, você pode [instalar o Subsistema Windows para Linux](#) para obter uma versão do Ubuntu integrada com o Windows e o Bash.

## Dependências do rails

O framework Rails 6.1.4.1 tem as dependências a seguir. Certifique-se de que você tem todas elas instaladas.

- Ruby 2.5.0 ou mais recente: para obter instruções de instalação, consulte [Configurar ambiente de desenvolvimento Ruby](#).

Neste tutorial, usamos Ruby 3.0.2 e a versão correspondente da plataforma do Elastic Beanstalk.

- Node.js: para obter instruções de instalação, consulte [Instalar o Node.js como o gerenciador de pacotes](#).
- Yarn: para obter instruções de instalação, consulte [Installation](#) no site do Yarn.

## Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma Ruby e aceite as configurações padrão e o código de exemplo.

Para iniciar um ambiente (console)

1. [Abra o console do Elastic Beanstalk usando este link pré-configurado:](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)  
[console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.

3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).

- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

#### Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

#### Note

O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para ter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon S3](#).

## Instalar o rails e gerar um site

Instale o Rails e suas dependências com o comando `gem`.

```
~$ gem install rails
Fetching: concurrent-ruby-1.1.9.gem
Successfully installed concurrent-ruby-1.1.9
Fetching: rack-2.2.3.gem
Successfully installed rack-2.2.3
...
```

Teste a instalação do Rails.

```
~$ rails --version
Rails 6.1.4.1
```



Use `rails new` com o nome do aplicativo para criar um novo projeto Rails.

```
~$ rails new ~/eb-rails
```

O Rails cria um diretório com o nome especificado, gera todos os arquivos necessários para executar um projeto de exemplo localmente e, em seguida, executa o bundler para instalar todas as dependências (Gems) definidas no Gemfile do projeto.

### Note

Este processo instala a versão mais recente do Puma para o projeto. Talvez essa versão seja diferente da versão que o Elastic Beanstalk fornece na versão da plataforma Ruby do seu ambiente. Para ver as versões do Puma fornecidas pelo Elastic Beanstalk, consulte [Histórico de plataforma Ruby](#) no Guia de plataformas do AWS Elastic Beanstalk. Para mais informações sobre a versão mais recente do Puma, consulte o site [Puma.io](#). Se houver uma incompatibilidade entre as duas versões do Puma, adote uma das seguintes opções:

- Use a versão do Puma instalada pelo comando `rails new` anterior. Nesse caso, é necessário adicionar um `Procfile` para que a plataforma use a versão do servidor Puma que você fornece. Para ter mais informações, consulte [Configurar o processo do aplicativo com um Procfile](#).
- Atualize a versão do Puma para ser consistente com a versão pré-instalada na versão da plataforma Ruby do seu ambiente. Para fazer isso, modifique a versão do Puma no [Gemfile](#) localizado na raiz do diretório de origem do projeto. Em seguida, execute `bundle update`. Para mais informações, consulte a página [bundle update](#) (atualização do pacote) no site Bundler.io.

Teste a instalação do Rails executando o projeto padrão localmente.

```
~$ cd eb-rails
~/eb-rails$ rails server
=> Booting Puma
=> Rails 6.1.4.1 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 5.5.2 (ruby 3.0.2-p107) ("Zawgyi")
* Min threads: 5
* Max threads: 5
```

```
* Environment: development
* PID: 77857
* Listening on http://127.0.0.1:3000
* Listening on http://[::1]:3000
Use Ctrl-C to stop
...
```

Abra `http://localhost:3000` em um navegador da web para ver o projeto padrão em ação.



Essa página só está visível em modo de desenvolvimento. Adicione algum conteúdo à página inicial da aplicação para oferecer suporte à implantação de produção no Elastic Beanstalk. Use `rails generate` para criar um controlador, uma rota e uma visualização de sua página de boas-vindas.

```
~/eb-rails$ rails generate controller WelcomePage welcome
 create app/controllers/welcome_page_controller.rb
 route get 'welcome_page/welcome'
 invoke erb
 create app/views/welcome_page
 create app/views/welcome_page/welcome.html.erb
```

```
invoke test_unit
create test/controllers/welcome_page_controller_test.rb
invoke helper
create app/helpers/welcome_page_helper.rb
invoke test_unit
invoke assets
invoke coffee
create app/assets/javascripts/welcome_page.coffee
invoke scss
create app/assets/stylesheets/welcome_page.scss.
```

Isso fornece tudo o que você precisa para acessar a página na `/welcome_page/welcome`. No entanto, antes de publicar as alterações, altere o conteúdo na visualização e adicione uma rota para que essa página apareça no nível superior do site.

Use um editor de texto para editar o conteúdo em `app/views/welcome_page/welcome.html.erb`. Para este exemplo, você usará `cat` para simplesmente substituir o conteúdo do arquivo existente.

Example `app/views/welcome_page/welcome.html.erb`

```
<h1>Welcome!</h1>
<p>This is the front page of my first Rails application on Elastic Beanstalk.</p>
```

Por fim, adicione a seguinte rota a `config/routes.rb`:

Example `config/routes.rb`

```
Rails.application.routes.draw do
 get 'welcome_page/welcome'
 root 'welcome_page#welcome'
```

Isso instrui o Rails a rotear as solicitações à raiz do site para o método de boas-vindas do controlador da página de boas-vindas, que renderiza o conteúdo na visualização de boas-vindas (`welcome.html.erb`).

Para que o Elastic Beanstalk implante com sucesso a aplicação na plataforma Ruby, precisamos atualizar o `Gemfile.lock`. Algumas dependências do `Gemfile.lock` talvez sejam específicas da plataforma. Portanto, precisamos adicionar **platform ruby** a `Gemfile.lock` para que todas as dependências necessárias sejam instaladas com a implantação.

## Example

```
~/eb-rails$ bundle lock --add-platform ruby
Fetching gem metadata from https://rubygems.org/.....
Resolving dependencies...
Writing lockfile to /Users/janedoe/EBDPT/RubyApps/eb-rails-doc-app/Gemfile.lock
```

## Definir as configurações do rails

Use o console do Elastic Beanstalk para configurar o Rails com propriedades do ambiente. Defina a propriedade do ambiente `SECRET_KEY_BASE` para uma string de até 256 caracteres alfanuméricos.

O Rails usa essa propriedade para criar chaves. Portanto, você deve mantê-la em segredo e não armazená-la no controle de origem em texto sem formatação. Em vez disso, você a fornece ao código do Rails no ambiente por meio de uma propriedade de ambiente.

Como configurar as propriedades do ambiente no console do Elastic Beanstalk

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Role para baixo até Propriedades do ambiente.
6. Selecione Adicionar propriedade do ambiente.
7. Insira os pares Nome e Valor da propriedade.
8. Se for necessário adicionar mais variáveis, repita a Etapa 6 e a Etapa 7.
9. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Agora você está pronto para implantar o site no ambiente.

## Implantar o aplicativo

Crie um [pacote de origem](#) com os arquivos criados pelo Rails. O comando a seguir cria um pacote de origem chamado `rails-default.zip`.

```
~/eb-rails$ zip ../rails-default.zip -r * .[^.]*
```

Faça upload do pacote de origem no Elastic Beanstalk para implantar o Rails no ambiente.

Para implantar um pacote de origem

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

## Limpeza

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. [O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente, como instâncias do Amazon EC2, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.](#)

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

**Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

## Próximas etapas

Para obter mais informações sobre o Rails, visite [rubyonrails.org](http://rubyonrails.org).

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de [linha de comando do Elastic Beanstalk](#) (EB CLI easy-to-use ) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, [configure um nome de domínio personalizado](#) para seu ambiente e [habilite HTTPS](#) para conexões seguras.

## Implantar uma aplicação sinatra no Elastic Beanstalk

Este passo a passo mostra como implantar uma aplicação web [Sinatra](#) simples no AWS Elastic Beanstalk.

### Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em [Conceitos básicos do Elastic Beanstalk](#) para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt (\$) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows 10, você pode [instalar o Subsistema Windows para Linux](#) para obter uma versão do Ubuntu integrada com o Windows e o Bash.

O Sinatra 2.1.0 requer o Ruby 2.3.0 ou mais recente. Neste tutorial, usamos Ruby 3.0.2 e a versão correspondente da plataforma do Elastic Beanstalk. Instale o Ruby seguindo as instruções em [Configurar ambiente de desenvolvimento Ruby](#).

## Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma Ruby e aceite as configurações padrão e o código de exemplo.

Para iniciar um ambiente (console)

1. [Abra o console do Elastic Beanstalk usando este link pré-configurado:](#)  
[console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced](https://console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=Tutorials&EnvironmentType=LoadBalanced)
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- Instância do EC2: uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança de instância: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga: um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).
- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

#### Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.



Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

### Note

O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para ter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon S3](#).

## Escrever um site sinatra básico

Como criar e implantar um aplicativo sinatra

1. Crie um arquivo de configuração chamado `config.ru` com o conteúdo a seguir.

Example `config.ru`

```
require './helloworld'
run Sinatra::Application
```

2. Crie um arquivo de código Ruby chamado `helloworld.rb` com o conteúdo a seguir.

Example `helloworld.rb`

```
require 'sinatra'
get '/' do
 "Hello World!"
end
```

3. Crie um Gemfile com o conteúdo a seguir.

Example Gemfile

```
source 'https://rubygems.org'
gem 'sinatra'
gem 'puma'
```

4. Execute a instalação do pacote para gerar o `Gemfile.lock`

## Example

```
~/eb-sinatra$ bundle install
Fetching gem metadata from https://rubygems.org/....
Resolving dependencies...
Using bundler 2.2.22
Using rack 2.2.3
...
```

5. Para que o Elastic Beanstalk implante com sucesso a aplicação na plataforma Ruby, precisamos atualizar o `Gemfile.lock`. Algumas dependências do `Gemfile.lock` talvez sejam específicas da plataforma. Portanto, precisamos adicionar **platform ruby** a `Gemfile.lock` para que todas as dependências necessárias sejam instaladas com a implantação.

## Example

```
~/eb-sinatra$ bundle lock --add-platform ruby
Fetching gem metadata from https://rubygems.org/....
Resolving dependencies...
Writing lockfile to /Users/janedoe/EBDPT/RubyApps/eb-sinatra/Gemfile.lock
```

6. Crie um arquivo Procfile com o seguinte conteúdo:

## Example Procfile

```
web: bundle exec puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

## Implantar o aplicativo

Crie um [pacote de origem](#) com os arquivos de origem. O comando a seguir cria um pacote de origem chamado `sinatra-default.zip`.


```
~/eb-sinatra$ zip ../sinatra-default.zip -r * .[^.]*
```

Faça upload do pacote de origem no Elastic Beanstalk para implantar o Sinatra no ambiente.

Para implantar um pacote de origem

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS

2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.


3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

## Limpeza

Ao terminar de trabalhar com o Elastic Beanstalk, você pode encerrar o ambiente. [O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente, como instâncias do Amazon EC2, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.](#)

Como encerrar o ambiente do Elastic Beanstalk

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Com o Elastic Beanstalk, é possível criar facilmente um ambiente para a aplicação a qualquer momento.

## Próximas etapas

Para mais informações sobre o Sinatra, acesse [sinatrarb.com](http://sinatrarb.com).

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de [linha de comando do Elastic Beanstalk](#) (EB CLI easy-to-use ) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, [configure um nome de domínio personalizado](#) para seu ambiente e [habilite HTTPS](#) para conexões seguras.

## Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicações Ruby

É possível usar uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS) para armazenar dados coletados e modificados pela aplicação. O banco de dados pode ser acoplado ao seu ambiente e gerenciado pelo Elastic Beanstalk ou pode ser criado como desacoplado e gerenciado externamente por outro serviço. Este tópico fornece instruções para criar o Amazon RDS usando o console do Elastic Beanstalk. O banco de dados será acoplado ao ambiente e gerenciado pelo Elastic Beanstalk. Para obter mais informações sobre como integrar um Amazon RDS com o Elastic Beanstalk, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

### Seções

- [Adicionar uma instância de banco de dados ao seu ambiente](#)
- [Download de adaptador](#)
- [Conectar-se a um banco de dados](#)

## Adicionar uma instância de banco de dados ao seu ambiente

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização do ambiente for concluída, o nome de host da instância de banco de dados e outras informações de conexão estarão disponíveis para o seu aplicativo por meio das seguintes propriedades de ambiente:

Nome da propriedade	Descrição	Valor da propriedade
RDS_HOSTNAME	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
RDS_PORT	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
RDS_DB_NAME	O nome do banco de dado, <b>ebdb</b> .	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
RDS_USERNAME	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Para obter mais informações sobre como configurar uma instância de banco de dados acoplada a um ambiente Elastic Beanstalk, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

## Download de adaptador

Adicione o adaptador de banco de dados ao [arquivo de gem](#) de seu projeto.

Exemplo Gemfile: Rails com o MySQL

```
source 'https://rubygems.org'
gem 'puma'
gem 'rails', '~> 6.1.4', '>= 6.1.4.1'
gem 'mysql2'
```

Gems de adaptadores comuns para Ruby

- MySQL – [mysql2](#)
- PostgreSQL – [pg](#)
- Oracle: [activerecord-oracle\\_enhanced-adapter](#)
- SQL Server – [activerecord-sqlserver-adapter](#)

## Conectar-se a um banco de dados

O Elastic Beanstalk fornece informações de conexão referentes às instâncias de banco de dados associadas nas propriedades de ambiente. Use `ENV['VARIABLE']` para ler as propriedades e configurar uma conexão de banco de dados.

Exemplo config/database.yml: configuração do banco de dados de Ruby on Rails (MySQL)

```
production:
 adapter: mysql2
 encoding: utf8
 database: <%= ENV['RDS_DB_NAME'] %>
 username: <%= ENV['RDS_USERNAME'] %>
 password: <%= ENV['RDS_PASSWORD'] %>
 host: <%= ENV['RDS_HOSTNAME'] %>
 port: <%= ENV['RDS_PORT'] %>
```

# Tutoriais e amostras

Tutoriais específicos de linguagem e estrutura estão espalhados por todo o Guia do AWS Elastic Beanstalk Desenvolvedor. Tutoriais novos e atualizados são adicionados a essa lista à medida que são publicados. As atualizações mais recentes são mostradas primeiro.

Esses tutoriais destinam-se a usuários intermediários e podem não conter instruções de etapas básicas, como cadastrar-se na AWS. Se esta é sua primeira vez usando o AWS Elastic Beanstalk, [confira o passo a passo de introdução para colocar seu primeiro ambiente do Elastic Beanstalk em funcionamento](#).

- Ruby on Rails: [Implantar uma aplicação rails no Elastic Beanstalk](#)
- Ruby e Sinatra: [Implantar uma aplicação sinatra no Elastic Beanstalk](#)
- Configuração de HA do PHP e MySQL: [Implantar uma aplicação PHP de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk](#)
- PHP e Laravel: [Implantar uma aplicação Laravel no Elastic Beanstalk](#)
- PHP e CakePHP: [Implantar uma aplicação CakePHP no Elastic Beanstalk](#)
- Configuração de HA do PHP e Drupal: [Implantar um site Drupal de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk](#)
- Configuração de PHP e WordPress HA - [Implantação de um WordPress site de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk](#)
- Node.js com configuração de HA do DynamoDB: [Implantar uma aplicação Node.js com o DynamoDB no Elastic Beanstalk](#)
- ASP.NET Core: [Tutorial: Implantação de um aplicativo ASP.NET Core com o Elastic Beanstalk](#)
- Python e Flask: [Implantar uma aplicação Flask no Elastic Beanstalk](#)
- Python e Django: [Implantar uma aplicação Django no Elastic Beanstalk](#)
- Node.js e Express: [Implantar uma aplicação Express no Elastic Beanstalk](#)
- Docker, PHP e nginx: [Ambientes do Docker gerenciado pelo ECS com o console do Elastic Beanstalk](#)

É possível fazer download das aplicações demonstrativas usadas pelo Elastic Beanstalk ao criar um ambiente, sem fornecer um pacote de origem com os seguintes links:

- Docker: [docker.zip](#)

- Docker com vários contêineres — [2.zip docker-multicontainer-v](#)
- Docker pré-configurado (Glassfish) — [1.zip docker-glassfish-v](#)
- Go: [go.zip](#)
- Corretto: [corretto.zip](#)
- Tomcat: [tomcat.zip](#)
- .NET Core no Linux — [dotnet-core-linux.zip](#)
- .NET Core — [dotnet-asp-windows.zip](#)
- Node.js: [nodejs.zip](#)
- PHP: [php.zip](#)
- Python: [python.zip](#)
- Ruby: [ruby.zip](#)

Aplicativos de amostra mais envolvidos que mostram o uso de estruturas, bibliotecas e ferramentas adicionais da Web estão disponíveis como projetos de código aberto em GitHub:

- [Balanceamento de carga WordPress \(tutorial\)](#) — Arquivos de configuração para instalação WordPress segura e execução em um ambiente Elastic Beanstalk com balanceamento de carga.
- [Drupal com balanceamento de carga \(tutorial\)](#): arquivos de configuração e instruções para instalar com segurança o Drupal e executá-lo em um ambiente do Elastic Beanstalk com balanceamento de carga.
- [Scorekeep](#) - API web RESTful que usa a estrutura Spring e a AWS SDK for Java para fornecer uma interface para criar e gerenciar usuários, sessões e jogos. A API faz parte do pacote de um aplicativo Web Angular 1.5 que a consome via HTTP. Inclui filiais que mostram a integração com o Amazon Cognito e o Amazon Relational Database Service. AWS X-Ray

O aplicativo usa recursos da plataforma Java SE para fazer download de dependências e criar na instância, minimizando o tamanho do pacote de origem. O aplicativo também inclui arquivos de configuração nginx que substituem a configuração padrão para servir o aplicativo web de front-end estaticamente na porta 80 através do proxy e encaminhar as solicitações para atalhos em /api para a API em execução em localhost:5000.

- [Tem cobras?](#) - Aplicativo Tomcat que mostra o uso do RDS em um aplicativo web Java EE no Elastic Beanstalk. O projeto mostra o uso de servlets, JSPs, Simple Tag Support, Tag Files, JDBC, SQL, Log4J, Bootstrap, Jackson e arquivos de configuração do Elastic Beanstalk.



- [Locust Load Generator](#) – esse projeto mostra o uso de recursos da Plataforma Java SE para instalar e executar [Locust](#), uma ferramenta de geração de carga escrita em Python. O projeto inclui arquivos de configuração que instalam e configuram o Locust, um script de compilação que configura uma tabela do DynamoDB, e um Procfile que executa o Locust.
- [Share Your Thoughts \(tutorial\)](#): aplicação em PHP que mostra o uso de MySQL no Amazon RDS, Composer e arquivos de configuração.
- [Uma nova inicialização \(tutorial\)](#) - exemplo de aplicativo Node.js que mostra o uso do DynamoDB, o SDK do Node.js, AWS do gerenciamento JavaScript de pacotes npm e dos arquivos de configuração.

# Gerenciar e configurar aplicações do Elastic Beanstalk

A primeira etapa para usar o AWS Elastic Beanstalk é criar uma aplicação, que representa sua aplicação web na AWS. No Elastic Beanstalk, uma aplicação serve como contêiner para os ambientes que executam sua aplicação Web e para as versões de seu código-fonte, as configurações salvas, os logs e outros artefatos da aplicação Web criados ao usar o Elastic Beanstalk.

Para criar um aplicativo.

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, escolha Applications (Aplicações) e Create a new application (Criar uma nova aplicação).
3. Use o formulário na tela para fornecer um nome de aplicativo.
4. Opcionalmente, forneça uma descrição e adicione chaves e valores de tag.
5. Escolha Create (Criar).

Elastic Beanstalk

## Create new application

**Application information**

Application Name

Maximum length of 100 characters, not including forward slash (/).

Description

**Tags**

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

Key	Value	
<input type="text"/>	<input type="text"/>	<input type="button" value="Remove tag"/>

50 remaining

Após a criação do aplicativo, o console solicita que você crie um ambiente para ele. Para obter informações detalhadas sobre todas as opções disponíveis, consulte [Criar um ambiente do Elastic Beanstalk](#).

Se você não precisar mais de um aplicativo, poderá excluí-lo.

**Warning**

A exclusão de um aplicativo encerra todos os ambientes associados e exclui todas as versões e configurações salvas do aplicativo.

## Para excluir um aplicativo

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, escolha Applications (Aplicativos) e selecione seu aplicativo na lista.
3. Escolha Actions (Ações) e Delete application (Excluir aplicativo).

## Tópicos

- [Console de gerenciamento de aplicações do Elastic Beanstalk](#)
- [Gerenciar versões de aplicativos](#)
- [Criar um pacote de origem do aplicativo](#)
- [Marcar recursos da aplicação do Elastic Beanstalk](#)

# Console de gerenciamento de aplicações do Elastic Beanstalk

Use o console do AWS Elastic Beanstalk para gerenciar aplicações, versões de aplicações e configurações salvas.

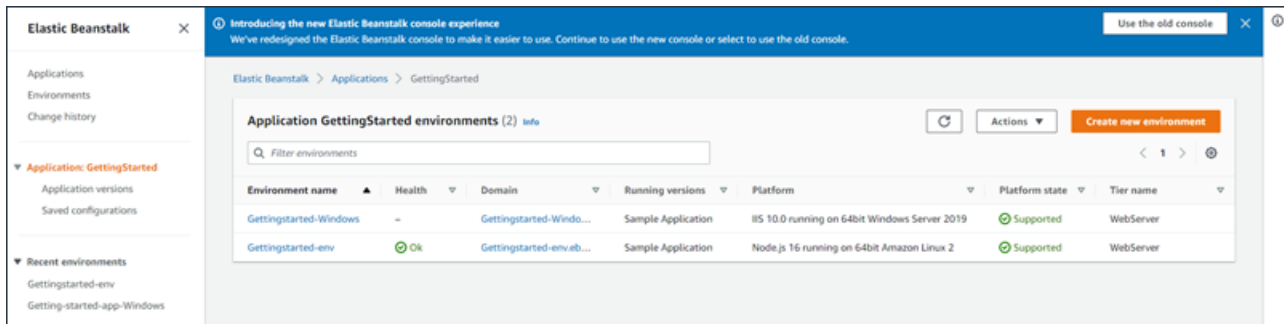
Para acessar o console de gerenciamento de aplicativos

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

### Note

Se você tiver muitos aplicativos, use a barra de pesquisa para filtrar a lista de aplicativos.

A página de visão geral do aplicativo exibe uma lista com uma visão geral de todos os ambientes associados ao aplicativo.



### 3. Você tem algumas maneiras de continuar:

- a. Escolha o menu suspenso Actions (Ações) e escolha uma das ações de gerenciamento de aplicativos. Para iniciar um ambiente neste aplicativo, é possível escolher diretamente Create a new environment (Criar um ambiente). Para obter mais detalhes, consulte [the section called “Criar ambientes”](#).
- b. Escolha um nome de ambiente para acessar o [console de gerenciamento de ambiente](#) desse ambiente em que é possível configurar, monitorar ou gerenciar o ambiente.
- c. Escolha Application versions (Versões do aplicativo) após o nome do aplicativo no painel de navegação para visualizar e gerenciar as versões do aplicativo.

Uma versão do aplicativo é uma versão carregada do código do aplicativo. É possível fazer upload de novas versões, implantar uma versão existente em qualquer um dos ambientes do aplicativo ou excluir versões antigas. Para obter mais informações, consulte [Gerenciar versões de aplicativos](#).

- d. Escolha Saved configurations (Configurações salvas) após o nome do aplicativo no painel de navegação para visualizar e gerenciar configurações salvas de ambientes em execução.

Uma configuração salva é um conjunto de configurações que você pode usar para restaurar configurações de um ambiente para um estado anterior ou para criar um ambiente com as mesmas configurações. Para mais informações, consulte [Usar configurações salvas do Elastic Beanstalk](#).

## Gerenciar versões de aplicativos

O Elastic Beanstalk cria uma versão da aplicação sempre que você faz upload do código-fonte. Isso normalmente ocorre quando você cria um ambiente ou faz upload e implanta o código usando o [console de gerenciamento do ambiente](#) ou a [CLI do EB](#). O Elastic Beanstalk exclui essas versões da aplicação de acordo com a política de ciclo de vida da aplicação e quando a aplicação é excluída.

Para obter detalhes sobre a política de ciclo de vida de aplicativo, consulte [Definir as configurações de ciclo de vida da versão do aplicativo](#).


Também é possível fazer upload de um pacote de origem sem implantá-lo usando o [console de gerenciamento da aplicação](#) ou com o comando [eb appversion](#) da EB CLI. O Elastic Beanstalk armazena pacotes de origem no Amazon Simple Storage Service (Amazon S3) e não os exclui automaticamente.

Você pode aplicar tags a uma versão do aplicativo ao criá-la e editar tags de versões existentes do aplicativo. Para obter mais detalhes, consulte [Atribuir tags em versões do aplicativo](#).

Para criar uma nova versão do aplicativo


Você também pode criar outra versão da aplicação usando a EB CLI. Para obter mais informações, consulte [eb appversion](#) no capítulo Comandos da EB CLI.

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

 Note

Se você tiver muitos aplicativos, use a barra de pesquisa para filtrar a lista de aplicativos.

3. No painel de navegação, encontre o nome do aplicativo e escolha Application versions (Versões do aplicativo).
4. Escolha Upload (Carregar). Use o formulário na tela para carregar o [pacote de origem](#) do aplicativo.

 Note

O tamanho máximo do arquivo do pacote de fontes é 500 MB.

5. Opcionalmente, forneça uma breve descrição e adicione chaves e valores de tag.
6. Escolha Upload (Carregar).

O arquivo que você especificou está associado ao seu aplicativo. Você pode implantar a versão do aplicativo em um ambiente novo ou existente.

Com o tempo, seu aplicativo pode acumular várias versões do aplicativo. Para economizar espaço de armazenamento e evitar atingir a [cota de versões da aplicação](#), é recomendável excluir versões da aplicação que não são mais necessárias.

### Note

A exclusão de uma versão do aplicativo não afeta os ambientes que a estão executando no momento.

Para excluir uma versão do aplicativo

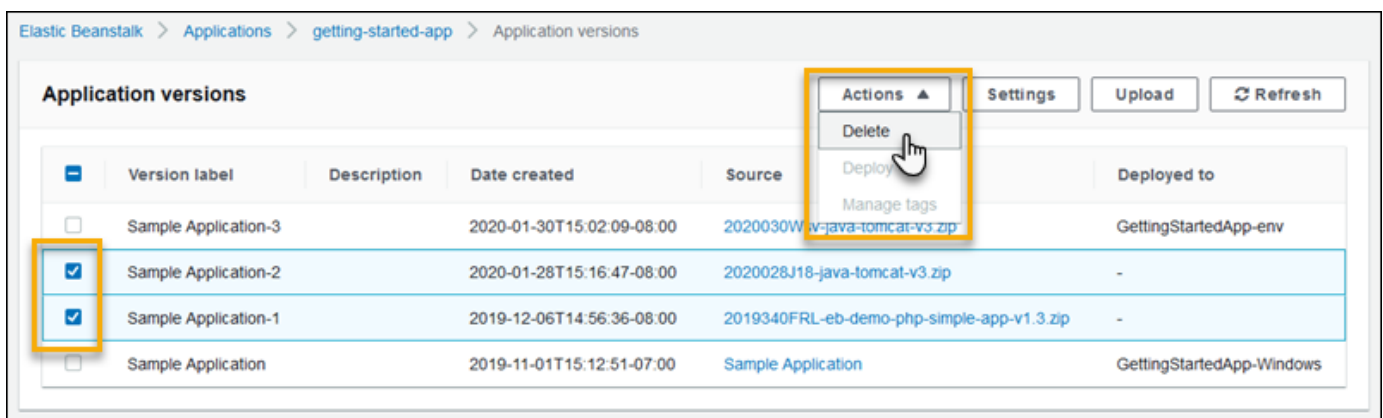
Você também pode excluir uma versão da aplicação usando a EB CLI. Para obter mais informações, consulte [eb appversion](#) no capítulo Comandos da EB CLI.

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

### Note

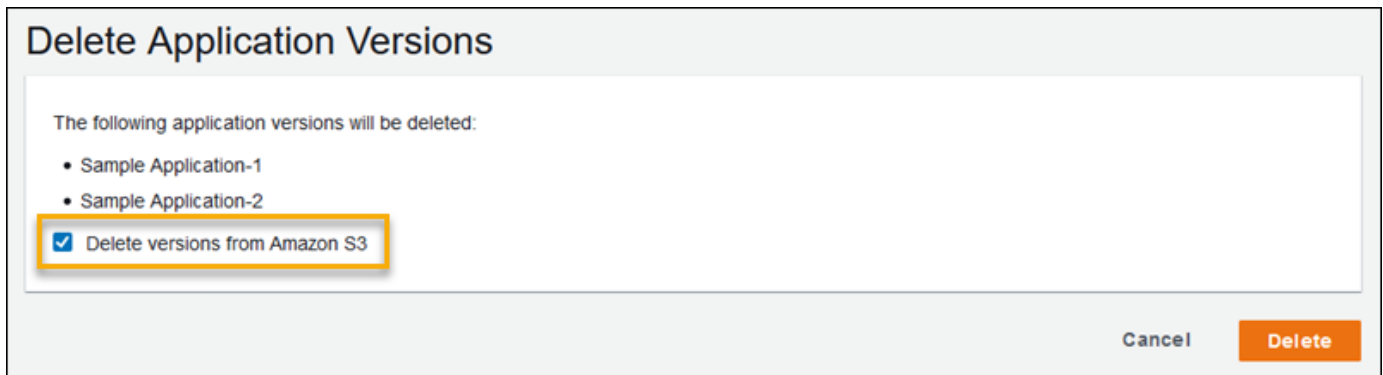
Se você tiver muitos aplicativos, use a barra de pesquisa para filtrar a lista de aplicativos.

3. No painel de navegação, encontre o nome do aplicativo e escolha Application versions (Versões do aplicativo).
4. Selecione uma ou mais versões do aplicativo que você deseja excluir.



5. Escolha Actions (Ações) e Delete (Excluir).

6. (Opcional) Para deixar o pacote de origem da aplicação para essas versões de aplicações no bucket do Amazon Simple Storage Service (Amazon S3), desmarque a caixa Delete versions from Amazon S3 (Excluir versões do Amazon S3).



7. Escolha Delete (Excluir).

Também é possível configurar o Elastic Beanstalk para excluir versões antigas automaticamente ao definir as configurações do ciclo de vida de versões da aplicação. Se você definir essas configurações de ciclo de vida do aplicativo, elas serão aplicadas a novas versões do aplicativo. Por exemplo, se você configurar, no máximo, 25 versões da aplicação, o Elastic Beanstalk excluirá a versão mais antiga quando você fizer upload da 26ª versão. Se você definir um período máximo de 90 dias, todas as versões com mais de 90 dias serão excluídas quando você fizer upload de uma nova versão. Para obter mais detalhes, consulte [the section called “Ciclo de vida da versão”](#).

Se você não optar por excluir o pacote de origem do Amazon S3, o Elastic Beanstalk ainda excluirá a versão de seus registros. No entanto, o pacote de origem é deixado no [bucket de armazenamento do Elastic Beanstalk](#). A cota de versões da aplicação se aplica apenas a versões monitoradas pelo Elastic Beanstalk. Portanto, se for necessário, você poderá excluir versões para se manter dentro da cota, mas retenha todos os pacotes de origem no Amazon S3.

#### Note

A cota de versões da aplicação não se aplica a pacotes de origem, mas ainda assim pode haver cobranças do Amazon S3 e reter informações pessoais além do tempo necessário. O Elastic Beanstalk nunca exclui pacotes de origem automaticamente. Você deve excluir os pacotes de origem quando não precisar mais deles.



## Definir as configurações de ciclo de vida da versão do aplicativo

Cada vez que você faz upload de uma nova versão da sua aplicação com o console do Elastic Beanstalk ou a CLI do EB, o Elastic Beanstalk cria uma [versão da aplicação](#). Se você não excluir as versões que não usa mais, acabará atingindo a [cota de versões da aplicação](#) e não poderá criar novas versões dela.

Você pode evitar atingir a cota aplicando uma política de ciclo de vida de versão do aplicativo aos aplicativos. Uma política de ciclo de vida instrui o Elastic Beanstalk a excluir versões da aplicação antigas ou excluí-las quando o total de versões de uma determinada aplicação excede um número especificado.

O Elastic Beanstalk aplica uma política de ciclo de vida de uma aplicação cada vez que uma nova versão da aplicação é aplicada e exclui até 100 versões cada vez que a política de ciclo de vida é aplicada. O Elastic Beanstalk exclui versões antigas depois de criar a nova versão e não considera a nova versão no número máximo de versões definido na política.

O Elastic Beanstalk não exclui versões da aplicação que estão sendo usadas por um ambiente nem versões da aplicação implantadas em ambientes que foram encerrados menos de dez semanas antes do acionamento da política.

A cota de versões do aplicativo é utilizada em todos os aplicativos em uma região. Se você tiver vários aplicativos, configure cada um com uma política de ciclo de vida adequada para evitar atingir a cota. Por exemplo, se você tiver 10 aplicações em uma região e a cota for 1.000 versões da aplicação, considere definir uma política de ciclo de vida com uma cota de 99 versões para todas as aplicações ou definir outros valores em cada aplicação, desde que o total seja inferior a 1.000 versões. O Elastic Beanstalk somente aplicará a política se a criação da versão da aplicação for bem-sucedida, portanto, se você já atingiu a cota, deverá excluir algumas versões manualmente antes de criar uma nova.

Por padrão, o Elastic Beanstalk deixa o [pacote de origem](#) da versão da aplicação no Amazon S3 para evitar a perda de dados. Você pode excluir esse pacote para economizar espaço.

É possível definir as configurações de ciclo de vida por meio da CLI e das APIs do Elastic Beanstalk. Consulte [eb appversion](#), [CreateApplication](#) (usando o parâmetro `ResourceLifecycleConfig`) e [UpdateApplicationResourceLifecycle](#) para obter mais detalhes.

### Definir as configurações de ciclo de vida do aplicativo no console

É possível especificar as configurações do ciclo de vida no console do Elastic Beanstalk.

## Como especificar configurações de ciclo de vida do seu aplicativo


1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

### Note

Se você tiver muitos aplicativos, use a barra de pesquisa para filtrar a lista de aplicativos.

3. No painel de navegação, encontre o nome do aplicativo e escolha Application versions (Versões do aplicativo).
4. Escolha Settings (Configurações).
5. Use o formulário na tela para definir as configurações do ciclo de vida do aplicativo.
6. Escolha Save (Salvar).

### Application version lifecycle settings ✕

Configure a lifecycle policy to limit the number of application versions to retain for future deployments. This policy will not delete application versions that are currently deployed or are in the process of being created. [Learn more](#) 

**Lifecycle policy**

Enable

**Lifecycle rule**

Set the application versions limit by total count

200 Application Versions

Set the application versions limit by age

180 days

**Retention**

Delete source bundle from S3

**Service role**

Na página de configurações, você pode fazer o seguinte:

- Defina as configurações do ciclo de vida com base na contagem total de versões do aplicativo ou na idade das versões do aplicativo.
- Especifique se deseja excluir o pacote de origem do S3 quando a versão do aplicativo for excluída.
- Especifique a função de serviço na qual a versão do aplicativo é excluída. Para incluir todas as permissões necessárias para a exclusão da versão, escolha a função de serviço padrão do Elastic Beanstalk, denominada `aws-elasticbeanstalk-service-role`, ou outra função de serviço usando as políticas de serviço gerenciadas do Elastic Beanstalk. Para obter mais informações, consulte [Gerenciar funções de serviço do Elastic Beanstalk](#).

## Atribuir tags em versões do aplicativo

Você pode aplicar tags às versões da aplicação do AWS Elastic Beanstalk. As tags são pares de chave-valor associados a recursos da AWS. Para obter informações sobre a atribuição de tags do recurso do Elastic Beanstalk, casos de uso, restrições de chave e valor de tag, além de tipos de recursos compatíveis, consulte [Marcar recursos da aplicação do Elastic Beanstalk](#).

Você pode especificar tags quando criar uma versão do aplicativo. Em uma versão de aplicativo existente, você pode adicionar ou remover tags e atualizar os valores de tags existentes. Você pode adicionar até 50 tags para cada versão do aplicativo.

### Adicionar tags durante a criação da versão do aplicativo

Ao usar o console do Elastic Beanstalk para [criar um ambiente](#) e optar por fazer upload de uma versão do código da aplicação, você pode especificar chaves e valores de tags para associar à nova versão da aplicação.

Também é possível usar o console do Elastic Beanstalk para [fazer upload de uma versão da aplicação](#) sem usá-lo imediatamente em um ambiente. Você pode especificar valores e chaves de tag ao carregar uma versão do aplicativo.

Com a AWS CLI ou outros clientes baseados em API, adicione tags usando o parâmetro `--tags` no comando [create-application-version](#).

```
$ aws elasticbeanstalk create-application-version \
 --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
 --application-name my-app --version-label v1
```


Quando você usa a CLI do EB para criar ou atualizar um ambiente, uma versão do aplicativo é criada a partir do código implementado. Não há uma maneira direta de atribuir uma tag a uma versão do aplicativo durante sua criação por meio da CLI do EB. Consulte a seção a seguir para saber como adicionar tags a uma versão do aplicativo existente.

### Gerenciar tags de uma versão do aplicativo existente

É possível adicionar, atualizar e excluir tags em uma versão existente da aplicação do Elastic Beanstalk.

Como gerenciar as tags de uma versão da aplicação usando o console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

 Note

Se você tiver muitos aplicativos, use a barra de pesquisa para filtrar a lista de aplicativos.

3. No painel de navegação, encontre o nome do aplicativo e escolha Application versions (Versões do aplicativo).
4. Selecione a versão do aplicativo que você deseja gerenciar.
5. Escolha Actions (Ações) e Manage tags (Gerenciar tags).
6. Use o formulário na tela para adicionar, atualizar ou excluir tags.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Se você usar a CLI do EB para atualizar a versão do aplicativo, use [eb tags](#) para adicionar, atualizar, excluir ou listar tags.

Por exemplo, o comando a seguir lista as tags em uma versão do aplicativo.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-version"
```

O comando a seguir atualiza a tag mytag1 e exclui a tag mytag2.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
 --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-version"
```

Para obter uma lista de opções e mais exemplos, consulte [eb tags](#).

Com a AWS CLI ou outros clientes baseados em API, use o comando [list-tags-for-resource](#) para listar as tags de uma versão da aplicação.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
"arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-
version"
```

Use o comando [update-tags-for-resource](#) para adicionar, atualizar ou excluir tags em uma versão do aplicativo.

```
$ aws elasticbeanstalk update-tags-for-resource \
--tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
--resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-
id:applicationversion/my-app/my-version"
```

Especifique as tags a serem adicionadas e atualizadas no parâmetro `--tags-to-add` do `update-tags-for-resource`. Uma tag não existente é adicionada, e o valor de uma tag existente é atualizado.

#### Note

Para usar alguns dos comandos da EB CLI e da AWS CLI com uma versão da aplicação do Elastic Beanstalk, você precisa do ARN da versão da aplicação. Você pode recuperar o ARN usando o comando a seguir.

```
$ aws elasticbeanstalk describe-application-versions --application-name my-app
--version-label my-version
```

## Criar um pacote de origem do aplicativo

Quando você usa o console do AWS Elastic Beanstalk para implantar uma nova aplicação ou uma versão da aplicação, é preciso fazer upload de um pacote de origem. O pacote de origem deve cumprir os seguintes requisitos:

- Consistir em um único arquivo ZIP ou WAR (você pode incluir vários arquivos WAR em seu arquivo ZIP)
- Não exceder 500 MB
- Não incluir uma pasta pai ou o diretório de nível superior (subdiretórios podem ser usados)

Se você deseja implantar um aplicativo do operador que processa tarefas periódicas em segundo plano, o pacote de origem do aplicativo deverá incluir também um arquivo `cron.yaml`. Para obter mais informações, consulte [Tarefas periódicas](#).

Se você estiver implantando sua aplicação com a Elastic Beanstalk Command Line Interface (EB CLI), o AWS Toolkit for Eclipse ou o AWS Toolkit for Visual Studio, o arquivo ZIP ou WAR será automaticamente estruturado de modo correto. Para obter mais informações, consulte [Usar a Interface de Linhas de Comando do Elastic Beanstalk \(CLI do EB\)](#), [Criar e implantar aplicações Java no Elastic Beanstalk](#) e [A AWS Toolkit for Visual Studio](#).

## Seções

- [Criar pacote de origem a partir da linha de comando](#)
- [Criar pacote de origem com o Git](#)
- [Compactar arquivos no Mac OS X Finder ou Windows Explorer](#)
- [Criar pacote de origem para um aplicativo .NET](#)
- [Testar pacote de origem](#)

## Criar pacote de origem a partir da linha de comando

Crie um pacote de origem usando o comando `zip`. Para incluir arquivos e pastas ocultos, use um padrão, como os seguintes.

```
~/myapp$ zip ../myapp.zip -r * .[^.]*
 adding: app.js (deflated 63%)
 adding: index.js (deflated 44%)
 adding: manual.js (deflated 64%)
 adding: package.json (deflated 40%)
 adding: restify.js (deflated 85%)
 adding: .ebextensions/ (stored 0%)
 adding: .ebextensions/xray.config (stored 0%)
```

Isso garante que os [arquivos de configuração](#) do Elastic Beanstalk e outros arquivos e pastas que começam com um ponto sejam incluídos no arquivo.

Para aplicativos web Tomcat, use `jar` para criar um arquivamento web.

```
~/myapp$ jar -cvf myapp.war .
```

Os comandos acima incluem arquivos ocultos que podem aumentar o tamanho do pacote de origem desnecessariamente. Para obter mais controle, use um padrão de arquivo mais detalhado ou [crie seu pacote de origem com o Git](#).

## Criar pacote de origem com o Git

Se você estiver usando o Git para gerenciar o código-fonte do aplicativo, use o comando `git archive` para criar o pacote de origem.

```
$ git archive -v -o myapp.zip --format=zip HEAD
```

`git archive` inclui apenas os arquivos que são armazenados em git e exclui arquivos ignorados e git. Isso ajuda a manter o pacote de origem o menor possível. Para obter mais informações, acesse a [página do manual git-archive](#).

## Compactar arquivos no Mac OS X Finder ou Windows Explorer

Quando você criar um arquivo ZIP no Mac OS X Finder ou Windows Explorer, certifique-se de compactar os arquivos e subpastas, em vez da pasta pai.

### Note

A interface gráfica de usuário (GUI) nos sistemas operacionais baseados em Mac OS X e Linux não exibe os arquivos e pastas com nomes que começam com um ponto (.). Use a linha de comando em vez da GUI para compactar o aplicativo se o arquivo ZIP deve incluir uma pasta oculta, como `.ebextensions`. Para os procedimentos de linha de comando para criar um arquivo ZIP no sistema operacional baseado em Mac OS X ou Linux, consulte [Criar pacote de origem a partir da linha de comando](#).

### Example

Suponhamos que você tenha uma pasta de projeto Python chamada `myapp`, que inclui os seguintes arquivos e subpastas:

```
myapplication.py
README.md
static/
```



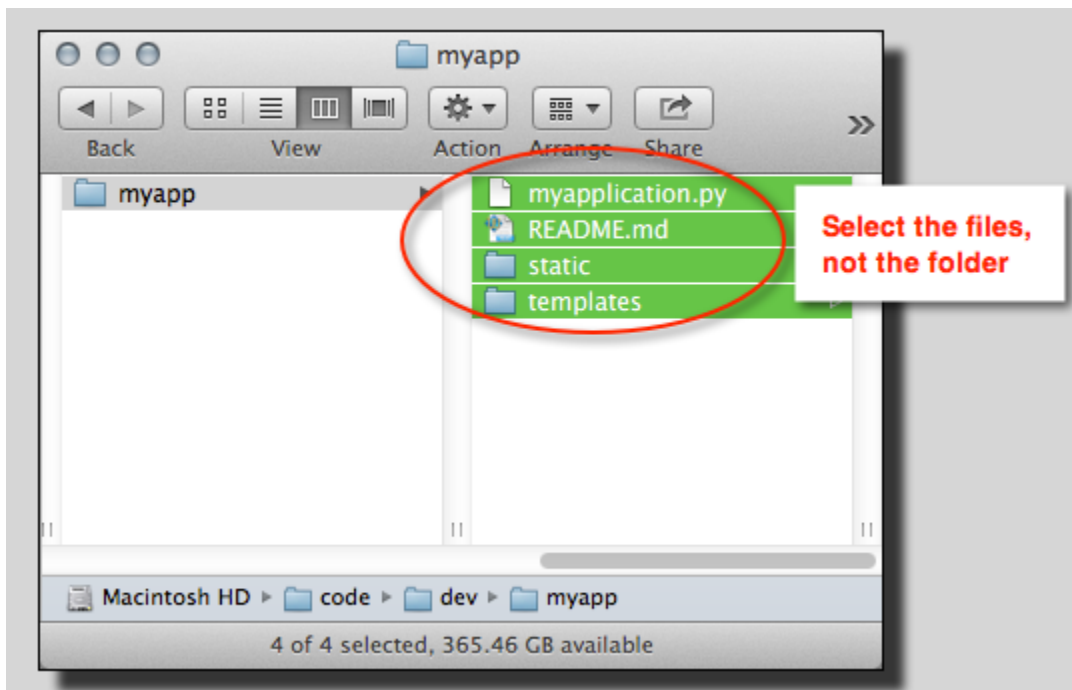
```
static/css
static/css/styles.css
static/img
static/img/favicon.ico
static/img/logo.png
templates/
templates/base.html
templates/index.html
```

Como observado na lista de requisitos acima, seu pacote de origem deve ser compactado sem uma pasta pai, para que a estrutura descompactada não inclua um diretório de nível superior extra. Neste exemplo, nenhuma pasta `myapp` deverá ser criada quando os arquivos forem descompactados (ou, na linha de comando, nenhum segmento `myapp` deverá ser adicionado aos caminhos de arquivo).

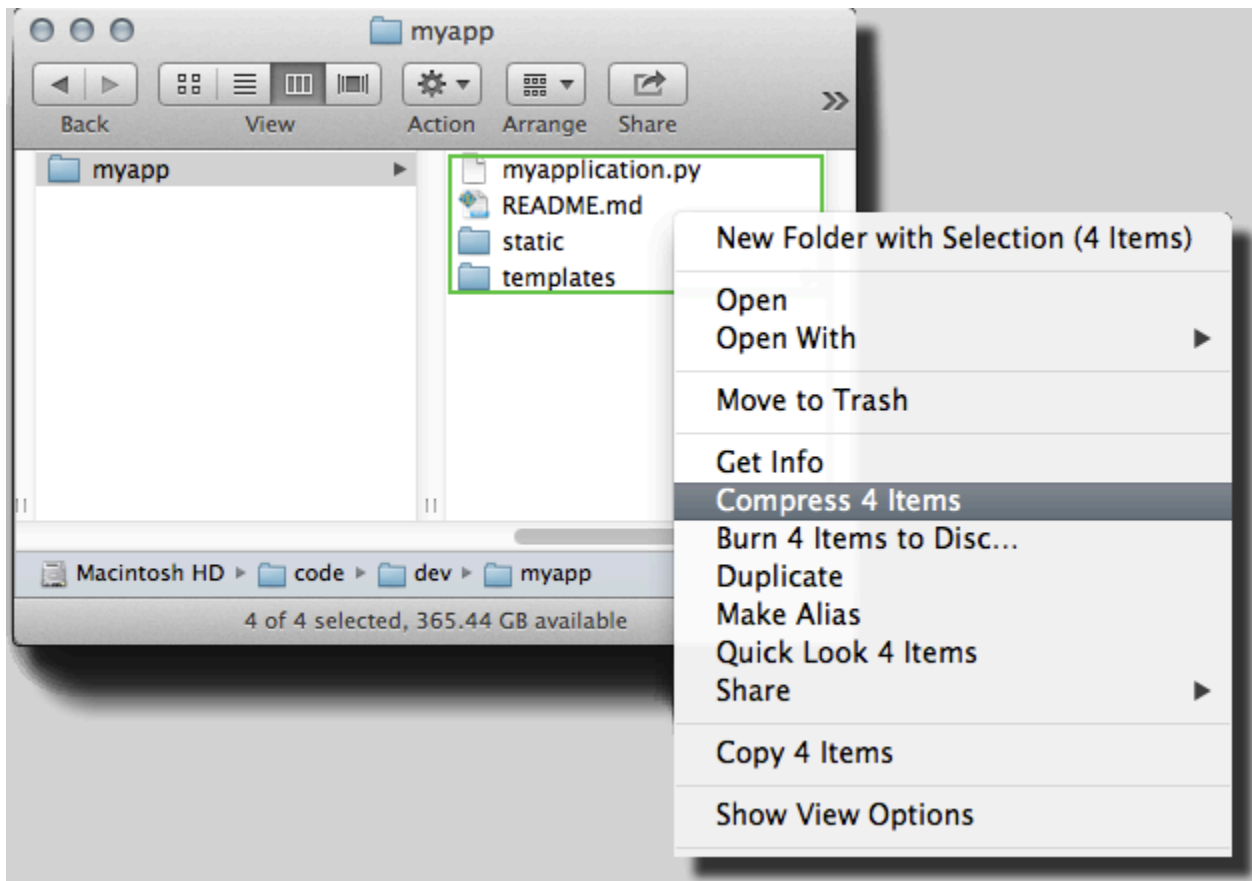
Este exemplo de estrutura de arquivos é usado em todo este tópico para ilustrar como compactar arquivos.

Para compactar arquivos no Mac OS X Finder

1. Abra a pasta de projeto de nível superior e selecione todos os arquivos e subpastas contidos nela. Não selecione a própria pasta de nível superior.

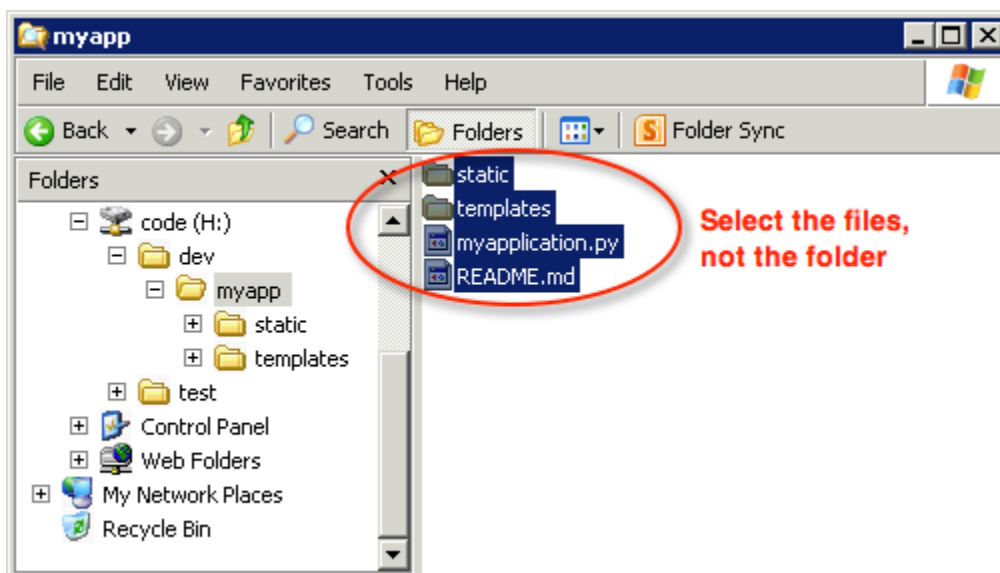


2. Clique com o botão direito do mouse nos arquivos selecionados e escolha Comprimir X itens, em que X é o número de arquivos e subpastas que você selecionou.

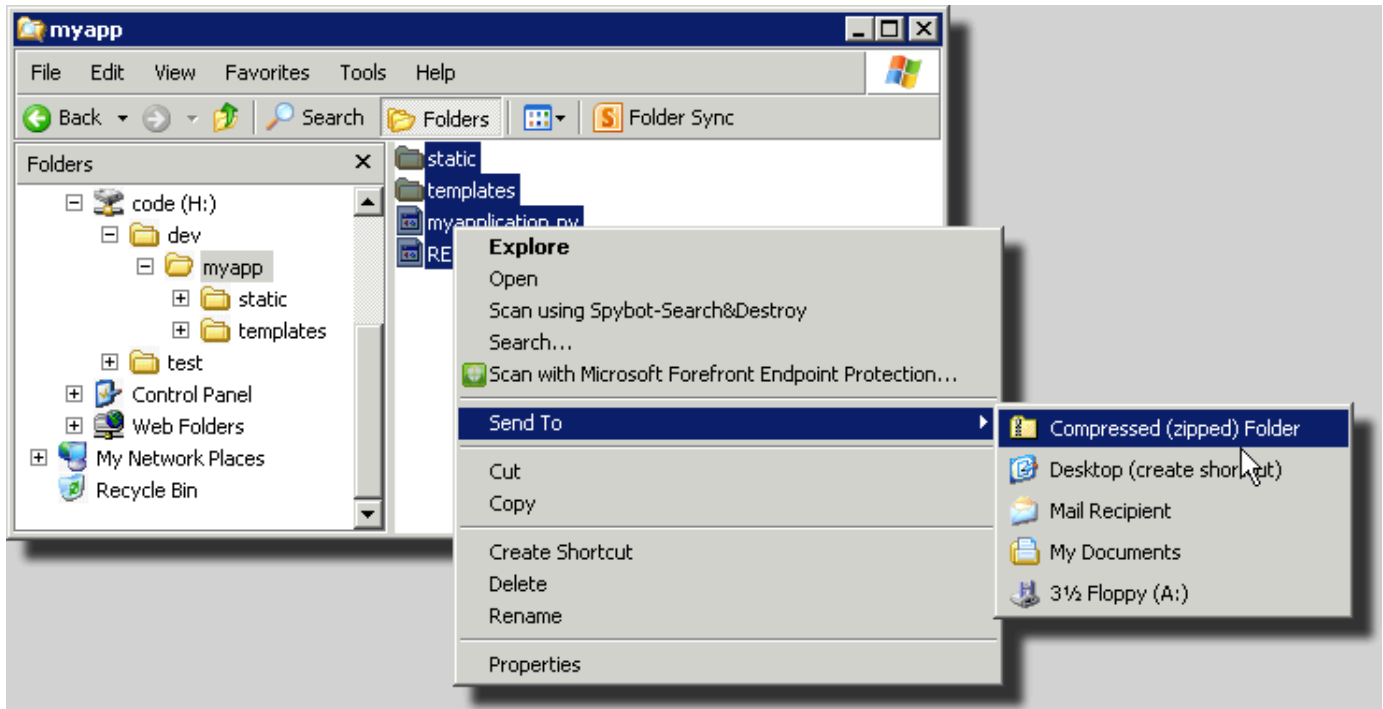


## Como compactar arquivos no Windows Explorer

1. Abra a pasta de projeto de nível superior e selecione todos os arquivos e subpastas contidos nela. Não selecione a própria pasta de nível superior.



2. Clique com o botão direito do mouse nos arquivos selecionados, escolha Enviar para e Pasta compactada (zipada).



## Criar pacote de origem para um aplicativo .NET

Se você usar o Visual Studio, poderá usar a ferramenta de implantação incluída no AWS Toolkit for Visual Studio para implantar a aplicação .NET no Elastic Beanstalk. Para obter mais informações, consulte [Implantar aplicações do Elastic Beanstalk em .NET usando a ferramenta de implantação](#).

Se você precisa criar manualmente um pacote de origem para o seu aplicativo .NET, não pode simplesmente criar um arquivo ZIP que contenha o diretório do projeto. É necessário criar um pacote de implantação da Web para seu projeto que seja adequado à implantação no Elastic Beanstalk. Há vários métodos que podem ser usados para criar um pacote de implantação:

- Crie o pacote de implantação usando o assistente Publicar na Web no Visual Studio. Para obter mais informações, acesse [Como criar um pacote de implantação Web no Visual Studio](#).

### Important

Ao criar o pacote de implantação da Web, inicie o Nome do site com Default Web Site.

- Se tiver um projeto .NET, você poderá criar o pacote de implantação usando o comando msbuild, conforme exibido no exemplo a seguir.

#### Important

O parâmetro DeployIisAppPath deve começar com Default Web Site.

```
C:/> msbuild <web_app>.csproj /t:Package /p:DeployIisAppPath="Default Web Site"
```

- Se você tem um projeto de site, pode usar a ferramenta de implantação da Web do IIS para criar o pacote de implantação. Para obter mais informações, acesse [Packaging and Restoring a Web site](#).

#### Important

O parâmetro apphostconfig deve começar com Default Web Site.

Se estiver implantando vários aplicativos ou um aplicativo ASP.NET Core, coloque sua .ebextensions pasta na raiz do pacote de origem, lado a lado com os pacotes do aplicativo e arquivo manifesto:

```
~/workspace/source-bundle/
|-- .ebextensions
| |-- environmentvariables.config
| |-- healthcheckurl.config
|-- AspNetCore101HelloWorld.zip
|-- AspNetCoreHelloWorld.zip
|-- aws-windows-deployment-manifest.json
|-- VS2015AspNetWebApiApp.zip
```

## Testar pacote de origem

Convém testar seu pacote de origem localmente antes de fazer upload para o Elastic Beanstalk. Como o Elastic Beanstalk usa basicamente a linha de comando para extrair os arquivos, é melhor fazer os testes a partir da linha de comando em vez de usar uma ferramenta GUI.

## Para testar a extração de arquivos no Mac OS X ou Linux

1. Abra uma janela do terminal (Mac OS X) ou conecte-se ao servidor Linux. Navegue até o diretório que contém o pacote de origem.
2. Com uso do comando `unzip` ou `tar xf`, descompacte o arquivo.
3. Certifique-se de que os arquivos descompactados aparecem na mesma pasta que o próprio arquivo, não em uma nova pasta ou diretório de nível superior.

### Note

Se você usar o Mac OS X Finder para descompactar o arquivo, será criada uma nova pasta de nível superior, independentemente de como você tenha estruturado o arquivo. Para obter melhores resultados, use a linha de comando.

## Para testar a extração de arquivos no Windows

1. Faça download ou instale um programa que permita extrair arquivos compactados por meio da linha de comando. Por exemplo, faça download do programa `unzip.exe` gratuito em <http://stahlforce.com/dev/index.php?tool=zipunzip>.
2. Se necessário, copie o arquivo executável no diretório que contém o pacote de origem. Se você tiver instalado uma ferramenta em todo o sistema, poderá ignorar esta etapa.
3. Com o uso do comando apropriado, descompacte o arquivo. Se você fez download do `unzip.exe` usando o link da etapa 1, o comando é `unzip <archive-name>`.
4. Certifique-se de que os arquivos descompactados aparecem na mesma pasta que o próprio arquivo, não em uma nova pasta ou diretório de nível superior.

## Marcar recursos da aplicação do Elastic Beanstalk

Você pode aplicar tags aos recursos das aplicações do AWS Elastic Beanstalk. As tags são pares de chave-valor associados a recursos da AWS. As tags podem ajudar você a classificar recursos. Elas são particularmente úteis se você gerencia muitos recursos como parte de várias aplicações da AWS.

Veja a seguir algumas maneiras de usar tags com recursos do Elastic Beanstalk:

- Estágios de implantação: identifique recursos associados a diferentes estágios da aplicação, como desenvolvimento, beta e produção.
- Alocação de custos: use relatórios de alocação de custos para acompanhar o uso de recursos da AWS associados a várias contas de despesas. Os relatórios incluem recursos marcados e não marcados e agregam os custos de acordo com as tags. Para obter informações sobre como os relatórios de alocação de custo usam etiquetas, consulte [Usar etiquetas de alocação de custos para relatórios de faturamento personalizados](#) no Guia do usuário do AWS Billing and Cost Management.
- Controle de acesso: use etiquetas para gerenciar permissões para solicitações e recursos. Por exemplo, um usuário que só pode criar e gerenciar ambientes beta deve ter acesso somente aos recursos do estágio beta. Para obter mais detalhes, consulte [Usar tags para controlar o acesso aos recursos do Elastic Beanstalk](#).

Você pode adicionar até 50 tags a cada recurso da . Os ambientes são um pouco diferentes: o Elastic Beanstalk adiciona três tags de sistema padrão aos ambientes, e não é possível editar nem excluir essas tags. Além das tags padrão, você pode adicionar até 47 tags adicionais a cada ambiente.

As seguintes restrições se aplicam a chaves e valores de tags:

- As chaves e os valores podem conter letras, números, espaço em branco e os seguintes símbolos: `_ . : / = + - @`
- As chaves podem conter até 127 caracteres. Os valores podem conter até 255 caracteres.

#### Note

Esses limites de tamanho são para caracteres Unicode em UTF-8. Para outras codificações multibyte, os limites podem ser menores.

- As chaves fazem distinção entre maiúsculas e minúsculas.
- As chaves de tag não podem começar com `aws:` ou `elasticbeanstalk:`.

## Propagação de tags para modelos de execução

O Elastic Beanstalk fornece uma opção para permitir a propagação de tags de ambiente para modelos de execução. Essa opção fornece suporte contínuo ao controle de acesso baseado em tags (TBAC) com modelos de execução.

### Note

As configurações de lançamento estão sendo descontinuadas e substituídas por modelos de execução. Para obter mais informações, consulte [Configurações de execução](#), no Guia do usuário do Amazon EC2 Auto Scaling.

Para evitar o tempo de inatividade da execução de instâncias do EC2, o AWS CloudFormation não propaga tags para modelos de execução existentes. Se houver um caso de uso que exija tags para os recursos do seu ambiente, você poderá habilitar o Elastic Beanstalk para criar modelos de execução com tags para esses recursos. Para fazer isso, defina a opção `LaunchTemplateTagPropagationEnabled` no namespace [aws:autoscaling:launchconfiguration](#) como `true`. O valor padrão é `false`.

O exemplo de [arquivo de configuração](#) a seguir permite a propagação de tags para modelos de execução.

```
option_settings:
 aws:autoscaling:launchconfiguration:
 LaunchTemplateTagPropagationEnabled: true
```

O Elastic Beanstalk só pode propagar tags para modelos de execução para os seguintes recursos:

- Volumes do EBS
- Instâncias do EC2
- Interfaces de rede do EC2
- Modelos de execução do AWS CloudFormation que definem um recurso

Essa restrição existe porque o CloudFormation só permite tags na criação de modelos para recursos específicos. Para obter mais informações, consulte [TagSpecification](#) no Guia do usuário do AWS CloudFormation.

### Important

- Alterar o valor dessa opção de `false` para `true` para um ambiente existente pode ser uma alteração significativa para tags já existentes.
- Quando esse recurso estiver habilitado, a propagação de tags exigirá a substituição do EC2, o que pode resultar em tempo de inatividade. É possível habilitar atualizações contínuas para aplicar alterações de configuração em lotes e evitar tempo de inatividade durante o processo de atualização. Para obter mais informações, consulte [Alterações de configuração](#).

Para obter mais informações sobre modelos de execução, consulte:

- [Modelos de execução](#) no Guia do usuário do Amazon EC2 Auto Scaling
- [Trabalhar com modelos](#) no Guia do usuário do AWS CloudFormation
- [Trechos de modelos do Elastic Beanstalk](#) no Guia do usuário do AWS CloudFormation

## Recursos que podem ser marcados

Veja a seguir os tipos de recursos do Elastic Beanstalk que podem ser marcados com tags, bem como os links para tópicos específicos sobre gerenciamento de tags para cada um deles:

- [Aplicativos](#)
- [Ambientes do](#)
- [Versões do aplicativo](#)
- [Configurações salvas](#)
- [Versões de plataforma personalizadas](#)

## Marcar aplicativos

Você pode aplicar tags as aplicações do AWS Elastic Beanstalk. As tags são pares de chave-valor associados a recursos da AWS. Para obter informações sobre a atribuição de tags do recurso do Elastic Beanstalk, casos de uso, restrições de chave e valor de tag, além de tipos de recursos compatíveis, consulte [Marcar recursos da aplicação do Elastic Beanstalk](#).



Você pode especificar tags quando cria um aplicativo. Em um aplicativo existente, você pode adicionar ou remover tags e atualizar os valores de tags existentes. Você pode adicionar até 50 tags para cada aplicativo.

## Adicionar tags durante a criação do aplicativo

Ao usar o console do Elastic Beanstalk para [criar uma aplicação](#), é possível especificar chaves e valores de tags na caixa de diálogo Create New Application (Criar nova aplicação).

Elastic Beanstalk

### Create new application

**Application information**

Application Name

Maximum length of 100 characters, not including forward slash (/).

Description

**Tags**

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

Key	Value	
<input type="text"/>	<input type="text"/>	<input type="button" value="Remove tag"/>

50 remaining

Se você usar a CLI do EB para criar um aplicativo, use a opção `--tags` com [eb init](#) para adicionar tags.

```
~/workspace/my-app$ eb init --tags mytag1=value1,mytag2=value2
```

Com a AWS CLI ou outros clientes baseados em API, adicione tags usando o parâmetro `--tags` no comando [create-application](#).

```
$ aws elasticbeanstalk create-application \
 --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
 --application-name my-app --version-label v1
```

## Gerenciar tags de um aplicativo existente

É possível adicionar, atualizar e excluir tags em uma aplicação existente do Elastic Beanstalk.

Como gerenciar as tags de uma aplicação no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

### Note

Se você tiver muitos aplicativos, use a barra de pesquisa para filtrar a lista de aplicativos.

3. Escolha Actions (Ações) e Manage tags (Gerenciar tags).
4. Use o formulário na tela para adicionar, atualizar ou excluir tags.
5. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Se você usar a CLI do EB para atualizar o aplicativo, use [eb tags](#) para adicionar, atualizar, excluir ou listar tags.

Por exemplo, o comando a seguir lista as tags em um aplicativo.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-
account-id:application/my-app"
```

O comando a seguir atualiza a tag `mytag1` e exclui a tag `mytag2`.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
 \
```

```
--resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Para obter uma lista de opções e mais exemplos, consulte [eb tags](#).

Com a AWS CLI ou outros clientes baseados em API, use o comando [list-tags-for-resource](#) para listar as tags de uma aplicação.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
"arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Use o comando [update-tags-for-resource](#) para adicionar, atualizar ou excluir tags em uma aplicação.

```
$ aws elasticbeanstalk update-tags-for-resource \
--tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
--resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-
app"
```

Especifique as tags a serem adicionadas e atualizadas no parâmetro `--tags-to-add` do `update-tags-for-resource`. Uma tag não existente é adicionada, e o valor de uma tag existente é atualizado.

#### Note

Para usar alguns dos comandos da EB CLI e da AWS CLI com uma aplicação do Elastic Beanstalk, você precisa do ARN da aplicação. Você pode recuperar o ARN usando o comando a seguir.

```
$ aws elasticbeanstalk describe-applications --application-names my-app
```

# Gerenciar ambientes

O AWS Elastic Beanstalk facilita a criação de novos ambientes para sua aplicação. Você pode criar e gerenciar ambientes separados para uso em desenvolvimento, teste e produção, e pode [implantar qualquer versão](#) do seu aplicativo em qualquer ambiente. Os ambientes podem ser de execução prolongada ou temporários. Quando você encerra um ambiente, pode salvar sua configuração para recriá-lo posteriormente.

À medida que desenvolve seu aplicativo, você o implanta com frequência, possivelmente em muitos ambientes diferentes para diversas finalidades. O Elastic Beanstalk permite [configurar como as implantações são executadas](#). Você pode fazer a implantação em todas as instâncias do seu ambiente simultaneamente ou dividi-la em lotes com implantações contínuas.

As [alterações de configuração](#) são processados separadamente das implantações e têm escopo próprio. Por exemplo, se você alterar o tipo das instâncias do EC2 que executam seu aplicativo, todas as instâncias deverão ser substituídas. Por outro lado, se você modificar a configuração do load balancer do ambiente, essa alteração poderá ser feita no local, sem interromper o serviço nem reduzir a capacidade. Você também pode aplicar alterações de configuração em lotes, que modificam as instâncias em seu ambiente com [atualizações de configuração contínuas](#).

## Note

Modifique os recursos no ambiente usando somente o Elastic Beanstalk. Se você modificar os recursos usando o console de outro serviço, os comandos da CLI ou os SDKs, o Elastic Beanstalk não poderá monitorar com precisão o estado desses recursos, e você não poderá salvar a configuração nem recriar o ambiente de forma confiável. Alterações fora de banda também podem causar problemas ao atualizar ou encerrar um ambiente.

Ao iniciar um ambiente, você escolhe uma versão de plataforma. Nós atualizamos as plataformas periodicamente com novas versões de plataforma para aprimorar o desempenho e lançar novos recursos. Você pode [atualizar seu ambiente para a versão de plataforma mais recente](#) a qualquer momento.

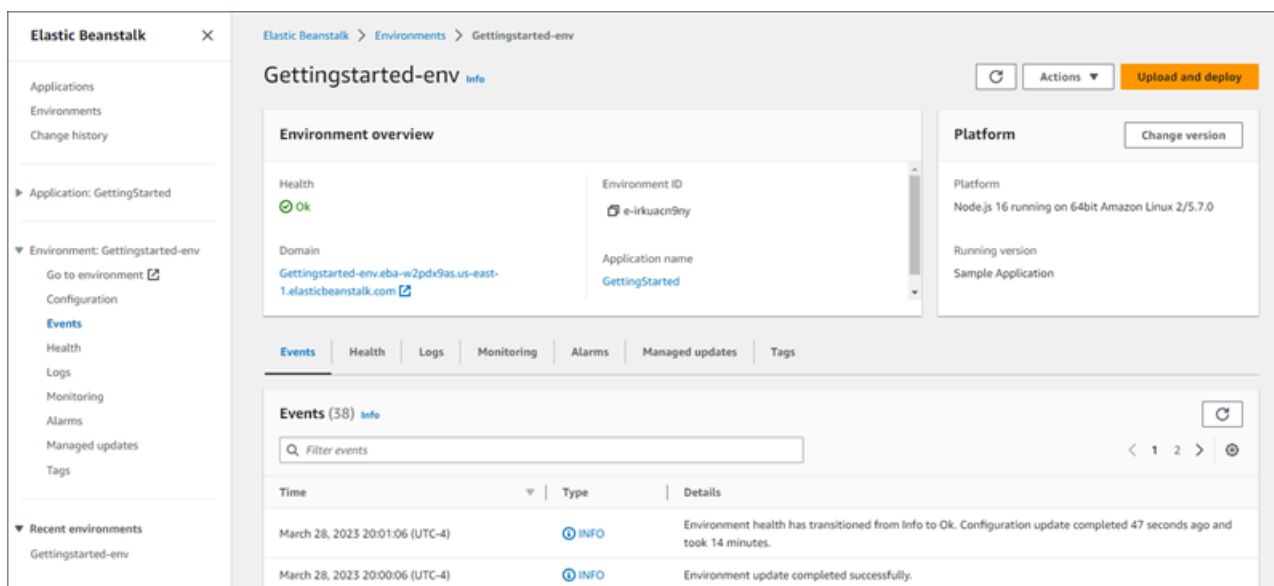
À medida que a complexidade de seu aplicativo aumenta, você pode dividi-lo em vários componentes, sendo cada um executado em um ambiente separado. Para cargas de trabalho de longa duração, é possível executar [ambientes de operador](#) que processam trabalhos a partir de uma fila do Amazon Simple Queue Service (Amazon SQS).

## Tópicos

- [Usar o console de gerenciamento de ambiente do Elastic Beanstalk](#)
- [Criar um ambiente do Elastic Beanstalk](#)
- [Implantar aplicações em ambientes do Elastic Beanstalk](#)
- [Alterações de configuração](#)
- [Atualizar a versão de plataforma do ambiente Elastic Beanstalk](#)
- [Cancelar atualizações de configuração do ambiente e implantações de aplicativos](#)
- [Recriar ambientes do Elastic Beanstalk](#)
- [Tipos de ambiente](#)
- [Ambientes de operador do Elastic Beanstalk](#)
- [Criar links entre ambientes do Elastic Beanstalk](#)

## Usar o console de gerenciamento de ambiente do Elastic Beanstalk

O console do Elastic Beanstalk fornece uma página Environment overview (Visão geral do ambiente) para você gerenciar cada um dos ambientes do AWS Elastic Beanstalk. Em uma página Environment overview (Visão geral do ambiente), você pode gerenciar a configuração do ambiente e realiza ações comuns. Essas ações incluem reiniciar os servidores Web em execução no seu ambiente, clonar seu ambiente ou reconstruir seu ambiente a partir do zero.



The screenshot displays the AWS Elastic Beanstalk console interface. On the left, a navigation sidebar shows 'Applications', 'Environments', and 'Change history'. The main content area is titled 'Gettingstarted-env' and includes an 'Info' icon, a refresh button, and an 'Actions' dropdown menu with an 'Upload and deploy' button. Below this, the 'Environment overview' section shows a 'Health' status of 'Ok' (green checkmark), 'Environment ID' 'e-irkuacn9ny', 'Domain' 'Gettingstarted-env.eba-w2pdx9as.us-east-1.elasticbeanstalk.com', and 'Application name' 'GettingStarted'. To the right, the 'Platform' section shows 'Node.js 16 running on 64bit Amazon Linux 2/5.7.0' and 'Running version' 'Sample Application'. Below the overview, a tabbed interface includes 'Events (38)', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags'. The 'Events' tab is active, showing a search bar and a table of events:

Time	Type	Details
March 28, 2023 20:01:06 (UTC-4)	INFO	Environment health has transitioned from Info to Ok. Configuration update completed 47 seconds ago and took 14 minutes.
March 28, 2023 20:00:06 (UTC-4)	INFO	Environment update completed successfully.

## Como acessar o console de gerenciamento do ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

A página Environment overview (Visão geral do ambiente) é exibida. O painel de navegação do console exibe o nome da aplicação à qual o ambiente pertence, com páginas de gerenciamento de aplicações relacionadas e o nome do ambiente, com páginas de gerenciamento de ambiente.

### Tópicos

- [Visão geral do ambiente](#)
- [Ações do ambiente](#)
- [Eventos](#)
- [Integridade](#)
- [Logs](#)
- [Monitorar](#)
- [alarmes](#)
- [Atualizações gerenciadas](#)
- [Tags](#)
- [Configuração](#)

## Visão geral do ambiente

Para visualizar a página Environment overview (Visão geral do ambiente), escolha o nome do ambiente no painel de navegação, se for o ambiente atual. Ou então, navegue até o ambiente a partir da página Applications (Aplicações) ou da lista principal de ambientes na página Environments (Ambientes).

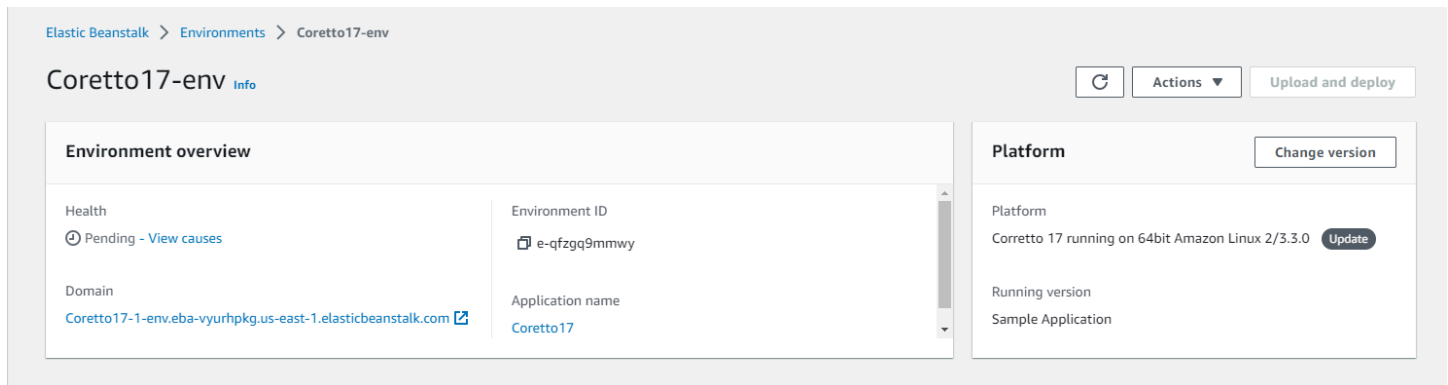
O painel superior na página de visão geral do ambiente exibe informações de nível superior sobre seu ambiente. Isso inclui o nome, o URL, o status de integridade atual, o nome da versão da

aplicação implantada no momento e a versão da plataforma na qual a aplicação é executada. Você pode ver os cinco eventos mais recentes do ambiente abaixo do painel de visão geral.

Clique em Refresh (Atualizar) para atualizar as informações mostradas. A página de visão geral contém as seguintes informações e opções.

## Integridade

A integridade geral do ambiente. Se a integridade do seu ambiente piorar, o link Exibir causas será exibido ao lado da integridade do ambiente. Selecione este link para ver a guia Integridade com mais detalhes.



The screenshot shows the AWS Elastic Beanstalk console interface for an environment named 'Coretto17-env'. The breadcrumb navigation at the top reads 'Elastic Beanstalk > Environments > Coretto17-env'. Below the environment name, there are buttons for 'Refresh', 'Actions', and 'Upload and deploy'. The main content is divided into two panels: 'Environment overview' and 'Platform'. The 'Environment overview' panel shows 'Health' as 'Pending - View causes', 'Environment ID' as 'e-qfzq9mmwy', and 'Domain' as 'Coretto17-1-env.eba-vyurhpkg.us-east-1.elasticbeanstalk.com'. The 'Platform' panel shows 'Platform' as 'Corretto 17 running on 64bit Amazon Linux 2/3.3.0' with an 'Update' button, and 'Running version' as 'Sample Application'.

## Domínio

O Domain (Domínio), ou URL, do ambiente encontra-se na parte superior da página Environment overview (Visão geral do ambiente), abaixo de Health (Integridade) do ambiente. Este é o URL do aplicativo web no qual o ambiente é executado.

## ID do ambiente

O ID do ambiente. Esse é um ID interno que é gerado quando o ambiente é criado.

## Application name

O nome da versão da aplicação que está implantada e em execução no ambiente.

## Versão em execução

O nome da versão do aplicativo que está implantado e em execução em seu ambiente. Escolha Upload and deploy (Fazer upload e implantar) para fazer upload de um [pacote de origem](#) e implantá-lo em seu ambiente. Essa opção cria uma nova versão do aplicativo.

## Plataforma

O nome da versão da plataforma em execução no seu ambiente. Normalmente, isso inclui a arquitetura, o sistema operacional (SO), a linguagem e o servidor de aplicações (coletivamente conhecidos como ramificação da plataforma), com um número de versão específico da plataforma.

Se a versão da plataforma não for a mais recente disponível, um rótulo de status será exibida ao lado dela na seção Plataforma. O rótulo Atualizar indica que, embora a versão da plataforma seja compatível, uma versão mais recente está disponível. A versão da plataforma também pode ser rotulada como Obsoleta ou Descontinuada. Selecione Alterar versão para atualizar sua ramificação da plataforma para uma versão mais recente. Para obter mais informações sobre os estados de uma versão da plataforma, consulte a seção Ramificação da plataforma no [Glossário de plataformas do Elastic Beanstalk](#).

The screenshot shows the AWS Elastic Beanstalk console interface for an environment named 'Coretto17-env'. The breadcrumb navigation at the top reads 'Elastic Beanstalk > Environments > Coretto17-env'. Below the environment name, there are buttons for 'Refresh', 'Actions', and 'Upload and deploy'. The main content is divided into two panels. The left panel, titled 'Environment overview', displays the following information: Health status is 'Ok' (indicated by a green checkmark), Environment ID is 'e-qfzq9mmwy', and Domain is 'Coretto17-1-env.eba-vyurhpkg.us-east-1.elasticbeanstalk.com'. The right panel, titled 'Platform', shows the current platform as 'Corretto 17 running on 64bit Amazon Linux 2/3.3.0' with an 'Update' button next to it. Below this, it lists the 'Running version' as 'Sample Application'.

### Guias de Environment overview (Visão geral do ambiente)

As guias exibidas na metade inferior da página contém informações mais detalhadas sobre o ambiente e fornecem acesso a recursos adicionais:

- **Events (Eventos):** mostra informações ou mensagens de erro do serviço do Elastic Beanstalk e de outros serviços cujos recursos são utilizados por esse ambiente.
- **Health (Integridade):** mostra o status e as informações detalhadas de integridade sobre as instâncias do Amazon EC2 que executam a aplicação.
- **Logs:** recupere e baixe os logs do Amazon EC2 no seu ambiente. Você pode recuperar logs completos ou as atividades recentes. Os logs recuperados ficam disponíveis por 15 minutos.
- **Monitoring (Monitoramento):** mostra as estatísticas do ambiente, por exemplo, a latência média e a utilização de CPU.
- **Alarms (Alarmes):** mostra os alarmes que você configurou para as métricas do ambiente. Você pode adicionar, modificar ou excluir alarmes nessa página.



- **Managed Updates (Atualizações gerenciadas):** mostra informações sobre as atualizações gerenciadas da plataforma e das substituições de instâncias já concluídas ou que ocorrerão em breve.
- **Tags:** mostra as tags de ambiente e permite que você as gerencie. Tags são pares de chave-valor que são aplicadas ao seu ambiente.

#### Note

O painel de navegação no lado esquerdo do console lista links com o mesmo nome das guias. Selecionar qualquer desses links exibirá o conteúdo da guia correspondente.

## Ações do ambiente

A página de visão geral do ambiente contém um menu Actions (Ações) que pode ser usada para executar operações comuns no seu ambiente. Esse menu é exibido no lado direito do cabeçalho do ambiente, ao lado da opção Create New Environment (Criar novo ambiente).

#### Note

Algumas ações só estão disponíveis sob certas condições, permanecendo desativadas até que as condições certas sejam atendidas.

## Carregar configuração

Carregue uma configuração salva anteriormente. As configurações são salvas no seu aplicativo e podem ser carregadas por qualquer ambiente associado. Se você fizer alterações na configuração do seu ambiente, poderá carregar uma configuração salva para desfazê-las. Você também pode carregar uma configuração que você salvou de um ambiente diferente que esteja executando o mesmo aplicativo para propagar as alterações de configuração entre eles.

## Salvar configuração

Salve a configuração atual do seu ambiente no seu aplicativo. Antes de fazer alterações na configuração do ambiente, salve a configuração atual para que você possa reverter posteriormente, se necessário. Você também pode aplicar uma configuração salva ao iniciar um novo ambiente.

## Swap Environment Domains (URLs) (Trocar domínios [URL] do ambiente)

Troque o CNAME do ambiente atual por um novo ambiente. Depois de uma troca de CNAME, todo o tráfego para o aplicativo que usa a URL do ambiente vai para o novo ambiente. Quando você estiver pronto para implantar uma nova versão de seu aplicativo, poderá iniciar um ambiente separado na nova versão. Quando o novo ambiente estiver pronto para começar a receber solicitações, execute uma troca de CNAME para começar a rotear o tráfego para o novo ambiente. Fazer isso não interrompe seus serviços. Para obter mais informações, consulte [Implantações azuis/verdes com o Elastic Beanstalk](#).

## Clonar ambiente

Inicie um novo ambiente com as mesmas configurações que o seu atual ambiente em execução.

## Clonar com a plataforma mais recente

Clone o seu ambiente atual com a versão mais recente da plataforma do Elastic Beanstalk em uso. Essa opção só fica disponível quando uma versão mais recente da plataforma do ambiente atual está disponível para uso.

## Abortar operação atual

Pare uma atualização de ambiente em andamento. Interromper uma operação pode fazer com que algumas das instâncias no seu ambiente fiquem em um estado diferente das outras, dependendo do progresso da operação. Essa opção está disponível apenas quando o ambiente está sendo atualizado.

## Reiniciar servidores de aplicativos

Reinicie o servidor web em execução nas instâncias do seu ambiente. Essa opção não termina nem reinicia qualquer recurso da AWS. Se seu ambiente estiver agindo de modo estranho em resposta a solicitações ruins, reiniciar o servidor de aplicativo pode restaurar temporariamente a funcionalidade, enquanto você soluciona a causa principal do problema.

## Recriar ambiente

Encerre todos os recursos no ambiente em execução e crie um novo ambiente com as mesmas configurações. Essa operação leva vários minutos, semelhante ao tempo necessário para implantar um novo ambiente a partir do zero. Todas as instâncias do Amazon RDS em execução na camada

de dados de seu ambiente são excluídas durante a recriação. Se você precisa dos dados, crie um snapshot. É possível criar um snapshot manualmente [no console do RDS](#) ou configurar política de exclusão da camada de dados para criar um snapshot automaticamente antes de excluir a instância. Essa é a configuração padrão quando você cria uma camada de dados.

## Encerrar ambiente

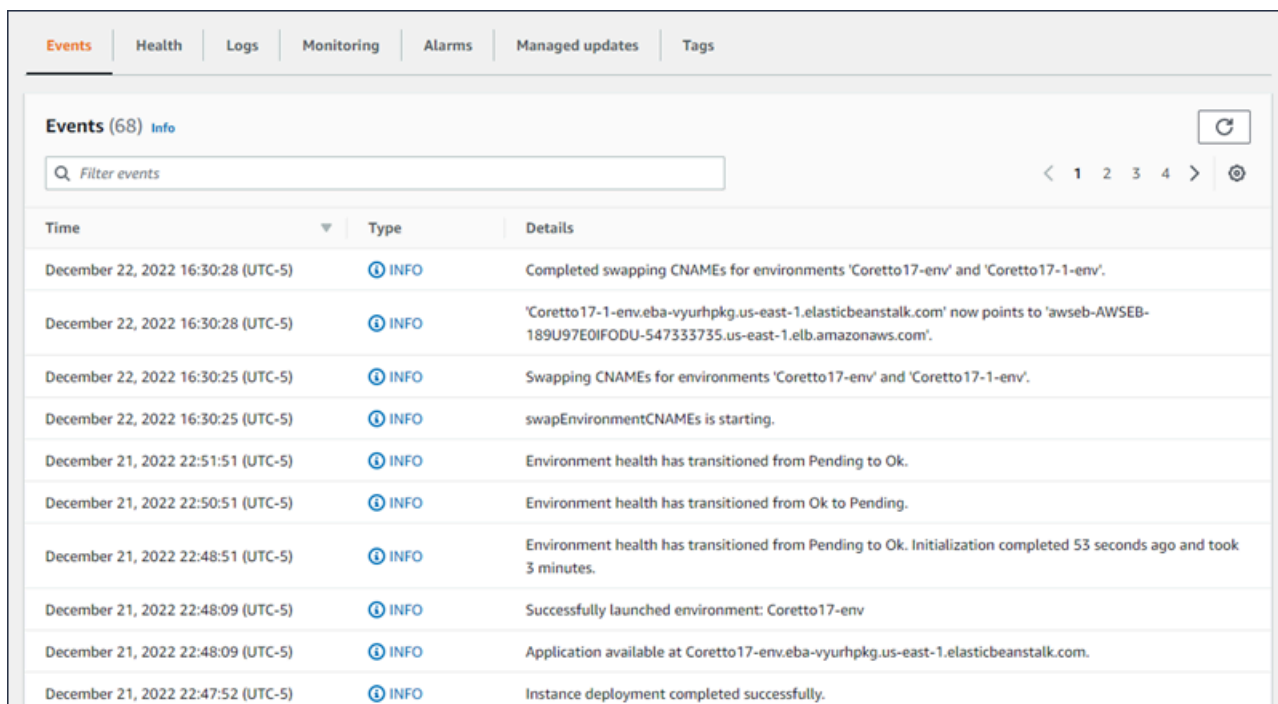
Encerre todos os recursos no ambiente em execução e remova o ambiente da aplicação. Se você tiver uma instância do RDS em execução em uma camada de dados e precisar reter os dados, certifique-se de que a política de exclusão de banco de dados esteja definida como Snapshot ou Retain. Para obter mais informações, consulte [Ciclo de vida do banco de dados](#) no capítulo Configuração de ambientes deste guia.

## Restaurar ambiente

Se o ambiente tiver sido encerrado em até uma hora, você poderá restaurá-lo dessa página. Depois de uma hora, você pode [restaurá-lo a partir da página de visão geral do aplicativo](#).

## Eventos

A página Events (Eventos) exibe o fluxo de eventos do ambiente. O Elastic Beanstalk envia mensagens de evento sempre que você interage com o ambiente e quando recursos do ambiente são criados ou modificados como resultado.



The screenshot shows the AWS Elastic Beanstalk console's 'Events' page. At the top, there are navigation tabs: Events (selected), Health, Logs, Monitoring, Alarms, Managed updates, and Tags. Below the tabs, the page title is 'Events (68) Info' with a refresh icon. A search bar labeled 'Filter events' is present. Below the search bar is a table of events with columns for Time, Type, and Details. The events listed are:

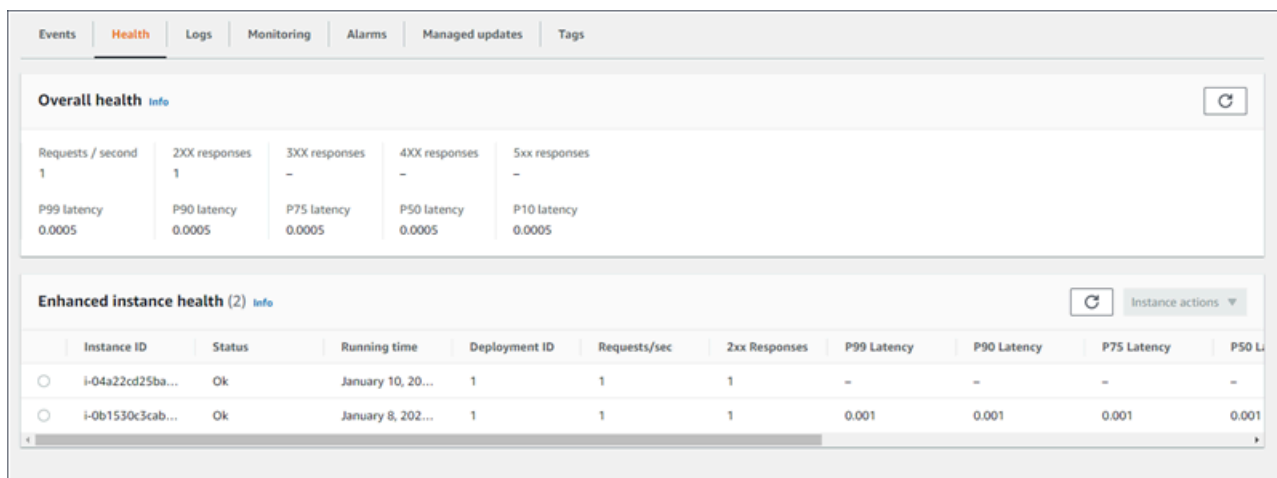
Time	Type	Details
December 22, 2022 16:30:28 (UTC-5)	INFO	Completed swapping CNAMEs for environments 'Coretto17-env' and 'Coretto17-1-env'.
December 22, 2022 16:30:28 (UTC-5)	INFO	'Coretto17-1-env.eba-vyurhpkg.us-east-1.elasticbeanstalk.com' now points to 'awseb-AWSEB-189U97E0IFODU-547333735.us-east-1.elb.amazonaws.com'.
December 22, 2022 16:30:25 (UTC-5)	INFO	Swapping CNAMEs for environments 'Coretto17-env' and 'Coretto17-1-env'.
December 22, 2022 16:30:25 (UTC-5)	INFO	swapEnvironmentCNAMEs is starting.
December 21, 2022 22:51:51 (UTC-5)	INFO	Environment health has transitioned from Pending to Ok.
December 21, 2022 22:50:51 (UTC-5)	INFO	Environment health has transitioned from Ok to Pending.
December 21, 2022 22:48:51 (UTC-5)	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 53 seconds ago and took 3 minutes.
December 21, 2022 22:48:09 (UTC-5)	INFO	Successfully launched environment: Coretto17-env
December 21, 2022 22:48:09 (UTC-5)	INFO	Application available at Coretto17-env.eba-vyurhpkg.us-east-1.elasticbeanstalk.com.
December 21, 2022 22:47:52 (UTC-5)	INFO	Instance deployment completed successfully.

Para obter mais informações, consulte [Visualizar o stream de eventos de um ambiente do Elastic Beanstalk](#).

## Integridade

Se o monitoramento avançado de integridade estiver habilitado, essa página mostrará informações de integridade em tempo real para as instâncias. O painel Overall health (Integridade geral) mostra os dados de integridade como uma média de todas as instâncias do ambiente combinadas. O painel Enhanced instance health (Integridade aprimorada de instâncias) mostra, em tempo real, informações sobre a integridade de cada instância individual no ambiente. O monitoramento de integridade avançado permite que o Elastic Beanstalk monitore de perto os recursos no ambiente para que possa avaliar a integridade de sua aplicação com mais precisão.

Quando o monitoramento avançado de integridade está ativado, essa página mostra informações sobre as solicitações atendidas pelas instâncias em seu ambiente e as métricas do sistema operacional, incluindo latência, carga e utilização de CPU.



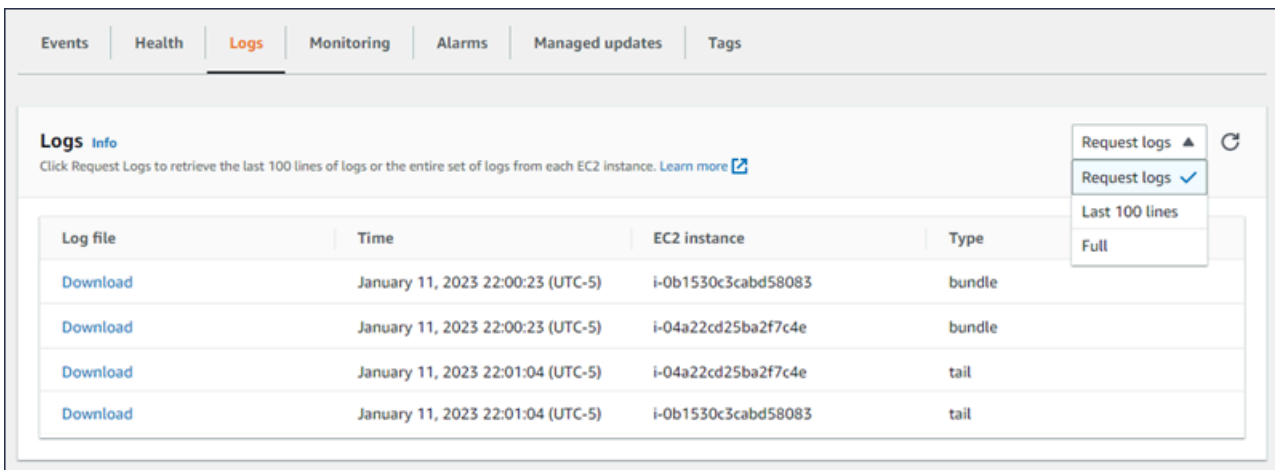
Instance ID	Status	Running time	Deployment ID	Requests/sec	2xx Responses	P99 Latency	P90 Latency	P75 Latency	P50 Latency
i-04a22cd25ba...	Ok	January 10, 20...	1	1	1	-	-	-	-
i-0b1530c3cab...	Ok	January 8, 202...	1	1	1	0.001	0.001	0.001	0.001

Para obter mais informações, consulte [Monitoramento e relatório de integridade aprimorada](#).

## Logs

A página Logs permite recuperar os logs das instâncias do EC2 em seu ambiente. Quando logs são solicitados, o Elastic Beanstalk envia um comando para as instâncias que, depois, faz upload de logs para o bucket de armazenamento do Elastic Beanstalk no Amazon S3. Quando logs são solicitados nesta página, o Elastic Beanstalk os exclui automaticamente do Amazon S3 após 15 minutos.

Também é possível configurar as instâncias do seu ambiente para fazer upload dos logs para o Amazon S3, para armazenamento permanente depois que forem alternadas localmente.



Events | Health | **Logs** | Monitoring | Alarms | Managed updates | Tags

**Logs** Info  
Click Request Logs to retrieve the last 100 lines of logs or the entire set of logs from each EC2 instance. [Learn more](#)

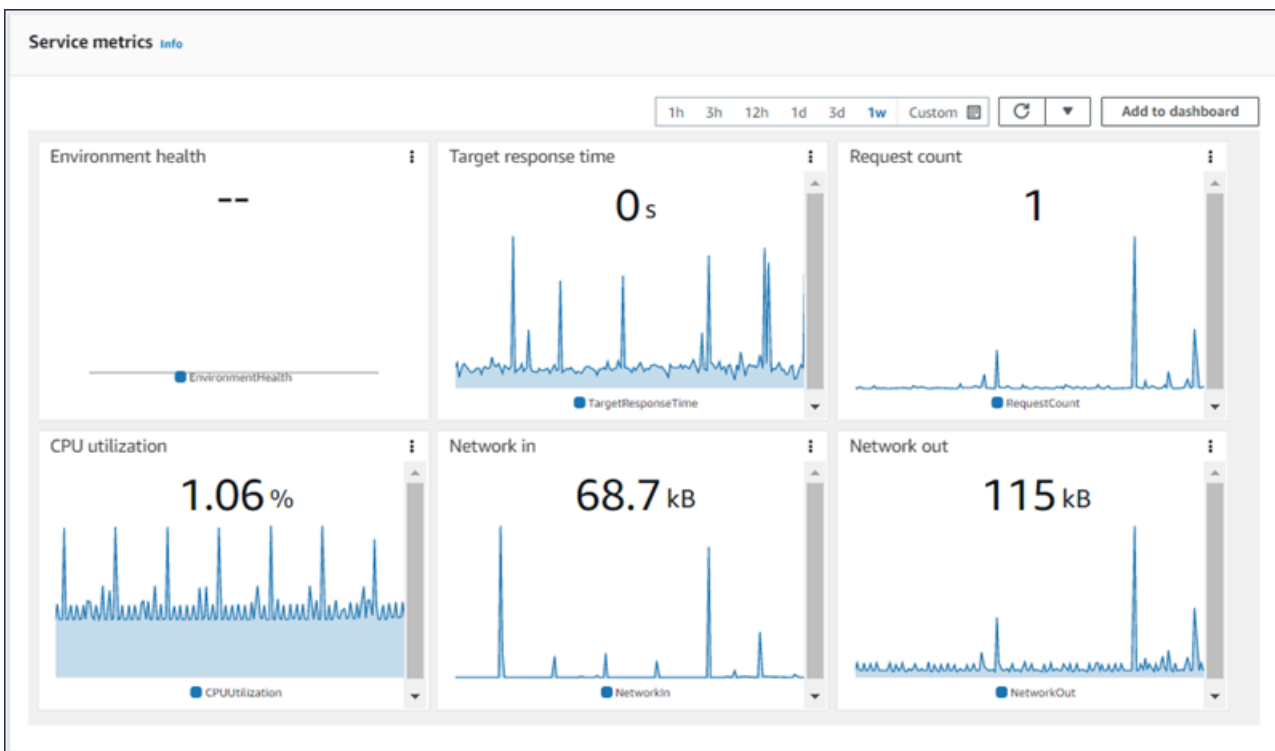
Request logs ▲ ↻  
Request logs ✓  
Last 100 lines  
Full

Log file	Time	EC2 instance	Type
Download	January 11, 2023 22:00:23 (UTC-5)	i-0b1530c3cabd58083	bundle
Download	January 11, 2023 22:00:23 (UTC-5)	i-04a22cd25ba2f7c4e	bundle
Download	January 11, 2023 22:01:04 (UTC-5)	i-04a22cd25ba2f7c4e	tail
Download	January 11, 2023 22:01:04 (UTC-5)	i-0b1530c3cabd58083	tail

Para obter mais informações, consulte [Visualizar logs de instâncias do Amazon EC2 no ambiente do Elastic Beanstalk](#).

## Monitorar

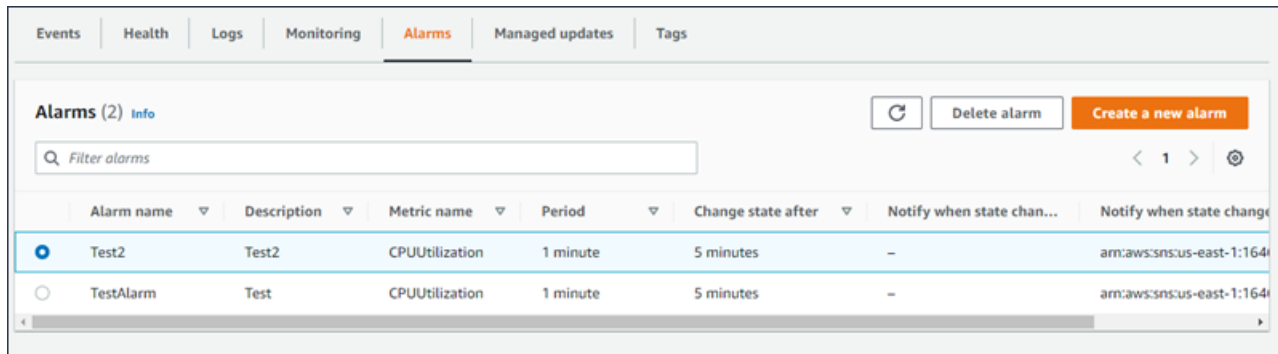
A página Monitoring (Monitoramento) exibe uma visão geral de informações de integridade de seu ambiente. Isso inclui o conjunto padrão de métricas fornecidas pelo Elastic Load Balancing e o Amazon EC2 e os gráficos que mostram como a integridade do ambiente mudou ao longo do tempo.



Para obter mais informações, consulte [Monitoramento da integridade do ambiente no Console de Gerenciamento da AWS](#).

## alarmes

A página Existing alarms (Alarmes existentes) exibe informações sobre todos os alarmes que você configurou em seu ambiente. Você pode usar as opções dessa página para modificar ou excluir alarmes.



Para obter mais informações, consulte [Gerenciar alarmes](#).

## Atualizações gerenciadas

A página Managed updates overview (Visão geral de atualizações gerenciadas) exibe informações sobre as atualizações gerenciadas da plataforma concluídas ou futuras e a substituição de instâncias.

O recurso de atualizações gerenciadas permite que você configure o ambiente para ser atualizado automaticamente com a última versão da plataforma durante uma janela de manutenção semanal de sua escolha. Entre as versões da plataforma, você pode optar por fazer com que seu ambiente substitua todas as instâncias do Amazon EC2 durante a janela de manutenção. Isso pode diminuir os problemas que ocorrem quando a aplicação é executada por longos períodos.

Para obter mais informações, consulte [Atualizações gerenciadas de plataforma](#).

Events | Health | Logs | Monitoring | Alarms | **Managed updates** | Tags

ⓘ A new platform version is available. A platform update has been scheduled to run during the next maintenance window, to perform the replacement immediately, choose **Apply now**.

Apply now

Managed updates history (3) [Info](#) ↻

< 1 > ⚙

Start time	Duration	Update information	Result
December 8, 2022 13:18:30 (UTC-5)	12:09	Platform update from 64bit Amazon Linux 2 runni...	Completed
November 10, 2022 13:00:28 (UTC-5)	13:14	Platform update from 64bit Amazon Linux 2 runni...	Completed
January 5, 2023 14:06:26 (UTC-5)	3:48	Platform update from 64bit Amazon Linux 2 runni...	Failed - RollbackSuccessful Successful abort of the Managed Action.

Para obter mais informações, consulte [Atualizações gerenciadas de plataforma](#).

## Tags

A página Tags exibe as tags que o Elastic Beanstalk aplicou ao ambiente quando o criou e as tags adicionadas. É possível adicionar, editar e excluir tags personalizadas. Não é possível editar ou excluir as tags aplicadas pelo Elastic Beanstalk.

As tags do ambiente são aplicadas a todos os recursos que o Elastic Beanstalk cria para oferecer suporte à aplicação.

Events | Health | Logs | Monitoring | Alarms | Managed updates | **Tags**

**Tags for Gettingstarted-env**

Apply up to 47 tags in addition to the default tags to the resources in your environment. You can use tags to group and filter your environments. A tag is a key-value pair. The key must be unique within the environment and is case-sensitive. [Learn more](#) ↗

Manage tags

Key	Value
Name	Gettingstarted-env
elasticbeanstalk:environment-name	Gettingstarted-env
elasticbeanstalk:environment-id	e-irkuacn9ny

Para obter mais informações, consulte [Marcar recursos em seus ambientes do Elastic Beanstalk](#).

## Configuração

A página Configuration (Configuração) mostra a configuração atual do ambiente e seus recursos, incluindo instâncias do Amazon EC2, um balanceador de carga, notificações e configurações de monitoramento de integridade. Use as configurações nessa página para personalizar o comportamento do seu ambiente durante as implantações, habilitar recursos adicionais e modificar o tipo de instância e outras configurações que você escolher durante a criação do ambiente.



Elastic Beanstalk > Environments > Gettingstarteda-env > Configuration

## Configuration Info

---

### Service access Info

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role  
arn:aws:iam::164656829171:role/aws-elasticbeanstalk-service-role

---

### Instance traffic and scaling Info

Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic. Configure the software that runs on your environment's instances by setting platform-specific options.

**Instances**  
IMDSv1  
Deactivated

**Capacity**

Environment type	Fleet composition	On-demand base
Load balanced	On-Demand instances	0
On-demand above base	Processor type	Instance types
70	x86_64	t2.micro,t2.small

**Load balancer**  
Load balancer type  
application

---

### Networking, database, and tags Info

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

**Network**

Load balancer visibility	Load balancer subnets
public	—

**Database**  
Has coupled database  
false

---

### Updates, monitoring, and logging Info

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

**Updates**

Managed updates	Update batch size	Deployment batch size
Deactivated	1	100
Deployment batch size type	Command timeout	Deployment policy
Percentage	600	AllAtOnce
Health threshold	Ignore health check	Instance replacement
Ok	false	false
Minimum capacity	Notifications email	
0	—	

Para obter mais informações, consulte [Configurar ambientes do Elastic Beanstalk](#).

# Criar um ambiente do Elastic Beanstalk

Um ambiente do AWS Elastic Beanstalk é um conjunto de recursos da AWS que executam uma versão da aplicação. Você pode implantar múltiplos ambientes quando precisar executar várias versões de um aplicativo. Por exemplo, você pode ter ambientes de desenvolvimento, integração e produção.

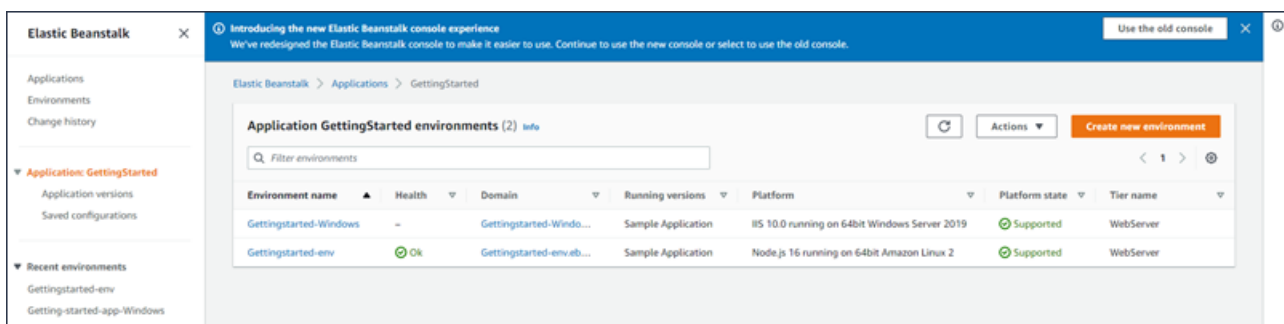
O procedimento a seguir inicia um novo ambiente executando o aplicativo padrão. Essas etapas são simplificadas para colocar seu ambiente em funcionamento rapidamente, usando valores padrão de opções. Para obter instruções detalhadas com descrições das muitas opções que podem ser usadas para configurar os recursos que o Elastic Beanstalk implanta em seu nome, consulte [O assistente de criação de novo ambiente](#).

## Observações

- Para obter instruções sobre a criação e o gerenciamento de ambientes com a CLI da EB, consulte [Gerenciar ambientes do Elastic Beanstalk com a CLI do EB](#).
- A criação de um ambiente requer as permissões na política gerenciada de acesso completo do Elastic Beanstalk. Para mais detalhes, consulte [Política de usuário do Elastic Beanstalk](#).

Para iniciar um ambiente com um aplicativo de exemplo (console)

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, escolha Aplicativos e o nome de um aplicativo existente na lista ou [crie um](#).
3. Na página de visão geral da aplicação, escolha Create a new environment (Criar um novo ambiente).



Isso inicia o assistente Create environment (Criar ambiente). O assistente fornece um conjunto de etapas para a criação de um novo ambiente.

Step 1  
**Configure environment**

---

Step 2  
Configure service access

---

Step 3 - optional  
Configure instance traffic and scaling

---

Step 4 - optional  
Set up networking, database, and tags

---

Step 5 - optional  
Configure updates, monitoring, and logging

---

Step 6  
Review

## Configure environment [Info](#)

### Environment tier [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- Web server environment**  
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
- Worker environment**  
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

### Application information [Info](#)

**Application name**  
GettingStarted  
Maximum length of 100 characters.

▶ **Application tags (optional)**

### Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

**Environment name**  
GettingStarted-env  
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

**Domain name**  
 .us-east-1.elasticbeanstalk.com

**Environment description**

### Platform [Info](#)

**Platform type**

- Managed platform**  
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)
- Custom platform**  
Platforms created and owned by you. This option is unavailable if you have no platforms.

**Platform**


**Platform branch**

**Platform version**

### Application code [Info](#)


- Sample application**
- Existing version**  
Application versions that you have uploaded.
- Upload your code**  
Upload a source bundle from your computer or copy one from Amazon S3.

4. Para nível de ambiente, escolha o [nível de ambiente](#) Web server environment (Ambiente do servidor Web) ou Worker environment (Ambiente do operador). Você não pode alterar o nível de um ambiente após a criação.

 Note

O [.NET na Plataforma de servidor Windows](#) não tem suporte para a camada de ambiente do operador.

5. Em Plataforma, selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.

 Note

O Elastic Beanstalk é compatível com várias [versões](#) para a maioria das plataformas listadas. Por padrão, o console seleciona a versão recomendada da plataforma e a ramificação da plataforma escolhida. Se a aplicação exigir outra versão, você poderá selecioná-la aqui. Para obter informações sobre versões de plataforma compatíveis, consulte [the section called “Plataformas compatíveis”](#).

6. Para Application code, escolha Sample application.
7. Em Configuration presets (Predefinições de configuração), escolha Single instance (Instância única).
8. Escolha Next (Próximo).
9. A página Configurar acesso ao serviço é exibida.

**Configure service access** [Info](#)

**Service access**  
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

**Service role**  
 Create and use new service role  
 Use an existing service role

**Existing service roles**  
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

**EC2 key pair**  
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

**EC2 instance profile**  
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

10. Escolha Usar um perfil de serviço existente em Perfil de serviço.

11. Em seguida, vamos nos concentrar na lista suspensa Perfil de instância do EC2. Os valores exibidos nessa lista suspensa podem variar, dependendo de a conta ter criado anteriormente um novo ambiente.

Escolha uma das opções a seguir com base nos valores exibidos na lista.

- Se `aws-elasticbeanstalk-ec2-role` for exibido na lista suspensa, selecione-o na lista suspensa Perfil de instância do EC2.
- Se outro valor for exibido na lista e for o perfil padrão de instância do EC2 para seus ambientes, selecione-o na lista suspensa Perfil de instância do EC2.
- Se a lista suspensa Perfil de instância do EC2 não mostrar nenhum valor para seleção, expanda o procedimento que se segue, Criar um perfil do IAM para ser o perfil de instância do EC2.

Conclua as etapas em Criar perfil do IAM para ser o perfil de instância do EC2 para criar um perfil do IAM que você possa selecionar posteriormente para ser o perfil de instância do EC2. Depois, retorne para esta etapa.

Agora que você criou um perfil do IAM e atualizou a lista, ele será exibido como uma opção na lista suspensa. Selecione o perfil do IAM que você acabou de criar na lista suspensa Perfil de instância do EC2.

12. Escolha Skip to Review (Pular para revisão) na página Configure service access (Configurar acesso a serviço).

Isso selecionará os valores padrão para essa etapa e pulará as etapas opcionais.

13. A página Review (Revisão) exibe um resumo de todas as suas escolhas.

Para personalizar ainda mais o ambiente, escolha Edit (Editar) ao lado da etapa que inclui os itens que você deseja configurar. Você pode configurar as opções a seguir somente durante a criação do ambiente:

- Nome do ambiente
- Nome de domínio
- Versão da plataforma
- Processor
- VPC
- Nível

Você pode alterar as configurações a seguir após a criação do ambiente, mas elas exigem que novas instâncias ou outros recursos sejam provisionados, e pode levar muito tempo para que elas sejam aplicadas:

- Tipo de instância, volume raiz, par de chaves e função do AWS Identity and Access Management (IAM)
- Banco de dados interno do Amazon RDS
- Load balancer

Para obter detalhes sobre todas as configurações disponíveis, consulte [O assistente de criação de novo ambiente](#).

14. Escolha Submit (enviar) na parte inferior da página para inicializar a criação do novo ambiente.

## Criar um perfil do IAM para ser o perfil de instância do EC2

The screenshot shows the 'Configure service access' dialog box in the AWS IAM console. The title is 'Configure service access' with an 'Info' link. Below the title is a 'Service access' section with a brief explanation and a 'Learn more' link. The 'Service role' section has two radio buttons: 'Create and use new service role' (unselected) and 'Use an existing service role' (selected). Under 'Existing service roles', there is a text prompt and a dropdown menu showing 'aws-elasticbeanstalk-service-role' with a refresh button. The 'EC2 key pair' section has a text prompt and a dropdown menu showing 'Choose a key pair' with a refresh button. The 'EC2 instance profile' section has a text prompt and a dropdown menu showing 'aws-elasticbeanstalk-ec2-role' with a refresh button. At the bottom left of the dialog is a 'View permission details' button. At the bottom right are four buttons: 'Cancel', 'Skip to review', 'Previous', and 'Next'.

Para criar um perfil do IAM para ser o perfil de instância do EC2

1. Escolha Visualizar detalhes de permissão. Essa opção exibe a lista suspensa Perfil de instância do EC2.

Uma janela modal intitulada Visualizar permissões de perfil de instância é exibida. Essa janela lista os perfis gerenciados que deverão ser anexados ao novo perfil de instância do EC2 que você criar. Ela também fornece um link para iniciar o console do IAM.

2. Escolha o link Console do IAM exibido na parte superior da janela.
3. No painel de navegação do console do IAM, escolha Roles (Perfis).
4. Selecione Create role (Criar função).
5. Em Tipo de entidade confiável, selecione Serviço da AWS.
6. Em Use case (Caso de uso), selecione EC2.
7. Escolha Next (Próximo).
8. Anexe as políticas gerenciadas apropriadas. Role a janela modal Visualizar permissões de perfil de instância para ver as políticas gerenciadas. As políticas também estão listadas aqui:



- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

9. Escolha Next (Próximo).
10. Insira um nome para a função.
11. (Opcional) Adicione tags à função.
12. Selecione Create role (Criar função).
13. Retorne à janela do console do Elastic Beanstalk que está aberta.
14. Feche a janela modal Visualizar permissões de perfil de instância.

 Important

Não feche a página do navegador que exibe o console do Elastic Beanstalk.

15. Escolha



(atualizar), ao lado da lista suspensa Perfil de instância do EC2.

Essa opção atualiza a lista suspensa para que o perfil que você acabou de criar seja exibido na lista suspensa.

Enquanto o Elastic Beanstalk cria seu ambiente, você será redirecionado para o [console do Elastic Beanstalk](#). Quando a integridade do ambiente ficar verde, escolha o URL próximo ao nome do ambiente para visualizar o aplicativo em execução. Esse URL costuma ser acessível pela internet, a menos que você configure seu ambiente para usar uma [VPC personalizada com um load balancer interno](#).

## Tópicos

- [O assistente de criação de novo ambiente](#)
- [Clonar um ambiente do Elastic Beanstalk](#)
- [Encerrar um ambiente do Elastic Beanstalk](#)
- [Criação de ambientes do Elastic Beanstalk com a CLI da AWS](#)
- [Criar ambientes do Elastic Beanstalk com a API](#)

- [Construção de um URL "Iniciar agora"](#)
- [Criar e atualizar grupos de ambientes do Elastic Beanstalk](#)

## O assistente de criação de novo ambiente

Em [Criar um ambiente do Elastic Beanstalk](#) mostramos como abrir o assistente Create new environment (Criar novo ambiente) e criar um ambiente rapidamente. Escolha Create environment (Criar ambiente) para iniciar um ambiente com um nome de ambiente padrão, um domínio gerado automaticamente, um código de exemplo do aplicativo e as configurações recomendadas.

Este tópico descreve o assistente Create new environment (Criar novo ambiente) e todas as maneiras de usá-lo para configurar o ambiente que você deseja criar.

### Página do assistente

O assistente Create environment (Cria ambiente) fornece um conjunto de etapas para a criação de um novo ambiente.

Step 1  
**Configure environment**

---

Step 2  
Configure service access

---

Step 3 - optional  
Configure instance traffic and scaling

---

Step 4 - optional  
Set up networking, database, and tags

---

Step 5 - optional  
Configure updates, monitoring, and logging

---

Step 6  
Review

## Configure environment [Info](#)

### Environment tier [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- Web server environment**  
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
- Worker environment**  
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

### Application information [Info](#)

**Application name**  
GettingStarted  
Maximum length of 100 characters.

▶ **Application tags (optional)**

### Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

**Environment name**  
GettingStarted-env  
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

**Domain name**  
 .us-east-1.elasticbeanstalk.com

**Environment description**

### Platform [Info](#)

**Platform type**

- Managed platform**  
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)
- Custom platform**  
Platforms created and owned by you. This option is unavailable if you have no platforms.

**Platform**

**Platform branch**

**Platform version**

### Application code [Info](#)



- Sample application**
- Existing version**  
Application versions that you have uploaded.
- Upload your code**  
Upload a source bundle from your computer or copy one from Amazon S3.

## Nível do ambiente

Em environment tier (nível de ambiente), escolha Web server environment (Ambiente do servidor Web) ou Worker environment (Ambiente do operador). Você não pode alterar o nível de um ambiente após a criação.

**Environment tier** [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- Web server environment**  
Run a website, web application, or web API that serves HTTP requests. [Learn more](#) 
- Worker environment**  
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#) 

### Note

O [.NET na Plataforma de servidor Windows](#) não tem suporte para a camada de ambiente do operador.

## Informações sobre a aplicação

Se você iniciou o assistente selecionando Create new environment (Criar novo ambiente) na página de Application overview (Visão geral da aplicação, o Application name (Nome da aplicação) foi preenchido previamente. Do contrário, insira um nome de aplicação. Opcionalmente, adicione [tags da aplicação](#).

### Application information [Info](#)

**Application name**

Maximum length of 100 characters.

▼ **Application tags (optional)**

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

No tags associated with the resource.

You can add 50 more tags.

## Informações sobre o ambiente

Defina o nome e o subdomínio do ambiente, e crie uma descrição para o ambiente. Lembre-se de que essas configurações de ambiente não podem ser alteradas após a criação do ambiente.

### Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

**Environment name**

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

**Domain name**

 .us-east-1.elasticbeanstalk.com 

**Environment description**

- **Name (Nome):** insira um nome para o ambiente. O formulário fornece um nome gerado.
- **Domain (Domínio):** (ambientes de servidor Web) insira um nome de domínio exclusivo para o ambiente. O nome padrão é o nome do ambiente. Você pode inserir outro nome de domínio. O Elastic Beanstalk usa esse nome para criar um CNAME exclusivo para o ambiente. Para

verificar se o nome do domínio desejado está disponível, escolha Check Availability (Verificar disponibilidade).

- Description (Descrição): insira uma descrição para esse ambiente.

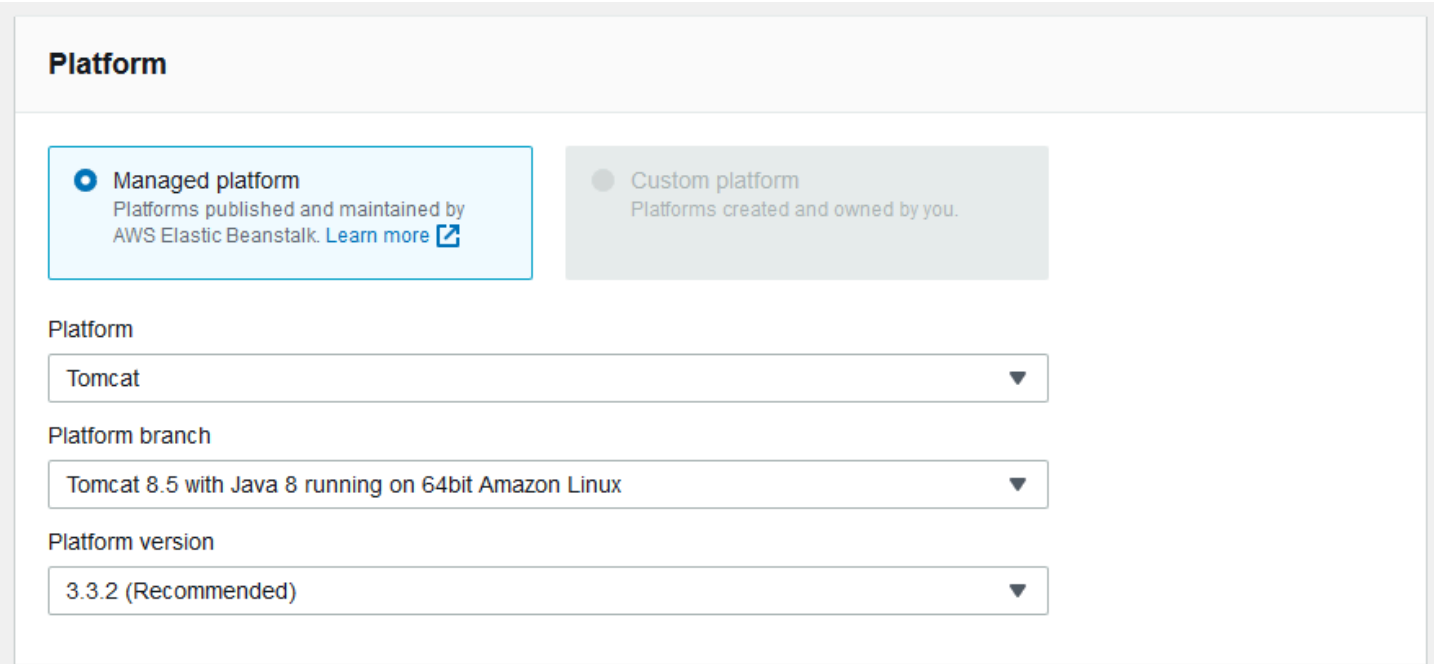
Selecionar uma plataforma para o novo ambiente

Você pode criar um novo ambiente a partir de dois tipos de plataformas:

- Plataforma gerenciada
- Plataforma personalizada

Plataforma gerenciada

Na maioria dos casos, é usada uma plataforma gerenciada do Elastic Beanstalk para seu novo ambiente. Quando o novo assistente de ambiente é iniciado, ele seleciona a opção Managed platform (Plataforma gerenciada) por padrão.



**Platform**

Managed platform  
Platforms published and maintained by AWS Elastic Beanstalk. [Learn more](#)

Custom platform  
Platforms created and owned by you.

Platform  
Tomcat

Platform branch  
Tomcat 8.5 with Java 8 running on 64bit Amazon Linux

Platform version  
3.3.2 (Recommended)

Selecione uma plataforma, uma ramificação de plataforma dentro dessa plataforma e uma versão específica da plataforma na ramificação. Quando você seleciona uma ramificação de plataforma, a versão recomendada dentro da ramificação é selecionada por padrão. Além disso, é possível selecionar qualquer versão de plataforma que você usou antes.

**Note**

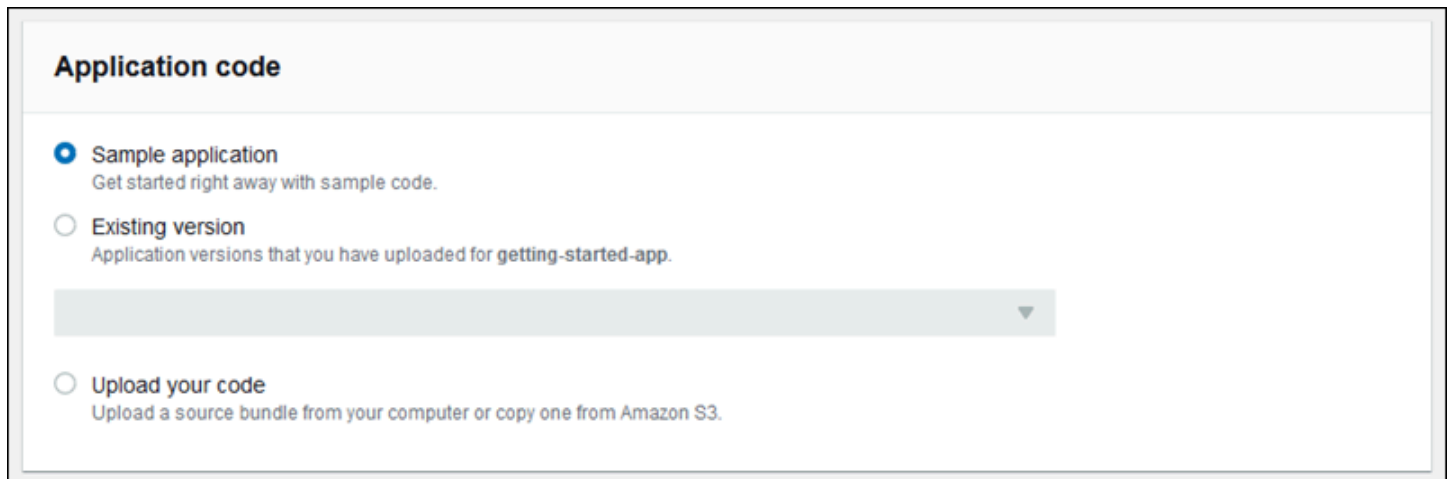
Para um ambiente de produção, recomendamos escolher uma versão de plataforma em uma ramificação compatível da plataforma. Para obter detalhes sobre estados de ramificação da plataforma, consulte a definição de Ramificação da plataforma no [the section called “Glossário de plataformas”](#).

## Plataforma personalizada

Se uma plataforma pronta para uso não atender às suas necessidades, crie um novo ambiente a partir de uma plataforma personalizada. Para especificar uma plataforma personalizada, escolha a opção Custom platform (Plataforma personalizada) e selecione uma das plataformas personalizadas disponíveis. Se não houver plataformas personalizadas disponíveis, essa opção ficará esmaecida.

## Fornecer o código do aplicativo

Agora que você selecionou a plataforma a ser usada, a próxima etapa é fornecer o código do aplicativo.



**Application code**

**Sample application**  
Get started right away with sample code.

**Existing version**  
Application versions that you have uploaded for getting-started-app.

**Upload your code**  
Upload a source bundle from your computer or copy one from Amazon S3.

Você tem várias opções:

- É possível usar a aplicação demonstrativa que o Elastic Beanstalk fornece para cada plataforma.
- Você pode usar o código que já implantou no Elastic Beanstalk. Escolha Existing version (Versão existente) e seu aplicativo na seção Application code (Código do aplicativo).
- Você pode fazer upload do novo código. Escolha Upload your code (Fazer upload do código) e Upload (Fazer upload). Você pode fazer upload do novo código da aplicação de um arquivo local ou especificar o URL do bucket do Amazon S3 que contém o código da aplicação.

**Note**

Dependendo da versão da plataforma selecionada, você pode fazer upload da aplicação em um [pacote de origem ZIP](#), um [arquivo WAR](#) ou uma [configuração do Docker de texto simples](#). O tamanho máximo do arquivo é 500 MB.

Quando você opta por fazer upload de um novo código, também é possível fornecer tags para associar ao novo código. Para obter mais informações sobre como marcar uma versão do aplicativo, consulte [the section called “Atribuir tags em versões do aplicativo”](#).



## Application code

- Sample application  
Get started right away with sample code.
- Existing version  
Application versions that you have uploaded for `getting-started-app`.

- Upload your code  
Upload a source bundle from your computer or copy one from Amazon S3.


### ▼ Source code origin

(Maximum size 512 MB)

- Local file
- Public S3 URL

 Choose file

File name : **java-tomcat-v3.zip**

 File successfully uploaded

Version label

Unique name for this version of your application code.

getting-started-app-source

### ▼ Application code tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

Key

Value

Remove tag

Add tag

50 remaining

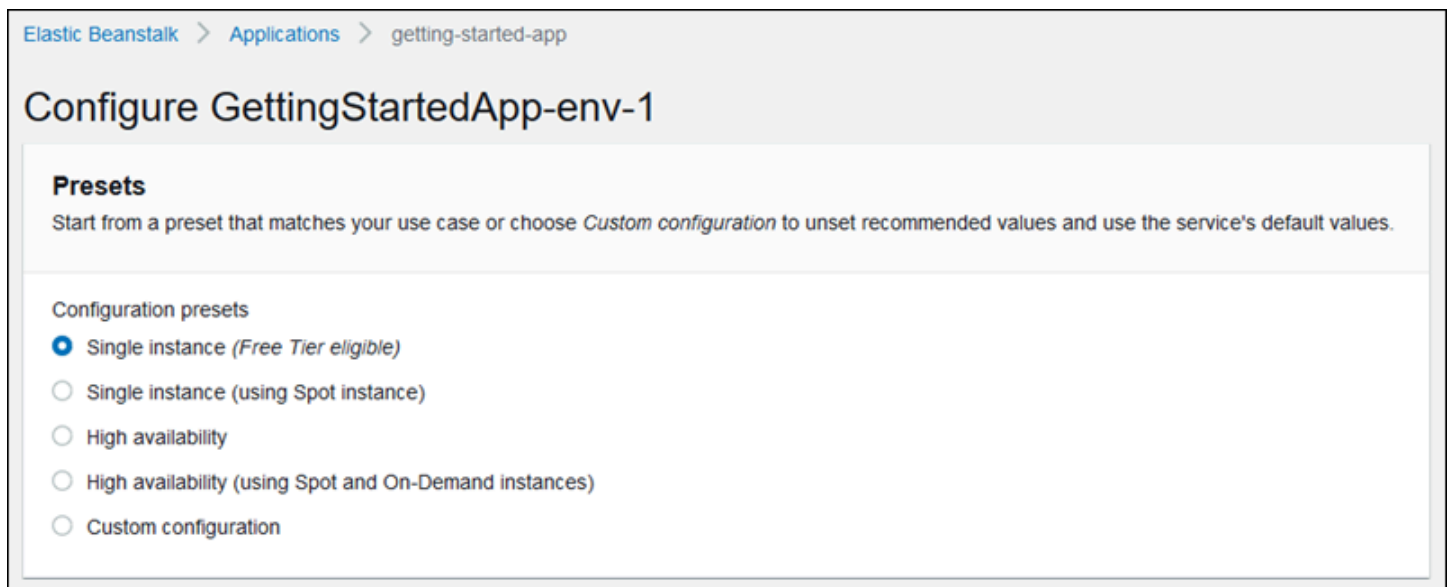
Para uma rápida criação do ambiente usando opções de configuração padrão, agora é possível escolher Create environment (Criar ambiente). Escolha Configure more options (Configurar mais opções) para fazer alterações adicionais na configuração, conforme descrito nas seções a seguir.

## Página de configuração do assistente

Quando você escolhe Configure more options (Configurar mais opções), o assistente exibe a página Configure (Configurar). Nessa página, é possível selecionar uma predefinição de configuração, alterar a versão da plataforma que deseja que seu ambiente use ou fazer escolhas de configuração específicas para o novo ambiente.

### Escolher uma configuração predefinida

Na seção Presets (Predefinições) da página, o Elastic Beanstalk fornece várias predefinições de configuração para diferentes casos de uso. Cada predefinição inclui valores recomendados para várias [opções de configuração](#).



As predefinições de High availability (Alta disponibilidade) incluem um load balancer e são recomendadas para ambientes de produção. Escolha essa opção se você quiser um ambiente que possa executar várias instâncias com alta disponibilidade e reduzir a resposta à carga. As predefinições de Single instance (Instância única) são recomendadas, principalmente, para desenvolvimento. Duas das predefinições permitem solicitações de instâncias spot. Para obter detalhes sobre a configuração de capacidade do Elastic Beanstalk, consulte [Grupo de Auto Scaling](#).

A última predefinição, Custom configuration (Configuração personalizada), remove todos os valores recomendados, exceto as configurações de função, e usa os padrões de API. Escolha essa opção se

you are implementing a package of origin with [configuration files](#) that define configuration options. A Custom configuration (Custom configuration) is also selected automatically when you modify the default configuration Low cost (Low cost) or High availability (High availability).

## Personalizar sua configuração

In addition to (or instead of) choosing a default configuration, you can adjust the [configuration options](#) in your environment. The Configure (Configure) assistant displays various categories of configuration. Each category of configuration displays a summary of values for a group of configuration definitions. Choose Edit (Edit) to edit this group of configurations.

## Categorias de configuração

- [Software configurations](#)
- [Instances](#)
- [Capacity](#)
- [Load balancer](#)
- [Continuous updates and deployments](#)
- [Security](#)
- [Monitor](#)
- [Managed updates](#)
- [Notifications](#)
- [Network](#)
- [Database](#)
- [Tags](#)
- [Operator environments](#)

## Configurações de software

Use the Modify software (Modify software) page to configure software on instances of Amazon Elastic Compute Cloud (Amazon EC2) that run your application. It is possible to configure environment properties, instance log storage, streaming and AWS X-Ray debugging and specific platform configurations. For more details, consult [the section called "Environment properties and software configurations"](#).

Elastic Beanstalk > Applications > getting-started-app

## Modify software

The following settings control platform behavior and let you pass key-value pairs in as OS environment variables. [Learn more](#)

### Platform options

Target .NET runtime  
4.0

Enable 32-bit applications  
False

### AWS X-Ray

X-Ray daemon

## Instâncias

Use a página de configuração [Modify instances](#) (Modificar instâncias) para configurar as instâncias do Amazon EC2 que executam sua aplicação. Para obter mais detalhes, consulte [the section called “Instâncias do Amazon EC2”](#).

Elastic Beanstalk > Applications > getting-started-app

## Modify instances

### Amazon CloudWatch monitoring

The time interval between when metrics are reported from the EC2 instances.

Monitoring interval  
5 minute

### Root volume (boot device)

Root volume type  
(Container default)

## Capacity

Use a página de configuração [Modify capacity](#) (Modificar capacidade) para configurar a capacidade computacional do ambiente e das definições [Auto Scaling group](#) (Grupo de Auto Scaling) para

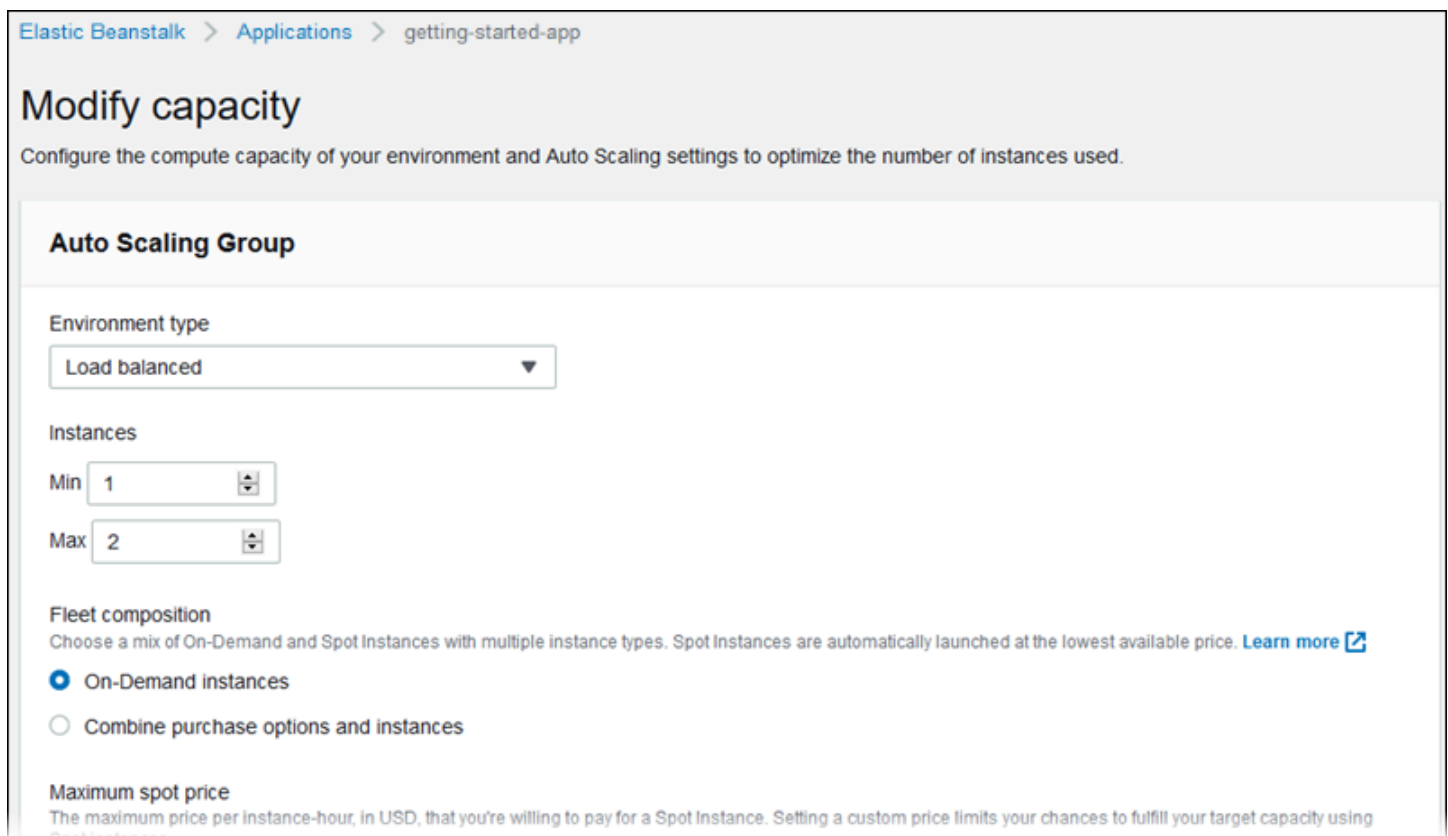
otimizar o número e o tipo de instâncias que você está usando. Também é possível alterar a capacidade do ambiente com base em triggers ou em uma programação.

Um ambiente com balanceamento de carga pode executar várias instâncias para alta disponibilidade e evitar o tempo de inatividade durante as atualizações de configuração e implantações. Em um ambiente com balanceamento de carga, o nome de domínio é mapeado para o load balancer. Em um ambiente de uma instância única, ele é mapeado para um endereço IP elástico na instância.

### Warning

Um ambiente de única instância não está pronto para produção. Se a instância tornar-se instável durante a implantação, ou o Elastic Beanstalk encerrar e reiniciar a instância durante uma atualização de configuração, sua aplicação poderá ficar indisponível durante um determinado período. Use ambientes de única instância para desenvolvimento, teste ou preparação. Use ambientes com balanceamento de carga para produção.

Para obter mais informações sobre configurações de capacidade do ambiente, consulte [the section called “Grupo de Auto Scaling”](#) e [the section called “Instâncias do Amazon EC2”](#).



Elastic Beanstalk > Applications > getting-started-app

## Modify capacity

Configure the compute capacity of your environment and Auto Scaling settings to optimize the number of instances used.

### Auto Scaling Group

Environment type

Load balanced

Instances

Min 1

Max 2

Fleet composition

Choose a mix of On-Demand and Spot Instances with multiple instance types. Spot Instances are automatically launched at the lowest available price. [Learn more](#)

On-Demand instances

Combine purchase options and instances

Maximum spot price

The maximum price per instance-hour, in USD, that you're willing to pay for a Spot Instance. Setting a custom price limits your chances to fulfill your target capacity using Spot Instances.

## Load balancer

Use a página de configuração Modify Load balancer (Modificar load balancer) para selecionar um tipo de load balancer e definir suas configurações. Em ambientes com balanceamento de carga, o load balancer do ambiente é o ponto de entrada de todo o tráfego de seu aplicativo. O Elastic Beanstalk é compatível com vários tipos de balanceador de carga. Por padrão, o console do Elastic Beanstalk cria um Application Load Balancer e o configura para servir o tráfego HTTP na porta 80.

### Note

É possível selecionar somente o tipo de load balancer do ambiente durante a criação do ambiente.

Para obter mais informações sobre os tipos e configurações de load balancer, consulte [the section called “Load balancer”](#) e [the section called “HTTPS”](#).

Elastic Beanstalk &gt; Applications &gt; getting-started-app

## Modify load balancer

### Application Load Balancer

Application layer load balancer—routing HTTP and HTTPS traffic based on protocol, port, and route to environment processes.

### Classic Load Balancer

*Previous generation* — HTTP, HTTPS, and TCP

### Network Load Balancer

Ultra-high performance and static IP addresses for your application.

### Application Load Balancer

You can specify listeners for your load balancer. Each listener routes incoming client traffic on a specified port using a specified protocol to your environment processes. By default, we've configured your load balancer with a standard web server on port 80.

Actions ▾

Add listener

<input type="checkbox"/>	Port	Protocol	SSL certificate	Enabled
<input type="checkbox"/>	80	HTTP	--	<input checked="" type="checkbox"/>

### Processes

For each environment process, you can specify the protocol and port that the load balancer uses to route requests to the process. You can

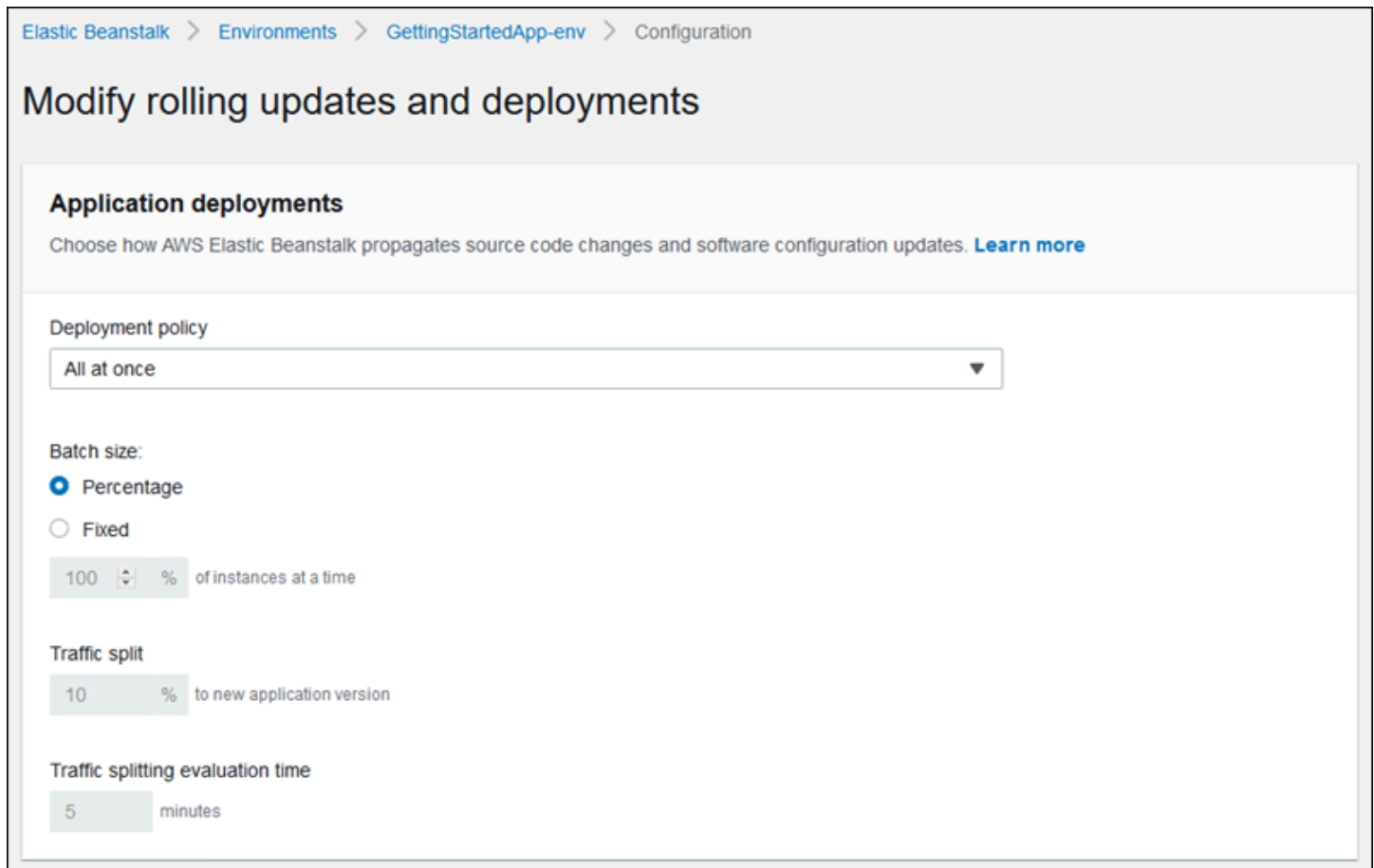
#### Note

A opção Classic Load Balancer (CLB) está desabilitada no assistente do console Criar ambiente. Se você tiver um ambiente existente configurado com um Classic Load Balancer, poderá criar um novo [clonando o ambiente existente](#) usando o console do Elastic Beanstalk ou a [CLI do EB](#). Você também tem a opção de usar a [CLI do EB](#) ou a [AWS CLI](#) para criar um novo ambiente configurado com um Classic Load Balancer. Essas ferramentas de linha de comando criarão um novo ambiente com um CLB mesmo que ainda não exista um em sua conta.

## Atualizações e implantações contínuas

Use a página de configuração `Modify rolling updates and deployments` (Modificar atualizações contínuas e implantações) para configurar como o Elastic Beanstalk processa implantações de aplicações e atualizações de configuração para seu ambiente.

As implantações de aplicativos ocorrem quando você carrega um pacote de origem de aplicativo atualizado e o implementa em seu ambiente. Para obter mais informações sobre como configurar implantações, consulte [the section called “Opções de implantação”](#).



The screenshot shows the AWS Elastic Beanstalk console interface for configuring application deployments. The breadcrumb navigation at the top reads: `Elastic Beanstalk > Environments > GettingStartedApp-env > Configuration`. The main heading is `Modify rolling updates and deployments`. Below this, there is a section titled `Application deployments` with a sub-header: `Choose how AWS Elastic Beanstalk propagates source code changes and software configuration updates. Learn more`. The configuration options are as follows:

- Deployment policy:** A dropdown menu currently set to `All at once`.
- Batch size:** Two radio buttons are present: `Percentage` (which is selected) and `Fixed`. Below the radio buttons is a text input field containing `100` followed by `% of instances at a time`.
- Traffic split:** A text input field containing `10` followed by `% to new application version`.
- Traffic splitting evaluation time:** A text input field containing `5` followed by `minutes`.

As alterações de configuração que modificam a [configuração de execução](#) ou [configurações de VPC](#) exigem o encerramento de todas as instâncias no ambiente e a substituição. Para obter mais informações sobre como definir o tipo de atualização e outras opções, consulte [the section called “Alterações de configuração”](#).



### Configuration updates

Changes to virtual machine settings and VPC configuration trigger rolling updates to replace the instances in your environment without downtime.  
[Learn more](#)

**Rolling update type**  
Rolling based on Health

**Batch size**  
1  
The maximum number of instances to replace in each phase of the update.

**Minimum capacity**  
1  
The minimum number of instances to keep in service at all times.

**Pause time**  
hh:mm:ss  
Pause the update for up to an hour between each batch.

## Segurança

Use a página [Configure service access](#) (Configurar acesso ao serviço) para configurar as definições de segurança dos serviços e das instâncias.

Para obter uma descrição dos conceitos de segurança do Elastic Beanstalk, consulte [Permissões](#).

A primeira vez que você criar um ambiente no console do Elastic Beanstalk, deverá criar um perfil de instância do EC2 com um conjunto padrão de permissões. Se a lista suspensa Perfil de instância do EC2 não mostrar nenhum valor para seleção, expanda o procedimento que se segue. Ele fornece as etapas para a criação de um perfil que você poderá selecionar posteriormente para ser o Perfil de instância do EC2.


Criar um perfil do IAM para ser o perfil de instância do EC2

Para criar um perfil do IAM para ser o perfil de instância do EC2

1. Escolha Visualizar detalhes de permissão. Essa opção exibe a lista suspensa Perfil de instância do EC2.

Uma janela modal intitulada Visualizar permissões de perfil de instância é exibida. Essa janela lista os perfis gerenciados que deverão ser anexados ao novo perfil de instância do EC2 que você criar. Ela também fornece um link para iniciar o console do IAM.

2. Escolha o link Console do IAM exibido na parte superior da janela.
3. No painel de navegação do console do IAM, escolha Roles (Perfis).
4. Selecione Create role (Criar função).
5. Em Tipo de entidade confiável, selecione Serviço da AWS.
6. Em Use case (Caso de uso), selecione EC2.
7. Escolha Next (Próximo).
8. Anexe as políticas gerenciadas apropriadas. Role a janela modal Visualizar permissões de perfil de instância para ver as políticas gerenciadas. As políticas também estão listadas aqui:
  - AWSElasticBeanstalkWebTier
  - AWSElasticBeanstalkWorkerTier
  - AWSElasticBeanstalkMulticontainerDocker
9. Escolha Next (Próximo).
10. Insira um nome para a função.
11. (Opcional) Adicione tags à função.
12. Selecione Create role (Criar função).
13. Retorne à janela do console do Elastic Beanstalk que está aberta.
14. Feche a janela modal Visualizar permissões de perfil de instância.

 Important

Não feche a página do navegador que exibe o console do Elastic Beanstalk.

15. Escolha



(atualizar), ao lado da lista suspensa Perfil de instância do EC2.

Essa opção atualiza a lista suspensa para que o perfil que você acabou de criar seja exibido na lista suspensa.

## Configure service access Info

**Service access**  
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

**Service role**

Create and use new service role

Use an existing service role

**Existing service roles**  
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

**EC2 key pair**  
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

**EC2 instance profile**  
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

## Monitorar

Use a página de configuração Modify monitoring (Modificar monitoramento) para configurar relatórios de integridade, regras de monitoramento e streaming de eventos de integridade. Para obter detalhes, consulte [the section called “Habilitar integridade aprimorada”](#), [the section called “Regras de integridade aperfeiçoada”](#), e [the section called “Streaming de integridade do ambiente”](#).

Elastic Beanstalk > Applications > getting-started-app

## Modify monitoring

### Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#).

System

Enhanced

Basic

CloudWatch Custom Metrics - Instance

Choose metrics

### Atualizações gerenciadas

Use a página de configuração Modify managed updates (Modificar atualizações gerenciadas) para configurar atualizações gerenciadas da plataforma. É possível decidir se deseja habilitá-los, definir a programação e configurar outras propriedades. Para obter mais detalhes, consulte [the section called “Atualizações gerenciadas”](#).

Elastic Beanstalk > Applications > getting-started-app

## Modify managed updates

**Managed platform updates**

Enable managed platform updates to apply platform updates automatically during a weekly maintenance window that you choose. Your application stays available during the update process.

Managed updates  
 Enabled

Weekly update window  
Tuesday at 12 : 00 UTC  
Any available managed updates will run between Tuesday, 4:00 AM and Tuesday, 6:00 AM (-0800 GMT).

Update level  
Minor and patch

Instance replacement  
If enabled, an instance replacement will be scheduled if no other updates are available.  
 Enabled

Cancel Save

## Notificações

Use a página de configuração Modify notifications (Modificar notificações) para especificar um endereço de e-mail para receber [notificações por e-mail](#) de eventos importantes do seu ambiente.

Elastic Beanstalk > Applications > getting-started-app

## Modify notifications

**Email notifications**

Enter an email address to receive email notifications for important events from your environment. [Learn more](#)

Email

## Rede


Se você criou uma [VPC personalizada](#), use a página de configuração Modify network (Modificar rede) a fim de configurar seu ambiente para usá-lo. Se você não escolher uma VPC, o Elastic Beanstalk usará a VPC e as sub-redes padrão.

Elastic Beanstalk > Applications > getting-started-app

## Modify network

### Virtual private cloud (VPC)

VPC  
Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. [Learn more](#)


vpc-0f9c96ae77f3c49c1 (172.31.0.0/16) | private-public 

[Create custom VPC](#)

### Load balancer settings

Assign your load balancer to a subnet in each Availability Zone (AZ) in which your application runs. For a publicly accessible application, set **Visibility** to **Public** and choose public subnets.

Visibility  
Make your load balancer internal if your application serves requests only from connected VPCs. Public load balancers serve requests from the internet.

Public 

### Load balancer subnets

## Banco de dados


Use a página de configuração *Modify database* (Modificar banco de dados) para adicionar um banco de dados do Amazon Relational Database Service (Amazon RDS) ao seu ambiente para desenvolvimento e teste. O Elastic Beanstalk fornece informações de conexão a suas instâncias definindo as propriedades do ambiente para o nome do host do banco de dados, o nome do usuário, a senha, o nome da tabela e a porta.

Para obter mais detalhes, consulte [the section called “Banco de dados”](#).

Elastic Beanstalk > Applications > getting-started-app

## Modify database

Add an Amazon RDS SQL database to your environment for development and testing. AWS Elastic Beanstalk provides connection information to your instances by setting environment properties for the database hostname, username, password, table name, and port. When you add a database to your environment, its lifecycle is tied to your environment's.

For production environments, you can configure your instances to connect to a database. [Learn more](#) 

### Restore a snapshot

Restore an existing snapshot in your account, or create a new database.

Snapshot

None



### Database settings

Choose an engine and instance type for your environment's database.

Engine

mysql

Engine version

## Tags

Use a página de configuração Modify tags (Modificar tags) para adicionar [tags](#) aos recursos no ambiente. Para obter mais informações sobre marcação do ambiente, consulte [Marcar recursos em seus ambientes do Elastic Beanstalk](#).



Elastic Beanstalk > Applications > getting-started-app

## Modify tags

Apply up to 50 tags to the resources in your environment in addition to the default tags.

Key	Value	
<input type="text" value="mytag1"/>	<input type="text" value="value1"/>	<input type="button" value="Remove"/>

49 remaining

### Ambientes de operador

Se você estiver criando um ambiente de nível do operador, use a página de configuração Modify worker (Modificar operador) para configurar o ambiente de operador. O daemon do operador nas instâncias no ambiente extrai itens de uma fila do Amazon Simple Queue Service (Amazon SQS) e os retransmite como mensagens de publicação para a aplicação de operador. É possível escolher a fila do Amazon SQS na qual o daemon do operador lê (gerada automaticamente ou existente). Também é possível configurar as mensagens que o daemon do operador envia para seu aplicativo.

Para obter mais informações, consulte [the section called “Ambientes de operador”](#).

Elastic Beanstalk > Applications > getting-started-app

## Modify worker

You can create a new Amazon SQS queue for your worker application or pull work items from an existing queue. The worker daemon on the instances in your environment pulls an item from the queue and relays it in the body of a POST request to a local HTTP path relative to localhost.

**Queue**

Worker queue

SQS queue from which to read work items.

**Messages**

HTTP path

## Clonar um ambiente do Elastic Beanstalk

É possível criar um ambiente do Elastic Beanstalk existente como a base para um novo ambiente clonando o ambiente existente. Por exemplo, você pode criar um clone para usar uma versão mais recente da ramificação da plataforma utilizada pela plataforma do ambiente original. O Elastic Beanstalk define o clone com as mesmas configurações de ambiente usadas pelo ambiente original. Ao clonar um ambiente existente, em vez de criar um novo ambiente, você não precisa definir manualmente configurações de opções, variáveis de ambiente e outras configurações. O Elastic Beanstalk também cria uma cópia de qualquer recurso da AWS associado ao ambiente original. No entanto, durante o processo de clonagem, o Elastic Beanstalk não copia dados do Amazon RDS para o clone. Depois de criar o ambiente de clone, você pode modificar as definições da configuração do ambiente, conforme necessário.

Só é possível clonar um ambiente para uma versão de plataforma diferente da mesma ramificação da plataforma. Não é garantido que uma ramificação de plataforma diferente seja compatível. Para usar uma ramificação de plataforma diferente, é necessário criar manualmente um ambiente, implantar o código do aplicativo e fazer as alterações necessárias no código e nas opções para garantir que o aplicativo funcione corretamente na nova ramificação da plataforma.

### Note

O Elastic Beanstalk não inclui nenhuma alteração não gerenciada feita nos recursos do clone. As alterações feitas nos recursos da AWS usando ferramentas diferentes do console do Elastic Beanstalk, ferramentas de linha de comando ou API são consideradas não gerenciadas.

## AWSConsole de Gerenciamento da


### Para clonar um ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, escolha Ações.
4. Escolha Ambiente de clonagem.
5. Na página Clone environment (Clonar ambiente), analise as informações na seção Original Environment (Ambiente original) para verificar se escolheu o ambiente com base no qual você deseja criar um clone.
6. Na seção New Environment (Novo ambiente), é possível alterar os valores de Environment name (Nome do ambiente), Environment URL (URL do ambiente), Description (Descrição), Platform version (Versão da plataforma) e Service role (Função de serviço) definidos automaticamente pelo Elastic Beanstalk com base no ambiente original.

 Note

Se a versão da plataforma usada no ambiente original não for a recomendada para uso na ramificação da plataforma, você será avisado de que uma versão da plataforma diferente é recomendada. Escolha a Platform version (Versão da plataforma), e será possível ver a versão recomendada da plataforma na lista, por exemplo, 3.3.2 (Recommended) (3.3.2 (Recomendado)).

Elastic Beanstalk > Environments > GettingStartedApp-env > Clone environment

# Clone environment

You can launch a new environment based on an existing environment's configuration settings while optionally choosing a different platform version for the new environment. [Learn more](#)

## Original environment

Environment name

GettingStartedApp-env

Environment URL

GettingStartedApp-env.gap8pzvmti.us-east-2.elasticbeanstalk.com

Platform

Tomcat 8.5 with Java 8 running on 64bit Amazon Linux/3.3.1

## New environment

Environment name

GettingStartedApp-env-1

Environment URL

gettingstartedapp-env-1 .us-east-2.elasticbeanstal

Description


Clone of GettingStartedApp-env

Platform branch

Tomcat 8.5 with Java 8 running on 64bit Amazon Linux

Platform version

3.3.1

 **Warning**  
A different platform version is recommended.

Service role

arn:aws:iam::123456789012:role/aws-elasticbeanstalk-service-role

## 7. Quando estiver pronto, escolha Clone (Clonar).

### Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB)

Use o comando `eb clone` para clonar um ambiente em execução, da seguinte maneira.

```
~/workspace/my-app$ eb clone my-env1
Enter name for Environment Clone
(default is my-env1-clone): my-env2
Enter DNS CNAME prefix
(default is my-env1-clone): my-env2
```

Você pode especificar o nome do ambiente de origem no comando de clone ou omiti-lo para clonar o ambiente padrão referente à pasta do projeto atual. A EB CLI solicita que você insira um nome e prefixo DNS para o novo ambiente.

Por padrão, o `eb clone` cria o novo ambiente com a versão mais recente disponível da plataforma do ambiente de origem. Para forçar a EB CLI a usar a mesma versão, mesmo que haja uma versão mais recente disponível, use a opção `--exact`.

```
~/workspace/my-app$ eb clone --exact
```

Para obter mais informações sobre esse comando, consulte [eb clone](#).

### Encerrar um ambiente do Elastic Beanstalk

Você pode terminar um ambiente do AWS Elastic Beanstalk em execução usando o console do Elastic Beanstalk. Fazendo isso, você evita incorrer em cobranças por não recursos da AWS não utilizados.

#### Note

Você sempre pode iniciar um novo ambiente usando a mesma versão mais tarde.

Se você tiver dados de um ambiente que deseja manter, defina a política de exclusão de banco de dados como `Retain` antes de encerrar o ambiente. Isso mantém o banco de dados operacional fora do Elastic Beanstalk. Depois disso, qualquer ambiente do Elastic Beanstalk deve se conectar a ele

como um banco de dados externo. Se você quiser fazer backup dos dados sem manter o banco de dados operacional, defina a política de exclusão para fazer um snapshot do banco de dados antes de encerrar o ambiente. Para obter mais informações, consulte [Ciclo de vida do banco de dados](#) no capítulo Configuração de ambientes deste guia.

O Elastic Beanstalk pode falhar ao encerrar seu ambiente. Um motivo comum para isso é que outro grupo de segurança do ambiente tem uma dependência do grupo de segurança do ambiente que você está tentando terminar. Para obter instruções sobre como evitar esse problema, consulte [Grupos de segurança](#) na página Instâncias do EC2 deste guia.

### Important

Se você encerrar um ambiente, também deverá excluir quaisquer mapeamentos CNAME criados, pois outros clientes poderão reutilizar um nome de host disponível. Certifique-se de excluir os registros DNS que apontam para seu ambiente encerrado para evitar uma entrada de DNS pendente. Uma entrada de DNS pendente pode expor o tráfego da Internet destinado ao seu domínio a vulnerabilidades de segurança. Também pode apresentar outros riscos.

Para obter mais informações, consulte [Proteção contra um registro de delegação pendente no Route 53](#) no Guia do desenvolvedor do Amazon Route 53. Você também pode aprender mais sobre entradas de DNS pendentes em [Proteções de domínio aprimoradas para solicitações do Amazon CloudFront](#) no blog de segurança da AWS.

## Console do Elastic Beanstalk

Para encerrar um ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

**Note**

Quando você termina o ambiente, o CNAME associado ao ambiente terminado torna-se disponível para qualquer pessoa utilizar.

O Elastic Beanstalk leva alguns minutos para terminar os recursos da AWS em execução no ambiente.

## AWS CLI

Para encerrar um ambiente

- Execute o comando a seguir.

```
$ aws elasticbeanstalk terminate-environment --environment-name my-env
```

## API

Para encerrar um ambiente

- Chame `TerminateEnvironment` com o seguinte parâmetro:

```
EnvironmentName = SampleAppEnv
```

```
https://elasticbeanstalk.us-west-2.amazon.com/?EnvironmentName=SampleAppEnv
&Operation=TerminateEnvironment
&AuthParams
```

## Criação de ambientes do Elastic Beanstalk com a CLI da AWS

[Para obter detalhes sobre os comandos da AWS CLI do Elastic Beanstalk, consulte a Referência de comandos da AWS CLI.](#)

1. Verifique se o CNAME do ambiente está disponível.

```
$ aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
{
 "Available": true,
 "FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"
}
```

2. Verifique se a versão do aplicativo existe.

```
$ aws elasticbeanstalk describe-application-versions --application-name my-app --
version-label v1
```

Se você não tiver uma versão do aplicativo para sua origem, crie-a. Por exemplo, o comando a seguir cria uma versão da aplicação a partir de um pacote de origem no Amazon Simple Storage Service (Amazon S3).

```
$ aws elasticbeanstalk create-application-version --application-name my-app --
version-label v1 --source-bundle S3Bucket=DOC-EXAMPLE-BUCKET,S3Key=my-source-
bundle.zip
```

3. Crie um modelo de configuração para o aplicativo.

```
$ aws elasticbeanstalk create-configuration-template --application-name my-app --
template-name v1 --solution-stack-name "64bit Amazon Linux 2015.03 v2.0.0 running
Ruby 2.2 (Passenger Standalone)"
```

4. Crie o ambiente.

```
$ aws elasticbeanstalk create-environment --cname-prefix my-cname --application-
name my-app --template-name v1 --version-label v1 --environment-name v1clone --
option-settings file://options.txt
```

As configurações de opção são definidas no arquivo options.txt:

```
[
 {
 "Namespace": "aws:autoscaling:launchconfiguration",
 "OptionName": "IamInstanceProfile",
 "Value": "aws-elasticbeanstalk-ec2-role"
 }
]
```



A configuração da opção acima define o perfil da instância do IAM. É possível especificar o ARN ou o nome do perfil.

5. Determine se o novo ambiente está verde e pronto.

```
$ aws elasticbeanstalk describe-environments --environment-names my-env
```

Se o novo ambiente não estiver verde e pronto, decida se deseja repetir a operação ou deixar o ambiente em seu estado atual para investigação. Certifique-se de encerrar o ambiente depois que você terminar e de limpar todos os recursos não utilizados.

#### Note

É possível ajustar o tempo limite se o ambiente não iniciar em um período razoável.

## Criar ambientes do Elastic Beanstalk com a API

1. Chame `CheckDNSAvailability` com o seguinte parâmetro:

- `CNAMEPrefix = SampleApp`

#### Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?CNAMEPrefix=sampleapplication
&Operation=CheckDNSAvailability
&AuthParams
```

2. Chame `DescribeApplicationVersions` com os seguintes parâmetros:

- `ApplicationName = SampleApp`
- `VersionLabel = Version2`

#### Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&VersionLabel=Version2
&Operation=DescribeApplicationVersions
```

```
&AuthParams
```

3. Chame `CreateConfigurationTemplate` com os seguintes parâmetros:

- `ApplicationName = SampleApp`
- `TemplateName = MyConfigTemplate`
- `SolutionStackName = 64bit%20Amazon%20Linux%202015.03%20v2.0.0%20running%20Ruby%202.2%20(Passenger%20Standalone)`

#### Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&TemplateName=MyConfigTemplate
&Operation=CreateConfigurationTemplate
&SolutionStackName=64bit%20Amazon%20Linux%202015.03%20v2.0.0%20running%20Ruby
%202.2%20(Passenger%20Standalone)
&AuthParams
```

4. Chame `CreateEnvironment` com um dos seguintes conjuntos de parâmetros.

a. Use o seguinte em um nível de ambiente de servidor Web:

- `EnvironmentName = SampleAppEnv2`
- `VersionLabel = Version2`
- `Description = description`
- `TemplateName = MyConfigTemplate`
- `ApplicationName = SampleApp`
- `CNAMEPrefix = sampleapplication`
- `OptionSettings.member.1.Namespace = aws:autoscaling:launchconfiguration`
- `OptionSettings.member.1.OptionName = IamInstanceProfile`
- `OptionSettings.member.1.Value = aws-elasticbeanstalk-ec2-role`

#### Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&VersionLabel=Version2
```

```
&EnvironmentName=SampleAppEnv2
&TemplateName=MyConfigTemplate
&CNAMEPrefix=sampleapplication
&Description=description
&Operation=CreateEnvironment
&OptionSettings.member.1.Namespace=aws%3Aautoscaling%3Alaunchconfiguration
&OptionSettings.member.1.OptionName=IamInstanceProfile
&OptionSettings.member.1.Value=aws-elasticbeanstalk-ec2-role
&AuthParams
```

b. Use o seguinte em um nível de ambiente de operador:

- EnvironmentName = SampleAppEnv2
- VersionLabel = Version2
- Description = description
- TemplateName = MyConfigTemplate
- ApplicationName = SampleApp
- Tier = Worker
- OptionSettings.member.1.Namespace = aws:autoscaling:launchconfiguration
- OptionSettings.member.1.OptionName = IamInstanceProfile
- OptionSettings.member.1.Value = aws-elasticbeanstalk-ec2-role
- OptionSettings.member.2.Namespace = aws:elasticbeanstalk:sqs
- OptionSettings.member.2.OptionName = WorkerQueueURL
- OptionSettings.member.2.Value = sqsd.elasticbeanstalk.us-east-2.amazonaws.com
- OptionSettings.member.3.Namespace = aws:elasticbeanstalk:sqs
- OptionSettings.member.3.OptionName = HttpPath
- OptionSettings.member.3.Value = /
- OptionSettings.member.4.Namespace = aws:elasticbeanstalk:sqs
- OptionSettings.member.4.OptionName = MimeType
- OptionSettings.member.4.Value = application/json
- OptionSettings.member.5.Namespace = aws:elasticbeanstalk:sqs
- OptionSettings.member.5.OptionName = HttpConnections

- `OptionSettings.member.5.Value = 75`
- `OptionSettings.member.6.Namespace = aws:elasticbeanstalk:sqs`
- `OptionSettings.member.6.OptionName = ConnectTimeout`
- `OptionSettings.member.6.Value = 10`
- `OptionSettings.member.7.Namespace = aws:elasticbeanstalk:sqs`
- `OptionSettings.member.7.OptionName = InactivityTimeout`
- `OptionSettings.member.7.Value = 10`
- `OptionSettings.member.8.Namespace = aws:elasticbeanstalk:sqs`
- `OptionSettings.member.8.OptionName = VisibilityTimeout`
- `OptionSettings.member.8.Value = 60`
- `OptionSettings.member.9.Namespace = aws:elasticbeanstalk:sqs`
- `OptionSettings.member.9.OptionName = RetentionPeriod`
- `OptionSettings.member.9.Value = 345600`

## Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&VersionLabel=Version2
&EnvironmentName=SampleAppEnv2
&TemplateName=MyConfigTemplate
&Description=description
&Tier=Worker
&Operation=CreateEnvironment
&OptionSettings.member.1.Namespace=aws%3Aautoscaling%3Alaunchconfiguration
&OptionSettings.member.1.OptionName=IamInstanceProfile
&OptionSettings.member.1.Value=aws-elasticbeanstalk-ec2-role
&OptionSettings.member.2.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.2.OptionName=WorkerQueueURL
&OptionSettings.member.2.Value=sqs.elasticbeanstalk.us-east-2.amazonaws.com
&OptionSettings.member.3.Namespace=aws%3elasticbeanstalk%3sqs
&OptionSettings.member.3.OptionName=HttpPath
&OptionSettings.member.3.Value=%2F
&OptionSettings.member.4.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.4.OptionName=MimeType
&OptionSettings.member.4.Value=application%2Fjson
&OptionSettings.member.5.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.5.OptionName=HttpConnections
```

```
&OptionSettings.member.5.Value=75
&OptionSettings.member.6.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.6.OptionName=ConnectTimeout
&OptionSettings.member.6.Value=10
&OptionSettings.member.7.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.7.OptionName=InactivityTimeout
&OptionSettings.member.7.Value=10
&OptionSettings.member.8.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.8.OptionName=VisibilityTimeout
&OptionSettings.member.8.Value=60
&OptionSettings.member.9.Namespace=aws%3Aelasticbeanstalk%3Asqs
&OptionSettings.member.9.OptionName=RetentionPeriod
&OptionSettings.member.9.Value=345600
&AuthParams
```

## Construção de um URL "Iniciar agora"

Você pode criar um localizador uniforme de recursos (URL) personalizado para que qualquer pessoa possa implantar e executar rapidamente uma aplicação Web predeterminada no AWS Elastic Beanstalk. Esse URL é chamado URL "Iniciar agora". Você pode precisar de um URL de início imediato, por exemplo, para demonstrar uma aplicação Web criada para ser executada no Elastic Beanstalk. Com a URL Launch Now (Iniciar agora), você pode usar parâmetros para adicionar antecipadamente as informações necessárias ao assistente de criação de aplicativo. Quando adiciona essa informação no assistente, qualquer pessoa pode usar o link do URL para iniciar um ambiente do Elastic Beanstalk com a fonte da aplicação Web em apenas algumas etapas. Isso significa que os usuários não precisam carregar ou especificar manualmente o local do pacote de origem da aplicação. Também não é necessário fornecer nenhuma informação adicional ao assistente.

Um URL de início imediato fornece ao Elastic Beanstalk as informações mínimas necessárias para criar uma aplicação: o nome da aplicação, a pilha de soluções, o tipo de instância e o tipo de ambiente. O Elastic Beanstalk usa valores padrão para os outros detalhes de configuração que não são explicitamente especificados no URL personalizado de início imediato.

Um URL "Iniciar agora" usa a sintaxe de URL padrão. Para obter mais informações, consulte [RFC 3986 – URI \(Uniform Resource Identifier\): sintaxe genérica](#).

### Parâmetros do URL

O URL deve conter os seguintes parâmetros, que diferenciam maiúsculas de minúsculas:

- **region (região):** especifique uma região da AWS Para obter uma lista de regiões compatíveis com o Elastic Beanstalk, consulte [AWS Elastic BeanstalkEndpoints e cotas](#) no Referência geral da AWS.
- **applicationName:** especifique o nome da aplicação. O Elastic Beanstalk exibe o nome da aplicação no console do Elastic Beanstalk para distingui-la de outras aplicações. Por padrão, o nome do aplicativo também forma a base do nome e URL do ambiente.
- **platform (plataforma):** especifique a versão de plataforma a ser usada para o ambiente. Use um dos métodos a seguir e codifique sua escolha com URL:
  - Especifique um ARN de plataforma sem uma versão. O Elastic Beanstalk seleciona a versão da plataforma mais recente da versão principal da plataforma correspondente. Por exemplo, para selecionar a versão mais recente da plataforma Python 3.6, especifique `Python 3.6 running on 64bit Amazon Linux`.
  - Especifique o nome da plataforma. O Elastic Beanstalk seleciona a versão mais recente do tempo de execução mais recente do idioma da plataforma (por exemplo, Python).

Para obter uma descrição de todas as plataformas disponíveis e suas versões, consulte [Plataformas compatíveis com Elastic Beanstalk](#).

Você pode usar a [AWS Command Line Interface](#) (AWS CLI) para obter uma lista de todas as versões da plataforma disponíveis com seus respectivos ARNs. O comando `list-platform-versions` lista informações detalhadas sobre todas as versões da plataforma disponíveis. Use o argumento `--filters` para limitar o escopo da lista. Por exemplo, você pode listar todas as versões da plataforma de um idioma específico.

O exemplo a seguir consulta todas as versões de plataforma Python e canaliza a saída por meio de uma série de comandos. O resultado é uma lista de ARNs de versão da plataforma (sem o final `/version`), em formato legível, sem codificação de URL.

```
$ aws elasticbeanstalk list-platform-versions --filters
 'Type="PlatformName",Operator="contains",Values="Python"' | grep PlatformArn | awk -
F '"" '{print $4}' | awk -F '/' '{print $2}'
Preconfigured Docker - Python 3.4 running on 64bit Debian
Preconfigured Docker - Python 3.4 running on 64bit Debian
Python 2.6 running on 32bit Amazon Linux
Python 2.6 running on 32bit Amazon Linux 2014.03
...
Python 3.6 running on 64bit Amazon Linux
```

O exemplo a seguir adiciona um comando Perl ao último exemplo, para codificar a saída com URL.

```
$ aws elasticbeanstalk list-platform-versions --filters
 'Type="PlatformName",Operator="contains",Values="Python"' | grep PlatformArn | awk
-F '""' '{print $4}' | awk -F '/' '{print $2}' | perl -MURI::Escape -ne 'chomp;print
uri_escape($_), "\n"'
Preconfigured%20Docker%20-%20Python%203.4%20running%20on%2064bit%20Debian
Preconfigured%20Docker%20-%20Python%203.4%20running%20on%2064bit%20Debian
Python%202.6%20running%20on%2032bit%20Amazon%20Linux
Python%202.6%20running%20on%2032bit%20Amazon%20Linux%202014.03
...
Python%203.6%20running%20on%2064bit%20Amazon%20Linux
```

Um URL "Iniciar agora" pode, opcionalmente, conter os parâmetros a seguir. Se você não incluir os parâmetros opcionais em seu URL Launch Now (Iniciar agora), o Elastic Beanstalk usará valores padrão para criar e executar a aplicação. Quando o parâmetro `sourceBundleUrl` é incluído, o Elastic Beanstalk usa a aplicação demonstrativa padrão para a plataforma (plataforma) especificada.

- `sourceBundleUrl`: especifique o local do pacote de origem da aplicação Web no formato de URL. Por exemplo, se você fez upload do pacote de origem em um bucket do Amazon S3, poderá especificar o valor do parâmetro `sourceBundleUrl` como `https://mybucket.s3.amazonaws.com/myobject`.

#### Note

Você pode especificar o valor do parâmetro `sourceBundleUrl` como um URL HTTP, mas o navegador da Web do usuário converterá caracteres, conforme necessário, aplicando a codificação de URL HTML.

- `environmentType`: especifique se o ambiente tem balanceamento de carga e é escalável ou se é de apenas uma única instância. Para obter mais informações, consulte [Tipos de ambiente](#). Você pode especificar `LoadBalancing` ou `SingleInstance` como o valor do parâmetro.
- `tierName`: especifique se o ambiente é compatível com uma aplicação Web que processe solicitações da Web ou com uma aplicação Web que execute trabalhos em segundo plano. Para obter mais informações, consulte [Ambientes de operador do Elastic Beanstalk](#). Você pode especificar `WebServer` ou `Worker`,

- `instanceType`: especifique um servidor com as características (incluindo o tamanho da memória e a capacidade da CPU) mais adequadas à aplicação. Para obter mais informações sobre as famílias e os tipos de instância do Amazon EC2, consulte [Tipos de instância](#), no Guia do usuário do Amazon EC2 para instâncias do Linux ou [Tipos de instância](#) no Guia do usuário do Amazon EC2 para instâncias do Windows. Para obter mais informações sobre os tipos de instâncias disponíveis nas regiões, consulte [Tipos de instância](#) no Guia do usuário do Amazon EC2 para instâncias do Linux ou [tipos de instância](#) no Guia do usuário do Amazon EC2 para instâncias do Windows.
- `withVpc`: especifique se o ambiente será criado em uma Amazon VPC. Você pode especificar `true` ou `false`. Para obter mais informações sobre como usar o Elastic Beanstalk com a Amazon VPC, consulte [Usar o Elastic Beanstalk com Amazon VPC](#).
- `withRds`: especifique uma instância do banco de dados do Amazon RDS será criada com esse ambiente. Para obter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon RDS](#). Você pode especificar `true` ou `false`.
- `rdsDBEngine`: especifique o mecanismo de banco de dados que deseja usar em suas instâncias do Amazon EC2 nesse ambiente. Especifique `mysql`, `oracle-se1`, `sqlserver-ex`, `sqlserver-web` ou `sqlserver-se`. O valor padrão é `mysql`.
- `rdsDBAllocatedStorage`: especifique o tamanho alocado de armazenamento do banco de dados em gigabytes (GB). Especifique os seguintes valores:
  - MySQL: 5 a 1024. O padrão é 5.
  - Oracle: 10 a 1024. O padrão é 10.
  - Microsoft SQL Server Express Edition – 30.
  - Microsoft SQL Server Web Edition – 30.
  - Microsoft SQL Server Standard Edition – 200.
- `rdsDBInstanceClass`: especifique o tipo de instância de banco de dados. O valor padrão é `db.t2.micro` (`db.m1.large` é para um ambiente que não esteja em execução em uma Amazon VPC). Para obter uma lista de classes de instância de banco de dados suportadas pelo Amazon RDS, consulte [DB Instance Class](#) (Classe de instância de banco de dados) no Amazon Relational Database Service User Guide (Guia do usuário do Amazon Relational Database Service).
- `rdsMultiAZDatabase`: especifique se o Elastic Beanstalk precisa criar a instância de banco de dados em várias zonas de disponibilidade. Você pode especificar `true` ou `false`. Para obter mais informações sobre implantações em várias zonas de disponibilidade com o Amazon RDS, acesse [Regions and Availability Zones](#) (Regiões e zonas de disponibilidade) no Amazon Relational Database Service User Guide (Guia do usuário do Amazon Relational Database Service).



- `rdsDBDeletionPolicy`: especifique se deseja excluir ou criar um snapshot da instância de banco de dados no encerramento do ambiente. Você pode especificar `Delete` ou `Snapshot`.

## Exemplo

Um exemplo de URL "Iniciar agora" está apresentado a seguir. Depois de criar seu próprio URL, você pode fornecê-lo aos usuários. Por exemplo, você pode incorporar a URL em uma página da Web ou em materiais de treinamento. Quando os usuários criam uma aplicação usando o URL Launch Now (Iniciar agora), o assistente de criação de aplicação do Elastic Beanstalk não requer entrada adicional.

```
https://console.aws.amazon.com/elasticbeanstalk/home?region=us-west-2#/newApplication?applicationName=YourCompanySampleApp&platform=PHP%207.3%20running%20on%2064bit%20Amazon%20Linux&sourceBundleUrl=http://s3.amazonaws.com/mybucket/myobject&environmentType=SingleInstance&tierName=WebServer&instanceType=m1.small&withVpc=true&
```

Quando os usuários escolhem um URL Launch Now (Iniciar agora), o Elastic Beanstalk exibe uma página semelhante à seguinte.



## Create a web app

Create a new application and environment with a sample application or your own code. By creating an environment, you allow AWS Elastic Beanstalk to manage AWS resources and permissions on your behalf. [Learn more](#)

### Application information

**Application name**

Up to 100 Unicode characters, not including forward slash (/).

### Environment information

Choose the name, subdomain, and description for your environment. These cannot be changed later.

**Environment name**

**Domain**

**Description**

### Base configuration

**Tier**  Web Server ([Choose tier](#))

**Platform**  Preconfigured platform

Platforms published and maintained by AWS Elastic Beanstalk.

Custom platform <sup>NEW</sup>

Platforms created and owned by you. [Learn more](#)

**Application code**  Sample application

Get started right away with sample code.

Upload your code

Upload a source bundle from your computer or copy one from Amazon S3.

ZIP or WAR

## Para usar o URL "Iniciar agora"

1. Escolha a URL Launch Now (Iniciar agora).
2. Quando o console do Elastic Beanstalk é aberto, na página Create a web app (Criar uma aplicação Web), escolha Review and launch (Revisar e iniciar) para visualizar as configurações que o Elastic Beanstalk usa para criar a aplicação e iniciar o ambiente no qual a aplicação é executada.
3. Na página Configure (Configurar), clique em Create app (Criar aplicativo) para criar o aplicativo.

## Criar e atualizar grupos de ambientes do Elastic Beanstalk

Com a API `Compose Environments` do AWS Elastic Beanstalk, é possível criar e atualizar grupos de ambientes do Elastic Beanstalk em uma única aplicação. Cada ambiente do grupo pode executar um componente separado de um aplicativo de arquitetura orientada a serviços. A API `Compose Environments` usa uma lista de versões do aplicativo e um nome de grupo opcional. O Elastic Beanstalk cria um ambiente para cada versão do aplicativo ou, se os ambientes já existem, implanta as versões do aplicativo neles.

Crie links entre os ambientes do Elastic Beanstalk para designar um ambiente como uma dependência do outro. Quando você cria um grupo de ambientes com a API `Compose Environments`, o Elastic Beanstalk cria ambientes dependentes somente depois que suas dependências estão em execução. Para obter mais informações sobre links de ambiente, consulte [Criar links entre ambientes do Elastic Beanstalk](#).

A API `Compose Environments` usa um [manifesto de ambiente](#) para armazenar detalhes de configuração que são compartilhados por grupos de ambientes. Cada aplicativo do componente deve ter um arquivo de configuração `env.yaml` em seu pacote de origem do aplicativo que especifique os parâmetros usados para criar o ambiente.

`Compose Environments` requer que `EnvironmentName` e `SolutionStack` sejam especificados no manifesto de ambiente para cada aplicativo do componente.

Use a API `Compose Environments` com a Elastic Beanstalk Command Line Interface (EB CLI), a AWS CLI ou um SDK. Para ver as instruções da EB CLI, consulte [Gerenciar vários ambientes do Elastic Beanstalk como um grupo com a CLI do EB](#).

## Uso da API **Compose Environments**

Por exemplo, você pode criar uma aplicação chamada `Media Library` que permita aos usuários fazer upload e gerenciar imagens e vídeos armazenados no Amazon Simple Storage Service (Amazon S3). O aplicativo tem um ambiente de front-end, `front`, que executa um aplicativo Web. Isso permite aos usuários fazer upload e download de arquivos individuais, visualizar sua biblioteca e iniciar trabalhos de processamento em lote.

Em vez de processar os trabalhos diretamente, a aplicação de front-end adiciona trabalhos a uma fila do Amazon SQS. O segundo ambiente, `worker`, obtém trabalhos da fila e os processa. O `worker` usa um tipo de instância G2, que tem uma GPU de alta performance e, ao mesmo tempo, `front` pode ser executado em um tipo de instância genérico mais econômico.

Organize a pasta do projeto, `Media Library`, em diretórios separados para cada componente, com cada diretório contendo um arquivo de definição de ambiente (`env.yaml`) com o código-fonte de cada:

```
~/workspace/media-library
|-- front
| `-- env.yaml
`-- worker
 `-- env.yaml
```

As seguintes listas mostram o arquivo `env.yaml` para cada aplicativo do componente.

### **~/workspace/media-library/front/env.yaml**

```
EnvironmentName: front+
EnvironmentLinks:
 "WORKERQUEUE" : "worker+"
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentTier:
 Name: WebServer
 Type: Standard
SolutionStack: 64bit Amazon Linux 2015.09 v2.0.4 running Java 8
OptionSettings:
 aws:autoscaling:launchconfiguration:
 InstanceType: m4.large
```

### **~/workspace/media-library/worker/env.yaml**

```

EnvironmentName: worker+
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentTier:
 Name: Worker
 Type: SQS/HTTP
SolutionStack: 64bit Amazon Linux 2015.09 v2.0.4 running Java 8
OptionSettings:
 aws:autoscaling:launchconfiguration:
 InstanceType: g2.2xlarge

```

Depois de [criar uma versão do aplicativo](#) para os componentes do aplicativo de front-end (front-v1) e operador (worker-v1), chame a API Compose Environments com os nomes de versão. Neste exemplo, usamos a AWS CLI para chamar a API.

```

Create application versions for each component:
~$ aws elasticbeanstalk create-application-version --application-name media-
library --version-label front-v1 --process --source-bundle S3Bucket="DOC-EXAMPLE-
BUCKET",S3Key="front-v1.zip"
{
 "ApplicationVersion": {
 "ApplicationName": "media-library",
 "VersionLabel": "front-v1",
 "Description": "",
 "DateCreated": "2015-11-03T23:01:25.412Z",
 "DateUpdated": "2015-11-03T23:01:25.412Z",
 "SourceBundle": {
 "S3Bucket": "DOC-EXAMPLE-BUCKET",
 "S3Key": "front-v1.zip"
 }
 }
}
~$ aws elasticbeanstalk create-application-version --application-name media-
library --version-label worker-v1 --process --source-bundle S3Bucket="DOC-EXAMPLE-
BUCKET",S3Key="worker-v1.zip"
{
 "ApplicationVersion": {
 "ApplicationName": "media-library",
 "VersionLabel": "worker-v1",
 "Description": "",
 "DateCreated": "2015-11-03T23:01:48.151Z",
 "DateUpdated": "2015-11-03T23:01:48.151Z",
 "SourceBundle": {

```

```
 "S3Bucket": "DOC-EXAMPLE-BUCKET",
 "S3Key": "worker-v1.zip"
 }
}
}
Create environments:
~$ aws elasticbeanstalk compose-environments --application-name media-library --group-name dev --version-labels front-v1 worker-v1
```

A terceira chamada cria dois ambientes, `front-dev` e `worker-dev`. A API cria os nomes dos ambientes ao concatenar o `EnvironmentName` especificado no arquivo `env.yml` com a opção `group name` especificada na chamada `Compose Environments`, separados por um hífen. O tamanho total dessas duas opções e o hífen não deve exceder o máximo permitido de 23 caracteres do nome do ambiente.

A aplicação em execução no ambiente `front-dev` pode acessar o nome da fila do Amazon SQS anexada ao ambiente `worker-dev` lendo a variável `WORKERQUEUE`. Para obter mais informações sobre links de ambiente, consulte [Criar links entre ambientes do Elastic Beanstalk](#).

## Implantar aplicações em ambientes do Elastic Beanstalk

Use o console AWS Elastic Beanstalk para fazer upload de um [pacote de fonte](#) atualizado e implante-o em seu ambiente do Elastic Beanstalk ou implante novamente uma versão carregada anteriormente.

Cada implantação é identificada por um ID. Os IDs de implantação começam em 1, com incrementos de um a cada implantação e alteração de configuração de instância. Se você habilitar os [relatórios avançados de integridade](#), o Elastic Beanstalk exibirá o ID da implantação no [console de integridade](#) e na [CLI do EB](#) ao relatar o status de integridade da instância. O ID de implantação ajuda a determinar o estado do ambiente quando uma atualização contínua falha.

O Elastic Beanstalk fornece várias políticas e configurações de implantação. Para obter detalhes sobre como configurar uma política e configurações adicionais, consulte [the section called “Opções de implantação”](#). A tabela a seguir lista as políticas e os tipos de ambientes compatíveis com elas.

## Políticas de implantação compatíveis

Política de implantação	Ambientes de carga equilibrada	Ambientes de instância única	Ambientes legados do Windows Server†
Tudo de uma vez	✓ Sim	✓ Sim	✓ Sim
Contínua	✓ Sim	× Não	✓ Sim
Contínua com lote adicional	✓ Sim	× Não	× Não
Imutável	✓ Sim	✓ Sim	× Não
Divisão de tráfego	✓ Sim (Application Load Balancer)	× Não	× Não

† Nessa tabela, um Ambiente legado do Windows Server é um ambiente baseado em uma [configuração de plataforma do Windows Server](#) que usa uma versão anterior do IIS 8.5.

### Warning

Algumas políticas substituem todas as instâncias durante a implantação ou a atualização. Por conta disso, todo o [equilíbrio de intermitência do Amazon EC2](#) é perdido. Isso acontece nos seguintes casos:

- Atualizações de plataforma gerenciada com substituição de instância habilitada
- Atualizações imutáveis
- Implantações com atualizações imutáveis ou divisão de tráfego habilitada

## Como escolher uma política de implantação

Escolher a política de implantação correta para seu aplicativo é uma questão com algumas considerações a fazer e que depende de suas necessidades específicas. A página [the section called “Opções de implantação”](#) tem mais informações sobre cada política e uma descrição detalhada do funcionamento de algumas delas.

A lista a seguir fornece informações resumidas sobre as diferentes políticas de implantação e inclui considerações relacionadas.



- **All at once (Tudo de uma vez):** o método de implantação mais rápido. Adequado se você puder aceitar uma pequena perda de serviço e se implantações rápidas forem importantes para você. Com esse método, o Elastic Beanstalk implanta a nova versão da aplicação em cada instância. Em seguida, talvez o proxy da web ou o servidor do aplicativo precise reiniciar. Como resultado, seu aplicativo pode ficar indisponível para usuários (ou ter baixa disponibilidade) por um breve período.
- **Rolling (Contínua):** evita o tempo de inatividade e minimiza a disponibilidade reduzida, a um custo de implantação mais longo. Adequado se você não puder aceitar nenhum período de perda total de serviço. Com esse método, seu aplicativo é implantado no ambiente um lote de instâncias por vez. A maior parte da largura de banda é mantida durante toda a implantação.
- **Rolling with additional batch (Contínua com lote adicional):** evita qualquer disponibilidade reduzida, ao custo de um tempo de implantação ainda mais longo em comparação com o método Rolling (Contínuo). Adequado se você precisar manter a mesma largura de banda durante toda a implantação. Com esse método, o Elastic Beanstalk inicia um lote extra de instâncias e executa uma implantação contínua. A inicialização do lote extra leva tempo e garante que a mesma largura de banda seja mantida durante toda a implantação.
- **Immutable (Imutável):** um método de implantação mais lento, que garante que a nova versão da aplicação seja sempre implantada em novas instâncias, em vez de atualizar instâncias existentes. Ele também tem a vantagem adicional de uma reversão rápida e segura em caso de falha na implantação. Com esse método, o Elastic Beanstalk executa uma [atualização imutável](#) para implantar sua aplicação. Em uma atualização imutável, um segundo grupo de Auto Scaling é iniciado no ambiente e a nova versão atende o tráfego junto com a versão antiga até que a nova instância passe nas verificações de integridade.
- **Traffic splitting (Divisão de tráfego):** um método de implantação de testes canary. Adequado se você deseja testar a integridade da sua nova versão do aplicativo usando uma parte do tráfego recebido, mantendo o restante do tráfego atendido pela versão antiga do aplicativo.

A tabela a seguir compara as propriedades dos métodos de implantação.



## Métodos de implantação

Método	Impacto de uma implantação malsucedida	Tempo de implantação	Tempo de inatividade de zero	Nenhuma alteração de DNS	Processo de reversão	Código implantado em
Tudo de uma vez	Tempo de inatividade	⊕	X Não	✓ Sim	Reimplantação manual	Instâncias existentes
Contínua	Lote único fora de serviço; qualquer lote bem-sucedido antes de uma falha ao executar uma nova versão do aplicativo	⊕	⊖ ✓ Sim	✓ Sim	Reimplantação manual	Instâncias existentes
Contínua com lote adicional	Mínimo, se o primeiro lote falhar; do contrário, é semelhante à Rolling (Contínua)	⊕	⊖ ✓ Sim	✓ Sim	Reimplantação manual	Instâncias novas e existentes
Imutável	Mínimo	⊕	⊖ ✓ Sim	✓ Sim	Encerra novas instâncias	Instâncias novas
Divisão de tráfego	Percentual de tráfego do cliente roteado para a nova versão afetada temporariamente	⊕	⊖ ✓ Sim	✓ Sim	Redirecionar tráfego e encerra novas instâncias	Instâncias novas

Método	Impacto de uma implantação malsucedida	Tempo de implantação	Tempo de inatividade de zero	Nenhuma alteração de DNS	Processo de reversão	Código implantado em
Azul/verde	Mínimo		 ✓ Sim	X Não	Alternar URL	Instâncias novas

† Varia, dependendo do tamanho do lote.

†† Varia de acordo com a configuração da opção de tempo momento de avaliação.

## Implantar uma nova versão do aplicativo

Você pode executar as implantações pelo painel do ambiente.

Como implantar uma nova versão da aplicação em um ambiente do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.


3. Escolha Upload and deploy (Fazer upload e implantar).
4. Use o formulário na tela para carregar o pacote de origem do aplicativo.
5. Escolha Implantar.

## Reimplantar a versão anterior

Você também pode implantar uma versão do aplicativo carregado anteriormente em qualquer um dos ambientes da página de versões do aplicativo.

Para implantar uma versão existente do aplicativo no ambiente atual

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

 Note

Se você tiver muitos aplicativos, use a barra de pesquisa para filtrar a lista de aplicativos.

3. No painel de navegação, encontre o nome do aplicativo e escolha Application versions (Versões do aplicativo).
4. Selecione a versão do aplicativo a ser implantada.
5. Escolha Actions (Ações) e escolha Deploy (Implantar).
6. Escolha um ambiente e escolha Deploy (Implantar).

## Outras formas de implantar seu aplicativo

Se você implanta com frequência, considere o uso da [Interface de Linhas de Comando do Elastic Beanstalk](#) (CLI do EB) para gerenciar os ambientes. A CLI do EB cria um repositório junto com o código-fonte. Ela também pode criar um pacote de origem, carregá-lo no Elastic Beanstalk e implantá-lo com um único comando.

Para implantações dependentes das alterações da configuração dos recursos ou de uma nova versão que não pode ser executada junto com a versão antiga, inicie um novo ambiente com a nova versão e faça uma troca de CNAME para uma [implantação azul/verde](#).

## Políticas e configurações de implantação

O AWS Elastic Beanstalk oferece várias opções de processamento das [implantações](#), incluindo as políticas de implantação (Tudo ao mesmo tempo, Contínua, Contínua com batch adicional, Imutável e Divisão de tráfego) e opções que permitem configurar o tamanho do lote e o comportamento da verificação de integridade durante as implantações. Por padrão, o ambiente usa implantações de uma só vez. Se você criou o ambiente com a CLI do EB e se ele for um ambiente de escalável (você não especificou a opção `--single`), ele usa implantações contínuas.

Com implantações contínuas, o Elastic Beanstalk divide as instâncias do Amazon EC2 do ambiente em lotes e implanta a nova versão da aplicação em um lote por vez. Ele deixa o restante das

instâncias no ambiente executando a versão antiga do aplicativo. Durante uma implantação contínua, algumas instâncias atenderão a solicitações com a versão antiga do aplicativo, enquanto as instâncias em lotes completos atenderão a outras solicitações com a nova versão. Para obter mais detalhes, consulte [the section called “Como funcionam as implantações gradativas”](#).

Para manter a capacidade total durante as implantações, você pode configurar o ambiente para iniciar um novo lote de instâncias antes de tirar qualquer instância de serviço. Essa opção é conhecida como uma implantação contínua com um lote adicional. Quando a implantação for concluída, o Elastic Beanstalk encerrará os lotes adicionais de instâncias.

As implantações imutáveis executam uma [atualização imutável](#) para iniciar um conjunto completo de novas instâncias executando a nova versão da aplicação em outro grupo de Auto Scaling junto com as instâncias que executam a versão antiga. As implantações imutáveis podem impedir problemas causados por implantações contínuas parcialmente concluídas. Se as novas instâncias não passarem pela verificação de integridade, o Elastic Beanstalk as encerrará, deixando as instâncias originais inalteradas.

As implantações de divisão de tráfego permitem executar testes canary como parte da implantação de aplicativos. Em uma implantação de divisão de tráfego, o Elastic Beanstalk executa um conjunto completo de novas instâncias, como durante uma implantação imutável. Depois, ele encaminha uma porcentagem especificada de tráfego de entrada do cliente para a nova versão do aplicativo por um período de avaliação especificado. Se as novas instâncias permanecerem íntegras, o Elastic Beanstalk encaminhará todo o tráfego para elas e encerrará as antigas. Se as novas instâncias não passarem nas verificações de integridade ou se você optar por interromper a implantação, o Elastic Beanstalk moverá o tráfego de volta para as instâncias antigas e encerrará as novas. Nunca há interrupção de serviço. Para obter mais detalhes, consulte [the section called “Como funcionam as implantações de divisão de tráfego”](#).

#### Warning

Algumas políticas substituem todas as instâncias durante a implantação ou a atualização. Por conta disso, todo o [equilíbrio de intermitência do Amazon EC2](#) é perdido. Isso acontece nos seguintes casos:

- Atualizações de plataforma gerenciada com substituição de instância habilitada
- Atualizações imutáveis
- Implantações com atualizações imutáveis ou divisão de tráfego habilitada

Se o seu aplicativo não passar em todas as verificações de integridade, mas ainda estiver funcionando corretamente em um status de integridade inferior, você poderá permitir que as instâncias passem pelas verificações de integridade com um status inferior, como Warning, modificando a opção Limite de integridade. Se suas implantações falham por não passarem nas verificações de integridade e você precisar forçar uma atualização, independentemente do status de integridade, especifique a opção Ignorar verificação de integridade.

Ao especificar um tamanho de lote para atualizações constantes, o Elastic Beanstalk também usa esse valor para implantar reinicializações de aplicações. Use as reinicializações contínuas quando precisar reiniciar o proxy e os servidores de aplicativos em execução em suas instâncias do ambiente sem tempo de inatividade.

## Configurar implantações de aplicativos

No [console de gerenciamento do ambiente](#), ative e configure as implantações em lote de versão do aplicativo editando Atualizações e implantações na página Configuração do ambiente.

Para configurar as implantações (console)

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Rolling updates and deployments (Atualizações e implantações contínuas), escolha Edit (Editar).
5. Na seção Application Deployments (Implantações de aplicativos), escolha uma Deployment policy (Política de implantação), as configurações de lote e as opções de verificação de integridade.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A seção Implantações de aplicativos da página Atualizações e implantações contínuas tem as seguintes opções para implantações de aplicativos:

- **Deployment policy (Política de implantação):** escolha uma das seguintes opções de implantação:
  - **All at once (Todos de uma vez):** implante a nova versão em todas as instâncias simultaneamente. Todas as instâncias em seu ambiente ficam fora de serviço por um curto período durante a implantação.
  - **Rolling (Contínua):** implante a nova versão em lotes. Cada lote é retirado de serviço durante a fase de implantação, reduzindo a capacidade de seu ambiente de acordo com o número de instâncias em um lote.
  - **Rolling with additional batch (Contínua com lote adicional):** implante a nova versão em lotes, mas primeiro inicie um novo lote de instâncias para garantir a capacidade total durante o processo de implantação.
  - **Immutable (Imutável):** implante a nova versão em um novo grupo de instâncias executando uma [atualização imutável](#).
  - **Traffic splitting (Divisão de tráfego):** implante a nova versão em um novo grupo de instâncias e divida temporariamente o tráfego de entrada do cliente entre a versão existente da aplicação e a nova.

Para as políticas Contínua e Contínua com lote adicional, você pode configurar:

- **Batch size (Tamanho do lote):** o tamanho do conjunto de instâncias para implantar em cada lote.

Escolha **Percentage (Porcentagem)** para configurar uma porcentagem do número total de instâncias do EC2 no grupo de Auto Scaling (até 100%), ou escolha **Fixed (Fixo)** para configurar um número fixo de instâncias (até a contagem máxima de instâncias na configuração de Auto Scaling do ambiente).

Para a política de implantação de divisão de tráfego, você pode fazer a seguinte configuração:

- **Traffic split (Divisão de tráfego):** a porcentagem inicial de tráfego de entrada do cliente que o Elastic Beanstalk desloca para instâncias de ambiente executando a nova versão da aplicação que está sendo implantada.
- **Traffic splitting evaluation time (Tempo de avaliação da divisão de tráfego):** o período, em minutos, que o Elastic Beanstalk aguarda após uma implantação íntegra inicial antes de prosseguir para deslocar todo o tráfego de entrada de cliente para a nova versão da aplicação que está sendo implantada.

Elastic Beanstalk > Environments > GettingStartedApp-env > Configuration

## Modify rolling updates and deployments

### Application deployments

Choose how AWS Elastic Beanstalk propagates source code changes and software configuration updates. [Learn more](#)

Deployment policy

All at once

Batch size:

Percentage

Fixed

100 % of instances at a time

Traffic split

10 % to new application version

Traffic splitting evaluation time

5 minutes

A seção Deployment preferences (Preferências de implantação) contém opções relacionadas a verificações de integridade.

- Ignore health check (Ignorar verificação de integridade): impede que uma implantação seja revertida quando um lote não fica íntegro dentro do Command timeout (Tempo limite do comando).
- Healthy threshold (Limite de integridade): reduz o limite no qual uma instância é considerada íntegra durante as implantações e atualizações contínuas e as atualizações imutáveis.
- Command timeout (Tempo limite do comando): o número de segundos de espera para que uma instância se torne íntegra antes de cancelar a implantação ou, se a opção Ignore health check (Ignorar verificação de integridade) estiver definida, prosseguir para o próximo lote.

**Deployment preferences**  
Customize health check requirements and deployment timeouts.

**Ignore health check**  
False  
Don't fail deployments due to health check failures.

**Healthy threshold**  
Ok  
Lower the threshold for an instance in a batch to pass health checks during an update or deployment.

**Command timeout**  
600  
Change the amount of time in seconds that AWS Elastic Beanstalk allows an instance to complete deployment commands.

## Como funcionam as implantações gradativas

Ao processar um lote, o Elastic Beanstalk separa todas as instâncias no lote do balanceador de carga, implanta a nova versão da aplicação e, depois, anexa as instâncias novamente. Se você tiver habilitado a [diminuição de conexão](#), o Elastic Beanstalk diminuirá as conexões existentes das instâncias do Amazon EC2 em cada lote antes de iniciar a implantação.

Depois de anexar novamente as instâncias em um lote ao balanceador de carga, o Elastic Load Balancing espera até que elas passem em um número mínimo de verificações de integridade do Elastic Load Balancing (o valor Healthy check count threshold (Limite de contagem de verificações de integridade)) e, depois, inicia o roteamento de tráfego para elas. Se nenhum [URL de verificação de integridade](#) estiver configurado, isso poderá acontecer muito rapidamente, porque uma instância passará pela verificação de integridade assim que puder aceitar uma conexão TCP. Se um URL de verificação de integridade estiver configurado, o load balancer não roteará o tráfego para as instâncias atualizadas até que retornem um código de status 200 OK em resposta a uma solicitação HTTP GET para o URL de verificação de integridade.

O Elastic Beanstalk aguarda até que todas as instâncias em um lote estejam íntegras antes de avançar para o próximo lote. Com os [relatórios básicos de integridade](#), a integridade da instância depende do status da verificação de integridade do Elastic Load Balancing. Quando todas as instâncias no lote passarem em verificações de integridade suficientes para serem consideradas íntegras pelo Elastic Load Balancing, o lote estará concluído. Se os [relatórios aprimorados de integridade](#) estiverem habilitados, o Elastic Beanstalk considerará vários outros fatores, incluindo o



resultado de solicitações recebidas. Com os relatórios de integridade avançados, todas as instâncias devem passar por 12 verificações de integridade consecutivas com [status OK](#) dentro de dois minutos para ambientes de servidor web e 18 verificações de integridade dentro de três minutos para ambientes de operador.

Se um lote de instâncias não se tornar íntegro dentro do [tempo limite do comando](#), a implantação falhará. Depois de uma implantação com falha, [verifique a integridade das instâncias no ambiente](#) para obter informações sobre a causa da falha. Em seguida, execute outra implantação com uma versão corrigida ou comprovadamente boa de seu aplicativo para reverter.

Se uma implantação falhar após um ou mais lotes serem concluídos com êxito, os lotes concluídos executar a nova versão de seu aplicativo enquanto os eventuais lotes pendentes continuam a executar a versão antiga. Você pode identificar a versão em execução nas instâncias em seu ambiente na [página de integridade](#) no console. Essa página exibe o ID da implantação mais recente executada em cada instância em seu ambiente. Se você encerrar instâncias a partir da implantação com falha, o Elastic Beanstalk as substituirá por instâncias executando a versão da aplicação a partir da implantação bem-sucedida mais recente.

## Como funcionam as implantações de divisão de tráfego

As implantações de divisão de tráfego permitem a execução de testes canary. Você direciona algum tráfego de entrada do cliente para a nova versão do aplicativo de modo a verificar a integridade do aplicativo antes de confirmar a nova versão e direcionar todo o tráfego para ele.

Durante uma implantação de divisão de tráfego, o Elastic Beanstalk cria um conjunto de instâncias em um grupo de Auto Scaling temporário separado. Depois, o Elastic Beanstalk instrui o balanceador de carga a direcionar uma certa porcentagem do tráfego de entrada do ambiente para as novas instâncias. Depois, durante um período configurado, o Elastic Beanstalk rastreia a integridade do novo conjunto de instâncias. Se tudo estiver bem, o Elastic Beanstalk vai deslocar o tráfego restante para as novas instâncias e anexá-las ao grupo de Auto Scaling original do ambiente, substituindo as instâncias antigas. Depois, o Elastic Beanstalk limpa, ou seja, encerra as instâncias antigas e remove o grupo temporário de Auto Scaling.

### Note

A capacidade do ambiente não muda durante uma implantação de divisão de tráfego. O Elastic Beanstalk inicia o mesmo número de instâncias no grupo temporário que há no grupo original de Auto Scaling no momento em que a implantação é iniciada. Ele mantém um número constante de instâncias nos dois grupos de Auto Scaling durante a implantação.

Leve esse fato em consideração ao definir o tempo de avaliação da divisão de tráfego do ambiente.

A reversão da implantação para a versão anterior do aplicativo é rápida e não afeta o serviço para o tráfego do cliente. Se as novas instâncias não passarem nas verificações de integridade ou se você optar por interromper a implantação, o Elastic Beanstalk moverá o tráfego de volta para as instâncias antigas e encerrará as novas. É possível cancelar implantações usando a página de visão geral do ambiente no console do Elastic Beanstalk e escolher **Abort current operation** (Anular operação atual) em **Environment actions** (Ações de ambiente). Também é possível chamar a API [AbortEnvironmentUpdate](#) ou o comando da AWS CLI equivalente.

As implantações de divisão de tráfego exigem um Application Load Balancer. O Elastic Beanstalk usa esse tipo de balanceador de carga por padrão quando você cria seu ambiente usando o console do Elastic Beanstalk ou a CLI do EB.

## Namespaces da opção de implantação

Você pode usar as [opções de configuração](#) no namespace [aws:elasticbeanstalk:command](#) para configurar suas implantações. Se você escolher a política de divisão de tráfego, opções adicionais para essa política estarão disponíveis no namespace [aws:elasticbeanstalk:trafficsplitting](#).

Use a opção `DeploymentPolicy` para definir o tipo de implantação. Os valores a seguir são aceitos:

- `AllAtOnce`: desabilita implantações contínuas e sempre implanta em todas as instâncias simultaneamente.
- `Rolling`: habilita as implantações contínuas padrão.
- `RollingWithAdditionalBatch`: inicia um lote adicional de instâncias antes de iniciar a implantação para manter a capacidade total.
- `Immutable`: executa uma [atualização imutável](#) para cada implantação.
- `TrafficSplitting`: executa implantações de divisão de tráfego para fazer testes canary nas suas implantações de aplicações.

Quando você habilitar as implantações constantes, defina as opções `BatchSize` e `BatchSizeType` para configurar o tamanho de cada lote. Por exemplo, para implantar 25% de todas as instâncias em cada lote, especifique as opções e os valores a seguir.

Example `.ebextensions/rolling-updates.config`

```
option_settings:
 aws:elasticbeanstalk:command:
 DeploymentPolicy: Rolling
 BatchSizeType: Percentage
 BatchSize: 25
```

Para implantar em cinco instâncias em cada lote, independentemente do número de instâncias em execução, e para abrir mais um lote de cinco instâncias executando a nova versão antes de colocar qualquer instância fora de serviço, especifique as opções e os valores a seguir.

Example `.ebextensions/rolling-additionalbatch.config`

```
option_settings:
 aws:elasticbeanstalk:command:
 DeploymentPolicy: RollingWithAdditionalBatch
 BatchSizeType: Fixed
 BatchSize: 5
```

Para executar uma atualização imutável para cada implantação com um limite de verificação de integridade Aviso e continuar com a implantação, mesmo se as instâncias em um lote não passarem nas verificações de integridade em um limite de tempo de 15 minutos, especifique as seguintes opções e valores.

Example `.ebextensions/immutable-ignorehealth.config`

```
option_settings:
 aws:elasticbeanstalk:command:
 DeploymentPolicy: Immutable
 HealthCheckSuccessThreshold: Warning
 IgnoreHealthCheck: true
 Timeout: "900"
```

Para executar implantações de divisão de tráfego, encaminhando 15% do tráfego do cliente para a nova versão do aplicativo e avaliando a integridade por 10 minutos, especifique as seguintes opções e valores.

## Example .ebextensions/traffic-splitting.config

```
option_settings:
 aws:elasticbeanstalk:command:
 DeploymentPolicy: TrafficSplitting
 aws:elasticbeanstalk:trafficsplitting:
 NewVersionPercent: "15"
 EvaluationTime: "10"
```

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte [Valores recomendados](#).

## Implantações azuis/verdes com o Elastic Beanstalk

Como o AWS Elastic Beanstalk realiza uma atualização in loco quando você atualiza as versões da aplicação, sua aplicação pode ficar indisponível para os usuários por um curto período. Para evitar isso, execute uma implantação azul/verde. Para fazer isso, implante um ambiente separado e, em seguida, troque os CNAMEs dos dois ambientes para redirecionar o tráfego para a nova versão instantaneamente.

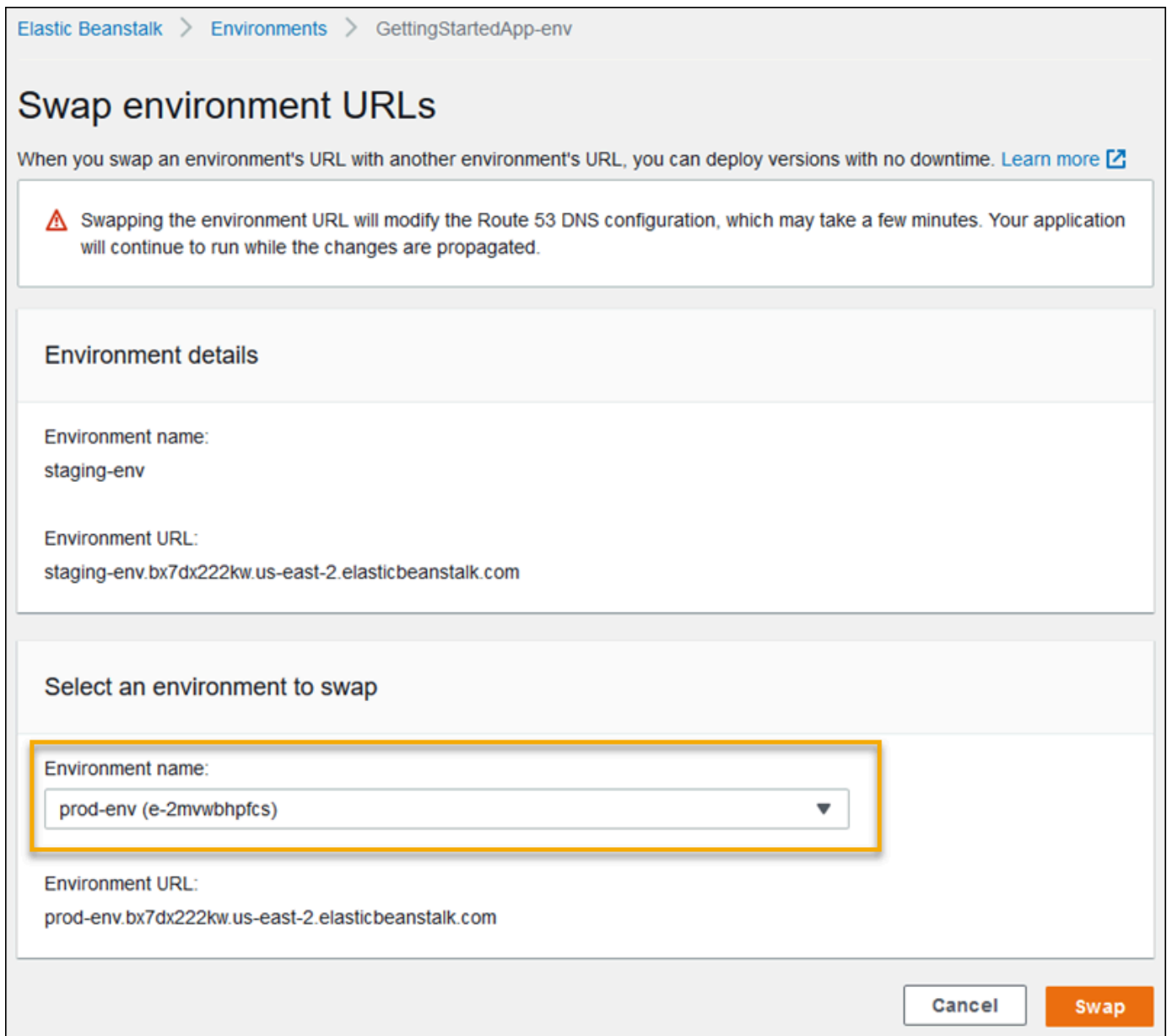
Uma implantação azul/verde também é necessária quando você deseja atualizar um ambiente para uma versão de plataforma incompatível. Para obter mais informações, consulte [the section called “Atualizações da plataforma”](#).

As implantações azuis/verdes exigem que o ambiente seja executado de forma independente do seu banco de dados de produção, caso a aplicação use um. Se seu ambiente incluir um banco de dados criado pelo Elastic Beanstalk em seu nome, o banco de dados e a conexão do ambiente não serão preservados, a menos que você realize ações específicas. Se você tiver um banco de dados que deseja reter, use uma das opções de ciclo de vida do banco de dados do Elastic Beanstalk. Você pode escolher a opção Reter para manter o banco de dados e o ambiente operacionais depois de desacoplar o banco de dados. Para obter mais informações, consulte [Ciclo de vida do banco de dados](#) no capítulo Configuração de ambientes deste guia.

Para instruções sobre como configurar sua aplicação para se conectar a uma instância externa do Amazon RDS (não gerenciada pelo Elastic Beanstalk), consulte [Usar o Elastic Beanstalk com o Amazon RDS](#).

## Para executar uma implantação azul/verde

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. [Clone seu ambiente atual](#) ou inicie um novo ambiente para executar a versão de plataforma desejada.
3. [Implante a nova versão da aplicação](#) no novo ambiente.
4. Teste a nova versão no novo ambiente.
5. Na página de visão geral do ambiente, escolha Actions (Ações) e depois Swap environment URLs (Trocar URLs do ambiente).
6. Em Environment name (Nome do ambiente), selecione o ambiente atual.



Elastic Beanstalk > Environments > GettingStartedApp-env

## Swap environment URLs

When you swap an environment's URL with another environment's URL, you can deploy versions with no downtime. [Learn more](#)

**⚠** Swapping the environment URL will modify the Route 53 DNS configuration, which may take a few minutes. Your application will continue to run while the changes are propagated.

### Environment details

Environment name:  
staging-env

Environment URL:  
staging-env.bx7dx222kw.us-east-2.elasticbeanstalk.com

### Select an environment to swap

Environment name:  
prod-env (e-2mwwbhpfc)

Environment URL:  
prod-env.bx7dx222kw.us-east-2.elasticbeanstalk.com

## 7. Escolha Swap.

O Elastic Beanstalk troca os registros de CNAME dos ambientes antigo e novo redirecionando o tráfego da versão antiga para a nova versão.

Depois que o Elastic Beanstalk concluir a operação de troca, verifique se o novo ambiente responde quando você tenta se conectar ao URL do ambiente antigo. No entanto, não encerre o ambiente antigo até as alterações de DNS serem propagadas e seus antigos registros DNS expirarem. Os servidores do DNS nem sempre limpam os registros antigos do cache com base na vida útil (TTL) que você define nos registros do DNS.

## Alterações de configuração

Quando você modifica as definições de opção de configuração na seção Configuration do [console de gerenciamento de ambiente](#), o AWS Elastic Beanstalk propaga a alteração para todos os recursos afetados. Esses recursos incluem o balanceador de carga que distribui o tráfego para as instâncias do Amazon EC2 que executam a aplicação, o grupo de Auto Scaling que gerencia essas instâncias e as próprias instâncias do EC2.

Várias alterações de configuração podem ser aplicadas a um ambiente em execução sem substituir as instâncias existentes. Por exemplo, a configuração de um [URL de verificação de integridade](#) aciona uma atualização de ambiente para modificar as configurações do load balancer, mas não causa tempo de inatividade, porque as instâncias que executam seu aplicativo continuam atendendo às solicitações enquanto a atualização é propagada.

As alterações de configuração que modificam a [configuração de execução](#) ou [configurações de VPC](#) exigem o encerramento de todas as instâncias no ambiente e a substituição. Por exemplo, quando você altera o tipo de instância ou a configuração de chaves SSH do ambiente, as instâncias do EC2 precisam ser encerradas e substituídas. O Elastic Beanstalk fornece várias políticas que determinam como essa substituição é feita.

- **Atualizações contínuas:** o Elastic Beanstalk aplica as alterações de configuração em lotes, mantendo um número mínimo de instâncias em execução e distribuindo o tráfego o tempo todo. Essa abordagem evita o tempo de inatividade durante o processo de atualização. Para obter mais detalhes, consulte [Atualizações contínuas](#).
- **Atualizações imutáveis:** o Elastic Beanstalk inicia um grupo temporário de Auto Scaling fora do ambiente com um conjunto separado de instâncias em execução com a nova configuração. Depois, o Elastic Beanstalk coloca essas instâncias atrás do balanceador de carga do ambiente. As instâncias antigas e novas distribuem o tráfego até que as novas instâncias passem nas verificações de integridade. Naquele momento, o Elastic Beanstalk move as novas instâncias para o grupo de Auto Scaling do ambiente e encerra o grupo temporário e as instâncias antigas. Para obter mais detalhes, consulte [Atualizações imutáveis](#).
- **Desativado:** o Elastic Beanstalk não tenta evitar o tempo de inatividade. Ele encerra as instâncias existentes do ambiente e as substitui por novas instâncias em execução com a nova configuração.

**⚠ Warning**

Algumas políticas substituem todas as instâncias durante a implantação ou a atualização. Por conta disso, todo o [equilíbrio de intermitência do Amazon EC2](#) é perdido. Isso acontece nos seguintes casos:

- Atualizações de plataforma gerenciada com substituição de instância habilitada
- Atualizações imutáveis
- Implantações com atualizações imutáveis ou divisão de tráfego habilitada

## Tipos de atualização compatíveis

Contínua da configuração de atualização	Ambientes de carga equilibrada	Ambientes de instância única	Ambientes legados do Windows Server†
Desabilitado	✓ Sim	✓ Sim	✓ Sim
Contínua com base na integridade	✓ Sim	× Não	✓ Sim
Contínua com base no tempo	✓ Sim	× Não	✓ Sim
Imutável	✓ Sim	✓ Sim	× Não

† Para a finalidade dessa tabela, um ambiente legado do Windows Server é um ambiente com base em uma [configuração de plataforma do Windows Server](#) que usa uma versão anterior do IIS 8.5.

## Tópicos

- [Atualizações de configuração de ambiente contínuas do Elastic Beanstalk](#)
- [Atualizações de ambiente imutáveis](#)

## Atualizações de configuração de ambiente contínuas do Elastic Beanstalk

Quando uma [alteração de configuração exige a substituição de instâncias](#), o Elastic Beanstalk pode executar a atualização em lotes para evitar tempo de inatividade enquanto a modificação é



propagada. Durante uma atualização contínua, a capacidade só é reduzida pelo tamanho de um único lote, o qual você pode configurar. O Elastic Beanstalk coloca um lote de instâncias fora de serviço e as encerra, para, depois, iniciar um lote com a nova configuração. Depois que o novo lote começar a atender a solicitações, o Elastic Beanstalk passa para o próximo lote.

Os lotes de atualização contínua da configuração podem ser processados periodicamente (com base no tempo), com um atraso entre cada lote ou de acordo com a integridade. Para atualizações contínuas baseadas no tempo, você pode configurar a quantidade de tempo que o Elastic Beanstalk aguarda depois de concluir a inicialização de um lote de instâncias antes de passar para o próximo. Essa pausa permite que seu aplicativo faça o bootstrap e comece a atender às solicitações.

De acordo com as atualizações contínuas baseadas na integridade, o Elastic Beanstalk aguarda até que as instâncias em um lote passem por verificações de integridade antes de mudar para o próximo lote. A integridade de uma instância é determinada pelo sistema de relatórios de integridade, o qual pode ser básico ou aprimorado. Com a [integridade básica](#), um lote é considerado íntegro assim que todas as suas instâncias passem nas verificações de integridade do Elastic Load Balancing (ELB).

Com os [relatórios de integridade aprimorada](#), todas as instâncias em um lote devem passar por várias verificações de integridade consecutivas antes que o Elastic Beanstalk vá para o próximo lote. Além das verificações de integridade do ELB, que verificam somente suas instâncias, a integridade aprimorada monitora os logs do aplicativo e o estado de outros recursos do ambiente. Em um ambiente de servidor web com integridade aprimorada, todas as instâncias devem passar por 12 verificações de integridade durante o período de dois minutos (18 verificações em três minutos para ambientes de operador). Se qualquer instância falhar em uma verificação de integridade, a contagem recomeça.

Se um lote não se tornar íntegro dentro do tempo limite da atualização contínua (o padrão é 30 minutos), a atualização será cancelada. O timeout de atualização contínua é uma [opção de configuração](#) que está disponível no namespace [aws:autoscaling:updatepolicy:rollingupdate](#). Caso a aplicação não passe nas verificações de integridade com o status Ok, mas esteja estável em outro nível, você pode definir a opção `HealthCheckSuccessThreshold` no namespace [aws:elasticbeanstalk:healthreporting:system](#) para alterar o nível no qual o Elastic Beanstalk considera uma instância como íntegra.

Se o processo de atualização contínua falhar, o Elastic Beanstalk iniciará outra atualização contínua para retornar à configuração anterior. Uma atualização contínua pode falhar devido a verificações de integridade com falha ou caso a execução de novas instâncias fizer com que você exceda as cotas na sua conta. Se você atingir uma cota para o número de instâncias do Amazon EC2, por exemplo,

a atualização contínua poderá falhar quando tentar provisionar um lote de novas instâncias. Nesse caso, a reversão também falhará.

Uma de reversão com falha encerra o processo de atualização e deixa o ambiente em um estado não íntegro. Os lotes não processados ainda estão executando instâncias com a configuração antiga, enquanto todos os lotes concluídos com êxito têm a nova configuração. Para corrigir um ambiente após uma reversão com falha, primeiro resolva o problema subjacente que causou a falha da atualização e, depois, inicie outra atualização do ambiente.

Um método alternativo é implantar a nova versão do aplicativo em um ambiente diferente e, depois, executar uma troca do CNAME para redirecionar o tráfego sem qualquer inatividade. Consulte [Implantações azuis/verdes com o Elastic Beanstalk](#) para obter mais informações.

## Atualizações contínuas versus implantações contínuas

As atualizações contínuas ocorrem quando você altera configurações que exigem novas instâncias do Amazon EC2 para serem provisionadas no ambiente. Isso inclui alterações na configuração do grupo de Auto Scaling como as configurações do tipo de instância e do par de chaves, bem como alterações nas configurações da VPC. Em uma atualização contínua, cada lote de instâncias é encerrado antes que um novo lote seja provisionado para substituí-lo.

As [implantações contínuas](#) ocorrem sempre que você implanta a aplicação e normalmente podem ser executadas sem a substituição de instâncias no ambiente. O Elastic Beanstalk coloca cada lote fora de serviço, implanta a nova versão da aplicação e, depois, coloca-o novamente em serviço.

A exceção é se você alterar as configurações que exigem a substituição da instância enquanto implanta uma nova versão do aplicativo. Por exemplo, se você alterar as configurações de [nome da chave](#) em um [arquivo de configuração](#) em seu pacote de origem e o implantar em seu ambiente, vai acionar uma atualização contínua. Em vez de implantar a nova versão do aplicativo para cada lote de instâncias existentes, um novo lote de instâncias é provisionado com a nova configuração. Nesse caso, uma implantação separada não ocorre, pois as novas instâncias são ativadas com a nova versão do aplicativo.

Sempre que novas instâncias são provisionadas como parte de uma atualização de ambiente, há uma fase de implantação em que o código-fonte do aplicativo é implantado nas novas instâncias, e qualquer definição da configuração que modifica o sistema operacional ou o software em instâncias é aplicada. As [Configurações de verificação de integridade da implantação](#) (Ignore health check (Ignorar verificação de integridade), Healthy threshold (Limite da integridade) Command timeout (Tempo limite do comando)) também se aplicam a atualizações contínuas baseadas na integridade e a atualizações imutáveis durante a fase de implantação.

## Configurar atualizações contínuas

É possível habilitar e configurar atualizações contínuas no console do Elastic Beanstalk.

Para habilitar atualizações contínuas

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Rolling updates and deployments (Atualizações e implantações contínuas), escolha Edit (Editar).
5. Na seção Configuration updates (Atualizações da configuração), para Rolling update type (Tipo de atualização contínua), selecione uma das opções de Rolling (Contínua).

**Configuration updates**  
Changes to virtual machine settings and VPC configuration trigger rolling updates to replace the instances in your environment without downtime.  
[Learn more](#)

Rolling update type  
Rolling based on Health

Batch size  
1  
The maximum number of instances to replace in each phase of the update.

Minimum capacity  
1  
The minimum number of instances to keep in service at all times.

Pause time  
hh:mm:ss  
Pause the update for up to an hour between each batch.

6. Selecione as configurações Batch size (Tamanho do lote), Minimum capacity (Capacidade mínima), e Pause time (Tempo de pausa).

7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A seção Configuration updates (Atualizações da configuração) da página Rolling updates and deployments (Atualizações contínuas e implantações) tem as seguintes opções para atualizações contínuas:

- Rolling update type (Tipo de atualização contínua): o Elastic Beanstalk aguarda a conclusão da atualização de um lote de instâncias para passar para o próximo lote e permitir que essas instâncias concluam o bootstrapping e comecem a atender ao tráfego. Escolha entre as seguintes opções:
  - Rolling based on Health (Contínua com base na integridade): aguarde até que as instâncias no lote atual se tornem íntegras antes de colocar instâncias em serviço e iniciar o próximo lote.
  - Rolling based on Time (Contínua com base no tempo): especifique um tempo de espera entre o início de novas instâncias e sua colocação em serviço para iniciar o próximo lote.
  - Immutable (Imutável): aplique a alteração de configuração a um novo grupo de instâncias executando uma [atualização imutável](#).
- Batch size (Tamanho do lote): o número de instâncias a serem substituídas em cada lote, entre **1** e **10000**. Por padrão, esse valor é um terço do tamanho mínimo do grupo de Auto Scaling arredondado para um número inteiro.
- Minimum capacity (Capacidade mínima): o número mínimo de instâncias para manter em execução enquanto outras instâncias estão sendo atualizadas, entre **0** e **9999**. O valor padrão é o tamanho mínimo do grupo de Auto Scaling ou um a menos do que o tamanho máximo do grupo de Auto Scaling, qualquer que seja o número menor.
- Pause time (Tempo de pausa) (somente baseado no tempo): a quantidade de tempo de espera após um lote ser atualizado antes de passar para o próximo lote para permitir que a aplicação comece a receber tráfego. Entre zero segundo e uma hora.

O namespace `aws:autoscaling:updatepolicy:rollingupdate`

Você também pode usar as [opções de configuração](#) no namespace [aws:autoscaling:updatepolicy:rollingupdate](#) para configurar as atualizações contínuas.

Use a opção `RollingUpdateEnabled` para habilitar atualizações contínuas e `RollingUpdateType` para escolher o tipo de atualização. Os valores a seguir são compatíveis com `RollingUpdateType`:

- **Health** – aguarde até que as instâncias no lote atual se tornem íntegras antes de colocá-las em serviço e iniciar o próximo lote.
- **Time** – especifique um tempo de espera entre a inicialização das novas instâncias e sua colocação em serviço para iniciar o próximo lote.
- **Immutable**: aplique a alteração de configuração a um novo grupo de instâncias executando uma [atualização imutável](#).

Quando você habilitar as atualizações constantes, defina as opções `MaxBatchSize` e `MinInstancesInService` para configurar o tamanho de cada lote. Para atualizações contínuas baseadas no tempo, também é possível configurar `PauseTime` e `Timeout`, respectivamente.

Por exemplo, para iniciar até cinco instâncias por vez mantendo pelo menos duas instâncias em serviço e aguardar cinco minutos e 30 segundos entre cada lote, especifique as opções e os valores a seguir.

Example `.ebextensions/timebased.config`

```
option_settings:
 aws:autoscaling:updatepolicy:rollingupdate:
 RollingUpdateEnabled: true
 MaxBatchSize: 5
 MinInstancesInService: 2
 RollingUpdateType: Time
 PauseTime: PT5M30S
```

Para habilitar atualizações contínuas baseadas na integridade, com um tempo limite de 45 minutos para cada lote, especifique as opções e os valores a seguir.

Example `.ebextensions/healthbased.config`

```
option_settings:
 aws:autoscaling:updatepolicy:rollingupdate:
 RollingUpdateEnabled: true
 MaxBatchSize: 5
 MinInstancesInService: 2
 RollingUpdateType: Health
 Timeout: PT45M
```

Os valores `Timeout` e `PauseTime` devem ser especificados na [duração ISO8601](#): `PT#H#M#S`, em que cada `#` é o número de horas, minutos ou segundos, respectivamente.

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte [Valores recomendados](#).

## Atualizações de ambiente imutáveis

Atualizações de ambiente imutável são uma alternativa a [atualizações contínuas](#). As atualizações de ambiente imutável garantem que as alterações de configuração que exigem a substituição de instâncias sejam aplicadas de forma eficiente e segura. Se uma atualização de ambiente imutável falhar, o processo de reversão precisará apenas do encerramento de um grupo de Auto Scaling. Uma atualização contínua com falha, por outro lado, requer a execução de uma atualização contínua adicional para reverter as alterações.

Para executar uma atualização de ambiente imutável, o Elastic Beanstalk cria um segundo grupo temporário de Auto Scaling por trás do balanceador de carga do ambiente para conter as novas instâncias. Primeiro, o Elastic Beanstalk inicia uma única instância com a nova configuração no novo grupo. Essa instância serve o tráfego junto com todas as instâncias no grupo original de Auto Scaling que estão executando a configuração anterior.

Depois que a primeira instância passa pelas verificações de integridade, o Elastic Beanstalk executa instâncias adicionais com a nova configuração que correspondem ao número de instâncias em execução no grupo original de Auto Scaling. Quando todas as novas instâncias passam por verificações de integridade, o Elastic Beanstalk as transfere para o grupo original de Auto Scaling e encerra o grupo temporário de Auto Scaling e as instâncias antigas.

### Note

Durante a atualização de ambiente imutável, a capacidade do ambiente dobra por um curto período quando as instâncias no novo grupo de Auto Scaling começam a atender às solicitações e antes de as instâncias do grupo original de Auto Scaling serem encerradas. Se o ambiente tiver muitas instâncias ou você tiver uma [cota de instância sob demanda](#) baixa, verifique se há capacidade suficiente para executar uma atualização de ambiente imutável. Se você estiver perto da cota, considere o uso de atualizações contínuas.

As atualizações imutáveis exigem um [relatório de integridade aprimorada](#) para avaliar a integridade de seu ambiente durante a atualização. Relatórios de integridade aprimorados combinam verificações de integridade padrão do load balancer com monitoramento de instância. Desse modo, ele verifica se as instâncias com a nova configuração estão [atendendo às solicitações com êxito](#).

Você também pode usar as atualizações imutáveis para implantar novas versões do seu aplicativo, como uma alternativa para implantações contínuas. Quando você [configura o Elastic Beanstalk para usar as atualizações imutáveis para implantações de aplicações](#), ele substitui todas as instâncias no ambiente cada vez que uma nova versão da aplicação é implantada. Se uma implantação de aplicação imutável falhar, o Elastic Beanstalk reverterá as alterações imediatamente ao encerrar o novo grupo de Auto Scaling. Isso pode impedir as implantações de frota parcial, que podem ocorrer quando uma implantação contínua falha após alguns lotes terem sido concluídos.

#### Warning

Algumas políticas substituem todas as instâncias durante a implantação ou a atualização. Por conta disso, todo o [equilíbrio de intermitência do Amazon EC2](#) é perdido. Isso acontece nos seguintes casos:

- Atualizações de plataforma gerenciada com substituição de instância habilitada
- Atualizações imutáveis
- Implantações com atualizações imutáveis ou divisão de tráfego habilitada

Se uma atualização imutável falhar, as novas instâncias farão upload de [logs do pacote](#) para o Amazon S3 antes que o Elastic Beanstalk as encerre. O Elastic Beanstalk deixa logs de uma atualização imutável com falha no Amazon S3 por uma hora antes de excluí-los, em vez de o padrão de 15 minutos para logs finais e em pacote.

#### Note

Se você usar as atualizações imutáveis para implantações de versão do aplicativo, mas não para a configuração, poderá encontrar um erro se tentar implantar uma versão do aplicativo com alterações de configuração que normalmente acionariam uma atualização contínua (por exemplo, configurações que alteram o tipo de instância). Para evitar isso, faça a alteração de configuração em uma atualização separada ou configure as atualizações imutáveis tanto para implantações quanto para alterações de configuração.

Não é possível realizar uma atualização imutável em conjunto com as alterações de configuração de recurso. Por exemplo, não é possível alterar [as configurações que exigem substituição de instância](#) e, ao mesmo tempo, atualizar outras configurações ou realizar uma implantação imutável com arquivos de configuração que alteram as definições de configuração ou recursos adicionais em seu

código-fonte. Se você tentar alterar as configurações de recurso (por exemplo, as configurações do balanceador de carga) e simultaneamente executar uma atualização imutável, o Elastic Beanstalk retornará um erro.

Se as alterações de configuração de recurso não são dependente de alteração em seu código-fonte ou na configuração de instância, execute-as em duas atualizações. Se elas são dependentes, realize uma [implantação azul/verde](#).

## Configurar as atualizações imutáveis

É possível habilitar e configurar atualizações imutáveis no console do Elastic Beanstalk.

Para habilitar as atualizações imutáveis (console)

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Rolling updates and deployments (Atualizações e implantações contínuas), escolha Edit (Editar).
5. Na seção Configuration Updates, defina o Rolling update type para Immutable.



### Configuration updates

Changes to virtual machine settings and VPC configuration trigger rolling updates to replace the instances in your environment without downtime. [Learn more](#)

Rolling update type

Immutable

Batch size

1

The maximum number of instances to replace in each phase of the update.

Minimum capacity

1

The minimum number of instances to keep in service at all times.

Pause time

hh:mm:ss

Pause the update for up to an hour between each batch.

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

## O namespace `aws:autoscaling:updatepolicy:rollingupdate`

Você também pode usar as opções de configuração no namespace `aws:autoscaling:updatepolicy:rollingupdate` para configurar as atualizações imutáveis. O exemplo de [arquivo de configuração](#) a seguir permite atualizações imutáveis para alterações de configuração.

Example `.ebextensions/immutable-updates.config`

```
option_settings:
 aws:autoscaling:updatepolicy:rollingupdate:
 RollingUpdateType: Immutable
```

O exemplo a seguir permite atualizações imutáveis tanto para alterações de configuração quanto para implantações.

Example `.ebextensions/immutable-all.config`

```
option_settings:
```

```
aws:autoscaling:updatepolicy:rollingupdate:
 RollingUpdateType: Immutable
aws:elasticbeanstalk:command:
 DeploymentPolicy: Immutable
```

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte [Valores recomendados](#).

## Atualizar a versão de plataforma do ambiente Elastic Beanstalk

### Important

#### Compatibilidade com TLS 1.2

Em 31 de dezembro de 2023, AWS começou a aplicar totalmente o TLS 1.2 em todos os endpoints AWS da API. Isso removeu a capacidade de usar as versões 1.0 e 1.1 do TLS com todas as AWS APIs. Este procedimento foi originalmente comunicado em [28 de junho de 2022](#). Para evitar o risco de impacto na disponibilidade, atualize suas versões da plataforma para uma versão mais nova o mais rápido possível.

#### Impacto potencial

As versões das plataformas Elastic Beanstalk que executam o TLS v1.1 ou anterior serão afetadas. Essa alteração afetará ações do ambiente que incluem, entre outras: implantações de configuração, implantações de aplicações, ajuste de escala automático, lançamento de novo ambiente, alternância de logs, relatórios de integridade aprimorados e publicação de logs de aplicações no bucket do Amazon S3 associado a suas aplicações.

#### Versões da plataforma Windows afetadas

Os clientes com ambientes Elastic Beanstalk na seguinte versão de plataforma são aconselhados a atualizar cada um de seus ambientes correspondentes para a plataforma Windows versão 2.8.3 ou posterior, lançada em [18 de fevereiro de 2022](#).

- Windows Server 2019 - versão da plataforma 2.8.2 ou anteriores

Os clientes com ambientes Elastic Beanstalk nas seguintes versões de plataforma são aconselhados a atualizar cada um de seus ambientes correspondentes para a plataforma Windows versão 2.10.7 ou posterior, lançada em [28 de dezembro de 2022](#).

- Windows Server 2016 - versão da plataforma 2.10.6 ou versões anteriores
- Windows Server 2012 — todas as versões da plataforma; essa plataforma foi retirada em [4 de dezembro de 2023](#)
- Windows Server 2008 — todas as versões da plataforma; essa plataforma foi desativada em [28 de outubro de 2019](#)

Para obter a lista das versões da plataforma Windows Server mais recentes e compatíveis, consulte [Plataformas compatíveis](#) no guia Plataformas do AWS Elastic Beanstalk .

Para obter detalhes e melhores práticas sobre como atualizar seu ambiente, leia as informações neste tópico.

### Plataformas Amazon Linux AMI (AL1)

Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI (AL1) como retiradas. Variações das plataformas Amazon Linux AMI (AL1) podem ser afetadas por essa alteração. Para evitar impacto na disponibilidade, recomendamos que você atualize seus ambientes Beanstalk baseados em AL1 para a versão mais recente da plataforma Amazon Linux 2 ou Amazon Linux 2023.

Para obter a lista das versões da plataforma Elastic Beanstalk mais recentes e compatíveis, consulte [Plataformas compatíveis](#) no guia Plataformas do AWS Elastic Beanstalk .

Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

O Elastic Beanstalk libera regularmente novas versões da plataforma para atualizar todas as [plataformas](#) baseadas no Linux e no Windows Server. As novas versões de plataforma fornecem atualizações para componentes de software existentes e suporte para novos recursos e opções de configuração. Para saber mais sobre plataformas e versões de plataformas, consulte [Glossário de plataformas do Elastic Beanstalk](#).

É possível usar o console do Elastic Beanstalk ou a CLI do EB para atualizar a versão da plataforma do ambiente. Dependendo da versão da plataforma que você deseja atualizar, o Elastic Beanstalk recomenda um dos dois métodos para realizar atualizações de plataforma.

- [Método 1: Atualizar a versão da plataforma do ambiente](#). Recomendamos esse método quando você está atualizando para a versão mais recente da plataforma em uma ramificação, com o mesmo runtime, servidor Web, servidor de aplicações e sistema operacional e sem alterar a versão principal da plataforma. Essa é a atualização de plataforma mais comum e rotineira.
- [Método 2: Executar uma implantação azul/verde](#). Recomendamos esse método quando você está atualizando para uma versão de plataforma em uma ramificação diferente, com um runtime, servidor Web, servidor de aplicações ou sistema operacional diferente, ou para outra versão principal da plataforma. Essa é uma boa abordagem quando você deseja aproveitar os novos recursos de runtime ou a funcionalidade do Elastic Beanstalk mais recente ou quando deseja sair de uma ramificação de plataforma defasada ou retirada.

[A migração de uma versão de plataforma legada](#) requer uma implantação azul/verde, pois essas versões de plataforma são incompatíveis com as versões atualmente compatíveis.

[A migração de uma aplicação Linux para o Amazon Linux 2](#) requer uma implantação azul/verde, pois as versões da plataforma Amazon Linux 2 são incompatíveis com versões anteriores da plataforma da AMI do Amazon Linux.

Para obter mais ajuda sobre como escolher o melhor método de atualização de plataforma, expanda a seção da plataforma do ambiente.

## Docker

Use o [Método 1](#) para executar atualizações da plataforma.

## Docker de vários contêineres

Use o [Método 1](#) para executar atualizações da plataforma.

## Docker pré-configurado

Considere os seguintes casos:

- Se você estiver migrando a aplicação para outra plataforma, por exemplo, do Go 1.4 (Docker) para o Go 1.11 ou do Python 3.4 (Docker) para o Python 3.6, use o [Método 2](#).

- Se você estiver migrando a aplicação para uma versão de contêiner diferente do Docker, por exemplo, da versão Glassfish 4.1 (Docker) para a Glassfish 5.0 (Docker), use o [Método 2](#).
- Se você estiver atualizando para uma versão de plataforma mais recente sem nenhuma alteração na versão do contêiner ou da versão principal, use o [Método 1](#).

## Go

Use o [Método 1](#) para executar atualizações da plataforma.

## Java SE

Considere os seguintes casos:

- Se você estiver migrando a aplicação para uma versão de runtime Java diferente, por exemplo, de Java 7 para Java 8, use o [Método 2](#).
- Se você estiver atualizando para uma versão de plataforma mais recente sem nenhuma alteração na versão de runtime, use o [Método 1](#).

## Java com Tomcat

Considere os seguintes casos:

- Se você estiver migrando a aplicação para uma versão de runtime Java diferente ou uma versão do servidor de aplicações Tomcat, por exemplo, do Java 7 com Tomcat 7 para o Java 8 com Tomcat 8.5, use o [Método 2](#).
- Se você estiver migrando a aplicação para as principais versões da plataforma Java com o Tomcat (v1.x.x, v2.x.x e v3.x.x), use o [Método 2](#).
- Se você estiver atualizando para uma versão de plataforma mais recente sem nenhuma alteração na versão de runtime, na versão do servidor de aplicações ou na versão principal, use o [Método 1](#).

## .NET no Windows Server com IIS

Considere os seguintes casos:

- Se você estiver migrando a aplicação para uma versão diferente do sistema operacional Windows, por exemplo, da versão Windows Server 2008 R2 para Windows Server 2016, use o [Método 2](#).

- Se você estiver migrando a aplicação para as principais versões da plataforma Windows Server, consulte [Migrar de versões principais anteriores da plataforma do Windows Server](#) e use o [Método 2](#).
- Se a aplicação estiver sendo executado em uma plataforma Windows Server V2.x.x e você estiver atualizando para uma versão de plataforma mais recente, use o [Método 1](#).

#### Note

[As versões da plataforma Windows Server](#) anteriores à v2 não são agrupadas semanticamente em versões. Só é possível iniciar a versão mais recente de cada uma das versões principais da plataforma do Windows Server e não é possível reverter após a atualização.

## Node.js

Use o [Método 2](#) para executar atualizações da plataforma.

## PHP

Considere os seguintes casos:

- Se você estiver migrando a aplicação para uma versão de runtime PHP diferente, por exemplo, da versão PHP 5.6 para PHP 7.2, use o [Método 2](#).
- Se você estiver migrando a aplicação para as principais versões da plataforma PHP (v1.x.x e v2.x.x), use o [Método 2](#).
- Se você estiver atualizando para uma versão de plataforma mais recente sem nenhuma alteração na versão de runtime ou da versão principal, use o [Método 1](#).

## Python

Considere os seguintes casos:

- Se você estiver migrando a aplicação para uma versão de runtime diferente do Python, por exemplo, da versão Python 2.7 para Python 3.6, use o [Método 2](#).
- Se você estiver migrando a aplicação para as principais versões da plataforma Python (v1.x.x e v2.x.x), use o [Método 2](#).

- Se você estiver atualizando para uma versão de plataforma mais recente sem nenhuma alteração na versão de tempo de execução ou da versão principal, use o [Método 1](#).

## Ruby

Considere os seguintes casos:

- Se você estiver migrando a aplicação para uma versão de runtime ou uma versão do servidor de aplicações Ruby diferente, por exemplo, da versão Ruby 2.3 com Puma para Ruby 2.6 com Puma, use o [Método 2](#).
- Se você estiver migrando a aplicação para as principais versões da plataforma Ruby (v1.x.x e v2.x.x), use o [Método 2](#).
- Se você estiver atualizando para uma versão de plataforma mais recente sem nenhuma alteração na versão de runtime, na versão do servidor de aplicações ou na versão principal, use o [Método 1](#).

## Método 1: Atualizar a versão da plataforma do ambiente

Use esse método para atualizar para a ramificação mais recente da plataforma do seu ambiente. Se você criou um ambiente anteriormente usando uma versão mais antiga da plataforma, ou atualizou o ambiente de uma versão mais antiga, também é possível esse método para reverter para uma versão anterior da plataforma, contanto que seja na mesma ramificação da plataforma.

Como atualizar a versão da plataforma de seu ambiente

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

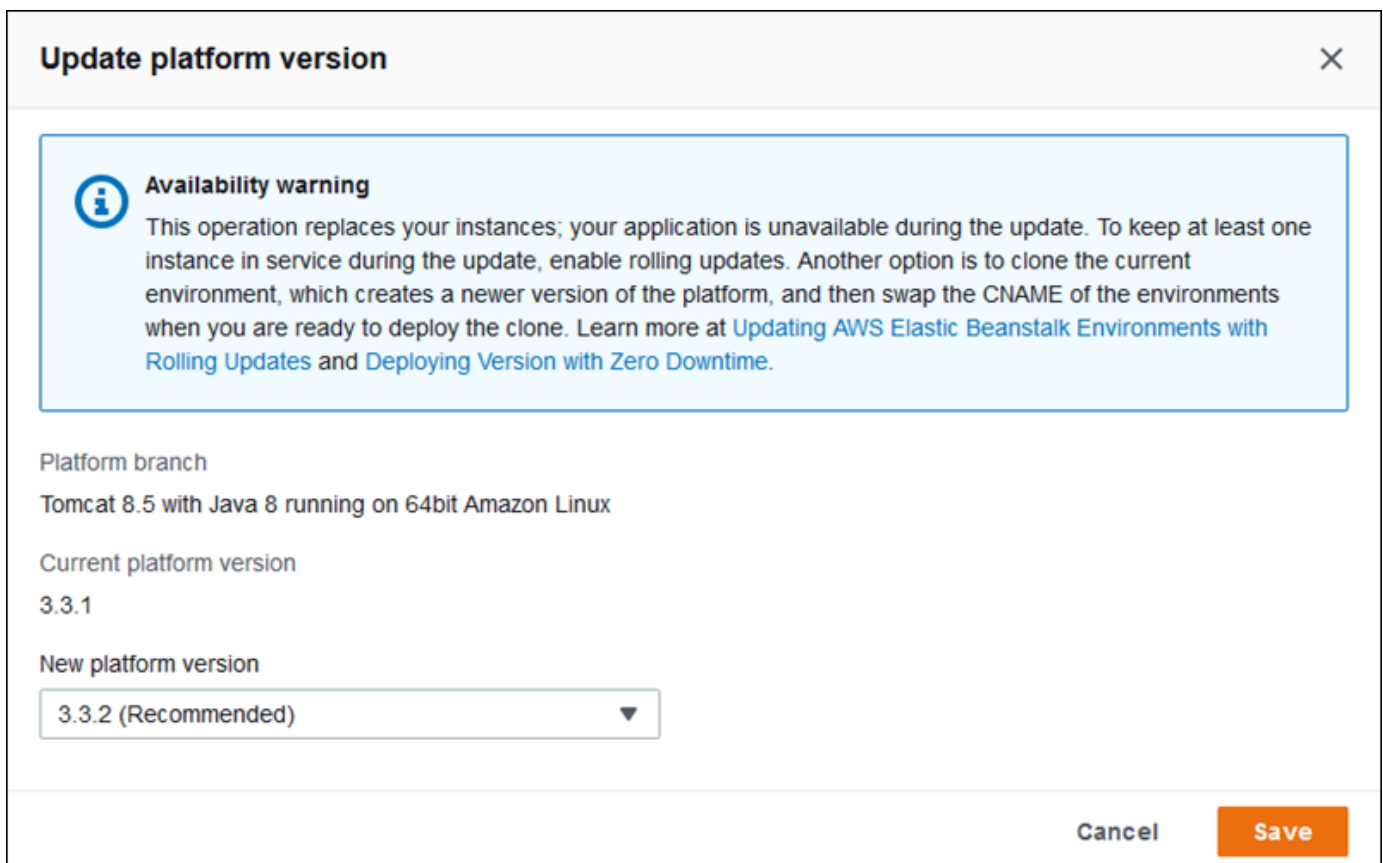
### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, em Platform (Plataforma), escolha Change (Alterar).



4. Na caixa de diálogo Update platform version (Atualizar versão da plataforma), selecione uma versão da plataforma. A versão mais recente (recomendada) da plataforma na ramificação é selecionada automaticamente. É possível atualizar para qualquer versão que você usou no passado.



5. Escolha Save (Salvar).



Para simplificar ainda mais as atualizações da plataforma, o Elastic Beanstalk pode gerenciá-las para você. É possível configurar o ambiente para aplicar atualizações de versão secundárias e de patch automaticamente durante uma janela de manutenção semanal configurável. O Elastic Beanstalk aplica atualizações gerenciadas sem tempo de inatividade nem redução na capacidade, e cancelará a atualização imediatamente se as instâncias que executam a aplicação na nova versão falharem nas verificações de integridade. Para obter mais detalhes, consulte [Atualizações gerenciadas de plataforma](#).

## Método 2: Executar uma implantação azul/verde

Use este método para atualizar para uma ramificação de plataforma diferente, com um runtime, servidor Web, servidor de aplicações ou sistema operacional diferente, ou para outra versão principal da plataforma. Normalmente, isso é necessário quando você deseja aproveitar os novos recursos de runtime ou a funcionalidade mais recente do Elastic Beanstalk. Também é necessário quando você está migrando para fora de uma ramificação defasada ou desativada da plataforma.

Quando você migra entre as principais versões de plataforma ou para as versões de plataforma com as principais atualizações de componentes, há uma probabilidade maior de que seu aplicativo, ou alguns aspectos dele, não funcione conforme o esperado na nova versão da plataforma e possa exigir alterações.

Antes de executar a migração, atualize sua máquina de desenvolvimento local para as versões de runtime mais novas e outros componentes da plataforma para a qual você planeja migrar. Verifique se o aplicativo ainda funciona conforme o esperado e faça as correções e alterações de código necessárias. Depois, use o procedimento de práticas recomendadas a seguir para migrar com segurança o ambiente para a nova versão da plataforma.

Para migrar o ambiente para uma versão de plataforma com atualizações importantes

1. [Crie um ambiente novo](#) usando a nova versão de plataforma de destino e implante o código do aplicativo nela. O novo ambiente deve estar na aplicação do Elastic Beanstalk que contém o ambiente para o qual você está migrando. Não encerre o ambiente existente.
2. Use o novo ambiente para migrar o aplicativo. Em particular:
  - Encontre e corrija quaisquer problemas de compatibilidade de aplicativos que você não conseguiu descobrir durante a fase de desenvolvimento.
  - Certifique-se de que as personalizações que seu aplicativo faz usando os [arquivos de configuração](#) funcionem corretamente no novo ambiente. Eles podem incluir configurações de

opções, pacotes adicionais instalados, políticas de segurança personalizadas e arquivos de script ou configuração instalados em instâncias de ambiente.

- Se o aplicativo usar uma imagem de máquina da Amazon (AMI), crie uma nova AMI personalizada com base na AMI da nova versão da plataforma. Para saber mais, consulte [Usar uma imagem de máquina da Amazon \(AMI\) personalizada](#). Especificamente, isso é necessário se o aplicativo usar a plataforma Windows Server com uma AMI personalizada e você estiver migrando para uma versão da plataforma Windows Server V2. Neste caso, consulte também [Migrar de versões principais anteriores da plataforma do Windows Server](#).

Itene os testes e implante suas correções até que você esteja satisfeito com o aplicativo no novo ambiente.

3. Torne o novo ambiente em seu ambiente de produção. Para isso, troque o CNAME dele pelo CNAME do ambiente de produção existente. Para obter mais detalhes, consulte [Implantações azuis/verdes com o Elastic Beanstalk](#).
4. Quando você estiver satisfeito com o estado do seu novo ambiente de produção, encerre o ambiente antigo. Para obter mais detalhes, consulte [Encerrar um ambiente do Elastic Beanstalk](#).

## Atualizações gerenciadas de plataforma

O AWS Elastic Beanstalk lança regularmente [atualizações de plataforma](#) para fornecer correções, atualizações de software e novos recursos. Com as atualizações gerenciadas de Plataforma, você pode configurar seu ambiente para atualizar automaticamente para a versão mais recente de uma Plataforma durante uma [janela de manutenção](#) programada. Seu aplicativo permanece em serviço durante o processo de atualização, sem redução da capacidade. As atualizações gerenciadas estão disponíveis em ambos os ambientes de instância única e com balanceamento de carga.

### Note

Esse recurso não está disponível em [versões da plataforma Windows Server](#) anteriores à versão 2 (v2).

Você pode configurar seu ambiente para aplicar automaticamente as [atualizações de versão de patch](#) ou ambas as atualizações de patch e de versão secundária. As atualizações gerenciadas da plataforma não são compatíveis com atualizações em ramificações da plataforma (atualizações para diferentes versões principais de componentes da plataforma, como sistema operacional, runtime

ou componentes do Elastic Beanstalk), pois elas podem apresentar alterações incompatíveis com versões anteriores.

Você também pode configurar o Elastic Beanstalk para substituir todas as instâncias no ambiente durante a janela de manutenção, mesmo se não houver uma atualização de plataforma disponível. A substituição de todas as instâncias em seu ambiente será útil se o seu aplicativo encontrar bugs ou problemas de memória quando executado por um longo período.

Em ambientes criados em 25 de novembro de 2019 ou posteriormente usando o console do Elastic Beanstalk, as atualizações gerenciadas são habilitadas por padrão sempre que possível. As atualizações gerenciadas exigem [integridade aprimorada](#) para serem habilitadas. A integridade aprimorada é habilitada por padrão quando você seleciona uma das [predefinições de configuração](#), e desabilitada quando você seleciona Custom configuration (Configuração personalizada). O console não pode habilitar atualizações gerenciadas para versões de plataforma mais antigas que não oferecem suporte à integridade aprimorada ou quando a integridade aprimorada está desabilitada. Quando o console habilita as atualizações gerenciadas para um novo ambiente, a Weekly update window (Janela de atualização semanal) é definida como um dia aleatório da semana em um horário aleatório. O Update level (Nível de atualização) é definido como Minor and patch (Menor e patch), e a Instance replacement (Substituição da instância) é desabilitada. Você pode desabilitar ou reconfigurar atualizações gerenciadas antes da etapa final de criação do ambiente.

Para um ambiente existente, use o console do Elastic Beanstalk a qualquer momento para configurar atualizações de plataforma gerenciadas.

#### Important

Um grande número de ambientes Beanstalk em uma conta AWS pode apresentar um risco de problemas de controle de utilização durante atualizações gerenciadas. Grande número é uma quantidade relativa que depende de quão rigorosamente você agenda as atualizações gerenciadas para seus ambientes. Mais de 200 ambientes em uma conta agendados de perto podem causar problemas de controle de utilização, embora um número menor também possa ser problemático.

Para equilibrar a carga de recursos para atualizações gerenciadas, recomendamos distribuir as janelas de manutenção programadas para os ambientes em uma conta.

Além disso, considere uma estratégia de várias contas. Para obter mais informações, consulte [Organizando seu AWS Ambiente usando várias contas](#) no site AWS Documentos técnicos e Guias.

## Como configurar atualizações de plataforma gerenciadas

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria Managed updates (Atualizações gerenciadas), escolha Edit (Editar).
5. Desabilite ou habilite as Managed Updates (Atualizações gerenciadas).
6. Se as atualizações gerenciadas estiverem habilitadas, selecione uma janela de manutenção e selecione um Update level (Nível de atualização).
7. (Opcional) Selecione Instance replacement (Substituição da instância) para habilitar a substituição semanal da instância.

Elastic Beanstalk > Environments > GettingStartedApp-env > Configuration

## Modify managed updates

**Managed platform updates**

Enable managed platform updates to apply platform updates automatically during a weekly maintenance window that you choose. Your application stays available during the update process.

**Managed updates**

Enabled

**Weekly update window**

Tuesday at 12 : 00 UTC

Any available managed updates will run between Tuesday, 4:00 AM and Tuesday, 6:00 AM (-0800 GMT).

**Update level**

Minor and patch

**Instance replacement**

If enabled, an instance replacement will be scheduled if no other updates are available.

Enabled

Cancel Continue Apply

8. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

As atualizações gerenciadas de Plataforma dependem de [relatórios de integridade aprimorada](#) para determinar se o seu aplicativo é íntegro o suficiente para considerar a atualização da Plataforma bem-sucedida. Para obter instruções, consulte [Habilitar relatórios de integridade aprimorada do Elastic Beanstalk](#).

### Seções

- [Permissões necessárias para executar atualizações gerenciadas de plataforma](#)
- [Janela de manutenção da atualização gerenciada](#)
- [Atualizações de versão secundária e patch](#)
- [Atualizações de ambiente imutáveis](#)
- [Gerenciar atualizações gerenciadas](#)
- [Namespaces da opção de ação gerenciada](#)

## Permissões necessárias para executar atualizações gerenciadas de plataforma

O Elastic Beanstalk precisa de permissão para iniciar uma atualização de plataforma em seu nome. Para obter essas permissões, o Elastic Beanstalk assume a função de serviço de atualizações gerenciadas. Quando a [função de serviço](#) padrão é usada para o ambiente, o console do Elastic Beanstalk também a usa como função de serviço de atualizações gerenciadas. O console atribui a política gerenciada [AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy](#) à sua função de serviço. Esta política tem todas as permissões que o Elastic Beanstalk precisa para executar atualizações de plataforma gerenciada.

Para obter detalhes sobre outras formas de definir a função de serviço de atualizações gerenciadas, consulte [the section called “Perfis de serviço”](#).

### Note

Se você usar [arquivos de configuração](#) para estender o ambiente a fim de incluir recursos adicionais, poderá ser necessário adicionar permissões à função de serviço de atualizações gerenciadas do ambiente. Normalmente, você precisa adicionar permissões ao fazer referência a esses recursos pelo nome em outras seções ou arquivos.

Se houver falha em uma atualização, você poderá localizar o motivo na página [Atualizações gerenciadas](#).

## Janela de manutenção da atualização gerenciada

Quando a AWS lança uma nova versão da plataforma do ambiente, o Elastic Beanstalk programa uma atualização gerenciada da plataforma durante a próxima janela de manutenção semanal. As janelas de manutenção são de duas horas. O Elastic Beanstalk inicia uma atualização agendada durante a janela de manutenção. Pode ser que a atualização não seja concluída antes do término das janelas.

### Note

Na maioria dos casos, o Elastic Beanstalk programa a atualização gerenciada para que ela ocorra durante a próxima janela de manutenção semanal. O sistema considera diversos aspectos de segurança e disponibilidade do serviço de atualização ao programar as atualizações gerenciadas. Em casos raros, uma atualização pode não ser programada para

a primeira janela de manutenção. Quando isso acontece, o sistema tenta novamente na próxima janela de manutenção. Para aplicar manualmente a atualização gerenciada, escolha **Apply now** (Aplicar agora) como explicado em [Gerenciar atualizações gerenciadas](#), nesta página.

## Atualizações de versão secundária e patch

Você pode habilitar as atualizações gerenciadas de plataforma para aplicar somente atualizações de versão de patch ou para ambas as atualizações de patch e de versão secundária. As atualizações de versão de patch fornecem correções de bugs e melhorias de performance e podem incluir alterações de configuração secundárias no software da instância, nos scripts e nas opções de configuração. Atualizações de versões secundárias são compatíveis com novos recursos do Elastic Beanstalk. Você não pode aplicar atualizações de versão principal, que podem fazer alterações incompatíveis com a versão anterior, com as atualizações gerenciadas da plataforma.

No número da versão de um plataforma, o segundo número é a versão de atualização secundária, e o terceiro número é a versão do patch. Por exemplo, a versão 2.0.7 de uma plataforma tem 0 como versão secundária e 7 como versão de patch.

## Atualizações de ambiente imutáveis

As atualizações gerenciadas de Plataforma executam [atualizações de ambiente imutáveis](#) para atualizar seu ambiente para uma nova versão de Plataforma. As atualizações imutáveis atualizam o ambiente sem retirar as instâncias de serviço ou modificar o ambiente, antes de confirmar que as instâncias que executam a nova versão foram aprovadas nas verificações de integridade.

Em uma atualização imutável, o Elastic Beanstalk implanta o mesmo número de instâncias que estão em execução no momento com a nova versão da plataforma. As novas instâncias começam a fazer solicitações juntamente com aquelas que executam a versão antiga. Se o novo conjunto de instâncias for aprovado em todas as verificações de integridade, o Elastic Beanstalk encerrará o conjunto antigo de instâncias, deixando apenas aquelas com a nova versão.

As atualizações gerenciadas de plataforma sempre realizam atualizações imutáveis, mesmo quando você as aplica fora da janela de manutenção. Se você alterar a versão da plataforma no Dashboard (Painel), o Elastic Beanstalk aplicará a política de atualização escolhida para as atualizações de configuração.

**⚠ Warning**

Algumas políticas substituem todas as instâncias durante a implantação ou a atualização. Por conta disso, todo o [equilíbrio de intermitência do Amazon EC2](#) é perdido. Isso acontece nos seguintes casos:

- Atualizações de plataforma gerenciada com substituição de instância habilitada
- Atualizações imutáveis
- Implantações com atualizações imutáveis ou divisão de tráfego habilitada

## Gerenciar atualizações gerenciadas

O console do Elastic Beanstalk exibe informações detalhadas sobre atualizações gerenciadas na página *Managed updates overview* (Visão geral de atualizações gerenciadas).

Para visualizar informações sobre atualizações gerenciadas (console)

1. Abra o [console do Elastic Beanstalk](#) e, na lista *Regions* (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione *Ambientes* e selecione o nome do ambiente na lista.

**ℹ Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha *Managed updates* (Atualizações gerenciadas).

A seção *Visão geral de atualizações gerenciadas* apresenta informações sobre as atualizações gerenciadas programadas e pendentes. A seção *History* lista as atualizações bem-sucedidas e as tentativas com falha.


Você pode aplicar uma atualização programada imediatamente, em vez de esperar pela janela de manutenção.

Para aplicar uma atualização gerenciada de plataforma imediatamente (console)

1. Abra o [console do Elastic Beanstalk](#) e, na lista *Regions* (Regiões), selecione a sua Região da AWS.



2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Managed updates (Atualizações gerenciadas).
4. Escolha Apply now.
5. Verifique os detalhes da atualização e escolha Apply (Aplicar).

Quando você aplica uma atualização gerenciada de plataforma fora da janela de manutenção, o Elastic Beanstalk realiza uma atualização imutável. Se você atualizar a plataforma do ambiente no [Dashboard \(Painel\)](#) ou usando um cliente diferente, o Elastic Beanstalk aplicará o tipo de atualização selecionado nas [alterações de configuração](#).

Se você não tem uma atualização gerenciada programada, talvez o ambiente já esteja executando a versão mais recente. Outros motivos para não ter uma atualização programada incluem:

- Uma atualização de [versão secundária](#) está disponível, mas o ambiente está configurado para aplicar automaticamente apenas as atualizações de versão de patch.
- O ambiente não foi verificado desde que a atualização foi lançada. O Elastic Beanstalk normalmente verifica se há atualizações a cada hora.
- Uma atualização está pendente ou já está em andamento.

Quando a janela de manutenção é iniciada ou quando você seleciona Apply now (Aplicar agora), as atualizações programadas entram no status pendente antes da execução.

## Namespaces da opção de ação gerenciada

Você pode usar as [opções de configuração](#) nos namespaces [aws:elasticbeanstalk:managedactions](#) e [aws:elasticbeanstalk:managedactions:platformupdate](#) para habilitar e configurar as atualizações gerenciadas de Plataforma.

A opção `ManagedActionsEnabled` ativa as atualizações gerenciadas de plataforma. Defina essa opção como `true` para habilitar as atualizações gerenciadas de plataforma e use as outras opções para configurar o comportamento de atualização.

Use `PreferredStartTime` para configurar o início da janela de manutenção semanal no formato *dia:hora:minuto*.

Defina `UpdateLevel` como `minor` ou `patch` para aplicar ambas as atualizações de versão secundária e de patch ou apenas as atualizações de versão de patch, respectivamente.

Quando as atualizações gerenciadas de plataforma são habilitadas, você pode permitir a substituição de instâncias definindo a opção `InstanceRefreshEnabled` como `true`. Quando essa configuração é habilitada, o Elastic Beanstalk executa uma atualização imutável no ambiente a cada semana, mesmo que não haja uma nova versão da plataforma disponível.

O [arquivo de configuração](#) de exemplo a seguir habilita as atualizações gerenciadas de plataforma para atualizações da versão de patch com uma janela de manutenção que começa às 9h UTC toda terça-feira.

Example `.ebextensions/managed-platform-update.config`

```
option_settings:
 aws:elasticbeanstalk:managedactions:
 ManagedActionsEnabled: true
 PreferredStartTime: "Tue:09:00"
 aws:elasticbeanstalk:managedactions:platformupdate:
 UpdateLevel: patch
 InstanceRefreshEnabled: true
```

## Migrar seu aplicativo de uma versão de plataforma legada

Se você tiver implantado uma aplicação do Elastic Beanstalk que usa uma versão de plataforma legada, deverá migrar a aplicação para um novo ambiente usando uma versão de plataforma não legada para ter acesso a novos recursos. Se não tiver certeza de estar executando o aplicativo usando uma versão de plataforma legada, verifique no console do Elastic Beanstalk. Para obter instruções, consulte [Como verificar se você está usando uma versão de plataforma legada](#).

Quais novos recursos as versões de plataforma legada não disponibilizam?

Plataformas herdadas não são compatíveis com os seguintes recursos:

- Arquivos de configuração, como descrito no tópico [Personalização avançada de ambiente com arquivos de configuração \(.ebextensions\)](#)

- Verificações de integridade do ELB, conforme descrito no tópico [Relatórios de integridade básica](#)
- Perfis de instância, conforme descrito no tópico [Gerenciar perfis de instância do Elastic Beanstalk](#)
- VPCs, como descrito no tópico [Usar o Elastic Beanstalk com Amazon VPC](#)
- Níveis de dados, como descrito no tópico [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#)
- Níveis de operador, como descrito no tópico [Ambientes de operador](#)
- Ambientes de instância única, como descrito no tópico [Tipos de ambiente](#)
- Tags, como descrito no tópico [Marcar recursos em seus ambientes do Elastic Beanstalk](#)
- Lançamento de atualizações, como descrito no tópico [Atualizações de configuração de ambiente contínuas do Elastic Beanstalk](#)

## Por que algumas versões de plataforma são marcadas como legadas?

Algumas versões mais antigas da plataforma não são compatíveis com os recursos mais recentes do Elastic Beanstalk. Essas versões são marcadas como (legacy) (herdado) na página de visão geral do ambiente no console do Elastic Beanstalk.

Como verificar se você está usando uma versão de plataforma legada

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.


3. Na página de visão geral do ambiente, visualize o nome da Platform (Plataforma).

O aplicativo está usando uma versão de plataforma legada se você vir (legacy) (legada) ao lado do nome da plataforma.

Para migrar seu aplicativo

1. Implante o aplicativo em um novo ambiente. Para obter instruções, vá para [Criar um ambiente do Elastic Beanstalk](#).

2. Se você tem uma instância de banco de dados do Amazon RDS, atualize seu security group de banco de dados para permitir o acesso ao security group do EC2 para o novo ambiente. Para obter instruções sobre como encontrar o nome do grupo de segurança do EC2 usando o Console de Gerenciamento da AWS, consulte [Grupos de segurança](#). Para obter mais informações sobre como configurar o grupo de segurança do EC2, acesse a seção “Autorização de acesso à rede para um grupo de segurança do Amazon EC2” de [Trabalhar com grupos de segurança de banco de dados](#) no Guia do usuário do Amazon Relational Database Service.
3. Troque o URL de seu ambiente. Para obter instruções, vá para [Implantações azuis/verdes com o Elastic Beanstalk](#).
4. Encerre seu ambiente antigo. Para obter instruções, vá para [Encerrar um ambiente do Elastic Beanstalk](#).

 Note

Se você usar o AWS Identity and Access Management (IAM), precisará atualizar suas políticas para incluir o AWS CloudFormation e o Amazon RDS (se aplicável). Para obter mais informações, consulte [Usar o Elastic Beanstalk com o AWS Identity and Access Management](#).

## Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2

Esta seção descreve como migrar sua aplicação usando um dos seguintes caminhos de migração.

- Migrar de uma ramificação da plataforma Amazon Linux 2 para uma ramificação da plataforma Amazon Linux 2023.
- Migrar de uma ramificação da plataforma Amazon Linux AMI (AL1) para uma ramificação da plataforma Amazon Linux 2023 (recomendado) ou Amazon Linux 2.

### Tópicos

- [Migração do Amazon Linux 2 para o Amazon Linux 2023](#)
- [Migração do Amazon Linux AMI \(AL1\) para o AL2 ou o AL2023](#)

## Migração do Amazon Linux 2 para o Amazon Linux 2023

Este tópico fornece orientações para ajudar a migrar sua aplicação de uma ramificação da plataforma Amazon Linux 2 para uma ramificação da plataforma Amazon Linux 2023.

### Diferenças e compatibilidade

Entre as plataformas AL2 e AL2023 do Elastic Beanstalk

Há um alto grau de compatibilidade entre as plataformas Elastic Beanstalk Amazon Linux 2 e Amazon Linux 2023. Embora haja algumas diferenças a serem observadas:

- Instance Metadata Service Version 1 (IMDSv1) – A configuração padrão da opção do [DisableIMDSv1](#) está nas plataformas AL2023 `true`. O padrão está `false` nas plataformas AL2.
- ferramenta de instância `pkg-repo` — A ferramenta [pkg-repo](#) não está disponível para ambientes executados nas plataformas AL2023. No entanto, você pode aplicar atualizações de pacote e sistema operacional manualmente a uma instância AL2023. Para obter mais informações, consulte [Gerenciamento de pacotes e atualizações do sistema operacional](#) no Guia do usuário do Amazon Linux 2023
- Configuração Apache HTTPd — O arquivo `httpd.conf` Apache para plataformas AL2023 tem algumas configurações diferentes das do AL2:
  - Negar acesso a todo o sistema de arquivos do servidor por padrão. Essas configurações são descritas em Proteger arquivos do servidor por padrão na página [Dicas de segurança](#) do site Apache.
  - Impeça que os usuários substituam os recursos de segurança que você configurou. A configuração nega o acesso à configuração `.htaccess` em todos os diretórios, exceto aqueles especificamente habilitados. Essa configuração é descrita em Protegendo as configurações do sistema na página [Dicas de segurança](#) do site Apache. A página [Tutorial do Apache HTTP Server: arquivos.htaccess](#) afirma que essa configuração pode ajudar a melhorar o desempenho.
  - Negue acesso a arquivos com padrão de nome `.ht*`. Essa configuração impede que os clientes da Web exibam arquivos `.htaccess` e `.htpasswd`.

Você pode alterar qualquer uma das definições de configuração acima para seu ambiente. Para obter mais informações, consulte [Estender as plataformas Linux do Elastic Beanstalk](#). Expanda o tópico Reverse Proxy para ver a seção Configurando o Apache HTTPD.

Entre sistemas operacionais Amazon Linux

Para saber mais sobre as diferenças entre o Amazon Linux 2 e o Amazon Linux 2023, consulte [Comparação entre o Amazon Linux 2 e o Amazon Linux 2023](#) no Guia do usuário do Amazon Linux 2023.

Para saber mais sobre o Amazon Linux 2023, consulte [O que é o Amazon Linux 2023?](#) no Guia do usuário do Amazon Linux 2023.

## Processo geral de migração

Quando você estiver pronto para ir para produção, o Elastic Beanstalk exigirá uma implantação azul/verde para executar a atualização. As etapas gerais de práticas recomendadas para a migração com um procedimento de implantação azul/verde são descritas a seguir.

### Preparar para testar a migração

Antes de implantar a aplicação e começar a testá-la, revise as informações na seção [Diferenças e compatibilidade](#) anterior. Veja também a referência citada nessa seção, [Comparação entre o Amazon Linux 2 e o Amazon Linux 2023](#) no Guia do usuário do Amazon Linux 2023. Anote as informações específicas desse conteúdo que podem ser usadas na aplicação e na configuração definida.

### Etapas de migração de alto nível

1. Crie um novo ambiente baseado em uma ramificação da plataforma AL2023.
2. Implante a aplicação no ambiente AL2023 de destino.

O ambiente de produção existente permanecerá ativo e não será afetado, enquanto você estiver fazendo interações por meio de testes e ajustes no novo ambiente.

3. Teste a aplicação minuciosamente no novo ambiente.
4. Quando o novo ambiente AL2023 de destino estiver pronto para entrar em produção, troque os CNAMEs dos dois ambientes para redirecionar o tráfego para o novo ambiente AL2023.

### Etapas de migração mais detalhadas e práticas recomendadas

Para obter um procedimento de implantação azul/verde mais detalhado, consulte [Implantações azuis/verdes com o Elastic Beanstalk](#)

Para obter orientações mais específicas e etapas detalhadas de práticas recomendadas, consulte [Método azul/verde](#).

## Mais referências para ajudar a planejar a migração

As referências a seguir podem oferecer informações adicionais para planejar sua migração.

- [Plataformas compatíveis com Elastic Beanstalk](#) em Plataformas AWS Elastic Beanstalk
- [Histórico de ramificações de plataforma retiradas](#)
- [the section called “Plataformas Linux”](#)
- [Perguntas frequentes sobre descontinuação de plataformas](#)

## Migração do Amazon Linux AMI (AL1) para o AL2 ou o AL2023

Se a aplicação do Elastic Beanstalk for baseada em uma ramificação de plataforma do Amazon Linux AMI, use esta seção para saber como migrar os ambientes da aplicação para o Amazon Linux 2 ou Amazon Linux 2023. As ramificações da plataforma da geração anterior baseadas no [Amazon Linux AMI](#) foram descontinuadas.

É altamente recomendável migrar para o Amazon Linux 2023, já que ele é mais recente do que o Amazon Linux 2. O encerramento do suporte ao sistema operacional Amazon Linux 2 ocorrerá antes do encerramento do suporte ao Amazon Linux 2023. Consequentemente, você se beneficiará de um período de suporte mais longo se migrar para o Amazon Linux 2023.

Há um alto grau de compatibilidade entre as plataformas Amazon Linux 2 e Amazon Linux 2023 do Elastic Beanstalk. Ainda assim, há diferenças em algumas áreas: a opção padrão do Instance Metadata Service Version 1 (IMDSv1), o suporte à ferramenta de instância pkg-repo e algumas configurações do Apache HTTPd. Para obter mais informações, consulte [Amazon Linux 2023](#).

### Diferenças e compatibilidade

Não há garantia de que ramificações de plataformas baseadas em AL2023/AL2 sejam compatíveis com versões anteriores das aplicações existentes. É importante ter em mente que, mesmo que o código da aplicação seja implantado com sucesso na nova versão da plataforma, ele poderá se comportar ou ser executado de forma diferente devido às diferenças de runtime e sistema operacional.

Embora a Amazon Linux AMI e o AL2023/AL2 compartilhem o mesmo kernel do Linux, eles diferem nos seguintes aspectos: sistema de inicialização, versões de `libc`, cadeia de ferramentas do compilador e vários pacotes. Para obter mais informações, consulte [Perguntas frequentes do Amazon Linux 2](#).

O serviço Elastic Beanstalk também atualizou as versões de runtime, ferramentas de compilação e outras dependências específicas de cada plataforma.

Portanto, recomendamos que você reserve um tempo para testar seu aplicativo minuciosamente em um ambiente de desenvolvimento e faça os ajustes necessários.

## Processo geral de migração

Quando você estiver pronto para ir para produção, o Elastic Beanstalk exigirá uma implantação azul/verde para executar a atualização. As etapas gerais de práticas recomendadas para a migração com um procedimento de implantação azul/verde são descritas a seguir.

### Preparar para testar a migração

Antes de a aplicação e começar a testá-la, revise as informações em [Considerações para todas as plataformas Linux](#) que serão apresentadas posteriormente neste tópico. Além disso, revise as informações aplicáveis à sua plataforma na seção [Considerações específicas da plataforma](#) a seguir. Anote as informações específicas desse conteúdo que podem ser usadas na aplicação e na configuração definida.

### Etapas de migração de alto nível

1. Crie um novo ambiente baseado em uma ramificação da plataforma AL2 ou AL2023. Recomendamos migrar para uma ramificação da plataforma AL2023.
2. Implante a aplicação no ambiente AL2023/AL2 de destino.

O ambiente de produção existente permanecerá ativo e não será afetado, enquanto você estiver fazendo interações por meio de testes e ajustes no novo ambiente.

3. Teste a aplicação minuciosamente no novo ambiente.
4. Quando o novo ambiente AL2023/AL2 de destino estiver pronto para entrar em produção, troque os CNAMEs dos dois ambientes para redirecionar o tráfego para o novo ambiente.

### Etapas de migração mais detalhadas e práticas recomendadas

Para obter um procedimento de implantação azul/verde mais detalhado, consulte [Implantações azuis/verdes com o Elastic Beanstalk](#)

Para obter orientações mais específicas e etapas detalhadas de práticas recomendadas, consulte [Método azul/verde](#).



## Mais referências para ajudar a planejar a migração

As referências a seguir podem oferecer informações adicionais para planejar sua migração.

- [Comparação entre o Amazon Linux 2 e o Amazon Linux 2023](#) no Guia do usuário do Amazon Linux 2023.
- [O que é o Amazon Linux 2023?](#) no Guia do usuário Amazon Linux 2023.
- [Plataformas compatíveis com Elastic Beanstalk](#) em Plataformas AWS Elastic Beanstalk
- [Histórico de ramificações de plataforma retiradas](#)
- [the section called “Plataformas Linux”](#)
- [Perguntas frequentes sobre descontinuação de plataformas](#)

## Considerações para todas as plataformas Linux

A tabela a seguir discute considerações das quais você deve estar ciente ao planejar uma migração de aplicação para o AL2023/AL2. Essas considerações se aplicam a qualquer uma das plataformas Linux do Elastic Beanstalk, independentemente de linguagens de programação específicas ou servidores de aplicações.

Área	Alterações e informações
Arquivos de configuração	<p>Nas plataformas AL2023/AL2, é possível usar os <a href="#">arquivos de configuração</a> como antes, e todas as seções funcionarão da mesma maneira. No entanto, configurações específicas podem não funcionar da mesma forma que funcionavam em plataformas da AMI do Amazon Linux anteriores. Por exemplo:</p> <ul style="list-style-type: none"> <li>• Alguns pacotes de software instalados por você usando um arquivo de configuração podem não estar disponíveis no AL2023/AL2 ou seus nomes podem ter sido alterados.</li> <li>• Algumas opções de configuração específica da plataforma mudaram de seus namespaces específicos da plataforma para namespaces diferentes e independentes de plataforma.</li> <li>• Os arquivos de configuração de proxy fornecidos no diretório <code>.ebextensions/nginx</code> devem ser movidos para o diretório de hooks da plataforma <code>.platform/nginx</code>. Para obter detalhes, expanda a seção Reverse Proxy Configuration</li> </ul>

Área	Alterações e informações
	<p>(Configuração do proxy reverso) em <a href="#">the section called “Estender plataformas Linux”</a>.</p> <p>Recomendamos o uso de hooks de plataforma para executar o código personalizado em instâncias do seu ambiente. Você ainda pode usar comandos e os comandos de contêiner em arquivos de configuração <code>.ebextensions</code>, apesar de não ser fácil trabalhar com eles. Por exemplo, escrever scripts de comando dentro de um arquivo YAML pode ser complicado e difícil de testar.</p> <p>Você ainda precisa usar arquivos de configuração <code>.ebextensions</code> para qualquer script que precise de uma referência a um recurso do AWS CloudFormation.</p>
Hooks de plataforma	<p>As plataformas AL2 introduzem uma nova maneira de estender a plataforma do seu ambiente adicionando arquivos executáveis para criar hooks de diretórios nas instâncias do ambiente. É possível que você possa ter usado <a href="#">hooks de plataforma personalizada</a> com versões anteriores da plataforma Linux. Esses hooks não foram projetados para plataformas gerenciadas e não eram compatíveis, mas podiam funcionar de maneira útil em alguns casos. Os hooks de plataforma personalizada não funcionam nas versões de plataforma AL2023/AL2. Você deve migrar todos os hooks para os novos hooks de plataforma. Para obter detalhes, expanda a seção Platform Hooks (Hooks de plataforma) em <a href="#">the section called “Estender plataformas Linux”</a>.</p>

Área	Alterações e informações
Servidores de proxy compatíveis	<p>As versões de plataformas do AL2023/AL2 são compatíveis com os mesmos servidores de proxy reverso que cada plataforma compatível em suas versões de plataforma da Amazon Linux AMI. Todas as versões de plataforma do AL2023/AL2 usam o nginx como servidor de proxy reverso padrão, com exceção das plataformas ECS e Docker. As plataformas Tomcat, Node.js, PHP e Python também oferecem suporte ao Apache HTTPD como uma alternativa. Todas as plataformas habilitam a configuração do servidor de proxy de forma uniforme, conforme descrito nesta seção. No entanto, configurar o servidor de proxy é ligeiramente diferente do que estava na AMI do Amazon Linux. Estas são as diferenças para todas as plataformas:</p> <ul style="list-style-type: none"><li>• O padrão é o nginx: o servidor de proxy padrão em todas as versões de plataforma do AL2023/AL2 é o nginx. Nas versões da plataforma da AMI do Amazon Linux de Tomcat, PHP e Python, o servidor de proxy padrão era Apache HTTPD.</li><li>• Namespace consistente: todas as versões de plataforma do AL2023/AL2 usam o namespace <code>aws:elasticbeanstalk:environment:proxy</code> para configurar o servidor de proxy. Nas versões da plataforma da AMI do Amazon Linux, essa foi uma decisão de acordo com a plataforma, e o Node.js usava um namespace diferente.</li><li>• Localização do arquivo de configuração: é necessário colocar arquivos de configuração de proxy nos diretórios <code>.platform/nginx</code> e <code>.platform/httpd</code> em todas as versões de plataforma do AL2023/AL2. Nas versões da plataforma da AMI do Amazon Linux, esses locais eram <code>.ebextensions/nginx</code> e <code>.ebextensions/httpd</code>, respectivamente.</li></ul> <p>Para ver as alterações de configuração de proxy específicas da plataforma, consulte <a href="#">the section called “Considerações específicas da plataforma”</a>. Para obter informações sobre a configuração do proxy nas plataformas AL2023/AL2, expanda a seção Configuração do proxy reverso em <a href="#">the section called “Estender plataformas Linux”</a>.</p>

Área	Alterações e informações
Alteração de configuração de proxy	<p>Algumas alterações de configuração de proxy se aplicam uniformemente a todas as plataformas, mas há também alterações de configuração de proxy específicas de cada plataforma. É importante consultar ambas para configurar os ambientes com precisão.</p> <ul style="list-style-type: none"><li>• All Plataformas (Todas as plataformas): expanda a seção Reverse Proxy Configuration (Configuração do proxy reverso) em <a href="#">the section called “Estender plataformas Linux”</a>.</li><li>• Platform-specific (Específico da plataforma): consulte <a href="#">the section called “Considerações específicas da plataforma”</a>.</li></ul>
Perfil da instância	<p>As plataformas AL2023/AL2 exigem que um perfil de instância seja configurado. A criação do ambiente pode ser bem-sucedida temporariamente sem um perfil, mas o ambiente poderá mostrar erros logo após a criação, quando as ações que exigem um perfil de instância começarem a falhar. Para obter mais detalhes, consulte <a href="#">the section called “Perfis de instância”</a>.</p>
Integridade avançada	<p>As versões de plataforma AL2023/AL2 habilitam a integridade avançada por padrão. Trata-se de uma alteração, se você não usa o console do Elastic Beanstalk para criar seus ambientes. O console permite a integridade aprimorada por padrão sempre que possível, independentemente da versão da plataforma. Para obter mais detalhes, consulte <a href="#">the section called “Monitoramento e relatório de integridade aprimorada”</a>.</p>
AMI personalizada	<p>Se o ambiente usar uma <a href="#">AMI personalizada</a>, crie uma AMI baseada no AL2023/AL2 para o novo ambiente usando uma plataforma AL2023/AL2 do Elastic Beanstalk.</p>
Plataformas personalizadas	<p>As AMIs gerenciadas das versões de plataforma do AL2023/AL2 não oferecem suporte a <a href="#">plataformas personalizadas</a>.</p>

## Considerações específicas da plataforma

Esta seção aborda considerações de migração específicas para determinadas plataformas Linux do Elastic Beanstalk.

### Docker

A família de ramificações de plataforma Docker baseadas no Amazon Linux AMI (AL1) inclui três ramificações de plataforma. Recomendamos um caminho de migração específico para cada.

Ramificação de plataforma AL1	Caminho de migração para o AL2023/AL2
Docker de vários contêineres gerenciado pelo Amazon ECS sendo executado no Amazon Linux AMI (AL1)	<p>Ramificações de plataforma AL2023/AL2 do Docker baseada no ECS</p> <p>As ramificações de plataforma AL2023/AL2 do Docker baseadas no ECS oferecem um caminho de migração direta para ambientes em execução na ramificação da plataforma AL1 do Docker de vários contêineres.</p> <ul style="list-style-type: none"> <li>• Como na ramificação AL1 do Docker de vários contêineres anterior, as ramificações de plataformas AL2023/AL2 mais novas usam o Amazon ECS para coordenar uma implantação de vários contêineres do Docker em um cluster do Amazon ECS em um ambiente do Elastic Beanstalk.</li> <li>• As ramificações de plataforma AL2023/AL2 são compatíveis com todos os recursos da ramificação AL1 do Docker de vários contêineres anterior.</li> <li>• As ramificações de plataforma AL2023/AL2 também oferecem suporte ao mesmo arquivo <code>DockerRun.aws.json</code> v2.</li> </ul> <p>Para obter mais informações sobre a migração de aplicações executadas na ramificação da plataforma Amazon Linux do Docker de vários contêineres para a ramificação da plataforma Amazon ECS em execução no AL2023/AL2, consulte <a href="#">???</a>.</p>
Docker em execução no Amazon	Docker em execução na ramificação da plataforma AL2023/AL2

Ramificação de plataforma AL1	Caminho de migração para o AL2023/AL2				
Linux AMI (AL1)  Docker pré-configurado (Glassfish 5.0) em execução no Amazon Linux AMI (AL1)	<p>Recomendamos migrar as aplicações em execução em ambientes baseados no Docker pré-configurado (Glassfish 5.0) ou no Docker em execução no Amazon Linux AMI (AL1) para ambientes baseados nas ramificações de plataforma Docker em execução no Amazon Linux 2 ou Docker em execução no AL2023.</p> <p>Se o ambiente for baseado na ramificação de plataforma Docker pré-configurado (Glassfish 5.0), consulte <a href="#">the section called “Tutorial do GlassFish no Docker: caminho para o Amazon Linux 2023”</a>.</p> <p>A tabela a seguir lista informações de migração específicas para a ramificação da plataforma Docker em execução no AL2023/AL2.</p>				
	<table border="1"> <thead> <tr> <th data-bbox="321 898 488 972">Área</th> <th data-bbox="488 898 1507 972">Alterações e informações</th> </tr> </thead> <tbody> <tr> <td data-bbox="321 972 488 1829">Armazenamento</td> <td data-bbox="488 972 1507 1829"> <p>O Elastic Beanstalk configura o Docker para usar <a href="#">drivers de armazenamento</a> para armazenar dados de contêineres e imagens do Docker. Na AMI do Amazon Linux, o Elastic Beanstalk usou o <a href="#">driver de armazenamento Device Mapper</a>. Para melhorar a performance, o Elastic Beanstalk provisionou um volume extra do Amazon EBS. Nas versões da plataforma Docker AL2023/AL2, o Elastic Beanstalk usa o <a href="#">driver de armazenamento OverlayFS</a> e alcança uma performance ainda melhor sem precisar mais de um volume separado.</p> <p>Com a AMI do Amazon Linux, se você usou a opção <code>BlockDeviceMappings</code> do namespace <code>aws:autoscaling:launchconfiguration</code> para adicionar volumes de armazenamento personalizados a um ambiente do Docker, recomendamos que você também adicione o volume do Amazon EBS <code>/dev/xvdcz</code> provisionado pelo Elastic Beanstalk. O Elastic Beanstalk não provisiona mais esse volume, portanto, é necessário removê-lo dos arquivos de configuração. Para obter mais detalhes, consulte <a href="#">the</a></p> </td> </tr> </tbody> </table>	Área	Alterações e informações	Armazenamento	<p>O Elastic Beanstalk configura o Docker para usar <a href="#">drivers de armazenamento</a> para armazenar dados de contêineres e imagens do Docker. Na AMI do Amazon Linux, o Elastic Beanstalk usou o <a href="#">driver de armazenamento Device Mapper</a>. Para melhorar a performance, o Elastic Beanstalk provisionou um volume extra do Amazon EBS. Nas versões da plataforma Docker AL2023/AL2, o Elastic Beanstalk usa o <a href="#">driver de armazenamento OverlayFS</a> e alcança uma performance ainda melhor sem precisar mais de um volume separado.</p> <p>Com a AMI do Amazon Linux, se você usou a opção <code>BlockDeviceMappings</code> do namespace <code>aws:autoscaling:launchconfiguration</code> para adicionar volumes de armazenamento personalizados a um ambiente do Docker, recomendamos que você também adicione o volume do Amazon EBS <code>/dev/xvdcz</code> provisionado pelo Elastic Beanstalk. O Elastic Beanstalk não provisiona mais esse volume, portanto, é necessário removê-lo dos arquivos de configuração. Para obter mais detalhes, consulte <a href="#">the</a></p>
Área	Alterações e informações				
Armazenamento	<p>O Elastic Beanstalk configura o Docker para usar <a href="#">drivers de armazenamento</a> para armazenar dados de contêineres e imagens do Docker. Na AMI do Amazon Linux, o Elastic Beanstalk usou o <a href="#">driver de armazenamento Device Mapper</a>. Para melhorar a performance, o Elastic Beanstalk provisionou um volume extra do Amazon EBS. Nas versões da plataforma Docker AL2023/AL2, o Elastic Beanstalk usa o <a href="#">driver de armazenamento OverlayFS</a> e alcança uma performance ainda melhor sem precisar mais de um volume separado.</p> <p>Com a AMI do Amazon Linux, se você usou a opção <code>BlockDeviceMappings</code> do namespace <code>aws:autoscaling:launchconfiguration</code> para adicionar volumes de armazenamento personalizados a um ambiente do Docker, recomendamos que você também adicione o volume do Amazon EBS <code>/dev/xvdcz</code> provisionado pelo Elastic Beanstalk. O Elastic Beanstalk não provisiona mais esse volume, portanto, é necessário removê-lo dos arquivos de configuração. Para obter mais detalhes, consulte <a href="#">the</a></p>				

Ramificação de plataforma AL1	Caminho de migração para o AL2023/AL2	
	Área	Alterações e informações
		<a href="#">section called “Configuração do Docker na AMI do Amazon Linux (que precede o Amazon Linux 2)”</a> .
	Autenticação de repositório privado	Ao fornecer um arquivo de autenticação gerado pelo Docker para se conectar a um repositório privado, não é mais necessário convertê-lo no formato mais antigo exigido pelas versões da plataforma Docker da AMI do Amazon Linux. As versões da plataforma Docker AL2023/AL2 são compatíveis com o novo formato. Para obter mais detalhes, consulte <a href="#">the section called “Usar imagens de um repositório privado”</a> .
	Servidor de proxy	As versões da plataforma Docker do AL2023/AL2 não são compatíveis com contêineres autônomos que não são executados com um servidor de proxy. Nas versões da plataforma da AMI do Docker do Amazon Linux, isso costumava ser possível pelo valor none da opção ProxyServer no namespace <code>aws:elasticbeanstalk:environment:proxy</code> .

Go

A tabela a seguir lista as informações de migração para as versões da plataforma AL2023/AL2 na [plataforma Go](#).

Área	Alterações e informações
Transmissão de porta	Nas plataformas AL2023/AL2, o Elastic Beanstalk não passa um valor de porta para o processo de aplicação por meio da variável de ambiente PORT. Você pode simular esse comportamento para o seu processo configurando uma propriedade de ambiente PORT por conta própria. No entanto, se você tiver vários processos e estiver contando com a transmissão de valores incrementais de porta do Elastic

Área	Alterações e informações
	Beanstalk para os seus processos (5000, 5100, 5200 etc.), deverá modificar a implementação. Para obter detalhes, expanda a seção Configuração de proxy reverso em <a href="#">the section called “Estender plataformas Linux”</a> .

## Amazon Corretto

A tabela a seguir lista as informações de migração para as ramificações de plataforma do Corretto na [plataforma Java SE](#).

Área	Alterações e informações
Corretto vs. OpenJDK	Para implementar a plataforma Java, Standard Edition (Java SE), as ramificações de plataforma AL2023/AL2 usam o <a href="#">Amazon Corretto</a> , uma distribuição da AWS do Open Java Development Kit (OpenJDK). As ramificações anteriores da plataforma Java SE do Elastic Beanstalk usam os pacotes OpenJDK incluídos na AMI do Amazon Linux.
Ferramentas de compilação	As plataformas AL2023/AL2 têm versões mais recentes das ferramentas de compilação: gradle, maven e ant.
Tratamento de arquivos JAR	Nas plataformas AL2023/AL2, se o pacote de origem (arquivo ZIP) contiver um único arquivo JAR e nenhum outro arquivo, o Elastic Beanstalk não alterará mais o nome do arquivo JAR para <code>application.jar</code> . A renomeação ocorrerá somente se você enviar um arquivo JAR sozinho, e não dentro de um arquivo ZIP.
Transmissão de porta	Nas plataformas AL2023/AL2, o Elastic Beanstalk não passa um valor de porta para o processo de aplicação por meio da variável de ambiente PORT. Você pode simular esse comportamento para o seu processo configurando uma propriedade de ambiente PORT por conta própria. No entanto, se você tiver vários processos e estiver contando com a transmissão de valores incrementais de porta do Elastic Beanstalk para os seus processos (5000, 5100, 5200 etc.), deverá modificar a implementação. Para obter detalhes, expanda a seção Configuração de proxy reverso em <a href="#">the section called “Estender plataformas Linux”</a> .



Área	Alterações e informações
Java 7	O Elastic Beanstalk não é compatível com uma ramificação de plataforma Java 7 do AL2023/AL2. Se você tiver uma aplicação Java 7, migre-a para Corretto 8 ou Corretto 11.

## Tomcat

A tabela a seguir lista as informações de migração para as versões da plataforma AL2023/AL2 na [plataforma Tomcat](#).

Área	Alterações e informações						
Opções de configuração	<p>Nas versões da plataforma AL2023/AL2, o Elastic Beanstalk é compatível apenas com um subconjunto das opções de configuração e valores de opção no namespace <code>aws:elasticbeanstalk:environment:proxy</code>. Veja a seguir as informações sobre a migração para cada opção.</p> <table border="1"> <thead> <tr> <th>Opção</th> <th>Informações sobre migração</th> </tr> </thead> <tbody> <tr> <td><code>GzipCompression</code></td> <td>Não compatível em versões de plataforma AL2023/AL2.</td> </tr> <tr> <td><code>ProxyServer</code></td> <td> <p>As versões da plataforma Tomcat do AL2023/AL2 são compatíveis com os servidores de proxy nginx e Apache HTTPD versão 2.4. No entanto, não há suporte para o Apache versão 2.2.</p> <p>Nas versões da plataforma da AMI do Amazon Linux, o proxy padrão era Apache 2.4. Se você usou a configuração de proxy padrão e adicionou arquivos de configuração de proxy personalizados, a configuração de proxy ainda deverá funcionar no AL2023/AL2. No entanto, se você usou o valor da opção <code>apache/2.2</code>, agora você precisa migrar a configuração de proxy para o Apache versão 2.4.</p> </td> </tr> </tbody> </table> <p>A opção <code>XX:MaxPermSize</code> no namespace <code>aws:elasticbeanstalk:container:tomcat:jvmoptions</code> não é compatível com as versões</p>	Opção	Informações sobre migração	<code>GzipCompression</code>	Não compatível em versões de plataforma AL2023/AL2.	<code>ProxyServer</code>	<p>As versões da plataforma Tomcat do AL2023/AL2 são compatíveis com os servidores de proxy nginx e Apache HTTPD versão 2.4. No entanto, não há suporte para o Apache versão 2.2.</p> <p>Nas versões da plataforma da AMI do Amazon Linux, o proxy padrão era Apache 2.4. Se você usou a configuração de proxy padrão e adicionou arquivos de configuração de proxy personalizados, a configuração de proxy ainda deverá funcionar no AL2023/AL2. No entanto, se você usou o valor da opção <code>apache/2.2</code>, agora você precisa migrar a configuração de proxy para o Apache versão 2.4.</p>
Opção	Informações sobre migração						
<code>GzipCompression</code>	Não compatível em versões de plataforma AL2023/AL2.						
<code>ProxyServer</code>	<p>As versões da plataforma Tomcat do AL2023/AL2 são compatíveis com os servidores de proxy nginx e Apache HTTPD versão 2.4. No entanto, não há suporte para o Apache versão 2.2.</p> <p>Nas versões da plataforma da AMI do Amazon Linux, o proxy padrão era Apache 2.4. Se você usou a configuração de proxy padrão e adicionou arquivos de configuração de proxy personalizados, a configuração de proxy ainda deverá funcionar no AL2023/AL2. No entanto, se você usou o valor da opção <code>apache/2.2</code>, agora você precisa migrar a configuração de proxy para o Apache versão 2.4.</p>						

Área	Alterações e informações
	de plataforma AL2023/AL2. A configuração de JVM para modificar o tamanho da geração permanente se aplica somente ao Java 7 e anterior e, portanto, não é aplicável às versões de plataforma AL2023/AL2.
Caminho do aplicativo	Nas plataformas AL2023/AL2, o caminho para o diretório da aplicação nas instâncias do Amazon EC2 do ambiente é <code>/var/app/current</code> . Era <code>/var/lib/tomcat8/webapps</code> em plataformas da AMI do Amazon Linux.

## Node.js

A tabela a seguir lista as informações de migração para as versões de plataforma AL2023/AL2 na [plataforma Node.js](#).

Área	Alterações e informações
Versões Node.js instaladas	<p>Em plataformas AL2023/AL2, o Elastic Beanstalk mantém várias ramificações da plataforma Node.js e instala somente a versão mais recente da versão principal do Node.js correspondente à ramificação da plataforma em cada versão. Por exemplo, cada versão da plataforma na ramificação da plataforma Node.js 12 tem somente o Node.js 12.x.y instalado por padrão. Nas versões da plataforma da AMI do Amazon Linux, instalamos as várias versões do Node.js em cada versão da plataforma e mantemos apenas uma única ramificação da plataforma.</p> <p>Escolha a ramificação da plataforma Node.js que corresponde à versão principal do Node.js de que seu aplicativo precisa.</p>
Nomes de arquivos de log do Apache HTTPD	<p>Nas plataformas AL2023/AL2, se você usar o servidor de proxy Apache HTTPD, os nomes dos arquivos de log HTTPD serão <code>access_log</code> e <code>error_log</code>, o que é consistente com todas as outras plataformas compatíveis com o Apache HTTPD. Nas versões da plataforma da AMI do Amazon Linux, esses arquivos de log eram chamados de <code>access.log</code> e <code>error.log</code>, respectivamente.</p> <p>Para obter detalhes sobre nomes de arquivos de log e locais de todas as plataformas, consulte <a href="#">the section called “Como o Elastic Beanstalk configura o CloudWatch Logs”</a>.</p>

Área	Alterações e informações										
Opções de configuração	<p>Nas plataformas AL2023/AL2, o Elastic Beanstalk não é compatível com as opções de configuração no namespace <code>aws:elasticbeanstalk:container:nodejs</code>. Algumas das opções têm alternativas. Veja a seguir as informações sobre a migração para cada opção.</p> <table border="1"> <thead> <tr> <th>Opção</th> <th>Informações sobre migração</th> </tr> </thead> <tbody> <tr> <td><code>NodeCommand</code></td> <td>Use um <code>scripts</code> ou a palavra-chave <code>Procfile</code> em um arquivo <code>package.json</code> para especificar o script de início.</td> </tr> <tr> <td><code>NodeVersion</code></td> <td>Use a palavra-chave <code>engines</code> em um arquivo <code>package.json</code> para especificar a versão do Node.js. Lembre-se de que é possível especificar somente uma versão do Node.js que corresponda à sua ramificação da plataforma. Por exemplo, se você estiver usando a ramificação da plataforma Node.js 12, será possível especificar somente uma versão 12.x.y do Node.js. Para obter mais detalhes, consulte <a href="#">the section called “Especificar as dependências do Node.js com um arquivo package.json”</a>.</td> </tr> <tr> <td><code>GzipCompression</code></td> <td>Não compatível em versões de plataforma AL2023/AL2.</td> </tr> <tr> <td><code>ProxyServer</code></td> <td> <p>Nas versões da plataforma Node.js do AL2023/AL2, essa opção foi movida para o namespace <code>aws:elasticbeanstalk:environment:proxy</code>. É possível escolher entre <code>nginx</code> (o padrão) e <code>apache</code>.</p> <p>As versões da plataforma Node.js do AL2023/AL2 não são compatíveis com aplicações autônomas que não são executadas por trás de um servidor de proxy. Nas versões da plataforma Node.js da AMI do Amazon Linux, isso costumava ser possível pelo valor <code>none</code> da opção <code>ProxyServer</code> no namespace <code>aws:elasticbeanstalk:container:nodejs</code>. Se o ambiente executar uma aplicação autônoma, atualize o código para escutar a porta</p> </td> </tr> </tbody> </table>	Opção	Informações sobre migração	<code>NodeCommand</code>	Use um <code>scripts</code> ou a palavra-chave <code>Procfile</code> em um arquivo <code>package.json</code> para especificar o script de início.	<code>NodeVersion</code>	Use a palavra-chave <code>engines</code> em um arquivo <code>package.json</code> para especificar a versão do Node.js. Lembre-se de que é possível especificar somente uma versão do Node.js que corresponda à sua ramificação da plataforma. Por exemplo, se você estiver usando a ramificação da plataforma Node.js 12, será possível especificar somente uma versão 12.x.y do Node.js. Para obter mais detalhes, consulte <a href="#">the section called “Especificar as dependências do Node.js com um arquivo package.json”</a> .	<code>GzipCompression</code>	Não compatível em versões de plataforma AL2023/AL2.	<code>ProxyServer</code>	<p>Nas versões da plataforma Node.js do AL2023/AL2, essa opção foi movida para o namespace <code>aws:elasticbeanstalk:environment:proxy</code>. É possível escolher entre <code>nginx</code> (o padrão) e <code>apache</code>.</p> <p>As versões da plataforma Node.js do AL2023/AL2 não são compatíveis com aplicações autônomas que não são executadas por trás de um servidor de proxy. Nas versões da plataforma Node.js da AMI do Amazon Linux, isso costumava ser possível pelo valor <code>none</code> da opção <code>ProxyServer</code> no namespace <code>aws:elasticbeanstalk:container:nodejs</code>. Se o ambiente executar uma aplicação autônoma, atualize o código para escutar a porta</p>
Opção	Informações sobre migração										
<code>NodeCommand</code>	Use um <code>scripts</code> ou a palavra-chave <code>Procfile</code> em um arquivo <code>package.json</code> para especificar o script de início.										
<code>NodeVersion</code>	Use a palavra-chave <code>engines</code> em um arquivo <code>package.json</code> para especificar a versão do Node.js. Lembre-se de que é possível especificar somente uma versão do Node.js que corresponda à sua ramificação da plataforma. Por exemplo, se você estiver usando a ramificação da plataforma Node.js 12, será possível especificar somente uma versão 12.x.y do Node.js. Para obter mais detalhes, consulte <a href="#">the section called “Especificar as dependências do Node.js com um arquivo package.json”</a> .										
<code>GzipCompression</code>	Não compatível em versões de plataforma AL2023/AL2.										
<code>ProxyServer</code>	<p>Nas versões da plataforma Node.js do AL2023/AL2, essa opção foi movida para o namespace <code>aws:elasticbeanstalk:environment:proxy</code>. É possível escolher entre <code>nginx</code> (o padrão) e <code>apache</code>.</p> <p>As versões da plataforma Node.js do AL2023/AL2 não são compatíveis com aplicações autônomas que não são executadas por trás de um servidor de proxy. Nas versões da plataforma Node.js da AMI do Amazon Linux, isso costumava ser possível pelo valor <code>none</code> da opção <code>ProxyServer</code> no namespace <code>aws:elasticbeanstalk:container:nodejs</code>. Se o ambiente executar uma aplicação autônoma, atualize o código para escutar a porta</p>										

Área	Alterações e informações	
	Opção	Informações sobre migração
		<p>para a qual servidor de proxy (nginx ou Apache) encaminha o tráfego.</p> <pre>var port = process.env.PORT    5000;  app.listen(port, function() {   console.log('Server running at http://127.0.0.1:%s',     port); });</pre>

## PHP

A tabela a seguir lista as informações de migração para as versões da plataforma AL2023/AL2 na [plataforma PHP](#).

Área	Alterações e informações
Processamento de arquivos PHP	Nas plataformas AL2023/AL2, os arquivos PHP são processados usando PHP-FPM (um gerenciador de processos CGI). Em plataformas da AMI do Amazon Linux, usamos mod_php (um módulo do Apache).
Servidor de proxy	<p>As versões da plataforma PHP do AL2023/AL2 são compatíveis com os servidores de proxy nginx e Apache HTTPD. O padrão é nginx.</p> <p>As versões da plataforma PHP da AMI do Amazon Linux eram compatíveis apenas com o Apache HTTPD. Se você adicionou arquivos de configuração personalizados do Apache, poderá definir a opção <code>ProxyServer</code> no namespace <code>aws:elasticbeanstalk:environment:proxy</code> como <code>apache</code>.</p>

## Python

A tabela a seguir lista as informações de migração para as versões da plataforma AL2023/AL2 na [plataforma Python](#).

Área	Alterações e informações
Servidor WSGI	<p>Nas plataformas AL2023/AL2, o <a href="#">Gunicorn</a> é o servidor WSGI padrão. Por padrão, o Gunicorn escuta na porta 8000. A porta pode ser diferente da utilizada pela aplicação na plataforma da AMI do Amazon Linux. Se você estiver definindo a opção <code>WSGIPath</code> do namespace <code>aws:elasticbeanstalk:container:python</code>, substitua o valor pela sintaxe de Gunicorn. Para obter mais detalhes, consulte <a href="#">the section called “Namespaces de configuração do Python”</a>.</p> <p>Como alternativa, você pode usar um <code>Procfile</code> para especificar e configurar o servidor WSGI. Para obter mais detalhes, consulte <a href="#">the section called “Procfile”</a>.</p>
Caminho do aplicativo	Nas plataformas AL2023/AL2, o caminho para o diretório da aplicação nas instâncias do Amazon EC2 do ambiente é <code>/var/app/current</code> . Era <code>/opt/python/current/app</code> em plataformas da AMI do Amazon Linux.
Servidor de proxy	<p>As versões da plataforma Python do AL2023/AL2 são compatíveis com os servidores de proxy nginx e Apache HTTPD. O padrão é nginx.</p> <p>As versões da plataforma Python da AMI do Amazon Linux eram compatíveis apenas com o Apache HTTPD. Se você adicionou arquivos de configuração personalizados do Apache, poderá definir a opção <code>ProxyServer</code> no namespace <code>aws:elasticbeanstalk:environment:proxy</code> como <code>apache</code>.</p>

## Ruby

A tabela a seguir lista as informações de migração para as versões da plataforma AL2023/AL2 na [plataforma Ruby](#).

Área	Alterações e informações
Versões Ruby instaladas	Nas plataformas AL2023/AL2, o Elastic Beanstalk instala somente a versão mais recente de uma única versão do Ruby, correspondente à ramificação da plataforma, em cada versão da plataforma. Por exemplo, cada versão da plataforma na ramificação da plataforma Ruby 2.6 tem somente o Ruby 2.6.x instalado. Nas versões da plataforma da AMI do Amazon Linux, instalamos as versões mais recentes de várias versões do Ruby, por exemplo, 2.4.x, 2.5.x e 2.6.x.

Área	Alterações e informações
	Se seu aplicativo usar uma versão do Ruby que não corresponda à ramificação da plataforma que você está usando, recomendamos mudar para uma ramificação de plataforma que tenha a versão do Ruby correta para seu aplicativo.
Servidor de aplicativos	<p>Nas plataformas AL2023/AL2, o Elastic Beanstalk instala apenas o servidor de aplicações Puma em todas as versões da plataforma Ruby. É possível usar um <code>Procfile</code> para iniciar um servidor de aplicativos diferente e um <code>Gemfile</code> para instalá-lo.</p> <p>Na plataforma da AMI do Amazon Linux, oferecemos suporte a dois tipos de ramificações da plataforma para cada versão Ruby: uma com o servidor de aplicações Puma e outra com o servidor de aplicações Passenger. Se o seu aplicativo usar o Passenger, é possível configurar seu ambiente Ruby para instalar e usar o Passenger.</p> <p>Para ter mais informações e exemplos, consulte <a href="#">the section called “A Plataforma Ruby”</a>.</p>

## Perguntas frequentes sobre descontinuação de plataformas

### Note

O Elastic Beanstalk descontinuou todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) em 18 de julho de 2022.

As respostas destas Perguntas frequentes referenciam os seguintes tópicos:

- [Política de suporte às plataformas do Elastic Beanstalk](#)
- [Histórico de ramificações de plataforma retiradas](#)
- [Plataformas compatíveis com Elastic Beanstalk](#) em Plataformas AWS Elastic Beanstalk
- [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#)
- [Perguntas frequentes do Amazon Linux 2](#).

## 1. O que significa a retirada de uma ramificação de plataforma?

Após a data de descontinuação de uma ramificação de plataforma ser anunciada, você não poderá mais criar um novo ambiente baseado na ramificação de plataforma retirada, a menos que já tenha um ambiente ativo baseado nessa ramificação de plataforma. Para obter mais informações, consulte a [pergunta frequentes nº 11](#). O Elastic Beanstalk deixará de fornecer novas atualizações de manutenção para essas ramificações de plataforma. Não é recomendável usar uma ramificação de plataforma retirada em ambientes de produção. Para obter mais informações, consulte a [pergunta frequentes nº 5](#).

## 2. Por que AWS aposentou as filiais de plataformas baseadas em AL1?

O Elastic Beanstalk retira ramificações de plataforma quando os componentes da plataforma ficam obsoletos ou são retirados pelos fornecedores. No presente caso, o Amazon Linux AMI (AL1) encerrou o suporte padrão a partir de [31 de dezembro de 2020](#). Embora o Elastic Beanstalk tenha continuado a oferecer plataformas baseadas no AL1 até 2022, desde então lançamos muitas plataformas baseadas no AL2 e AL2023 que oferecem os recursos mais recentes. Para que os clientes continuem a se beneficiar da segurança e dos recursos mais recentes, é fundamental que migrem para plataformas baseadas em AL2 ou AL2023.

## 3. Quais ramificações de plataforma serão retiradas?

Para obter uma lista de componentes de plataforma e ramificações de plataforma que foram retirados, consulte [Histórico de ramificações de plataforma retiradas](#).

## 4. Quais são as plataformas compatíveis atualmente?

Consulte [Plataformas compatíveis com Elastic Beanstalk](#) em Plataformas do AWS Elastic Beanstalk

## 5. O Elastic Beanstalk removerá ou encerrará algum componente do ambiente após a retirada?

Nossa política para filiais de plataformas desativadas não remove o acesso aos ambientes nem exclui recursos. Porém, um ambiente baseado em uma ramificação de plataforma retirada pode levar a uma situação imprevisível, pois o Elastic Beanstalk não pode fornecer atualizações de segurança, suporte técnico ou hotfixes para ramificações de plataforma retiradas, uma vez que o fornecedor o componente definiu que o componente atingiu o Fim da Vida Útil (EOL). Por exemplo,

uma vulnerabilidade de segurança prejudicial e crítica pode surgir em um ambiente sendo executado em uma ramificação de plataforma retirada. Ou uma ação da API do EB pode parar de funcionar no ambiente se, com o tempo, tornar-se incompatível com o serviço Elastic Beanstalk. Quanto mais tempo um ambiente baseado em uma ramificação de plataforma retirada permanecer ativo, maior a chance de que esses tipos de riscos ocorram.

Se seu aplicativo tiver problemas ao ser executado em uma ramificação de plataforma descontinuada e você não conseguir migrá-lo para uma plataforma compatível, precisará considerar outras alternativas. Soluções alternativas incluem o encapsulamento da aplicação em uma imagem do Docker para executá-la como um contêiner do Docker. Isso permitiria que um cliente usasse qualquer uma de nossas soluções Docker, como nossas plataformas Docker Elastic Beanstalk AL2023/AL2 ou outros serviços baseados em Docker, como Amazon ECS ou Amazon EKS. As alternativas que não são do Docker incluem nosso AWS CodeDeploy serviço, que permite a personalização completa dos tempos de execução que você deseja.

## 6. Posso enviar uma solicitação para adiar a data de retirada?

Não. Após a data de retirada, os ambientes existentes continuarão a funcionar. Porém, o Elastic Beanstalk não fornecerá mais atualizações de manutenção e segurança da plataforma. Portanto, é fundamental migrar para AL2 ou AL2023 se você ainda executa aplicações em uma plataforma baseada em AL1. Para obter mais informações sobre riscos e soluções alternativas, consulte a [pergunta frequente nº 5](#).

## 7. Quais serão as soluções alternativas se eu não puder concluir a migração para o AL2 ou AL2023 no prazo?

Os clientes podem continuar a executar o ambiente, embora seja muito recomendável planejar a migração de todos os ambientes do Elastic Beanstalk para uma versão de plataforma compatível. Isso minimizará o risco e proporcionará os benefícios contínuos das importantes melhorias de funcionalidade, performance e segurança oferecidas nas versões mais recentes. Para obter mais informações sobre riscos e soluções alternativas, consulte a [pergunta frequente nº 5](#).

## 8. Qual é o processo recomendado para migrar para as plataformas AL2 ou AL2023?

Para obter instruções abrangentes para a migração do AL1 para o AL2023/AL2, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#). Este tópico explica que o Elastic Beanstalk exige uma implantação azul/verde para realizar a atualização.



## 9. Se eu tiver um ambiente em execução em uma plataforma retirada, qual seria o impacto?

Um ambiente baseado em uma ramificação de plataforma retirada pode levar a uma situação imprevisível, pois o Elastic Beanstalk não pode fornecer atualizações de segurança, suporte técnico ou hotfixes para ramificações de plataforma retiradas, uma vez que o fornecedor definiu que o componente atingiu o Fim da Vida Útil (EOL). Por exemplo, uma vulnerabilidade de segurança prejudicial e crítica pode surgir em um ambiente sendo executado em uma ramificação de plataforma retirada. Ou uma ação da API do EB pode parar de funcionar no ambiente se, com o tempo, tornar-se incompatível com o serviço Elastic Beanstalk. Quanto mais tempo um ambiente baseado em uma ramificação de plataforma retirada permanecer ativo, maior a chance de que esses tipos de riscos ocorram. Para obter mais informações, consulte a [pergunta frequentes nº 5](#).

## 10. O que acontece 90 dias após a data de retirada?

Nossa política para filiais de plataformas desativadas não remove o acesso aos ambientes nem exclui recursos. Porém, lembre-se de que um ambiente baseado em uma ramificação de plataforma retirada pode levar a uma situação imprevisível, pois o Elastic Beanstalk não pode fornecer atualizações de segurança, suporte técnico ou hotfixes para ramificações de plataforma retiradas, uma vez que o fornecedor definiu que o componente atingiu o Fim da Vida Útil (EOL). Por exemplo, uma vulnerabilidade de segurança prejudicial e crítica pode surgir em um ambiente sendo executado em uma ramificação de plataforma retirada. Ou uma ação da API do EB pode parar de funcionar no ambiente se, com o tempo, tornar-se incompatível com o serviço Elastic Beanstalk. Quanto mais tempo um ambiente baseado em uma ramificação de plataforma retirada permanecer ativo, maior a chance de que esses tipos de riscos ocorram. Para obter mais informações, consulte a [pergunta frequentes nº 5](#).

## 11. Posso criar um novo ambiente baseado em uma plataforma retirada?

Você pode criar um novo ambiente baseado em uma ramificação de plataforma retirada se já tiver usado essa ramificação de plataforma para criar um ambiente existente usando a mesma conta e na mesma região. A ramificação descontinuada da plataforma não estará disponível no console do Elastic Beanstalk. Porém, para clientes que tenham ambientes existentes baseados em uma ramificação de plataforma retirada, ela estará disponível por meio da CLI do EB, da API do EB e da AWS CLI. Além disso, os clientes existentes podem usar os consoles [Clone environment](#) (Clonar ambiente) e [Rebuild environment](#) (Recriar ambiente). Porém, lembre-se de que um ambiente baseado em uma ramificação de plataforma retirada pode levar a uma situação imprevisível. Para obter mais informações, consulte a [pergunta frequentes nº 5](#).

12. Se eu tiver um ambiente existente em execução em uma ramificação de plataforma desativada, até quando poderei criar um novo ambiente com base na ramificação de plataforma descontinuada? Posso fazer isso usando console, CLI ou API?

Você pode criar o ambiente após a data de aposentadoria. Porém, lembre-se de que uma ramificação de plataforma retirada pode levar a uma situação imprevisível. Quanto mais tempo após a retirada esse ambiente for criado ou permanecer ativo, maior será o risco de problemas inesperados. Para obter mais informações sobre a criação de um novo ambiente de desenvolvimento, consulte a [pergunta frequente nº 11](#).

13. É possível clonar ou recriar um ambiente baseado em plataforma retirada?

Sim. Você pode fazer isso usando os consoles [Clone environment](#) (Clonar ambiente) e [Rebuild environment](#) (Recriar ambiente). Você também pode usar a CLI do EB, a API do EB e. AWS CLI. Para obter mais informações sobre a criação de um novo ambiente de desenvolvimento, consulte a [pergunta frequente nº 11](#).

Porém, é muito recomendável planejar a migração de todos os ambientes do Elastic Beanstalk para uma versão de plataforma compatível. Isso minimizará o risco e proporcionará os benefícios contínuos das importantes melhorias de funcionalidade, performance e segurança oferecidas nas versões mais recentes. Para obter mais informações sobre riscos e soluções alternativas, consulte a [pergunta frequente nº 5](#).

14. Após a data de aposentadoria, o que aconteceria com os AWS recursos do meu ambiente do Elastic Beanstalk, baseado em uma ramificação de plataforma desativada? Por exemplo, se a instância do EC2 em execução for encerrada, o Elastic Beanstalk poderia iniciar uma nova instância do EC2 baseada no AL1 para manter capacidade?

Os recursos do ambiente permaneceriam ativos e continuariam funcionando. E, sim, o Elastic Beanstalk será dimensionado automaticamente para as instâncias do EC2 no AL1 do ambiente. Porém, o Elastic Beanstalk deixará de fornecer novas atualizações de manutenção da plataforma para o ambiente, o que pode, com o tempo, levar o ambiente a uma situação imprevisível. Para obter mais informações, consulte a [pergunta frequentes nº 5](#).

## 15. Quais são as principais diferenças entre os sistemas operacionais AL2023/AL2 e Amazon Linux AMI (AL1)? Como as ramificações de plataforma AL2023/AL2 do Elastic Beanstalk são afetadas?

Embora o Amazon Linux AMI e o AL2023/AL2 compartilhem o mesmo kernel do Linux, eles diferem em sistema de inicialização, nas versões de `libc`, na cadeia de ferramentas do compilador e em vários pacotes. Para obter mais informações, consulte [Perguntas frequentes do Amazon Linux 2](#).

O serviço Elastic Beanstalk também atualizou as versões de runtime, ferramentas de compilação e outras dependências específicas de cada plataforma. Não há garantia de que ramificações de plataformas baseadas em AL2023/AL2 sejam compatíveis com versões anteriores das aplicações existentes. Além disso, mesmo que o código do aplicativo seja implantado com sucesso na nova versão da plataforma, ele pode se comportar ou ser executado de forma diferente devido às diferenças de runtime e sistema operacional. Para obter uma lista e a descrição das configurações e personalizações que você precisará revisar e testar, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

## Cancelar atualizações de configuração do ambiente e implantações de aplicativos

Você pode cancelar atualizações em andamento que são acionadas por alterações da configuração do ambiente. Também pode cancelar a implantação de uma nova versão do aplicativo em andamento. Por exemplo, pode ser necessário cancelar uma atualização caso você decida continuar usando a configuração de ambiente existente em vez de aplicar novas configurações do ambiente. Ou talvez você perceba que a nova versão do aplicativo que está sendo implantada tem problemas que impedirão sua inicialização ou execução correta. Cancelando uma atualização de ambiente ou de versão de aplicativo, você pode iniciar uma nova tentativa de atualizar o ambiente ou a versão do aplicativo sem ter que esperar até o fim do processo de atualização ou implantação.

### Note

Durante a fase de limpeza em que recursos antigos desnecessários são removidos, após a atualização do último lote de instâncias, você não pode mais cancelar a atualização.

O Elastic Beanstalk executa a reversão do mesmo modo como executou a última atualização bem-sucedida. Por exemplo, se as atualizações contínuas baseadas no tempo estiverem habilitadas


no seu ambiente, o Elastic Beanstalk aguardará o tempo de espera especificado entre a reversão das alterações em um lote de instâncias antes de reverter as alterações do próximo lote. Caso você tenha ativado recentemente as atualizações contínuas, mas a última atualização que fez nas configurações do ambiente não teve atualizações contínuas, o Elastic Beanstalk executará a reversão em todas as instâncias simultaneamente.

Não é possível impedir que o Elastic Beanstalk reverta à configuração de ambiente anterior depois que ele começa a cancelar a atualização. O processo de reversão continua até que todas as instâncias no ambiente tenham a configuração de ambiente anterior ou até que o processo de reversão falhe. Para implantações de versão do aplicativo, o cancelamento da implantação apenas interrompe a implantação. Algumas instâncias terão a nova versão do aplicativo enquanto outras continuarão a executar a versão do aplicativo existente. Você pode implantar a mesma versão do aplicativo ou implantar outra mais tarde.

Para obter mais informações sobre atualizações contínuas, consulte [Atualizações de configuração de ambiente contínuas do Elastic Beanstalk](#). Para obter mais informações sobre as implantações de versão do aplicativo em lote, consulte [Políticas e configurações de implantação](#).

Para cancelar uma atualização

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, escolha Actions (Ações) e depois Abort current operation (Abortar operação atual).

## Recriar ambientes do Elastic Beanstalk

O seu ambiente do AWS Elastic Beanstalk poderá ficar inutilizável se você não usar a funcionalidade do Elastic Beanstalk para modificar ou terminar os recursos da AWS subjacentes do ambiente. Se isso acontecer, você poderá reconstruir o ambiente para tentar restaurá-lo a um estado operacional. A reconstrução de um ambiente encerra todos os seus recursos e os substitui por novos recursos com a mesma configuração.

Você também pode reconstruir ambientes em até seis semanas (42 dias) de seu encerramento. Ao recriar, o Elastic Beanstalk tenta criar outro ambiente com o mesmo nome, ID e configuração.

## Reconstrução de um ambiente em execução

É possível recriar um ambiente por meio do console do Elastic Beanstalk ou usando a API `RebuildEnvironment`.

Para reconstruir um ambiente em execução (console)

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions e Rebuild environment.
4. Escolha Rebuild.

A reconstrução de um ambiente em execução cria novos recursos com a mesma configuração dos antigos, no entanto, os IDs de recurso são diferentes e os dados dos recursos antigos não são restaurados. Por exemplo, a recriação de um ambiente com uma instância de banco de dados do Amazon RDS cria outro banco de dados com a mesma configuração, mas não aplica um snapshot ao novo banco de dados.

Para reconstruir um ambiente em execução com a API do Elastic Beanstalk, use a ação [RebuildEnvironment](#) com a AWS CLI ou o AWS SDK.

```
$ aws elasticbeanstalk rebuild-environment --environment-id e-vdnftxubwq
```

## Reconstruir um ambiente encerrado

É possível recriar e restaurar um ambiente encerrado ao usar o console do Elastic Beanstalk, a CLI do EB ou a API `RebuildEnvironment`.

**Note**

A menos que você esteja usando seu próprio nome de domínio personalizado com o ambiente encerrado, o ambiente usará um subdomínio de elasticbeanstalk.com. Esses subdomínios são compartilhados dentro de uma região do Elastic Beanstalk. Portanto, eles podem ser usados por qualquer ambiente criado por qualquer cliente na mesma região. Embora seu ambiente esteja encerrado, um outro ambiente pode usar seu subdomínio. Nesse caso, a reconstrução não ocorrerá.

Evite esse problema usando um domínio personalizado. Para mais detalhes, consulte [Nome de domínio do ambiente do Elastic Beanstalk](#).

Os ambientes encerrados recentemente são exibidos na visão geral do aplicativo por até uma hora. Durante esse período, você pode visualizar eventos do ambiente em seu [painel](#) e usar a ação [Restore environment](#) para reconstruí-lo.

Para reconstruir um ambiente que não está mais visível, use a opção Restore terminated environment na página do aplicativo.

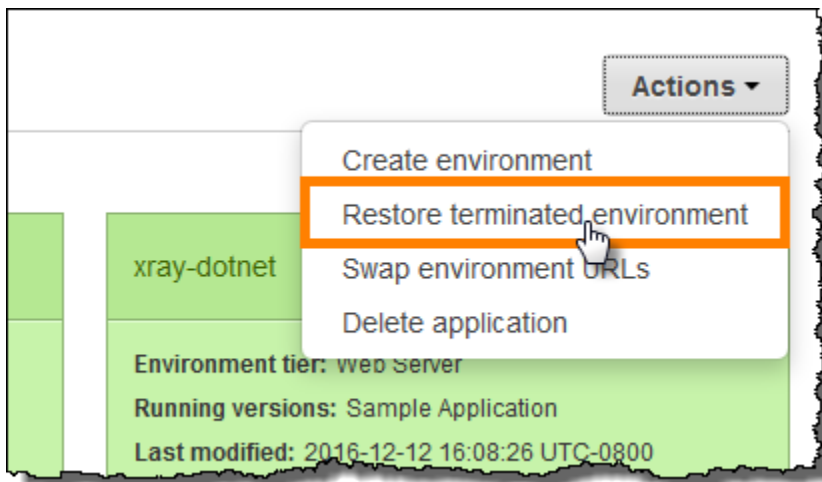
Para reconstruir um ambiente encerrado (console)

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

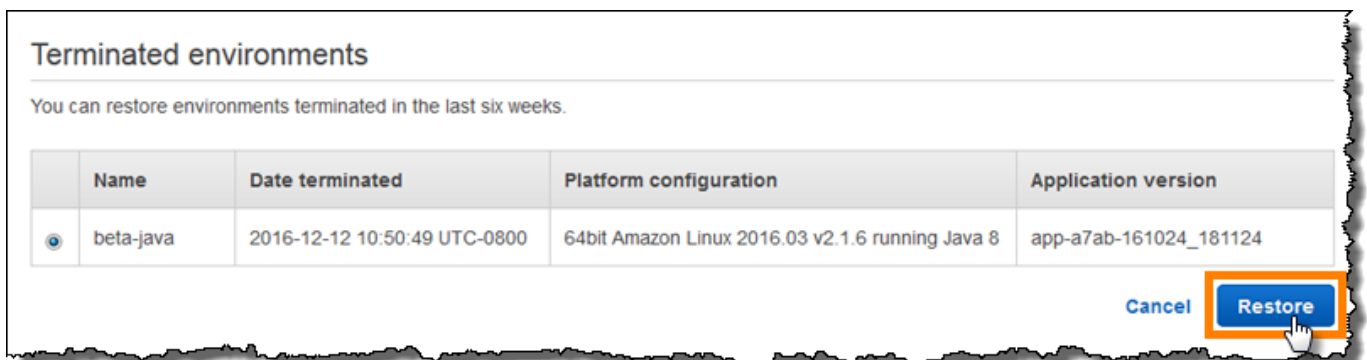
**Note**

Se você tiver muitos aplicativos, use a barra de pesquisa para filtrar a lista de aplicativos.

3. Escolha Actions e, em seguida, Restore terminated environment.



4. Escolha um ambiente encerrado.
5. Escolha Restore.



O Elastic Beanstalk tenta criar um novo ambiente com o mesmo nome, ID e configuração. Se já existir um ambiente com o mesmo nome ou URL, quando você tentar reconstruir, a ação falhará. A exclusão da versão do aplicativo que foi implantada no ambiente também causará falha na reconstrução.

Se você usa a EB CLI para gerenciar seu ambiente, use o comando `eb restore` para reconstruir um ambiente encerrado.

```
$ eb restore e-vdnftxubwq
```

Consulte [eb restore](#) para obter mais informações.

Para reconstruir um ambiente terminado com a API do Elastic Beanstalk, use a ação [RebuildEnvironment](#) com a AWS CLI ou o AWS SDK.

```
$ aws elasticbeanstalk rebuild-environment --environment-id e-vdnftxubwq
```

## Tipos de ambiente

No AWS Elastic Beanstalk, é possível criar um ambiente escalável e com balanceamento de carga ou um ambiente de instância única. O tipo de ambiente necessário depende do aplicativo que você implanta. Por exemplo, é possível desenvolver e testar um aplicativo em um ambiente de instância única para reduzir custos e, depois, atualizá-lo para um ambiente escalável e com balanceamento de carga quando a aplicação estiver pronta para produção.

### Note

Uma camada de ambiente de operador para um aplicativo web que processa tarefas em segundo plano não inclui um load balancer. No entanto, um ambiente de operador é expandido de modo eficiente adicionando instâncias ao grupo de Auto Scaling para processar dados da fila do Amazon SQS quando a carga necessita.

## Ambiente escalável e com balanceamento de carga

Um ambiente com carga balanceada e escalável usa os serviços de Auto Scaling do Amazon EC2 e Elastic Load Balancing para provisionar as instâncias do Amazon EC2 necessárias para a aplicação implantada. O Auto Scaling do Amazon EC2 inicia automaticamente as instâncias adicionais para acomodar a crescente carga na aplicação. Se a carga na aplicação diminuir, o Auto Scaling do Amazon EC2 interromperá as instâncias, mas deixando sempre o número mínimo especificado de instâncias em execução. Se a aplicação exigir escalabilidade com a opção de execução em várias zonas de disponibilidade, use um ambiente escalável e com balanceamento de carga. Se não tiver certeza sobre qual tipo de ambiente selecionar, você poderá escolher um e, se necessário, mudar o tipo do ambiente mais tarde.

## Ambiente de instância única

Um ambiente de instância única contém uma instância do Amazon EC2 com um endereço IP elástico. Um ambiente de instância única não tem um load balancer, o que pode ajudar a reduzir custos em comparação com um ambiente escalável e com balanceamento de carga. Embora o ambiente de instância única não use o serviço Auto Scaling do Amazon EC2, todas as configurações



dos números mínimo e máximo de instâncias e da capacidade desejada são definidas como 1. Dessa forma, as novas instâncias não são iniciadas para acomodar a crescente carga no aplicativo.

Use um ambiente de instância única se você espera que o aplicativo de produção tenha pouco tráfego ou se você está fazendo um desenvolvimento remoto. Se você não tiver certeza sobre qual tipo de ambiente selecionar, poderá escolher um e, se necessário, alterá-lo posteriormente. Para obter mais informações, consulte [Alterar tipo de ambiente](#).

## Alterar tipo de ambiente

É possível alterar o tipo de ambiente para um de instância única ou escalável e com balanceamento de carga editando a configuração do ambiente. Em alguns casos, convém alterar de um tipo de ambiente para outro. Por exemplo, suponha que você desenvolveu e testou um aplicativo em um ambiente de instância única para reduzir custos. Quando a aplicação estiver pronta para produção, você poderá alterar o tipo de ambiente para um escalável e com balanceamento de carga, de modo que ele possa ser dimensionado para atender às demandas dos clientes.

Para alterar um tipo de ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria Capacity (Capacidade), escolha Edit (Editar).
5. Na lista Environment Type (Tipo de ambiente), selecione o tipo de ambiente desejado.

Elastic Beanstalk > Environments > GettingStartedApp-env > Configuration

## Modify capacity

Configure the compute capacity of your environment and Auto Scaling settings to optimize the number of instances used.

### Auto Scaling Group

Environment type  
Load balanced

Instances  
Min 1  
Max 2

Fleet composition  
Choose a mix of On-Demand and Spot Instances with multiple instance types. Spot Instances are automatically launched at the lowest available price. [Learn more](#)

On-Demand instances  
 Combine purchase options and instances

Maximum spot price  
The maximum price per instance-hour, in USD, that you're willing to pay for a Spot Instance. Setting a custom price limits your chances to fulfill your target capacity using [Spot Instances](#).

## 6. Escolha Save (Salvar).

Pode levar algum tempo para o ambiente ser atualizado enquanto o Elastic Beanstalk provisiona os recursos da AWS.

Se o ambiente estiver em uma VPC, selecione sub-redes para colocar as instâncias do Elastic Load Balancing e do Amazon EC2. Cada zona de disponibilidade na qual seu aplicativo é executado deve ter as duas. Para mais detalhes, consulte [Usar o Elastic Beanstalk com Amazon VPC](#).

## Ambientes de operador do Elastic Beanstalk

Se a sua aplicação do AWS Elastic Beanstalk executa operações ou fluxos de trabalho que demoram muito para serem concluídos, é possível descarregar essas tarefas em um ambiente de operador dedicado. O desacoplamento do front-end do aplicativo web de um processo que executa operações de bloqueio é uma maneira comum de garantir que seu aplicativo mantenha a capacidade de resposta sob carga.

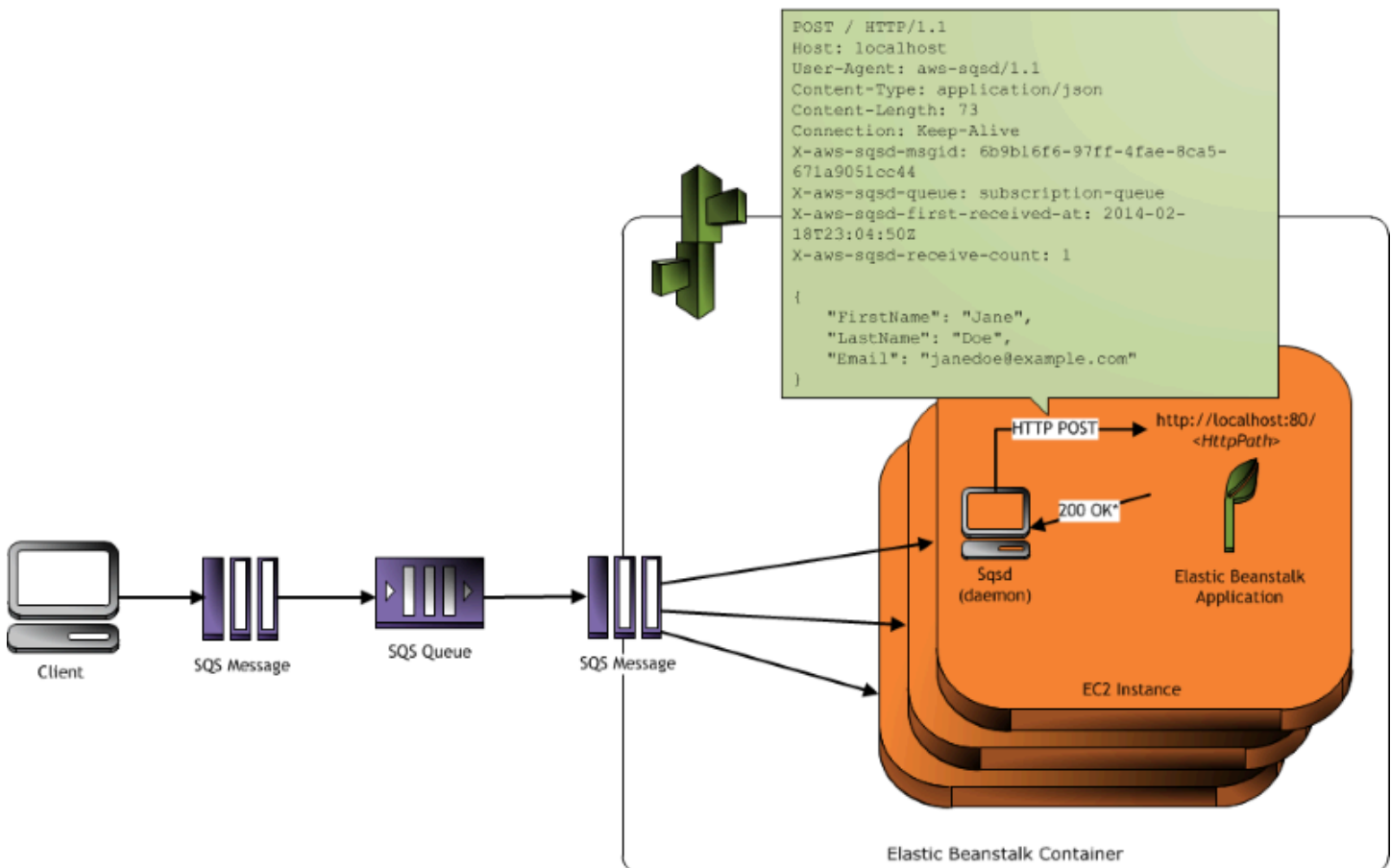
Uma tarefa de longa execução é alguma coisa que aumenta significativamente o tempo necessário para concluir uma solicitação, como o processamento de imagens ou vídeos, o envio de e-mail ou a geração de um arquivo ZIP. Essas operações podem levar apenas um ou dois segundos para serem

concluídas, mas um atraso de alguns segundos é muito para uma solicitação web que, de outra forma, seria concluída em menos de 500 ms.

Uma opção é criar um processo de operador localmente, retornar o êxito e processar a tarefa de maneira assíncrona. Isso funcionará, se a instância puder acompanhar todas as tarefas que são enviadas a ela. Sob carga elevada, no entanto, uma instância pode ficar sobrecarregada com tarefas em segundo plano e parar de responder às solicitações de prioridade mais alta. Se cada usuário puder gerar várias tarefas, o aumento na carga pode não corresponder a um aumento nos usuários, o que dificulta expandir o nível do servidor Web de modo eficaz.

Para evitar a execução de tarefas de longa execução localmente, é possível usar AWS SDK como linguagem de programação para enviá-las a uma fila do Amazon Simple Queue Service (Amazon SQS) e executar o processo que as executa em um conjunto separado de instâncias. Depois, você cria essas instâncias de operador para obter itens da fila apenas quando elas têm capacidade para executá-los, impedindo que se tornem sobrecarregadas.

Os ambientes de operador do Elastic Beanstalk simplificam esse processo gerenciando a fila do Amazon SQS e executando um [processo de daemon](#) em cada instância que lê a fila para você. Quando o daemon extrai um item da fila, envia uma solicitação HTTP POST localmente para `http://localhost/`, na porta 80, com o conteúdo da mensagem da fila no corpo. Tudo o que seu aplicativo precisa fazer é realizar a tarefa de longa execução em resposta ao POST. Você pode [configurar o daemon](#) para publicar em um caminho diferente, usar um tipo MIME que não seja aplicativo/JSON, conectar-se a uma fila existente ou personalizar conexões (máximo de solicitações simultâneas), tempos limite e novas tentativas.



\* HTTP Response of 200 OK = delete the message  
 Any other HTTP Response = retry the message after the VisibilityTimeout period  
 No response = retry the message after the InactivityTimeout period

Com as [tarefas periódicas](#), você também pode configurar o daemon do operador para colocar as mensagens na fila com base em uma programação cron. Cada tarefa periódica pode executar POST em um caminho diferente. Para habilitar as tarefas periódicas, inclua um arquivo YAML em seu código-fonte que define a programação e o caminho para cada tarefa.

### Note

O [.NET na Plataforma de Windows Server](#) não tem suporte para ambientes de operador.

## Seções

- [Daemon SQS do ambiente de operador](#)
- [Filas de mensagens mortas](#)
- [Tarefas periódicas](#)

- [Usar o Amazon CloudWatch para escalabilidade automática em níveis do ambiente de operador](#)
- [Configurar ambientes de operador](#)

## Daemon SQS do ambiente de operador

Os ambientes de operador executam um processo de daemon fornecido pelo Elastic Beanstalk. Esse daemon é atualizado regularmente para adicionar recursos e corrigir erros. Para obter a versão mais recente do daemon, atualize para a última [versão da Plataforma](#).

Quando a aplicação no ambiente de operador retorna uma resposta 200 OK para confirmar que ele recebeu e processou a solicitação com êxito, o daemon envia uma chamada DeleteMessage à fila do Amazon SQS para excluir a mensagem da fila. Se a aplicação retornar qualquer resposta diferente de 200 OK, o Elastic Beanstalk aguardará para recolocar a mensagem na fila após o período ErrorVisibilityTimeout configurado. Se não houver nenhuma resposta, o Elastic Beanstalk aguardará para recolocar a mensagem na fila após o período InactivityTimeout para que a mensagem seja disponibilizada para outra tentativa de processamento.

### Note

As propriedades das filas do Amazon SQS (ordem das mensagens, entrega pelo menos uma vez e amostragem de mensagens) podem afetar o modo como você cria uma aplicação Web para um ambiente de operador. Para obter mais informações, consulte [Propriedades das filas distribuídas](#) no [Guia do desenvolvedor do Amazon Simple Queue Service](#).

O Amazon SQS exclui automaticamente as mensagens que estão na fila há mais tempo do que o RetentionPeriod configurado.

O daemon define os seguintes cabeçalhos HTTP.

### Cabeçalhos HTTP

Name (Nome)	Value (Valor)
User-Agent	aws-sqsd
	aws-sqsd/1.1 1

## Cabeçalhos HTTP

<code>X-Aws-Sqs-Message-Id</code>	ID de mensagens do SQS, usado para detectar enxurradas de mensagens (um número excepcionalmente alto de novas mensagens).
<code>X-Aws-Sqs-Queue</code>	Nome da fila do SQS.
<code>X-Aws-Sqs-First-Received-At</code>	Horário UTC, no <a href="#">formato ISO 8601</a> , quando a mensagem foi recebida pela primeira vez.
<code>X-Aws-Sqs-Receive-Count</code>	Contagem de recebimento de mensagens do SQS.
<code>X-Aws-Sqs-Attr-<i>message-attribute-name</i></code>	Atributos de mensagem personalizados concedidos à mensagem que está sendo processada. O <code>message-attribute-name</code> é o nome do atributo de mensagem real. Todos os atributos de mensagem de string e número são adicionados ao cabeçalho. Os atributos binários são descartados e não são incluídos no cabeçalho.
<code>Content-Type</code>	Configuração de tipo Mime. Por padrão, <code>application/json</code> .

## Filas de mensagens mortas

Os ambientes de operador do Elastic Beanstalk são compatíveis com filas de mensagens mortas do Amazon Simple Queue Service (Amazon SQS). Dead letter queue (fila de mensagens mortas) é uma fila onde outras filas (origem) podem enviar mensagens que, por algum motivo, não puderam ser processadas com êxito. Um benefício importante de usar filas de mensagens mortas é a capacidade de segregar e isolar as mensagens que não foram processadas com êxito. Em seguida, você pode analisar todas as mensagens enviadas para a fila de mensagens mortas para tentar determinar por que elas não foram processadas com êxito.

Se você especificar uma fila do Amazon SQS gerada automaticamente no momento em que criar o nível de ambiente de operador, uma fila de mensagens mortas será habilitada por padrão para um

ambiente de operador. Se você escolher uma fila do SQS existente para seu ambiente de operador, deve usar o SQS para configurar uma fila de mensagens mortas de forma independente. Para obter informações sobre como usar o SQS para configurar uma fila de mensagens mortas, consulte [Usar fila de mensagens mortas do Amazon SQS](#).

Você não pode desabilitar filas de mensagens mortas. As mensagens que não puderem ser entregues serão eventualmente enviadas a uma fila de mensagens mortas. No entanto, você pode efetivamente desabilitar esse recurso definindo a opção `MaxRetries` como o valor máximo válido de 100.

Se uma fila de mensagens mortas não estiver configurada para a fila do Amazon SQS do ambiente de operador, o Amazon SQS manterá as mensagens na fila até que o período de retenção expire. Para obter detalhes sobre como configurar o período de retenção, consulte [the section called “Configurar ambientes de operador”](#).

#### Note

A opção `MaxRetries` do Elastic Beanstalk é equivalente à opção `MaxReceiveCount` do SQS. Se o ambiente de operador não usar uma fila do SQS gerada automaticamente, use a opção `MaxReceiveCount` no SQS para desabilitar efetivamente a fila de mensagens mortas. Para obter mais informações, consulte [Usar filas de mensagens mortas do Amazon SQS](#).

Para obter mais informações sobre o ciclo de vida de uma mensagem do SQS, acesse [Ciclo de vida de mensagens](#).

## Tarefas periódicas

Você pode definir tarefas periódicas em um arquivo denominado `cron.yaml` em seu pacote de origem para adicionar trabalhos à fila do seu ambiente de operador automaticamente em intervalos regulares.

Por exemplo, o arquivo `cron.yaml` a seguir cria duas tarefas periódicas. O primeiro é executado a cada 12 horas e o segundo é executado às 11 horas UTC todos os dias.

Example `cron.yaml`

```
version: 1
```

```
cron:
- name: "backup-job"
 url: "/backup"
 schedule: "0 */12 * * *"
- name: "audit"
 url: "/audit"
 schedule: "0 23 * * *"
```

O **name** deve ser exclusivo para cada tarefa. O URL é o caminho para o qual a solicitação POST é enviada para acionar o trabalho. A programação é uma [expressão CRON](#) que determina quando a tarefa é executada.

Quando uma tarefa é executada, o daemon publica uma mensagem na fila do SQS do ambiente com um cabeçalho indicando o trabalho que precisa ser realizado. Qualquer instância no ambiente pode selecionar a mensagem e processar o trabalho.

#### Note

Se você configurar o ambiente de operador com uma fila do SQS existente e escolher uma [fila FIFO do Amazon SQS](#), as tarefas periódicas não serão compatíveis.

O Elastic Beanstalk usa a eleição de líder para determinar qual instância no ambiente de operador coloca a tarefa periódica em fila. Cada instância tenta se tornar líder gravando em uma tabela do Amazon DynamoDB. A primeira instância que for bem-sucedida será líder e deverá continuar gravando na tabela para manter o status de líder. Se o líder ficar fora de serviço, outra instância rapidamente assumirá seu lugar.

Para tarefas periódicas, o daemon do operador define os seguintes cabeçalhos adicionais.

#### Cabeçalhos HTTP

Name (Nome)	Value (Valor)
X-Aws-Sqsd-Taskname	Para tarefas periódicas, o nome da tarefa a ser executada.
X-Aws-Sqsd-Scheduled-At	Horário programado para a tarefa periódica



## Cabeçalhos HTTP

X-Aws-Sqs-Id

Número da conta da AWS do remetente da mensagem

## Usar o Amazon CloudWatch para escalabilidade automática em níveis do ambiente de operador

Juntos, o Amazon EC2 Auto Scaling e o CloudWatch monitoram a utilização da CPU das instâncias em execução no ambiente de operador. A maneira como você configura o limite de escalação automática para a capacidade da CPU determina quantas instâncias o grupo de Auto Scaling executa para gerenciar adequadamente o throughput de mensagens na fila do Amazon SQS. Cada instância do EC2 publica suas métricas de utilização da CPU no CloudWatch. O Amazon EC2 Auto Scaling recupera do CloudWatch o uso médio da CPU em todas as instâncias no ambiente de operador. Você configura o limite superior e inferior e quantas instâncias devem ser adicionadas ou encerradas de acordo com a capacidade da CPU. Quando o Amazon EC2 Auto Scaling detecta que você atingiu o limite superior especificado na capacidade da CPU, o Elastic Beanstalk cria novas instâncias no ambiente de operador. As instâncias são excluídas quando a carga da CPU volta a ficar abaixo do limite.

### Note

As mensagens que não foram processadas no momento em que uma instância foi encerrada são retornadas para a fila na qual elas podem ser processadas por outro daemon em uma instância que ainda está em execução.

Você também pode definir outros alarmes do CloudWatch, conforme necessário, usando o console do Elastic Beanstalk, a CLI ou o arquivo de opções. Para obter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon CloudWatch](#) e [Criar um grupo de Auto Scaling com políticas de escalabilidade em etapas](#).

## Configurar ambientes de operador

É possível gerenciar a configuração de um ambiente de operador editando a configuração Worker Configuration (Configuração do operador) na página Configuration (Configuração) no [console de gerenciamento do ambiente](#).

[Elastic Beanstalk](#) > [Environments](#) > [GettingStartedApp-env](#) > Configuration

## Modify worker

You can create a new Amazon SQS queue for your worker application or pull work items from an existing queue. The worker daemon on the instances in your environment pulls an item from the queue and relays it in the body of a POST request to a local HTTP path relative to localhost.

### Queue

Worker queue



SQS queue from which to read work items.

### Messages

HTTP path

The daemon pulls items from the Amazon SQS queue and posts them locally to this path.

MIME type

Change the MIME type of the POST requests that the worker daemon sends to your application.

HTTP connections

Maximum number of concurrent connections to the application.

Visibility timeout

seconds

The amount of time to lock an incoming message for processing before returning it to the queue.

Error visibility timeout

seconds

The amount of time to wait before resending a message after an error response from the application.

### ▼ Advanced options

The following settings control advanced behavior of the worker tier daemon. [Learn more](#)

Max retries

Maximum number of retries after which the message is discarded.

Connection timeout

Inactivity timeout

**Note**

Você pode configurar o caminho do URL para publicar mensagens na fila de operadores, mas não pode configurar a porta IP. O Elastic Beanstalk sempre publica mensagens da fila do operador na porta 80. O aplicativo do ambiente de operador ou seu proxy deve ouvir na porta 80.

Para configurar o daemon do operador

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

**Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Worker (Operador), escolha Edit (Editar).

A página de configuração Modify worker (Modificar operador) tem as seguintes opções.

Na seção Queue (Fila):

- Worker queue (Fila do operador): especifique a fila do Amazon SQS de leitura do daemon. Se tiver uma, você poderá escolher uma fila existente. Se você escolher Autogenerated queue (Fila gerada automaticamente), o Elastic Beanstalk criará uma nova fila do Amazon SQS e um Worker queue URL (URL da fila do operador) correspondente.

**Note**

Quando você escolhe Autogenerated queue (Fila gerada automaticamente), a fila criada pelo Elastic Beanstalk é uma fila [padrão](#) do Amazon SQS. Ao escolher uma fila existente, você pode fornecer uma fila padrão ou uma fila [FIFO](#) do Amazon SQS. Lembre-se de que, se você fornecer uma fila FIFO, as [tarefas periódicas](#) não serão compatíveis.

- **Worker queue URL** ((URL da fila do operador)): se você escolher uma **Worker queue** (Fila do operador) existente, essa configuração exibirá o URL associado a essa fila do Amazon SQS.

Na seção **Messages** (Mensagens):

- **HTTP path** (Caminho HTTP): especifique o caminho relativo para a aplicação que receberá os dados da fila do Amazon SQS. Os dados são inseridos no corpo de uma mensagem HTTP POST. O valor padrão é `/`.
- **MIME type** (Tipo MIME): indica o tipo MIME usado pela mensagem HTTP POST. O valor padrão é `application/json`. No entanto, qualquer valor é válido, pois você pode criar e, em seguida, especificar seu próprio tipo MIME.
- **HTTP connections** (Conexões HTTP): especifique o número máximo de conexões simultâneas que o daemon pode estabelecer com qualquer aplicação em uma instância do Amazon EC2. O padrão é **50**. Você pode especificar de **1** a **100**.
- **Visibility timeout** (Tempo limite de visibilidade): indique por quanto tempo (em segundos) uma mensagem recebida da fila do Amazon SQS fica bloqueada para processamento. Após o término do tempo configurado, a mensagem ficará novamente visível na fila para outro daemon ler. Escolha um valor superior ao que seu aplicativo precisa para processar mensagens, até **43200** segundos.
- **Error visibility timeout** (Tempo limite de visibilidade de erros): indique quanto tempo (em segundos) o Elastic Beanstalk deve aguardar para retornar uma mensagem à fila do Amazon SQS após uma falha com erro explícito ao tentar processá-la. Você pode especificar de **0** a **43200** segundos.

Na seção **Advanced options**:

- **Max retries** (Máximo de tentativas): especifique o número máximo de vezes que o Elastic Beanstalk tenta enviar a mensagem à fila do Amazon SQS antes de movê-la para a [fila de mensagens mortas](#). O valor padrão é **10**. Você pode especificar de **1** a **100**.

#### Note

A opção **Max retries** (Máximo de novas tentativas) só é aplicável a filas do Amazon SQS configuradas com uma fila de mensagens mortas. Para qualquer fila do Amazon SQS que não esteja configurada com uma fila de mensagens mortas, o Amazon SQS retém

as mensagens na fila e as processa até a expiração do período especificado pela opção `Retention period` (Período de retenção).

- `Connection timeout` (Tempo limite de conexão): indique por quanto tempo (em segundos) aguardar por conexões bem-sucedidas com uma aplicação. O valor padrão é **5**. Você pode especificar de **1** a **60** segundos.
- `Inactivity timeout` (Tempo limite de inatividade): indique por quanto tempo (em segundos) aguardar por uma resposta para uma conexão existente com uma aplicação. O valor padrão é **180**. Você pode especificar de **1** a **36000** segundos.
- `Retention period` (Período de retenção): indique por quanto tempo (em segundos) uma mensagem é válida e processada ativamente. O valor padrão é **345600**. Você pode especificar de **60** a **1209600** segundos.

Se você usa uma fila do Amazon SQS existente, as configurações que você define ao criar um ambiente de operador podem entrar em conflito com as configurações definidas diretamente no Amazon SQS. Por exemplo, se você configurar um ambiente de operador com um valor de `RetentionPeriod` maior que o valor de `MessageRetentionPeriod` definido no Amazon SQS, este excluirá a mensagem quando ela exceder o `MessageRetentionPeriod`.

Por outro lado, se o valor `RetentionPeriod` definido na configuração do ambiente de operador for menor do que o valor de `MessageRetentionPeriod` definido no Amazon SQS, o daemon excluirá a mensagem antes de o Amazon SQS poder excluí-la. Em `VisibilityTimeout`, o valor definido para o daemon nas configurações do ambiente de operador substitui a configuração `VisibilityTimeout` do Amazon SQS. Certifique-se de que as mensagens sejam excluídas adequadamente comparando suas configurações do Elastic Beanstalk às configurações do Amazon SQS.

## Criar links entre ambientes do Elastic Beanstalk

À medida que seu aplicativo aumenta de tamanho e complexidade, você pode dividi-lo em componentes com diferentes ciclos de vida operacionais e de desenvolvimento. Ao executar serviços menores que interagem com os outros por uma interface bem definida, as equipes podem trabalhar de forma independente, e implantações podem ser de menor risco. O AWS Elastic Beanstalk permite vincular seus ambientes para compartilhar informações entre componentes que dependem um do outro.

**Note**

Atualmente, o Elastic Beanstalk é compatível com links de ambiente para todas as plataformas, exceto Docker de vários contêineres.

Com os links de ambiente, você pode especificar as conexões entre os ambientes de componentes do aplicativo como referências nomeadas. Quando um ambiente que define um link é criado, o Elastic Beanstalk define uma variável de ambiente com o mesmo nome do link. O valor da variável é o endpoint que você pode usar para se conectar a outro componente, que pode ser um servidor Web ou ambiente de operador.

Por exemplo, se o aplicativo é composto de um front-end que coleta endereços de e-mail e um operador que envia e-mail de boas-vindas para os endereços coletados pelo front-end, você pode criar um link para o operador do front-end e o front-end descobrir automaticamente o endpoint (URL da fila) para o operador.

Defina links para outros ambientes em um [manifesto de ambiente](#), um arquivo em formato YAML nomeado `env.yaml` na raiz da origem do aplicativo. O seguinte manifesto define um link para um ambiente denominado operador:

**~/workspace/my-app/frontend/env.yaml**

```
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentLinks:
 "WORKERQUEUE": "worker"
```

Quando você cria um ambiente com uma versão da aplicação que inclui o manifesto de ambiente acima, o Elastic Beanstalk procura um ambiente chamado `worker` que pertence à mesma aplicação. Se esse ambiente existir, o Elastic Beanstalk criará uma propriedade de ambiente chamada `WORKERQUEUE`. O valor de `WORKERQUEUE` é o URL da fila do Amazon SQS. O aplicativo de front-end pode ler essa propriedade da mesma maneira que uma variável de ambiente. Para obter mais detalhes, consulte [Manifesto de ambiente \(env.yaml\)](#).

Para usar os links de ambiente, adicione o manifesto de ambiente à fonte da aplicação e faça upload dele com a EB CLI, AWS CLI ou um SDK. Se você usar AWS CLI ou um SDK, defina o sinalizador `process` quando chamar `CreateApplicationVersion`:

```
$ aws elasticbeanstalk create-application-version --process --application-name
my-app --version-label frontend-v1 --source-bundle S3Bucket="DOC-EXAMPLE-
BUCKET",S3Key="front-v1.zip"
```

Essa opção instrui o Elastic Beanstalk a validar o manifesto de ambiente e os arquivos de configuração no pacote de origem quando você cria a versão da aplicação. A EB CLI define esse sinalizador automaticamente quando você tem um manifesto de ambiente no diretório do projeto.

Crie seus ambientes normalmente usando qualquer cliente. Quando for necessário encerrar ambientes, primeiro faça isso com o vínculo. Se um ambiente estiver vinculado a qualquer outro ambiente, o Elastic Beanstalk impedirá o encerramento do ambiente vinculado. Para substituir essa proteção, use o sinalizador `ForceTerminate`. Esse parâmetro está disponível na AWS CLI como `--force-terminate`:

```
$ aws elasticbeanstalk terminate-environment --force-terminate --environment-name
worker
```

# Configurar ambientes do Elastic Beanstalk

AWS Elastic Beanstalk fornece uma ampla variedade de opções para personalizar os recursos em seu ambiente e o comportamento e as configurações da plataforma do Elastic Beanstalk. Quando você cria um ambiente de servidor Web, o Elastic Beanstalk cria vários recursos para oferecer suporte à operação da aplicação.

- **Instância do EC2:** uma máquina virtual do Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicações Web na plataforma escolhida.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- **Grupo de segurança de instância:** um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do load balancer chegue à instância do EC2 que executa seu aplicativo Web. Por padrão, o tráfego não é permitido em outras portas.
- **Balanceador de carga:** um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- **Grupo de segurança do balanceador de carga:** um grupo de segurança do Amazon EC2 configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- **Grupo de Auto Scaling:** grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- **Bucket do Amazon S3:** um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- **CloudWatch Alarmes da Amazon** — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- **AWS CloudFormation stack** — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no [console do AWS CloudFormation](#).



- Nome de domínio: um nome de domínio que encaminha para a aplicação Web no formato *subdomínio.região.elasticbeanstalk.com*.

#### Note

Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio `elasticbeanstalk.com` é registrado na [Lista Pública de Sufixos \(PSL\)](#). Para maior segurança, recomendamos que você use cookies com um prefixo `__Host-` se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações do Elastic Beanstalk. Essa prática ajudará a defender seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página [Set-Cookie](#) na Mozilla Developer Network.

Este tópico se concentra nas opções de configuração de recursos disponíveis no console do Elastic Beanstalk. Os tópicos a seguir mostram como configurar seu ambiente no console. Eles também descrevem os namespaces subjacentes que correspondem às opções do console a serem usadas com arquivos de configuração ou com as opções de configuração da API. Para saber mais sobre métodos de configuração avançados, consulte [Configurar ambientes \(avançado\)](#).

#### Tópicos

- [Configuração do ambiente usando o console do Elastic Beanstalk](#)
- [Instâncias do Amazon EC2 para seu ambiente do Elastic Beanstalk](#)
- [Grupo do Auto Scaling para o seu ambiente do Elastic Beanstalk](#)
- [Balanceador de carga do ambiente do Elastic Beanstalk](#)
- [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#)
- [Segurança do ambiente AWS Elastic Beanstalk](#)
- [Marcar recursos em seus ambientes do Elastic Beanstalk](#)
- [Propriedades de ambientes e outras configurações de software](#)
- [Notificações do ambiente do Elastic Beanstalk com o Amazon SNS](#)
- [Configurar a Amazon Virtual Private Cloud \(Amazon VPC\) com o Elastic Beanstalk](#)
- [Nome de domínio do ambiente do Elastic Beanstalk](#)

# Configuração do ambiente usando o console do Elastic Beanstalk

É possível usar o console do Elastic Beanstalk para visualizar e modificar muitas [opções de configuração](#) do seu ambiente e seus recursos. Você pode personalizar o comportamento do ambiente durante as implantações, habilitar recursos adicionais e modificar o tipo de instância e outras configurações que você definiu durante a criação do ambiente.

Como visualizar um resumo da configuração de seu ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

## Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).

## Página de configuração

A página Configuration overview (Visão geral de configuração) mostra um conjunto de categorias de configuração. Cada categoria de configuração resume o estado atual de um conjunto de opções relacionadas.

Elastic Beanstalk > Environments > Gettingstarteda-env > Configuration

## Configuration Info

[Cancel](#) [Review changes](#) [Apply changes](#)

### Service access Info

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances. [Edit](#)

Service role  
arn:aws:iam::164656829171:role/aws-elasticbeanstalk-service-role

### Instance traffic and scaling Info

Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic. Configure the software that runs on your environment's instances by setting platform-specific options. [Edit](#)

**Instances**  
IMDSv1  
Deactivated

**Capacity**

Environment type	Fleet composition	On-demand base
Load balanced	On-Demand instances	0
On-demand above base	Processor type	Instance types
70	x86_64	t2.micro,t2.small

**Load balancer**  
Load balancer type  
application

### Networking, database, and tags Info

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment. [Edit](#)

**Network**

Load balancer visibility	Load balancer subnets
public	—

**Database**  
Has coupled database  
false

### Updates, monitoring, and logging Info

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources. [Edit](#)

**Updates**

Managed updates	Update batch size	Deployment batch size
Deactivated	1	100
Deployment batch size type	Command timeout	Deployment policy
Percentage	600	AllAtOnce
Health threshold	Ignore health check	Instance replacement
Ok	false	false
Minimum capacity	Notifications email	
0	—	

Escolha Edit (Editar) em uma categoria de configuração para acessar uma página de configurações relacionadas, na qual é possível ver os valores de opção completos e fazer alterações. Quando terminar de visualizar e modificar as opções, você poderá escolher uma das seguintes ações:

- **Cancel (Cancelar):** volte ao painel do ambiente sem aplicar suas alterações de configuração. Ao escolher Cancel (Cancelar), o console perde as alterações pendentes feitas em qualquer categoria de configuração.

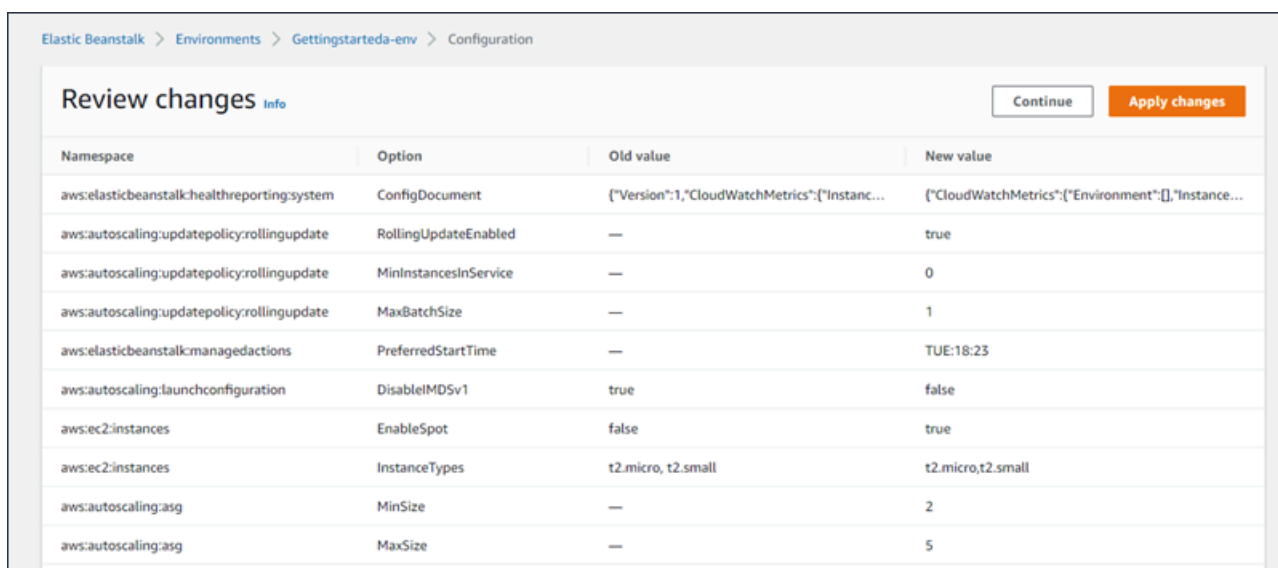
Você também pode cancelar as alterações de configuração, escolhendo outra página do console, como Events (Eventos) ou Logs. Neste caso, se houver alterações de configuração pendente, o console solicitará que você confirme que aceita perdê-las.

- **Review changes (Revisar alterações):** obtenha um resumo de todas as alterações pendentes feitas em qualquer uma das categorias de configuração. Para obter mais detalhes, consulte [Página Revisar alterações](#).
- **Apply changes (Aplicar alterações):** aplique no seu ambiente todas as alterações feitas nas categorias de configuração. Em alguns casos, será solicitado que você confirme uma consequência de uma das suas decisões de configuração.

## Página Revisar alterações

A página Review Changes (Revisar alterações) exibe uma tabela que mostra todas as alterações pendentes de opções feitas em qualquer uma das categorias de configuração e que não foram aplicadas ainda ao seu ambiente.

As tabelas listam cada opção como uma combinação de Namespace e Option (Opção) com a qual o Elastic Beanstalk a identifica. Para obter mais detalhes, consulte [Opções de configuração](#).



Namespace	Option	Old value	New value
aws:elasticbeanstalk:healthreporting:system	ConfigDocument	{"Version":1,"CloudWatchMetrics":{"Instanc...	{"CloudWatchMetrics":{"Environment":"","Instanc...
aws:autoscaling:updatepolicy:rollingupdate	RollingUpdateEnabled	—	true
aws:autoscaling:updatepolicy:rollingupdate	MinInstancesInService	—	0
aws:autoscaling:updatepolicy:rollingupdate	MaxBatchSize	—	1
aws:elasticbeanstalk:managedactions	PreferredStartTime	—	TUE:18:23
aws:autoscaling:launchconfiguration	DisableIMDSv1	true	false
aws:ec2:instances	EnableSpot	false	true
aws:ec2:instances	InstanceTypes	t2.micro, t2.small	t2.micro,t2.small
aws:autoscaling:asg	MinSize	—	2
aws:autoscaling:asg	MaxSize	—	5

Quando terminar de revisar suas alterações, você poderá escolher uma das seguintes ações:

- Continue (Continuar): voltar para a página Configuration Overview (Visão geral da configuração). Em seguida, você pode continuar a fazer alterações ou aplicar as pendentes.
- Apply changes (Aplicar alterações): aplique no seu ambiente todas as alterações feitas nas categorias de configuração. Em alguns casos, será solicitado que você confirme uma consequência de uma das suas decisões de configuração.

## Instâncias do Amazon EC2 para seu ambiente do Elastic Beanstalk

Quando você cria um ambiente de servidor Web, o AWS Elastic Beanstalk cria uma ou mais máquinas virtuais do Amazon Elastic Compute Cloud (Amazon EC2), conhecidas como instâncias.

As instâncias em seu ambiente são configuradas para executar aplicativos web na plataforma de sua preferência. Você pode fazer alterações em várias propriedades e comportamentos das instâncias do seu ambiente ao criar o ambiente ou depois que ele já estiver em execução. Ou, você já pode fazer essas alterações modificando o código-fonte que implanta no ambiente. Para obter mais informações, consulte [the section called “Opções de configuração”](#).

### Note

O [grupo de Auto Scaling](#) no ambiente gerencia as instâncias do Amazon EC2 que executam a aplicação. Quando você faz alterações de configuração descritas nesta página, a configuração de execução também é alterada. A configuração de execução é um modelo de lançamento da Amazon EC2 ou um recurso de configuração de execução do grupo do Auto Scaling. Essa alteração requer a [substituição de todas as instâncias](#). Ele também aciona uma [atualização contínua](#) ou [atualização imutável](#), dependendo de qual delas está configurada.

O Elastic Beanstalk é compatível com várias [opções de compra de instâncias](#): Instâncias sob demanda, Instâncias reservadas e Instâncias spot do Amazon EC2. Uma instância sob demanda é um pay-as-you-go recurso. Não é necessário um compromisso de longo prazo quando você a usa. Uma instância reservada é um desconto de faturamento pré-compra, aplicado automaticamente às instâncias sob demanda correspondentes em seu ambiente. Uma instância spot é uma instância do Amazon EC2 não utilizada que está disponível por um valor mais baixo que o preço sob demanda. Você pode habilitar instâncias spot em seu ambiente definindo uma única opção. Você pode configurar o uso de instância spot, incluindo a combinação de instâncias sob demanda e spot, usando opções adicionais. Para obter mais informações, consulte [Grupo de Auto Scaling](#).

## Seções

- [Tipos de instância do Amazon EC2](#)
- [Configurar instâncias do Amazon EC2 para seu ambiente](#)
- [Configurar instâncias do EC2 da AWS para seu ambiente usando a AWS CLI](#)
- [Recomendações para ambientes de primeiro grupo Graviton arm64](#)
- [O namespace `aws:autoscaling:launchconfiguration`](#)
- [Configurar o serviço de metadados da instância nas instâncias do ambiente](#)

## Tipos de instância do Amazon EC2

Quando você cria um novo ambiente, o Elastic Beanstalk provisiona instâncias do Amazon EC2 baseadas nos tipos de instância do Amazon EC2 que você escolher. O tipo de instância escolhido determina o hardware do host que executa suas instâncias. Os tipos de instância do EC2 podem ser categorizados por qual arquitetura de processador cada um se baseia. O Elastic Beanstalk é compatível com tipos de instância com base na seguinte arquitetura de processador: Arquitetura Arm Graviton da AWS de 64 bits (arm64), arquitetura de 64 bits (x86) e arquitetura de 32 bits (i386). O Elastic Beanstalk seleciona a arquitetura do processador x86 por padrão quando você cria um novo ambiente.

### Note

A arquitetura i386 de 32 bits não é mais compatível com a maioria das plataformas do Elastic Beanstalk. Então recomendamos que você escolha os tipos de arquitetura x86 ou arm64. O Elastic Beanstalk fornece [Opções de configuração](#) para o tipo de instância do processador i386 no namespace da [`aws:ec2:instances`](#).

Todos os tipos de instância na configuração de um determinado ambiente do Elastic Beanstalk devem ter o mesmo tipo de arquitetura de processador. Suponha que você adicione um novo tipo de instância a um ambiente existente que já tenha um tipo de instância `t2.medium`, que é baseado na arquitetura x86. Você só pode adicionar outro tipo de instância da mesma arquitetura, como `t2.small`. Se você quiser substituir os tipos de instância existentes por aqueles de uma arquitetura diferente, você pode fazer isso. Mas certifique-se de que todos os tipos de instância no comando sejam baseados no mesmo tipo de arquitetura.

O Elastic Beanstalk adiciona regularmente suporte para novos tipos de instância compatíveis depois que são apresentados pelo Amazon EC2. Para obter mais informações sobre o tipo de instância disponível, consulte [tipo de instância](#), no Guia do usuário do Amazon EC2 para instâncias do Linux ou [tipos de instância](#) no Guia do usuário do Amazon EC2 para instâncias do Windows.

#### Note

O Elastic Beanstalk agora é compatível com Graviton em todas as plataformas mais recentes do Amazon Linux 2 em todas as regiões da AWS compatíveis com Graviton. Para obter mais informações sobre como criar um ambiente do Elastic Beanstalk com tipos de instâncias baseados em arm64, consulte [Configurar instâncias do Amazon EC2 para seu ambiente](#). Crie novos ambientes que executam instâncias do Amazon EC2 na arquitetura arm64 e migre suas aplicações para lá com as [opções de implantação](#) no Elastic Beanstalk. Para saber mais sobre os processadores baseados em Graviton arm64, consulte estes recursos da AWS:

- Benefícios – [O Processador Graviton da AWS](#)
- Introdução e outros tópicos, como considerações específicas do idioma — Artigo sobre como [começar](#) a usar o Graviton AWS GitHub

## Configurar instâncias do Amazon EC2 para seu ambiente

Você pode criar ou modificar a configuração da instância do Amazon EC2 do ambiente do Elastic Beanstalk no console do Elastic Beanstalk.

#### Note

Embora o console do Elastic Beanstalk não ofereça a opção de alterar a arquitetura do processador de um ambiente existente, você pode fazer isso com o AWS CLI. Por exemplo, com os comandos, consulte [Configurar instâncias do EC2 da AWS para seu ambiente usando a AWS CLI](#).

Como configurar instâncias do Amazon EC2 no console do Elastic Beanstalk durante a criação de ambientes:

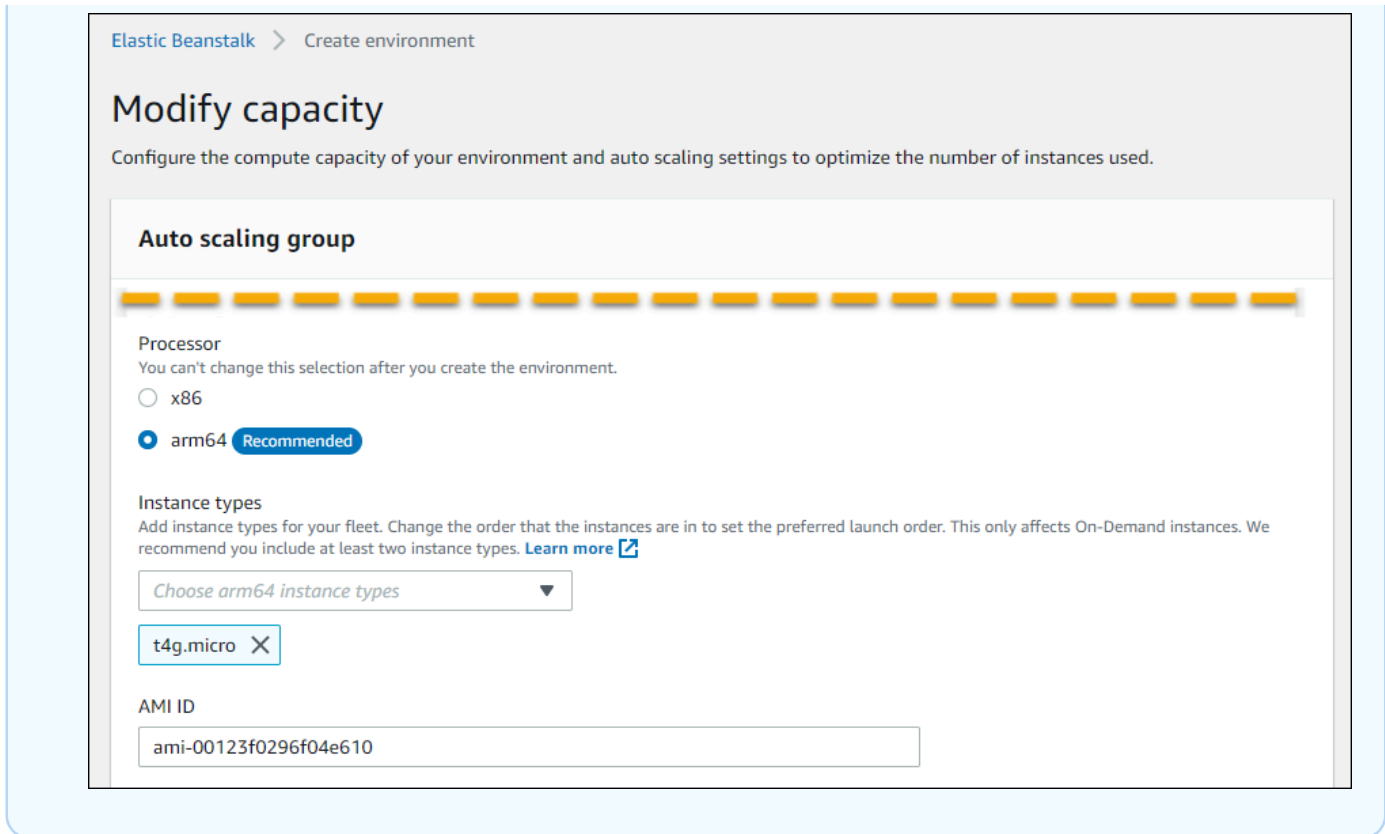
1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, escolha Ambientes.
3. Escolha [Create a new environment \(Criar um ambiente\)](#) para começar a criar seu ambiente.
4. Na página principal do assistente, antes de escolher Create environment, escolha Configure more options.
5. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar). Faça alterações nas configurações dessa categoria e escolha Apply (Aplicar). Para obter descrições de configuração, consulte a seção [the section called “Configurações de categoria de instâncias”](#) nesta página.
6. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar). Faça alterações nas configurações dessa categoria e escolha Continue (Continuar). Para obter descrições de configuração, consulte a seção [the section called “Configurações da categoria de capacidade”](#) nesta página.

 Selecione a arquitetura do processador.

Role para baixo até Processador para selecionar uma arquitetura de processador para suas instâncias do EC2. O console lista as arquiteturas de processador compatíveis com a plataforma que você escolheu anteriormente no painel Criar ambiente.

Se não encontrar a arquitetura do processador que você precisa, volte à lista de categorias de configuração da lista para selecionar uma plataforma compatível com ela. No painel Modificar capacidade, escolha Cancelar. Em seguida, escolha Alterar a versão da plataforma para escolher novas configurações de plataforma. Em seguida, na categoria de configuração Capacidade, escolha Editar para ver as opções de arquitetura de processador novamente.





7. Escolha Save e, em seguida, faça as outras alterações de configuração de acordo com seu ambiente.
8. Selecione Create environment (Criar ambiente).

Como configurar um ambiente de execução para instâncias do Amazon EC2 no console do Elastic Beanstalk:

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

#### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).

4. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar). Faça alterações nas configurações dessa categoria e escolha Apply (Aplicar). Para obter descrições de configuração, consulte a seção [the section called “Configurações de categoria de instâncias”](#) nesta página.
5. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar). Faça alterações nas configurações dessa categoria e escolha Continue (Continuar). Para obter descrições de configuração, consulte a seção [the section called “Configurações da categoria de capacidade”](#) nesta página.

## Configurações de categoria de instâncias

As configurações a seguir, relacionadas a instâncias do Amazon EC2, estão disponíveis na categoria de configuração Instances (Instâncias).

### Opções

- [Intervalo de monitoramento](#)
- [Volume raiz \(dispositivo de inicialização\)](#)
- [Serviço de metadados da instância](#)
- [Grupos de segurança](#)

## Modify instances

### Amazon CloudWatch monitoring

The time interval between when metrics are reported from the EC2 instances.

Monitoring interval

5 minute ▼

### Root volume (boot device)

Root volume type

(Container default) ▼

Size

The number of gigabytes of the root volume attached to each instance.

GB

IOPS

Input/output operations per second for a provisioned IOPS (SSD) volume.

100 IOPS

Throughput

The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance

MiB/s

### Instance metadata service (IMDS)

Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, disable IMDSv1. [Learn more](#)

Disable IMDSv1

With the current setting, the environment enables both IMDSv1 and IMDSv2.

Disabled

### EC2 security groups

	Group name	Group ID	Name
<input type="checkbox"/>	aws-ec2-elasticbeanstalk-ec2-1	sg-027aafe45182f171f	WinTest-dev
<input type="checkbox"/>	aws-ec2-elasticbeanstalk-ec2-2	sg-020e30e60b3e80c5b	WinTest-dev
<input type="checkbox"/>	aws-ec2-elasticbeanstalk-ec2-3	sg-03879e31c4ebe98ea	Gettingstarted-env
<input checked="" type="checkbox"/>	aws-ec2-elasticbeanstalk-ec2-4	sg-05b1982101cf211ef	Gettingstarted-env
<input type="checkbox"/>	default	sg-3527cd14	

Cancel

Continue

Apply

## Intervalo de monitoramento

Por padrão, as instâncias em seu ambiente publicam [métricas básicas de saúde](#) CloudWatch na Amazon em intervalos de cinco minutos, sem custo adicional.

Para obter relatórios mais detalhados, você pode definir o intervalo de monitoramento como 1 minuto para aumentar a frequência com que os recursos em seu ambiente publicam [métricas básicas de CloudWatch saúde](#). CloudWatch as taxas de serviço se aplicam a métricas de intervalo de um minuto. Para obter mais informações, consulte [Amazon CloudWatch](#).

## Volume raiz (dispositivo de inicialização)

Cada instância do ambiente é configurado com um volume raiz. O volume raiz é o dispositivo de blocos do Amazon EBS anexado à instância para armazenar o sistema operacional, as bibliotecas, os scripts e o código-fonte da aplicação. Por padrão, todas as plataformas usam dispositivos de blocos SSD de uso geral para armazenamento.

Você pode modificar o Root volume type (Tipo de volume raiz) para usar armazenamento magnético ou tipos de volume SSD de IOPS provisionadas e, se necessário, aumentar o tamanho do volume. Para os volumes de IOPS provisionadas, selecione também o número de IOPS a provisionar. O throughput só é aplicável a tipos de volume gp3 SSD. Você pode inserir o throughput desejado para provisionamento. Ela pode variar entre 125 a 1000 mebibytes por segundo (MiB/s). Selecione o tipo de volume que atenda aos seus requisitos de performance e preço.

Para obter mais informações, consulte [Tipos de volume do Amazon EBS](#) no Guia do usuário do Amazon EC2 para instâncias do Linux e [Detalhes do produto Amazon EBS](#).

## Serviço de metadados da instância

O serviço de metadados da instância (IMDS) é um componente na instância que o código na instância usa para acessar metadados da instância com segurança. O código pode acessar metadados de instância em uma instância em execução usando um dos seguintes métodos: Eles são o serviço de metadados da instância Versão 1 (IMDSv1) ou serviço de metadados da instância Versão 2 (IMDSv2). O IMDSv2 é mais seguro. Desabilite o IMDSv1 para aplicar o IMDSv2. Para ter mais informações, consulte [the section called "IMDS"](#).

### Note

A seção do IMDS nesta página de configuração aparecem apenas para as versões de plataforma que são compatíveis com IMDSv2.

## Grupos de segurança

Os grupos de segurança anexados a suas instâncias determinam qual tráfego tem permissão para chegar às instâncias. Eles também determinam qual tráfego tem permissão para sair das instâncias. O Elastic Beanstalk cria um grupo de segurança que permite tráfego do balanceador de carga nas portas padrão para HTTP (80) e HTTPS (443).

Você pode especificar os grupos de segurança adicionais que você tenha criado para permitir o tráfego em outras portas ou de outras fontes. Por exemplo, você pode criar um grupo de segurança para acessar SSH que permita o tráfego de entrada na porta 22 a partir de um intervalo restrito de endereços IP. Caso contrário, para segurança adicional, crie um que permita tráfego de um host bastion ao qual somente você pode acessar.

### Note

Para permitir tráfego entre as instâncias do ambiente A e as instâncias do ambiente B, você pode adicionar uma regra ao grupo de segurança que o Elastic Beanstalk anexou ao ambiente B. Depois, você pode especificar o grupo de segurança que o Elastic Beanstalk anexou ao ambiente A. Isso permite tráfego de entrada e tráfego de saída nas instâncias do ambiente A. No entanto, isso cria uma dependência entre os dois grupos de segurança. Se mais tarde você tentar encerrar o ambiente A, o Elastic Beanstalk não poderá excluir o grupo de segurança do ambiente, pois o grupo de segurança do ambiente B depende dele. Por isso, recomendamos primeiro criar um grupo de segurança separado. Depois, anexe-o ao ambiente A e especifique-o em uma regra do grupo de segurança do ambiente B.

Para obter mais informações sobre os grupos de segurança do Amazon EC2, consulte [Grupos de segurança do Amazon EC2](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

## Configurações da categoria de capacidade

As configurações a seguir, relacionadas a instâncias do Amazon EC2, estão disponíveis na categoria de configuração Capacity (Capacidade).

### Opções

- [Tipos de instância](#)
- [ID de AMI](#)

Elastic Beanstalk > Environments > Gettingstartedz-env > Configuration

## Modify capacity

Configure the compute capacity of your environment and auto scaling settings to optimize the number of instances used.

### Auto scaling group

---

**Instance types**  
Add instance types for your fleet. Change the order that the instances are in to set the preferred launch order. This only affects On-Demand instances. We recommend you include at least two instance types. [Learn more](#)

-- Choose instance types --

t4g.medium X t4g.2xlarge X

**AMI ID**  
ami-00123f0296f04e610

## Tipos de instância

A configuração de tipos de instância determina o tipo de instância do Amazon EC2 iniciada para executar sua aplicação. Esta página de configuração mostra uma lista de Tipos de instância. Você pode selecionar um ou mais tipo de instância. O console do Elastic Beanstalk exibe apenas os tipos de instância com base na arquitetura do processador configurada para seu ambiente. Portanto, você só pode adicionar tipos de instância da mesma arquitetura de processador.

### Note

Embora o console do Elastic Beanstalk não ofereça a opção de alterar a arquitetura do processador de um ambiente existente, você pode fazer isso com o AWS CLI. Por exemplo, com os comandos, consulte [Configurar instâncias do EC2 da AWS para seu ambiente usando a AWS CLI](#).

Escolha uma instância com capacidade suficiente para executar sua aplicação sob carga, mas não com tanta capacidade que ela fique ociosa a maior parte do tempo. Para fins de desenvolvimento, a família t2 de instâncias fornece poder de processamento moderado com capacidade de intermitência por curtos períodos de tempo. Para aplicações de alta disponibilidade e grande escala, use um grupo de instâncias para garantir que a capacidade não seja afetada significativamente se uma instância parar de funcionar. Comece com um tipo de instância que permita executar cinco instâncias com carga moderada durante o horário normal. Se alguma instância falhar, as demais poderão absorver o restante do tráfego. O buffer de capacidade também permite tempo para o ambiente expandir conforme o tráfego começa a subir durante horários de pico.

Para obter mais informações sobre as famílias e os tipos de instância do Amazon EC2, consulte [Tipos de instância](#), no Guia do usuário do Amazon EC2 para instâncias do Linux ou [Tipos de instância](#) no Guia do usuário do Amazon EC2 para instâncias do Windows. Para determinar quais tipos de instâncias disponíveis atendem aos seus requisitos e quais as regiões suportadas, consulte [Tipos de instância](#) no Guia do usuário do Amazon EC2 para instâncias do Linux ou [Tipos de instância disponíveis](#) no Guia do usuário do Amazon EC2 para instâncias do Windows.

## ID de AMI

A imagem de máquina da Amazon (AMI) é a imagem de máquina do Amazon Linux ou do Windows Server que o Elastic Beanstalk usa para iniciar instâncias do Amazon EC2 no ambiente. O Elastic Beanstalk fornece imagens de máquinas que contêm as ferramentas e os recursos necessários para executar a aplicação.

O Elastic Beanstalk seleciona uma AMI padrão para o ambiente com base na região, versão da plataforma e arquitetura do processador escolhida. Se você tiver criado uma [AMI personalizada](#), substitua o ID padrão da AMI pelo seu próprio padrão personalizado.

## Configurar instâncias do EC2 da AWS para seu ambiente usando a AWS CLI

Usar a Interface de linha de comando da AWS (AWS CLI) para criar e configurar ambientes do Elastic Beanstalk usando comandos no shell da linha de comando. Esta seção fornece exemplos de comandos [create-environment](#) e [update-environment](#)

Os dois primeiros exemplos criam um novo ambiente. O comando especifica um tipo de instâncias do Amazon EC2, t4g.small, baseado na arquitetura do processador arm64. O Elastic Beanstalk usa como padrão a ID de imagem (AMI) para as instâncias do EC2 com base na região, na versão da plataforma e no tipo de instância. O tipo de instância corresponde a uma arquitetura de processador. O parâmetro `solution-stack-name` se aplica à versão da plataforma.

Example 1 – criar um novo ambiente baseado em arm64 (opções de namespace inline)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings \

```

```
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role \
Namespace=aws:ec2:instances,OptionName=InstanceTypes,Value=t4g.small
```

Outra opção é usar um arquivo `options.json` para especificar as opções de namespace em vez de incluí-las inline.

Example 2 – criar um novo ambiente baseado em arm64 (opções de namespace no arquivo `options.json`)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings file://options.json
```

Example

```
example options.json ###
[
 {
 "Namespace": "aws:autoscaling:launchconfiguration",
 "OptionName": "IamInstanceProfile",
 "Value": "aws-elasticbeanstalk-ec2-role"
 },
 {
 "Namespace": "aws:ec2:instances",
 "OptionName": "InstanceTypes",
 "Value": "t4g.small"
 }
]
```

Os próximos dois exemplos atualizam a configuração de um ambiente existente com o comando [update-environment](#). Neste exemplo, estamos adicionando outro tipo de instância que também se baseia na arquitetura do processador arm64. Para ambientes existentes, todos os tipos de instância adicionados devem ter a mesma arquitetura de processador. Se você quiser substituir os tipos de instância existentes por aqueles de uma arquitetura diferente, você pode fazer isso. Mas certifique-se de que todos os tipos de instância no comando tenham o mesmo tipo de arquitetura.



### Example 3 – atualize um ambiente baseado em arm64 existente (opções de namespace inline)

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role \
Namespace=aws:ec2:instances,OptionName=InstanceTypes,Value=t4g.small,t4g.micro
```

Outra opção é usar um arquivo `options.json` para especificar as opções de namespace em vez de incluí-las inline.

### Example 4 – atualize um ambiente baseado em arm64 existente (opções de namespace no arquivo `options.json`)

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings file://options.json
```

### Example

```
example options.json ###
[
 {
 "Namespace": "aws:autoscaling:launchconfiguration",
 "OptionName": "IamInstanceProfile",
 "Value": "aws-elasticbeanstalk-ec2-role"
 },
 {
 "Namespace": "aws:ec2:instances",
 "OptionName": "InstanceTypes",
 "Value": "t4g.small, t4g.micro"
 }
]
```

Os próximos dois exemplos mostram mais comandos de [create-environment](#). Esses exemplos não fornecem valores para InstanceTypes. Quando valores InstanceTypes não são especificados, o Elastic Beanstalk usa como padrão a arquitetura de processador baseada em x86. A ID de imagem (AMI) para as instâncias do EC2 do ambiente será padrão de acordo com a Região, a versão da plataforma e o tipo de instância padrão. O tipo de instância corresponde a uma arquitetura de processador.

Example 5 – criar um novo ambiente baseado em x86 (opções de namespace inline)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role
```

Outra opção é usar um arquivo `options.json` para especificar as opções de namespace em vez de incluí-las inline.

Example 6 – criar um novo ambiente baseado em x86 (opções de namespace no arquivo `options.json`)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings file://options.json
```

Example

```
example options.json ###
[
 {
 "Namespace": "aws:autoscaling:launchconfiguration",
 "OptionName": "IamInstanceProfile",
 "Value": "aws-elasticbeanstalk-ec2-role"
 }
]
```

]

## Recomendações para ambientes de primeiro grupo Graviton arm64

### Note

Esta seção só se aplica a um subconjunto de clientes. Se você criou um novo ambiente com os tipos de instância baseados no Graviton arm64 antes de 24 de novembro de 2021, as informações nesta seção talvez se apliquem a você.

### Ações recomendadas para ambientes de primeiro grupo Graviton arm64

A partir de outubro e novembro de 2021, o Elastic Beanstalk começou a adicionar grupos de suporte para processadores Graviton arm64 em algumas regiões e para algumas versões da plataforma. Esse primeiro grupo foi anunciado nas Notas de Release do AWS Elastic Beanstalk de [13 de outubro](#), [21 de outubro](#) e [19 de novembro](#) de 2021. Se você criou ambientes baseados em arm64, as instruções informaram para configurar a instância com AMIs personalizadas fornecidas nas notas de release. Agora que o suporte aprimorado para Graviton arm64 está disponível, o Elastic Beanstalk usa como padrão as AMIs para os tipos de instância arm64 nas versões mais recentes da plataforma.

Se você criou ambientes com a AMIs personalizadas fornecidas nas releases do primeiro grupo, recomendamos que você faça o seguinte para manter um ambiente saudável e funcional.

1. Remova as AMIs personalizadas do seu ambiente.
2. Atualize o ambiente para a versão mais recente da plataforma.
3. Configure as [atualizações de plataforma gerenciada](#) para fazer upgrade automaticamente para a versão mais recente da plataforma durante uma janela de manutenção programada.

### Note

O Elastic Beanstalk não substituirá automaticamente as AMIs personalizadas. Você deve excluir as AMIs personalizadas na Etapa 1, para que a próxima atualização da plataforma na Etapa 2 as atualize.

O procedimento a seguir orienta você por essas etapas. O exemplos da AWS CLI se aplicam a um ambiente que foi criado com as informações a seguir.

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role \
Namespace=aws:ec2:instances,OptionName=InstanceTypes,Value=t4g.small \
Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId,Value=ami-0fbdb88ce139244bf
```

Como atualizar ambientes arm64 criados no primeiro grupo de suporte do Graviton arm64

1. Execute [update-environment](#) para remover as configurações de AMIs personalizadas.

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--environment-name my-env \
--options-to-remove \
Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId
```

2. Atualize o ambiente para a versão mais recente da plataforma. Escolha uma das seguintes opções.
  - Opção de console: use o console do Elastic Beanstalk para atualizar a versão da plataforma. Para obter mais informações, consulte [Atualizar a versão da plataforma do seu ambiente](#).
  - Opção da AWS CLI: execute o comando [update-environment](#) da AWS, especificando a versão da plataforma disponível mais recente.

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.9 running Docker"
```

**Note**

O [list-available-solution-stacks](#) comando fornece uma lista das versões da plataforma disponíveis para sua conta em uma AWS região.

```
aws elasticbeanstalk list-available-solution-stacks --region us-east-1 --
query SolutionStacks
```

- Use o console do Elastic Beanstalk para configurar atualizações de plataforma gerenciada para seu ambiente. As atualizações da plataforma gerenciada fazem upgrade automaticamente do seu ambiente para a versão mais recente da plataforma durante uma janela de manutenção programada. Sua aplicação permanece em serviço durante o processo de atualização. Para obter mais informações, consulte [atualizações da plataforma gerenciada](#).

## O namespace `aws:autoscaling:launchconfiguration`.

Você pode usar as [opções de configuração](#) no namespace [aws:autoscaling:launchconfiguration](#) para configurar as instâncias para o seu ambiente, inclusive opções adicionais que não estão disponíveis no console.

O exemplo de [arquivo de configuração](#) a seguir usa as opções de configuração básicas que são mostradas neste tópico. Por exemplo, ele usa a opção `DisableIMDSv1`, que é discutida em [IMDS](#). Ele também usa as opções `EC2KeyName` e `IamInstanceProfile`, discutidas em [Segurança](#), e a opção `BlockDeviceMappings`, que não está disponível no console.

```
option_settings:
 aws:autoscaling:launchconfiguration:
 SecurityGroups: my-securitygroup
 MonitoringInterval: "1 minute"
 DisableIMDSv1: false
 EC2KeyName: my-keypair
 IamInstanceProfile: "aws-elasticbeanstalk-ec2-role"
 BlockDeviceMappings: "/dev/sdj=:100,/dev/sdh=snap-51eef269,/dev/sdb=ephemeral0"
```

Você pode usar `BlockDeviceMappings` para configurar dispositivos de blocos adicionais para as instâncias. Para obter mais informações, consulte [Mapeamento de dispositivos de blocos](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte [Valores recomendados](#).

## Configurar o serviço de metadados da instância nas instâncias do ambiente

Metadados de instância são dados relacionados a uma instância do Amazon Elastic Compute Cloud (Amazon EC2) que as aplicações podem usar para configurar ou gerenciar a instância em execução. O serviço de metadados da instância (IMDS) é um componente na instância que o código na instância usa para acessar metadados da instância com segurança. Esse código pode ser código da plataforma do Elastic Beanstalk nas instâncias do ambiente, o SDK da AWS que a aplicação pode estar usando ou até mesmo o código da própria aplicação. Para obter mais informações, consulte [Metadados de instância e dados do usuário](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

O código pode acessar metadados de instância de uma instância em execução usando um dos dois métodos: serviço de metadados de instância versão 1 (IMDSv1) ou serviço de metadados de instância versão 2 (IMDSv2). O IMDSv2 usa solicitações orientadas para sessão e mitiga vários tipos de vulnerabilidades que podem ser usadas para tentar acessar o IMDS. Para obter informações sobre esses dois métodos, consulte [Configuração do serviço de metadados de instância](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

### Seções

- [Suporte de plataforma ao IMDS](#)
- [Escolher métodos de IMDS](#)
- [Configurar o IMDS usando o console do Elastic Beanstalk](#)
- [O namespace aws:autoscaling:launchconfiguration](#)

## Suporte de plataforma ao IMDS

Versões mais antigas da plataforma do Elastic Beanstalk eram compatíveis com IMDSv1. As versões mais recentes da plataforma do Elastic Beanstalk (todas as [versões da plataforma do Amazon Linux 2](#)) são compatíveis com IMDSv1 e IMDSv2. É possível configurar o ambiente para oferecer suporte aos dois métodos (o padrão) ou desabilitar o IMDSv1.

**Note**

Desabilitar o IMDSv1 requer o uso dos modelos de lançamento do Amazon EC2. Quando você configura esse recurso durante a criação ou nas atualizações do ambiente, o Elastic Beanstalk tenta configurar o ambiente para usar os modelos de lançamento do Amazon EC2 (se o ambiente ainda não os estiver usando). Nesse caso, se a sua política de usuário não tiver as permissões necessárias, a criação ou as atualizações do ambiente poderão falhar. Portanto, recomendamos que você use nossa política de usuário gerenciada ou adicione as permissões necessárias às suas políticas personalizadas. Para obter detalhes sobre as permissões necessárias, consulte [the section called “Criar uma política de usuário personalizada”](#).

## Escolher métodos de IMDS

Ao tomar uma decisão sobre os métodos de IMDS que você deseja que o ambiente suporte, considere os seguintes casos de uso do:


- **SDK da AWS:** se sua aplicação usar um SDK da AWS, lembre-se de usar a versão mais recente do SDK. Os AWS SDKs fazem chamadas de IMDS e as versões mais recentes do SDK usam o IMDSv2 sempre que possível. Se você algum dia desabilitar o IMDSv1, ou se sua aplicação usar uma versão antiga do SDK, as chamadas de IMDS podem falhar.
- **Código da sua aplicação:** se a aplicação fizer chamadas de IMDS, considere usar o SDK da AWS para poder fazer as chamadas em vez de fazer solicitações HTTP diretas. Assim, não é necessário fazer alterações no código para alternar entre os métodos de IMDS. O AWS SDK usa o IMDSv2 sempre que possível.
- **Código da plataforma do Elastic Beanstalk:** nosso código faz chamadas de IMDS por meio da AWS SDK e, portanto, usa o IMDSv2 em todas as versões de plataforma de suporte. Se o código usa um AWS SDK atualizado e faz todas as chamadas de IMDS por meio do SDK, é possível desabilitar o IMDSv1 com segurança.

## Configurar o IMDS usando o console do Elastic Beanstalk

É possível modificar a configuração da instância do Amazon EC2 do ambiente do Elastic Beanstalk no console do Elastic Beanstalk.

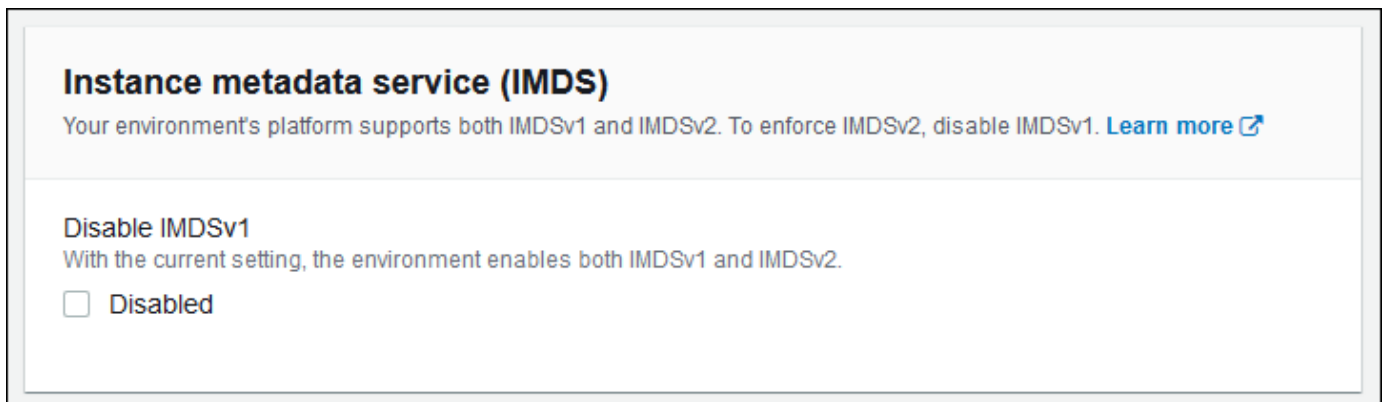
Como configurar o IMDS em suas instâncias do Amazon EC2 no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar).



5. Defina Desabilitar o IMDSv1 para aplicar o IMDSv2. Desmarque Desabilitar o IMDSv1 para habilitar o IMDSv1 e o IMDSv2.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

## O namespace aws:autoscaling:launchconfiguration

É possível usar uma [opção de configuração](#) no namespace [aws:autoscaling:launchconfiguration](#) para configurar o IMDS nas instâncias do ambiente.

O exemplo de [arquivo de configuração](#) a seguir desabilita o IMDSv1 usando a opção `DisableIMDSv1`.

```
option_settings:
 aws:autoscaling:launchconfiguration:
 DisableIMDSv1: true
```



## Grupo do Auto Scaling para o seu ambiente do Elastic Beanstalk

Seu ambiente AWS Elastic Beanstalk inclui um grupo do Auto Scaling que gerencia as [instâncias do Amazon EC2](#) no seu ambiente. Em um ambiente de instância única, o grupo de Auto Scaling garante que sempre haja uma instância em execução. Em um ambiente com balanceamento de carga, o grupo é configurado com um intervalo de instâncias a serem executadas, e o Auto Scaling adiciona ou remove instâncias conforme necessário com base na carga.

O grupo do Auto Scaling também aplica a configuração de execução para as instâncias no seu ambiente. Você pode [modificar a configuração de execução](#) para alterar o tipo de instância, o par de chaves, o armazenamento do Amazon Elastic Block Store (Amazon EBS) e outras configurações que só podem ser definidas quando você inicia uma instância.

O grupo de Auto Scaling usa dois alarmes do Amazon CloudWatch para acionar operações de escalabilidade. Os acionadores padrão são escalados quando a média de tráfego de rede de saída de cada instância é mais alta que 6 MiB ou mais baixa que 2 MiB durante um período de cinco minutos. Para usar o Auto Scaling com eficiência, [configure acionadores](#) que são apropriados para sua aplicação, tipo de instância e requisitos de serviço. Você pode dimensionar com base em várias estatísticas, incluindo latência, E/S de disco, utilização de CPU e a contagem de solicitações.

Para otimizar a utilização de instâncias do Amazon EC2 pelo seu ambiente por meio de períodos previsíveis de tráfego de pico, [configure o grupo de Auto Scaling para alterar a contagem de instâncias dele com base em um agendamento](#). Você pode programar alterações em sua configuração do grupo que sejam repetidas uma vez por dia ou uma vez por semana, ou programar as alterações de uma só vez para se preparar para eventos de marketing que irão gerar uma grande quantidade de tráfego para o seu site.

Como opção, o Elastic Beanstalk pode combinar instâncias sob demanda e [Spot](#) para o seu ambiente. Você pode configurar o Amazon EC2 Auto Scaling para monitorar e responder automaticamente às alterações que afetam a disponibilidade de suas instâncias spot habilitando o [Rebalanceamento de capacidade](#).

O Auto Scaling monitora a integridade de cada instância do Amazon EC2 que ele executa. Se alguma instância for encerrada inesperadamente, o Auto Scaling detectará o encerramento e executará uma instância de substituição. Para configurar o grupo para usar o mecanismo de verificação de integridade do balanceador de carga, consulte [Configuração da verificação de integridade do Auto Scaling](#).

É possível configurar o Auto Scaling para seu ambiente usando o [console do Elastic Beanstalk](#), a [CLI do EB](#) ou as [opções de configuração](#).

## Tópicos

- [Suporte à instância spot](#)
- [Configuração do grupo de Auto Scaling usando o console do Elastic Beanstalk](#)
- [Configuração do grupo de Auto Scaling usando a CLI do EB](#)
- [Opções de configuração](#)
- [Gatilhos do ajuste de escala automático](#)
- [Ações programadas de ajuste de escala automático](#)
- [Configuração da verificação de integridade do Auto Scaling](#)

## Suporte à instância spot

Para aproveitar as [Instâncias spot](#) do Amazon EC2, você pode habilitar uma opção Spot para o seu ambiente. O grupo de Auto Scaling do ambiente combina opções de compra do Amazon EC2 e mantém uma combinação de instâncias sob demanda e spot.

Este tópico descreve os seguintes métodos para habilitar solicitações de instância spot para seu ambiente:

- O console do Elastic Beanstalk — Para obter mais informações, consulte [Composição da frota em the section called “Configuração do grupo de Auto Scaling usando o console do Elastic Beanstalk”](#).
- A CLI do EB - Para obter mais informações, consulte [the section called “Configuração do grupo de Auto Scaling usando a CLI do EB”](#).
- A `aws:ec2:instances` Opção de configuração de namespace — Para obter mais informações, consulte [the section called “Opções de configuração”](#).

### Important

A demanda por instâncias Spot pode variar significativamente de um momento para outro, e a disponibilidade das instâncias Spot também pode variar significativamente dependendo de quantas instâncias do Amazon EC2 não utilizadas estão disponíveis. É sempre possível que sua instância spot seja interrompida.

Para ajudar a minimizar o impacto dessas interrupções em sua aplicação, você pode habilitar a opção Rebalanceamento de capacidade incluída no Amazon EC2 Auto Scaling. Com esse recurso habilitado, o EC2 tenta substituir automaticamente instâncias spot em um grupo do Auto Scaling antes que elas sejam interrompidas. Para habilitar esse recurso, use o console do Elastic Beanstalk para [configurar o grupo do Auto Scaling](#). Como alternativa, é possível definir a `EnableCapacityRebalancing` [opção de configuração](#) do Elastic Beanstalk no `true` namespace [aws:autoscaling:asg](#).

Para obter mais informações, consulte [Rebalanceamento de capacidade](#) no Manual do usuário do Amazon EC2 Auto Scaling e [Interrupções de instâncias spot](#) no Manual do usuário do Amazon EC2 para instâncias do Linux.

O Elastic Beanstalk fornece várias opções de configuração para oferecer suporte ao recurso spot. Elas são abordadas nas seções a seguir sobre como configurar o grupo do Auto Scaling.

Duas dessas opções, no namespace [aws:ec2:instances](#) merecem atenção especial:

- `SpotFleetOnDemandBase`
- `SpotFleetOnDemandAboveBasePercentage`

Essas duas opções se correlacionam com a `MinSize` opção no namespace [aws:autoscaling:asg](#):

- Somente `MinSize` determina a capacidade inicial do seu ambiente — o número de instâncias que você deseja executar no mínimo.
- `SpotFleetOnDemandBase` não afeta a capacidade inicial. Quando o Spot está habilitado, essa opção determina somente quantas instâncias sob demanda são provisionadas antes de qualquer instância spot ser considerada.
- Considere quando `SpotFleetOnDemandBase` for menor que `MinSize`. Você ainda vai ter exatas `MinSize` instâncias como capacidade inicial. No mínimo `SpotFleetOnDemandBase` delas devem ser instâncias sob demanda.
- Considere quando `SpotFleetOnDemandBase` for maior que `MinSize`. À medida que o ambiente tiver aumento da escala, você terá, pelo menos, uma quantidade adicional de instâncias igual à diferença entre os dois valores. Em outras palavras, você tem a garantia de que obterá pelo menos uma (`SpotFleetOnDemandBase - MinSize`) instância adicional sob demanda antes de atender ao `SpotFleetOnDemandBase` requisito.

Em ambientes de produção, as instâncias spot são particularmente úteis como parte de um ambiente escalável e com balanceamento de carga. Não recomendamos o uso de spot em um ambiente de instância única. Se as instâncias spot não estiverem disponíveis, você poderá perder toda a capacidade (uma única instância) do seu ambiente. Você ainda pode usar uma instância spot em um ambiente de instância única para desenvolvimento ou teste. Neste caso, certifique-se de definir ambas `SpotFleetOnDemandBase` e `SpotFleetOnDemandAboveBasePercentage` de zerar. Quaisquer outras configurações resultam em uma instância sob demanda.

### Observações

- Algumas contas da AWS mais antigas podem fornecer o Elastic Beanstalk com tipos de instância padrão que não oferecem suporte a instâncias spot (por exemplo, t1.micro). Se você habilitar solicitações de instância spot e vir o erro Nenhum dos tipos de instância especificados oferece suporte a spot, certifique-se de configurar os tipos de instância que oferecem suporte a spot. Para escolher os tipos de instância spot, use o [Spot Instance Advisor](#).
- Habilitar solicitações de instância spot requer o uso de modelos de execução do Amazon EC2. Quando você configura esse recurso durante a criação ou nas atualizações do ambiente, o Elastic Beanstalk tenta configurar o ambiente para usar modelos de lançamento do Amazon EC2 (se o ambiente ainda não os estiver usando). Nesse caso, se a sua política de usuário não tiver as permissões necessárias, a criação ou as atualizações do ambiente poderão falhar. Portanto, recomendamos que você use nossa política de usuário gerenciada ou adicione as permissões necessárias às suas políticas personalizadas. Para obter detalhes sobre as permissões necessárias, consulte [the section called “Criar uma política de usuário personalizada”](#).

Os exemplos a seguir demonstram diferentes cenários de configurações das várias opções de escalabilidade. Todos os exemplos pressupõem um ambiente com balanceamento de carga e com solicitações de instância spot habilitadas.

## Example 1: Sob demanda e spot como parte da capacidade inicial

### Configurações de opção

Opção	Namespace	Valor
MinSize	aws:autoscaling:asg	10
MaxSize	aws:autoscaling:asg	24
SpotFleetOnDemandBase	aws:ec2:instances	4
SpotFleetOnDemandAboveBasePercentage	aws:ec2:instances	50

Neste exemplo, o ambiente começa com dez instâncias, das quais sete são sob demanda (quatro de base e 50% das seis acima da base) e três são spot. O ambiente pode ser expandido até 24 instâncias. Conforme ele é expandido, a quantidade sob demanda na parte da frota acima das quatro instâncias sob demanda de base é mantida em 50%, até um máximo de 24 instâncias no geral, das quais 14 são sob demanda (quatro de base e 50% das 20 acima da base) e dez são spot.

## Example 2: Toda a capacidade inicial sob demanda

### Configurações de opção

Opção	Namespace	Valor
MinSize	aws:autoscaling:asg	4
MaxSize	aws:autoscaling:asg	24
SpotFleetOnDemandBase	aws:ec2:instances	4
SpotFleetOnDemandAboveBasePercentage	aws:ec2:instances	50

Neste exemplo, o ambiente começa com quatro instâncias, todas sob demanda. O ambiente pode ser expandido até 24 instâncias. Conforme ele é expandido, a quantidade sob demanda na parte da

frota acima das quatro instâncias sob demanda de base é mantida em 50%, até um máximo de 24 instâncias no geral, das quais 14 são sob demanda (quatro de base e 50% das 20 acima da base) e dez são spot.

### Example 3: Base adicional sob demanda além da capacidade inicial

#### Configurações de opção

Opção	Namespace	Valor
MinSize	aws:autoscaling:asg	3
MaxSize	aws:autoscaling:asg	24
SpotFleetOnDemandBase	aws:ec2:instances	4
SpotFleetOnDemandAboveBasePercentage	aws:ec2:instances	50

Neste exemplo, o ambiente começa com três instâncias, todas sob demanda. O ambiente pode ser expandido até 24 instâncias. A primeira instância adicional acima das três iniciais é sob demanda, para concluir as quatro instâncias sob demanda de base. À medida que ele é expandido ainda mais, a quantidade sob demanda, na parte da frota acima das quatro instâncias sob demanda de base, é mantida em 50%, até um máximo de 24 instâncias no geral, das quais 14 são sob demanda (quatro de base e 50% das 20 acima da base) e dez são spot.

## Configuração do grupo de Auto Scaling usando o console do Elastic Beanstalk

É possível configurar como o Auto Scaling funciona editando as opções de Capacidade na página de Configuração do ambiente no [console do Elastic Beanstalk](#).


Como configurar o grupo de Auto Scaling no console Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Na seção Grupo do Auto Scaling, defina as configurações a seguir.
  - Tipo de ambiente: selecione Carga balanceada.
  - Mínimo de instâncias: o número mínimo de instâncias do EC2 que o grupo deve conter a qualquer momento. O grupo começa com o número mínimo de contagens e adiciona instâncias quando a condição de acionador aumentar a escala na vertical é atendida.
  - Máximo de instâncias: o número máximo de instâncias do EC2 que o grupo deve conter a qualquer momento.

 Note

Se você usar atualizações cumulativas, verifique se a contagem máxima de instância é mais alta que a configuração do [Minimum instances in service \(Mínimo de instâncias em serviço\)](#) para atualizações cumulativas.

- Composição da frota— O padrão é instâncias sob demanda. Para habilitar as solicitações de instância spot , selecione Opções e Instâncias de compras combinadas.

As opções a seguir são habilitadas se você selecionar Habilitar as solicitações de Instância Spot:

- Preço Spot máximo — Para obter recomendações sobre opções de preço máximo para instâncias spot, consulte [Histórico de definição de preço da instância spot](#) no Manual do usuário do Amazon EC2 para instâncias do Linux.
- Base sob demanda – O número mínimo de instâncias sob demanda que o grupo do Auto Scaling provisiona antes de considerar as instâncias spot à medida que o ambiente tiver aumento de escala.
- Sob demanda acima da base – O percentual de instâncias sob demanda como parte da capacidade adicional que o grupo do Auto Scaling provisiona além das instâncias sob demanda.

**Note**

As opções On-Demand base (Base sob demanda) e On-Demand above base (Sob demanda acima da base) correlacionam-se com as de opções das instâncias Min (Mín.) e Max (Máx.) listadas anteriormente. Para obter mais informações sobre essas opções e exemplos, consulte [the section called “Suporte à instância spot”](#).

- Habilitar rebalanceamento de capacidade – Essa opção só é relevante quando há pelo menos uma instância spot em seu grupo do Auto Scaling. Quando esse recurso é habilitado, o EC2 tenta automaticamente substituir instâncias spot no grupo do Auto Scaling antes que elas sejam interrompidas, minimizando as interrupções de instância spot em suas aplicações. Para obter mais informações, consulte [Rebalanceamento de capacidade](#) no Manual do usuário do Amazon EC2 Auto Scaling.
- Tipo de instância: o tipo de instância do Amazon EC2 executada para iniciar sua aplicação. Para obter mais detalhes, consulte [the section called “Tipos de instância”](#).
- ID da AMI: a imagem da máquina que o Elastic Beanstalk usa para iniciar instâncias do Amazon EC2 no ambiente. Para obter mais detalhes, consulte [the section called “ID de AMI”](#).
- Availability Zones (Zonas de disponibilidade): escolha o número de zonas de disponibilidade nas quais distribuir as instâncias do ambiente. Por padrão, o grupo de Auto Scaling executa instâncias uniformemente em todas as regiões utilizáveis. Para concentrar suas instâncias em um número menor de zonas, escolha o número de zonas a serem usadas. Para ambientes de produção, use pelo menos duas zonas para garantir que seu aplicativo está disponível caso uma zona de disponibilidade fique fora.
- Placement (Posicionamento) (opcional): escolha as zonas de disponibilidade a serem usadas. Use esta configuração se suas instâncias precisarem se conectar a recursos em regiões específicas ou se você tiver adquirido [instâncias reservadas](#), que são específicas da região. Se você executar seu ambiente em uma VPC personalizada, não poderá configurar essa opção. Em uma VPC personalizada, é possível escolher as zonas de disponibilidade para as sub-redes que você atribui para o seu ambiente.
- Scaling cooldown (Desaquecimento da escalabilidade): o tempo de espera, em segundos, para que as instâncias sejam executadas ou encerradas depois da escalabilidade, antes de continuar a avaliar triggers. Para obter mais informações, consulte [Desaquecimento da escalabilidade](#).



Elastic Beanstalk > Environments > Gettingstarted-env > Configuration

## Modify capacity

Configure the compute capacity of your environment and Auto Scaling settings to optimize the number of instances used.

### Auto Scaling Group

Environment type  
Load balanced

Instances  
Min 1  
Max 4

Fleet composition  
Choose a mix of On-Demand and Spot Instances with multiple instance types. Spot Instances are automatically launched at the lowest available price. [Learn more](#)

On-Demand instances  
 Combine purchase options and instances

Maximum spot price  
The maximum price per instance-hour, in USD, that you're willing to pay for a Spot Instance. Setting a custom price limits your chances to fulfill your target capacity using Spot instances.  
 Default - the On-Demand price for each instance type (recommended)  
 Set your maximum price

On-Demand base  
The minimum number of On-Demand Instances that your Auto Scaling group provisions before considering Spot Instances as your environment scales out.  
0

On-Demand above base  
The percentage of On-Demand Instances as part of any additional capacity that your Auto Scaling group provisions beyond the On-Demand base instances.  
70 %

Enable Capacity Rebalancing  
Specifies whether to enable the Capacity Rebalancing feature for Spot Instances in your Auto Scaling Group. This option is only relevant when EnableSpot is true in the aws:ec2:instances namespace, and there is at least one Spot Instance in your Auto Scaling group.  
 Enabled

Instance types  
Add acceptable instance types for your fleet. Change their order to set the launch priority of On-Demand Instances. This order doesn't affect Spot Instances. We recommend a minimum of two instance types. [Learn more](#)  
-- Choose Instance Types --  
t2.micro (1vCPUs, 1GiB) X t2.small (1vCPUs, 2GiB) X

AMI ID  
ami-9999999999999999

Availability Zones  
Number of Availability Zones (AZs) to use.  
Any

Placement  
Specify Availability Zones (AZs) to use.  
-- Choose Availability Zones (AZs) --

Scaling cooldown  
360 seconds

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

## Configuração do grupo de Auto Scaling usando a CLI do EB

Ao criar um ambiente usando o comando [eb create](#), você pode especificar algumas opções relacionadas ao grupo do Auto Scaling no seu ambiente. Essas são algumas das opções que ajudam você a controlar a capacidade do seu ambiente.

### --single

Cria o ambiente com uma instância do Amazon EC2 e nenhum balanceador de carga. Se você não usa essa opção, um balanceador de carga é adicionado ao ambiente que é criado.

### --enable-spot

Habilita solicitações de instância spot para seu ambiente.

As opções a seguir para o comando [eb create](#) podem ser utilizadas somente com `--enable-spot`.

### --instance-types

Lista os tipos de instâncias do Amazon EC2 que você deseja que seu ambiente use.

### --spot-max-price

O preço máximo por hora, em USD, que você está disposto a pagar por uma instância spot. Para obter recomendações sobre opções de preço máximo para Instâncias spot, consulte o [Histórico de definição de preço da instância spot](#) no Manual do usuário do Amazon EC2 para instâncias do Linux.

### --on-demand-base-capacity

O número mínimo de instâncias sob demanda que o grupo de Auto Scaling provisiona antes de considerar as instâncias spot à medida que o ambiente é dimensionado.

### --on-demand-above-base-capacity

A porcentagem de instâncias sob demanda como parte da capacidade adicional que o grupo de Auto Scaling provisiona além do número de instâncias especificado pela opção `--on-demand-base-capacity`.

O exemplo a seguir cria um ambiente e configura o grupo de Auto Scaling para habilitar solicitações de instância spot para o novo ambiente. Para este exemplo, três tipos de instância podem ser usados.

```
$ eb create --enable-spot --instance-types "t2.micro,t3.micro,t3.small"
```

### ⚠ Important

Há outra opção nomeada de forma semelhante chamada `--instance-type` (sem "s") que a EB CLI reconhece somente ao processar instâncias sob demanda. Não use `--instance-type` (sem "s") com a opção `--enable-spot`. Se você fizer isso, a EB CLI o ignorará. Em vez disso, use `--instance-types` (com "s") com a opção `--enable-spot`.

## Opções de configuração

O Elastic Beanstalk fornece [opções de configuração](#) para as configurações do Auto Scaling em dois namespaces: [aws:autoscaling:asg](#) e [aws:ec2:instances](#).

### O namespace `aws:autoscaling:asg`

O namespace [aws:autoscaling:asg](#) fornece opções para escalabilidade e disponibilidade geral.

O [arquivo de configuração](#) demonstrativo a seguir configura o grupo de Auto Scaling para usar de duas a quatro instâncias, zonas de disponibilidade específicas e um período de desaquecimento de 12 minutos (720 segundos). O Rebalanceamento de capacidade para instâncias spot está habilitado. Esta última opção só tem efeito se `EnableSpot` for definida como `true` no [aws:ec2:instances](#) namespace, conforme mostrado no exemplo do arquivo de configuração após este.

```
option_settings:
 aws:autoscaling:asg:
 Availability Zones: Any
 Cooldown: '720'
 Custom Availability Zones: 'us-west-2a,us-west-2b'
 MaxSize: '4'
 MinSize: '2'
 EnableCapacityRebalancing: true
```

### O namespace `aws:ec2:instances`

O namespace [aws:ec2:instances](#) fornece opções relacionadas às instâncias do ambiente, incluindo o gerenciamento de instância spot. Ele complementa [aws:autoscaling:launchconfiguration](#) e [aws:autoscaling:asg](#).

Quando você atualiza a configuração do ambiente e remove um ou mais tipos de instância da opção `InstanceTypes`, o Elastic Beanstalk encerra todas as instâncias do Amazon EC2 em execução em qualquer um dos tipos de instância removidos. O grupo de Auto Scaling do seu ambiente executa novas instâncias, conforme necessário, para completar a capacidade desejada, usando os tipos de instância especificados atuais.

O exemplo a seguir de [arquivo de configuração](#) configura o grupo de Auto Scaling para habilitar solicitações de instância spot para seu ambiente. É possível usar três tipos de instância. Pelo menos uma instância sob demanda é usada para capacidade de linha de base, e um percentual sustentado de 33% de instâncias sob demanda é usado para qualquer capacidade adicional.

```
option_settings:
 aws:ec2:instances:
 EnableSpot: true
 InstanceTypes: 't2.micro,t3.micro,t3.small'
 SpotFleetOnDemandBase: '1'
 SpotFleetOnDemandAboveBasePercentage: '33'
```

Para escolher os tipos de instância spot, use o [Spot Instance Advisor](#).

## Gatilhos do ajuste de escala automático

O grupo de Auto Scaling no ambiente do Elastic Beanstalk usa dois alarmes do Amazon CloudWatch para acionar operações de escalabilidade. Os triggers padrão são dimensionados quando a média de tráfego de rede de saída de cada instância é mais alta que 6 MB ou mais baixa que 2 MB durante um período de cinco minutos. Para usar o Amazon EC2 Auto Scaling com eficiência, configure triggers que são apropriados para sua aplicação, tipo de instância e requisitos de serviço. Você pode dimensionar com base em várias estatísticas, incluindo latência, E/S de disco, utilização de CPU e a contagem de solicitações.

Para obter mais informações sobre métricas e alarmes do CloudWatch, consulte [Conceitos do Amazon CloudWatch](#) no Guia do usuário do Amazon CloudWatch.

## Configurar triggers de Auto Scaling

É possível configurar os triggers que ajustam o número de instâncias no grupo de Auto Scaling do seu ambiente no console do Elastic Beanstalk.

## Como configurar triggers no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Na seção Scaling triggers (Escalabilidade de triggers), defina as seguintes configurações:
  - Metric (Métrica): métrica usada para o trigger de Auto Scaling.
  - Statistic (Estatística): cálculo de estatísticas que o trigger deve usar, como Average.
  - Unit (Unidade): unidade da métrica do trigger, como Bytes.
  - Period (Período): especifica com que frequência o Amazon CloudWatch avalia as métricas do trigger.
  - Breach duration (Duração da ruptura): quantidade de tempo, em minutos, em que uma métrica pode estar fora dos limites superiores e inferiores antes de acionar uma operação de escalabilidade.
  - Upper threshold (Limite superior): se a métrica ficar acima desse número pela duração da violação, uma operação de escalabilidade será acionada.
  - Scale up increment (Incremento de aumento): a quantidade de instâncias do Amazon EC2 que devem ser adicionadas ao executar uma ação de escalabilidade.
  - Lower threshold (Limite inferior): se a métrica ficar abaixo desse número pela duração da violação, uma operação de escalabilidade será acionada.
  - Scale down increment (Incremento de redução): a quantidade de instâncias do Amazon EC2 que devem ser removidas ao executar uma ação de escalabilidade.

### Scaling triggers

**Metric**  
Change the metric that is monitored to determine if the environment's capacity is too low or too high.

NetworkOut ▼

**Statistic**  
Choose how the metric is interpreted.

Average ▼

**Unit**

Bytes ▼

**Period**  
The period between metric evaluations.

5 Min

**Breach duration**  
The amount of time a metric can exceed a threshold before triggering a scaling operation.

5 Min

**Upper threshold**

6000000 Bytes

**Scale up increment**

1 EC2 instances

**Lower threshold**

2000000 Bytes

**Scale down increment**

-1 EC2 instances

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

## O namespace `aws:autoscaling:trigger`

O Elastic Beanstalk fornece [opções de configuração](#) para configurações de Auto Scaling no namespace [aws:autoscaling:trigger](#). As configurações neste namespace estão organizadas pelo recurso ao qual elas se aplicam.

```
option_settings:
 AWSEBAutoScalingScaleDownPolicy.aws:autoscaling:trigger:
```

```
LowerBreachScaleIncrement: '-1'
AWSEBAutoScalingScaleUpPolicy.aws:autoscaling:trigger:
 UpperBreachScaleIncrement: '1'
AWSEBCloudwatchAlarmHigh.aws:autoscaling:trigger:
 UpperThreshold: '6000000'
AWSEBCloudwatchAlarmLow.aws:autoscaling:trigger:
 BreachDuration: '5'
 EvaluationPeriods: '1'
 LowerThreshold: '2000000'
 MeasureName: NetworkOut
 Period: '5'
 Statistic: Average
 Unit: Bytes
```

## Ações programadas de ajuste de escala automático

Para otimizar a utilização de instâncias do Amazon EC2 pelo seu ambiente por meio de períodos previsíveis de tráfego de pico, configure o grupo de Amazon EC2 Auto Scaling para alterar a contagem de instâncias dele com base em uma programação. Você pode configurar seu ambiente com uma ação recorrente para expandir todos os dias de manhã e reduzir à noite, quando o tráfego é baixo. Por exemplo, se você tiver um evento de marketing que irá direcionar o tráfego para seu site por um período limitado, você poderá programar um evento único para expandir quando ele começar e outro para reduzir quando ele terminar.

Você pode definir até 120 ações programadas ativas por ambiente. O Elastic Beanstalk também retém até 150 ações programadas expiradas que você pode reutilizar atualizando suas configurações.

### Configurar ações programadas

Você pode criar ações programadas para o grupo de Auto Scaling do ambiente no console do Elastic Beanstalk.

Como configurar ações programadas no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

**Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Na seção Time-based Scaling (Escalabilidade baseada em tempo), escolha Add scheduled action (Adicionar ação programada).

**Time-based Scaling**  
Use the following settings to control time-based scaling actions. [Learn more](#)

Current status  
1 instance(s) in service, Min: 1, Max: 2

Time zone  
 UTC  
 Local

Actions ▾ **Add scheduled action**

<input type="checkbox"/>	Name	Min	Max	Desired	Next occurrence (UTC)
No scheduled actions					

6. Preencha as seguintes configurações de ações programadas:
  - Name (Nome): especifique um nome exclusivo de até 255 caracteres alfanuméricos sem espaços.
  - Instances (Instâncias): selecione a contagem de instâncias mínimas e máximas para se inscrever no grupo de Auto Scaling.
  - Desired capacity (Capacidade desejada) (opcional): defina a capacidade inicial desejada para o grupo de Auto Scaling. Depois que a ação programada for aplicada, os triggers ajustarão a capacidade desejada com base em suas configurações.
  - Occurrence (Ocorrência): escolha Recurring (Recorrente) para repetir a ação de escalabilidade em uma programação.
  - Start time (Hora de início): para ações únicas, selecione a data e a hora para executar a ação.



Para ações recorrentes, é opcional definir um horário de início. Especifique-o para escolher a hora de início em que a ação é executada. Após esse horário, a ação se repete de acordo com a expressão Recurrence (Recorrência).

- Recurrence (Recorrência): Use uma expressão [Cron](#) para especificar a frequência com a qual você deseja executar a ação programada. Por exemplo, `30 6 * * 2` executa a ação todas as terças-feiras às 6:30 horas UTC.
- End time (Hora de término) (opcional): opcional para ações recorrentes. Se especificado, a ação se repete de acordo com a expressão Recurrence (Recorrência) e não será executada novamente após esse horário.

Quando uma ação programada é finalizada, o Auto Scaling não volta automaticamente às suas configurações anteriores. Configure uma segunda ação programada para retornar o Auto Scaling para as configurações originais, conforme necessário.

7. Escolha Adicionar.
8. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

#### Note

Ações agendadas não serão salvas até serem aplicadas.

## O namespace `aws:autoscaling:scheduledaction`

Se você precisar configurar um grande número de ações programadas, poderá usar [arquivos de configuração](#) ou [a API do Elastic Beanstalk](#) para aplicar alterações da opção de configuração de um arquivo YAML ou JSON. Esses métodos também permitem que você acesse a opção [Suspend](#) para desativar uma ação programada recorrente temporariamente.

#### Note

Ao trabalhar com as opções de configuração de ações programadas fora do console, use o formato de tempo da ISO 8601 para especificar horários de início e término em UTC. Por exemplo, `2015-04-28T04:07:02Z`. Para obter mais informações sobre o formato de hora da ISO 8601, consulte [Formatos de data e hora](#). As datas devem ser exclusivas em todas as ações programadas.

O Elastic Beanstalk oferece opções de configuração para definições de ações programadas no namespace [aws:autoscaling:scheduledaction](#). Use o campo `resource_name` para especificar o nome da ação programada.

#### Example Scheduled-scale-up-specific-time-long.config

Este arquivo de configuração instrui o Elastic Beanstalk a aumentar de cinco instâncias para 10 instâncias em 2015-12-12T00:00:00Z.

```
option_settings:
 - namespace: aws:autoscaling:scheduledaction
 resource_name: ScheduledScaleUpSpecificTime
 option_name: MinSize
 value: '5'
 - namespace: aws:autoscaling:scheduledaction
 resource_name: ScheduledScaleUpSpecificTime
 option_name: MaxSize
 value: '10'
 - namespace: aws:autoscaling:scheduledaction
 resource_name: ScheduledScaleUpSpecificTime
 option_name: DesiredCapacity
 value: '5'
 - namespace: aws:autoscaling:scheduledaction
 resource_name: ScheduledScaleUpSpecificTime
 option_name: StartTime
 value: '2015-12-12T00:00:00Z'
```

#### Example Scheduled-scale-up-specific-time.config

Para usar a sintaxe abreviada com a EB CLI ou com arquivos de configuração, coloque o nome do recurso antes do namespace.

```
option_settings:
 ScheduledScaleUpSpecificTime.aws:autoscaling:scheduledaction:
 MinSize: '5'
 MaxSize: '10'
 DesiredCapacity: '5'
 StartTime: '2015-12-12T00:00:00Z'
```

#### Example Scheduled-scale-down-specific-time.config

Esse arquivo de configuração instrui o Elastic Beanstalk a reduzir em 2015-12-12T07:00:00Z.

```
option_settings:
 ScheduledScaleDownSpecificTime.aws:autoscaling:scheduledaction:
 MinSize: '1'
 MaxSize: '1'
 DesiredCapacity: '1'
 StartTime: '2015-12-12T07:00:00Z'
```

### Example Scheduled-periodic-scale-up.config

Este arquivo de configuração instrui o Elastic Beanstalk a fazer escalonamento horizontal todos os dias às 9h. A ação foi programada para começar em 14 de maio de 2015 e terminar em 12 de janeiro de 2016.

```
option_settings:
 ScheduledPeriodicScaleUp.aws:autoscaling:scheduledaction:
 MinSize: '5'
 MaxSize: '10'
 DesiredCapacity: '5'
 StartTime: '2015-05-14T07:00:00Z'
 EndTime: '2016-01-12T07:00:00Z'
 Recurrence: 0 9 * * *
```

### Example Scheduled-periodic-scale-down.config

Este arquivo de configuração instrui o Elastic Beanstalk a reduzir para nenhuma instância em execução todos os dias às 18h. Se você sabe que seu aplicativo está ocioso principalmente fora do horário comercial, você pode criar uma ação programada semelhante. Se o aplicativo tiver de ser desativado fora do horário comercial, altere MaxSize para 0.

```
option_settings:
 ScheduledPeriodicScaleDown.aws:autoscaling:scheduledaction:
 MinSize: '0'
 MaxSize: '1'
 DesiredCapacity: '0'
 StartTime: '2015-05-14T07:00:00Z'
 EndTime: '2016-01-12T07:00:00Z'
 Recurrence: 0 18 * * *
```

## Example Scheduled-weekend-scale-down.config

Este arquivo de configuração instrui o Elastic Beanstalk a reduzir todas as sextas-feiras às 18h. Se você sabe que seu aplicativo não recebe tanto tráfego durante o fim de semana, pode criar uma ação programada semelhante.

```
option_settings:
 ScheduledWeekendScaleDown.aws:autoscaling:scheduledaction:
 MinSize: '1'
 MaxSize: '4'
 DesiredCapacity: '1'
 StartTime: '2015-12-12T07:00:00Z'
 EndTime: '2016-01-12T07:00:00Z'
 Recurrence: 0 18 * * 5
```

## Configuração da verificação de integridade do Auto Scaling

O Amazon EC2 Auto Scaling monitora a integridade de cada instância do Amazon Elastic Compute Cloud (Amazon EC2) iniciada. Se alguma instância for encerrada inesperadamente, o Auto Scaling detectará o encerramento e executará uma instância de substituição. Por padrão, o grupo de Auto Scaling criado para seu ambiente usa [verificações de status do Amazon EC2](#). Se uma instância no ambiente não passar em uma verificação de status do Amazon EC2, o Auto Scaling a desabilitará e a substituirá.

As verificações de status do Amazon EC2 abrangem somente a integridade da instância, e não a integridade da aplicação, do servidor ou de qualquer contêiner do Docker executado na instância. Se a aplicação falhar, mas a instância na qual ela é executada ainda estiver íntegra, ela poderá ser retirada do balanceador de carga, mas não será substituída automaticamente pelo Auto Scaling. O comportamento padrão é adequado para resolução de problemas. Se o Auto Scaling substituir a instância logo após a falha na aplicação, talvez você não perceba que ocorreu um erro, mesmo se ele falhar imediatamente após a inicialização.

Se você quiser que o Auto Scaling substitua instâncias cuja aplicação parou de responder, poderá usar um [arquivo de configuração](#) para configurar o grupo de Auto Scaling para usar verificações de integridade do Elastic Load Balancing. O exemplo a seguir define o grupo para usar as verificações de integridade do balanceador de carga, além da verificação de status do Amazon EC2, para determinar a integridade de uma instância.

## Example .ebextensions/autoscaling.config

```
Resources:
 AWSEBAutoScalingGroup:
 Type: "AWS::AutoScaling::AutoScalingGroup"
 Properties:
 HealthCheckType: ELB
 HealthCheckGracePeriod: 300
```

Para obter mais informações sobre as propriedades `HealthCheckType` e `HealthCheckGracePeriod`, consulte [AWS::AutoScaling::AutoScalingGroup](#) no Manual do usuário do AWS CloudFormation e [Verificações de integridade para instâncias do Auto Scaling](#) no Manual do usuário do Amazon EC2 Auto Scaling.

Por padrão, a verificação de integridade do Elastic Load Balancing é configurada para tentar uma conexão TCP com sua instância na porta 80. Isso confirma que o servidor web em execução na instância está aceitando conexões. No entanto, você pode desejar [personalizar a verificação da integridade do load balancer](#) para garantir que seu aplicativo, e não apenas o servidor web, está em bom estado. A configuração do período de carência define por quantos segundos uma instância pode ser reprovada na verificação de integridade sem ser encerrada e substituída. As instâncias podem ser recuperadas depois de serem retiradas do load balancer, portanto, forneça um tempo adequado suficiente para seu aplicativo.

## Balanceador de carga do ambiente do Elastic Beanstalk

Um load balancer distribui o tráfego entre as instâncias do ambiente. Quando você [habilita o balanceamento de carga](#), a AWS Elastic Beanstalk cria um balanceador de carga do [Elastic Load Balancing](#) dedicado ao ambiente. O Elastic Beanstalk gerencia totalmente esse balanceador de carga, cuidando das configurações de segurança e encerrando o balanceador de carga quando o ambiente é encerrado.

Também há a opção de compartilhar um balanceador de carga em vários ambientes do Elastic Beanstalk. Com um load balancer compartilhado, você economiza no custo operacional evitando um load balancer dedicado para cada ambiente. Você também assume mais responsabilidade de gerenciamento pelo load balancer compartilhado usado pelos ambientes.

O Elastic Load Balancing tem estes tipos de balanceador de carga:

- [Classic Load Balancer](#): o balanceador de carga da geração anterior. Rotas de tráfego de solicitação HTTP, HTTPS ou TCP para portas diferentes em instâncias de ambiente.
- [Application Load Balancer](#): um balanceador de carga da camada de aplicação. Rotas de tráfego de solicitação HTTP ou HTTPS para portas diferentes em instâncias de ambiente com base no caminho da solicitação.
- [Network Load Balancer](#): um balanceador de carga de camada de rede. Rotas de tráfego de solicitação TCP para portas diferentes em instâncias de ambiente. Oferece suporte a verificações de integridade ativas e passivas.

O Elastic Beanstalk é compatível com todos os três tipos de balanceador de carga. A tabela a seguir mostra quais tipos você pode usar com os dois padrões de uso:

Tipo de load balancer	Dedicado	Compartilhado
Classic Load Balancer	✓ Sim	× Não
Application Load Balancer	✓ Sim	✓ Sim
Network Load Balancer	✓ Sim	× Não

#### Note

A opção Classic Load Balancer (CLB) está desabilitada no assistente do console Criar ambiente. Se você tiver um ambiente existente configurado com um Classic Load Balancer, poderá criar um novo [clonando o ambiente existente](#) usando o console do Elastic Beanstalk ou a [CLI do EB](#). Você também tem a opção de usar a [CLI do EB](#) ou a [AWS CLI](#) para criar um novo ambiente configurado com um Classic Load Balancer. Essas ferramentas de linha de comando criarão um novo ambiente com um CLB mesmo que ainda não exista um em sua conta.

Por padrão, o Elastic Beanstalk cria um Application Load Balancer para seu ambiente quando você habilita o balanceamento de carga com o console Elastic Beanstalk ou a CLI do EB. Ele configura o load balancer para ouvir o tráfego HTTP na porta 80 e encaminhar esse tráfego para instâncias na mesma porta. Você só pode escolher o tipo de load balancer que seu ambiente usará

durante a criação do ambiente. Posteriormente, você pode alterar as configurações para gerenciar o comportamento do load balancer do ambiente em execução, mas não pode alterar seu tipo.

#### Note

Seu ambiente deve estar em uma VPC com sub-redes em pelo menos duas zonas de disponibilidade para criar um Application Load Balancer. Todas as novas contas de AWS incluem VPCs padrão que atendem a esse requisito.

Consulte os tópicos a seguir para saber mais sobre cada tipo de balanceador de carga compatível com o Elastic Beanstalk, sua funcionalidade, como configurá-lo e gerenciá-lo em um ambiente do Elastic Balanceador e como configurar um balanceador de carga para [fazer upload de logs de acesso](#) para o Amazon S3.

#### Tópicos

- [Configuração de um Classic Load Balancer](#)
- [Configuração de um Application Load Balancer](#)
- [Configurar um Application Load Balancer compartilhado](#)
- [Configurar um Network Load Balancer](#)
- [Configurar logs de acesso](#)

## Configuração de um Classic Load Balancer

Quando você [habilita o balanceamento de carga](#), o ambiente do AWS Elastic Beanstalk é equipado com um balanceador de carga do Elastic Load Balancing para distribuir o tráfego entre as instâncias do ambiente. O Elastic Load Balancing é compatível com vários tipos de balanceador de carga. Para saber mais sobre eles, consulte o [Guia do usuário do Elastic Load Balancing](#). O Elastic Beanstalk pode criar um balanceador de carga para você ou permitir a especificação de um balanceador de carga compartilhado que você criou.

Este tópico descreve a configuração de um [Classic Load Balancer](#) que o Elastic Beanstalk cria e dedica ao seu ambiente. Para obter informações sobre como configurar todos os tipos de balanceador de carga compatíveis com o Elastic Beanstalk, consulte [Balanceador de carga do ambiente do Elastic Beanstalk](#).

**Note**

Você só pode escolher o tipo de load balancer que seu ambiente usará durante a criação do ambiente. Posteriormente, você pode alterar as configurações para gerenciar o comportamento do load balancer do ambiente em execução, mas não pode alterar seu tipo.

## Introdução

Um [Classic Load Balancer](#) é o balanceador de carga da geração anterior do Elastic Load Balancing. Oferece suporte ao roteamento de tráfego de solicitação HTTP, HTTPS ou TCP para portas diferentes em instâncias de ambiente.

Quando seu ambiente usa um Classic Load Balancer, o Elastic Beanstalk o configura por padrão para [ouvir](#) o tráfego HTTP na porta 80 e encaminhá-lo para instâncias na mesma porta. Embora você não possa excluir o ouvinte padrão da porta 80, você pode desativá-lo, o que obtém a mesma funcionalidade ao bloquear o tráfego. Observe que você pode adicionar ou excluir outros ouvintes. Para oferecer suporte a conexões seguras, configure o load balancer com um listener na porta 443 e um certificado TLS.

O balanceador de carga usa uma [verificação de integridade](#) para determinar se as instâncias do Amazon EC2 que executam sua aplicação estão íntegras. A verificação de integridade faz uma solicitação para um URL especificado em um intervalo de tempo definido. Se o URL retornar uma mensagem de erro ou não retornar dentro do tempo limite especificado, essa verificação falhará.

Se a aplicação tiver melhor performance atendendo a várias solicitações do mesmo cliente em um único servidor, configure o balanceador de carga para usar [sticky sessions](#). Com as sticky sessions, o balanceador de carga adiciona um cookie às respostas HTTP que identifica a instância do Amazon EC2 que atendeu à solicitação. Quando uma solicitação subsequente é recebida do mesmo cliente, o load balancer usa o cookie para enviar a solicitação para a mesma instância.

Com o [balanceamento de carga entre zonas](#), cada nó do balanceador de carga do seu Classic Load Balancer distribui solicitações uniformemente a todas as instâncias registradas em todas as zonas de disponibilidade habilitadas. Se o balanceamento de carga entre zonas estiver desabilitado, cada nó do balanceador de carga distribuirá solicitações uniformemente às instâncias registradas somente em sua zona de disponibilidade.

Quando uma instância é removida do balanceador de carga por não ser íntegra ou por redução do ambiente, a [diminuição de conexão](#) dá à instância tempo de concluir as solicitações antes de fechar



a conexão entre a instância e o balanceador de carga. Você pode alterar o tempo concedido às instâncias para enviar uma resposta ou pode desabilitar completamente a drenagem de conexão.

#### Note

A diminuição de conexão é habilitada por padrão quando um ambiente é criado com o console do Elastic Beanstalk ou a CLI do EB. Para os outros clientes, é possível ativá-la com [opções de configuração](#).

Você pode usar as configurações avançadas do load balancer para configurar listeners em portas arbitrárias, modificar configurações adicionais de sticky session e configurar o load balancer para conectar-se com segurança a instâncias do EC2. Essas configurações estão disponíveis por meio das [opções de configuração](#) que podem ser definidas com os arquivos de configuração no código-fonte ou diretamente em um ambiente usando a API do Elastic Beanstalk. Muitas dessas configurações também estão disponíveis no console do Elastic Beanstalk. Além disso, é possível configurar um balanceador de carga para [fazer upload de logs de acesso](#) no Amazon S3.

## Configurar um Classic Load Balancer usando o console do Elastic Beanstalk

É possível usar o console do Elastic Beanstalk para configurar portas, certificado HTTPS e outras configurações de um Classic Load Balancer durante a criação do ambiente ou mais tarde, quando o ambiente estiver em execução.

#### Note

A opção Classic Load Balancer (CLB) está desabilitada no assistente do console Criar ambiente. Se você tiver um ambiente existente configurado com um Classic Load Balancer, poderá criar um novo [clonando o ambiente existente](#) usando o console do Elastic Beanstalk ou a [CLI do EB](#). Você também tem a opção de usar a [CLI do EB](#) ou a [AWS CLI](#) para criar um novo ambiente configurado com um Classic Load Balancer. Essas ferramentas de linha de comando criarão um novo ambiente com um CLB mesmo que ainda não exista um em sua conta.

## Como configurar o Classic Load Balancer de um ambiente em execução no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Load balancer, escolha Edit (Editar).

### Note

Se a categoria de configuração Load balancer não tem um botão Edit (Editar), seu ambiente não tem um load balancer. Para aprender como configurar um, consulte [Alterar tipo de ambiente](#).

5. Faça as alterações de configuração do Classic Load Balancer necessárias ao seu ambiente.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

## Configurações do Classic Load Balancer

- [Listeners](#)
- [Sessões](#)
- [Balanceamento de carga entre zonas](#)
- [Drenagem de conexão](#)
- [Verificação de integridade](#)

## Listeners

Use esta lista para especificar listeners para seu load balancer. Cada listener direciona o tráfego de entrada do cliente em uma porta especificada usando um protocolo especificado para suas instâncias. Inicialmente, a lista mostra o listener padrão, que roteia o tráfego HTTP de entrada na

porta 80 para os servidores de instância do seu ambiente que estão observando o tráfego HTTP na porta 80.

### Note

Embora você não possa excluir o ouvinte padrão da porta 80, você pode desativá-lo, o que obtém a mesma funcionalidade ao bloquear o tráfego.

#### Classic Load Balancer

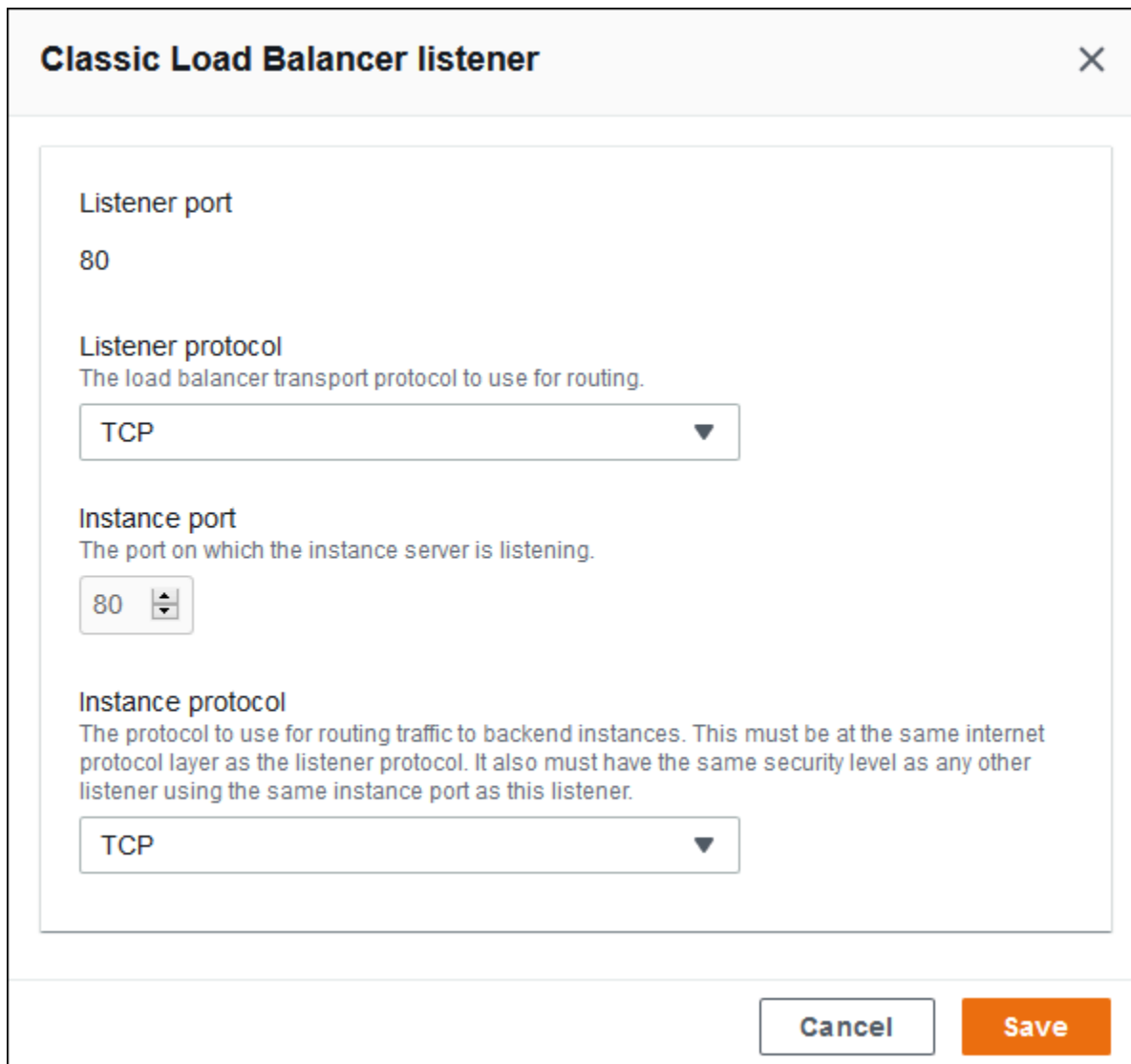
You can specify listeners for your load balancer. Each listener routes incoming client traffic on a specified port using a specified protocol to your instances. By default, we've configured your load balancer with a standard web server on port 80.

<input type="checkbox"/>	Port	Protocol	Instance port	Instance protocol	SSL certificate	Enabled
<input type="checkbox"/>	80	HTTP	80	HTTP	--	<input checked="" type="checkbox"/>

Para configurar um listener existente

1. Marque a caixa de seleção ao lado da entrada da tabela, escolha Actions, e depois escolha a ação desejada.
2. Se escolher Edit, use a caixa de diálogo Classic Load Balancer listener para editar as configurações e, em seguida, escolha Save.

Por exemplo, você pode editar o listener padrão e alterar Protocol de HTTP para TCP se você quiser que o load balancer encaminhe uma solicitação como está. Isso impede que o load balancer reescreva cabeçalhos (incluindo X-Forwarded-For). A técnica não funciona com sticky sessions.



The image shows a dialog box titled "Classic Load Balancer listener" with a close button (X) in the top right corner. The dialog contains the following configuration fields:

- Listener port:** A text input field containing the value "80".
- Listener protocol:** A dropdown menu with the text "The load balancer transport protocol to use for routing." and the value "TCP" selected.
- Instance port:** A spinner control with the value "80" and up/down arrows.
- Instance protocol:** A dropdown menu with the text "The protocol to use for routing traffic to backend instances. This must be at the same internet protocol layer as the listener protocol. It also must have the same security level as any other listener using the same instance port as this listener." and the value "TCP" selected.

At the bottom right of the dialog, there are two buttons: "Cancel" and "Save".

Para adicionar um listener

1. Escolha Add listener.
2. Na caixa de diálogo Classic Load Balancer listener, defina as configurações desejadas e, em seguida, escolha Add.

Adicionar um listener seguro é um caso de uso comum. O exemplo na imagem a seguir adiciona um listener ao tráfego HTTPS na porta 443. Esse listener direciona o tráfego de entrada para os servidores de instância do ambiente que observam o tráfego HTTPS na porta 443.

Antes de configurar um listener HTTPS, verifique se você tem um certificado SSL válido. Execute um destes procedimentos:

- Se o AWS Certificate Manager (ACM) estiver [disponível na sua região da AWS](#), crie ou importe um certificado usando o ACM. Para obter mais informações sobre como solicitar um certificado do ACM, consulte [Solicitação de um certificado](#) no Manual do usuário do AWS Certificate Manager. Para obter mais informações sobre a importação de certificados de terceiros para o ACM, consulte [Importação de certificados](#) no Manual do usuário do AWS Certificate Manager.
- Se o ACM não estiver [disponível em sua região da AWS](#), carregue o certificado e a chave existente para o IAM. Para obter mais informações sobre a criação e o upload de certificados para o IAM, consulte [Trabalhar com certificados de servidor](#) no Guia do usuário do IAM.

Para obter mais detalhes sobre como configurar o HTTPS e trabalhar com os certificados no Elastic Beanstalk, consulte [Configurar HTTPS para o seu ambiente do Elastic Beanstalk](#).

Para SSL certificate, escolha o ARN do seu certificado SSL. Por exemplo,  
arn:aws:iam::123456789012:server-certificate/abc/certs/build ou  
arn:aws:acm:us-  
east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678.


### Classic Load Balancer listener ✕

**Listener port**  
443

**Listener protocol**  
The load balancer transport protocol to use for routing.  
HTTPS

**Instance port**  
The port on which the instance server is listening.  
443

**Instance protocol**  
The protocol to use for routing traffic to backend instances. This must be at the same internet protocol layer as the listener protocol. It also must have the same security level as any other listener using the same instance port as this listener.  
HTTPS

**SSL certificate**  
arn:aws:acm:us-east-2:123456789012:certific... 

Para obter detalhes sobre como configurar o HTTPS e trabalhar com os certificados no Elastic Beanstalk, consulte [Configurar HTTPS para o seu ambiente do Elastic Beanstalk](#).

## Sessões

Selecione ou desmarque a caixa **Perdurabilidade da sessão** para habilitar ou desabilitar as sticky sessions. Use **Duração do cookie** para configurar a duração de uma sticky session até **1000000** segundos. Na lista **Load balancer ports (Portas do load balancer)**, selecione portas do listener às quais a política padrão (**AWSEB-ELB-StickinessPolicy**) se aplica.

## Sessions

The following settings let you control whether the load balancer routes requests for the same session to the Amazon EC2 instance with the smallest load, or consistently to the same instance.

Session stickiness enabled

### Cookie duration

Lifetime of the sticky session cookie between an Amazon EC2 instance and the load balancer.

0 seconds

### Load balancer ports

List of the listener ports that the default policy (AWSEB-ELB-StickinessPolicy) applies to.

Choose load balancer ports

80
443

## Balanceamento de carga entre zonas

Selecione ou desmarque a caixa Load balancing across multiple Availability Zones enabled para habilitar ou desabilitar o balanceamento de carga entre zonas.

## Cross-zone load balancing

Load balancing across multiple Availability Zones enabled

## Drenagem de conexão

Selecione ou desmarque a caixa Diminuição de conexão habilitada para habilitar ou desabilitar a diminuição de conexão. Defina Tempo limite de drenagem até **3600** segundos.

## Connection draining

Connection draining enabled

### Draining timeout

Maximum time that the load balancer maintains connections to an Amazon EC2 instance before forcibly closing connections.

20 seconds

## Verificação de integridade

Use as configurações a seguir para configurar as verificações de integridade do load balancer:

- **Health check path (Caminho da verificação de integridade):** o caminho para o qual o balanceador de carga envia solicitações de verificação de integridade. Se você não definir o caminho, o load balancer tentará fazer uma conexão TCP na porta 80 para verificar a integridade.
- **Timeout (Tempo limite):** a quantidade de tempo, em segundos, para aguardar uma resposta de verificação de integridade.
- **Interval (Intervalo):** a quantidade de tempo, em segundos, entre as verificações de integridade de uma instância individual. O intervalo deve ser maior que o tempo limite.
- **Unhealthy threshold (Limite não íntegro), Healthy threshold (Limite íntegro):** o número de verificações de integridade que devem falhar ou passar, respectivamente, antes que o Elastic Load Balancing altere o estado de integridade de uma instância.

### Health check

**Health check path**  
Path to which ELB sends an HTTP GET request to verify instance health.

  
**Timeout**  
Amount of time to wait for a health check response.

5 seconds

**Interval**  
Amount of time between health checks of an individual instance. The interval must be greater than the timeout.

10 seconds

**Unhealthy threshold**  
The number of consecutive health check failures required to designate the instance as unhealthy.

5 requests

**Healthy threshold**  
The number of consecutive successful health checks required to designate the instance as healthy.

3 requests

Cancel Save



**Note**

A verificação de integridade do Elastic Load Balancing não afeta o comportamento da verificação de integridade do grupo de Auto Scaling de um ambiente. As instâncias que falham em uma verificação de integridade do Elastic Load Balancing não são substituídas automaticamente pelo Amazon EC2 Auto Scaling, a menos que você configure manualmente o Amazon EC2 Auto Scaling para fazer isso. Para mais detalhes, consulte [Configuração da verificação de integridade do Auto Scaling](#).

Para obter mais informações sobre verificações de integridade e como elas influenciam a integridade geral do ambiente, consulte [Relatórios de integridade básica](#).

## Configurar um Classic Load Balancer usando a CLI do EB

A EB CLI solicita que você escolha um tipo de load balancer ao executar o [eb create](#).

```
$ eb create
Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
1) classic
2) application
3) network
(default is 1):
```

Pressione Enter para selecionar `classic`.

Você também pode especificar um tipo de load balancer usando a opção `--elb-type`.

```
$ eb create test-env --elb-type classic
```

## Namespaces de configuração do Classic Load Balancer

É possível encontrar configurações relacionadas aos Classic Load Balancers nos seguintes namespaces:

- [aws:elb:healthcheck](#): configure os limites, o intervalo de verificação e o tempo limite das verificações de integridade do balanceador de carga.
- [aws:elasticbeanstalk:application](#): configure o URL da verificação de integridade.
- [aws:elb:loadbalancer](#): habilitar balanceamento de carga entre zonas. Atribua security groups ao load balancer e substitua o security group padrão que o Elastic Beanstalk cria. Esse namespace também inclui opções obsoletas de configuração de ouvintes padrão e seguros que foram substituídas pelas opções no namespace `aws:elb:listener`.
- [aws:elb:listener](#): configure o listener padrão na porta 80, um listener seguro na porta 443 ou outros listeners para qualquer protocolo em qualquer porta. Se você especificar `aws:elb:listener` como o namespace, as configurações se aplicarão ao listener padrão na porta 80. Se você especificar uma porta (por exemplo, `aws:elb:listener:443`), um listener será configurado nessa porta.
- [aws:elb:policies](#): defina configurações adicionais para o balanceador de carga. Use as opções desse namespace para configurar listeners em portas arbitrárias, modificar configurações adicionais de sticky session e configurar o balanceador de carga para conectar-se com segurança a instâncias do Amazon EC2.

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte [Valores recomendados](#).

Example `.ebextensions/loadbalancer-terminatehttps.config`

O exemplo de arquivo de configuração a seguir cria um listener HTTPS na porta 443, atribui um certificado que o load balancer usa para encerrar a conexão segura e desabilita o listener padrão na porta 80. O load balancer encaminha as solicitações descritografadas para as instâncias do EC2 de seu ambiente em HTTP: 80.

```
option_settings:
 aws:elb:listener:443:
 ListenerProtocol: HTTPS
 SSLCertificateId: arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678
 InstancePort: 80
 InstanceProtocol: HTTP
 aws:elb:listener:
 ListenerEnabled: false
```

## Configuração de um Application Load Balancer

Quando você [habilita o balanceamento de carga](#), o ambiente do AWS Elastic Beanstalk é equipado com um balanceador de carga do Elastic Load Balancing para distribuir o tráfego entre as instâncias do ambiente. O Elastic Load Balancing é compatível com vários tipos de balanceador de carga. Para saber mais sobre eles, consulte o [Guia do usuário do Elastic Load Balancing](#). O Elastic Beanstalk pode criar um balanceador de carga para você ou permitir a especificação de um balanceador de carga compartilhado que você criou.

Este tópico descreve a configuração de um [Application Load Balancer](#) que o Elastic Beanstalk cria e dedica ao seu ambiente. Consulte também [the section called “Application Load Balancer compartilhado”](#). Para obter informações sobre como configurar todos os tipos de balanceador de carga compatíveis com o Elastic Beanstalk, consulte [the section called “Load balancer”](#).

### Note

Você só pode escolher o tipo de load balancer que seu ambiente usará durante a criação do ambiente. Você pode alterar as configurações para gerenciar o comportamento do load balancer do ambiente em execução, mas não pode alterar seu tipo. Também não é possível alternar de um load balancer dedicado para um compartilhado ou vice-versa.

## Introdução

Um Application Load Balancer inspeciona o tráfego na camada de protocolo de rede da aplicação para identificar o caminho do pedido de modo que ele possa direcionar solicitações de caminhos diferentes para destinos diferentes.

Quando seu ambiente usa um Application Load Balancer, o Elastic Beanstalk o configura por padrão para executar a mesma função que um Classic Load Balancer. O listener padrão aceita solicitações de HTTP na porta 80 e as distribui para as instâncias no ambiente. É possível adicionar um listener seguro na porta 443 com um certificado para descriptografar o tráfego de HTTPS, configurar o comportamento de verificação de integridade e enviar os registros de acesso do balanceador de carga para um bucket do Amazon Simple Storage Service (Amazon S3).

### Note

Ao contrário de um Classic Load Balancer ou um Network Load Balancer, um Application Load Balancer não pode ter listeners TCP ou SSL/TLS de camada de transporte (camada 4).

Suporta apenas listeners HTTP e HTTPS. Além disso, não é possível usar a autenticação de back-end para autenticar conexões HTTPS entre o load balancer e as instâncias de back-end.

Em um ambiente do Elastic Beanstalk, é possível usar um Application Load Balancer para direcionar o tráfego de determinados caminhos para um processo diferente em suas instâncias do servidor Web. Com um Classic Load Balancer, todo o tráfego para o listener é roteado para um único processo nas instâncias de back-end. Com um Application Load Balancer, é possível configurar várias regras no listener para rotear as solicitações para determinados caminhos para um processo de back-end diferente. Configure cada processo com a porta na qual o processo escuta.

Por exemplo, é possível executar um processo de login separadamente do aplicativo principal. Embora o aplicativo principal nas instâncias do seu ambiente aceite a maioria das solicitações e observe a porta 80, seu processo de login observará a porta 5000 e aceitará solicitações para o caminho `/login`. Todas as solicitações de entrada dos clientes vêm na porta 80. Com um Application Load Balancer, é possível configurar um único listener para o tráfego de entrada na porta 80, com duas regras que direcionam o tráfego para dois processos separados, dependendo do caminho na solicitação. Adicione uma regra personalizada que direcione o tráfego para `/login` para o processo de logon que escuta na porta 5000. A regra padrão encaminha todos os outros tráfegos para o processo principal do aplicativo, observando a porta 80.

Uma regra do Application Load Balancer mapeia uma solicitação para um grupo de destino. No Elastic Beanstalk, um grupo de destino é representado por um processo. Você pode configurar um processo com configurações de protocolo, porta e verificação de integridade. O processo representa a execução nas instâncias em seu ambiente. O processo padrão é um listener na porta 80 do proxy reverso (nginx ou Apache) que é executado na frente de seu aplicativo.

#### Note

Fora do Elastic Beanstalk, um grupo de destino é mapeado para um grupo de instâncias. Um listener pode usar regras e grupos de destino para rotear o tráfego para instâncias diferentes com base no caminho. Dentro do Elastic Beanstalk, todas as instâncias no ambiente são idênticas, então a distinção é feita entre os processos escutando em portas diferentes.

Um Classic Load Balancer usa um único caminho de verificação de integridade para todo o ambiente. Com um Application Load Balancer, cada processo tem um caminho de verificação

de integridade separado que é monitorado pelo balanceador de carga e pelo monitoramento de integridade aprimorado do Elastic Beanstalk.

Para usar o Application Load Balancer, seu ambiente deve estar em uma VPC personalizada ou padrão e deve ter uma função de serviço com o conjunto padrão de permissões.

Se você tiver uma função de serviço mais antiga, talvez seja necessário [atualizar as permissões](#) dela para incluir `elasticloadbalancing:DescribeTargetHealth` e `elasticloadbalancing:DescribeLoadBalancers`. Para obter mais informações sobre os Application Load Balancers, consulte [O que é Application Load Balancer?](#).

#### Note

A verificação de integridade do Application Load Balancer não usa o caminho da verificação de integridade do Elastic Beanstalk. Em vez disso, ele usa o caminho específico configurado para cada processo separadamente.

## Configurar um Application Load Balancer usando o console do Elastic Beanstalk

É possível usar o console do Elastic Beanstalk para configurar os listeners, processos e regras de um Application Load Balancer durante a criação do ambiente ou posterior quando seu ambiente estiver em execução.

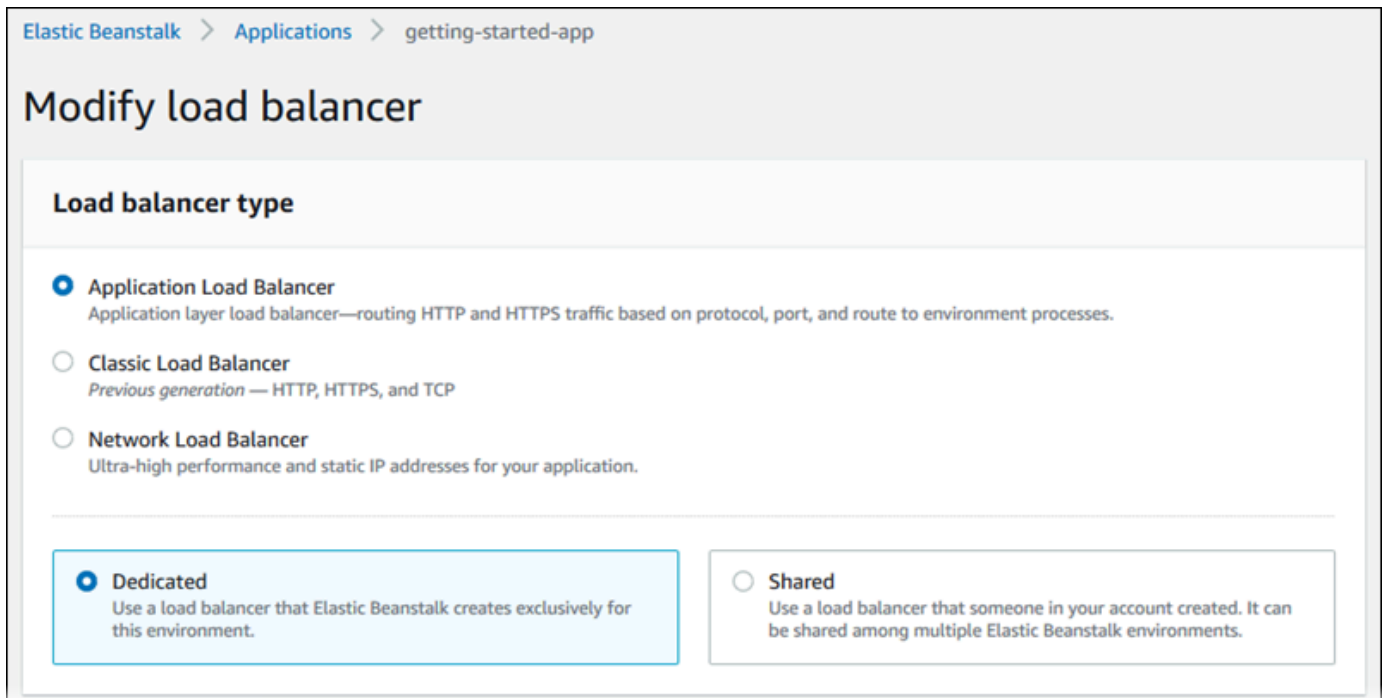
Como configurar um Application Load Balancer no console do Elastic Beanstalk durante a criação do ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, escolha Environments (Ambientes).
3. Escolha [Create a new environment \(Criar um ambiente\)](#) para começar a criar seu ambiente.
4. Na página principal do assistente, antes de escolher Create environment, escolha Configure more options.
5. Escolha a predefinição de configuração High availability (Alta disponibilidade).

Se preferir, na categoria de configuração Capacity (Capacidade), configure o tipo de ambiente Load balanced (Carga equilibrada). Para obter mais detalhes, consulte [Capacity](#).

6. Na categoria de configuração Load balancer, escolha Edit (Editar).

7. Selecione as opções Application Load Balancer e Dedicated (Dedicado) se elas ainda não estiverem selecionadas.



8. Faça todas as alterações de configuração do Application Load Balancer necessárias ao seu ambiente.
9. Escolha Save e, em seguida, faça as outras alterações de configuração de acordo com seu ambiente.
10. Selecione Create environment (Criar ambiente).

Como configurar o Application Load Balancer de um ambiente em execução no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

**Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Load balancer, escolha Edit (Editar).

**Note**

Se a categoria de configuração Load balancer não tem um botão Edit (Editar), seu ambiente não tem um load balancer. Para aprender como configurar um, consulte [Alterar tipo de ambiente](#).

5. Faça as alterações de configuração do Application Load Balancer necessárias ao seu ambiente.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

## Configurações do Application Load Balancer

- [Listeners](#)
- [Processos](#)
- [Regras](#)
- [Captura de log de acesso](#)

### Listeners

Use esta lista para especificar listeners para seu load balancer. Cada listener direciona o tráfego de entrada do cliente em uma porta especificada usando um protocolo especificado para um ou mais processos nas suas instâncias. Inicialmente, a lista mostra o listener padrão, que encaminha o tráfego HTTP de entrada na porta 80 para um processo chamado padrão.

**Listeners**

You can specify listeners for your load balancer. Each listener routes incoming client traffic on a specified port using a specified protocol to your environment processes. By default, we've configured your load balancer with a standard web server on port 80.

Actions ▾ + Add listener

<input type="checkbox"/>	Port	Protocol	SSL certificate	Default process	Enabled
<input type="checkbox"/>	80	HTTP	--	default	<input checked="" type="checkbox"/>

### Para configurar um listener existente

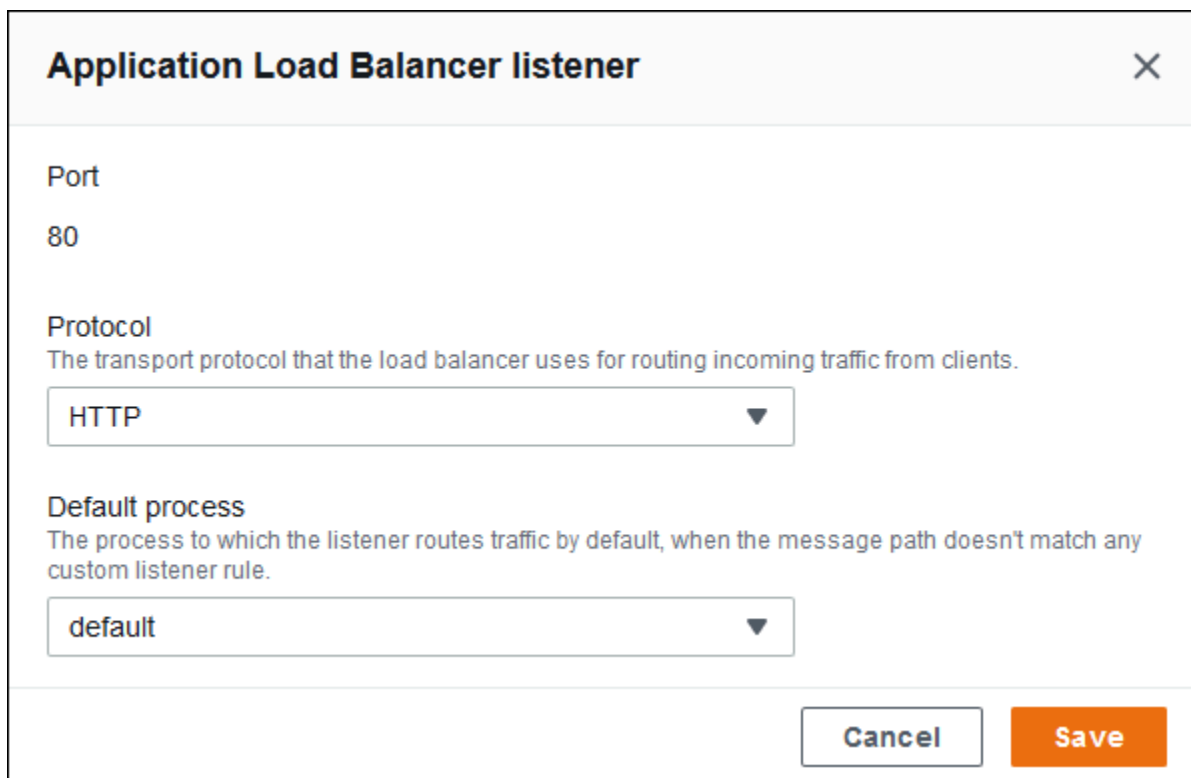
1. Marque a caixa de seleção ao lado da entrada da tabela e, em seguida, escolha Actions, Edit.

2. Use a caixa de diálogo Application Load Balancer listener para editar as configurações e, em seguida, escolha Save.

Para adicionar um listener

1. Escolha Add listener.
2. Na caixa de diálogo Application Load Balancer listener (Listener do Application Load Balancer), defina as configurações desejadas e selecione Add (Adicionar).

Use as configurações da caixa de diálogo Listener do Application Load Balancer para escolher a porta e o protocolo nos quais o ouvinte escuta o tráfego, bem como o processo para encaminhar o tráfego. Se você escolher o protocolo HTTPS, defina as configurações de SSL.



The screenshot shows a dialog box titled "Application Load Balancer listener" with a close button (X) in the top right corner. The dialog contains the following fields:

- Port:** A text input field containing the value "80".
- Protocol:** A dropdown menu with the text "The transport protocol that the load balancer uses for routing incoming traffic from clients." below it. The selected option is "HTTP".
- Default process:** A dropdown menu with the text "The process to which the listener routes traffic by default, when the message path doesn't match any custom listener rule." below it. The selected option is "default".

At the bottom right of the dialog, there are two buttons: "Cancel" and "Save".

Antes de configurar um listener HTTPS, verifique se você tem um certificado SSL válido. Faça um dos seguintes procedimentos:

- Se o AWS Certificate Manager (ACM) estiver [disponível na sua região da AWS](#), crie ou importe um certificado usando o ACM. Para obter mais informações sobre como solicitar um certificado do ACM, consulte [Solicitação de um certificado](#) no Manual do usuário do AWS Certificate Manager.



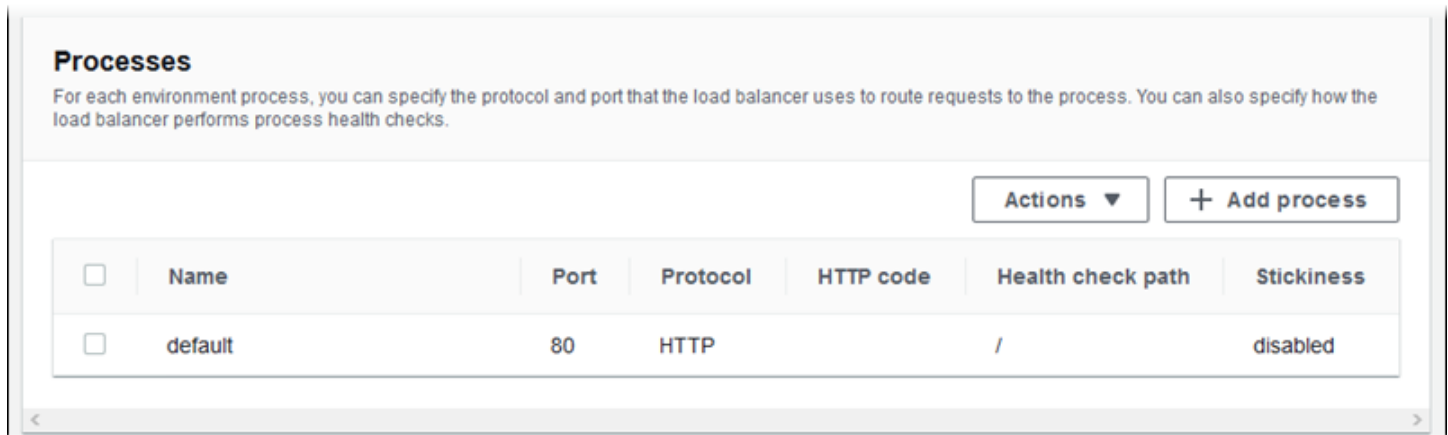
Para obter mais informações sobre a importação de certificados de terceiros para o ACM, consulte [Importação de certificados](#) no Manual do usuário do AWS Certificate Manager.

- Se o ACM não estiver [disponível em sua região da AWS](#), carregue o certificado e a chave existente para o IAM. Para obter mais informações sobre a criação e o upload de certificados para o IAM, consulte [Trabalhar com certificados de servidor](#) no Guia do usuário do IAM.

Para obter mais detalhes sobre como configurar o HTTPS e trabalhar com os certificados no Elastic Beanstalk, consulte [Configurar HTTPS para o seu ambiente do Elastic Beanstalk](#).

## Processos

Use esta lista para especificar processos para seu load balancer. Um processo é um destino para os listeners direcionarem o tráfego. Cada listener direciona o tráfego de entrada do cliente em uma porta especificada usando um protocolo especificado para um ou mais processos nas suas instâncias. Inicialmente, a lista mostra o processo padrão, que observa o tráfego HTTP de entrada na porta 80.



**Processes**

For each environment process, you can specify the protocol and port that the load balancer uses to route requests to the process. You can also specify how the load balancer performs process health checks.

Actions ▾ + Add process

<input type="checkbox"/>	Name	Port	Protocol	HTTP code	Health check path	Stickiness
<input type="checkbox"/>	default	80	HTTP		/	disabled

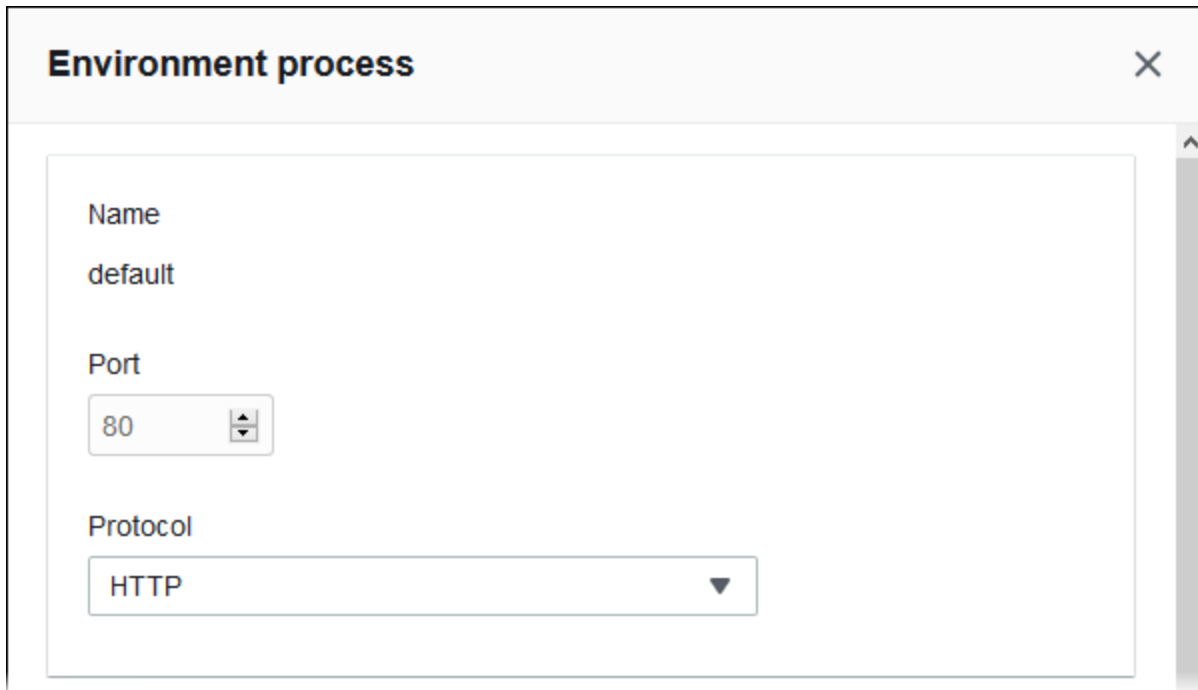
Você pode editar as configurações de um processo existente ou adicionar um novo processo. Para começar a editar um processo na lista ou adicionar um processo a ele, use as mesmas etapas listadas para [lista de listener](#). A caixa de diálogo Environment process é exibida.

Configurações da caixa de diálogo Environment process (Processo do ambiente) do Application Load Balancer

- [Definição](#)
- [Verificação de integridade](#)
- [Sessões](#)

## Definição

Use estas configurações para definir o processo Name, Port e Protocol em que as solicitações são observadas.



The screenshot shows a dialog box titled "Environment process" with a close button (X) in the top right corner. Inside the dialog, there are three configuration fields:

- Name:** A text input field containing the value "default".
- Port:** A numeric input field containing the value "80".
- Protocol:** A dropdown menu with "HTTP" selected.

## Verificação de integridade

Use as configurações a seguir para configurar as verificações de integridade do processo:

- **HTTP code (Código de HTTP):** o código de status HTTP que designa um processo íntegro.
- **Path (Caminho):** o caminho da solicitação de verificação de integridade para o processo.
- **Timeout (Tempo limite):** a quantidade de tempo, em segundos, para aguardar uma resposta de verificação de integridade.
- **Interval (Intervalo):** a quantidade de tempo, em segundos, entre as verificações de integridade de uma instância individual. O intervalo deve ser maior que o tempo limite.
- **Unhealthy threshold (Limite não íntegro), Healthy threshold (Limite íntegro):** o número de verificações de integridade que devem falhar ou passar, respectivamente, antes que o Elastic Load Balancing altere o estado de integridade de uma instância.
- **Deregistration delay (Atraso do cancelamento de registro):** a quantidade de tempo, em segundos, para aguardar que as solicitações ativas sejam concluídas antes de cancelar o registro de uma instância.

## Health check

### HTTP code

HTTP status code of a healthy instance in your environment.

### Path

Path to which the load balancer sends HTTP health check requests.

### Timeout

Amount of time to wait for a health check response.

 seconds

### Interval

Amount of time between health checks of an individual instance. The interval must be greater than the timeout.

 seconds

### Unhealthy threshold

The number of consecutive health check failures required to designate the instance as unhealthy.

 requests

### Healthy threshold

The number of consecutive successful health checks required to designate the instance as healthy.

 requests

### Deregistration delay

Amount of time to wait for active requests to complete before deregistering.

 seconds

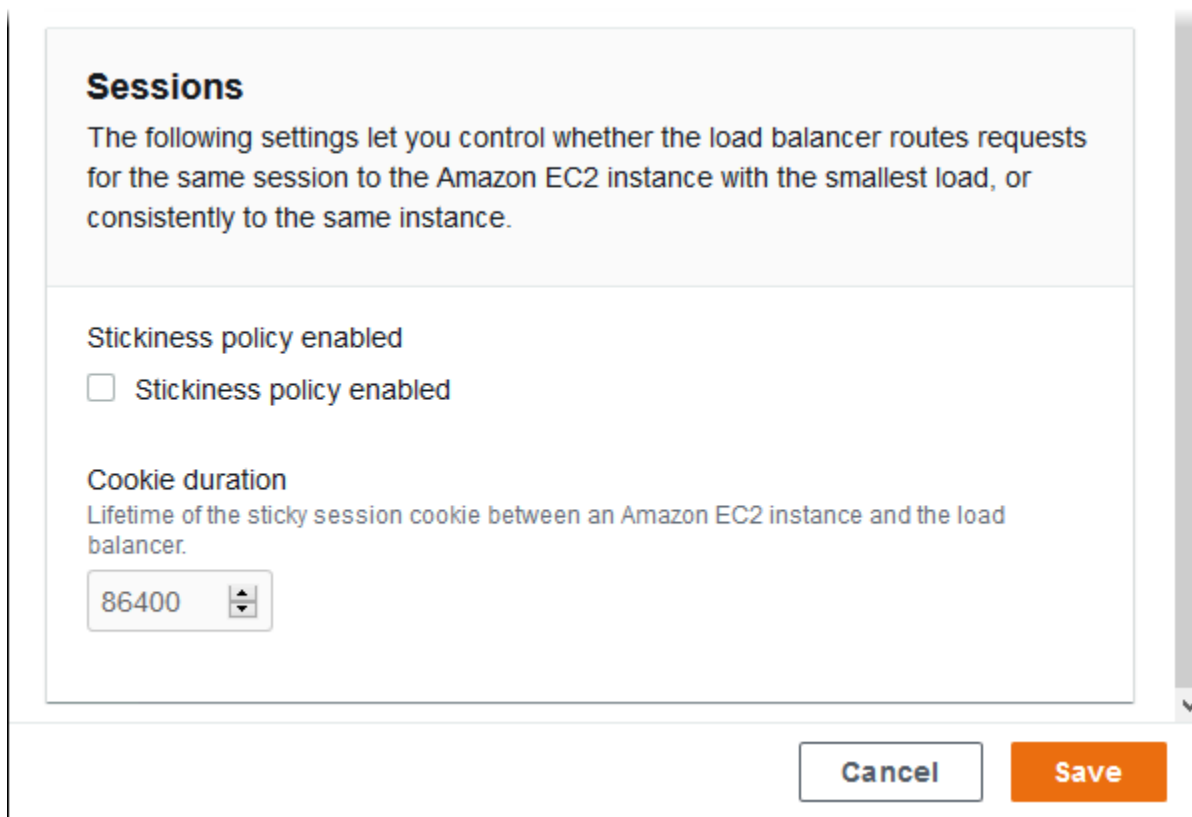
**Note**

A verificação de integridade do Elastic Load Balancing não afeta o comportamento da verificação de integridade do grupo de Auto Scaling de um ambiente. As instâncias que falham em uma verificação de integridade do Elastic Load Balancing não são substituídas automaticamente pelo Amazon EC2 Auto Scaling, a menos que você configure manualmente o Amazon EC2 Auto Scaling para fazer isso. Para mais detalhes, consulte [Configuração da verificação de integridade do Auto Scaling](#).

Para obter mais informações sobre verificações de integridade e como elas influenciam a integridade geral do ambiente, consulte [Relatórios de integridade básica](#).

**Sessões**

Selecione ou desmarque a caixa Política de perdurabilidade habilitada para habilitar ou desabilitar as sticky sessions. Use Duração do cookie para configurar a duração de uma sticky session até **604800** segundos.



**Sessions**

The following settings let you control whether the load balancer routes requests for the same session to the Amazon EC2 instance with the smallest load, or consistently to the same instance.

Stickiness policy enabled

Stickiness policy enabled

Cookie duration

Lifetime of the sticky session cookie between an Amazon EC2 instance and the load balancer.

86400

Cancel Save

## Regras

Use esta lista para especificar regras de listener para seu load balancer. Uma regra mapeia solicitações que o listener recebe em um padrão de caminho específico para um processo de destino. Cada listener pode ter várias regras, roteando solicitações em diferentes caminhos para diferentes processos nas suas instâncias.

As regras têm prioridades numéricas que determinam a precedência em que são aplicadas às solicitações recebidas. Para cada novo listener adicionado, o Elastic Beanstalk adiciona uma regra padrão que direciona todo o tráfego do listener para o processo padrão. A precedência da regra padrão é a mais baixa. Ela é aplicada se nenhuma outra regra para o mesmo listener corresponder à solicitação recebida. Inicialmente, se você não tiver adicionado regras personalizadas, a lista estará vazia. As regras padrão de todos os listeners não são exibidas.

**Rules**

Your load balancer routes requests to environment processes based on rules. Rules are evaluated by priority in ascending numerical order. Elastic Beanstalk configures a default rule for each listener. Each default rule routes all traffic to the default process associated with each listener, and has the last priority among all rules of that listener. If a request doesn't match the conditions for any other rule, a default rule routes the request to the listener's default process.

Actions ▾ + Add rule

Name	Listener port	Priority	Host headers	Path patterns	Process
No additional listener rules are currently configured. Choose Add rule to add a listener rule.					

Você pode editar as configurações de uma regra existente ou adicionar uma nova regra. Para começar a editar uma regra na lista ou adicionar uma regra a ela, use as mesmas etapas listadas para [lista de listener](#). A caixa de diálogo Listener rule é exibida com as seguintes configurações:

- **Name (Nome):** o nome da regra.
- **Listener port (Porta do listener):** porta do listener ao qual a regra se aplica.
- **Priority (Prioridade):** prioridade da regra. Um número de prioridade mais baixa tem maior precedência. As prioridades das regras de um listener devem ser únicas.
- **Match conditions (Condições de correspondência):** uma lista de condições do URL de solicitação às quais a regra se aplica. Existem dois tipos de condições: HostHeader (a parte do domínio do

URL) e PathPattern (a parte do caminho do URL). É possível adicionar até cinco condições. Cada valor de condição tem até 128 caracteres de comprimento e pode incluir caracteres curinga.

- Process (Processo): o processo para o qual o balanceador de carga direciona solicitações que correspondem à regra.

Ao editar qualquer regra existente, você não pode alterar Name nem Listener port.

**Listener rule** [X]

**Name**  
images

**Listener port**  
80 ▼

**Priority**  
Evaluated in ascending numerical order. Must be unique across all rules.  
1 ▲▼

**Match conditions**  
A listener rule can have up to five match conditions.

Type	Value	
PathPattern ▼	/images/*	Remove

Add condition

**Process**  
images ▼

Cancel Add

## Captura de log de acesso

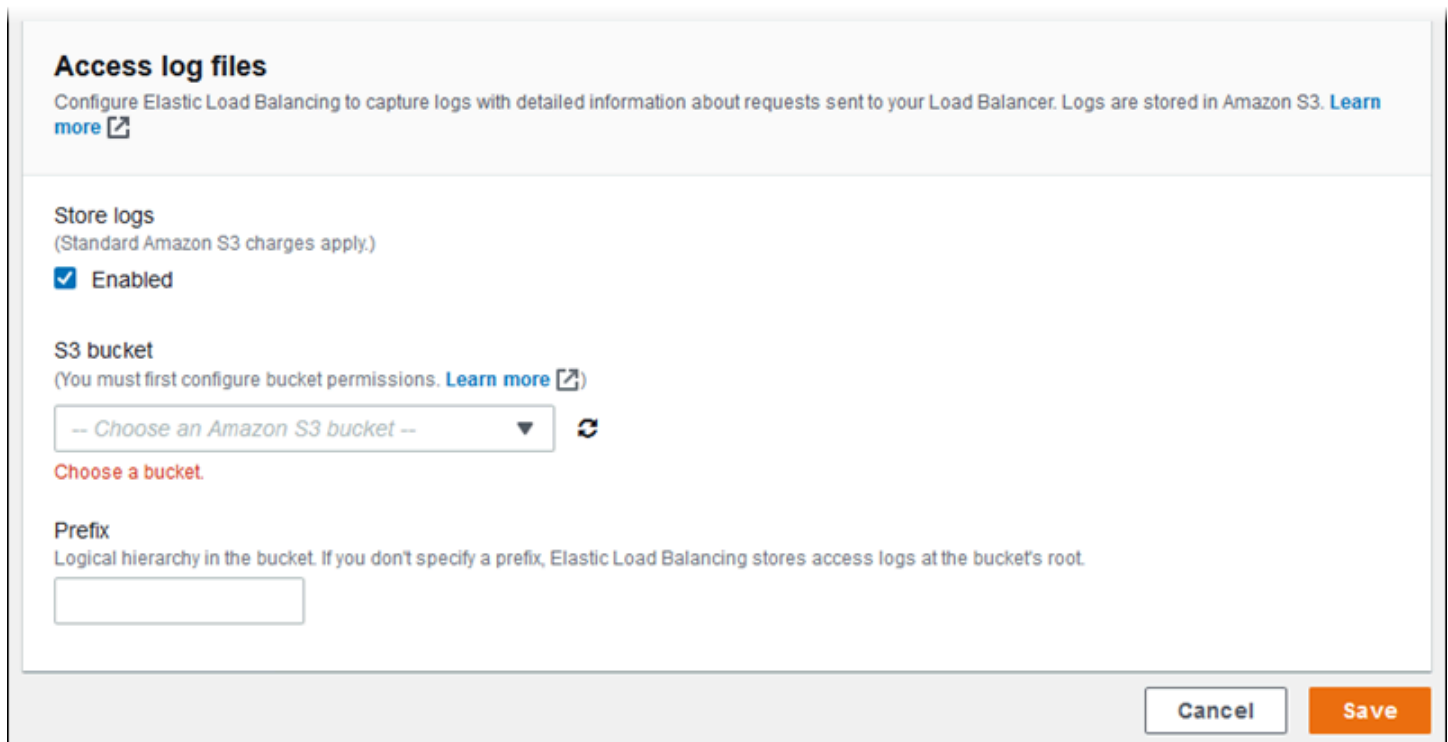
Use essas configurações para configurar o Elastic Load Balancing para capturar logs com informações detalhadas sobre solicitações enviadas ao Application Load Balancer. A captura de log

de acesso está desativada por padrão. Quando Store logs (Armazenar logs) está habilitado, o Elastic Load Balancing armazena os logs no S3 bucket (Bucket do S3) configurado. A configuração Prefix (Prefixo) especifica uma pasta de nível superior no bucket para os logs. O Elastic Load Balancing coloca os logs em uma pasta denominada AWSLogs sob seu prefixo. Se você não especificar um prefixo, o Elastic Load Balancing coloca sua pasta no nível raiz do bucket.

### Note

Se o bucket do Amazon S3 que você configura para a captura de log de acesso não for o bucket criado pelo Elastic Beanstalk para a conta, certifique-se de adicionar uma política de usuário com as permissões apropriadas aos usuários da AWS Identity and Access Management (IAM). As [políticas de usuário gerenciadas](#) que o Elastic Beanstalk fornece apenas permissões de cobertura para recursos gerenciados pelo Elastic Beanstalk.

Para obter detalhes sobre logs de acesso, incluindo permissões e outros requisitos, consulte [Registros de acesso para o Application Load Balancer](#).



**Access log files**  
Configure Elastic Load Balancing to capture logs with detailed information about requests sent to your Load Balancer. Logs are stored in Amazon S3. [Learn more](#)

**Store logs**  
(Standard Amazon S3 charges apply.)  
 Enabled

**S3 bucket**  
(You must first configure bucket permissions. [Learn more](#))  
-- Choose an Amazon S3 bucket --  
Choose a bucket.

**Prefix**  
Logical hierarchy in the bucket. If you don't specify a prefix, Elastic Load Balancing stores access logs at the bucket's root.

Cancel Save

## Exemplo: Application Load Balancer com um listener seguro e dois processos

Neste exemplo, seu aplicativo exige criptografia de tráfego completa e um processo separado para lidar com solicitações administrativas.

Para configurar o Application Load Balancer do ambiente de modo a atender esses requisitos, remova o listener padrão, adicione um listener HTTPS, indique que o processo padrão escuta a porta 443 em HTTPS e adicione um processo e uma regra de listener ao tráfego administrativo em um caminho diferente.

Para configurar o load balancer neste exemplo

1. Adicione um listener seguro. Em Porta, digite **443**. Em Protocolo, selecione **HTTPS**. Em Certificado SSL, escolha o ARN do seu certificado SSL. Por exemplo, **arn:aws:iam::123456789012:server-certificate/abc/certs/build** ou **arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678**.

Em Processo padrão, mantenha **default** selecionado.



### Application Load Balancer listener ✕

**Port**  
443

**Protocol**  
The transport protocol that the load balancer uses for routing incoming traffic from clients.  
HTTPS

**SSL certificate**  
arn:aws:acm:us-east-2:123456789012:certific...

**SSL policy**  
The Secure Sockets Layer (SSL) negotiation configuration, known as a security policy, that this load balancer uses to negotiate SSL connections with clients.  
ELBSecurityPolicy-2016-08

**Default process**  
The process to which the listener routes traffic by default, when the message path doesn't match any custom listener rule.  
default

Agora você pode ver seu listener adicional na lista.

<input type="checkbox"/>	Port	Protocol	SSL certificate	Default process	Enabled
<input type="checkbox"/>	80	HTTP	--	default	<input checked="" type="checkbox"/>
<input type="checkbox"/>	443	HTTPS	arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678	default	<input checked="" type="checkbox"/>

- Desative o listener HTTP da porta padrão 80. Para o listener padrão, desative a opção Habilitada.

<input type="checkbox"/>	Port	Protocol	SSL certificate	Default process	Enabled
<input type="checkbox"/>	80	HTTP	--	default	<input type="checkbox"/>
<input type="checkbox"/>	443	HTTPS	arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678	default	<input checked="" type="checkbox"/>

- Configure o processo padrão como HTTPS. Selecione o processo padrão e, em seguida, para Actions, escolha Edit. Em Porta, digite **443**. Em Protocolo, selecione **HTTPS**.

### Environment process

Name  
default

Port  
443

Protocol  
HTTPS

- Adicione um processo administrativo. Para Name (Nome), digite **admin**. Em Porta, digite **443**. Em Protocolo, selecione **HTTPS**. Em Verificação de integridade, para Caminho, digite **/admin**.

**Environment process** [X]

**Name**

admin

**Port**

443

**Protocol**

HTTPS

**Health check**

**HTTP code**  
HTTP status code of a healthy instance in your environment.

200

**Path**  
Path to which the load balancer sends HTTP health check requests.

/admin

5. Adicione uma regra para o tráfego de administrador. Para Name (Nome), digite **admin**. Para Listener port, digite **443**. Em Match conditions (Condições de correspondência), adicione um PathPattern com o valor **/admin/\***. Em Processo, selecione **admin**.

### Listener rule ✕

**Name**  
admin

**Listener port**  
443 ▼

**Priority**  
Evaluated in ascending numerical order. Must be unique across all rules.  
1 ▲▼

**Match conditions**  
A listener rule can have up to five match conditions.

Type	Value	
PathPattern ▼	/admin/*	Remove

Add condition

**Process**  
admin ▼

Cancel Add

## Configurar um Application Load Balancer usando a CLI do EB

A EB CLI solicita que você escolha um tipo de load balancer ao executar o [eb create](#).

```
$ eb create
Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
```

```
1) classic
2) application
3) network
(default is 2):
```

Você também pode especificar um tipo de load balancer com a opção `--elb-type`.

```
$ eb create test-env --elb-type application
```

## Namespaces do Application Load Balancer

É possível encontrar configurações relacionadas aos Application Load Balancers nos seguintes namespaces:

- [aws:elasticbeanstalk:environment](#): selecione o tipo de balanceador de carga para o ambiente. O valor de um Application Load Balancer é `application`.

Não é possível definir essa opção nos arquivos de configuração ([.Ebextensions](#)).

- [aws:elbv2:loadbalancer](#): defina logs de acesso e outras configurações que se aplicam ao Application Load Balancer como um todo.
- [aws:elbv2:listener](#): configure listeners no Application Load Balancer. Essas configurações são mapeadas para as configurações em `aws:elb:listener` para Classic Load Balancers.
- [aws:elbv2:listenerrule](#): configure as regras que roteiam o tráfego para os diferentes processos, dependendo do caminho da solicitação. As regras são exclusivas dos Application Load Balancers.
- [aws:elasticbeanstalk:environment:process](#): configure as verificações de integridade e especifique a porta e o protocolo para os processos que são executados nas instâncias de seu ambiente. A porta e as configurações de protocolo mapeiam a porta e as configurações de protocolo da instância em `aws:elb:listener` para um listener no Classic Load Balancer. As configurações de verificação de integridade são mapeadas para as configurações nos namespaces `aws:elb:healthcheck` e `aws:elasticbeanstalk:application`.

Example `.ebextensions/alb-access-logs.config`

O arquivo de configuração a seguir permite fazer upload do log de acesso para um ambiente com um Application Load Balancer.

```
option_settings:
```

```
aws:elbv2:loadbalancer:
 AccessLogsS3Bucket: DOC-EXAMPLE-BUCKET
 AccessLogsS3Enabled: 'true'
 AccessLogsS3Prefix: beanstalk-alb
```

### Example .ebextensions/alb-default-process.config

O arquivo de configuração a seguir modifica as configurações de verificação de integridade e de durabilidade no processo padrão.

```
option_settings:
 aws:elasticbeanstalk:environment:process:default:
 DeregistrationDelay: '20'
 HealthCheckInterval: '15'
 HealthCheckPath: /
 HealthCheckTimeout: '5'
 HealthyThresholdCount: '3'
 UnhealthyThresholdCount: '5'
 Port: '80'
 Protocol: HTTP
 StickinessEnabled: 'true'
 StickinessLBCookieDuration: '43200'
```

### Example .ebextensions/alb seguro-listener.config

O arquivo de configuração a seguir adiciona um listener seguro e um processo de correspondência na porta 443.

```
option_settings:
 aws:elbv2:listener:443:
 DefaultProcess: https
 ListenerEnabled: 'true'
 Protocol: HTTPS
 SSLCertificateArns: arn:aws:acm:us-
east-2:123456789012:certificate/21324896-0fa4-412b-bf6f-f362d6eb6dd7
 aws:elasticbeanstalk:environment:process:https:
 Port: '443'
 Protocol: HTTPS
```

## Example .ebextensions/alb-admin-rule.config

O arquivo de configuração a seguir adiciona um listener seguro com uma regra que roteia o tráfego com um caminho de solicitação de /admin para um processo chamado admin que escuta na porta 4.443.

```
option_settings:
 aws:elbv2:listener:443:
 DefaultProcess: https
 ListenerEnabled: 'true'
 Protocol: HTTPS
 Rules: admin
 SSLCertificateArns: arn:aws:acm:us-east-2:123456789012:certificate/21324896-0fa4-412b-bf6f-f362d6eb6dd7
 aws:elasticbeanstalk:environment:process:https:
 Port: '443'
 Protocol: HTTPS
 aws:elasticbeanstalk:environment:process:admin:
 HealthCheckPath: /admin
 Port: '4443'
 Protocol: HTTPS
 aws:elbv2:listenerrule:admin:
 PathPatterns: /admin/*
 Priority: 1
 Process: admin
```

## Configurar um Application Load Balancer compartilhado

Quando você [habilita o balanceamento de carga](#), o ambiente do AWS Elastic Beanstalk é equipado com um balanceador de carga do Elastic Load Balancing para distribuir o tráfego entre as instâncias do ambiente. O Elastic Load Balancing é compatível com vários tipos de balanceador de carga. Para saber mais sobre eles, consulte o [Guia do usuário do Elastic Load Balancing](#). O Elastic Beanstalk pode criar um balanceador de carga para você ou permitir a especificação de um balanceador de carga compartilhado que você criou.

Este tópico descreve a configuração de um [Application Load Balancer](#) compartilhado que você cria e associa ao ambiente. Consulte também [the section called “Application Load Balancer”](#). Para obter informações sobre como configurar todos os tipos de balanceador de carga compatíveis com o Elastic Beanstalk, consulte [Balanceador de carga do ambiente do Elastic Beanstalk](#).

**Note**

Você só pode escolher o tipo de load balancer que seu ambiente usará durante a criação do ambiente. Você pode alterar as configurações para gerenciar o comportamento do load balancer do ambiente em execução, mas não pode alterar seu tipo. Também não é possível alternar de um load balancer dedicado para um compartilhado ou vice-versa.

## Introdução

Um balanceador de carga compartilhado é um balanceador de carga criado e gerenciado usando o serviço Amazon Elastic Compute Cloud (Amazon EC2) e, depois, é usado em vários ambientes do Elastic Beanstalk.

Quando um ambiente é criado com balanceamento de carga e escalabilidade automática e um Application Load Balancer é utilizado, o Elastic Beanstalk cria um balanceador de carga dedicado ao ambiente por padrão. Para saber o que é um Application Load Balancer e como ele funciona em um ambiente do Elastic Beanstalk, consulte a [introdução](#) à configuração de um Application Load Balancer para Elastic Beanstalk.

Em algumas situações, você pode economizar o custo de ter vários load balancers dedicados. Isso pode ser útil quando você tem vários ambientes, por exemplo, se a aplicação for um pacote de microsserviços em vez de um serviço monolítico. Nesses casos, é possível optar por usar um load balancer compartilhado.

Para usar um balanceador de carga compartilhado, primeiro crie-o no Amazon EC2 e adicione um ou mais listeners. Durante a criação de um ambiente do Elastic Beanstalk, é necessário fornecer o balanceador de carga e escolher uma porta do listener. O Elastic Beanstalk associa o listener ao processo padrão no ambiente. É possível adicionar regras de listener personalizadas para rotear o tráfego de cabeçalhos e caminhos de host específicos para outros processos do ambiente.

O Elastic Beanstalk adiciona uma tag ao balanceador de carga compartilhado. O nome da tag é `elasticbeanstalk:shared-elb-environment-count`, e seu valor é o número de ambientes que compartilham esse load balancer.

Usar um load balancer compartilhado é diferente de usar um dedicado de diversas maneiras.



Referente	Application Load Balancer dedicado	Application Load Balancer compartilhado
Gerenciamento	<p>O Elastic Beanstalk cria e gerencia o balanceador de carga, os listeners, as regras de listener e os processos (grupos de destino). O Elastic Beanstalk também os remove quando o ambiente é encerrado. O Elastic Beanstalk pode definir a captura de logs de acesso do balanceador de carga, caso escolha essa opção.</p>	<p>O balanceador de carga e os listeners são criados e gerenciados fora do Elastic Beanstalk. O Elastic Beanstalk cria e gerencia uma regra padrão e um processo padrão, e regras e processos podem ser adicionados. O Elastic Beanstalk remove as regras e os processos de listener que foram adicionados durante a criação do ambiente.</p>
Regras do listener	<p>O Elastic Beanstalk cria uma regra padrão para cada listener a fim de rotear todo o tráfego para o processo padrão do listener.</p>	<p>O Elastic Beanstalk associa uma regra padrão apenas a um listener da porta 80, se existir. Se você escolher uma porta de listener padrão diferente, precisará associar a regra padrão a ela (o console do Elastic Beanstalk e a CLI do EB fazem isso por você).</p> <p>Para resolver conflitos de condição de regra de listener em ambientes que compartilham o balanceador de carga, o Elastic Balanceador adiciona o CNAME do ambiente à regra de listener como uma condição de cabeçalho de host.</p> <p>O Elastic Beanstalk trata as configurações de prioridade de regra como relativas entre ambientes que compartilham o balanceador de carga e as mapeia para prioridades absolutas durante a criação.</p>
Grupos de segurança	<p>O Elastic Beanstalk cria um grupo de segurança padrão e o anexa ao balanceador de carga.</p>	<p>É possível configurar um ou mais grupos de segurança a serem usados para o load balancer. Caso contrário, o Elastic Beanstalk</p>

Referente	Application Load Balancer dedicado	Application Load Balancer compartilhado
		<p>verifica se um grupo de segurança existente gerenciado pelo Elastic Beanstalk já está conectado ao balanceador de carga. Caso contrário, o Elastic Beanstalk cria um grupo de segurança e o anexa ao balanceador de carga. O Elastic Beanstalk exclui esse grupo de segurança quando o último ambiente que compartilha o balanceador de carga é encerrado.</p>
Atualizações	<p>É possível atualizar o Application Load Balancer após a criação do ambiente. É possível editar listeners, regras de listener e processos. É possível configurar a captura de log de acesso do load balancer.</p>	<p>Não é possível usar o Elastic Beanstalk para configurar a captura de logs de acesso no Application Load Balancer nem atualizar listeners e regras de listener após a criação do ambiente. Só é possível atualizar processos (grupos de destino). Para configurar a captura de logs de acesso e atualizar listeners e regras de listener, use o Amazon EC2.</p>

## Configurar um Application Load Balancer compartilhado usando o console do Elastic Beanstalk

É possível usar o console do Elastic Beanstalk para configurar um Application Load Balancer compartilhado durante a criação do ambiente. É possível selecionar um dos load balancers compartilháveis da conta para uso no ambiente, selecionar a porta de listener padrão e configurar processos adicionais e regras de listener.

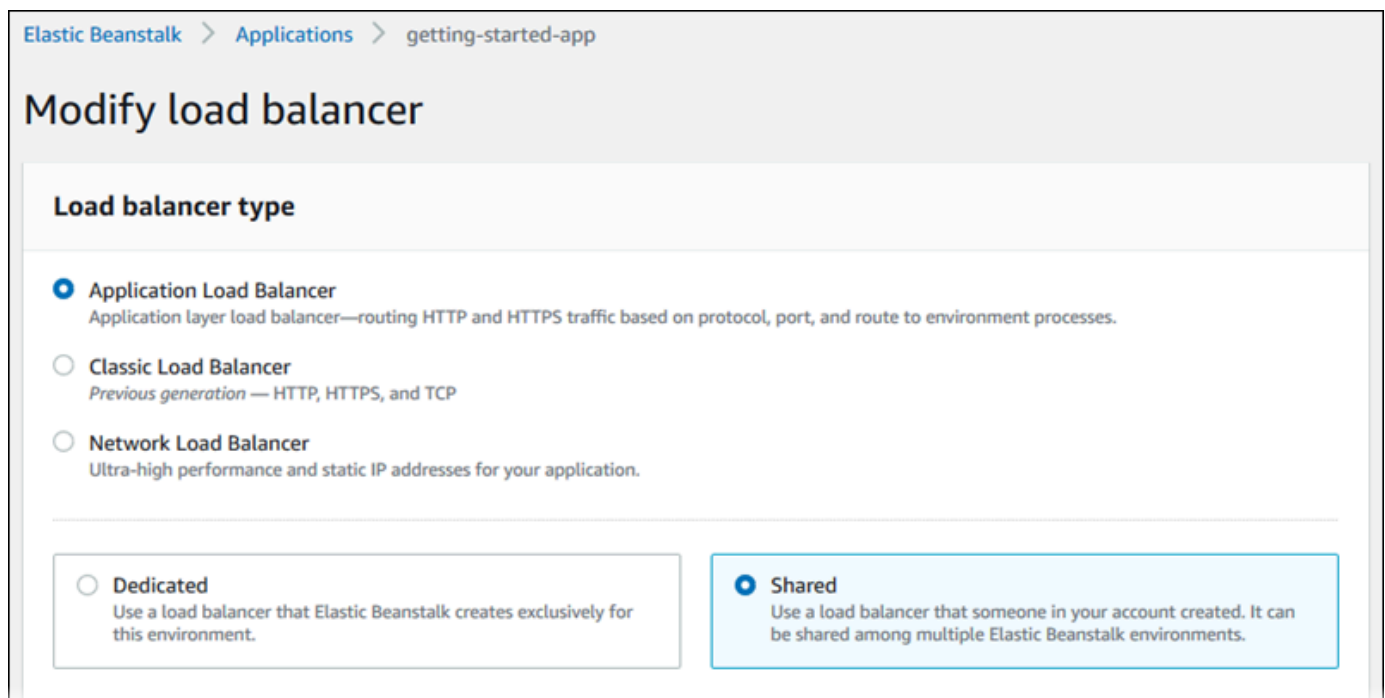
Não é possível editar a configuração do Application Load Balancer compartilhado no console do Application Load Balancer após a criação do ambiente. Para configurar listeners, regras de listener, processos (grupos de destino) e captura de logs de acesso, use o Amazon EC2.

Como configurar um Application Load Balancer no console do Elastic Beanstalk durante a criação do ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, escolha Environments (Ambientes).
3. Escolha [Create a new environment \(Criar um ambiente\)](#) para começar a criar seu ambiente.
4. Na página principal do assistente, antes de escolher Create environment, escolha Configure more options.
5. Escolha a predefinição de configuração High availability (Alta disponibilidade).

Se preferir, na categoria de configuração Capacity (Capacidade), configure o tipo de ambiente Load balanced (Carga equilibrada). Para obter mais detalhes, consulte [Capacity](#).

6. Na categoria de configuração Load balancer, escolha Edit (Editar).
7. Selecione a opção Application Load Balancer se ainda não estiver selecionada e, depois, selecione a opção Shared (Compartilhado).



8. Faça todas as alterações de configuração do Application Load Balancer compartilhado necessárias ao seu ambiente.
9. Escolha Save e, em seguida, faça as outras alterações de configuração de acordo com seu ambiente.

## 10. Selecione Create environment (Criar ambiente).

### Configurações do Application Load Balancer compartilhado

- [Application Load Balancer compartilhado](#)
- [Processos](#)
- [Regras](#)

### Application Load Balancer compartilhado

Use esta seção para escolher um Application Load Balancer compartilhado para seu ambiente e configurar o roteamento de tráfego padrão.

Antes de configurar um Application Load Balancer compartilhado aqui, use o Amazon EC2 para definir pelo menos um Application Load Balancer para compartilhamento com pelo menos um ouvinte, em sua conta. Se você ainda não fez isso, poderá escolher Manage load balancers (Gerenciar balanceadores de carga). O Elastic Beanstalk abre o console do Amazon EC2 em uma nova guia do navegador.

Quando terminar de configurar balanceadores de carga compartilhados fora do Elastic Beanstalk, defina as configurações a seguir nesta seção do console:

- Load balancer ARN (ARN do balanceador de carga): o balanceador de carga compartilhado a ser usado nesse ambiente. Selecione o load balancer na lista ou insira o nome de recurso da Amazon (ARN) de um load balancer.
- Default listener port (Porta do listener padrão): uma porta do listener que o balanceador de carga compartilhado escuta. Faça sua escolha em uma lista de portas de listener existentes. O tráfego desse listener com o CNAME do ambiente no cabeçalho do host é roteado para um processo padrão nesse ambiente.

### Shared Application Load Balancer

Select a shared load balancer and default listener for your environment. To manage load balancers and listeners, choose **Manage load balancers**.

[Manage load balancers](#)

Load balancer ARN

arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/example-sh: X ↻

Must be an active Application Load Balancer in vpc-5732152e

Default listener

The default process and rule are associated with this listener.

80 (HTTP) ▼ ↻

## Processos

Use essa lista para especificar processos para o load balancer compartilhado. Um processo é um destino para os listeners direcionarem o tráfego. Inicialmente, a lista mostra o processo padrão, que recebe tráfego do listener padrão.

### Processes

For each environment process, you can specify the protocol and port that the load balancer uses to route requests to the process. You can also specify how the load balancer performs process health checks.

[Actions](#) ▼ [+ Add process](#)

<input type="checkbox"/>	Name	Port	Protocol	HTTP code	Health check path	Stickiness
<input type="checkbox"/>	default	80	HTTP		/	disabled

## Como configurar um processo existente

1. Marque a caixa de seleção ao lado da entrada da tabela e, em seguida, escolha **Actions**, **Edit**.
2. Use a caixa de diálogo **Environment process** (Processo do ambiente) para editar as configurações e selecione **Save** (Salvar).

## Como adicionar um processo

1. Selecione **Add process** (Adicionar processo).

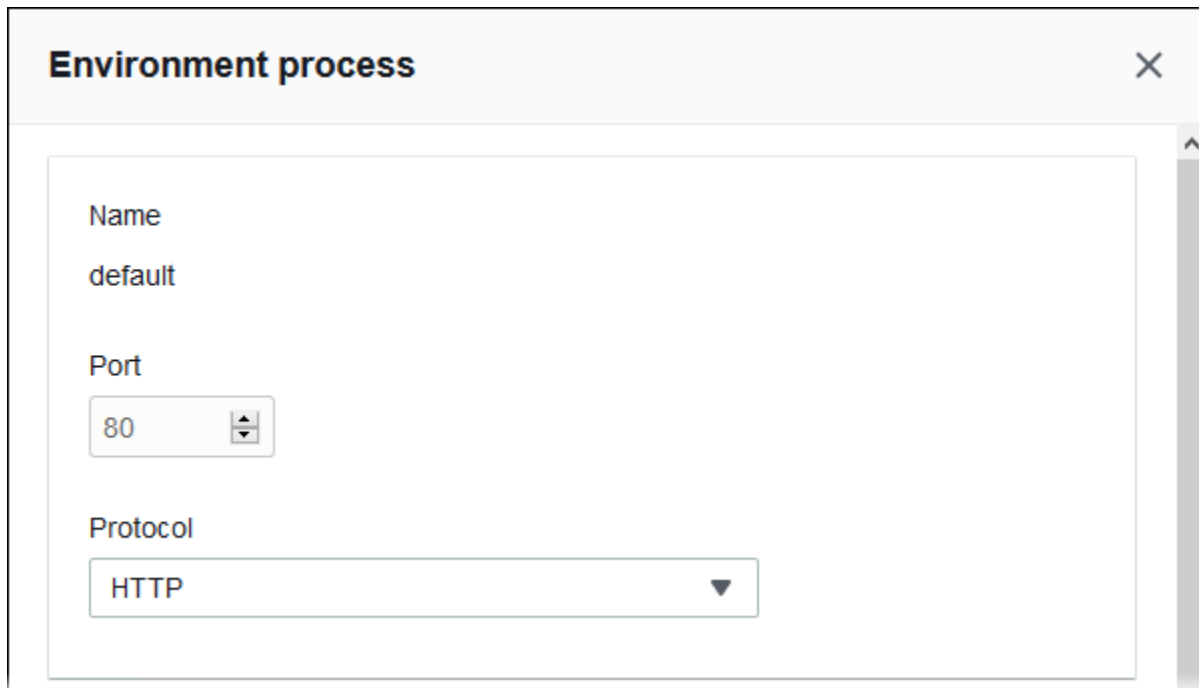
2. Na caixa de diálogo Environment Process (Processo do ambiente), defina as configurações desejadas e selecione Add (Adicionar).

Configurações da caixa de diálogo Environment process (Processo do ambiente) do Application Load Balancer

- [Definição](#)
- [Verificação de integridade](#)
- [Sessões](#)

### Definição

Use estas configurações para definir o processo Name, Port e Protocol em que as solicitações são observadas.



The screenshot shows a dialog box titled "Environment process" with a close button (X) in the top right corner. The dialog contains three configuration fields:

- Name:** A text input field containing the value "default".
- Port:** A spinner control showing the value "80".
- Protocol:** A dropdown menu with "HTTP" selected.

### Verificação de integridade

Use as configurações a seguir para configurar as verificações de integridade do processo:

- **HTTP code (Código de HTTP):** o código de status HTTP que designa um processo íntegro.
- **Path (Caminho):** o caminho da solicitação de verificação de integridade para o processo.
- **Timeout (Tempo limite):** a quantidade de tempo, em segundos, para aguardar uma resposta de verificação de integridade.

- **Interval (Intervalo):** a quantidade de tempo, em segundos, entre as verificações de integridade de uma instância individual. O intervalo deve ser maior que o tempo limite.
- **Unhealthy threshold (Limite não íntegro), Healthy threshold (Limite íntegro):** o número de verificações de integridade que devem falhar ou passar, respectivamente, antes que o Elastic Load Balancing altere o estado de integridade de uma instância.
- **Deregistration delay (Atraso do cancelamento de registro):** a quantidade de tempo, em segundos, para aguardar que as solicitações ativas sejam concluídas antes de cancelar o registro de uma instância.

## Health check

### HTTP code

HTTP status code of a healthy instance in your environment.

### Path

Path to which the load balancer sends HTTP health check requests.

### Timeout

Amount of time to wait for a health check response.

 seconds

### Interval

Amount of time between health checks of an individual instance. The interval must be greater than the timeout.

 seconds

### Unhealthy threshold

The number of consecutive health check failures required to designate the instance as unhealthy.

 requests

### Healthy threshold

The number of consecutive successful health checks required to designate the instance as healthy.

 requests

### Deregistration delay

Amount of time to wait for active requests to complete before deregistering.

 seconds



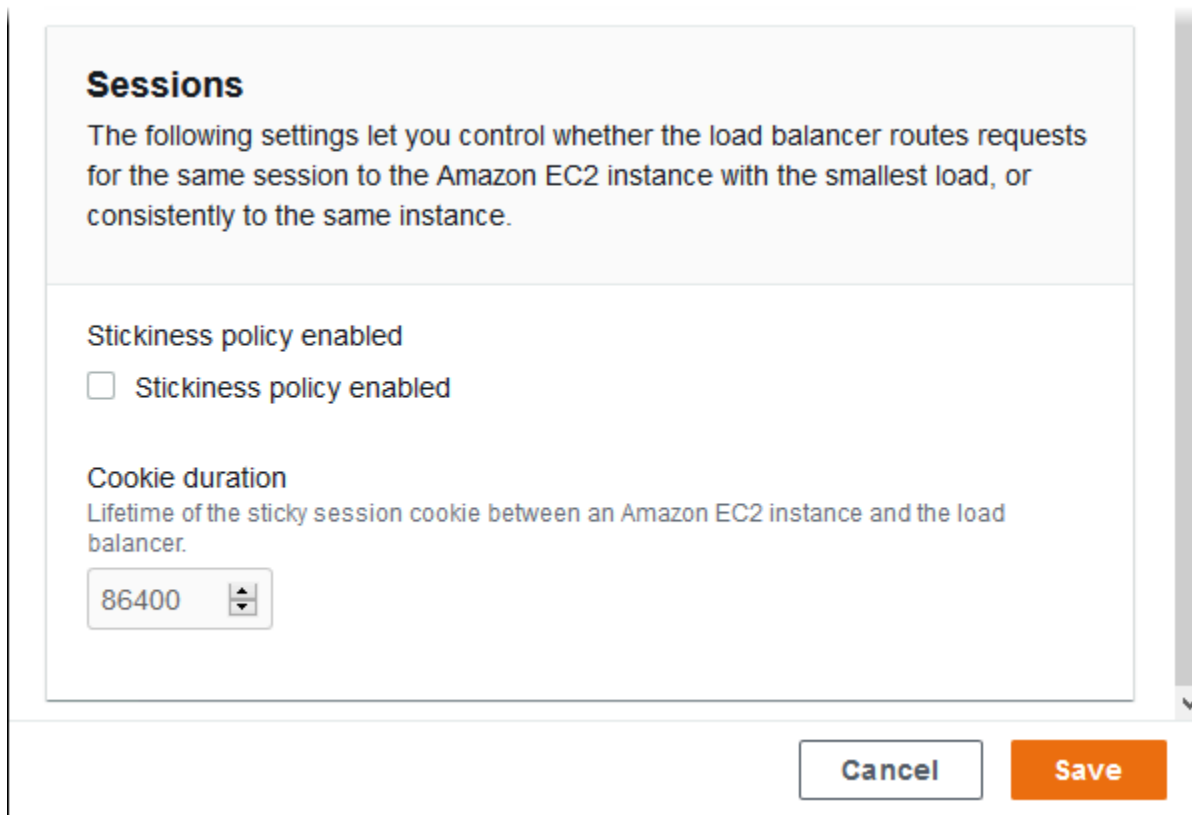
**Note**

A verificação de integridade do Elastic Load Balancing não afeta o comportamento da verificação de integridade do grupo de Auto Scaling de um ambiente. As instâncias que falham em uma verificação de integridade do Elastic Load Balancing não são substituídas automaticamente pelo Amazon EC2 Auto Scaling, a menos que você configure manualmente o Amazon EC2 Auto Scaling para fazer isso. Para mais detalhes, consulte [Configuração da verificação de integridade do Auto Scaling](#).

Para obter mais informações sobre verificações de integridade e como elas influenciam a integridade geral do ambiente, consulte [Relatórios de integridade básica](#).

**Sessões**

Selecione ou desmarque a caixa Política de perdurabilidade habilitada para habilitar ou desabilitar as sticky sessions. Use Duração do cookie para configurar a duração de uma sticky session até **604800** segundos.



**Sessions**

The following settings let you control whether the load balancer routes requests for the same session to the Amazon EC2 instance with the smallest load, or consistently to the same instance.

Stickiness policy enabled

Stickiness policy enabled

Cookie duration

Lifetime of the sticky session cookie between an Amazon EC2 instance and the load balancer.

86400

Cancel Save

## Regras

Use essa lista para especificar regras de listener para o load balancer compartilhado. Uma regra mapeia solicitações que o listener recebe em um padrão de caminho específico para um processo de destino. Cada listener pode ter várias regras, roteando solicitações em caminhos diferentes para processos diferentes em instâncias dos diferentes ambientes que compartilham o listener.

As regras têm prioridades numéricas que determinam a precedência em que são aplicadas às solicitações recebidas. O Elastic Beanstalk adiciona uma regra padrão que roteia todo o tráfego do listener padrão para o processo padrão do seu novo ambiente. A precedência da regra padrão é a mais baixa. Ela é aplicada se nenhuma outra regra para o mesmo listener corresponder à solicitação recebida. Inicialmente, se você não tiver adicionado regras personalizadas, a lista estará vazia. A regra padrão não é exibida.

**Rules**

Your load balancer routes requests to environment processes based on rules. Rules are evaluated by priority in ascending numerical order. If the shared load balancer has existing rules configured, this environment's rules are adjusted to have lower priority than existing rules. You can manage rules across environments in the EC2 console.

Elastic Beanstalk configures a default rule for this environment. This rule routes all traffic from the default listener on port **80** to the default process, and has the last priority among all rules of this environment. If a request doesn't match the conditions for any other rule, the default rule routes the request to the default process.

**Shared load balancer environment rules**

After environment creation, you can't add or edit rules for this environment using Elastic Beanstalk. When you terminate the environment, listener rules created outside of Elastic Beanstalk aren't automatically removed by Elastic Beanstalk.

Actions ▾ + Add rule

	Name	Listener port	Priority	Host headers	Path patterns	Process
No additional listener rules are currently configured. Choose <b>Add rule</b> to add a listener rule.						

Cancel Save

Você pode editar as configurações de uma regra existente ou adicionar uma nova regra. Para começar a editar uma regra na lista ou adicionar uma regra a ela, use as mesmas etapas listadas para a [lista de processos](#). A caixa de diálogo Listener rule é exibida com as seguintes configurações:

- Name (Nome): o nome da regra.
- Listener port (Porta do listener): porta do listener ao qual a regra se aplica.

- **Priority (Prioridade):** prioridade da regra. Um número de prioridade mais baixa tem maior precedência. As prioridades das regras de um listener devem ser únicas. O Elastic Beanstalk trata as prioridades de regras como relativas em ambientes de compartilhamento e as mapeia para prioridades absolutas durante a criação.
- **Match conditions (Condições de correspondência):** uma lista de condições do URL de solicitação às quais a regra se aplica. Existem dois tipos de condições: **HostHeader** (a parte do domínio do URL) e **PathPattern** (a parte do caminho do URL). Uma condição é reservada para o subdomínio do ambiente e é possível adicionar até quatro condições. Cada valor de condição tem até 128 caracteres de comprimento e pode incluir caracteres curinga.
- **Process (Processo):** o processo para o qual o balanceador de carga direciona solicitações que correspondem à regra.

### Listener rule ✕

**Name**  
images

**Listener port**  
80 ▼

**Priority**  
Evaluated in ascending numerical order. Must be unique across all rules.  
1 ▲▼

**Match conditions**  
A listener rule can have up to five match conditions.

Type	Value	
PathPattern ▼	/images/*	Remove

Add condition

**Process**  
images ▼

Cancel Add

Exemplo: usar um Application Load Balancer compartilhado para uma aplicação segura baseado em microsserviços

Neste exemplo, a aplicação consiste em vários microsserviços, cada um implementado como um ambiente do Elastic Beanstalk. Além disso, você precisa de criptografia de tráfego de ponta a ponta. Vamos demonstrar um dos ambientes de microsserviço, que tem um processo principal para solicitações de usuários e um processo separado para lidar com solicitações administrativas.

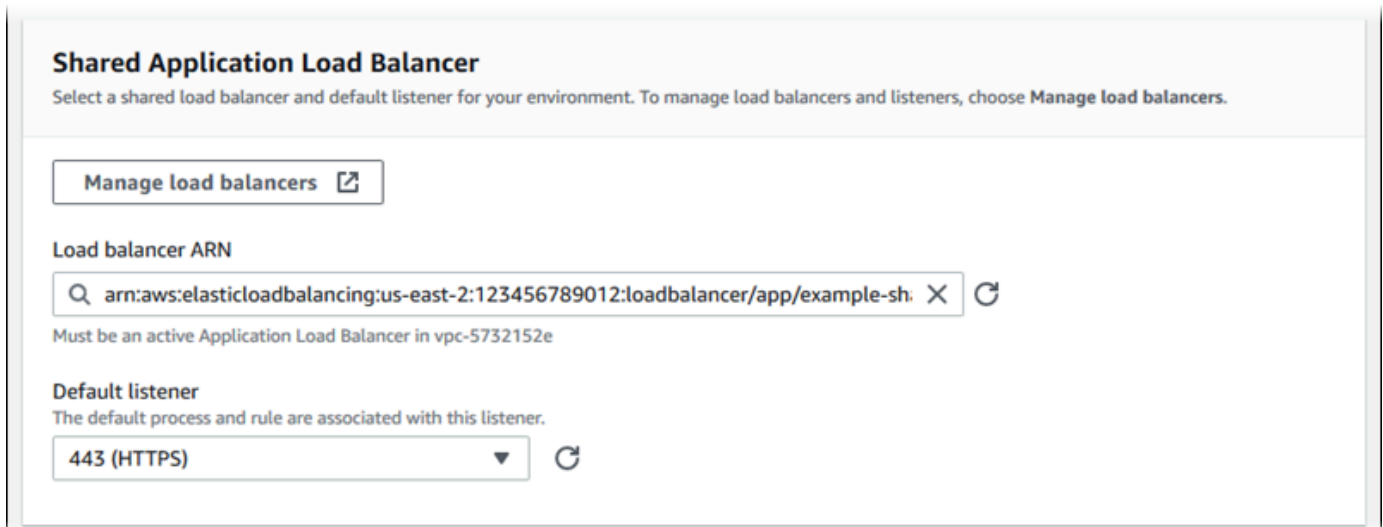
Para atender a esses requisitos, use o Amazon EC2 para criar um Application Load Balancer que você compartilhará entre seus microsserviços. Adicione um listener seguro na porta 443 e no protocolo HTTPS. Depois, adicione vários certificados SSL ao listener: um por domínio de

microsserviço. Para obter detalhes sobre como criar o Application Load Balancer e o listener seguro, consulte [Criar um Application Load Balancer](#) e [Criar um listener HTTPS para o Application Load Balancer](#) no Guia do usuário para Application Load Balancers.

No Elastic Beanstalk, configure cada ambiente de microsserviço para usar o Application Load Balancer compartilhado e defina a porta do listener padrão como 443. No caso do ambiente específico que estamos demonstrando aqui, indique que o processo padrão escuta a porta 443 em HTTPS e adicione um processo e uma regra de listener ao tráfego administrativo em um caminho diferente.

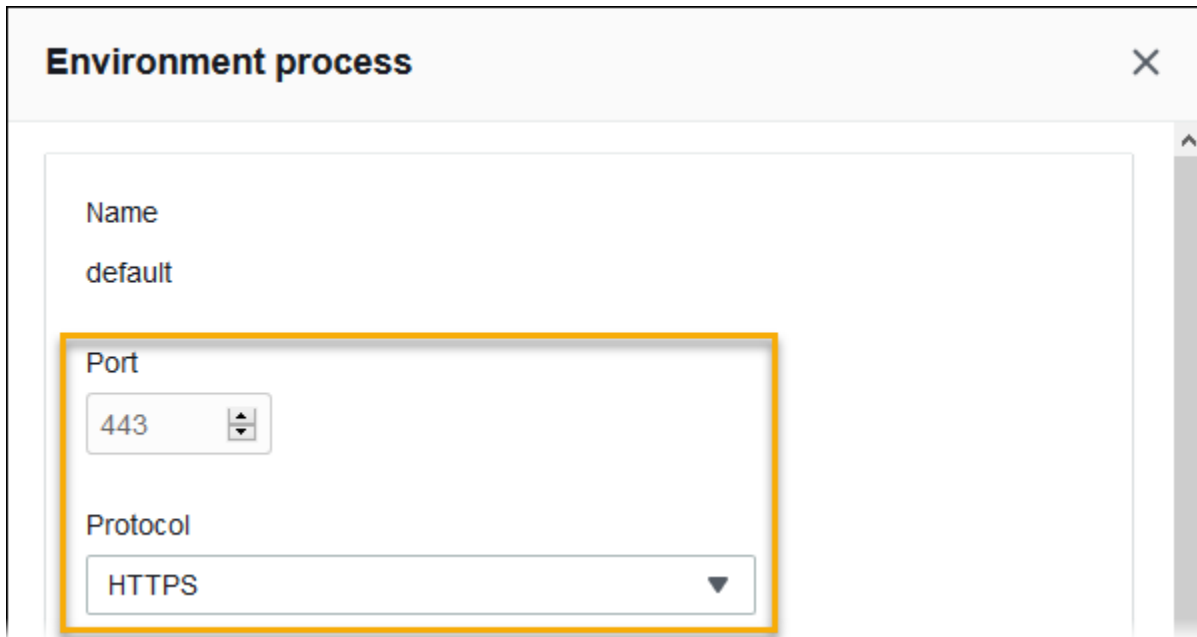
Como configurar o load balancer compartilhado neste exemplo

1. Na seção Shared Application Load Balancer (Application Load Balancer compartilhado), selecione o balanceador de carga e, depois, para Default listener port (Porta do listener padrão), selecione **443**. A porta do listener já deverá estar selecionada se for o único listener que o load balancer possui.



The screenshot shows the configuration interface for a Shared Application Load Balancer. At the top, it says "Shared Application Load Balancer" and "Select a shared load balancer and default listener for your environment. To manage load balancers and listeners, choose Manage load balancers." Below this is a button labeled "Manage load balancers" with an external link icon. Underneath is a text input field for "Load balancer ARN" containing the value "arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/example-sh" with a search icon on the left, a clear icon (X), and a refresh icon (C). Below the input field is a note: "Must be an active Application Load Balancer in vpc-5732152e". At the bottom, there is a "Default listener" section with the text "The default process and rule are associated with this listener." and a dropdown menu currently showing "443 (HTTPS)" with a refresh icon (C) to its right.

2. Configure o processo padrão como HTTPS. Selecione o processo padrão e, em seguida, para Actions, escolha Edit. Em Port (Porta), insira **443**. Em Protocolo, selecione **HTTPS**.



**Environment process** [X]

Name  
default

Port  
443

Protocol  
HTTPS

3. Adicione um processo administrativo. Em Name (Nome), insira **admin**. Em Port (Porta), insira **443**. Em Protocolo, selecione **HTTPS**. Em Health check (Verificação de integridade), para Path (Caminho), insira **/admin**.

**Environment process** [X]

Name  
admin

Port  
443

Protocol  
HTTPS

**Health check**

HTTP code  
HTTP status code of a healthy instance in your environment.  
200

Path  
Path to which the load balancer sends HTTP health check requests.  
/admin

4. Adicione uma regra para o tráfego de administrador. Em Name (Nome), insira **admin**. Em Listener port (Porta do listener), insira **443**. Em Match conditions (Condições de correspondência), adicione um PathPattern com o valor **/admin/\***. Em Processo, selecione **admin**.

### Listener rule ✕

**Name**  
admin

**Listener port**  
443 ▼

**Priority**  
Evaluated in ascending numerical order. Must be unique across all rules.  
1

**Match conditions**  
A listener rule can have up to five match conditions.

Type	Value	
PathPattern ▼	/admin/*	Remove

Add condition

**Process**  
admin ▼

Cancel Add

## Configurar um Application Load Balancer compartilhado usando a CLI do EB

A EB CLI solicita que você escolha um tipo de load balancer ao executar o [eb create](#). Se você escolher `application` (o padrão) e sua conta tiver pelo menos um Application Load Balancer compartilhável, a CLI do EB também perguntará se você deseja usar um Application Load Balancer compartilhado. Se responder `y`, você também será solicitado a selecionar o load balancer e a porta padrão.

```
$ eb create
Enter Environment Name
```



```
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
1) classic
2) application
3) network
(default is 2):

Your account has one or more sharable load balancers. Would you like your new
environment to use a shared load balancer?(y/N) y

Select a shared load balancer
1)MySharedALB1 - arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/
MySharedALB1/6d69caa75b15d46e
2)MySharedALB2 - arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/
MySharedALB2/e574ea4c37ad2ec8
(default is 1): 2

Select a listener port for your shared load balancer
1) 80
2) 100
3) 443
(default is 1): 3
```

Também é possível especificar um load balancer compartilhado usando opções de comando.

```
$ eb create test-env --elb-type application --shared-lb MySharedALB2 --shared-lb-
port 443
```

## Namespaces do Application Load Balancer compartilhado

Você pode encontrar configurações relacionadas aos Application Load Balancers compartilhados nos seguintes namespaces:

- [aws:elasticbeanstalk:environment](#): escolha o tipo de balanceador de carga para o ambiente e informe o Elastic Beanstalk que você usará um balanceador de carga compartilhado.

Não é possível definir essas duas opções em arquivos de configuração ([.Ebextensions](#)).

- [aws:elbv2:loadbalancer](#): configure o ARN e os grupos de segurança do Application Load Balancer compartilhado.

- [aws:elbv2:listener](#): associe os listeners do Application Load Balancer compartilhado aos processos de ambiente listando regras do listener.
- [aws:elbv2:listenerrule](#): configure as regras do listener que roteiam o tráfego para os diferentes processos, dependendo do caminho da solicitação. As regras são exclusivas dos Application Load Balancers, sejam eles dedicados ou compartilhados.
- [aws:elasticbeanstalk:environment:process](#): configure as verificações de integridade e especifique a porta e o protocolo para os processos que são executados nas instâncias de seu ambiente.

Example `.ebextensions/application-load-balancer-shared.config`

Para começar a usar um Application Load Balancer compartilhado, use o console do Elastic Beanstalk, a CLI do EB ou a API para definir o tipo de balanceador de carga como `application` e optar por usar um balanceador de carga compartilhado. Use um [arquivo de configuração](#) para configurar o load balancer compartilhado.

```
option_settings:
 aws:elbv2:loadbalancer:
 SharedLoadBalancer: arn:aws:elasticloadbalancing:us-
east-2:123456789012:loadbalancer/app/MySharedALB2/e574ea4c37ad2ec8
```

#### Note

Só é possível configurar essa opção durante a criação do ambiente.

Example `.ebextensions/alb-shared-secure-listener.config`

O arquivo de configuração a seguir seleciona um listener seguro padrão na porta 443 para o load balancer compartilhado e define o processo padrão para escutar a porta 443.

```
option_settings:
 aws:elbv2:loadbalancer:
 SharedLoadBalancer: arn:aws:elasticloadbalancing:us-
east-2:123456789012:loadbalancer/app/MySharedALB2/e574ea4c37ad2ec8
 aws:elbv2:listener:443:
 rules: default
 aws:elasticbeanstalk:environment:process:default:
```

```
Port: '443'
Protocol: HTTPS
```

### Example .ebextensions/alb-shared-admin-rule.config

O arquivo de configuração a seguir se baseia no exemplo anterior e adiciona uma regra que roteia o tráfego com um caminho de solicitação de /admin para um processo chamado admin que escuta na porta 4443.

```
option_settings:
 aws:elbv2:loadbalancer:
 SharedLoadBalancer: arn:aws:elasticloadbalancing:us-
east-2:123456789012:loadbalancer/app/MySharedALB2/e574ea4c37ad2ec8
 aws:elbv2:listener:443:
 rules: default,admin
 aws:elasticbeanstalk:environment:process:default:
 Port: '443'
 Protocol: HTTPS
 aws:elasticbeanstalk:environment:process:admin:
 HealthCheckPath: /admin
 Port: '4443'
 Protocol: HTTPS
 aws:elbv2:listenerrule:admin:
 PathPatterns: /admin/*
 Priority: 1
 Process: admin
```

## Configurar um Network Load Balancer

Quando você [habilita o balanceamento de carga](#), o ambiente do AWS Elastic Beanstalk é equipado com um balanceador de carga do Elastic Load Balancing para distribuir o tráfego entre as instâncias do ambiente. O Elastic Load Balancing é compatível com vários tipos de balanceador de carga. Para saber mais sobre eles, consulte o [Guia do usuário do Elastic Load Balancing](#). O Elastic Beanstalk pode criar um balanceador de carga para você ou permitir a especificação de um balanceador de carga compartilhado que você criou.

Este tópico descreve a configuração de um [Network Load Balancer](#) que o Elastic Beanstalk cria e dedica ao seu ambiente. Para obter informações sobre como configurar todos os tipos de balanceador de carga compatíveis com o Elastic Beanstalk, consulte [Balanceador de carga do ambiente do Elastic Beanstalk](#).

**Note**

Você só pode escolher o tipo de load balancer que seu ambiente usará durante a criação do ambiente. Você pode alterar as configurações para gerenciar o comportamento do load balancer do ambiente em execução, mas não pode alterar seu tipo.

## Introdução

Com o Network Load Balancer, o listener padrão aceita solicitações de TCP na porta 80 e as distribui para as instâncias no seu ambiente. Você pode configurar o comportamento de verificação de integridade, configurar uma porta do listener ou adicionar um listener a outra porta.

**Note**

Ao contrário de um Classic Load Balancer ou um Application Load Balancer, um Network Load Balancer não pode ter listeners HTTP ou HTTPS da camada de aplicação (camada 7). Ele só oferece suporte aos listeners de TCP da camada de transporte (camada 4). O tráfego HTTP e HTTPS podem ser roteados para o ambiente por TCP. Para estabelecer conexões HTTPS seguras entre os clientes da Web e seu ambiente, instale um [certificado autoassinado](#) nas instâncias do ambiente e configure as instâncias para escutar na porta apropriada (normalmente 443) e encerrar as conexões HTTPS. A configuração varia de acordo com a plataforma. Para obter instruções, consulte [Configurar o aplicativo para encerrar conexões HTTPS na instância](#). Depois, configure o Network Load Balancer para adicionar um listener que é mapeado para um processo que escuta na porta apropriada.

Um Network Load Balancer é compatível com verificações de integridade ativas. Essas verificações são baseadas em mensagens para o caminho da raiz (/). Além disso, um Network Load Balancer é compatível com verificações de integridade passivas. Ele detecta automaticamente as instâncias de back-end defeituosas e roteia o tráfego somente para instâncias íntegras.

## Configurar um Network Load Balancer usando o console do Elastic Beanstalk

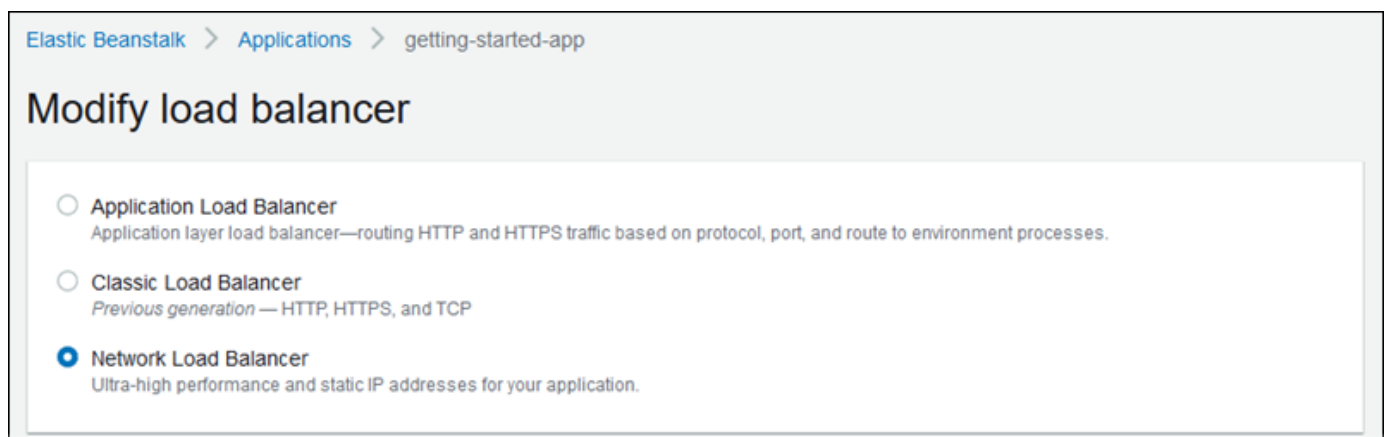
É possível usar o console do Elastic Beanstalk para configurar os listeners e os processos de um Network Load Balancer durante a criação do ambiente ou mais tarde quando seu ambiente estiver em execução.

Como configurar um Network Load Balancer no console do Elastic Beanstalk durante a criação do ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, escolha Environments (Ambientes).
3. Escolha [Create a new environment \(Criar um ambiente\)](#) para começar a criar seu ambiente.
4. Na página principal do assistente, antes de escolher Create environment, escolha Configure more options.
5. Escolha a predefinição de configuração High availability (Alta disponibilidade).

Se preferir, na categoria de configuração Capacity (Capacidade), configure o tipo de ambiente Load balanced (Carga equilibrada). Para obter mais detalhes, consulte [Capacity](#).

6. Na categoria de configuração Load balancer, escolha Edit (Editar).
7. Selecione a opção Network Load Balancer se ainda não estiver selecionada.




8. Faça todas as alterações de configuração do Network Load Balancer necessárias ao seu ambiente.
9. Escolha Save e, em seguida, faça as outras alterações de configuração de acordo com seu ambiente.
10. Selecione Create environment (Criar ambiente).

Como configurar o Network Load Balancer de um ambiente em execução no console do Elastic Beanstalk


1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.

2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Load balancer, escolha Edit (Editar).

 Note

Se a categoria de configuração Load balancer não tem um botão Edit (Editar), seu ambiente não tem um load balancer. Para aprender como configurar um, consulte [Alterar tipo de ambiente](#).

5. Faça as alterações de configuração do Network Load Balancer necessárias ao seu ambiente.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

## Configurações do Network Load Balancer

- [Listeners](#)
- [Processos](#)

### Listeners

Use esta lista para especificar listeners para seu load balancer. Cada listener direciona o tráfego de entrada do cliente em uma porta especificada para um processo em suas instâncias. Inicialmente, a lista mostra o listener padrão, que encaminha o tráfego de entrada na porta 80 para um processo chamado default, que observa a porta 80.

### Network Load Balancer

You can specify listeners for your load balancer. Each listener routes incoming client traffic on a specified port using TCP to an environment process (specified by the port that the process listens on). By default, we've configured your load balancer with a listener on port 80 that routes traffic to a default process listening on port 80.

Actions ▼ Add listener

<input type="checkbox"/>	Listener port	Process port	Protocol	Enabled
<input type="checkbox"/>	80	80	TCP	<input checked="" type="checkbox"/>

Para configurar um listener existente

1. Marque a caixa de seleção ao lado da entrada da tabela e, em seguida, escolha Actions, Edit.
2. Use a caixa de diálogo Network Load Balancer listener para editar as configurações e, em seguida, escolha Save.

Para adicionar um listener

1. Escolha Add listener.
2. Na caixa de diálogo Network Load Balancer listener, defina as configurações obrigatórias e, em seguida, escolha Add.

Use a caixa de diálogo Network Load Balancer listener para configurar a porta na qual o ouvinte atende ao tráfego e para escolher o processo para o qual você deseja rotear o tráfego (especificado pela porta na qual o processo escuta).

### Network Load Balancer listener ✕

**Listener port**  
80

**Protocol**  
The transport protocol that the load balancer uses for routing incoming traffic from clients.  
TCP ▼

**Process port**  
The port to which this listener routes traffic. It determines the environment process that receives traffic from the listener.  
80 ▼

**Cancel** **Save**

## Processos

Use esta lista para especificar processos para seu load balancer. Um processo é um destino para os listeners direcionarem o tráfego. Cada listener direciona o tráfego de entrada do cliente em uma porta especificada para um processo em suas instâncias. Inicialmente, a lista mostra o processo padrão, que observa o tráfego de entrada na porta 80.



### Processes

For each environment process, you can specify the port that the load balancer uses to route requests to the process. You can also specify how the load balancer performs process health checks.

Actions ▼ Add process

<input type="checkbox"/>	Process name	Process port	Interval	Healthy threshold	Unhealthy threshold
<input type="checkbox"/>	default	80	10	5	5

Cancel Save

Você pode editar as configurações de um processo existente ou adicionar um novo processo. Para começar a editar um processo na lista ou adicionar um processo a ele, use as mesmas etapas listadas para [lista de listener](#). A caixa de diálogo Environment process é exibida.

Configurações da caixa de diálogo Environment process (Processo do ambiente) do Network Load Balancer

- [Definição](#)
- [Verificação de integridade](#)

### Definição

Use estas configurações para definir o processo Name e Process port no qual as solicitações são observadas.



**Environment process** [X]

Name  
default

Process port  
80


### Verificação de integridade

Use as configurações a seguir para configurar as verificações de integridade do processo:

- **Interval (Intervalo):** a quantidade de tempo, em segundos, entre as verificações de integridade de uma instância individual.
- **Healthy threshold (Limite íntegro):** o número de verificações de integridade que devem ser aprovadas antes de o Elastic Load Balancing alterar o estado de integridade de uma instância. (Para Network Load Balancer, Unhealthy threshold (Limite não íntegro) é uma configuração somente leitura que é sempre igual ao valor limite de integridade.)
- **Deregistration delay (Atraso do cancelamento de registro):** a quantidade de tempo, em segundos, para aguardar que as solicitações ativas sejam concluídas antes de cancelar o registro de uma instância.


## Health check

**Interval**  
Amount of time between health checks of an individual instance.


10 

seconds


**Healthy threshold**  
The number of consecutive successful health checks required to designate the instance as healthy.

5  requests

**Unhealthy threshold**  
The number of consecutive health check failures required to designate the instance as unhealthy.

5  requests

**Deregistration delay**  
Amount of time to wait for active requests to complete before deregistering.

20  seconds

### Note

A verificação de integridade do Elastic Load Balancing não afeta o comportamento da verificação de integridade do grupo de Auto Scaling de um ambiente. As instâncias que falham em uma verificação de integridade do Elastic Load Balancing não serão automaticamente substituídas pelo Amazon EC2 Auto Scaling, a menos que você configure manualmente o Amazon EC2 Auto Scaling para fazê-lo. Para mais detalhes, consulte [Configuração da verificação de integridade do Auto Scaling](#).

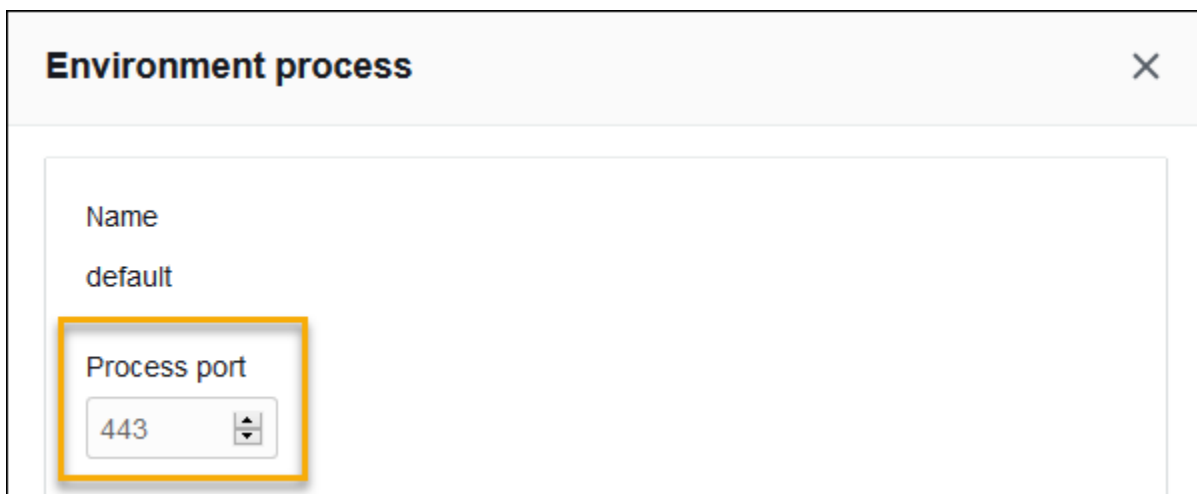
Para obter mais informações sobre verificações de integridade e como elas influenciam a integridade geral do ambiente, consulte [Relatórios de integridade básica](#).

## Exemplo: Network Load Balancer para um ambiente com criptografia de ponta a ponta

Neste exemplo, o aplicativo requer criptografia de tráfego ponta a ponta. Para configurar o Network Load Balancer do ambiente para atender a esses requisitos, configure o processo padrão para escutar na porta 443, adicione um listener à porta 443 que roteia o tráfego para o processo padrão e desabilite o listener padrão.

Para configurar o load balancer neste exemplo

1. Configure o processo padrão. Selecione o processo padrão e, em seguida, em Actions, escolha Edit. Para Process port, digite 443.



2. Adicione um listener da porta 443. Adicione um novo listener. Para Listener port, digite 443. Para Process port, certifique-se de selecionar 443.

### Network Load Balancer listener ✕

**Listener port**

443

**Protocol**  
The transport protocol that the load balancer uses for routing incoming traffic from clients.

TCP

**Process port**

The port to which this listener routes traffic. It determines the environment process that receives traffic from the listener.

443

Agora você pode ver seu listener adicional na lista.

<input type="checkbox"/>	Listener port	Process port	Protocol	Enabled
<input type="checkbox"/>	80	443	TCP	<input checked="" type="checkbox"/>
<input type="checkbox"/>	443	443	TCP	<input checked="" type="checkbox"/>

- Desative o listener da porta padrão 80. Para o listener padrão, desative a opção Habilitada.

<input type="checkbox"/>	Listener port	Process port	Protocol	Enabled
<input type="checkbox"/>	80	443	TCP	<input type="checkbox"/>
<input type="checkbox"/>	443	443	TCP	<input checked="" type="checkbox"/>

## Configurar um Network Load Balancer usando a CLI do EB

A EB CLI solicita que você escolha um tipo de load balancer ao executar o [eb create](#).

```
$ eb create
Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
1) classic
2) application
3) network
(default is 1): 3
```

Você também pode especificar um tipo de load balancer com a opção `--elb-type`.

```
$ eb create test-env --elb-type network
```

## Namespaces do Network Load Balancer

É possível encontrar as configurações relacionadas aos Network Load Balancers nos seguintes namespaces:

- [aws:elasticbeanstalk:environment](#): selecione o tipo de balanceador de carga para o ambiente. O valor de um Network Load Balancer é `network`.
- [aws:elbv2:listener](#): configure os listeners no Network Load Balancer. Essas configurações são mapeadas para as configurações em `aws:elb:listener` para Classic Load Balancers.
- [aws:elasticbeanstalk:environment:process](#): configure as verificações de integridade e especifique a porta e o protocolo para os processos que são executados nas instâncias de seu ambiente. A porta e as configurações de protocolo mapeiam a porta e as configurações de protocolo da instância em `aws:elb:listener` para um listener no Classic Load Balancer. As configurações de verificação de integridade são mapeadas para as configurações nos namespaces `aws:elb:healthcheck` e `aws:elasticbeanstalk:application`.

Example `.ebextensions/network-load-balancer.config`

Para começar a usar um Network Load Balancer, use um [arquivo de configuração](#) para definir o tipo de balanceador de carga como `network`.

```
option_settings:
 aws:elasticbeanstalk:environment:
```

```
LoadBalancerType: network
```

**Note**

Você pode definir o tipo de load balancer apenas durante a criação do ambiente.

**Example .ebextensions/nlb-default-process.config**

O arquivo de configuração a seguir modifica as configurações de verificação de integridade no processo padrão.

```
option_settings:
 aws:elasticbeanstalk:environment:process:default:
 DeregistrationDelay: '20'
 HealthCheckInterval: '10'
 HealthyThresholdCount: '5'
 UnhealthyThresholdCount: '5'
 Port: '80'
 Protocol: TCP
```

**Example .ebextensions/nlb-secure-listener.config**

O arquivo de configuração a seguir adiciona um listener para o tráfego seguro na porta 443 e um processo de destino correspondente com listener na porta 443.

```
option_settings:
 aws:elbv2:listener:443:
 DefaultProcess: https
 ListenerEnabled: 'true'
 aws:elasticbeanstalk:environment:process:https:
 Port: '443'
```

A opção `DefaultProcess` é assim denominada em virtude dos Application Load Balancers, que podem ter listeners não padrão na mesma porta para tráfego para caminhos específicos (consulte [Application Load Balancer](#) para obter detalhes). Para um Network Load Balancer, a opção especifica o único processo de destino para esse listener.

Neste exemplo, chamamos o processo de `https` porque ele faz o listening do tráfego seguro (HTTPS). O listener envia o tráfego para o processo na porta designada usando o protocolo TCP,

pois o Network Load Balancer funciona apenas com TCP. Isso está correto, porque o tráfego de rede para HTTP e HTTPS é implementado na parte superior do TCP.

## Configurar logs de acesso

Use os [arquivos de configuração](#) para configurar o balanceador de carga do ambiente a fazer upload de logs de acesso em um bucket do Amazon S3. Veja os arquivos de configuração de exemplo a seguir no GitHub para obter instruções:

- [loadbalancer-accesslogs-existingbucket.config](#): configure o balanceador de carga para fazer upload de logs de acesso em um bucket existente do Amazon S3.
- [loadbalancer-accesslogs-newbucket.config](#): configure o balanceador de carga para fazer upload de logs de acesso em um novo bucket.

## Adição de um banco de dados ao seu ambiente do Elastic Beanstalk

O Elastic Beanstalk fornece integração com o [Amazon Relational Database Service \(Amazon RDS\)](#). Você pode usar o Elastic Beanstalk para adicionar um banco de dados MySQL, PostgreSQL, Oracle ou SQL Server ao ambiente durante ou após a criação do ambiente. Quando você adiciona uma instância de banco de dados, o Elastic Beanstalk fornece as informações de conexão à sua aplicação. Ele faz isso definindo as propriedades do ambiente para o nome de host do banco de dados, a porta, o nome de usuário, a senha e o nome do banco de dados.

Se nunca usou uma instância de banco de dados com sua aplicação, recomendamos que você primeiro use o processo descrito neste tópico para adicionar um banco de dados a um ambiente de teste com o serviço Elastic Beanstalk. Fazendo isso, você pode verificar se a aplicação pode ler as propriedades do ambiente, criar uma sequência de conexão e conectar-se a uma instância de banco de dados, sem o trabalho de configuração adicional necessário para um banco de dados externo ao Elastic Beanstalk.

Depois de verificar se a aplicação funciona corretamente com o banco de dados, você pode considerar a possibilidade de passar para um ambiente de produção. Neste ponto, você tem a opção de desacoplar o banco de dados do ambiente do Elastic Beanstalk para passar para uma configuração que ofereça mais flexibilidade. O banco de dados desacoplado pode permanecer operacional como uma instância de banco de dados externa do Amazon RDS. A integridade do ambiente não é afetada pelo desacoplamento do banco de dados. Se precisar terminar o ambiente,



you will be able to do it and also choose the option to keep the database available and operational outside of Elastic Beanstalk.

Using an external database has several advantages. You can connect to a database outside of various environments, use types of databases that are not compatible with integrated databases and perform blue/green deployments. As an alternative to using an external database created by Elastic Beanstalk, you can also create a database instance outside of your Elastic Beanstalk environment. Both options result in a database instance external to your Elastic Beanstalk environment and require additional configurations for the security group and connection sequence. For more information, consult [Using Amazon RDS with Elastic Beanstalk](#).

## Seções

- [Database lifecycle](#)
- [Adding an Amazon RDS database instance to your environment using the console](#)
- [Connecting to a database](#)
- [Configuring an integrated RDS database instance using the console](#)
- [Configuring an integrated RDS database instance using configuration files](#)
- [Detaching an Amazon RDS database instance using the console](#)
- [Detaching an Amazon RDS database instance using configuration files](#)

## Ciclo de vida do banco de dados

You can choose what you want to happen to the database after you detach it from your Elastic Beanstalk environment. The options you can choose are collectively called exclusion policies. The exclusion policies that follow will apply to a database after you [detach it from an Elastic Beanstalk environment](#) or terminate the Elastic Beanstalk environment.

- **Snapshot:** Elastic Beanstalk creates a snapshot of the database before you terminate it. You can restore a database from a snapshot when you add a database instance to an Elastic Beanstalk environment or create an autonomous database. For more information about how to create a new autonomous database instance from a snapshot, consult [Restoring from a DB snapshot \(Restaurar de um snapshot do banco de dados\)](#) in the Amazon RDS User Guide (Amazon RDS User Guide). There may be a charge for

armazenamento de snapshots de banco de dados. Para obter mais informações, consulte a seção [Armazenamento de backup em Definição de preço do Amazon RDS](#).

- Excluir: o Elastic Beanstalk termina o banco de dados. Após ser terminada, a instância do banco de dados não está mais disponível para nenhuma operação.
- Reter: a instância do banco de dados não foi terminada. Ela permanece disponível e operacional, embora desacoplada do Elastic Beanstalk. Então, você pode configurar um ou vários ambientes para conectar ao banco de dados como uma instância de banco de dados externa do Amazon RDS. Para obter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon RDS](#).

## Adição de uma instância de banco de dados do Amazon RDS ao seu ambiente usando o console

É possível adicionar uma instância de banco de dados ao seu ambiente usando o console do Elastic Beanstalk.

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Você pode configurar as seguintes opções:

- Snapshot: escolha um snapshot de banco de dados existente. O Elastic Beanstalk restaura o snapshot e o adiciona ao ambiente. O valor padrão é Nenhum. Quando o valor padrão é None (Nenhum), você pode configurar um novo banco de dados usando as outras configurações dessa página.

- Engine (Mecanismo): escolha um mecanismo de banco de dados.
- Engine version (Versão do mecanismo): escolha uma versão específica do mecanismo de banco de dados.
- Instance class (Classe de instância): escolha uma classe de instância de banco de dados. Para obter mais informações sobre classes de instância de banco de dados, consulte <https://aws.amazon.com/rds/>.
- Storage (Armazenamento): escolha a quantidade de armazenamento que deve ser provisionada para seu banco de dados. Você pode aumentar o armazenamento alocado posteriormente, mas não pode diminuí-lo. Para obter mais informações sobre a alocação de armazenamento, consulte [Features \(Recursos\)](#).
- Username (Nome de usuário): insira um nome de usuário de sua escolha usando uma combinação de apenas números e letras.
- Password (Senha): insira uma senha de sua preferência que contenha de 8 a 16 caracteres ASCII imprimíveis (exceto /, \ e @).
- Availability (Disponibilidade): escolha High (Multi-AZ) (Alta [Multi-AZ]) para executar um backup quente em uma segunda zona de disponibilidade para alta disponibilidade.
- Database deletion policy (política de exclusão do banco de dados): a política de exclusão determina o que acontece com o banco de dados após ele ser [desacoplado](#) do seu ambiente. Pode ser um dos seguintes valores: Create Snapshot, Retain ou Delete. Esses valores são descritos em [Ciclo de vida do banco de dados](#) neste mesmo tópico.

#### Note

O Elastic Beanstalk cria um usuário principal para o banco de dados usando o nome de usuário e a senha fornecidos. Para saber mais sobre o usuário principal e seus privilégios, consulte [Master User Account Privileges \(Privilégios da conta de usuário principal\)](#).

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização é concluída, o novo banco de dados é acoplado ao seu ambiente. O nome do host e outras informações de conexão da instância de banco de dados estão disponíveis para a sua aplicação por meio das seguintes propriedades do ambiente:

Nome da propriedade	Descrição	Valor da propriedade
RDS_HOSTNAME	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
RDS_PORT	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
RDS_DB_NAME	O nome do banco de dado, <b>ebdb</b> .	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
RDS_USERNAME	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

## Conectar ao banco de dados

Use as informações de conectividade para conectar ao banco de dados, de dentro da aplicação, por meio das variáveis do ambiente. Para obter mais informações sobre como usar o Amazon RDS com seus aplicativos, consulte os tópicos a seguir.

- Java SE: [Conectar a um banco de dados \(plataformas Java SE\)](#)
- Java com Tomcat: [Conectar a um banco de dados \(plataformas Tomcat\)](#)
- Node.js: [Conectar-se a um banco de dados](#)
- .NET: [Conectar-se a um banco de dados](#)

- PHP: [Conexão com banco de dados com um PDO ou MySQLi](#)
- Python: [Conectar-se a um banco de dados](#)
- Ruby: [Conectar-se a um banco de dados](#)

## Configuração de uma instância de banco de dados do RDS integrada, usando o console

É possível visualizar e modificar as configurações de sua instância de banco de dados na seção Database (Banco de dados) na página Configuration (Configuração) do ambiente no [console do Elastic Beanstalk](#).

Como configurar a instância de banco de dados do seu ambiente no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).

Você pode modificar as configurações de Instance class (Classe de instância), Storage (Armazenamento), Password (Senha), Availability (Disponibilidade) e Database deletion policy (política de exclusão de banco de dados) após a criação do banco de dados. Se você alterar a classe de instância, o Elastic Beanstalk provisionará novamente a instância de banco de dados.

Se você não precisa mais do Elastic Beanstalk para associar o banco de dados ao ambiente, pode optar por desacoplá-lo selecionando Decouple database (Desacoplar banco de dados). É importante entender as opções e as considerações envolvidas nessa operação. Para obter mais informações, consulte [the section called “Desacoplamento de uma instância de banco de dados do RDS usando o console”](#).

**i** Aviso

Não modifique as configurações na instância do banco de dados desacoplado fora da funcionalidade que é fornecida pelo Elastic Beanstalk (por exemplo, no console do Amazon RDS). Se fizer isso, a configuração de banco de dados do Amazon RDS poderá ficar fora de sincronia com a definição do ambiente. Quando você atualiza ou reinicia o ambiente, as configurações especificadas no ambiente substituem todas as configurações feitas fora do Elastic Beanstalk.

Se você precisar modificar as configurações não compatíveis diretamente com o Elastic Beanstalk, use [arquivos de configuração](#) do Elastic Beanstalk.

## Configuração de uma instância de banco de dados do RDS integrada usando arquivos de configuração

Você pode configurar a instância de banco de dados do seu ambiente usando [arquivos de configuração](#). Use as opções no namespace [aws:rds:dbinstance](#). O exemplo a seguir modifica o tamanho de armazenamento alocado do banco de dados para 100 GB.

Example `.ebextensions/db-instance-options.config`

```
option_settings:
 aws:rds:dbinstance:
 DBAllocatedStorage: 100
```

Se quiser configurar propriedades de instância de banco de dados não compatíveis com o Elastic Beanstalk, você ainda pode usar um arquivo de configuração e especificar as configurações usando a chave `resources`. O exemplo a seguir define valores para as propriedades `StorageType` e `Iops` do Amazon RDS.

Example `.ebextensions/db-instance-properties.config`

```
Resources:
 AWSEBRDSDatabase:
 Type: AWS::RDS::DBInstance
 Properties:
 StorageType: io1
 Iops: 1000
```

## Desacoplamento de uma instância de banco de dados do RDS usando o console

Você pode desacoplar seu banco de dados de um ambiente do Elastic Beanstalk sem afetar a integridade do ambiente. Considere os seguintes requisitos antes de desacoplar o banco de dados:

- O que deverá acontecer com o banco de dados depois que ele for desacoplado?

Você pode optar por criar um snapshot do banco de dados e depois terminá-lo, manter o banco de dados operacional como um banco de dados autônomo externo ao Elastic Beanstalk ou excluir permanentemente o banco de dados. A configuração Database deletion policy (política de exclusão do banco de dados) determina esse resultado. Para obter uma descrição detalhada das políticas de exclusão, consulte [Ciclo de vida do banco de dados](#) neste mesmo tópico.

- Você precisa fazer algumas alterações nas configurações do banco de dados antes de desacoplá-lo?

Se precisar fazer alterações de configuração no banco de dados, você deverá aplicá-las antes de desacoplar o banco de dados. Isso inclui alterações na Database deletion policy (política de exclusão do banco de dados). Alterações pendentes que sejam enviadas ao mesmo tempo que a configuração Decouple database (Desacoplar banco de dados) serão ignoradas, e somente a configuração de desacoplamento será aplicada.

Para desacoplar uma instância de banco de dados de um ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).

5. Revise todos os valores das configurações na seção Database settings (Configurações do banco de dados), especialmente Database deletion policy (política de exclusão do banco de dados), que determina o que acontece com o banco de dados após ele ser desacoplado.

**Database settings**  
Choose an engine and Instance type for your environment's database.

**Engine**  
mysql

**Engine version**  
--

**Instance class**  
db.t2.micro

**Storage**  
Choose a number between 5 GB and 1024 GB.  
5

**Username**  
test

**Password**  
\*\*\*\*\*

**Availability**  
Low (one AZ)

**Database deletion policy**  
This policy applies when you decouple a database or terminate the environment coupled to it.

- Create snapshot**  
Elastic Beanstalk saves a snapshot of the database and then deletes it. You can restore a database from a snapshot when you add a DB to an Elastic Beanstalk environment or when you create a standalone database. You might incur charges for storing database snapshots.
- Retain**  
The decoupled database will remain available and operational external to Elastic Beanstalk.
- Delete**  
Elastic Beanstalk terminates the database. The database will no longer be available.

Cancel Continue Apply

Se todas as outras configurações estiverem corretas, pule para a Etapa 6 para desacoplar o banco de dados.

### Warning

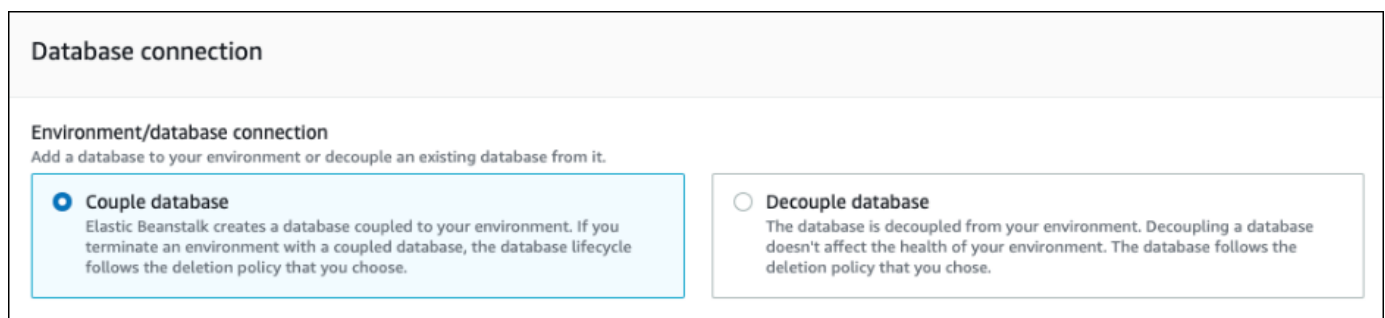
É importante aplicar a configuração Database deletion policy (política de exclusão do banco de dados) separada de Decouple database (Desacoplar banco de dados). Se você selecionar Apply (Aplicar) com a intenção de salvar Decouple database (Desacoplar banco de dados) e uma política de exclusão do banco de dados recém-selecionada, a nova política de exclusão escolhida será ignorada. O Elastic Beanstalk



desacoplará o banco de dados seguindo a política de exclusão definida anteriormente. Se a política de exclusão definida anteriormente for `Delete` ou `Create Snapshot`, você correrá o risco de perder o banco de dados em vez de seguir a política pendente pretendida.

Se alguma das configurações exigir atualizações, faça o seguinte:

1. Faça as modificações necessárias no painel `Database settings` (Configurações do banco de dados).
2. Escolha `Apply` (Aplicar). Salvar as alterações de configuração do banco de dados levará alguns minutos.
3. Volte para a Etapa 3 e escolha `Configuration` (Configuração) no painel de navegação.
6. Vá para a seção `Database connection` (Conexão do banco de dados) do painel.



**Database connection**

**Environment/database connection**  
Add a database to your environment or decouple an existing database from it.

**Couple database**  
Elastic Beanstalk creates a database coupled to your environment. If you terminate an environment with a coupled database, the database lifecycle follows the deletion policy that you choose.

**Decouple database**  
The database is decoupled from your environment. Decoupling a database doesn't affect the health of your environment. The database follows the deletion policy that you chose.

7. Escolha `Decouple database` (Desacoplar banco de dados).
8. Escolha `Apply` (Aplicar) para iniciar a operação de desacoplamento do banco de dados.

A configuração de política de exclusão determina o resultado para o banco de dados e o tempo necessário para desacoplá-lo.

- Se a política de exclusão for definida como `Delete`, o banco de dados será excluído. A operação pode levar aproximadamente de 10 a 20 minutos, dependendo do tamanho do banco de dados.
- Se a política de exclusão for definida como `Snapshot`, um snapshot do banco de dados será criado. Em seguida, o banco de dados será excluído. O tempo necessário para esse processo varia de acordo com o tamanho do banco de dados.

- Se a política de exclusão for definida como `Retain`, o banco de dados permanecerá operacional, fora do ambiente do Elastic Beanstalk. Geralmente, desacoplar um banco de dados leva menos de cinco minutos.

Se decidiu reter o banco de dados externo ao seu ambiente do Elastic Beanstalk, você precisará realizar etapas adicionais para configurá-lo. Para obter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon RDS](#). Se planeja usar o banco de dados que você desacopla para um ambiente de produção, verifique se o tipo de armazenamento usado pelo banco de dados é adequado para sua workload. Para obter mais informações, consulte [DB Instance Storage](#) (Armazenamento de instâncias de banco de dados) e [Modifying a DB instance](#) (Modificar uma instância de banco de dados) no Amazon RDS User Guide (Guia do usuário do Amazon RDS).

## Desacoplamento de uma instância de banco de dados do RDS usando arquivos de configuração

Você pode desacoplar sua instância de banco de dados de um ambiente do Elastic Beanstalk sem afetar a integridade do ambiente. A instância de banco de dados segue a política de exclusão de banco de dados que foi aplicada quando o banco de dados foi desacoplado.

Ambas as opções necessárias para desacoplar o banco de dados estão no namespace [the section called “aws:rds:dbinstance”](#). Eles são os seguintes:

- A opção `DBDeletionPolicy` define a política de exclusão. Pode ser um dos seguintes valores: `Snapshot`, `Delete` ou `Retain`. Esses valores são descritos em [Ciclo de vida do banco de dados](#) neste mesmo tópico.
- A opção `HasCoupledDatabase` determina se o ambiente tem um banco de dados acoplado.
  - Se definida como `true`, o Elastic Beanstalk cria uma nova instância de banco de dados acoplada ao seu ambiente.
  - Se definida como `false`, o Elastic Beanstalk começa a desacoplar a instância de banco de dados do seu ambiente.

Se quiser alterar a configuração do banco de dados antes de desacoplá-la, aplique todas as alterações de configuração primeiro, em uma operação separada. Isso inclui alterar a configuração de `DBDeletionPolicy`. Depois que as alterações forem aplicadas, execute um comando separado para definir a opção de desacoplamento. Se você enviar outras configurações ao mesmo tempo que

a configuração de desacoplamento, as outras configurações da opção serão ignoradas enquanto a configuração de desacoplamento for aplicada.

#### Warning

É importante que você execute os comandos para aplicar as configurações `DBDeletionPolicy` e `HasCoupledDatabase` como duas operações separadas. Se a política de exclusão ativa já estiver definida como `Delete` ou `Snapshot`, você correrá o risco de perder o banco de dados. O banco de dados segue a política de exclusão que está ativa no momento, em vez da política de exclusão pendente que você pretendia.

Para desacoplar uma instância de banco de dados de um ambiente

Siga estas etapas para desacoplar o banco de dados do ambiente do Elastic Beanstalk. Você pode usar a CLI do EB ou a AWS CLI para realizar as etapas. Para obter mais informações, consulte [Personalização avançada do ambiente com arquivos de configuração](#).

1. Se quiser alterar a política de exclusão, configure um arquivo de configuração no formato a seguir. Neste exemplo, a política de exclusão está definida para reter.

#### Example

```
option_settings:
 aws:rds:dbinstance:
 DBDeletionPolicy: Retain
```

2. Execute o comando usando sua ferramenta preferida para atualizar a configuração.
3. Configure um arquivo de configuração para definir `HasCoupledDatabase` como `false`.

#### Example

```
option_settings:
 aws:rds:dbinstance:
 HasCoupledDatabase: false
```

4. Execute o comando usando sua ferramenta preferida para atualizar a configuração.

A configuração de política de exclusão determina o resultado para o banco de dados e o tempo necessário para desacoplá-lo.

- Se a política de exclusão for definida como `Delete`, o banco de dados será excluído. A operação pode levar aproximadamente de 10 a 20 minutos, dependendo do tamanho do banco de dados.
- Se a política de exclusão for definida como `Snapshot`, um snapshot do banco de dados será criado. Em seguida, o banco de dados será excluído. O tempo necessário para esse processo varia de acordo com o tamanho do banco de dados.
- Se a política de exclusão for definida como `Retain`, o banco de dados permanecerá operacional, fora do ambiente do Elastic Beanstalk. Geralmente, desacoplar um banco de dados leva menos de cinco minutos.

Se decidiu reter o banco de dados externo ao seu ambiente do Elastic Beanstalk, você precisará realizar etapas adicionais para configurá-lo. Para obter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon RDS](#). Se planeja usar o banco de dados que você desacopla para um ambiente de produção, verifique se o tipo de armazenamento usado pelo banco de dados é adequado para sua workload. Para obter mais informações, consulte [DB Instance Storage](#) (Armazenamento de instâncias de banco de dados) e [Modifying a DB instance](#) (Modificar uma instância de banco de dados) no Amazon RDS User Guide (Guia do usuário do Amazon RDS).

## Segurança do ambiente AWS Elastic Beanstalk

O Elastic Beanstalk fornece várias opções que controlam o acesso ao serviço (segurança) do ambiente e das instâncias do Amazon EC2 nesse ambiente. Este tópico discute a configuração dessas opções.

### Seções

- [Configurar a segurança do ambiente](#)
- [Namespaces de configuração de segurança do ambiente](#)


## Configurar a segurança do ambiente

É possível modificar a configuração de segurança do ambiente do Elastic Beanstalk no console do Elastic Beanstalk.

Para configurar o acesso ao serviço (segurança) do ambiente no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.

2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Service access (Acesso ao serviço), escolha Edit (Editar).

As seguintes configurações estão disponíveis.

### Configurações

- [Função de serviço](#)
- [EC2 key pair](#)
- [Perfil de instância do IAM](#)

Elastic Beanstalk > Environments > Gettingstarted-env > Configuration

## Configure service access Info

**Service access**  
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

**Service role**  
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

**EC2 key pair**  
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

**EC2 instance profile**  
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

## Função de serviço

Selecione uma [função de serviço](#) a ser associada ao ambiente do Elastic Beanstalk. O Elastic Beanstalk assume a função de serviço quando acessa outros produtos da AWS em seu nome. Para obter mais detalhes, consulte [Gerenciar funções de serviço do Elastic Beanstalk](#).

## EC2 key pair

É possível fazer login com segurança nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) provisionadas para a sua aplicação do Elastic Beanstalk com um par de chaves do Amazon EC2. Para obter instruções sobre como criar um par de chaves, consulte [Criar um par de chaves usando o Amazon EC2](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

### Note

Quando você cria um par de chaves, o Amazon EC2 armazena uma cópia de sua chave pública. Se você não precisar mais usá-la para se conectar a nenhuma instância do ambiente, poderá excluí-la do Amazon EC2. Para obter detalhes, consulte [Excluir um par de chaves](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

Escolha um EC2 key pair (Par de chaves do EC2) no menu suspenso para atribuí-lo às instâncias do ambiente. Quando você atribui um par de chaves, a chave pública é armazenada na instância para autenticar a chave privada, que você armazena localmente. A chave privada nunca é armazenada na AWS.

Para obter mais informações sobre a conexão com instâncias do Amazon EC2, consulte [Conectar-se à instância](#) e [Conectar-se a instâncias Linux/UNIX a partir do Windows usando PuTTY](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

## Perfil de instância do IAM

Um [perfil de instância](#) do EC2 é um perfil do IAM que é aplicado às instâncias iniciadas no ambiente do Elastic Beanstalk. As instâncias do Amazon EC2 assumem a função do perfil da instância para assinar solicitações para a AWS e acessar APIs, por exemplo, para fazer upload de logs no Amazon S3.

Na primeira vez que você cria um ambiente no console do Elastic Beanstalk, o Elastic Beanstalk solicita que você crie um perfil de instância com um conjunto padrão de permissões. Você pode

adicionar permissões a esse perfil para conceder acesso a suas instâncias para outros produtos da AWS. Para obter mais detalhes, consulte [Gerenciar perfis de instância do Elastic Beanstalk](#).

### Note

Anteriormente, o Elastic Beanstalk criava um perfil de instância padrão do EC2 denominado `aws-elasticbeanstalk-ec2-role` a primeira vez que uma conta da AWS criava um ambiente. Esse perfil de instância incluía as políticas gerenciadas padrão. Se sua conta já tiver esse perfil de instância, ele permanecerá disponível para você atribuí-lo aos seus ambientes.

Porém, as diretrizes de segurança recentes da AWS não permitem que um serviço da AWS crie automaticamente perfis com políticas de confiança para outros serviços da AWS, neste caso o EC2. Por causa dessas diretrizes de segurança, o Elastic Beanstalk não cria mais um perfil de instância padrão `aws-elasticbeanstalk-ec2-role`.

## Namespaces de configuração de segurança do ambiente

O Elastic Beanstalk fornece [opções de configuração](#) nos namespaces a seguir para permitir que você personalize a segurança do ambiente:

- [aws:elasticbeanstalk:environment](#): configure a função de serviço do ambiente usando a opção `ServiceRole`.
- [aws:autoscaling:launchconfiguration](#): configure permissões para as instâncias do Amazon EC2 do ambiente usando as opções `EC2KeyName` e `IamInstanceProfile`.

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte [Valores recomendados](#).

## Marcar recursos em seus ambientes do Elastic Beanstalk

Você pode aplicar tags aos ambientes do AWS Elastic Beanstalk. As tags são pares de chave-valor associados a recursos da AWS. Para obter informações sobre a atribuição de tags do recurso do Elastic Beanstalk, casos de uso, restrições de chave e valor de tag, além de tipos de recursos compatíveis, consulte [Marcar recursos da aplicação do Elastic Beanstalk](#).

O Elastic Beanstalk aplica tags de ambiente ao próprio recurso de ambiente, bem como a outros recursos da AWS criados pelo Elastic Beanstalk para o ambiente. É possível usar tags para gerenciar permissões no nível de recurso específico em um ambiente. Para obter mais informações, consulte [Marcar recursos do Amazon EC2](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Por padrão, o Elastic Beanstalk aplica algumas tags ao seu ambiente:

- `elasticbeanstalk:environment-name`: o nome do ambiente.
- `elasticbeanstalk:environment-id`: o ID do ambiente.
- `Name`: o nome do ambiente também. `Name` é usado no painel do Amazon EC2 para identificar e classificar recursos.

Você não pode editar essas etiquetas padrão.

É possível especificar tags ao criar o ambiente do Elastic Beanstalk. Em um ambiente existente, você pode adicionar ou remover tags e atualizar os valores de tags existentes. Um ambiente pode ter até 50 tags, incluindo as tags padrão.

## Adicionar tags durante a criação do ambiente

Ao usar o console do Elastic Beanstalk para criar um ambiente, é possível especificar chaves e valores de tags na página `Modify tags` (Modificar tags) do assistente [Create New Environment \(Criar ambiente\)](#).

Elastic Beanstalk > Applications > getting-started-app

### Modify tags

Apply up to 50 tags to the resources in your environment in addition to the default tags.

Key	Value	
mytag1	value1	Remove

Add tag

49 remaining

Cancel Save



Se você usar a CLI do EB para criar um ambiente, use a opção `--tags` com [eb create](#) para adicionar tags.

```
~/workspace/my-app$ eb create --tags mytag1=value1,mytag2=value2
```

Com a AWS CLI ou outros clientes baseados em API, use o parâmetro `--tags` no comando [create-environment](#).

```
$ aws elasticbeanstalk create-environment \
 --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
 --application-name my-app --environment-name my-env --name-prefix my-app --
 version-label v1 --template-name my-saved-config
```

As [configurações salvas](#) incluem as tags definidas pelo usuário. Quando você aplica uma configuração salva que contém tags durante a criação do ambiente, essas tags são aplicadas ao novo ambiente, desde que você não especifique nenhuma tag nova. Se você adicionar tags a um ambiente usando um dos métodos anteriores, qualquer tag definida na configuração salva será descartada.

## Gerenciar tags de um ambiente existente

É possível adicionar, atualizar e excluir tags em um ambiente existente do Elastic Beanstalk. O Elastic Beanstalk aplica as alterações aos recursos do seu ambiente.

No entanto, não é possível editar as tags padrão que o Elastic Beanstalk aplica a seu ambiente.

Como gerenciar as tags de um ambiente no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, selecione Tags.

A página de gerenciamento de tags mostra a lista de tags que existem atualmente no ambiente.

Elastic Beanstalk > Environments > GettingStartedApp-env > Tags

### Tags for GettingStartedApp-env

Apply up to 47 tags in addition to the default tags to the resources in your environment. You can use tags to group and filter your environments. A tag is a key-value pair. The key must be unique within the environment and is case-sensitive. [Learn more](#)

Key	Value	
elasticbeanstalk:environment-id	e-cubmdjm6ga	
elasticbeanstalk:environment-name	GettingStartedApp-env	
Name	GettingStartedApp-env	
mytag1	value1	Remove
mytag2	value2	Remove

45 remaining

#### 4. Adicione, atualize ou exclua tags:

- Para adicionar uma tag, insira-a nas caixas vazias na parte inferior da lista. Para adicionar outra tag, escolha Add tag (Adicionar tag) e o Elastic Beanstalk adicionará outro par de caixas vazias.
- Para atualizar uma chave ou valor da tag, edite a caixa respectiva na linha da tag.
- Para excluir uma tag, escolha Remove (Remover) ao lado da caixa de valor da tag.

#### 5. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Se você usar a CLI do EB para atualizar o ambiente, use [eb tags](#) para adicionar, atualizar, excluir ou listar tags.

Por exemplo, o comando a seguir lista as tags em seu ambiente padrão.

```
~/workspace/my-app$ eb tags --list
```

O comando a seguir atualiza a tag mytag1 e exclui a tag mytag2.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2
```

Para obter uma lista de opções e mais exemplos, consulte [eb tags](#).

Com a AWS CLI ou outros clientes baseados em API, use o comando [list-tags-for-resource](#) para listar as tags de um ambiente.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
"arn:aws:elasticbeanstalk:us-east-2:my-account-id:environment/my-app/my-env"
```

Use o comando [update-tags-for-resource](#) para adicionar, atualizar ou excluir tags em um ambiente.

```
$ aws elasticbeanstalk update-tags-for-resource \
--tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
--resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:environment/my-
app/my-env"
```

Especifique as tags a serem adicionadas e atualizadas no parâmetro `--tags-to-add` do `update-tags-for-resource`. Uma tag não existente é adicionada, e o valor de uma tag existente é atualizado.

#### Note

Para usar esses dois comandos da AWS CLI com um ambiente do Elastic Beanstalk, você precisa do ARN do ambiente. Você pode recuperar o ARN usando o comando a seguir.

```
$ aws elasticbeanstalk describe-environments
```

## Propriedades de ambientes e outras configurações de software

A página de configuração Configurar atualizações, monitoramento e logs permite configurar o software nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) que executam a aplicação. É possível configurar propriedades do ambiente, armazenamento de logs de instância, streaming e depuração do AWS X-Ray e configurações específicas da plataforma.

### Tópicos

- [Definir configurações específicas da plataforma](#)

- [Configurar propriedades do ambiente \(variáveis de ambiente\)](#)
- [Namespaces de configurações de software](#)
- [Acessar propriedades de ambiente](#)
- [Configurar depuração do AWS X-Ray](#)
- [Visualizar seus logs de ambiente do Elastic Beanstalk](#)

## Definir configurações específicas da plataforma

Além do conjunto padrão de opções disponíveis para todos os ambientes, a maioria das plataformas do Elastic Beanstalk permite que você use configurações específicas de linguagens ou de estruturas de trabalho. Eles aparecem na seção Plataforma de software da página Configurar atualizações, monitoramento e logs e podem assumir as seguintes formas.

- Propriedades predefinidas do ambiente: a Plataforma Ruby utiliza propriedades do ambiente para configurações de estruturas de trabalho, como RACK\_ENV e BUNDLE\_WITHOUT.
- Propriedades de placeholder do ambiente: a plataforma Tomcat define uma propriedade de ambiente chamada JDBC\_CONNECTION\_STRING que não tem nenhum valor definido. Esse tipo de configuração era mais comuns em versões mais antigas da plataforma.
- Opções de configuração: a maioria das plataformas define as [opções de configuração](#) em namespaces específicos da plataforma ou compartilhados, como `aws:elasticbeanstalk:xray` ou `aws:elasticbeanstalk:container:python`.

Como definir configurações específicas da plataforma no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

5. Em Software da plataforma), faça as alterações necessárias na configuração da opção.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Para obter mais informações sobre as opções específicas da plataforma e sobre como obter valores de propriedade de ambiente no seu código, consulte o tópico da plataforma para sua linguagem ou estrutura:

- Docker: [the section called “Configuração de ambientes”](#)
- Go: [Usar a plataforma Go do Elastic Beanstalk](#)
- Java SE: [Usar a plataforma Java SE do Elastic Beanstalk](#)
- Tomcat: [Usar a plataforma Tomcat do Elastic Beanstalk](#)
- .NET Core no Linux: [Usar a plataforma .NET Core no Linux](#)
- .NET: [Usar a plataforma .NET do Elastic Beanstalk](#)
- Node.js: [Usar a plataforma Node.js do Elastic Beanstalk](#)
- PHP: [Usar a plataforma PHP do Elastic Beanstalk](#)
- Python: [Usar a plataforma Python do Elastic Beanstalk](#)
- Ruby: [Usar a plataforma Ruby do Elastic Beanstalk](#)

## Configurar propriedades do ambiente (variáveis de ambiente)

É possível usar as propriedades do ambiente (também conhecidas como variáveis de ambiente) para passar segredos, endpoints, configurações de depuração e outras informações para a aplicação. As propriedades do ambiente ajudam você a executar seu aplicativo em vários ambientes para finalidades diferentes, como desenvolvimento, teste, preparação e produção.

Além disso, quando você [adiciona um banco de dados ao ambiente](#), o Elastic Beanstalk define propriedades do ambiente, como `RDS_HOSTNAME`, que pode ser lido no código da aplicação para criar um objeto ou string de conexão.


### Variáveis de ambiente

Na maioria dos casos, as propriedades de ambientes são transferidas para o seu aplicativo como variáveis de ambiente, mas o comportamento depende da Plataforma. Por exemplo, [a Plataforma Java SE](#) define variáveis de ambiente que você recupera com `System.getenv`, enquanto a [Plataforma Tomcat](#) define as propriedades do sistema Java que você recupera

com `System.getProperty`. Em geral, as propriedades não são visíveis se você se conectar a uma instância e executar `env`.

Como configurar as propriedades do ambiente no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Role para baixo até Propriedades do ambiente.
6. Selecione Adicionar propriedade do ambiente.
7. Insira os pares Nome e Valor da propriedade.
8. Se for necessário adicionar mais variáveis, repita a Etapa 6 e a Etapa 7.
9. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Limites de propriedades de ambiente

- As chaves podem conter caracteres alfanuméricos e os seguintes símbolos: `_ . : / + \ - @`

Os símbolos listados são válidos para chaves de propriedade do ambiente, mas podem não ser válidos para nomes de variáveis de ambiente na plataforma do seu ambiente. Para compatibilidade com todas as plataformas, limite as propriedades do ambiente ao padrão a seguir: `[A-Z_][A-Z0-9_]*`

- Os valores podem conter caracteres alfanuméricos, espaço em branco e os seguintes símbolos: `_ . : / = + \ - @ ' "`

**Note**

Alguns caracteres em valores de propriedades de ambiente devem ser escapados. Use o caractere de barra invertida (\) para representar alguns caracteres especiais e de controle. A lista a seguir inclui exemplos para representar alguns caracteres que precisam ser escapados:

- barra invertida (\): para representar, use \\
- aspas simples ('): para representar, use \'
- aspas duplas ("): para representar, use \"

- As chaves e os valores diferenciam letras maiúsculas de minúsculas.
- O tamanho total de todas as propriedades de ambiente não pode exceder 4.096 bytes quando forem armazenadas como sequências de caracteres com o formato de *chave=valor*.

## Namespaces de configurações de software

Você pode usar um [arquivo de configuração](#) para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. As opções de configuração podem ser definidas pelo serviço Elastic Beanstalk ou pela plataforma utilizada, e são organizadas em namespaces.

É possível usar os [arquivos de configuração](#) do Elastic Beanstalk para definir propriedades do ambiente e opções de configuração em seu código-fonte. Use o [aws:elasticbeanstalk:application:environment namespace](#) para definir propriedades do ambiente.

Example `.ebextensions/options.config`

```
option_settings:
 aws:elasticbeanstalk:application:environment:
 API_ENDPOINT: www.example.com/api
```

Se você usar arquivos de configuração ou modelos do AWS CloudFormation para criar [recursos personalizados](#), poderá usar uma função do AWS CloudFormation para obter informações sobre o recurso e atribuí-lo dinamicamente a uma propriedade do ambiente durante a implantação. O exemplo a seguir do repositório [elastic-beanstalk-samples](#) GitHub usa a [função Ref](#) para obter o ARN

de um tópico do Amazon SNS que ele cria, e o atribui a uma propriedade de ambiente denominada NOTIFICATION\_TOPIC.

### Observações

- Se você usar uma função do AWS CloudFormation para definir uma propriedade de ambiente, o console do Elastic Beanstalk exibirá o valor da propriedade antes que a função seja avaliada. Você pode usar o [script de plataforma get-config](#) para confirmar os valores das propriedades do ambiente que estão disponíveis para seu aplicativo.
- A plataforma [Docker de vários contêineres](#) não usa o AWS CloudFormation para criar recursos de contêiner. Como resultado, essa Plataforma não oferece suporte para definir as propriedades de ambiente usando as funções AWS CloudFormation.

### Example [.Ebextensions/sns-topic.config](#)

```
Resources:
 NotificationTopic:
 Type: AWS::SNS::Topic

option_settings:
 aws:elasticbeanstalk:application:environment:
 NOTIFICATION_TOPIC: `{"Ref" : "NotificationTopic"}`
```

Você também pode usar esse recurso para propagar informações de [pseudoparâmetros do AWS CloudFormation](#). Este exemplo obtém a região atual e a atribui a uma propriedade chamada AWS\_REGION.

### Example [.Ebextensions/env-regionname.config](#)

```
option_settings:
 aws:elasticbeanstalk:application:environment:
 AWS_REGION: `{"Ref" : "AWS::Region"}`
```

A maioria das plataformas do Elastic Beanstalk define namespaces adicionais com opções para configurar o software que é executado na instância, como o proxy inverso que retransmite solicitações para sua aplicação. Para obter mais informações sobre os namespaces disponíveis para a sua plataforma, consulte a seguir:



- Go: [Namespace de configuração do Go](#)
- Java SE: [Namespace de configuração do Java SE](#)
- Tomcat: [Namespaces de configuração do Tomcat](#)
- .NET Core no Linux: [Namespace de configuração do .NET Core no Linux](#)
- .NET: [O namespace aws:elasticbeanstalk:container:dotnet:appool](#)
- Node.js: [Namespace de configuração do Node.js](#)
- PHP: [O namespace aws:elasticbeanstalk:container:php:phpini](#)
- Python: [Namespaces de configuração do Python](#)
- Ruby: [Namespaces de configuração do Ruby](#)

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte [Opções de configuração](#) para obter mais informações.

## Acessar propriedades de ambiente

Na maioria dos casos, você acessa as propriedades do ambiente no código do seu aplicativo, como uma variável de ambiente. Porém, em geral, as propriedades do ambiente são transferidas apenas para o aplicativo e não podem ser visualizadas conectando uma instância em seu ambiente e executando-a env.

- [Go](#): `os.Getenv`

```
endpoint := os.Getenv("API_ENDPOINT")
```

- [Java SE](#): `System.getenv`

```
String endpoint = System.getenv("API_ENDPOINT");
```

- [Tomcat](#): `System.getProperty`

```
String endpoint = System.getProperty("API_ENDPOINT");
```

- [.NET Core no Linux](#): `Environment.GetEnvironmentVariable`

```
string endpoint = Environment.GetEnvironmentVariable("API_ENDPOINT");
```

- [.NET](#): `appConfig`

```
NameValueCollection appConfig = ConfigurationManager.AppSettings;
string endpoint = appConfig["API_ENDPOINT"];
```

- [Node.js](#): `process.env`

```
var endpoint = process.env.API_ENDPOINT
```

- [PHP](#): `$_SERVER`

```
$endpoint = $_SERVER['API_ENDPOINT'];
```

- [Python](#): `os.environ`

```
import os
endpoint = os.environ['API_ENDPOINT']
```

- [Ruby](#): `ENV`

```
endpoint = ENV['API_ENDPOINT']
```

Fora do código do aplicativo, como em um script que é executado durante a implantação, você pode acessar as propriedades do ambiente com a [get-config Plataforma de script](#). Consulte o repositório GitHub do [elastic-beanstalk-samples](#) para obter exemplos de configurações que usam `get-config`.

## Configurar depuração do AWS X-Ray

Use o console do AWS Elastic Beanstalk ou um arquivo de configuração para executar o daemon do AWS X-Ray nas instâncias do ambiente. O X-Ray é um produto da AWS que coleta dados sobre as solicitações que o aplicativo atende, e os usa para construir um mapa de serviços que pode ser usado para identificar problemas no aplicativo e oportunidades de otimização.

### Note

Algumas regiões não oferecem o X-Ray. Se você criar um ambiente em uma dessas regiões, não poderá executar o daemon do X-Ray nas instâncias no ambiente.

Para obter informações sobre os serviços da AWS oferecidos em cada região, consulte a [Tabela de regiões](#).



O X-Ray fornece um SDK que você pode usar para instrumentar o código da aplicação, e uma aplicação daemon que transmite as informações de depuração do SDK para a API do X-Ray.

### Plataformas compatíveis

É possível usar o SDK do X-Ray com as seguintes plataformas do Elastic Beanstalk:

- Go - versão 2.9.1 e posterior
- Java 8 – versão 2.3.0 e posterior
- Java 8 com Tomcat 8 – versão 2.4.0 e posterior
- Node.js - versão 3.2.0 e posterior

- Windows Server – todas as versões de plataforma lançadas em 18 de dezembro de 2016 ou posteriormente
- Python – versão 2.5.0 e posterior

Nas plataformas compatíveis, é possível usar uma opção de configuração para executar o daemon do X-Ray nas instâncias do ambiente. Você pode habilitar o daemon no [console do Elastic Beanstalk](#) ou usando um [arquivo de configuração](#).

Para fazer upload de dados no X-Ray, o daemon do X-Ray exige permissões do IAM na política gerenciada AWSXrayWriteOnlyAccess. Essas permissões são incluídas no [perfil da instância do Elastic Beanstalk](#). Se você não usa o perfil de instância padrão, consulte [Conceder Permissão do Daemon para Enviar Dados ao X-Ray](#) no Guia do desenvolvedor do AWS X-Ray.

A depuração com o X-Ray requer o uso do SDK do X-Ray. Consulte os [Conceitos básicos do AWS X-Ray](#) no Guia do Desenvolvedor do AWS X-Ray para obter instruções e aplicações de amostra.

Se você usa uma versão de plataforma que não inclui o daemon, ainda é possível executá-lo com um script em um arquivo de configuração. Para obter mais informações, consulte [Fazer Download e Executar o Daemon do X-Ray Manualmente \(Avançado\)](#) no Guia do desenvolvedor do AWS X-Ray.

## Seções

- [Configurar depuração](#)
- [O namespace aws:elasticbeanstalk:xray](#)

## Configurar depuração

É possível habilitar o daemon do X-Ray em um ambiente em execução no console do Elastic Beanstalk.

Como habilitar a depuração no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

**Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Na seção Amazon X-Ray, selecione Ativado.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Essa opção pode ser ativada também durante a criação do ambiente. Para obter mais informações, consulte [O assistente de criação de novo ambiente](#).

## O namespace aws:elasticbeanstalk:xray

Use a opção XRayEnabled no namespace aws:elasticbeanstalk:xray para habilitar a depuração.

Para habilitar a depuração automaticamente quando você implanta o aplicativo, defina a opção em um [arquivo de configuração](#) no código-fonte, como a seguir.

Example .ebextensions/debugging.config

```
option_settings:
 aws:elasticbeanstalk:xray:
 XRayEnabled: true
```

## Visualizar seus logs de ambiente do Elastic Beanstalk

O AWS Elastic Beanstalk fornece duas maneiras de visualizar regularmente os logs das instâncias do Amazon EC2 que executam sua aplicação:

- Configure seu ambiente do Elastic Beanstalk para fazer upload de logs de instância alternados para o bucket do Amazon S3 do ambiente.
- Configure o ambiente para fazer streaming de logs de instâncias para o Amazon CloudWatch Logs.

Quando você configura o streaming de logs de instâncias para o CloudWatch Logs, o Elastic Beanstalk cria grupos de logs do CloudWatch Logs para logs de proxy e implantação nas instâncias do Amazon EC2 e transfere esses arquivos de log para o CloudWatch Logs em tempo real. Para obter mais informações sobre logs de instância, visite [Visualizar logs de instâncias do Amazon EC2 no ambiente do Elastic Beanstalk](#).

Além de logs da instância, se você habilitar a [integridade avançada](#) para seu ambiente, poderá configurar o ambiente para fazer streaming de informações de integridade para o CloudWatch Logs. Quando o status de integridade do ambiente for alterado, o Elastic Beanstalk adicionará um registro a um grupo de logs de integridade com o novo status e uma descrição da causa da alteração. Para obter informações sobre o streaming de integridade do ambiente, visite [Fazer streaming de informações de integridade do ambiente do Elastic Beanstalk para o Amazon CloudWatch Logs](#).

## Configurar visualização de logs de instância

Para visualizar logs de instância, você pode habilitar a alternância de logs de instância e o streaming de logs no console do Elastic Beanstalk.

Como configurar a alternância e o streaming de logs de instância no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Na seção Armazenamento de logs do S3, selecione Ativado embaixo de Alternar logs para habilitar o upload de logs alternados para o Amazon S3.
6. na seção Instance log streaming to CloudWatch Logs (Streaming de logs de instância para logs do CloudWatch), defina as seguintes configurações:
  - Streaming de logs: selecione Ativado para habilitar o streaming de logs.
  - Retention (Retenção): especifique o número de dias para reter os logs no CloudWatch Logs.

- Lifecycle (Ciclo de vida): defina como Delete logs upon termination (Excluir logs após o encerramento) para excluir os logs do CloudWatch Logs imediatamente se o ambiente for encerrado, em vez de esperar que eles expirem.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Depois de ativar o streaming de logs, você poderá voltar para a categoria ou a página de configuração Software para encontrar o link Log Groups (Grupos de logs). Clique nesse link para ver os logs de instância no console do CloudWatch.

## Configurar visualização do log de integridade do ambiente

Para visualizar logs de integridade do ambiente, é possível habilitar o streaming de logs de integridade do ambiente no console do Elastic Beanstalk.

Como configurar o streaming de logs de integridade do ambiente no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Vá para a seção Monitoramento.
6. Em Health event streaming to CloudWatch Logs (Streaming de eventos de integridade para logs do CloudWatch), defina as seguintes configurações:
  - Streaming de logs: escolha Ativado para habilitar o streaming de logs.
  - Retention (Retenção): especifique o número de dias para reter os logs no CloudWatch Logs.
  - Lifecycle (Ciclo de vida): defina como Delete logs upon termination (Excluir logs após o encerramento) para excluir os logs do CloudWatch Logs imediatamente se o ambiente for encerrado, em vez de esperar que eles expirem.

7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

## Namespaces de visualização de logs

Os seguintes namespaces contêm configurações para visualização de logs:

- [aws:elasticbeanstalk:hostmanager](#): configure o upload de logs alternados para o Amazon S3.
- [aws:elasticbeanstalk:cloudwatch:logs](#): configure o streaming de logs de instância para o CloudWatch.
- [aws:elasticbeanstalk:cloudwatch:logs:health](#): configure o streaming de integridade do ambiente para o CloudWatch.

## Notificações do ambiente do Elastic Beanstalk com o Amazon SNS

É possível configurar o ambiente AWS Elastic Beanstalk do Elastic Beanstalk para usar o Amazon Simple Notification Service (Amazon SNS) para notificá-lo sobre eventos importantes que afetam a aplicação. Para receber e-mails da AWS sempre que ocorrer um erro ou a integridade do seu ambiente mudar, especifique um endereço de e-mail durante ou após a criação do ambiente.

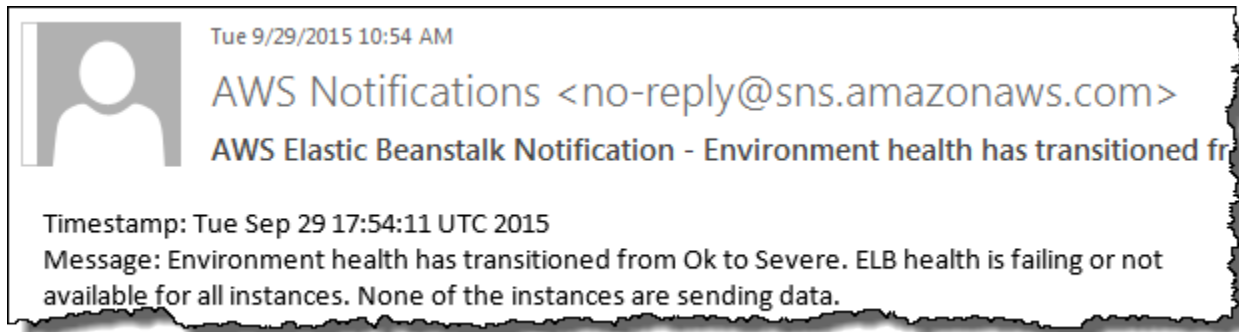
### Note

O Elastic Beanstalk usa o Amazon SNS para notificações. Para obter informações sobre a definição de preço do Amazon SNS, consulte <https://aws.amazon.com/sns/pricing/>.

Quando você configura notificações para o seu ambiente, o Elastic Beanstalk cria um tópico do Amazon SNS para ele em seu nome. Para enviar mensagens para um tópico do Amazon SNS, o Elastic Beanstalk precisa ter a permissão necessária. Para obter mais informações, consulte [Configurar permissões para enviar notificações](#).

Quando ocorre um [evento](#) notável, o Elastic Beanstalk envia uma mensagem para o tópico. Em seguida, o Amazon SNS retransmite as mensagens recebidas para os assinantes do tópico. Os eventos notáveis incluem erros de criação do ambiente e qualquer alteração na [integridade do ambiente e da instância](#). Eventos para as operações do Amazon EC2 Auto Scaling (como adicionar e remover instâncias do ambiente) e outros eventos informativos não acionam notificações.





Você pode inserir um endereço de e-mail no console do Elastic Beanstalk quando criar um ambiente ou em outro momento posterior. Isso criará um tópico do Amazon SNS e você estará inscrito nele. O Elastic Beanstalk gerencia o ciclo de vida do tópico e o exclui quando o ambiente é encerrado ou quando você remove seu endereço de e-mail no [console de gerenciamento do ambiente](#).

O namespace do `aws:elasticbeanstalk:sns:topics` oferece opções para configurar um tópico do Amazon SNS usando arquivos de configuração, uma CLI ou um SDK. Usando um desses métodos, você pode configurar o tipo de assinante e o endpoint. Para o tipo de assinante, você pode escolher uma fila do Amazon SQS ou uma URL HTTP.

Só é possível ativar e desativar notificações do Amazon SNS. A frequência de notificações enviadas ao tópico pode ser alta, dependendo do tamanho e da composição do seu ambiente. Há outras opções para configurar notificações a serem enviadas em circunstâncias específicas. Com o Amazon EventBridge, é possível [configurar regras orientadas por eventos](#) que notificam você quando o Elastic Beanstalk emite eventos que atendem a critérios específicos. Ou, alternativamente, você pode [configurar o seu ambiente para publicar métricas personalizadas](#) e [definir alarmes do Amazon CloudWatch](#) para avisar você quando essas métricas atingirem um limite não íntegro.

## Configurar notificações usando o console do Elastic Beanstalk

Você pode inserir um endereço de e-mail no console do Elastic Beanstalk para criar um tópico do Amazon SNS para o seu ambiente.

Como configurar notificações no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

**Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Role para baixo até a seção Notificações por e-mail.
6. Insira um endereço de e-mail.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Quando você insere um endereço de e-mail para notificações, o Elastic Beanstalk cria um tópico do Amazon SNS para seu ambiente e adiciona uma inscrição. O Amazon SNS envia um e-mail para o endereço inscrito para confirmar a inscrição. Você deve clicar no link do e-mail de confirmação para ativar a inscrição e receber notificações.

## Configurar notificações usando opções de configuração

Use as opções no [namespace `aws:elasticbeanstalk:sns:topics`](#) para configurar as notificações do Amazon SNS para o seu ambiente. Você pode definir estas opções usando [arquivos de configuração](#), uma CLI ou um SDK.

- **Endpoint da notificação:** o endereço de e-mail, a fila do Amazon SQS ou a URL para enviar as notificações. Definir essa opção cria uma fila do SQS e uma inscrição para o endpoint especificado. Se o endpoint não for um endereço de e-mail, você deve definir também a opção `Notification Protocol`. O SNS valida o valor de `Notification Endpoint` de acordo com o valor de `Notification Protocol`. A configuração dessa opção várias vezes cria inscrições adicionais no tópico. Se você remover esta opção, o tópico será excluído.
- **Protocolo de notificação:** protocolo usado para enviar notificações ao `Notification Endpoint`. Essa opção usa como valor padrão `email`. Defina essa opção como `email-json` para enviar e-mails em formato JSON, `http` ou `https` para publicar notificações no formato JSON em um endpoint HTTP, ou `sqs` para enviar notificações para uma fila do SQS.

**Note**

Notificações do AWS Lambda não são suportadas.

- ARN do tópico de notificação ARN: depois de definir um endpoint de notificação para seu ambiente, leia esta configuração para obter o ARN do tópico do SNS. Você também pode definir esta opção para usar um tópico do SNS existente para notificações. Um tópico anexado ao seu ambiente por meio dessa opção não é excluído quando ela é alterada ou se o ambiente for encerrado.

Para configurar as notificações do Amazon SNS, você precisa ter as permissões necessárias. Se o seu usuário do IAM usar a política de usuário gerenciado [AdministratorAccess-AWSElasticBeanstalk](#) do Elastic Beanstalk, é provável que você já tenha as permissões necessárias para configurar o tópico padrão do Amazon SNS que o Elastic Beanstalk cria para o seu ambiente. Contudo, se você configurar um tópico do Amazon SNS não gerenciado pelo Elastic Beanstalk, será necessário adicionar a política a seguir à sua função de usuário.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "sns:SetTopicAttributes",
 "sns:GetTopicAttributes",
 "sns:Subscribe",
 "sns:Unsubscribe",
 "sns:Publish"
],
 "Resource": [
 "arn:aws:sns:us-east-2:123456789012:sns_topic_name"
]
 }
]
}
```

- Nome do tópico de identificação: defina esta opção para personalizar o nome do tópico do Amazon SNS usado para notificações de ambiente. Se já houver um tópico com o mesmo nome, o Elastic Beanstalk anexará o tópico ao ambiente.

**⚠ Warning**

Se você anexar um tópico do SNS existente a um ambiente com `Notification Topic Name`, o Elastic Beanstalk excluirá o tópico quando você encerrar o ambiente ou alterar essa configuração futuramente.

Se você alterar essa opção, o `Notification Topic ARN` também será alterado. Se um tópico já está conectado ao ambiente, o Elastic Beanstalk exclui o antigo e cria um novo tópico e assinatura.

Ao usar um nome de tópico personalizado, você também deve fornecer um ARN de um tópico personalizado criado externamente. A política de usuário gerenciado não detecta automaticamente um tópico com um nome personalizado, portanto, você deve fornecer permissões personalizadas do Amazon SNS aos usuários do IAM. Use uma política semelhante à que foi utilizada para um ARN de tópico personalizado, mas faça as inclusões a seguir:

- Inclua mais duas ações na lista de `Actions`, especificamente: `sns:CreateTopic`, `sns>DeleteTopic`
- Se estiver alterando o nome `Notification Topic Name` de um tópico personalizado para outro, você também deve incluir os ARNs de ambos os tópicos na lista `Resource`. Com alternativa, inclua uma expressão regular que abranja ambos os tópicos. Dessa maneira, o Elastic Beanstalk tem permissões para excluir o tópico antigo e criar o novo.

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte [Valores recomendados](#).

## Configurar permissões para enviar notificações

Esta seção discute considerações de segurança relacionadas a notificações que utilizam o Amazon SNS. Há dois casos distintos:

- Use o tópico padrão do Amazon SNS criado pelo Elastic Beanstalk para o seu ambiente.
- Forneça um tópico externo do Amazon SNS por meio de opções de configuração.

A política de acesso padrão para tópicos do Amazon SNS permite que apenas o proprietário do tópico publique nele ou assine ele. Contudo, com a configuração de política adequada, o Elastic Beanstalk pode receber permissão para publicar em um tópico do Amazon SNS em um dos dois casos descritos nesta seção. As subseções a seguir fornecem mais informações.

## Permissões para um tópico padrão

Quando você configura notificações para o ambiente, o Elastic Beanstalk cria um tópico do Amazon SNS para ele. Para enviar mensagens para um tópico do Amazon SNS, o Elastic Beanstalk precisa ter a permissão necessária. Se o seu ambiente usa a [função de serviço](#) que o console do Elastic Beanstalk ou a CLI do EB geraram para ele, ou a [função vinculada ao serviço de monitoramento](#) da sua conta, você não precisará fazer mais nada. Essas funções gerenciadas incluem a permissão necessária que permite que o Elastic Beanstalk envie mensagens para o tópico do Amazon SNS.

No entanto, se você tiver fornecido uma função de serviço personalizada ao criar o seu ambiente, certifique-se de que ela inclui a política a seguir.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "sns:Publish"
],
 "Resource": [
 "arn:aws:sns:us-east-2:123456789012:ElasticBeanstalkNotifications*"
]
 }
]
}
```

## Permissões para um tópico externo

O [Configurar notificações usando opções de configuração](#) explica como substituir o tópico do Amazon SNS que o Elastic Beanstalk fornece por outro tópico do Amazon SNS. Se você substituiu o tópico, o Elastic Beanstalk deverá verificar se você tem permissão para publicar neste tópico do SNS para poder associar o tópico do SNS ao ambiente. Você deve ter `sns:Publish`. A função de serviço usa a mesma permissão. Para verificar se esse é o caso, o Elastic Beanstalk envia uma notificação de teste ao SNS como parte da sua ação para criar ou atualizar o ambiente. Se este teste

falhar, sua tentativa de criar ou atualizar o ambiente também falhará. O Elastic Beanstalk exibe uma mensagem que explica o motivo dessa falha.

Se você fornecer uma função de serviço personalizada para o seu ambiente, certifique-se de que a sua função de serviço personalizada inclua a política a seguir para permitir que o Elastic Beanstalk envie mensagens para o tópico do Amazon SNS. No código a seguir, substitua *sns\_topic\_name* pelo nome do tópico do Amazon SNS fornecido nas opções de configuração.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "sns:Publish"
],
 "Resource": [
 "arn:aws:sns:us-east-2:123456789012:sns_topic_name"
]
 }
]
}
```

Para obter mais informações sobre o controle de acesso do Amazon SNS, consulte [Casos de exemplo para o controle de acesso do Amazon SNS](#) no Guia do desenvolvedor do Amazon Simple Notification Service.

## Configurar a Amazon Virtual Private Cloud (Amazon VPC) com o Elastic Beanstalk

A [Amazon Virtual Private Cloud](#) (Amazon VPC) é o serviço de rede que direciona o tráfego com segurança para as instâncias do EC2 que executam sua aplicação no Elastic Beanstalk. Se você não configurar uma VPC ao executar o ambiente, o Elastic Beanstalk usará a VPC padrão.

Você pode executar seu ambiente em uma VPC personalizada para customizar as configurações de rede e de segurança. Com o Elastic Beanstalk, é possível escolher quais sub-redes serão usadas nos seus recursos e como configurar endereços IP para as instâncias e o balanceador de carga no ambiente. Um ambiente é bloqueado para uma VPC quando você o cria. No entanto, é possível alterar as configurações de sub-rede e de endereço IP em um ambiente em execução.

**Note**

Se você criou a conta da AWS antes de 4 de dezembro de 2013, talvez tenha ambientes que usem a configuração de rede do Amazon EC2-Classik em algumas regiões da AWS em vez da Amazon VPC. Para obter informações sobre como migrar os ambientes de uma configuração de rede do EC2-Classik para uma VPC, consulte [Migrar ambientes do Elastic Beanstalk do EC2-Classik para uma VPC](#).

## Definir as configurações da VPC no console do Elastic Beanstalk

Se você escolheu uma VPC personalizada ao criar seu ambiente, poderá modificar as configurações de VPC no console do Elastic Beanstalk.

Para definir as configurações da VPC do seu ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

**Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Network (Rede), escolha Edit (Editar).

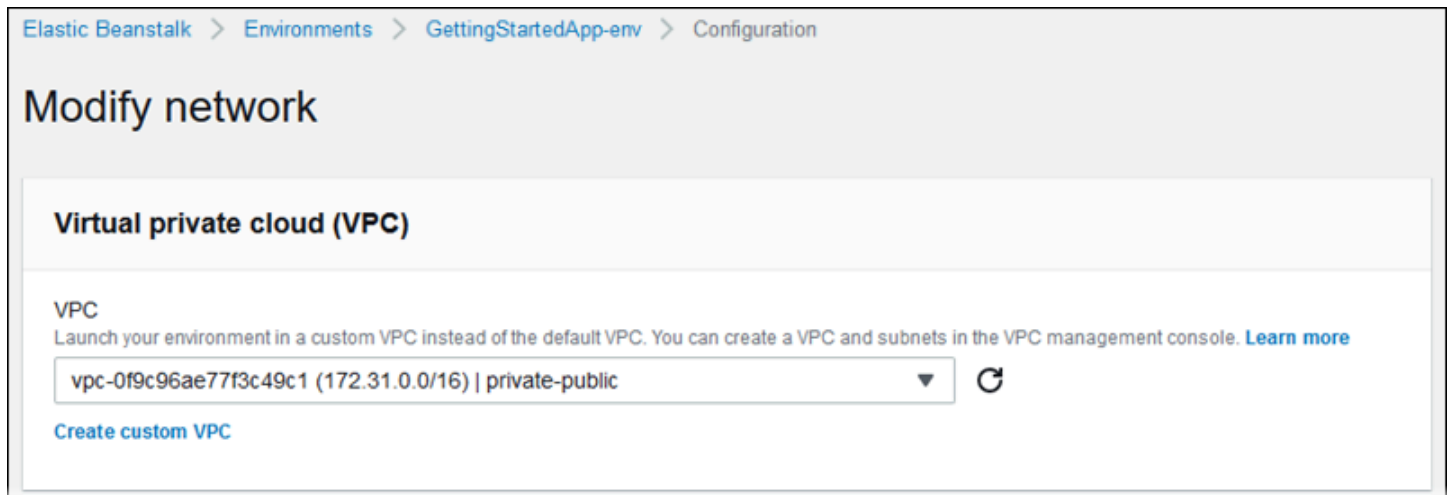
As seguintes configurações estão disponíveis.

### Opções

- [VPC](#)
- [Visibilidade do load balancer](#)
- [Sub-redes do load balancer](#)
- [Endereço IP público da instância](#)
- [Sub-redes de instância](#)
- [Sub-redes do banco de dados](#)

## VPC

Escolha uma VPC em seu ambiente. Você só pode alterar essa configuração durante a criação do ambiente.



## Visibilidade do load balancer

Para um ambiente com balanceamento de carga, escolha o esquema do load balancer. Por padrão, o load balancer é público e apresenta um endereço IP público e um nome de domínio. Se seu aplicativo somente veicular tráfego de dentro da sua VPC ou VPN conectada, desmarque essa opção e escolha as sub-redes privadas para que seu load balancer torná-lo interno e desative o acesso da Internet.

## Sub-redes do load balancer

Para um ambiente com balanceamento de carga, escolha as sub-redes que o load balancer usa para veicular o tráfego. Para um aplicativo público, escolha sub-redes públicas. Use sub-redes em várias zonas de disponibilidade para alta disponibilidade. Para um aplicativo interno, escolha sub-redes privadas e desative a visibilidade do load balancer.



## Load balancer settings

Assign your load balancer to a subnet in each Availability Zone (AZ) in which your application runs. For a publicly accessible application, set **Visibility** to **Public** and choose public subnets.

### Visibility

Make your load balancer internal if your application serves requests only from connected VPCs. Public load balancers serve requests from the Internet.

Public ▼

### Load balancer subnets

<input type="checkbox"/>	Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/>	us-east-2a	subnet-04a707767b8ca8023	172.31.0.0/24	public-a
<input type="checkbox"/>	us-east-2a	subnet-0c559ebeb1a89adb4	172.31.3.0/24	private-a
<input checked="" type="checkbox"/>	us-east-2b	subnet-034a813125cd2077a	172.31.2.0/24	private-b
<input type="checkbox"/>	us-east-2b	subnet-09a24e24e7f7359fa	172.31.1.0/24	public-b

## Endereço IP público da instância

Se você escolher sub-redes públicas para suas instâncias de aplicativos, ative endereços IP públicos para torná-los roteáveis a partir da Internet.

## Sub-redes de instância

Escolha sub-redes para suas instâncias de aplicativo. Escolha pelo menos uma sub-rede para cada zona de disponibilidade que seu load balancer usa. Se você escolher sub-redes privadas para suas instâncias, sua VPC precisará ter um gateway NAT em uma sub-rede pública que as instâncias possam usar para acessar a Internet.

### Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances.

Public IP address  
Assign a public IP address to the Amazon EC2 instances in your environment.

#### Instance subnets

	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	us-east-2a	subnet-04a707767b8ca8023	172.31.0.0/24	public-a
<input checked="" type="checkbox"/>	us-east-2a	subnet-0c559eeeb1a89adb4	172.31.3.0/24	private-a
<input type="checkbox"/>	us-east-2b	subnet-034a813125cd2077a	172.31.2.0/24	private-b
<input checked="" type="checkbox"/>	us-east-2b	subnet-09a24e24e7f7359fa	172.31.1.0/24	public-b

Cancel Continue Apply

## Sub-redes do banco de dados

Quando você executa um banco de dados do Amazon RDS anexado ao ambiente do Elastic Beanstalk, escolha as sub-redes para suas instâncias de banco de dados. Para alta disponibilidade, torne o banco de dados multi-AZ e escolha uma sub-rede para cada zona de disponibilidade. Para garantir que seu aplicativo possa se conectar ao banco de dados, execute ambos nas mesmas sub-redes.

## O namespace `aws:ec2:vpc`

Você pode usar as opções de configuração no namespace [aws:ec2:vpc](#) para definir as configurações de rede do seu ambiente.

O seguinte [arquivo de configuração](#) usa opções neste namespace para definir a VPC e as sub-redes do ambiente em uma configuração público-privada. Para definir o ID da VPC em um arquivo de configuração, o arquivo precisará ser incluído no pacote de origem do aplicativo durante a criação do ambiente. Consulte [Definir as opções de configuração durante a criação do ambiente](#) para outros métodos de definição dessas configurações durante a criação do ambiente.

## Example .ebextensions/vpc.config: público-privado

```
option_settings:
 aws:ec2:vpc:
 VPCId: vpc-087a68c03b9c50c84
 AssociatePublicIpAddress: 'false'
 ELBScheme: public
 ELBSubnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
 Subnets: subnet-026c6117b178a9c45,subnet-0839e902f656e8bd1
```

Este exemplo mostra uma configuração público-pública, em que o load balancer e as instâncias do EC2 são executadas nas mesmas sub-redes públicas.

## Example .ebextensions/vpc.config: público-privado

```
option_settings:
 aws:ec2:vpc:
 VPCId: vpc-087a68c03b9c50c84
 AssociatePublicIpAddress: 'true'
 ELBScheme: public
 ELBSubnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
 Subnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
```

## Migrar ambientes do Elastic Beanstalk do EC2-Classic para uma VPC

Este tópico descreve as diferentes opções para migrar seus ambientes do Elastic Beanstalk de uma plataforma de rede EC2-Classic para uma rede [Amazon Virtual Private Cloud](#) (Amazon VPC).

Se você criou a conta da AWS antes de 4 de dezembro de 2013, talvez tenha ambientes que usem a configuração da rede EC2-Classic em algumas regiões da Regiões da AWS. Todas as contas da AWS criadas a partir de 4 de dezembro de 2013 já têm uma VPC padrão em todas as regiões da AWS. As únicas exceções são se o Amazon EC2-Classic foi habilitado como resultado de uma solicitação de suporte.

### Note

Você pode visualizar as definições da configuração de rede do ambiente na categoria Network configuration (Configuração de rede) na página [Configuration overview](#) (Visão geral da configuração) do [console do Elastic Beanstalk](#).

## Por que você deve migrar

O Amazon EC2-Classic atingirá o fim do suporte padrão em 15 de agosto de 2022. Para evitar interrupções em suas workloads, recomendamos que você migre do Amazon EC2-Classic para uma VPC antes de 15 de agosto de 2022. Também solicitamos que você não inicie nenhum recurso da AWS no Amazon EC2-Classic no futuro e, em vez disso, use a Amazon VPC.

Quando migrar seus ambientes Elastic Beanstalk do Amazon EC2-Classic para a Amazon VPC, você deve criar uma nova conta da AWS. Você também deve recriar seus ambientes do AWS EC2-Classic em sua nova conta da AWS. Não é necessário fazer nenhum trabalho adicional de configuração de seus ambientes para usar a VPC padrão. Se a VPC padrão não atender aos seus requisitos, você pode criar uma VPC personalizada manualmente e associá-la aos seus ambientes.

Como alternativa, se a conta da AWS existente tiver recursos que não podem ser migrados para uma nova conta da AWS, você pode adicionar uma VPC à conta atual. Depois, configure seus ambientes para usar a VPC.

Para obter mais informações, consulte a postagem do blog [EC2-Classic Networking is Retiring - Here's How to Prepare \(O EC2-Classic está sendo descontinuado - Veja como se preparar\)](#).

## Migrar um ambiente do EC2-Classic para uma nova conta da AWS (recomendado)

Se você ainda não tem uma conta da AWS criada a partir de 4 de dezembro de 2013, crie uma nova conta. Você migrará seus ambientes para essa nova conta.

1. A nova conta da AWS fornece uma VPC padrão para os ambientes da AWS. Se você não precisar criar uma VPC personalizada, vá para a etapa 2.

É possível criar uma VPC personalizada de uma das seguintes maneiras:

- Crie uma VPC rapidamente usando o assistente do console da Amazon VPC, como uma das opções de configuração disponíveis. Para obter mais informações, consulte [Configurações do assistente do console da Amazon VPC](#).
- Crie uma VPC personalizada no console da Amazon VPC se você tem requisitos mais específicos para a sua VPC. Recomendamos que você faça isso, por exemplo, se seu caso de uso exigir um número específico de sub-redes. Para obter mais informações, consulte [VPCs e sub-redes](#).
- Crie uma VPC usando o repositório [elastic-beanstalk-samples](#) no site do GitHub se preferir usar modelos do AWS CloudFormation com os seus ambientes do Elastic Beanstalk. Este

repositório inclui modelos de AWS CloudFormation. Para obter mais informações, consulte . [Usar o Elastic Beanstalk com Amazon VPC](#).

### Note

Você também pode criar uma VPC personalizada ao mesmo tempo que recria o ambiente na nova conta da AWS usando o [assistente de criação de ambiente](#). Se você usar o assistente e optar por criar uma VPC personalizada, o assistente redireciona você para o console da Amazon VPC.

2. Em sua nova conta da AWS, crie um novo ambiente. Recomendamos que o ambiente inclua a mesma configuração de ambiente existente na conta da AWS da qual você está migrando. Isso pode ser feito usando uma das abordagens a seguir.

### Note

Caso o novo ambiente deva continuar usando o mesmo CNAME após a migração, termine o ambiente original na plataforma EC2-Classic. Isso libera o CNAME para uso. Entretanto, fazer isso pode causar tempo de inatividade para esse ambiente, e há o risco de que outro cliente selecione o CNAME no período entre o término do ambiente do EC2-Classic e a criação do novo ambiente. Para obter mais informações, consulte . [Encerrar um ambiente do Elastic Beanstalk](#).

Para ambientes que têm seu próprio nome de domínio proprietário, o CNAME não tem esse problema. Você pode apenas atualizar o Domain Name System (DNS) para encaminhar solicitações ao novo CNAME.

- Use o [assistente de criação de novo ambiente](#) no [console do Elastic Beanstalk](#). O assistente fornece uma opção para criar uma VPC personalizada. Se você não optar por criar uma VPC personalizada, uma VPC padrão é atribuída.
- Use a interface de linha de comando do Elastic Beanstalk (EB CLI) para recriar o seu ambiente na nova conta da AWS. Um dos [exemplos](#) na descrição do comando `eb create` demonstra a criação de um ambiente em uma VPC personalizada. Se você não fornecer o ID de uma VPC, o ambiente usa a VPC padrão.

Usando essa abordagem, você pode usar um arquivo de configurações salvo nas duas contas da AWS. Assim, você não precisa inserir manualmente todas as informações

de configuração. Mas, você deve salvar as definições de configuração para o ambiente do EC2-Classico que está migrando com o comando [eb config save](#). Copie o arquivo de configuração salvo em um novo diretório para o ambiente da nova conta.

#### Note

Você deve editar alguns dos dados no arquivo de configuração salvo antes de usá-lo na nova conta. Você deve também atualizar as informações relativas à conta antiga com os dados corretos da nova conta. Por exemplo, deve substituir o nome do recurso da Amazon (ARN) da função do AWS Identity and Access Management (IAM) pelo ARN da função do AIM para a nova conta.

Se você usar o comando [eb create](#) com `cfg`, o novo ambiente é criado usando o arquivo de configuração salvo especificado. Para obter mais informações, consulte [Usar configurações salvas do Elastic Beanstalk](#).

## Migrar um ambiente do EC2-Classico dentro da mesma conta da AWS

Sua conta da AWS existente pode ter recursos que não podem ser migrados para uma nova conta da AWS. Nesse caso, será necessário recriar os seus ambientes e configurar uma VPC manualmente para cada ambiente criado.

### Migrar os ambientes para uma VPC personalizada

#### Pré-requisitos


Antes de começar, você deve ter uma VPC. É possível criar uma VPC não padrão (personalizada) de uma das seguintes maneiras:

- Crie uma VPC rapidamente usando o assistente do console da Amazon VPC, com uma das opções de configuração disponíveis. Para obter mais informações, consulte [Configurações do assistente do console da Amazon VPC](#).
- Crie uma VPC personalizada no console da Amazon VPC se você tem requisitos mais específicos para a sua VPC. Recomendamos que você faça isso, por exemplo, se seu caso de uso exigir um número específico de sub-redes. Para obter mais informações, consulte [VPCs e sub-redes](#).
- Crie uma VPC usando o repositório [elastic-beanstalk-samples](#) no site do GitHub se preferir usar modelos do AWS CloudFormation com os seus ambientes do Elastic Beanstalk. Este repositório

inclui modelos de AWS CloudFormation. Para obter mais informações, consulte . [Usar o Elastic Beanstalk com Amazon VPC](#).

Nas etapas a seguir, use o ID da VPC gerada e os IDs de sub-rede ao configurar a VPC no novo ambiente.

1. Crie um ambiente que inclua a mesma configuração do ambiente existente. Isso pode ser feito usando uma das abordagens a seguir.

 Note

O recurso Saved Configurations (Configurações salvas) pode ajudar a recriar os ambientes na nova conta. Esse recurso pode salvar a configuração de um ambiente, para que você possa aplicá-lo ao criar ou atualizar outros ambientes. Para obter mais informações, consulte . [Usar configurações salvas do Elastic Beanstalk](#).

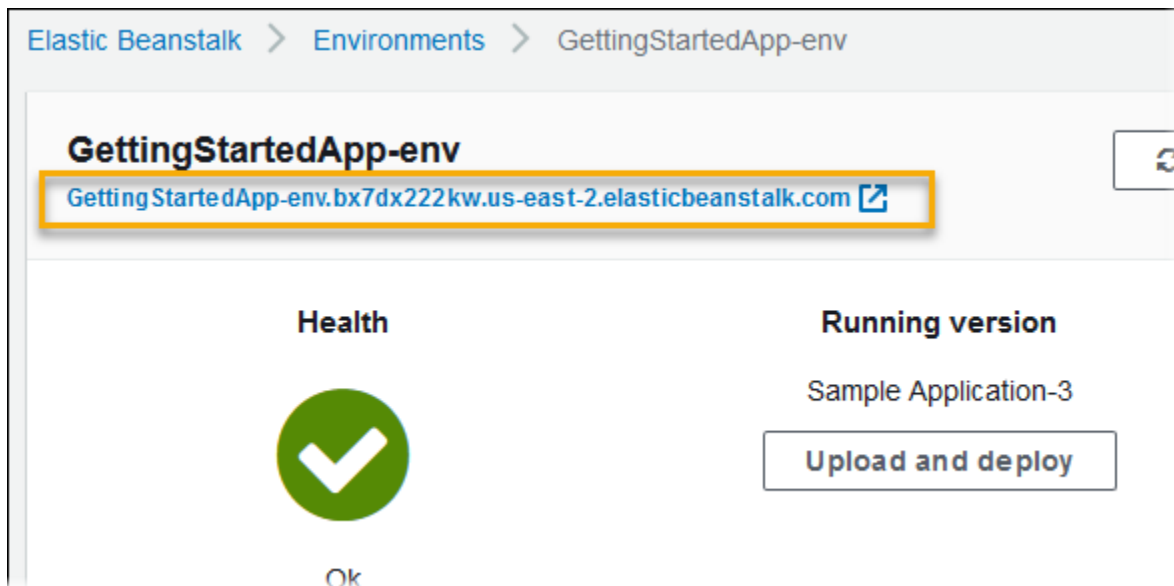
- Usando o [console do Elastic Beanstalk](#), aplique uma configuração salva do ambiente do EC2-Classic quando configurar o novo ambiente. Essa configuração usará a VPC. Para obter mais informações, consulte . [Usar configurações salvas do Elastic Beanstalk](#).
  - Com a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB), execute o comando `eb create` para recriar o ambiente. Forneça os parâmetros do ambiente original e do identificador da VPC. Um dos [exemplos](#) na descrição do comando `eb create` mostra como criar um ambiente em uma VPC personalizada.
  - Use o AWS Command Line Interface (AWS CLI) e recrie o ambiente usando o comando `elasticbeanstalk create-environment`. Forneça os parâmetros do ambiente original com o identificador da VPC. Para obter instruções, consulte [Criação de ambientes do Elastic Beanstalk com a CLI da AWS](#).
2. Troque os CNAMEs do ambiente atual pelos do novo ambiente. Dessa forma, o novo ambiente que você criou pode ser referenciado com o endereço já conhecido. Você pode usar a EB CLI ou a AWS CLI.
    - Usando a CLI do EB, troque os CNAMEs do ambiente executando o comando `eb swap`. Para obter mais informações, consulte . [Usar a Interface de Linhas de Comando do Elastic Beanstalk \(CLI do EB\)](#).

- Usando a AWS CLI, troque os CNAMEs do ambiente com o comando [elasticbeanstalk swap-environment-cnames](#). Para obter mais informações, consulte a [Referência de comandos da AWS CLI](#).

## Nome de domínio do ambiente do Elastic Beanstalk

Por padrão, seu ambiente está disponível para os usuários em um subdomínio do `elasticbeanstalk.com`. Quando você [cria um ambiente](#), pode escolher um nome de host para seu aplicativo. O subdomínio e o domínio são preenchidos automaticamente para `region.elasticbeanstalk.com`.

Para direcionar usuários para o ambiente do Elastic Beanstalk, registre um registro CNAME que aponte para o balanceador de carga do ambiente. É possível ver o URL da aplicação do ambiente com o valor atual do CNAME na página de [visão geral do ambiente](#) do console do Elastic Beanstalk.



Escolha o URL na página de visão geral ou escolha Go to environment (Acessar o ambiente) no painel de navegação para navegar até a página da web do aplicativo.

Você pode alterar o CNAME em seu ambiente substituindo-o pelo CNAME de outro ambiente. Para obter instruções, consulte [Implantações azuis/verdes com o Elastic Beanstalk](#).

Se você tem um nome de domínio, pode usar o Amazon Route 53 para resolvê-lo para o ambiente. É possível comprar um nome de domínio com o Amazon Route 53 ou usar um adquirido de outro provedor.



Para comprar um nome de domínio com o Route 53, consulte [Inscrever um novo domínio](#) no Guia do desenvolvedor do Amazon Route 53.

Para saber mais sobre o uso de um domínio personalizado, consulte [Encaminhamento de tráfego para um ambiente do AWS Elastic Beanstalk](#) no Guia do desenvolvedor do Amazon Route 53.

 Important

Se você encerrar um ambiente, também deverá excluir quaisquer mapeamentos CNAME criados, pois outros clientes poderão reutilizar um nome de host disponível. Certifique-se de excluir os registros DNS que apontam para seu ambiente encerrado para evitar uma entrada de DNS pendente. Uma entrada de DNS pendente pode expor o tráfego da Internet destinado ao seu domínio a vulnerabilidades de segurança. Também pode apresentar outros riscos.

Para obter mais informações, consulte [Proteção contra um registro de delegação pendente no Route 53](#) no Guia do desenvolvedor do Amazon Route 53. Você também pode aprender mais sobre entradas de DNS pendentes em [Proteções de domínio aprimoradas para solicitações do Amazon CloudFront](#) no blog de segurança da AWS.

# Configurar ambientes do Elastic Beanstalk (avançado)

Quando você cria um ambiente do AWS Elastic Beanstalk, o Elastic Beanstalk provisiona e configura todos os recursos da AWS necessários para executar e oferecer suporte a sua aplicação. Além de configurar os metadados e o comportamento de atualização do ambiente, você pode personalizar esses recursos fornecendo valores para [opções de configuração](#). Por exemplo, é possível adicionar uma fila do Amazon SQS e um alarme sobre o comprimento da fila, ou adicionar um cluster do Amazon ElastiCache.

A maioria das opções de configuração tem valores padrão que são aplicados automaticamente pelo Elastic Beanstalk. É possível substituir esses padrões com os arquivos de configuração, configurações salvas, opções da linha de comando ou chamando diretamente a API do Elastic Beanstalk. A CLI do EB e o console do Elastic Beanstalk também aplicam valores recomendados para algumas opções.

Você pode personalizar facilmente seu ambiente ao mesmo tempo que implanta a versão do aplicativo incluindo um arquivo de configuração com o pacote de origem. Ao personalizar o software em sua instância, é mais vantajoso usar um arquivo de configuração do que criar uma AMI personalizada, pois não é necessário manter um conjunto de AMIs.

Ao implantar seus aplicativos, você pode personalizar e configurar o software do qual o aplicativo depende. Esses arquivos podem ser dependências exigidas pela aplicação - por exemplo, pacotes adicionais do repositório yum - ou podem ser arquivos de configuração, como uma modificação para `httpd.conf` para anular configurações específicas que são usadas como padrão pelo AWS Elastic Beanstalk.

## Tópicos

- [Opções de configuração](#)
- [Personalização avançada de ambiente com arquivos de configuração \(.ebextensions\)](#)
- [Usar configurações salvas do Elastic Beanstalk](#)
- [Manifesto de ambiente \(env.yaml\)](#)
- [Usar uma imagem de máquina da Amazon \(AMI\) personalizada](#)
- [Fornecer arquivos estáticos](#)
- [Configurar HTTPS para o seu ambiente do Elastic Beanstalk](#)

## Opções de configuração

O Elastic Beanstalk define um grande número de opções de configuração que podem ser usadas para configurar o comportamento de seu ambiente e os recursos contidos neles. As opções de configuração são organizadas em namespaces, como `aws:autoscaling:asg`, que define opções para um grupo de Auto Scaling do ambiente.

O console do Elastic Beanstalk e a CLI do EB definem opções de configuração ao criar um ambiente, incluindo opções definidas explicitamente e [valores recomendados](#) definidos pelo cliente. Você também pode definir as opções de configuração em configurações salvas e em arquivos de configuração. Se a mesma opção for definida em várias localizações, o valor usado é determinado pela [ordem de precedência](#).

As definições de opção de configuração podem ser elaboradas em formato de texto e salvas antes da criação do ambiente, aplicadas durante a criação do ambiente usando qualquer cliente compatível e adicionadas, modificadas ou removidas após a criação do ambiente. Para ver uma discriminação detalhada de todos os métodos disponíveis para trabalhar com as opções de configuração em cada um desses três estágios, leia os seguintes tópicos:

- [Definir as opções de configuração antes da criação do ambiente](#)
- [Definir as opções de configuração durante a criação do ambiente](#)
- [Definir as opções de configuração depois da criação do ambiente](#)

Para obter uma lista completa de namespaces e opções, incluindo os valores padrão e compatíveis de cada um deles, consulte [Opções gerais para todos os ambientes](#) e [Opções específicas para a plataforma](#).

## Precedência

Durante a criação do ambiente, as opções de configuração são aplicadas de várias fontes com a seguinte precedência, da mais alta para a mais baixa:

- Configurações aplicadas diretamente ao ambiente: configurações especificadas durante uma operação de criação ou de atualização do ambiente na API do Elastic Beanstalk por qualquer cliente, incluindo o console do Elastic Beanstalk, EB CLI, AWS CLI, e SDKs. O console do Elastic Beanstalk e a CLI do EB também aplicam os [valores recomendados](#) de algumas opções aplicáveis nesse nível, a menos que sejam substituídos.

- **Configurações salvas:** as configurações para quaisquer opções que não são aplicadas diretamente ao ambiente são carregadas de uma configuração salva, se especificado.
- **Arquivos de configuração (.ebextensions):** as configurações para quaisquer opções que não são aplicadas diretamente ao ambiente, e também não são especificadas em uma configuração salva, são carregadas de arquivos de configuração na pasta `.ebextensions` na raiz do pacote de origem da aplicação.

Os arquivos de configuração são executados em ordem alfabética. Por exemplo, `.ebextensions/01run.config` é executado antes de `.ebextensions/02do.config`.

- **Valores padrão:** se uma opção de configuração tem um valor padrão, ela é aplicada apenas quando a opção não é definida em nenhum dos níveis acima.

Se a mesma opção de configuração for definida em mais de um local, a configuração com a precedência mais alta será aplicada. Quando uma configuração é aplicada de uma configuração salva ou de configurações aplicadas diretamente ao ambiente, ela é armazenada como parte da configuração do ambiente. Essas configurações podem ser removidas [com a AWS CLI](#) ou [com a EB CLI](#).

As configurações nos arquivos de configuração não são aplicadas diretamente ao ambiente e não podem ser removidas sem modificar os arquivos de configuração e implantar uma nova versão do aplicativo. Se uma configuração aplicada com um dos outros métodos for removida, a mesma configuração será carregada dos arquivos de configuração no pacote de origem.

Por exemplo, vamos supor que você defina o número mínimo de instâncias no ambiente como 5 durante a criação dele, usando o console do Elastic Beanstalk, uma opção de linha de comando ou uma configuração salva. O pacote de origem do seu aplicativo também inclui um arquivo de configuração que define o número mínimo de instâncias como 2.

Quando você cria o ambiente, o Elastic Beanstalk define a opção `MinSize` no namespace `aws:autoscaling:asg` como 5. Em seguida, se você remover a opção da configuração do ambiente, o valor no arquivo de configuração será carregado, e o número mínimo de instâncias será definido como 2. Na sequência, se você remover o arquivo de configuração do pacote de origem e reimplantá-lo, o Elastic Beanstalk usará a configuração padrão de 1.

## Valores recomendados

A Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB) e o console do Elastic Beanstalk oferecem valores recomendados para algumas opções de configuração. Esses valores podem

ser diferentes dos valores padrão e são definidos no nível da API quando o ambiente é criado. Os valores recomendados permitem que o Elastic Beanstalk melhore a configuração do ambiente padrão sem fazer alterações na API incompatíveis com as versões anteriores.

Por exemplo, a CLI do EB e o console do Elastic beanstalk definem a opção de configuração para o tipo de instância do EC2 (InstanceType no namespace `aws:autoscaling:launchconfiguration`). Cada cliente dispõe de um método diferente para substituir a configuração padrão. No console, você pode escolher um tipo de instância diferente em um menu suspenso na página Configuration Details do assistente Create New Environment. Com a EB CLI, você pode usar o parâmetro `--instance_type` para [eb create](#).

Como os valores recomendados são definidos no nível da API, eles substituirão os valores para as mesmas opções que você definir nos arquivos de configuração ou nas configurações salvas. As seguintes opções são definidas:

#### Console do Elastic Beanstalk

- Namespace: `aws:autoscaling:launchconfiguration`  
Nomes das opções: `IamInstanceProfile`, `EC2KeyName`, `InstanceType`
- Namespace: `aws:autoscaling:updatepolicy:rollingupdate`  
Nomes das opções: `RollingUpdateType` e `RollingUpdateEnabled`
- Namespace: `aws:elasticbeanstalk:application`  
Nome da opção: `Application Healthcheck URL`
- Namespace: `aws:elasticbeanstalk:command`  
Nome da opção: `DeploymentPolicy`, `BatchSize` e `BatchSizeType`
- Namespace: `aws:elasticbeanstalk:environment`  
Nome da opção: `ServiceRole`
- Namespace: `aws:elasticbeanstalk:healthreporting:system`  
Nome da opção: `SystemType` e `HealthCheckSuccessThreshold`
- Namespace: `aws:elasticbeanstalk:sns:topics`  
Nome da opção: `Notification Endpoint`
- Namespace: `aws:elasticbeanstalk:sqs`

Nome da opção: `HttpConnections`

- Namespace: `aws:elb:loadbalancer`

Nome da opção: `CrossZone`

- Namespace: `aws:elb:policies`

Nomes das opções: `ConnectionDrainingTimeout` e `ConnectionDrainingEnabled`

## EB CLI

- Namespace: `aws:autoscaling:launchconfiguration`

Nomes das opções: `IamInstanceProfile`, `InstanceType`

- Namespace: `aws:autoscaling:updatepolicy:rollingupdate`

Nomes das opções: `RollingUpdateType` e `RollingUpdateEnabled`

- Namespace: `aws:elasticbeanstalk:command`

Nome da opção: `BatchSize` e `BatchSizeType`

- Namespace: `aws:elasticbeanstalk:environment`

Nome da opção: `ServiceRole`

- Namespace: `aws:elasticbeanstalk:healthreporting:system`

Nome da opção: `SystemType`

- Namespace: `aws:elb:loadbalancer`

Nome da opção: `CrossZone`

- Namespace: `aws:elb:policies`

Nomes das opções: `ConnectionDrainingEnabled`

## Definir as opções de configuração antes da criação do ambiente

O AWS Elastic Beanstalk é compatível com um grande número de [opções de configuração](#) que permitem modificar as configurações aplicadas aos recursos do seu ambiente. Várias dessas opções

têm valores padrão que podem ser substituídos para personalizar o ambiente. Outras opções podem ser configuradas para ativar recursos adicionais.

O Elastic Beanstalk é compatível com dois métodos de salvamento das definições de opção de configuração. Os arquivos de configuração no formato JSON ou YAML podem ser incluídos no código-fonte do aplicativo em um diretório chamado `.ebextensions` e implantados como parte do seu pacote de origem do aplicativo. Crie e gerencie os arquivos de configuração localmente.

Configurações salvas são modelos que você cria em um ambiente em execução ou no arquivo de opções do JSON e armazena no Elastic Beanstalk. As configurações salvas existentes também podem ser estendidas para criar uma nova configuração.

#### Note

As configurações definidas nos arquivos de configuração e as configurações salvas têm menor precedência do que as configurações definidas durante ou após a criação do ambiente, inclusive os valores recomendados aplicados pelo console do Elastic Beanstalk e pela [CLI do EB](#). Para mais detalhes, consulte [Precedência](#).

As opções também podem ser especificadas em um documento JSON e fornecidas diretamente ao Elastic Beanstalk quando você criar ou atualizar um ambiente com a EB CLI ou a AWS CLI. As opções fornecidas diretamente ao Elastic Beanstalk dessa forma substituem todos os outros métodos.

Para uma lista completa de opções disponíveis, consulte [Opções de configuração](#).

#### Métodos

- [Arquivos de configuração \(.ebextensions\)](#)
- [Configurações salvas](#)
- [Documento JSON](#)
- [Configuração da EB CLI](#)

### Arquivos de configuração (**.ebextensions**)

Use `.ebextensions` para configurar as opções necessárias ao funcionamento do aplicativo e forneça valores padrão para outras opções que podem ser substituídas em um nível mais alto de

**precedência.** As opções especificadas em `.ebextensions` têm o nível mais baixo de precedência e são substituídas por configurações de qualquer outro nível.

Para usar arquivos de configuração, crie uma pasta chamada `.ebextensions` no nível superior do código-fonte do projeto. Adicione um arquivo com a extensão `.config` e especifique opções da seguinte maneira:

```
option_settings:
 - namespace: namespace
 option_name: option name
 value: option value
 - namespace: namespace
 option_name: option name
 value: option value
```

Por exemplo, o seguinte arquivo de configuração define o URL de verificação de integridade do aplicativo como `/health`:

`healthcheckurl.config`

```
option_settings:
 - namespace: aws:elasticbeanstalk:application
 option_name: Application Healthcheck URL
 value: /health
```

Em JSON:

```
{
 "option_settings" :
 [
 {
 "namespace" : "aws:elasticbeanstalk:application",
 "option_name" : "Application Healthcheck URL",
 "value" : "/health"
 }
]
}
```

Isso configura o balanceador de carga do Elastic Load Balancing no ambiente do Elastic Beanstalk para fazer uma solicitação HTTP para o caminho `/health` para cada instância do EC2 para determinar se ela está íntegra ou não.



**Note**

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Inclua o diretório `.ebextensions` em seu [Pacote de origem da aplicação](#) e implante-o no ambiente do Elastic Beanstalk novo ou já existente.

Os arquivos de configuração oferecem suporte a várias seções, além de `option_settings` para personalizar o software e arquivos que são executados nos servidores em seu ambiente. Para obter mais informações, consulte [.Ebextensions](#).

## Configurações salvas

Crie uma configuração salva para salvar as configurações aplicadas a um ambiente existente durante ou após a criação do ambiente, usando o console do Elastic Beanstalk, a EB CLI ou a AWS CLI. As configurações salvas pertencem a um aplicativo e podem ser aplicadas a ambientes novos ou existentes desse aplicativo.

### Cientes

- [Console do Elastic Beanstalk](#)
- [EB CLI](#)
- [AWS CLI](#)

### Console do Elastic Beanstalk

Como criar uma configuração salva (console do Elastic Beanstalk)

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

**Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Save Configuration (Salvar configuração).
4. Use a caixa de diálogo na tela para concluir a ação.

As configurações salvas são armazenadas no bucket do S3 do Elastic Beanstalk em uma pasta com o mesmo nome da sua aplicação. Por exemplo, as configurações de uma aplicação chamada `my-app` na região `us-west-2` da conta número `123456789012` podem ser encontradas em `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app`.

## EB CLI

A [EB CLI](#) também fornece subcomandos para interagir com configurações salvas em [eb config](#):

Para criar uma configuração salva (EB CLI)

1. Salve a configuração atual associada ao ambiente:

```
~/project$ eb config save --cfg my-app-v1
```

A EB CLI salva a configuração em `~/project/.elasticbeanstalk/saved_configs/my-app-v1.cfg.yml`

2. Modifique a configuração salva localmente, se necessário.
3. Faça upload da configuração salva no S3:

```
~/project$ eb config put my-app-v1
```

## AWS CLI

Crie uma configuração salva de um ambiente em execução com `aws elasticbeanstalk create-configuration-template`

Para criar uma configuração salva (AWS CLI)

1. Identifique o ID do ambiente do Elastic Beanstalk com `describe-environments`:

```
$ aws elasticbeanstalk describe-environments --environment-name my-env
{
 "Environments": [
 {
```

```

 "ApplicationName": "my-env",
 "EnvironmentName": "my-env",
 "VersionLabel": "89df",
 "Status": "Ready",
 "Description": "Environment created from the EB CLI using \"eb create
\",
 "EnvironmentId": "e-vcghmm2zwk",
 "EndpointURL": "awseb-e-v-AWSEBLoa-1JUM8159RA11M-43V6ZI1194.us-
west-2.elb.amazonaws.com",
 "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.2 running Multi-
container Docker 1.7.1 (Generic)",
 "CNAME": "my-env-nfptuqaper.elasticbeanstalk.com",
 "Health": "Green",
 "AbortableOperationInProgress": false,
 "Tier": {
 "Version": " ",
 "Type": "Standard",
 "Name": "WebServer"
 },
 "HealthStatus": "Ok",
 "DateUpdated": "2015-10-01T00:24:04.045Z",
 "DateCreated": "2015-09-30T23:27:55.768Z"
 }
]
}

```

## 2. Salve a configuração atual do ambiente com `create-configuration-template`:

```

$ aws elasticbeanstalk create-configuration-template --environment-id e-vcghmm2zwk
--application-name my-app --template-name v1

```

O Elastic Beanstalk salva a configuração no bucket do Elastic Beanstalk no Amazon S3.

## Documento JSON

Se você usar a AWS CLI para criar e atualizar ambientes, poderá também fornecer opções de configuração no formato JSON. Uma biblioteca de arquivos de configuração em JSON será útil se você usar a AWS CLI para criar e gerenciar ambientes.

Por exemplo, o seguinte documento JSON define o URL de verificação de integridade do aplicativo como `/health`:

```
~/ebconfigs/healthcheckurl.json
```

```
[
 {
 "Namespace": "aws:elasticbeanstalk:application",
 "OptionName": "Application Healthcheck URL",
 "Value": "/health"
 }
]
```

## Configuração da EB CLI

Além de ser compatível com configurações salvas e configuração de ambiente direta com comandos `eb config`, a CLI do EB tem um arquivo de configuração com uma opção chamada `default_ec2_keyname` que pode ser usada para especificar um par de chaves do Amazon EC2 para acesso SSH a instâncias do ambiente. A EB CLI usa essa opção para definir a opção de configuração `EC2KeyName` no namespace `aws:autoscaling:launchconfiguration`.

```
~/workspace/my-app/.elasticbeanstalk/config.yml
```

```
branch-defaults:
 master:
 environment: my-env
 develop:
 environment: my-env-dev
deploy:
 artifact: ROOT.war
global:
 application_name: my-app
 default_ec2_keyname: my-keypair
 default_platform: Tomcat 8 Java 8
 default_region: us-west-2
 profile: null
 sc: git
```

## Definir as opções de configuração durante a criação do ambiente

Quando você cria um ambiente do AWS Elastic Beanstalk usando o console do Elastic Beanstalk, a EB CLI, a AWS CLI, um SDK ou a API do Elastic Beanstalk, pode especificar valores para as opções de configuração a fim de personalizar o ambiente e os recursos da AWS que são iniciados nele.

Para qualquer outro fim que não seja uma alteração de configuração isolada, é possível [armazenar os arquivos de configuração](#) localmente, em seu pacote de origem ou no Amazon S3.

Este tópico inclui procedimentos para todos os métodos de definição de opções de configuração durante a criação do ambiente.

## Cientes

- [No console do Elastic Beanstalk](#)
- [Uso da EB CLI](#)
- [Usando o AWS CLI](#)

## No console do Elastic Beanstalk

Ao criar um ambiente do Elastic Beanstalk no console do Elastic Beanstalk, é possível especificar as opções de configuração usando os arquivos de configuração, as configurações salvas e os formulários no assistente Create New Environment (Criar novo ambiente).

## Métodos

- [Usar arquivos de configuração \(.ebextensions\)](#)
- [Usar uma configuração salva](#)
- [Usar o assistente de novo ambiente](#)

## Usar arquivos de configuração (.ebextensions)

Inclua os arquivos `.config` em seu [pacote de origem do aplicativo](#) em uma pasta chamada `.ebextensions`.

Para obter detalhes sobre arquivos de configuração, consulte [.Ebextensions](#).

```
~/workspace/my-app-v1.zip
|-- .ebextensions
| |-- environmentvariables.config
| `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Faça upload do pacote de origem para o Elastic Beanstalk normalmente, durante a [criação do ambiente](#).

O console do Elastic Beanstalk aplica [valores recomendados](#) a algumas opções de configuração e tem campos de formulário para outras. As opções configuradas pelo console do Elastic Beanstalk são aplicadas diretamente ao ambiente e substituem aquelas nos arquivos de configuração.

### Usar uma configuração salva

Quando você cria um novo ambiente usando o console do Elastic Beanstalk, uma das primeiras etapas é escolher uma configuração. A configuração pode ser uma [predefined configuration \(configuração predefinida\)](#), normalmente a versão mais recente de uma Plataforma, como o PHP ou o Tomcat, ou pode ser uma saved configuration (configuração salva).

Como aplicar uma configuração salva durante a criação do ambiente (console do Elastic Beanstalk)

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

#### Note

Se você tiver muitos aplicativos, use a barra de pesquisa para filtrar a lista de aplicativos.

3. No painel de navegação, encontre o nome do aplicativo e escolha Saved configurations (Configurações salvas).
4. Selecione a configuração salva que deseja aplicar e escolha Launch environment (Iniciar ambiente).
5. Continue no assistente para criar seu ambiente.

As configurações salvas são específicas ao aplicativo. Consulte [Configurações salvas](#) para obter detalhes sobre como criar configurações salvas.

### Usar o assistente de novo ambiente

A maioria das opções de configuração padrão estão disponíveis na página Configure more options (Configurar mais opções) do [Assistente de configuração de novo ambiente](#). Se você criar um banco de dados do Amazon RDS ou configurar uma VPC para o ambiente, as opções de configuração adicionais estarão disponíveis nas páginas desses recursos.

Como definir opções de configuração durante a criação do ambiente (console do Elastic Beanstalk)

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, escolha Applications (Aplicativos).
3. Selecione ou [crie](#) um aplicativo.
4. Escolha Actions (Ações) e Create environment (Criar ambiente).
5. Continue no assistente e escolha Configure more options (Configurar mais opções).
6. Escolha qualquer uma das configuration presets (predefinições da configuração) e escolha Edit (Editar) em uma ou mais das categorias de configuração para alterar um grupo de opções de configuração relacionado.
7. Ao concluir as seleções das opções, escolha Create environment (Criar ambiente).

Todas as opções que você definir no assistente de novo ambiente serão definidas diretamente no ambiente e substituirão qualquer configuração de opção nas configurações salvas ou nos arquivos de configuração (.ebextensions) que você aplicar. Você poderá remover as configurações após a criação do ambiente usando a [EB CLI](#) ou a [AWS CLI](#) para permitir que as definições nas configurações salvas ou nos arquivos de configuração sejam exibidas.

Para obter detalhes sobre o novo assistente de ambiente, consulte [O assistente de criação de novo ambiente](#).

## Uso da EB CLI

### Métodos

- [Usar arquivos de configuração \(.ebextensions\)](#)
- [Usar as configurações salvas](#)
- [Usar as opções de linha de comando](#)

### Usar arquivos de configuração (.ebextensions)

Inclua os arquivos .config na pasta do seu projeto em .ebextensions para implantá-los com o código do seu aplicativo.

Para obter detalhes sobre arquivos de configuração, consulte [.Ebextensions](#).

```
~/workspace/my-app/
|-- .ebextensions
| |-- environmentvariables.config
| `-- healthcheckurl.config
|-- .elasticbeanstalk
| `-- config.yml
|-- index.php
`-- styles.css
```

Crie seu ambiente e implante o código-fonte nele com `eb create`.

```
~/workspace/my-app$ eb create my-env
```

### Usar as configurações salvas

Para aplicar uma configuração salva ao criar um ambiente com [eb create](#), use a opção `--cfg`.

```
~/workspace/my-app$ eb create --cfg savedconfig
```

É possível armazenar a configuração salva na pasta do projeto ou no local de armazenamento do Elastic Beanstalk no Amazon S3. No exemplo anterior, a EB CLI primeiro procura um arquivo de configuração salvo chamado `savedconfig.cfg.yml` na pasta `.elasticbeanstalk/saved_configs/`. Não inclua as extensões de arquivo (`.cfg.yml`) ao aplicar uma configuração salva com `--cfg`.

```
~/workspace/my-app/
|-- .ebextensions
| `-- healthcheckurl.config
|-- .elasticbeanstalk
| |-- saved_configs
| | `-- savedconfig.cfg.yml
| `-- config.yml
|-- index.php
`-- styles.css
```

Se a CLI do EB não encontrar a configuração localmente, ela verificará no local de armazenamento do Elastic Beanstalk no Amazon S3. Para obter detalhes sobre como criar, editar, e fazer upload das configurações salvas, consulte [Configurações salvas](#).



## Usar as opções de linha de comando

O comando `eb create` da EB CLI tem várias [opções](#) que você pode usar para definir as opções de configuração durante a criação do ambiente. Você pode usar essas opções para adicionar um banco de dados do RDS a seu ambiente, configurar uma VPC ou substituir os [valores recomendados](#).

Por exemplo, a EB CLI usa o tipo de instância `t2.micro` por padrão. Para escolher um tipo de instância diferente, use a opção `--instance_type`.

```
$ eb create my-env --instance_type t2.medium
```

Para criar uma instância de banco de dados do Amazon RDS e anexá-la ao ambiente, use as opções `--database`.

```
$ eb create --database.engine postgres --database.username dbuser
```

Se você não inserir o nome do ambiente, a senha do banco de dados ou qualquer outro parâmetro necessário para criar o ambiente, a EB CLI solicitará que você os insira.

Consulte [eb create](#) para obter uma lista completa de opções disponíveis e exemplos de uso.

## Usando o AWS CLI

Quando você usa o comando `create-environment` para criar um ambiente do Elastic Beanstalk com a AWS CLI, a AWS CLI não aplica nenhum [valor recomendado](#). Todas as opções de configuração são definidas nos arquivos de configuração no pacote de origem especificado por você.

### Métodos

- [Usar arquivos de configuração \(.ebextensions\)](#)
- [Usar uma configuração salva](#)
- [Usar as opções de linha de comando](#)

### Usar arquivos de configuração (.ebextensions)

Para aplicar os arquivos de configuração a um ambiente criado com a AWS CLI, inclua-os no pacote de fonte da aplicação do qual fará upload para o Amazon S3.

Para obter detalhes sobre arquivos de configuração, consulte [.Ebextensions](#).

```
~/workspace/my-app-v1.zip
```

```
|-- .ebextensions
| |-- environmentvariables.config
| `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Para fazer upload de um pacote de fonte da aplicação e criar um ambiente com a AWS CLI

1. Se você ainda não tiver um bucket do Elastic Beanstalk no Amazon S3, crie um com `create-storage-location`.

```
$ aws elasticbeanstalk create-storage-location
{
 "S3Bucket": "elasticbeanstalk-us-west-2-123456789012"
}
```

2. Faça upload do pacote de origem da aplicação para o Amazon S3.

```
$ aws s3 cp sourcebundle.zip s3://elasticbeanstalk-us-west-2-123456789012/my-app/sourcebundle.zip
```

3. Crie a versão do aplicativo.

```
$ aws elasticbeanstalk create-application-version --application-name my-app --
version-label v1 --description MyAppv1 --source-bundle S3Bucket="elasticbeanstalk-
us-west-2-123456789012",S3Key="my-app/sourcebundle.zip" --auto-create-application
```

4. Criar o ambiente

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-
name my-env --version-label v1 --solution-stack-name "64bit Amazon Linux 2015.03
v2.0.0 running Tomcat 8 Java 8"
```

Usar uma configuração salva

Para aplicar uma configuração salva a um ambiente durante a criação, use o parâmetro `--template-name`.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name
my-env --template-name savedconfig --version-label v1
```

Quando você especificar uma configuração salva, não especifique também o nome da pilha da solução. As configurações salvas já especificam uma pilha de soluções, e o Elastic Beanstalk retornará um erro se você tentar usar as duas opções.

Usar as opções de linha de comando

Use o parâmetro `--option-settings` para especificar as opções de configuração no formato JSON.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name my-env --version-label v1 --template-name savedconfig --option-settings '[
 {
 "Namespace": "aws:elasticbeanstalk:application",
 "OptionName": "Application Healthcheck URL",
 "Value": "/health"
 }
]
```

Para carregar o JSON de um arquivo, use o prefixo `file://`.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name my-env --version-label v1 --template-name savedconfig --option-settings file://healthcheckurl.json
```

O Elastic Beanstalk aplica as configurações de opção especificadas com a opção `--option-settings` diretamente ao ambiente. Se as mesmas opções forem especificadas em uma configuração salva ou arquivo de configuração, `--option-settings` substituirá esses valores.

## Definir as opções de configuração depois da criação do ambiente

Você pode modificar as configurações de opções em um ambiente em execução, aplicando configurações salvas, carregando um novo pacote de origem com os arquivos de configuração (`.ebextensions`) ou usando um documento JSON. A CLI do EB e o console do Elastic Beanstalk também têm a funcionalidade específica do cliente para configurar e atualizar as opções de configuração.

Ao definir ou alterar uma opção de configuração, você pode acionar uma atualização completa do ambiente, dependendo da severidade da alteração. Por exemplo, alterações nas opções no [aws:autoscaling:launchconfiguration](#), como `InstanceType`, exigem que as instâncias do Amazon EC2 no ambiente sejam provisionadas novamente. Isso aciona uma [atualização](#)

[contínua](#). Outras alterações de configuração podem ser aplicadas sem qualquer interrupção ou novo provisionamento.

Você pode remover as configurações de opção de um ambiente com os comandos da EB CLI ou da AWS CLI. Remover uma opção que foi definida diretamente em um ambiente em um nível de API permite que as configurações em arquivos de configuração, que são, caso contrário, mascaradas por configurações aplicadas diretamente a um ambiente, surjam e entrem em vigor.

As definições em configurações salvas e em arquivos de configuração podem ser substituídas definindo a mesma opção diretamente no ambiente com um dos outros métodos de configuração. No entanto, essas definições só podem ser removidas completamente aplicando uma atualização de uma configuração salva ou de um arquivo de configuração. Quando uma opção não for definida em uma configuração salva, em um arquivo de configuração ou diretamente em um ambiente, o valor padrão será aplicado, se houver. Para mais detalhes, consulte [Precedência](#).

## Cientes

- [O console do Elastic Beanstalk](#).
- [A EB CLI](#)
- [A AWS CLI](#)

## O console do Elastic Beanstalk.

É possível atualizar as definições de opções de configuração no console do Elastic Beanstalk implantando um pacote de origem da aplicação que contenha os arquivos de configuração, aplicando uma configuração salva ou modificando o ambiente diretamente com a página Configuration (Configuração) no console de gerenciamento do ambiente.

## Métodos

- [Usar arquivos de configuração \(.ebextensions\)](#)
- [Usar uma configuração salva](#)
- [Usar o console do Elastic Beanstalk](#)


## Usar arquivos de configuração (.ebextensions)

Atualize os arquivos de configuração no diretório de origem, crie um novo pacote de origem e implante a nova versão no ambiente do Elastic Beanstalk para aplicar as alterações.

Para obter detalhes sobre arquivos de configuração, consulte [.Ebextensions](#).

Para implantar um pacote de origem

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.


As alterações feitas em arquivos de configuração não substituem as configurações de opção em configurações salvas ou em configurações aplicadas diretamente ao ambiente no nível de API. Consulte [Precedência](#) para obter mais detalhes.

Usar uma configuração salva

Aplique uma configuração salva a um ambiente em execução para aplicar as configurações de opção que ele define.

Como aplicar uma configuração salva a um ambiente em execução (console do Elastic Beanstalk)

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

 Note

Se você tiver muitos aplicativos, use a barra de pesquisa para filtrar a lista de aplicativos.

3. No painel de navegação, encontre o nome do aplicativo e escolha Saved configurations (Configurações salvas).
4. Selecione a configuração salva que deseja aplicar e escolha Load (Carregar).
5. Selecione um ambiente e escolha Load.

As configurações definidas em uma configuração salva substituem as configurações em arquivos de configuração e são substituídas pelas configurações definidas usando o console de gerenciamento do ambiente.


Consulte [Configurações salvas](#) para obter detalhes sobre como criar configurações salvas.

Usar o console do Elastic Beanstalk

O console do Elastic Beanstalk apresenta várias opções de configuração na página Configuration (Configuração) para cada ambiente.

Como alterar as opções de configuração em um ambiente em execução (console do Elastic Beanstalk)

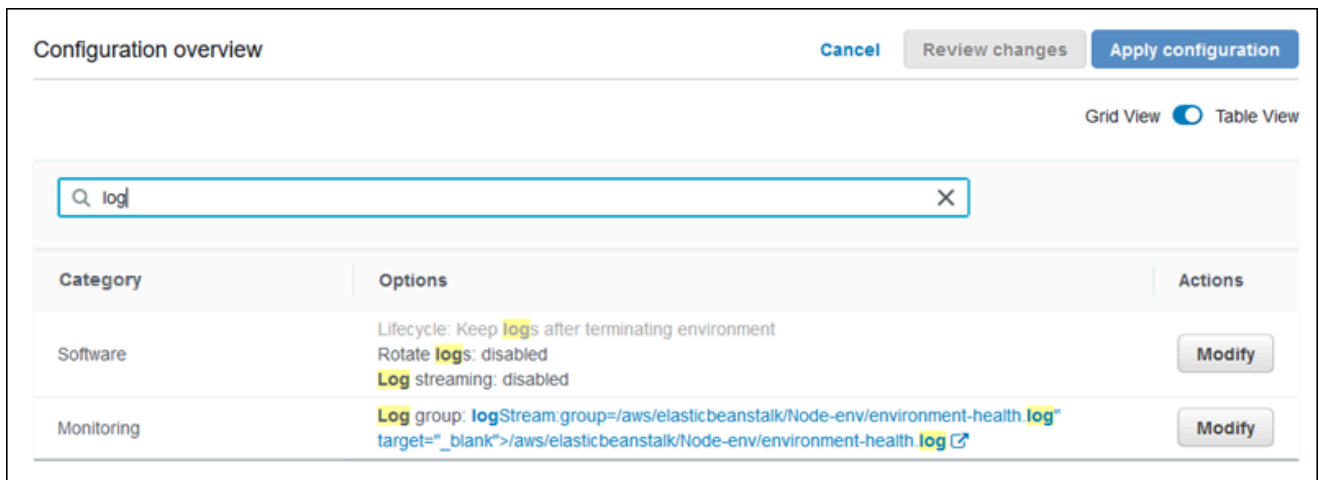
1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Encontre a página de configuração a qual deseja editar:
  - Se você vir a opção que lhe interessa ou souber a categoria de configuração em que ela está, escolha Edit (Editar) na categoria de configuração dela.
  - Para procurar uma opção, habilite a Table View (Exibição em tabela) e insira os termos de pesquisa na caixa de pesquisa. Conforme você digita, a lista diminui e mostra apenas as opções que correspondem aos seus termos de pesquisa.

Quando vir a opção que você está procurando, escolha Edit (Editar) na categoria de configuração em que ela está.



5. Altere as configurações e selecione Save (Salvar).
6. Repita as duas etapas anteriores em outras categorias de configuração, conforme necessário.
7. Escolha Apply (Aplicar).

As alterações feitas nas opções de configuração no console de gerenciamento do ambiente são aplicadas diretamente ao ambiente. Essas alterações substituem as definições das mesmas opções em arquivos de configuração ou em configurações salvas. Para obter mais detalhes, consulte [Precedence \(Precedência\)](#).

Para obter detalhes sobre como alterar as opções de configuração em um ambiente em execução usando o console do Elastic Beanstalk, consulte os tópicos em [Configurar ambientes do Elastic Beanstalk](#).

## A EB CLI

Você pode atualizar as definições de opção de configuração com a EB CLI implantando um código-fonte que contenha os arquivos de configuração, aplicando as definições de uma configuração salva ou modificando a configuração do ambiente diretamente com o comando `eb config`.

### Métodos

- [Usar arquivos de configuração \(.ebextensions\)](#)
- [Usar uma configuração salva](#)
- [Usar o eb config](#)
- [Usar o eb setenv](#)

## Usar arquivos de configuração (.ebextensions)

Inclua os arquivos `.config` na pasta do seu projeto em `.ebextensions` para implantá-los com o código do seu aplicativo.

Para obter detalhes sobre arquivos de configuração, consulte [Ebextensions](#).

```
~/workspace/my-app/
|-- .ebextensions
| |-- environmentvariables.config
| `-- healthcheckurl.config
|-- .elasticbeanstalk
| `-- config.yml
|-- index.php
`-- styles.css
```

Implante seu código-fonte com `eb deploy`.

```
~/workspace/my-app$ eb deploy
```

## Usar uma configuração salva

Você pode usar o comando `eb config` para aplicar uma configuração salva a um ambiente em execução. Use a opção `--cfg` com o nome da configuração salva para aplicar suas configurações ao seu ambiente.

```
$ eb config --cfg v1
```

Neste exemplo, `v1` é o nome de um [criado anteriormente e o arquivo de configuração salvo](#).

As configurações aplicadas a um ambiente com esse comando substituem as configurações que foram aplicadas durante a criação do ambiente e as configurações definidas em arquivos de configuração em seu pacote de origem do aplicativo.

## Usar o `eb config`

O comando `eb config` da EB CLI permite que você defina e remova configurações de opção diretamente em um ambiente usando um editor de texto.

Quando você executa `eb config`, a EB CLI exibe as configurações aplicadas em seu ambiente de todas as origens, inclusive de arquivos de configuração, configurações salvas, valores recomendados, opções definidas diretamente no ambiente e padrões de API.



**Note**

eb config não exibe as propriedades do ambiente. Para definir as propriedades do ambiente que você pode ler a partir do seu aplicativo, use [eb setenv](#).

O exemplo a seguir mostra as configurações aplicadas no namespace `aws:autoscaling:launchconfiguration`. Essas configurações incluem:

- Dois valores recomendados, para `IamInstanceProfile` e `InstanceType`, aplicados pela EB CLI durante a criação do ambiente.
- A opção `EC2KeyName` definida diretamente no ambiente durante a criação com base na configuração do repositório.
- Os valores padrão da API para as outras opções.

```
ApplicationName: tomcat
DateUpdated: 2015-09-30 22:51:07+00:00
EnvironmentName: tomcat
SolutionStackName: 64bit Amazon Linux 2015.03 v2.0.1 running Tomcat 8 Java 8
settings:
...
aws:autoscaling:launchconfiguration:
 BlockDeviceMappings: null
 EC2KeyName: my-key
 IamInstanceProfile: aws-elasticbeanstalk-ec2-role
 ImageId: ami-1f316660
 InstanceType: t2.micro
...
```

Para definir ou alterar as opções de configuração com `eb config`

1. Execute `eb config` para visualizar a configuração de seu ambiente.

```
~/workspace/my-app/$ eb config
```

2. Altere qualquer um dos valores de configuração usando o editor de texto padrão.

```
aws:autoscaling:launchconfiguration:
 BlockDeviceMappings: null
```

```
EC2KeyName: my-key
IamInstanceProfile: aws-elasticbeanstalk-ec2-role
ImageId: ami-1f316660
InstanceType: t2.medium
```

3. Salve o arquivo de configuração temporário e saia.
4. A EB CLI atualiza a configuração do ambiente.

Definir as opções de configuração com `eb config` substitui as definições de todas as outras fontes.

Você também pode remover as opções de seu ambiente com `eb config`.

Para remover as opções de configuração (EB CLI)

1. Execute `eb config` para visualizar a configuração de seu ambiente.

```
~/workspace/my-app/$ eb config
```

2. Substitua qualquer valor mostrado com a string `null`. Você também pode excluir toda a linha que contém a opção que você deseja remover.

```
aws:autoscaling:launchconfiguration:
 BlockDeviceMappings: null
 EC2KeyName: my-key
 IamInstanceProfile: aws-elasticbeanstalk-ec2-role
 ImageId: ami-1f316660
 InstanceType: null
```

3. Salve o arquivo de configuração temporário e saia.
4. A EB CLI atualiza a configuração do ambiente.

Remover as opções de seu ambiente com `eb config` permite que as configurações para as mesmas opções surjam a partir dos arquivos de configuração em seu pacote de origem do aplicativo.

Consulte [Precedência](#) para obter mais detalhes.

### Usar o `eb setenv`

Para definir as propriedades de ambiente com a EB CLI, use `eb setenv`.

```
~/workspace/my-app/$ eb setenv ENVVAR=TEST
```

```
INFO: Environment update is starting.
INFO: Updating environment my-env's configuration settings.
INFO: Environment health has transitioned from Ok to Info. Command is executing on all
instances.
INFO: Successfully deployed new configuration to environment.
```

Esse comando define as propriedades de ambiente no [aws:elasticbeanstalk:application:environment namespace](#). As propriedades do ambiente definidas com `eb setenv` ficam disponíveis para seu aplicativo após um curto processo de atualização.

Visualize as propriedades do ambiente definidas em seu ambiente com `eb printenv`.

```
~/workspace/my-app/$ eb printenv
Environment Variables:
 ENVVAR = TEST
```

## A AWS CLI

Você pode atualizar as definições da opção de configuração com a AWS CLI implantando um pacote de fonte que contenha os arquivos de configuração, aplicando uma configuração salva armazenada remotamente ou modificando o ambiente diretamente com o comando `aws elasticbeanstalk update-environment`.

### Métodos

- [Usar arquivos de configuração \(.ebextensions\)](#)
- [Usar uma configuração salva](#)
- [Usar as opções de linha de comando](#)

### Usar arquivos de configuração (**.ebextensions**)

Para aplicar os arquivos de configuração a um ambiente em execução com a AWS CLI, inclua-os no pacote de fonte da aplicação da qual você deseja fazer upload para o Amazon S3.

Para obter detalhes sobre arquivos de configuração, consulte [.Ebextensions](#).

```
~/workspace/my-app-v1.zip
|-- .ebextensions
```

```
| |-- environmentvariables.config
| |-- healthcheckurl.config
|-- index.php
|-- styles.css
```

Para fazer upload de um pacote de fonte da aplicação e aplicá-lo em um ambiente em execução (AWS CLI)

1. Se você ainda não tiver um bucket do Elastic Beanstalk no Amazon S3, crie um com `create-storage-location`:

```
$ aws elasticbeanstalk create-storage-location
{
 "S3Bucket": "elasticbeanstalk-us-west-2-123456789012"
}
```

2. Faça upload do pacote de origem da aplicação para o Amazon S3.

```
$ aws s3 cp sourcebundlev2.zip s3://elasticbeanstalk-us-west-2-123456789012/my-app/sourcebundlev2.zip
```

3. Crie a versão do aplicativo.

```
$ aws elasticbeanstalk create-application-version --application-name my-app --version-label v2 --description MyAppv2 --source-bundle S3Bucket="elasticbeanstalk-us-west-2-123456789012",S3Key="my-app/sourcebundlev2.zip"
```

4. Atualize o ambiente.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --version-label v2
```

### Usar uma configuração salva

Você pode aplicar uma configuração salva em um ambiente em execução com a opção `--template-name` no comando `aws elasticbeanstalk update-environment`.

A configuração salva deve estar no bucket do Elastic Beanstalk em um caminho com nome baseado na aplicação em `resources/templates`. Por exemplo, o modelo `v1` para a aplicação `my-app` na

região Oeste dos EUA (Oregon) (us-west-2) para a conta 123456789012 está localizado em `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app/v1`

Para aplicar uma configuração salva a um ambiente em execução (AWS CLI)

- Especifique a configuração salva em uma chamada `update-environment` com a opção `--template-name`.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --template-name v1
```

O Elastic Beanstalk coloca as configurações salvas neste local quando são criadas com `aws elasticbeanstalk create-configuration-template`. Você também pode modificar as configurações salvas localmente e colocá-las nesse local por conta própria.

Usar as opções de linha de comando

Para alterar as opções de configuração com um documento JSON (AWS CLI)

1. Defina as configurações de opção no formato JSON em um arquivo local.
2. Execute `update-environment` com a opção `--option-settings`.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --option-settings file:///~/ebconfigs/as-zero.json
```

Neste exemplo, `as-zero.json` define as opções que configuram o ambiente com um mínimo e um máximo de instâncias zero. Isso interrompe as instâncias no ambiente sem encerrar o ambiente.

**~/ebconfigs/as-zero.json**

```
[
 {
 "Namespace": "aws:autoscaling:asg",
 "OptionName": "MinSize",
 "Value": "0"
 },
 {
 "Namespace": "aws:autoscaling:asg",
 "OptionName": "MaxSize",
```

```
 "Value": "0"
 },
 {
 "Namespace": "aws:autoscaling:updatepolicy:rollingupdate",
 "OptionName": "RollingUpdateEnabled",
 "Value": "false"
 }
]
```

### Note

Definir as opções de configuração com `update-environment` substitui as definições de todas as outras fontes.

Você também pode remover as opções de seu ambiente com `update-environment`.

Para remover as opções de configuração (AWS CLI)

- Execute o comando `update-environment` com a opção `--options-to-remove`.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --options-to-remove Namespace=aws:autoscaling:launchconfiguration,OptionName=InstanceType
```

Remover as opções de seu ambiente com `update-environment` permite que as configurações para as mesmas opções surjam a partir dos arquivos de configuração em seu pacote de origem do aplicativo. Se uma opção não for configurada usando qualquer um desses métodos, o valor padrão da API será aplicado, se houver. Consulte [Precedência](#) para obter mais detalhes.

## Opções gerais para todos os ambientes

### Namespaces

- [aws:autoscaling:asg](#)
- [aws:autoscaling:launchconfiguration](#)
- [aws:autoscaling:scheduledaction](#)
- [aws:autoscaling:trigger](#)
- [aws:autoscaling:updatepolicy:rollingupdate](#)

- [aws:ec2:instances](#)
- [aws:ec2:vpc](#)
- [aws:elasticbeanstalk:application](#)
- [aws:elasticbeanstalk:application:environment](#)
- [aws:elasticbeanstalk:cloudwatch:logs](#)
- [aws:elasticbeanstalk:cloudwatch:logs:health](#)
- [aws:elasticbeanstalk:command](#)
- [aws:elasticbeanstalk:environment](#)
- [aws:elasticbeanstalk:environment:process:default](#)
- [aws:elasticbeanstalk:environment:process:process\\_name](#)
- [aws:elasticbeanstalk:environment:proxy:staticfiles](#)
- [aws:elasticbeanstalk:healthreporting:system](#)
- [aws:elasticbeanstalk:hostmanager](#)
- [aws:elasticbeanstalk:managedactions](#)
- [aws:elasticbeanstalk:managedactions:platformupdate](#)
- [aws:elasticbeanstalk:monitoring](#)
- [aws:elasticbeanstalk:sns:topics](#)
- [aws:elasticbeanstalk:sqs](#)
- [aws:elasticbeanstalk:trafficsplitting](#)
- [aws:elasticbeanstalk:xray](#)
- [aws:elb:healthcheck](#)
- [aws:elb:loadbalancer](#)
- [aws:elb:listener](#)
- [aws:elb:listener:listener\\_port](#)
- [aws:elb:policies](#)
- [aws:elb:policies:policy\\_name](#)
- [aws:elbv2:listener:default](#)
- [aws:elbv2:listener:listener\\_port](#)
- [aws:elbv2:listenerrule:rule\\_name](#)
- [aws:elbv2:loadbalancer](#)

- [aws:rds:dbinstance](#)

## aws:autoscaling:asg

Configure o grupo de Auto Scaling do ambiente. Para obter mais informações, consulte [the section called “Grupo de Auto Scaling”](#).

Namespace: **aws:autoscaling:asg**

Nome	Descrição	Padrão	Valores válidos
Availability Zones	As zonas de disponibilidade (AZs) são locais distintos dentro de uma região da AWS e são criadas para ficarem isoladas de falhas em outras AZs. Elas oferecem conectividade de rede de baixa latência e custo reduzido para outras ZDs na mesma região. Escolha o número de AZs para suas instâncias.	Any	Any Any 1 Any 2 Any 3
Cooldown	Os períodos de desaquecimento ajudam a evitar que o Amazon EC2 Auto Scaling inicie ações de escalabilidade adicionais antes que os efeitos de atividades anteriores sejam visíveis. O período de desaquecimento é a quantidade de tempo, em segundos, após a conclusão de uma ação de escalabilidade antes que uma outra ação de escalabilidade possa ser iniciada.	360	0 para 10000
Custom Availability Zones	Defina as AZs para suas instâncias.	Nenhum	us-east-1a us-east-1b us-east-1c us-east-1d



Nome	Descrição	Padrão	Valores válidos
			us-east-1e eu-central-1-1
EnableCapacityRebalancing	<p>Especifica se o recurso Rebalanceamento de capacidade deve ser habilitado para instâncias spot no seu grupo do Auto Scaling. Para obter mais informações, consulte <a href="#">Rebalanceamento de capacidade</a> no Manual do usuário do Amazon EC2 Auto Scaling.</p> <p>Essa opção só é relevante quando EnableSpot é definido como true no namespace- <a href="#">aws:ec2:instances</a> e quando há pelo menos uma instância spot no grupo do Auto Scaling.</p>	false	true false
MinSize	O número mínimo de instâncias que você deseja no grupo de Auto Scaling.	1	1 para 10000
MaxSize	O número máximo de instâncias desejadas no grupo de Auto Scaling.	4	1 para 10000

## aws:autoscaling:launchconfiguration


Configure as instâncias do Amazon Elastic Compute Cloud (Amazon EC2) do seu ambiente.

As instâncias do seu ambiente são criadas usando um modelo de lançamento do Amazon EC2 ou um recurso de configuração de execução de grupo do Auto Scaling. As opções a seguir funcionam com ambos os tipos de recursos.

Para obter mais informações, consulte [the section called “Instâncias do Amazon EC2”](#). Você também consultar mais informações sobre o Amazon Elastic Block Store (EBS) no capítulo [Amazon EBS](#)


do Amazon EC2 User Guide for Linux Instances (Guia do usuário do Amazon EC2 para instâncias Linux).

Namespace: **aws:autoscaling:launchconfiguration**


Nome	Descrição	Padrão	Valores válidos
DisableIMDSv1	<p>Defina como <code>true</code> para desabilitar o serviço de metadados de instância versão 1 (IMDSv1).</p> <p>As instâncias do seu ambiente são padronizadas da seguinte forma, com base no sistema operacional da plataforma:</p> <ul style="list-style-type: none"> <li>• Servidor do Windows, AL2 e anterior – habilite IMDSv1 e IMDSv2</li> <li>• AL2023 — habilite somente IMDSv2</li> </ul> <p>Para obter mais informações, consulte <a href="#">Configurar o serviço de metadados de instância</a> (Amazon Linux) ou <a href="#">Configurando o serviço de metadados da instância</a> (servidor do Windows).</p>	<p><code>false</code>— plataforma as baseadas em servidor Windows, Amazon Linux 2 e versões anteriores</p> <p><code>true</code>— plataforma as baseadas no Amazon Linux 2023</p>	<p><code>true</code></p> <p><code>false</code></p>
EC2KeyName	<p>Você pode usar um par de chaves para fazer login com segurança em sua instância do EC2.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um <a href="#">arquivo de configuração</a>. O</p> </div>	Nenhum	

Nome	Descrição	Padrão	Valores válidos
	console e substitui essa opção por um <a href="#">valor recomendado</a> .		
iamInstanceProfile	<p>Um perfil da instância habilita que os usuários do AWS Identity and Access Management (IAM) e os produtos da AWS tenham acesso a credenciais de segurança temporárias para fazer chamadas à API da AWS. Especifique o nome do perfil de instância ou o ARN dele.</p> <p>Exemplos:</p> <ul style="list-style-type: none"> <li>aws-elasticbeanstalk-ec2-role</li> <li>arn:aws:iam::123456789012:instance-profile/aws-elasticbeanstalk-ec2-role</li> </ul>	Nenhum	Nome do perfil de instância ou ARN.
	<p><b>Note</b></p> <p>Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um <a href="#">arquivo de configuração</a>. O console e a CLI do EB substituem essa opção por um <a href="#">valor recomendado</a>.</p>		

Nome	Descrição	Padrão	Valores válidos
ImageId	Você pode substituir a Imagem de máquina da Amazon (AMI) padrão especificando o seu próprio ID de AMI personalizado.  Exemplo: <code>ami-1f316660</code>	Nenhum	


Nome	Descrição	Padrão	Valores válidos
InstanceType	<p>O tipo de instância usado para executar a aplicação em um ambiente do Elastic Beanstalk.</p> <div data-bbox="326 401 889 1675" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Important</b></p> <p>A opção InstanceType está obsoleta. Ela foi substituída pela opção mais nova e mais eficiente InstanceTypes no namespace <a href="#">aws:ec2:instances</a>. Você pode usar essa nova opção para especificar uma lista de um ou mais tipos de instância para seu ambiente. O primeiro valor nessa lista é equivalente ao valor da opção InstanceType, incluída no namespace <code>aws:autoscaling:launchconfiguration</code> que é descrito aqui. Recomendamos que você especifique os tipos de instância usando a nova opção. Se especificada, a nova opção tem precedência sobre a anterior. Para obter mais informações, consulte <a href="#">the section called “O namespace aws:ec2:instances”</a>.</p> </div> <p>Os tipos de instância disponíveis dependem das zonas de disponibilidade e da região usadas. Se você</p>	Varia por conta e região.	<p>Um tipo de instância do EC2</p> <p>Varia de acordo com a conta, região e zona de disponibilidade. Você pode obter uma lista de tipos de instância do Amazon EC2 filtrados por esses valores. Para obter mais informações, consulte <a href="#">Tipos de instância disponíveis</a> no Guia do usuário do Amazon EC2 para instâncias do Linux ou <a href="#">tipos de instância</a> no Guia do usuário do Amazon EC2 para instâncias do Windows.</p>

Nome	Descrição	Padrão	Valores válidos
	<p>escolher uma sub-rede, a zona de disponibilidade que contém essa sub-rede determinará os tipos de instância disponíveis.</p> <ul style="list-style-type: none"><li>• O Elastic Beanstalk não é compatível com tipos de instância Mac do Amazon EC2.</li><li>• Para obter mais informações sobre as famílias e os tipos de instância do Amazon EC2, consulte <a href="#">Tipos de instância</a>, no Guia do usuário do Amazon EC2 para instâncias do Linux ou <a href="#">Tipos de instância</a> no Guia do usuário do Amazon EC2 para instâncias do Windows.</li><li>• Para obter mais informações os tipos de instâncias disponíveis nas regiões, consulte <a href="#">Tipos de instância disponíveis</a> no Guia do usuário do Amazon EC2 para instâncias do Linux ou <a href="#">tipos de instância</a> no Guia do usuário do Amazon EC2 para instâncias do Windows.</li></ul>		

 **Note**

Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um [arquivo de configuração](#). O console e a CLI do EB substituem essa

Nome	Descrição	Padrão	Valores válidos
	opção por um <a href="#">valor recomendado</a> .		

Nome	Descrição	Padrão	Valores válidos
LaunchTemplateTagPropagationEnabled	<p>Defina como <code>true</code> para habilitar a propagação de tags de ambiente para os modelos de execução de recursos específicos provisionados para o ambiente.</p> <p>O Elastic Beanstalk só pode propagar tags para modelos de execução para os seguintes recursos:</p> <ul style="list-style-type: none"><li>• Volumes do EBS</li><li>• Instâncias do EC2</li><li>• Interfaces de rede do EC2</li><li>• Modelos de execução do AWS CloudFormation que definem um recurso</li></ul> <p>Essa restrição existe porque o CloudFormation só permite tags na criação de modelos para recursos específicos. Para obter mais informações, consulte <a href="#">TagSpecification</a> no Guia do usuário do AWS CloudFormation.</p> <div data-bbox="326 1367 889 1862" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p> <b>Important</b></p><ul style="list-style-type: none"><li>• Alterar o valor dessa opção de <code>false</code> para <code>true</code> para um ambiente existente pode ser uma alteração significativa para tags já existentes.</li><li>• Quando esse recurso estiver habilitado, a propagação de tags exigirá a substituição</li></ul></div>	<code>false</code>	<code>true</code> <code>false</code>



Nome	Descrição	Padrão	Valores válidos
	<p>do EC2, o que pode resultar em tempo de inatividade. É possível habilitar atualizações contínuas para aplicar alterações de configuração em lotes e evitar tempo de inatividade durante o processo de atualização. Para obter mais informações, consulte <a href="#">Alterações de configuração</a>.</p> <p>Para obter mais informações sobre modelos de execução, consulte:</p> <ul style="list-style-type: none"> <li>• <a href="#">Modelos de execução</a> no Guia do usuário do Amazon EC2 Auto Scaling</li> <li>• <a href="#">Trabalhar com modelos</a> no Guia do usuário do AWS CloudFormation</li> <li>• <a href="#">Trechos de modelos do Elastic Beanstalk</a> no Guia do usuário do AWS CloudFormation</li> </ul> <p>Para obter mais informações sobre essa opção, consulte <a href="#">Propagação de tags para modelos de execução</a>.</p>		
Monitorin gInterval	O intervalo (em minutos) em que você deseja que as métricas do Amazon CloudWatch sejam retornadas.	5 minute	1 minute  5 minute


Nome	Descrição	Padrão	Valores válidos
SecurityGroups	<p>Lista os grupos de segurança do Amazon EC2 para atribuir às instâncias do EC2 no grupo de Auto Scaling para definir regras de firewall para as instâncias.</p> <p>É possível fornecer uma única string de valores separados por vírgulas que contém o nome de grupos de segurança do Amazon EC2 existentes ou referências aos recursos <code>AWS::EC2::SecurityGroup</code> criados no modelo. Os nomes de security group diferenciam letras maiúsculas de minúsculas.</p> <p>Se você usar a <a href="#">Amazon Virtual Private Cloud</a> (Amazon VPC) com o Elastic Beanstalk para que suas instâncias sejam executadas em uma virtual private cloud (VPC), especifique os IDs de grupos de segurança em vez de seus nomes.</p>	<code>elasticbeanstalk-default</code>	

Nome	Descrição	Padrão	Valores válidos
SSHSrcRestriction	<p>Usado para bloquear o acesso SSH a um ambiente. Por exemplo, você pode bloquear o acesso SSH para instâncias do EC2 de modo que apenas um bastion host possa acessar as instâncias na sub-rede privada.</p> <p>Essa string tem o seguinte formato:</p> <p><i>protocol, fromPort, toPort, source_restriction</i></p> <p><i>protocolo</i></p> <p>O protocolo para a regra de entrada.</p> <p><i>fromPort</i></p> <p>O número da porta inicial.</p> <p><i>toPort</i></p> <p>O número da porta final.</p> <p><i>source_restriction</i></p> <p>O intervalo de CIDR ou o nome de um grupo de segurança pelo qual o tráfego deve ser roteado. Para especificar um grupo de segurança a partir de outra conta (somente EC2-Classical, deve estar na mesma região), inclua o ID da conta antes do nome do grupo de segurança. Use o formato a seguir: <i>other_account_id /security_group_name</i> . Se você usar</p>	Nenhum	

Nome	Descrição	Padrão	Valores válidos
	<p>a <a href="#">Amazon Virtual Private Cloud</a> (Amazon VPC) com o Elastic Beanstalk para que suas instâncias sejam executadas em uma virtual private cloud (VPC), especifique o ID do grupo de segurança em vez de seu nome.</p> <p>Exemplo: <code>tcp, 22, 22, 54.240.196.185/32</code></p> <p>Exemplo: <code>tcp, 22, 22, my-security-group</code></p> <p>Exemplo (EC2-Classic): <code>tcp, 22, 22, 123456789012/their-security-group</code></p> <p>Exemplo (VPC): <code>tcp, 22, 22, sg-903004f8</code></p>		

Nome	Descrição	Padrão	Valores válidos
BlockDeviceMappings	<p>Anexe os volumes adicionais do Amazon EBS ou volumes de armazenamento de instância em todas as instâncias no grupo de Auto Scaling.</p> <p>Ao mapear volumes de armazenamento de instâncias, é necessário mapear apenas o nome do dispositivo para um nome de volume. No entanto, recomendamos que, ao mapear volumes do Amazon EBS, você também especifique alguns ou todos os campos a seguir (cada campo deve ser separado por dois pontos):</p> <ul style="list-style-type: none"> <li>• ID do snapshot</li> <li>• tamanho, em GB</li> <li>• excluir no encerramento (<code>true</code> ou <code>false</code>)</li> <li>• tipo de armazenamento (somente para <code>gp3</code>, <code>gp2</code>, <code>standard</code>, <code>st1</code>, <code>sc1</code> ou <code>io1</code>)</li> <li>• IOPS (somente para <code>gp3</code> ou <code>io1</code>)</li> <li>• throughput (somente para <code>gp3</code>)</li> </ul> <p>O exemplo a seguir anexa três volumes do Amazon EBS, um volume vazio de 100 GB <code>gp2</code> e um snapshot, um volume <code>io1</code> vazio de 20 GB com 2.000 IOPS provisionadas, e um volume de armazenamento de instâncias <code>ephemeral0</code>. Vários volumes de armazenamento de</p>	Nenhum	<ul style="list-style-type: none"> <li>• tamanho: deve ser entre 500 e 16384 GiB</li> <li>• throughput: deve ser entre 125 a 1000 mebibytes por segundo (MiB/s)</li> </ul>

Nome	Descrição	Padrão	Valores válidos
	<p>instâncias podem ser anexados se o tipo de instância for compatível com ele.</p> <pre>/dev/sdj=:100:true:gp2,/dev/sdh=snap-51eef269,/dev/sdi=:20:true:io1:2000,/dev/sdb=ephemeral0</pre>		
RootVolumeType	<p>Tipo de volume (SSD magnético, de uso geral ou SSD de IOPS provisionado) a ser usado para o volume raiz do Amazon EBS anexado às instâncias do EC2 para o seu ambiente.</p>	<p>Varia de acordo com a plataforma.</p>	<p>standard para armazenamento magnético.</p> <p>gp2 ou gp3 para SSD de uso geral.</p> <p>io1 para SSD com IOPS provisionadas.</p>
RootVolumeSize	<p>A capacidade de armazenamento do volume raiz do Amazon EBS em GB.</p> <p>Obrigatório se você definir RootVolumeType para SSD IOPS provisionadas.</p> <p>Por exemplo, "64".</p>	<p>Varia por plataforma para armazenamento magnético e SSD de uso geral.</p> <p>Nenhum para SSD IOPS provisionadas.</p>	<p>10 a 16384 GB para SSD de uso geral e SSD IOPS provisionadas.</p> <p>8 a 1024 GB para magnético.</p>

Nome	Descrição	Padrão	Valores válidos
RootVolumeIOPS	<p>As operações de entrada e saída por segundo (IOPS) desejadas para um volume raiz SSD com IOPS provisionadas ou para um volume raiz SSD gp3 de uso geral.</p> <p>A razão máxima entre IOPS e tamanho de volume é de 500 para 1. Por exemplo, um volume com 3000 IOPS deve ter pelo menos 6 GiB.</p>	Nenhum	<p>100 para 20000 para volumes raízes SSD io1 com IOPS provisionadas.</p> <p>3000 para 16000 para volumes raízes SSD gp3 de uso geral.</p>
RootVolumeThroughput	<p>O throughput desejado de mebibytes por segundo (MiB/s) para provisionar para o volume raiz do Amazon EBS associado à instância do EC2 de seu ambiente.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Essa opção só é aplicável a armazenamento do tipo gp3.</p> </div>	Nenhum	125 para 1000

## aws:autoscaling:scheduledaction

Configure [ações programadas](#) para o grupo de Auto Scaling do ambiente. Para cada ação, especifique um opção `resource_name` além do nome, namespace e valor da opção para cada configuração. Consulte [O namespace aws:autoscaling:scheduledaction](#) para ver exemplos.

### Namespace: **aws:autoscaling:scheduledaction**

Nome	Descrição	Padrão	Valores válidos
StartTime	Para ações únicas, escolha a data e a hora para executar a ação. Para	Nenhum	Um <a href="#">time stamp da ISO-8601</a>

Nome	Descrição	Padrão	Valores válidos
	ações recorrentes, escolha quando ativar a ação.		exclusivo em todas as ações de escalabilidade agendadas.
EndTime	<p>Uma data e hora no futuro (no fuso horário UTC/GMT) de quando você deseja que a ação de escalabilidade programada pare de repetir. Se você não especificar um EndTime, a ação será repetida de acordo com a expressão Recurrence .</p> <p>Exemplo: 2015-04-28T04:07:2Z</p> <p>Quando uma ação programada é finalizada, o Amazon EC2 Auto Scaling não retorna automaticamente para as configurações anteriores. Configure uma segunda ação programada para retornar às definições originais, conforme necessário.</p>	Nenhum	Um <a href="#">time stamp da ISO-8601</a> exclusivo em todas as ações de escalabilidade agendadas.
MaxSize	A contagem máxima de instâncias a ser aplicada quando a ação é executada.	Nenhum	0 para 10000
MinSize	A contagem mínima de instâncias a ser aplicada quando a ação é executada.	Nenhum	0 para 10000



Nome	Descrição	Padrão	Valores válidos
DesiredCapacity	Defina a capacidade inicial desejada para o grupo de Auto Scaling. Depois que a ação programada for aplicada, os triggers ajustarão a capacidade desejada com base em suas configurações.	Nenhum	0 para 10000
Recurrence	A frequência com que você deseja que a ação programada ocorra. Se você não especificar uma recorrência, a ação de escalabilidade ocorrerá somente uma vez, conforme especificado pelo <code>StartTime</code> .	Nenhum	Uma expressão <a href="#">Cron</a> .
Suspend	Configure como <code>true</code> para desativar uma ação recorrente programada temporariamente.	<code>false</code>	<code>true</code> <code>false</code>

## aws:autoscaling:trigger

Configure os triggers de escalabilidade para o grupo de Auto Scaling do ambiente.

### Note


Três opções nesse namespace determinam por quanto tempo uma métrica de acionador pode permanecer além dos limites definidos antes que o acionador seja disparado. Essas opções são relacionadas da seguinte forma:

$$\text{BreachDuration} = \text{Period} * \text{EvaluationPeriods}$$

Os valores padrão para essas opções (5, 5 e 1, respectivamente) resolvem essa equação.

Se você especificar valores inconsistentes, o Elastic Beanstalk poderá modificar um dos valores para que a equação ainda seja resolvida.

Namespace: **aws:autoscaling:trigger**

Nome	Descrição	Padrão	Valores válidos
BreachDuration	A quantidade de tempo, em minutos, que uma métrica pode estar além do limite definido (conforme especificado no <code>UpperThreshold</code> e <code>LowerThreshold</code> ) antes que o gatilho seja invocado.	5	1 para 600
LowerBreachScaleIncrement	Quantas instâncias do Amazon EC2 remover ao executar uma ação de escalabilidade.	-1	
LowerThreshold	Se a medida ficar abaixo desse número pela duração da violação, um gatilho é invocado.	2000000	0 para 20000000
MeasureName	A métrica usada para o acionador de Auto Scaling.  <div data-bbox="462 1096 1010 1850" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> <b>Note</b></p> <p><code>HealthyHostCount</code> , <code>UnhealthyHostCount</code> e <code>TargetResponseTime</code> são aplicáveis apenas a ambientes com um balanceador de carga dedicado. Estes não são valores de métrica válidos para ambientes configurados com um balanceador de carga compartilhado. Para obter mais informações sobre os tipos de balanceador de carga, consulte <a href="#">Balancead</a></p> </div>	NetworkOut	CPUUtilization  NetworkIn  NetworkOut  DiskWriteOps  DiskReadBytes  DiskReadOps  DiskWriteBytes  Latency  RequestCount

Nome	Descrição	Padrão	Valores válidos
	<p><a href="#">or de carga do ambiente do Elastic Beanstalk.</a></p>		<p>HealthyHostCount</p> <p>UnhealthyHostCount</p> <p>TargetResponseTime</p>
Period	<p>Especifica com que frequência o Amazon CloudWatch avalia as métricas do trigger. O valor é o número de minutos entre dois períodos consecutivos.</p>	5	1 para 600
EvaluationPeriods	<p>O número de períodos de avaliação consecutivos usados para determinar se está ocorrendo uma falha.</p>	1	1 para 600
Statistic	<p>A estatística que o acionador deve usar, como Average.</p>	Average	<p>Minimum</p> <p>Maximum</p> <p>Sum</p> <p>Average</p>

Nome	Descrição	Padrão	Valores válidos
Unit	Unidade de medição do gatilho, como Bytes.	Bytes	Seconds Percent Bytes Bits Count Bytes/Second Bits/Second Count/Second None
UpperBreachScaleIncrement	Especifica quantas instâncias do Amazon EC2 adicionar ao executar uma ação de escalabilidade.	1	
UpperThreshold	Se a medida ficar acima desse número pela duração da violação, um gatilho será invocado.	6000000	0 para 20000000

## aws:autoscaling:updatepolicy:rollingupdate

Configure atualizações contínuas para o grupo de Auto Scaling do ambiente.

Namespace: **aws:autoscaling:updatepolicy:rollingupdate**


Nome	Descrição	Padrão	Valores válidos
MaxBatchSize	O número de instâncias incluídas em cada lote da atualização contínua.	Um terço do tamanho mínimo do grupo de Auto Scaling arredonda	1 para 10000

Nome	Descrição	Padrão	Valores válidos
		do para o próximo valor inteiro.	
MinInstancesInService	O número mínimo de instâncias que devem estar em serviço dentro do grupo de Auto Scaling enquanto outras instâncias são encerradas.	O tamanho mínimo do grupo de Auto Scaling ou um inferior ao tamanho máximo do grupo de Auto Scaling, qualquer que seja o número menor.	0 para 9999

Nome	Descrição	Padrão	Valores válidos
RollingUpdateEnabled	<p>Se true, ele permite atualizações contínuas para um ambiente. As atualizações contínuas são úteis quando você precisa fazer pequenas atualizações frequentes na aplicação de software do Elastic Beanstalk e quer evitar tempo de inatividade da aplicação.</p> <p>A configuração desse valor para verdadeiro ativará automaticamente as opções <code>MaxBatchSize</code>, <code>MinInstancesInService</code> e <code>PauseTime</code>. A definição de qualquer uma dessas opções também define automaticamente a opção de valor <code>RollingUpdateEnabled</code> para true. A definição dessa opção para false</p>	false	true false

Nome	Descrição	Padrão	Valores válidos
	<p>desativa as atualizações contínuas.</p> <div data-bbox="558 331 878 1318" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um <a href="#">arquivo de configuração</a>. O console e a CLI do EB substituem essa opção por um <a href="#">valor recomendado</a>.</p></div>		

Nome	Descrição	Padrão	Valores válidos
RollingUpdateType	<p>Isso inclui três tipos: atualizações contínuas baseadas em tempo, atualizações contínuas baseadas em saúde e atualizações imutáveis.</p> <p>Atualizações contínuas com base em tempo aplicam um <code>PauseTime</code> entre lotes. Atualizações contínuas baseadas na integridade aguardam até que as novas instâncias passem nas verificações de integridade antes de mudar para o próximo lote. <a href="#">Atualizações imutáveis</a> iniciam um conjunto completo de instâncias em um novo grupo de Auto Scaling.</p>	Time	Time Health Immutable

 **Note**

Se você usar o console do Elastic Beanstalk ou



Nome	Descrição	Padrão	Valores válidos
	<p>a CLI do EB para criar um ambiente, não será possível definir essa opção em um <a href="#">arquivo de configuração</a>. O console e a CLI do EB substituem essa opção por um <a href="#">valor recomendado</a>.</p>		
PauseTime	<p>A quantidade de tempo (em segundos, minutos ou horas) que o serviço do Elastic Beanstalk aguarda após a conclusão das atualizações para um lote de instâncias e antes que ele prossiga para o próximo lote.</p>	<p>Calculado automaticamente com base no tipo de instância e contêiner.</p>	<p>PT0S* (0 segundos) a PT1H (1 hora)</p>

Nome	Descrição	Padrão	Valores válidos
Timeout	O tempo máximo de espera (em minutos ou horas) para que todas as instâncias em um lote de instâncias passem pelas verificações de integridade antes de cancelar a atualização.	PT30M (30 minutos)	PT5M* (5 minutos) a PT1H (1 hora)  *Formato da <a href="#">duração de ISO8601</a> : PT#H#M#S em que cada # é o número de horas, minutos e segundos, respectivamente.

## aws:ec2:instances


Configure as instâncias do seu ambiente, incluindo opções de spot. Este namespace complementa [aws:autoscaling:launchconfiguration](#) e [aws:autoscaling:asg](#).

Para obter mais informações, consulte [the section called “Grupo de Auto Scaling”](#).

Namespace: **aws:ec2:instances**


Nome	Descrição	Padrão	Valores válidos
EnableSpot	Habilite solicitações de instância spot para seu ambiente. Quando <code>false</code> , algumas opções neste namespace não entram em vigor.	<code>false</code>	<code>true</code>  <code>false</code>
InstanceTypes	Uma lista separada por vírgulas dos tipos de instância que você deseja que o ambiente use (por exemplo, <code>t2.micro,t3.micro</code> ).  Quando as instâncias Spot não estão ativadas ( <code>EnableSpot</code> é <code>false</code> ),	Uma lista de dois tipos de instâncias.	Um a dez tipos de instância do EC2  Recomendamos pelo menos dois  Varia de acordo com a conta, região e zona de disponibilidade. Você

Nome	Descrição	Padrão	Valores válidos
	<p>apenas o primeiro tipo de instância na lista é usado.</p> <p>O primeiro tipo de instância na lista nesta opção é equivalente ao valor da opção <code>InstanceType</code> no namespace <a href="#">aws:autoscaling:launchconfiguration</a>. Não recomendamos usar essa última opção porque ela está obsoleta. Se você especificar ambos, o primeiro tipo de instância na lista na opção <code>InstanceTypes</code> será usado e <code>InstanceType</code> será ignorado.</p> <p>Os tipos de instância disponíveis dependem das zonas de disponibilidade e da região usadas. Se você escolher uma sub-rede, a zona de disponibilidade que contém essa sub-rede determinará os tipos de instância disponíveis.</p> <ul style="list-style-type: none"> <li>• O Elastic Beanstalk não é compatível com tipos de instância Mac do Amazon EC2.</li> <li>• Para obter mais informações sobre as famílias e os tipos de instância do Amazon EC2, consulte <a href="#">Tipos de instância</a>, no Guia do usuário do Amazon EC2 para instâncias do Linux ou <a href="#">Tipos de instância</a> no Guia do usuário do Amazon EC2 para instâncias do Windows.</li> </ul>	<p>Varia por conta e região.</p>	<p>pode obter uma lista de tipos de instância do Amazon EC2 filtrados por esses valores. Para obter mais informações, consulte <a href="#">Tipos de instância disponíveis</a> no Guia do usuário do Amazon EC2 para instâncias do Linux ou <a href="#">tipos de instância</a> no Guia do usuário do Amazon EC2 para instâncias do Windows.</p> <p>Os tipos de instância devem fazer parte da mesma arquitetura (arm64, x86_64, i386).</p> <p>Supported Architectures também faz parte desse namespace. Se você fornecer valores para Supported Architectures, o(s) valor(es) inserido(s) para <code>InstanceTypes</code> deve(m) pertencer a uma, e apenas uma, arquitetura que você fornece</p>

Nome	Descrição	Padrão	Valores válidos
	<ul style="list-style-type: none"> <li>Para obter mais informações os tipos de instâncias disponíveis nas regiões, consulte <a href="#">Tipos de instância disponíveis</a> no Guia do usuário do Amazon EC2 para instâncias do Linux ou <a href="#">tipos de instância</a> no Guia do usuário do Amazon EC2 para instâncias do Windows.</li> </ul> <div data-bbox="448 653 997 1493" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>Algumas contas da AWS mais antigas podem fornecer o Elastic Beanstalk com tipos de instância padrão que não oferecem suporte a instâncias spot (por exemplo, t1.micro). Se você ativar solicitações de instância Spot e receber um erro sobre um tipo de instância que não é compatível com Spot, configure tipos de instância que garantam essa compatibilidade. Para escolher os tipos de instância spot, use o <a href="#">Spot Instance Advisor</a>.</p> </div> <p>Quando você atualiza a configuração do ambiente e remove um ou mais tipos de instância da opção InstanceTypes, o Elastic Beanstalk encerra todas as instâncias do Amazon EC2 em execução em</p>		para Supported Architectures.

Nome	Descrição	Padrão	Valores válidos
	qualquer um dos tipos de instância removidos. O grupo de Auto Scaling do seu ambiente executa novas instâncias, conforme necessário, para completar a capacidade desejada, usando os tipos de instância especificados atuais.		
SpotFleetOnDemandBase	O número mínimo de instâncias sob demanda que o grupo de Auto Scaling provisiona antes de considerar as instâncias spot à medida que o ambiente é dimensionado.  Esta opção é relevante somente quando <code>EnableSpot</code> é <code>true</code> .	0	0 para a opção <code>MaxSize</code> no namespace <a href="#">aws:autoscaling:asg</a>
SpotFleetOnDemandAboveBasePercentage	O percentual de instâncias sob demanda como parte da capacidade adicional que o grupo de Auto Scaling provisiona além das instâncias <code>SpotOnDemandBase</code> .  Esta opção é relevante somente quando <code>EnableSpot</code> é <code>true</code> .	0 para um ambiente de instância única  70 para um ambiente com balanceamento de carga	0 para 100

Nome	Descrição	Padrão	Valores válidos
SpotMaxPrice	<p>O preço máximo por hora, em US \$, que você está disposto a pagar por uma instância spot. Para obter recomendações sobre opções de preço máximo para instâncias spot, consulte o <a href="#">Histórico de definição de preço da instância spot</a> no Manual do usuário do Amazon EC2 para instâncias do Linux.</p> <p>Esta opção é relevante somente quando <code>EnableSpot</code> é <code>true</code>.</p>	<p>Preço sob demanda para cada tipo de instância. O valor da opção neste caso é <code>null</code>.</p>	<p><code>0.001</code> para <code>20.0</code> <code>null</code></p>

Nome	Descrição	Padrão	Valores válidos
Arquiteturas permitidas	<p>Uma lista separada por vírgulas dos tipos de arquitetura da instância do EC2 que você usará para seu ambiente.</p> <p>O Elastic Beanstalk é compatível com tipos de instância baseados nas seguintes arquiteturas de processador:</p> <ul style="list-style-type: none"> <li>• Arquitetura Arm Graviton 64 bits (arm64) da AWS</li> <li>• Arquitetura de 64 bits (x86_64)</li> <li>• Arquitetura de 32 bits (i386)</li> </ul> <p>Para obter mais informações sobre arquitetura de processador e tipos de instância do Amazon EC2, consulte <a href="#">the section called “Tipos de instância do Amazon EC2”</a>.</p>	Nenhum	<p>arm64</p> <p>x86_64</p> <p>i386</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>A arquitetura de 32 bits i386 não é compatível com a maioria das plataformas do Elastic Beanstalk. Recomendamos que você escolha os tipos de arquitetura x86_64 ou arm64 em vez disso.</p> </div>

## aws:ec2:vpc

Configure seu ambiente para iniciar recursos em uma [Amazon Virtual Private Cloud](#) (Amazon VPC) personalizada. Se você não definir configurações neste namespace, o Elastic Beanstalk executará recursos na VPC padrão.

Namespace: **aws:ec2:vpc**

Nome	Descrição	Padrão	Valores válidos
VPCId	O ID da sua Amazon VPC.	Nenhum	

Nome	Descrição	Padrão	Valores válidos
Subnets	Os IDs da sub-rede ou sub-redes do grupo de Auto Scaling. Se você tiver várias sub-redes, especifique o valor como uma string separada por vírgulas simples de IDs de sub-redes (por exemplo, "subnet-11111111, subnet-22222222" ).	Nenhum	
ELBSubnets	Os IDs da sub-rede ou das sub-redes do elastic load balancer. Se você tiver várias sub-redes, especifique o valor como uma string separada por vírgulas simples de IDs de sub-redes (por exemplo, "subnet-11111111, subnet-22222222" ).	Nenhum	
ELBScheme	Especifique <code>internal</code> se você deseja criar um balanceador de carga interno na Amazon VPC para que a aplicação do Elastic Beanstalk não seja acessada de fora dela. Se você especificar um valor diferente de <code>public</code> ou <code>internal</code> , o Elastic Beanstalk ignorará o valor.	<code>public</code>	<code>public</code> <code>internal</code>
DBSubnets	Contém os IDs de sub-redes de banco de dados. Opção usada somente se você deseja adicionar uma instância de banco de dados do Amazon RDS como parte do seu aplicativo. Se você tiver várias sub-redes , especifique o valor como uma string separada por vírgulas simples de IDs de sub-redes (por exemplo, "subnet-11111111, subnet-22222222" ).	Nenhum	



Nome	Descrição	Padrão	Valores válidos
AssociatePublicIpAddress	<p>Especifica se é para executar instâncias com endereços IP públicos na Amazon VPC. Instâncias com endereços IP públicos não exigem dispositivo NAT para comunicação com a Internet. Você deve definir o valor como <code>true</code> para incluir o balanceador de carga e as instâncias em uma única sub-rede pública.</p> <p>Essa opção não tem efeito em um ambiente de instância única, o qual sempre tem uma única instância do Amazon EC2 com um endereço IP elástico. A opção é relevante para ambientes escaláveis e com balanceamento de carga.</p>	Nenhum	<code>true</code> <code>false</code>

## aws:elasticbeanstalk:application

Configure um caminho de verificação de integridade para seu aplicativo. Para obter mais informações, consulte [Relatórios de integridade básica](#).

Namespace: **aws:elasticbeanstalk:application**

Nome	Descrição	Padrão	Valores válidos
URL de verificação de integridade do aplicativo	<p>O caminho para o qual as solicitações de verificação de integridade são enviadas. Se esse caminho não estiver definido, o balanceador de carga tentará fazer uma conexão TCP na porta 80 para verificar o status de integridade do aplicativo. Configure para um caminho iniciando com <code>/</code> para enviar uma solicitação HTTP GET para esse caminho. Você também pode incluir um protocolo (HTTP, HTTPS, TCP ou SSL) e uma</p>	Nenhum	<p>Os valores válidos são:</p> <p><code>/</code> (HTTP GET para caminho raiz)</p> <p><code>/health</code></p> <p><code>HTTPS:443/</code></p> <p><code>HTTPS:443/ health</code></p>

Nome	Descrição	Padrão	Valores válidos
	<p>porta antes do caminho para verificar a conectividade HTTPS ou usar uma porta não padrão.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p><b>Note</b></p> <p>Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um <a href="#">arquivo de configuração</a>. O console e substitui essa opção por um <a href="#">valor recomendado</a>.</p> </div>		

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte [Valores recomendados](#).

## aws:elasticbeanstalk:application:environment

Configure propriedades do ambiente para seu aplicativo.

Namespace: **aws:elasticbeanstalk:application:environment**

Nome	Descrição	Padrão	Valores válidos
Qualquer nome de variável de ambiente.	Passe em pares de chave-val or.	Nenhum	Qualquer valor de variável de ambiente.

Consulte [Propriedades de ambientes e outras configurações de software](#) para obter mais informações.

## aws:elasticbeanstalk:cloudwatch:logs

Configure o streaming de logs de instâncias para seu aplicativo.

Namespace: **aws:elasticbeanstalk:cloudwatch:logs**

Nome	Descrição	Padrão	Valores válidos
StreamLogs	Especifica se é necessário criar grupos nos logs do CloudWatch Logs para logs de proxy e implantação e fazer streaming dos logs de cada instância no ambiente.	false	true false
DeleteOnTerminate	Especifica se os grupos de log devem ser excluídos quando o ambiente é encerrado. Se for false, os logs serão mantidos por RetentionInDays dias.	false	true false
RetentionInDays	O número de dias que os eventos de log serão mantidos antes que expirem.	7	1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653

## aws:elasticbeanstalk:cloudwatch:logs:health

Configure o streaming de logs de integridade do ambiente para seu aplicativo.

Namespace: **aws:elasticbeanstalk:cloudwatch:logs:health**

Nome	Descrição	Padrão	Valores válidos
HealthStreamingEnabled	Para ambientes com relatórios de integridade aprimorada habilitados, especifica se deve ser criado um grupo no CloudWatch Logs para integridade	false	true false


Nome	Descrição	Padrão	Valores válidos
	do ambiente e arquivar dados de integridade do ambiente do Elastic Beanstalk. Para obter informações sobre como habilitar a integridade avançada, visite <a href="https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/healthreporting-system.html">aws:elasticbeanstalk:healthreporting:system</a> .		
DeleteOnTerminate	Especifica se o grupo de logs deve ser excluído quando o ambiente for encerrado. Se false, os dados de integridade são mantidos por RetentionInDays dias.	false	true false
RetentionInDays	O número de dias para manter os dados de integridade de arquivados antes expirem.	7	1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653

## aws:elasticbeanstalk:command

Configure a política de implantação para o código do aplicativo. Para obter mais informações, consulte [the section called “Opções de implantação”](#).

Namespace: **aws:elasticbeanstalk:command**

Nome	Descrição	Padrão	Valores válidos
DeploymentPolicy	<p>Escolha uma <a href="#">política de implantação</a> para implantações de versões do aplicativo.</p> <div data-bbox="391 472 1057 884" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p><b>Note</b></p> <p>Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um <a href="#">arquivo de configuração</a>. O console e substitui essa opção por um <a href="#">valor recomendado</a>.</p> </div>	AllAtOnce	<p>AllAtOnce</p> <p>Rolling</p> <p>RollingWithAdditionalBatch</p> <p>Immutable</p> <p>TrafficSplitting</p>
Timeout	<p>A quantidade de tempo, em segundos, para aguardar uma instância para concluir a execução de comandos.</p> <p>O Elastic Beanstalk adiciona internamente 240 segundos (quatro minutos) ao valor Timeout. Por exemplo, o tempo limite efetivo por padrão é 840 segundos (600 + 240) ou 14 minutos.</p>	600	1 para 3600
BatchSizeType	<p>O tipo de número especificado no BatchSize.</p> <div data-bbox="391 1465 1057 1793" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p><b>Note</b></p> <p>Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um <a href="#">arquivo de configuração</a>. O console e a CLI do EB</p> </div>	Porcentagem	<p>Porcentagem</p> <p>Fixed</p>


Nome	Descrição	Padrão	Valores válidos
	<p>substituem essa opção por um <a href="#">valor recomendado</a>.</p>		
BatchSize	<p>A porcentagem ou o número fixo de instâncias do Amazon EC2 no grupo Auto Scaling para executar implantações simultaneamente. Os valores válidos variam dependendo da configuração do BatchSizeType usado.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um <a href="#">arquivo de configuração</a>. O console e a CLI do EB substituem essa opção por um <a href="#">valor recomendado</a>.</p> </div>	100	<p>1 para 100 (Porcentagem).</p> <p>1 a <a href="#">aws:autoscaling:asg::MaxSize</a> (Fixed)</p>
IgnoreHealthCheck	Não cancele uma implantação devido a reprovação nas verificações de integridade.	false	<p>true</p> <p>false</p>

## aws:elasticbeanstalk:environment

Configure a arquitetura e a função de serviço de seu ambiente.

Namespace: **aws:elasticbeanstalk:environment**

Nome	Descrição	Padrão	Valores válidos
EnvironmentType	Defina como SingleInstance para executar uma instância do EC2 sem balanceador de carga.	LoadBalanced	SingleInstance  LoadBalanced
ServiceRole	<p>O nome de uma função do IAM que o Elastic Beanstalk usa para gerenciar recursos do ambiente. Especifique um nome de função (opcionalmente acompanhados de prefixo com um caminho personalizado) ou o ARN dele.</p> <p>Exemplos:</p> <ul style="list-style-type: none"> <li>aws-elasticbeanstalk-service-role</li> <li><i>custom-path /custom-role</i></li> <li>arn:aws:iam::123456789012:role/aws-elasticbeanstalk-service-role</li> </ul>	Nenhum	Nome da função do IAM, caminho/nome ou ARN

 **Note**

Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um [arquivo de configuração](#). O console e a CLI do EB substituem essa opção por um [valor recomendado](#).

Nome	Descrição	Padrão	Valores válidos
LoadBalancerType	O tipo de balanceador de carga para o seu ambiente. Para obter mais informações, consulte <a href="#">the section called “Load balancer”</a> .	classic	classic application network
LoadBalancerIsShared	Especifica se o balanceador de carga do ambiente é dedicado ou compartilhado. Essa opção só pode ser definida para um Application Load Balancer. Ela não pode ser alterada após a criação do ambiente.  Se false, o ambiente terá seu próprio balanceador de carga dedicado, criado e gerenciado pelo Elastic Beanstalk. Se true, o ambiente usará um balanceador de carga compartilhado, criado por você e especificado na opção SharedLoadBalancer do namespace <a href="#">aws:elbv2:loadbalancer</a> .	false	true false

## aws:elasticbeanstalk:environment:process:default

Configure o processo padrão de seu ambiente.

Namespace: **aws:elasticbeanstalk:environment:process:default**

Nome	Descrição	Padrão	Valores válidos
DeregistrationDelay	A quantidade de tempo, em segundos, para aguardar a conclusão das solicitações ativas	20	0 para 3600



Nome	Descrição	Padrão	Valores válidos
	antes de cancelar o registro.		
HealthCheckInterval	O intervalo de tempo, em segundos, que o Elastic Load Balancing verifica a integridade das instâncias do Amazon EC2 do seu aplicativo.	Com Classic ou Application Load Balancer: 15  Com Network Load Balancer: 30	Com Classic ou Application Load Balancer: 5 a 300  Com Network Load Balancer: 10, 30
HealthCheckPath	O caminho para o qual as solicitações HTTP para verificações de integridade são enviadas.	/	Um caminho roteável.
HealthCheckTimeout	A quantidade de tempo, em segundos, para aguardar uma resposta durante uma verificação de integridade.  Essa opção só é aplicável a ambientes com um Application Load Balancer.	5	1 para 60
HealthyThresholdCount	O número de solicitações consecutivas bem-sucedidas antes de o Elastic Load Balancing alterar o status de integridade da instância.	Com Classic ou Application Load Balancer: 3  Com Network Load Balancer: 5	2 para 10

Nome	Descrição	Padrão	Valores válidos
MatcherHTTPCode	<p>Uma lista separada por vírgulas de códigos HTTP que indicam que uma instância está íntegra.</p> <p>Essa opção só é aplicável a ambientes com um Network ou Application Load Balancer.</p>	200	<p>Com Application Load Balancer: 200 a 499</p> <p>Com Network Load Balancer: 200 a 399</p>
Port	Porta que o processo escuta.	80	1 para 65535
Protocol	<p>O protocolo que o processo usa.</p> <p>Com Application Load Balancer, você só pode definir essa opção como HTTP ou HTTPS.</p> <p>Com Network Load Balancer, você só pode definir essa opção como TCP.</p>	<p>Com Classic ou Application Load Balancer: HTTP</p> <p>Com Network Load Balancer: TCP</p>	<p>TCP</p> <p>HTTP</p> <p>HTTPS</p>

Nome	Descrição	Padrão	Valores válidos
StickinessEnabled	<p>Defina como verdadeiro para ativar sticky sessions.</p> <p>Essa opção só é aplicável a ambientes com um Application Load Balancer.</p>	'false'	'false'  'true'
StickinessLBCookieDuration	<p>A duração, em segundos, do cookie de sessão do sticky.</p> <p>Essa opção só é aplicável a ambientes com um Application Load Balancer.</p>	86400 (um dia)	1 para 604800
StickinessType	<p>Defina como <code>lb_cookie</code> para usar cookies nas sticky sessions.</p> <p>Essa opção só é aplicável a ambientes com um Application Load Balancer.</p>	<code>lb_cookie</code>	<code>lb_cookie</code>
UnhealthyThresholdCount	<p>Solicitações consecutivas sem êxito antes de o Elastic Load Balancing alterar o status de integridade da instância.</p>	5	2 para 10

**aws:elasticbeanstalk:environment:process:process\_name**

Configure processos adicionais para o seu ambiente.

Namespace: **aws:elasticbeanstalk:environment:process:process\_name**

Nome	Descrição	Padrão	Valores válidos
DeregistrationDelay	A quantidade de tempo, em segundos, para aguardar a conclusão das solicitações ativas antes de cancelar o registro.	20	0 para 3600
HealthCheckInterval	O intervalo, em segundos, que o Elastic Load Balancing verifica a integridade das instâncias do Amazon EC2 para seu aplicativo.	Com Classic ou Application Load Balancer: 15  Com Network Load Balancer: 30	Com Classic ou Application Load Balancer: 5 a 300  Com Network Load Balancer: 10, 30
HealthCheckPath	O caminho para o qual as solicitações HTTP para verificações de integridade são enviadas.	/	Um caminho roteável.
HealthCheckTimeout	A quantidade de tempo, em segundos, para aguardar uma resposta durante uma verificação de integridade.	5	1 para 60

Nome	Descrição	Padrão	Valores válidos
	Essa opção só é aplicável a ambientes com um Application Load Balancer.		
HealthyThresholdCount	O número de solicitações consecutivas bem-sucedidas antes de o Elastic Load Balancing alterar o status de integridade da instância.	Com Classic ou Application Load Balancer: 3  Com Network Load Balancer: 5	2 para 10
MatcherHTTPCode	Uma lista separada por vírgulas de códigos HTTP que indicam que uma instância está íntegra.  Essa opção só é aplicável a ambientes com um Network ou Application Load Balancer.	200	Com Application Load Balancer: 200 a 499  Com Network Load Balancer: 200 a 399
Port	A porta que o processo escuta.	80	1 para 65535

Nome	Descrição	Padrão	Valores válidos
Protocol	<p>O protocolo que o processo usa.</p> <p>Com Application Load Balancer, você só pode definir essa opção como HTTP ou HTTPS.</p> <p>Com Network Load Balancer, você só pode definir essa opção como TCP.</p>	<p>Com Classic ou Application Load Balancer: HTTP</p> <p>Com Network Load Balancer: TCP</p>	<p>TCP</p> <p>HTTP</p> <p>HTTPS</p>
StickinessEnabled	<p>Defina como verdadeiro para ativar sticky sessions.</p> <p>Essa opção só é aplicável a ambientes com um Application Load Balancer.</p>	'false'	'false' 'true'
StickinessLBCookieDuration	<p>A duração, em segundos, do cookie de sessão do sticky.</p> <p>Essa opção só é aplicável a ambientes com um Application Load Balancer.</p>	86400 (um dia)	1 para 604800

Nome	Descrição	Padrão	Valores válidos
StickinessType	<p>Defina como <code>lb_cookie</code> para usar cookies nas sticky sessions.</p> <p>Essa opção só é aplicável a ambientes com um Application Load Balancer.</p>	<code>lb_cookie</code>	<code>lb_cookie</code>
UnhealthyThresholdCount	Solicitações consecutivas sem êxito antes de o Elastic Load Balancing alterar o status de integridade da instância.	5	2 para 10

## aws:elasticbeanstalk:environment:proxy:staticfiles

Você pode usar o namespace a seguir para configurar o servidor de proxy para atender arquivos estáticos. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo. Isso reduz o número de solicitações que seu aplicativo deve processar.

Mapeie um caminho fornecido pelo servidor de proxy para uma pasta no código-fonte que contenha ativos estáticos. Cada opção definida neste namespace mapeia um caminho diferente.

### Note

Esse namespace se aplica a ramificações da plataforma baseadas no Amazon Linux 2 e versões posteriores. Se o ambiente usar uma versão de plataforma baseada na AMI do Amazon Linux (anterior ao Amazon Linux 2), consulte [the section called “Opções específicas para a plataforma”](#) para os namespaces de arquivos estáticos específicos da plataforma.


Namespace: **aws:elasticbeanstalk:environment:proxy:staticfiles**

Nome	Valor
<p>O caminho no qual o servidor proxy serve os arquivos. Inicie o valor com /.</p> <p>Por exemplo, especifique <code>/imagens</code> para servir arquivos em <code>subdomain.eelasticbeanstalk.com/imagens</code>.</p>	<p>O nome da pasta que contém os arquivos.</p> <p>Por exemplo, especifique <code>staticimages</code> para servir arquivos de uma pasta denominada <code>staticimages</code> no nível mais alto do seu pacote de origem.</p>



## aws:elasticbeanstalk:healthreporting:system

Configure relatórios de integridade aprimorados para o seu ambiente.

Namespace: **aws:elasticbeanstalk:healthreporting:system**

Nome	Descrição	Padrão	Valores válidos
SystemType	<p>O sistema de relatórios de integridade (<a href="#">básicos</a> ou <a href="#">aprimorados</a>). Os relatórios de integridade aprimorados requerem uma <a href="#">função de serviço</a> e uma <a href="#">versão da plataforma</a> versão 2 ou mais nova.</p> <div data-bbox="423 1283 1122 1696" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um <a href="#">arquivo de configuração</a>. O console e a CLI do EB substituem essa opção por um <a href="#">valor recomendado</a>.</p> </div>	basic	basic enhanced
ConfigDocument	Um documento JSON que descreve as métricas da instância e do ambiente para publicar no CloudWatch.	Nenhum	



Nome	Descrição	Padrão	Valores válidos
EnhancedHealthAuthEnabled	<p>Permite a autorização para a API interna que o Elastic Beanstalk usa para comunicar informações de integridade aprimorada das instâncias do ambiente para o serviço do Elastic Beanstalk.</p> <p>Para obter mais informações, consulte <a href="#">the section called “Funções de integridade aperfeiçoada”</a>.</p> <div data-bbox="423 667 1122 984" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Essa opção só é aplicável a relatórios de integridade aprimorados (como quando <code>SystemType</code> está definido como <code>enhanced</code>).</p> </div>	true	true false
HealthCheckSuccessThreshold	<p>Reduz o limite para que as instâncias passem nas verificações de integridade.</p> <div data-bbox="423 1150 1122 1556" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um <a href="#">arquivo de configuração</a>. O console e substitui essa opção por um <a href="#">valor recomendado</a>.</p> </div>	0k	0k Warning Degraded Severe

## aws:elasticbeanstalk:hostmanager

Configure as instâncias do EC2 no ambiente para fazer upload de logs alternados para o Amazon S3.

Namespace: **aws:elasticbeanstalk:hostmanager**

Nome	Descrição	Padrão	Valores válidos
LogPublic ationControl	Copie os arquivos de log das instâncias do Amazon EC2 da sua aplicação para o bucket do Amazon S3 que está associado ao aplicativo.	false	true  false

## aws:elasticbeanstalk:managedactions

Configure as atualizações gerenciadas da plataforma para o ambiente.

Namespace: **aws:elasticbeanstalk:managedactions**

Nome	Descrição	Padrão	Valores válidos
ManagedActionsEnabled	Ative as <a href="#">atualizações gerenciadas da Plataforma</a> .  Ao definir essa opção para true, também é necessário especificar PreferredStartTime e <a href="#">UpdateLevel</a> .	false	true  false
PreferredStartTime	Configure uma janela de manutenção para ações gerenciadas em UTC.  Por exemplo, "Tue:09:00" .	Nenhum	Dia e hora no <i>dia:hora:minuto</i> format.
ServiceRoleForManagedUpdates	O nome de uma função do IAM que o Elastic Beanstalk usa para executar atualizações gerenciadas de plataforma para o ambiente.	Nenhum	Igual a ServiceRole ou

Nome	Descrição	Padrão	Valores válidos
	É possível usar a mesma função especificada para a opção <code>ServiceRole</code> do namespace <code>aws:elasticbeanstalk:environment</code> ou a <a href="#">função vinculada ao serviço de atualizações gerenciadas</a> da conta. No último caso, se a conta ainda não tiver uma função vinculada ao serviço de atualizações gerenciadas, o Elastic Beanstalk a criará.		<code>AWSServiceRoleForElasticBeanstalkManagedUpdates</code>

`aws:elasticbeanstalk:managedactions:platformupdate`

Configure as atualizações gerenciadas da plataforma para o ambiente.

Namespace: **`aws:elasticbeanstalk:managedactions:platformupdate`**


Nome	Descrição	Padrão	Valores válidos
<code>UpdateLevel</code>	O nível mais alto de atualização a aplicar com atualizações gerenciadas da plataforma. As Plataformas são versões <i>principais</i> <i>secundárias</i> <i>patch</i> . Por exemplo, a versão 2.0.8 tem uma versão principal 2, uma versão secundária 0 e uma versão de patch 8.	Nenhum	<code>patch</code> somente para atualizações de versão de patch.  <code>menor</code> para atualizações de versão secundária e de patch.

Nome	Descrição	Padrão	Valores válidos
InstanceRefreshEnabled	<p>Ative substituição de instância semanal.</p> <p>Isso requer que <code>ManagedActionsEnabled</code> seja definido como <code>true</code>.</p>	false	<p>true</p> <p>false</p>

## aws:elasticbeanstalk:monitoring

Configure seu ambiente para encerrar instâncias do EC2 que falharem nas verificações de integridade.

Namespace: **aws:elasticbeanstalk:monitoring**

Nome	Descrição	Padrão	Valores válidos
Automatically Terminate Unhealthy Instances	<p>Encerre uma instância se ela falhar nas verificações de integridade.</p> <div data-bbox="500 1224 954 1885" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Essa opção só é compatível com <a href="#">ambientes legados</a>. Ela determina a integridade de uma instância com base em poder acessá-la e em outras métricas com base em instâncias.</p> <p>O Elastic Beanstalk não fornece uma maneira de encerrar instâncias</p> </div>	true	<p>true</p> <p>false</p>

Nome	Descrição	Padrão	Valores válidos
	automaticamente com base na integridade da aplicação.		

## aws:elasticbeanstalk:sns:topics

Configure as notificações de seu ambiente.

Namespace: **aws:elasticbeanstalk:sns:topics**

Nome	Descrição	Padrão	Valores válidos
Notification Endpoint	O endpoint em que você deseja ser notificado sobre eventos importantes que afetam a aplicação.	Nenhum	
	<div data-bbox="378 1060 881 1566" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p><b>Note</b></p> <p>Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um <a href="#">arquivo de configuração</a>. O console e substitui essa opção por um <a href="#">valor recomendado</a>.</p> </div>		
Notification Protocol	O protocolo usado para enviar notificações ao endpoint.	email	http https email email-json


Nome	Descrição	Padrão	Valores válidos
			sqs
Notification Topic ARN	O nome de recurso da Amazon (ARN) do tópico no qual você se inscreveu.	Nenhum	
Notification Topic Name	O nome do tópico que você assinou.	Nenhum	

aws:elasticbeanstalk:sqsd


Configure a fila do Amazon SQS para um ambiente de operador.

Namespace: **aws:elasticbeanstalk:sqsd**

Nome	Descrição	Padrão	Valores válidos
WorkerQueueURL	A URL da fila em que o daemon no nível de ambiente de trabalho lê as mensagens	gerado automaticamente	Se você não especificar um valor, o Elastic Beanstalk criará automaticamente uma fila.

 **Note**

Quando você não especifica um valor, a fila que o Elastic Beanstalk cria automaticamente é uma fila [padrão](#) do Amazon SQS. Quando você fornece um valor, pode informar o URL de uma fila padrão ou [FIFO](#) do Amazon SQS. Lembre-se de que, se você fornecer uma fila FIFO,

Nome	Descrição	Padrão	Valores válidos
	as <a href="#">tarefas periódicas</a> não serão compatíveis.		
HttpPath	O caminho relativo para a aplicação para a qual as mensagens HTTP POST são enviadas.	/	
MimeType	O tipo MIME da mensagem enviada na solicitação HTTP POST.	application/json	application/json application/x-www-form-urlencoded application/xml text/plain Tipo MIME personalizado.
HttpConnections	O número máximo de conexões simultâneas para quaisquer aplicações dentro de uma instância do Amazon EC2.	50	1 para 100
	<p> <b>Note</b></p> <p>Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um <a href="#">arquivo de configuração</a>. O console e substitui essa opção por um <a href="#">valor recomendado</a>.</p>		

Nome	Descrição	Padrão	Valores válidos
ConnectTimeout	O tempo, em segundos, a aguardar por conexões bem-sucedidas com um aplicativo.	5	1 para 60
InactivityTimeout	O tempo, em segundos, a aguardar por uma resposta para uma conexão existente com um aplicativo. A mensagem é reprocessada até que o daemon receba uma resposta 200 (OK) do aplicativo no nível do ambiente de operador ou <code>RetentionPeriod</code> expire.	299	1 para 36000
VisibilityTimeout	O tempo, em segundos, em que uma mensagem recebida da fila do Amazon SQS fica bloqueada para processamento. Após o término do tempo configurado, a mensagem ficará novamente visível na fila para qualquer outro daemon ler.	300	0 para 43200
ErrorVisibilityTimeout	O tempo, em segundos, que o Elastic Beanstalk deve aguardar para retornar uma mensagem à fila do Amazon SQS após uma falha com erro explícito ao tentar processá-la.	2 segundos	0 a 43200 segundos
RetentionPeriod	A quantidade de tempo, em segundos, em que uma mensagem é válida e é ativamente processada.	345600	60 para 1209600



Nome	Descrição	Padrão	Valores válidos
MaxRetries	O número máximo de vezes em que o Elastic Beanstalk tenta enviar a mensagem à aplicação Web que vai processá-la antes de movê-la para a dead-letter queue.	10	1 para 100

## aws:elasticbeanstalk:trafficsplitting

Configure implantações de divisão de tráfego para o ambiente.

Esse namespace se aplica quando você define a opção `DeploymentPolicy` do namespace [aws:elasticbeanstalk:command](#) como `TrafficSplitting`. Para obter mais informações sobre políticas de implantação, consulte [the section called “Opções de implantação”](#).

Namespace: **aws:elasticbeanstalk:trafficsplitting**

Nome	Descrição	Padrão	Valores válidos
NewVersionPercent	A porcentagem inicial de tráfego de entrada do cliente que o Elastic Beanstalk desloca para instâncias de ambiente executando a nova versão da aplicação que está sendo implantada.	10	1 para 100
EvaluationTime	O período, em minutos, que o Elastic Beanstalk aguarda após uma implantação íntegra inicial antes de prosseguir para deslocar todo o tráfego de entrada de cliente para a nova versão da aplicação que está sendo implantada.	5	3 para 600

## aws:elasticbeanstalk:xray

Execute o daemon do AWS X-Ray para retransmitir informações de rastreamento de sua aplicação [integrada ao X-Ray](#).

Namespace: **aws:elasticbeanstalk:xray**

Nome	Descrição	Padrão	Valores válidos
XRayEnabled	Defina como <code>true</code> para executar o daemon do X-Ray nas instâncias do ambiente.	<code>false</code>	<code>true</code> <code>false</code>

## aws:elb:healthcheck

Configure as verificações de integridade de um Classic Load Balancer.

Namespace: **aws:elb:healthcheck**

Nome	Descrição	Padrão	Valores válidos
HealthyThreshold	O número de solicitações consecutivas bem-sucedidas antes de o Elastic Load Balancing alterar o status de integridade da instância.	3	2 para 10
Interval	O intervalo em que o Elastic Load Balancing verifica a integridade das instâncias do Amazon EC2 do seu aplicativo.	10	5 para 300
Timeout	A quantidade de tempo, em segundos, que o Elastic Load Balancing aguardará por uma resposta antes de considerar a instância não responsiva.	5	2 para 60
UnhealthyThreshold	Solicitações consecutivas sem êxito antes de o Elastic Load Balancing alterar o status de integridade da instância.	5	2 para 10
(obsoleto) Target	O destino em uma instância de back-end para a qual as verificações de integridade são enviadas. Em vez dessa	TCP:80	Destino no

Nome	Descrição	Padrão	Valores válidos
	opção, use Application Healthcheck URL no namespace <a href="#">aws:elasticbeanstalk:application</a> .		formato <i>PROTOCOLO:PORTA/CAMINHO</i>


## aws:elb:loadbalancer

Configure o Classic Load Balancer do ambiente.

Várias das opções neste namespace foram substituídas por opções específicas do listener no namespace [aws:elb:listener](#). Como essas opções que não são mais suportadas, você só pode configurar dois listeners (um seguro e um inseguro) em portas padrão.

Namespace: **aws:elb:loadbalancer**

Name (Nome)	Descrição	Padrão	Valores válidos
CrossZone	Configure o balanceador de carga para rotear o tráfego uniformemente em todas as instâncias em todas as zonas de disponibilidade, em vez de apenas dentro de cada região.	false	true false

 **Note**

Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um [arquivo de configuração](#). O console e a CLI do EB substituem essa opção por um [valor recomendado](#).

Name (Nome)	Descrição	Padrão	Valores válidos
SecurityGroups	Atribua um ou mais grupos de segurança que você criou ao balanceador de carga.	Nenhum	Um ou mais IDs de security groups
ManagedSecurityGroup	<p>Atribua um security group existente ao balanceador de carga do seu ambiente, em vez de criar um novo. Para usar essa configuração, atualize a configuração <code>SecurityGroups</code> neste namespace para incluir a ID do security group e remova a ID do grupo de segurança criada automaticamente, se uma foi criada.</p> <p>Para permitir o tráfego do balanceador de carga para as instâncias do EC2 do seu ambiente, o Elastic Beanstalk adiciona uma regra ao security group das instâncias que permite o tráfego de entrada do security group gerenciado.</p>	Nenhum	Um ID de security group.
(obsoleto) LoadBalancerHTTPPort	Porta de escuta do listener não seguro.	80	OFF 80
(obsoleto) LoadBalancerPortProtocol	O protocolo para usar no listener não seguro.	HTTP	HTTP TCP
(obsoleto) LoadBalancerHTTPSPort	A porta de escuta do listener seguro.	OFF	OFF 443 8443

Name (Nome)	Descrição	Padrão	Valores válidos
(obsoleto) LoadBalancerSSLPortProtocol	O protocolo para usar no listener seguro.	HTTPS	HTTPS SSL
(obsoleto) SSLCertificateId	O Amazon Resource Name (ARN) de um certificado SSL para vincular ao listener seguro.	Nenhum	

## aws:elb:listener

Configure o listener padrão (porta 80) em um Classic Load Balancer.

Namespace: **aws:elb:listener**

Name (Nome)	Descrição	Padrão	Valores válidos
ListenerProtocol	O protocolo usado pelo listener.	HTTP	HTTP TCP
InstancePort	A porta que esse listener usa para se comunicar com as instâncias do EC2.	80	1 para 65535
InstanceProtocol	<p>O protocolo que esse listener usa para se comunicar com as instâncias do EC2.</p> <p>Deve estar na mesma camada de protocolo de internet que <code>ListenerProtocol</code>. Também deve ter o mesmo nível de segurança que qualquer outro listener usando o mesmo <code>InstancePort</code> que esse listener.</p> <p>Por exemplo, se <code>ListenerProtocol</code> for HTTPS (camada de aplicação, usando uma conexão segura), você poderá definir <code>InstanceProtocol</code> como HTTP</p>	<p>HTTP quando <code>ListenerProtocol</code> é HTTP</p> <p>TCP quando <code>ListenerProtocol</code> é TCP</p>	<p>HTTP ou HTTPS quando <code>ListenerProtocol</code> é HTTP ou HTTPS</p> <p>TCP ou SSL quando <code>ListenerProtocol</code></p>

Name (Nome)	Descrição	Padrão	Valores válidos
	(também na camada de aplicação, usando uma conexão não segura). Se, além disso, você definir <code>InstancePort</code> como <code>80</code> , precisará definir <code>InstanceProtocol</code> como <code>HTTP</code> em todos os outros listeners, com <code>InstancePort</code> definido como <code>80</code> .		é <code>TCP</code> ou <code>SSL</code>
<code>PolicyNames</code>	Uma lista separada por vírgulas de nomes de políticas para aplicar à porta desse listener. Sugerimos que você use a opção <code>LoadBalancerPorts</code> do namespace <a href="#">aws:elb:policies</a> .	Nenhum	
<code>ListenerEnabled</code>	Especifica se esse listener está ativado. Se você especificar <code>false</code> , o listener não será incluído no balanceador de carga.	<code>true</code>	<code>true</code> <code>false</code>

## aws:elb:listener:listener\_port

Configure listeners adicionais em um Classic Load Balancer.

Namespace: **aws:elb:listener:*listener\_port***

Name (Nome)	Descrição	Padrão	Valores válidos
<code>ListenerProtocol</code>	O protocolo usado pelo listener.	<code>HTTP</code>	<code>HTTP</code> <code>HTTPS</code> <code>TCP</code> <code>SSL</code>
<code>InstancePort</code>	A porta que esse listener usa para se comunicar com as instâncias do EC2.	Igual a <i>listener_port</i> .	1 para 65535
<code>InstanceProtocol</code>	O protocolo que esse listener usa para se comunicar com as instâncias do EC2.	<code>HTTP</code> quando	<code>HTTP</code> ou <code>HTTPS</code>

Name (Nome)	Descrição	Padrão	Valores válidos
	<p>Deve estar na mesma camada de protocolo de internet que <code>ListenerProtocol</code>. Também deve ter o mesmo nível de segurança que qualquer outro listener usando o mesmo <code>InstancePort</code> que esse listener.</p> <p>Por exemplo, se <code>ListenerProtocol</code> for HTTPS (camada de aplicação, usando uma conexão segura), você poderá definir <code>InstanceProtocol</code> como HTTP (também na camada de aplicação, usando uma conexão não segura). Se, além disso, você definir <code>InstancePort</code> como 80, precisará definir <code>InstanceProtocol</code> como HTTP em todos os outros listeners, com <code>InstancePort</code> definido como 80.</p>	<p><code>ListenerProtocol</code> é HTTP ou HTTPS</p> <p>TCP</p> <p>quando <code>ListenerProtocol</code> é TCP ou SSL</p>	<p>quando <code>ListenerProtocol</code> é HTTP ou HTTPS</p> <p>TCP ou SSL</p> <p>quando <code>ListenerProtocol</code> é TCP ou SSL</p>
PolicyNames	<p>Uma lista separada por vírgulas de nomes de políticas para aplicar à porta desse listener. Sugerimos que você use a opção <code>LoadBalancerPorts</code> do namespace <a href="https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features-on-windows.html#aws-elb-policies">aws:elb:policies</a>.</p>	Nenhum	
SSLCertificateId	<p>O nome de recurso da Amazon (ARN) de um certificado SSL a ser vinculado ao listener.</p>	Nenhum	


Name (Nome)	Descrição	Padrão	Valores válidos
ListenerEnabled	Especifica se esse listener está ativado. Se você especificar <code>false</code> , o listener não será incluído no balanceador de carga.	<code>true</code> se qualquer outra opção estiver definida. <code>false</code> , se for o contrário.	<code>true</code> <code>false</code>

## aws:elb:policies

Modifique o padrão global de durabilidade e as políticas globais de um Classic Load Balancer.


Namespace: **aws:elb:policies**

Nome	Descrição	Padrão	Valores válidos
ConnectionDrainingEnabled	Especifica se o balanceador de carga mantém as conexões existentes com a instâncias que deixaram de ser íntegras ou tiveram o registro cancelado para concluir as solicitações em andamento.	<code>false</code>	<code>true</code> <code>false</code>

 **Note**

Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um [arquivo de configuração](#). O console e a CLI do EB substitue



Nome	Descrição	Padrão	Valores válidos
	<p>m essa opção por um <a href="#">valor recomendado</a>.</p>		
ConnectionDrainingTimeout	<p>O número máximo de segundos que o balanceador de carga mantém conexões existentes com uma instância durante a drenagem da conexão antes de forçar o fechamento das conexões.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um <a href="#">arquivo de configuração</a>. O console e substitui essa opção por um <a href="#">valor recomendado</a>.</p> </div>	20	1 para 3600
ConnectionSettingIdleTimeout	<p>A quantidade de tempo, em segundos, que o balanceador de carga aguarda que todos os dados sejam enviados ou recebidos pela conexão. Se nenhum dado tiver sido enviado ou recebido após esse período, o balanceador de carga fechará a conexão.</p>	60	1 para 3600
LoadBalancerPorts	<p>Uma lista separada por vírgulas das portas do listener em que a política padrão (AWSEB-ELB-StickinessPolicy ) se aplica.</p>	Nenhum	Você pode usar :all para indicar todas as portas do listener

Nome	Descrição	Padrão	Valores válidos
Stickiness Cookie Expiration	O tempo, em segundos, pelo qual cada cookie é válido. Usa a política padrão (AWSEB-ELB-StickinessPolicy ).	0	0 para 1000000
Stickiness Policy	Vincula a sessão do usuário a uma determinada instância do servidor, para que todas as solicitações desse usuário durante a sessão sejam enviadas para a mesma instância do servidor. Usa a política padrão (AWSEB-ELB-StickinessPolicy ).	false	true false

## aws:elb:policies:policy\_name

Crie políticas adicionais de balanceador de carga para um Classic Load Balancer.

Namespace: **aws:elb:policies:policy\_name**

Nome	Descrição	Padrão	Valores válidos
CookieName	O nome do cookie gerado pelo aplicativo que controla os tempos de vida da sessão de uma política AppCookieStickinessPolicyType . Essa política pode ser associada somente a listeners HTTP/HTTPS.	Nenhum	
InstancePorts	Uma lista separada por vírgulas das portas de instâncias em que a política aplica.	Nenhum	Uma lista de portas, ou :all
LoadBalancerPorts	Uma lista separada por vírgulas das portas de listeners em que a política aplica.	Nenhum	Uma lista de portas, ou :all

Nome	Descrição	Padrão	Valores válidos
ProxyProtocol	Para uma política <code>ProxyProtocolPolicyType</code> , especifique se deseja ou não incluir o endereço IP e a porta da solicitação de origem de mensagens TCP. Essa política pode ser associada somente a listeners TCP/SSL.	Nenhum	true false
PublicKey	O conteúdo de uma chave pública para uma política <code>PublicKeyPolicyType</code> a ser utilizada na autenticação do servidor ou servidores de backend. Esta política não pode ser aplicada diretamente a servidores de backend ou listeners. Ela deve fazer parte de uma política de <code>BackendServerAuthenticationPolicyType</code> .	Nenhum	
PublicKeyPolicyNames	Uma lista separada por vírgulas de nomes de políticas (com base em políticas <code>PublicKeyPolicyType</code> ) para uma política <code>BackendServerAuthenticationPolicyType</code> que controla a autenticação em um ou mais servidores de backend. Essa política só pode ser associada a servidores de backend que estejam usando HTTPS/SSL.	Nenhum	

Nome	Descrição	Padrão	Valores válidos
SSLProtocols	Uma lista separada por vírgulas de protocolos SSL a serem habilitados para uma política de <code>SSLNegotiationPolicyType</code> que define as cifras e os protocolos que são aceitos pelo balanceador de carga. Essa política pode ser associada somente a listeners HTTPS/SSL.	Nenhum	
SSLReferencePolicy	O nome de uma política de segurança predefinida que segue as práticas recomendadas de segurança da AWS e que você deseja ativar para uma política <code>SSLNegotiationPolicyType</code> que define as cifras e os protocolos que são aceitos pelo balanceador de carga. Essa política pode ser associada somente a listeners HTTPS/SSL.	Nenhum	
Stickiness Cookie Expiration	O tempo, em segundos, pelo qual cada cookie é válido.	0	0 para 1000000
Stickiness Policy	Vincula a sessão do usuário a uma determinada instância do servidor, para que todas as solicitações desse usuário durante a sessão sejam enviadas para a mesma instância do servidor.	false	true false

## aws:elbv2:listener:default

Configure o listener padrão (porta 80) em um Application Load Balancer ou um Network Load Balancer.

Esse namespace não se aplica a um ambiente que usa um balanceador de carga compartilhado. Os balanceadores de carga compartilhados não têm um listener padrão.

Namespace: **aws:elbv2:listener:default**

Nome	Descrição	Padrão	Valores válidos
DefaultProcess	O nome do <a href="#">processo</a> ao qual encaminhar o tráfego quando nenhuma regra se aplica.	default	Um nome de processo.
ListenerEnabled	Defina como <code>false</code> para desativar o listener. Você pode usar essa opção para desativar o listener padrão na porta 80.	true	true false
Protocol	Protocolo de tráfego a processar.	Com Application Load Balancer: HTTP  Com Network Load Balancer: TCP	Com Application Load Balancer: HTTP, HTTPS  Com Network Load Balancer: TCP
Rules	A lista de <a href="#">regras</a> que se aplicam ao listener  Essa opção só é aplicável a ambientes com um Application Load Balancer.	Nenhum	Uma lista separada por vírgulas de nomes de regras.
SSLCertificateArns	O nome de recurso da Amazon (ARN) do certificado SSL a ser vinculado ao listener.	Nenhum	O ARN de um certificado armazenado no IAM ou no ACM.

Nome	Descrição	Padrão	Valores válidos
	Essa opção só é aplicável a ambientes com um Application Load Balancer.		
SSLPolicy	<p>Especifique uma política de segurança para aplicar ao listener.</p> <p>Essa opção só é aplicável a ambientes com um Application Load Balancer.</p>	Nenhum (padrão do ELB)	O nome de uma política de segurança do balanceador de carga.

## aws:elbv2:listener:listener\_port

Configure listeners adicionais em um Application Load Balancer ou um Network Load Balancer.

### Note

Para um Application Load Balancer compartilhado, você pode especificar apenas a opção `Rule`. As outras opções não são aplicáveis aos balanceadores de carga compartilhados.

Namespace: **aws:elbv2:listener:*listener\_port***

Nome	Descrição	Padrão	Valores válidos
DefaultProcess	O nome do <a href="#">processo</a> ao qual encaminhar o tráfego quando não houver regras correspondentes.	default	Um nome de processo.
ListenerEnabled	Defina como <code>false</code> para desativar o	true	true

Nome	Descrição	Padrão	Valores válidos
	listener. Você pode usar essa opção para desativar o listener padrão na porta 80.		false
Protocol	Protocolo de tráfego a processar.	Com Application Load Balancer: HTTP  Com Network Load Balancer: TCP	Com Application Load Balancer: HTTP, HTTPS  Com Network Load Balancer: TCP
Rules	Lista de <a href="#">regras</a> que se aplicam ao listener  Essa opção só é aplicável a ambientes com um Application Load Balancer.  Se o ambiente usar um Application Load Balancer compartilhado e você não especificar essa opção para nenhum listener, o Elastic Beanstalk associará automaticamente a regra default a um listener de porta 80.	Nenhum	Uma lista separada por vírgulas de nomes de regras.

Nome	Descrição	Padrão	Valores válidos
SSLCertificateArns	<p>O nome de recurso da Amazon (ARN) do certificado SSL a ser vinculado ao listener.</p> <p>Essa opção só é aplicável a ambientes com um Application Load Balancer.</p>	Nenhum	O ARN de um certificado armazenado no IAM ou no ACM.
SSLPolicy	<p>Especifique uma política de segurança para aplicar ao listener.</p> <p>Essa opção só é aplicável a ambientes com um Application Load Balancer.</p>	Nenhum (padrão do ELB)	O nome de uma política de segurança do balanceador de carga.

### aws:elbv2:listenerrule:rule\_name

Defina regras de listener para um Application Load Balancer. Se uma solicitação corresponder aos nomes ou caminhos do host em uma regra, o balanceador de carga a encaminhará para o processo especificado. Para usar uma regra, adicione-a a um listener com a opção `Rules` no namespace [aws:elbv2:listener:listener\\_port](#).

#### Note

Este namespace não é aplicável a ambientes com um Network Load Balancer.



Namespace: **aws:elbv2:listenerrule:rule\_name**

Nome	Descrição	Padrão	Valores válidos
HostHeaders	Uma lista de nomes de host para corresponder. Por exemplo, <code>my.example.com</code> .	Load balancer dedicado: nenhum  Load balancer compartilhado: o CNAME do ambiente	<p>Cada nome pode conter até 128 caracteres. Um padrão pode incluir letras maiúsculas e minúsculas, números, hífen (-) e até três caracteres curinga (* corresponde a zero ou mais caracteres; ? corresponde a exatamente um caractere). Você pode listar mais de um nome, cada um separado por uma vírgula. O Application Load Balancer é compatível com até cinco regras HostHeader e PathPattern combinadas.</p> <p>Para obter mais informações, consulte <a href="#">Condições de host</a> no Guia do Usuário para o Application Load Balancers.</p>
PathPatterns	Os padrões de caminho a serem adotados (por exemplo, <code>/img/*</code> ).	Nenhum	Cada padrão pode conter até 128 caracteres. Um

Nome	Descrição	Padrão	Valores válidos
	Essa opção só é aplicável a ambientes com um Application Load Balancer.		<p>padrão pode incluir letras maiúsculas e minúsculas, números, hífen (-) e até três caracteres curinga (* corresponde a zero ou mais caracteres; ? corresponde a exatamente um caractere). É possível adicionar vários padrões de caminho separados por vírgulas. O Application Load Balancer é compatível com até cinco regras HostHeader e PathPattern combinadas.</p> <p>Para obter mais informações, consulte as <a href="#">Condições de caminho</a> no Guia do Usuário para Application Load Balancers.</p>

Nome	Descrição	Padrão	Valores válidos
Priority	<p>A precedência dessa regra quando há várias regras correspondentes. O menor número tem precedência. Duas regras não podem ter a mesma prioridade.</p> <p>Com um balanceador de carga compartilhado, o Elastic Beanstalk trata as prioridades de regras como relativas em ambientes de compartilhamento e as mapeia para prioridades absolutas durante a criação.</p>	1	1 para 1000
Process	O nome do <a href="#">processo</a> ao qual encaminhar o tráfego quando essa regra corresponde à solicitação.	default	Um nome de processo.

## aws:elbv2:loadbalancer

Configure um Application Load Balancer.

Para um balanceador de carga compartilhado, somente as opções `SharedLoadBalancer` e `SecurityGroups` são válidas.

### Note

Este namespace não é aplicável a ambientes com um Network Load Balancer.

## Namespace: **aws:elbv2:loadbalancer**

Nome	Descrição	Padrão	Valores válidos
AccessLogsS3Bucket	O bucket do Amazon S3 no qual os logs de acesso são armazenados. O bucket deve estar na mesma região que o ambiente e permitir acesso de gravação ao balanceador de carga.	Nenhum	Nome de um bucket.

Nome	Descrição	Padrão	Valores válidos
AccessLogsS3Enabled	Habilite o acesso ao armazenamento de logs.	false	true false
AccessLogsS3Prefix	O prefixo para prefixar nomes de logs de acesso. Por padrão, o balanceador de carga faz upload de logs para um diretório chamado AWSLogs no bucket que você especificar. Especifique um prefixo para colocar o diretório AWSLogs dentro de outro diretório.	Nenhum	
IdleTimeout	A quantidade de tempo, em segundos, para aguardar a conclusão de uma solicitação antes de fechar conexões com o cliente e a instância.	Nenhum	1 para 3600

Nome	Descrição	Padrão	Valores válidos
ManagedSecurityGroup	<p>Atribua um grupo de segurança existente ao balanceador de carga do ambiente em vez de criar um novo. Para usar essa configuração, atualize a definição <code>SecurityGroups</code> neste namespace para incluir o ID do security group e remover o ID do security group criado automaticamente, se houver um.</p> <p>Para permitir o tráfego do balanceador de carga para as instâncias do EC2 para o seu ambiente, o Elastic Beanstalk adiciona uma regra ao security group de suas instâncias que permite o tráfego de entrada do security group gerenciado.</p>	O security group que o Elastic Beanstalk cria para o balanceador de carga.	Um ID de security group.

Nome	Descrição	Padrão	Valores válidos
SecurityGroups	<p>Lista de security groups para anexar ao balanceador de carga.</p> <p>Para um balanceador de carga compartilhado, se você não especificar esse valor, o Elastic Beanstalk verificará se um grupo de segurança existente que ele gerencia já está conectado ao balanceador de carga. Se um não estiver conectado ao balanceador de carga, o Elastic Beanstalk cria um security group e o anexará ao balanceador de carga. O Elastic Beanstalk exclui esse grupo de segurança quando o último ambiente que compartilha o balanceador de carga é encerrado.</p> <p>Os grupos de segurança do balanceador de carga são usados para configurar a regra de entrada do grupo de segurança da instância do Amazon EC2.</p>	O grupo de segurança que o Elastic Beanstalk cria para o balanceador de carga.	Lista separada por vírgulas de IDs do grupo de segurança.

Nome	Descrição	Padrão	Valores válidos
SharedLoadBalancer	<p>O nome de recurso da Amazon (ARN) do balanceador de carga. Essa opção é relevante apenas para um Application Load Balancer. Ela é necessária quando a opção <code>LoadBalancerIsShared</code> do namespace <a href="#">aws:elasticbeanstalk:environment</a> está definida como <code>true</code>. Não é possível alterar o ARN do balanceador de carga compartilhado após a criação do ambiente.</p> <p>Critérios para um valor válido:</p> <ul style="list-style-type: none"><li>• Ele deve ser um balanceador de carga ativo e válido na região da AWS onde o ambiente está localizado.</li><li>• Ele deve estar na mesma Amazon Virtual Private Cloud (Amazon VPC) que o ambiente.</li><li>• Ele não pode ser um balanceador de carga criado pelo Elastic Beanstalk como balanceador de carga dedicado para outro ambiente. Você pode identificar esses balanceadores de carga dedicados usando o prefixo <code>awseb-</code>.</li></ul> <p>Exemplo:</p> <pre>arn:aws:elasticloadbalancing:us-east-2:123456789012:lo</pre>	Nenhum	ARN de um balanceador de carga válido que atende a todos os critérios descritos aqui.

Nome	Descrição	Padrão	Valores válidos
	adbouncer/app/FrontEndLB/0dbf78d8ad96abbc		


## aws:rds:dbinstance

Configure uma instância de banco de dados anexada do Amazon RDS.

Namespace: **aws:rds:dbinstance**


Nome	Descrição	Padrão	Valores válidos
DBAllocatedStorage	O tamanho de armazenamento do banco de dados alocado, especificado em gigabytes.	MySQL: 5 Oracle: 10 sqlserver-se: 200 sqlserver-ex: 30 sqlserver-web: 30	MySQL: 5-1024 Oracle: 10-1024 sqlserver: não pode ser modificado
DBDeletionPolicy	Especifica se é necessário reter, excluir ou fazer um snapshot da instância de banco de dados quando um ambiente é terminado.  Essa opção funciona em conjunto com <code>HasCoupledDatabase</code> , também uma opção desse namespace.	Delete	Delete Retain Snapshot



Nome	Descrição	Padrão	Valores válidos
	<div style="border: 1px solid #f08080; padding: 10px; background-color: #fff9e6;"> <p> <b>Warning</b></p> <p>A exclusão de uma instância de banco de dados resulta em perda de dados permanente.</p> </div>		
DBEngine	O nome do mecanismo de banco de dados a utilizar nessa instância.	mysql	mysql oracle-se1 sqlserver-ex sqlserver-web sqlserver-se postgres
DBEngineVersion	O número da versão do mecanismo de banco de dados.	5.5	
DBInstanceClass	O tipo de instância de banco de dados.	db.t2.micro  (db.m1.large para um ambiente que não está em execução em uma Amazon VPC)	Para obter mais informações, consulte <a href="#">Classe de instância do banco de dados</a> no Guia do usuário do Amazon RDS.

Nome	Descrição	Padrão	Valores válidos
DBPassword	O nome da senha mestra do usuário para a instância de banco de dados.	Nenhum	
DBSnapshotIdentifier	O identificador para restaurar o DB snapshot.	Nenhum	
DBUser	O nome do usuário mestre da instância do banco de dados.	ebroot	

Nome	Descrição	Padrão	Valores válidos
HasCoupledDatabase	<p>Especifica se uma instância de banco de dados está acoplada ao ambiente. Se definida como <code>true</code>, o Elastic Beanstalk cria uma nova instância de banco de dados acoplada ao seu ambiente. Se definida como <code>false</code>, o Elastic Beanstalk inicia o desacoplamento da instância de banco de dados do seu ambiente.</p> <p>Essa opção funciona em conjunto com <code>DBDeletionPolicy</code>, também uma opção desse namespace.</p>	<code>false</code>	<code>true</code> <code>false</code>

 **Note**

Nota: se você voltar esse valor para `true` após desacoplar o banco de dados anterior, o Elastic Beanstalk cria um novo banco de dados com as configurações de banco de dados anteriores. No entanto, para manter a segurança do seu ambiente, ele não retém as configurações existentes de `DBUser` e `DBPassword`. Você precisa especificar `DBUser` e `DBPassword` novamente.

Nome	Descrição	Padrão	Valores válidos
MultiAZDatabase	Especifica se a implantação Multi-AZ de uma instância de banco de dados precisa ser criada. Para obter mais informações sobre implantações Multi-AZ com o Amazon Relational Database Service (RDS), consulte as <a href="#">Regiões e zonas de disponibilidade</a> no Guia do usuário do Amazon Relational Database Service.	false	true false

## Opções específicas para a plataforma

Algumas plataformas do Elastic Beanstalk definem namespaces de opção que são específicos da plataforma. Esses namespaces e suas opções estão listados abaixo para cada plataforma.

### Note

Anteriormente, em versões de plataforma baseadas na AMI do Amazon Linux (anterior ao Amazon Linux 2), os dois recursos a seguir e seus respectivos namespaces eram considerados recursos específicos da plataforma e foram listados aqui por plataforma:

- Configuração de proxy para arquivos estáticos – [aws:elasticbeanstalk:environment:proxy:staticfiles](#)
- AWS X-Ray Suporte do – [aws:elasticbeanstalk:xray](#)

Nas versões da plataforma do Amazon Linux 2, o Elastic Beanstalk implementa esses recursos de forma consistente em todas as plataformas de suporte. O namespace relacionado agora está listado na página [the section called “Opções gerais”](#). Mantivemos somente a menção a eles nesta página para plataformas que tinham namespaces de nomes diferentes.

## Plataformas

- [Opções para a plataforma Docker](#)
- [Opções para a plataforma Go](#)

- [Opções para a plataforma Java SE](#)
- [Java com opções para a plataforma Tomcat](#)
- [Opções da plataforma .NET Core no Linux](#)
- [Opções para a plataforma .NET](#)
- [Opções para a plataforma Node.js](#)
- [Opções para a plataforma PHP](#)
- [Opções para a plataforma Python](#)
- [Opções para a plataforma Ruby](#)

## Opções para a plataforma Docker

As opções de configuração específicas do Docker a seguir são aplicáveis a plataformas do Docker e do Docker pré-configurado.

### Note

Essas opções de configuração não se aplicam a:

- a plataforma Docker (Amazon Linux 2) com Docker Compose
- a plataforma Docker de vários contêineres (AMI do Amazon Linux)

Namespace: **aws:elasticbeanstalk:environment:proxy**

Name (Nome)	Descrição	Padrão	Valores válidos
ProxyServer	Especifica o servidor web a ser usado como um proxy.	nginx	nginx  none: AMI do Amazon Linux e Docker com DC somente

## Opções para a plataforma Go

Opções de plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2)

Namespace: **aws:elasticbeanstalk:container:golang:staticfiles**

Você pode usar o namespace a seguir para configurar o servidor de proxy para atender arquivos estáticos. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo. Isso reduz o número de solicitações que seu aplicativo deve processar.

Mapeie um caminho fornecido pelo servidor de proxy para uma pasta no código-fonte que contenha ativos estáticos. Cada opção definida neste namespace mapeia um caminho diferente.

Name (Nome)	Value (Valor)
Caminho em que o servidor de proxy servirá os arquivos.  Exemplo: /images para servir arquivos em <i>subdomain</i> .elasticbeanstalk.com/images .	Nome da pasta que contém os arquivos.  Exemplo: <code>staticimages</code> para servir arquivos de uma pasta denominada <code>staticimages</code> no nível mais alto do seu pacote de origem.

## Opções para a plataforma Java SE

Opções de plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2)

Namespace: **aws:elasticbeanstalk:container:java:staticfiles**

Você pode usar o namespace a seguir para configurar o servidor de proxy para atender arquivos estáticos. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo. Isso reduz o número de solicitações que seu aplicativo deve processar.

Mapeie um caminho fornecido pelo servidor de proxy para uma pasta no código-fonte que contenha ativos estáticos. Cada opção definida neste namespace mapeia um caminho diferente.

Name (Nome)	Value (Valor)
Caminho em que o servidor de proxy servirá os arquivos.  Exemplo: /images para servir arquivos em <i>subdomain</i> .elasticbeanstalk.com/images .	Nome da pasta que contém os arquivos.  Exemplo: staticimages para servir arquivos de uma pasta denominada staticimages no nível mais alto do seu pacote de origem.

## Java com opções para a plataforma Tomcat


Namespace: **aws:elasticbeanstalk:application:environment**

Name (Nome)	Descrição	Padrão	Valores válidos
JDBC_CONNECTION_STRING	A string de conexão a um banco de dados externo.	n/a	n/a

Consulte [Propriedades de ambientes e outras configurações de software](#) para obter mais informações.

Namespace: **aws:elasticbeanstalk:container:tomcat:jvmoptions**

Name (Nome)	Descrição	Padrão	Valores válidos
JVM Options	Passe as opções da linha de comando para a JVM no startup.	n/a	n/a
Xmx	Tamanhos máximos de heaps da JVM.	256m	n/a
XX:MaxPermSize	Seção do heap da JVM usada para armazenar definições de classe e metadados associados.	64m	n/a

Name (Nome)	Descrição	Padrão	Valores válidos
	<p> <b>Note</b></p> <p>Esta opção só se aplica a versões Java anteriores ao Java 8 e não é compatível com plataformas Tomcat do Elastic Beanstalk baseadas no Amazon Linux 2 ou posterior.</p>		
Xms	Tamanhos iniciais de heap da JVM.	256m	n/a
<i>optionName</i>	Especifique as opções arbitrárias de JVM além daqueles definidos pela plataforma Tomcat.	n/a	n/a

#### Namespace: **aws:elasticbeanstalk:environment:proxy**

Name (Nome)	Descrição	Padrão	Valores válidos
GzipCompression	<p>Defina como <code>false</code> para desativar a compactação de resposta.</p> <p>Válido somente em versões da plataforma da AMI do Amazon Linux (anterior ao Amazon Linux 2).</p>	true	true false
ProxyServer	<p>Defina o proxy a ser usado nas instâncias do ambiente. Se você definir essa opção como <code>apache</code>, o Elastic Beanstalk usará o <a href="#">Apache 2.4</a>.</p> <p>Defina como <code>apache/2.2</code> se seu aplicativo não estiver pronto para migrar do <a href="#">Apache 2.2</a> devido a definições de configuração de proxy incompatíveis. Esse valor só é</p>	<p>nginx (Amazon Linux 2)</p> <p>apache (AMI do Amazon Linux)</p>	<p>apache</p> <p>apache/2.2 – somente AMI do Amazon Linux</p> <p>nginx</p>



Name (Nome)	Descrição	Padrão	Valores válidos
	<p>válido nas versões da plataforma da AMI do Amazon Linux (anterior ao Amazon Linux 2).</p> <p>Defina como <code>nginx</code> para usar o <a href="#">nginx</a>. Este é o padrão a partir das versões da plataforma do Amazon Linux 2.</p> <p>Para obter mais informações, consulte <a href="#">Configurar servidor de proxy do seu ambiente Tomcat</a>.</p>		

## Opções da plataforma .NET Core no Linux

Namespace: **aws:elasticbeanstalk:environment:proxy**

Name (Nome)	Descrição	Padrão	Valores válidos
ProxyServer	Especifica o servidor web a ser usado como um proxy.	nginx	nginx none

## Opções para a plataforma .NET

Namespace: **aws:elasticbeanstalk:container:dotnet:apppool**

Name (Nome)	Descrição	Padrão	Valores válidos
Target Runtime	Escolha a versão do .NET Framework para o seu aplicativo.	4.0	2.0 4.0
Enable 32-bit Applications	Defina como <code>True</code> para executar aplicativos de 32 bits.	False	True False

## Opções para a plataforma Node.js

Namespace: **aws:elasticbeanstalk:environment:proxy**

Name (Nome)	Descrição	Padrão	Valores válidos
ProxyServer	Defina o proxy a ser usado nas instâncias do ambiente.	nginx	apache nginx

Opções de plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2)

Namespace: **aws:elasticbeanstalk:container:nodejs**

Name (Nome)	Descrição	Padrão	Valores válidos
NodeCommand	Comando usado para iniciar o aplicativo Node.js. Se uma string vazia for especificada, <code>app.js</code> será usado, seguido por <code>server.js</code> e <code>npm start</code> , nesta ordem.	""	n/a
NodeVersion	<p>Versão do Node.js. Por exemplo, 4.4.6</p> <p>As versões do Node.js compatíveis variam entre as versões da plataforma do Node.js. Consulte <a href="#">Node.js</a> no documento Plataformas do AWS Elastic Beanstalk para obter uma lista das versões compatíveis no momento.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p><b>Note</b></p> <p>Quando o suporte oferecido para a versão do Node.js que você estiver usando for removido da plataforma, será necessário alterar ou remover a configuração da versão antes de</p> </div>	varia	varia

Name (Nome)	Descrição	Padrão	Valores válidos
	<p>fazer a <a href="#">atualização da plataforma</a>. Isso pode ocorrer quando uma vulnerabilidade de segurança é identificada para uma ou mais versões do Node.js.</p> <p>Quando isso ocorrer, tentar atualizar para uma nova versão da plataforma não compatível com a <a href="#">NodeVersion</a> configurada falhará. Para evitar a necessidade de criar um novo ambiente, altere a opção de configuração NodeVersion para uma versão de Node.js que tenha suporte na versão de plataforma antiga e na nova, ou <a href="#">remova a configuração de opção</a> e execute a atualização da plataforma.</p>		
GzipCompression	Especifica se a compactação gzip está habilitada. Se o ProxyServer estiver definido como none, a compactação gzip será desativada.	false	true false
ProxyServer	Especifica qual servidor da web deve ser usado em conexões via proxy ao Node.js. Se o ProxyServer estiver definido como none, os mapeamentos de arquivos estáticos não terão efeito e a compactação gzip será desabilitada.	nginx	apache nginx none

## Namespace: `aws:elasticbeanstalk:container:nodejs:staticfiles`

Você pode usar o namespace a seguir para configurar o servidor de proxy para atender arquivos estáticos. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo. Isso reduz o número de solicitações que seu aplicativo deve processar.

Mapeie um caminho fornecido pelo servidor de proxy para uma pasta no código-fonte que contenha ativos estáticos. Cada opção definida neste namespace mapeia um caminho diferente.

### Note

As configurações de arquivos estáticos não são aplicáveis se `aws:elasticbeanstalk:container:nodejs::ProxyFiles` estiver definido como `none`.

Name (Nome)	Value (Valor)
<p>Caminho em que o servidor de proxy servirá os arquivos.</p> <p>Exemplo: <code>/images</code> para servir arquivos em <code>subdomain .elasticbeanstalk.com/images</code>.</p>	<p>Nome da pasta que contém os arquivos.</p> <p>Exemplo: <code>staticimages</code> para servir arquivos de uma pasta denominada <code>staticimages</code> no nível mais alto do seu pacote de origem.</p>

## Opções para a plataforma PHP


### Namespace: `aws:elasticbeanstalk:container:php:phpini`

Name (Nome)	Descrição	Padrão	Valores válidos
<code>document_root</code>	Especifique o diretório filho do seu projeto que é tratado como raiz da web voltada para o público.	<code>/</code>	Uma string em branco é tratada como <code>/</code> ou especifica uma string começando com <code>/</code>

Name (Nome)	Descrição	Padrão	Valores válidos
memory_limit	Quantidade de memória alocada para o ambiente PHP.	256M	n/a
zlib.output_compression	Especifica se o PHP deve usar compactação para saída.	Off	On Off true false
allow_url_fopen	Especifica se as funções de arquivos do PHP têm permissão para recuperar dados de locais remotos, como sites ou servidores de FTP.	On	On Off true false
display_errors	Especifica se as mensagens de erro devem ser parte da saída.	Off	On Off
max_execution_time	Define o tempo máximo em segundos que um script tem permissão para ser executado antes que o ambiente o encerre.	60	0 a 9223372036854775807 (PHP_INT_MAX)
composer_options	Define opções personalizadas a serem usadas ao instalar dependências usando o Composer por meio de composer.phar install. Para obter mais informações, incluindo as opções disponíveis, acesse <a href="http://getcomposer.org/doc/03-cli.md#install">http://getcomposer.org/doc/03-cli.md#install</a> .	n/a	n/a

Namespace: **aws:elasticbeanstalk:environment:proxy**

Name (Nome)	Descrição	Padrão	Valores válidos
ProxyServer	Defina o proxy a ser usado nas instâncias do ambiente.	nginx	apache nginx

 Note

Para obter mais informações sobre a plataforma PHP, consulte [Usar a plataforma PHP do Elastic Beanstalk](#).

## Opções para a plataforma Python

Namespace: **aws:elasticbeanstalk:application:environment**

Name (Nome)	Descrição	Padrão	Valores válidos
DJANGO_SETTINGS_MODULE	Especifica qual arquivo de configurações deve ser usado.	n/a	n/a

Consulte [Propriedades de ambientes e outras configurações de software](#) para obter mais informações.

Namespace: **aws:elasticbeanstalk:container:python**

Name (Nome)	Descrição	Padrão	Valores válidos
WSGIPath	O arquivo que contém o aplicativo WSGI. Este arquivo deve ter um <code>application</code> chamável.	Nas versões da plataforma Python do Amazon	n/a

Name (Nome)	Descrição	Padrão	Valores válidos
		Linux 2: application  Nas versões da plataforma Python da AMI do Amazon Linux: application.py	
NumProcesses	O número de processos daemon que deve ser iniciado para o grupo de processo ao executar aplicativos WSGI.	1	n/a
NumThreads	O número de threads a serem criados para lidar com as solicitações em cada processo daemon no processo de grupo ao executar aplicativos WSGI.	15	n/a

Namespace: **aws:elasticbeanstalk:environment:proxy**

Name (Nome)	Descrição	Padrão	Valores válidos
ProxyServer	Defina o proxy a ser usado nas instâncias do ambiente.	nginx	apache  nginx

## Opções de plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2)

Namespace: **aws:elasticbeanstalk:container:python:staticfiles**

Você pode usar o namespace a seguir para configurar o servidor de proxy para atender arquivos estáticos. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo. Isso reduz o número de solicitações que seu aplicativo deve processar.

Mapeie um caminho fornecido pelo servidor de proxy para uma pasta no código-fonte que contenha ativos estáticos. Cada opção definida neste namespace mapeia um caminho diferente.

Por padrão, o servidor de proxy em um ambiente Python atende todos os arquivos de uma pasta chamada `static` no caminho `/static`.

Namespace: **aws:elasticbeanstalk:container:python:staticfiles**

Name (Nome)	Value (Valor)
Caminho em que o servidor de proxy servirá os arquivos.	Nome da pasta que contém os arquivos.
Exemplo: <code>/images</code> para servir arquivos em <code>subdomain.eleasticbeanstalk.com/images</code> .	Exemplo: <code>staticimages</code> para servir arquivos de uma pasta denominada <code>staticimages</code> no nível mais alto do seu pacote de origem.

## Opções para a plataforma Ruby

Namespace: **aws:elasticbeanstalk:application:environment**

Name (Nome)	Descrição	Padrão	Valores válidos
<code>RAILS_SKIP_MIGRATIONS</code>	Especifica se <code>`rake db:migrations`</code> deve ser executado em nome dos aplicativos dos usuários ou se deve ser ignorado. Aplicável somente para aplicativos Rails 3.	<code>false</code>	<code>true</code> <code>false</code>



Name (Nome)	Descrição	Padrão	Valores válidos
RAILS_SKIP_ASSET_COMPILE	Especifica se o contêiner deve executar <code>`rake assets:precompile`</code> em nome dos aplicativos dos usuários ou se deve ser ignorado. Também aplicável somente para aplicativos Rails 3.	false	true false
BUNDLE_WITHOUT	Uma lista separada por dois pontos (:) de grupos a serem ignorados ao instalar dependências de um Gemfile.	test:development	n/a
RACK_ENV	Especifica em que estágio do ambiente um aplicativo pode ser executado. Exemplos de ambientes comuns incluem desenvolvimento, produção, teste.	production	n/a

Consulte [Propriedades de ambientes e outras configurações de software](#) para obter mais informações.

## Opções personalizadas

Use o namespace `aws:elasticbeanstalk:customoption` para definir opções e valores que possam ser lidos em blocos `Resources` de outros arquivos de configuração. Use as opções personalizadas para coletar as configurações especificadas pelo usuário em um único arquivo de configuração.

Por exemplo, você pode ter um arquivo de configuração complexa que define um recurso que pode ser configurado pelo usuário que inicia o ambiente. Se você usar `Fn::GetOptionSetting` para recuperar o valor de uma opção personalizada, poderá colocar a definição dessa opção em um arquivo de configuração diferente, para que seja detectada e modificada com mais facilidade pelo usuário.

Além disso, como são opções de configuração, as opções personalizadas podem ser definidas no nível de API para substituir valores definidos em um arquivo de configuração. Para obter mais informações, consulte [Precedência](#).

As opções personalizadas são definidas como qualquer outra opção:

```
option_settings:
 aws:elasticbeanstalk:customoption:
 option name: option value
```

Por exemplo, o seguinte arquivo de configuração cria uma opção chamada ELBAlarmEmail e define o valor como `someone@example.com`:

```
option_settings:
 aws:elasticbeanstalk:customoption:
 ELBAlarmEmail: someone@example.com
```

Em outras páginas, um arquivo de configuração define um tópico SNS que lê a opção com `Fn::GetOptionSetting` para preencher o valor do atributo `Endpoint`:

```
Resources:
 MySNSTopic:
 Type: AWS::SNS::Topic
 Properties:
 Subscription:
 - Endpoint:
 Fn::GetOptionSetting:
 OptionName: ELBAlarmEmail
 DefaultValue: nobody@example.com
 Protocol: email
```

Encontre mais snippets demonstrativos usando `Fn::GetOptionSetting` em [Adicionar e personalizar recursos do ambiente do Elastic Beanstalk](#).

## Personalização avançada de ambiente com arquivos de configuração (**.ebextensions**)

Você pode adicionar arquivos de AWS Elastic Beanstalk configuração (`.ebextensions`) ao código-fonte do seu aplicativo web para configurar seu ambiente e personalizar os AWS recursos que ele contém. [Os arquivos de configuração são documentos no formato YAML ou JSON com uma extensão de `.config` arquivo que você coloca em uma pasta chamada `.ebextensions` e implanta no pacote de origem do aplicativo.](#)

## Example .ebextensions/ .config network-load-balancer

Este exemplo facilita uma alteração de configuração. Ele modifica uma opção de configuração para definir o tipo de balanceador de carga do ambiente como Network Load Balancer.

```
option_settings:
 aws:elasticbeanstalk:environment:
 LoadBalancerType: network
```

Recomendamos usar o YAML para seus arquivos de configuração, pois é mais legível do que o JSON. YAML é compatível com comentários, comandos de várias linhas, várias alternativas para usar aspas e muito mais. No entanto, é possível fazer qualquer alteração de configuração em arquivos de configuração do Elastic Beanstalk de forma idêntica usando YAML ou JSON.

### Dica

Quando você estiver desenvolvendo ou testando novos arquivos de configuração, inicie um ambiente limpo executando o aplicativo padrão e implante nele. Arquivos de configuração formatados incorretamente causarão falha irrecoverável do início de um novo ambiente.

A seção `option_settings` de um arquivo de configuração define valores para as [opções de configuração](#). As opções de configuração permitem que você configure seu ambiente do Elastic Beanstalk AWS, os recursos nele contidos e o software que executa seu aplicativo. Os arquivos de configuração são apenas uma das várias maneiras de definir as opções de configuração.

A [Resourceção](#) permite que você personalize ainda mais os recursos no ambiente do seu aplicativo e defina AWS recursos adicionais além da funcionalidade fornecida pelas opções de configuração. Você pode adicionar e configurar qualquer recurso suportado pelo AWS CloudFormation, que o Elastic Beanstalk usa para criar ambientes.

As outras seções do arquivo de configuração (`packages`, `sources`, `files`, `users`, `groups`, `commands`, `container_commands` e `services`) permitem configurar as instâncias do EC2 que são iniciadas no ambiente. Sempre que um servidor é iniciado no ambiente, o Elastic Beanstalk executa as operações definidas nessas seções para preparar o sistema operacional e o sistema de armazenamento para a aplicação.

Para obter exemplos de `.ebextensions` usados com frequência, consulte os [Arquivos de configuração do repositório do Elastic Beanstalk](#).

## Requisitos

- **Localização** — O Elastic Beanstalk `.ebextensions` processará todas as pastas presentes na sua implantação. No entanto, recomendamos que você coloque todos os seus arquivos de configuração em uma única pasta, chamada `.ebextensions`, na raiz do pacote de origem. Pastas que começam com um ponto podem ser ocultas por navegadores de arquivos, portanto, certifique-se de que a pasta é adicionada quando você criar seu pacote de origem. Para ter mais informações, consulte [Criar um pacote de origem do aplicativo](#).
- **Nomenclatura**: os arquivos de configuração devem ter a extensão de arquivo `.config`.
- **Formatação**: os arquivos de configuração devem ser compatíveis com as especificações de YAML ou JSON.

Ao usar o YAML, sempre use espaços para recuar as chaves em diferentes níveis de aninhamento. Para obter mais informações sobre YAML, consulte [YAML Ain't Markup Language \(YAML™\) versão 1.1](#).

- **Exclusividade**: use cada chave somente uma vez em cada arquivo de configuração.

### Aviso

Se você usar uma chave (por exemplo, `option_settings`) duas vezes no mesmo arquivo de configuração, uma das seções será descartada. Combine seções duplicadas em uma única seção ou coloque-as em arquivos de configuração separados.

O processo de implantação varia ligeiramente, dependendo do cliente que você usa para gerenciar seus ambientes. Para detalhes, consulte estas seções:

- [Console do Elastic Beanstalk](#)
- [EB CLI](#)
- [AWS CLI](#)

## Tópicos

- [Configurações de opção](#)
- [Personalizar software em servidores Linux](#)
- [Personalizar software em Windows Servers](#)
- [Adicionar e personalizar recursos do ambiente do Elastic Beanstalk](#)

## Configurações de opção

É possível usar a chave `option_settings` para modificar a configuração do Elastic Beanstalk e definir variáveis que podem ser recuperadas da aplicação usando variáveis de ambiente. Alguns namespaces permitem estender o número e especificar os nomes dos parâmetros. Para obter uma lista de namespaces e as opções de configuração, consulte [Opções de configuração](#).

É possível também aplicar as configurações de opção diretamente a um ambiente durante a criação ou a atualização dele. As configurações aplicadas diretamente ao ambiente substituem aquelas referentes às mesmas opções nos arquivos de configuração. Se você remover as configurações de uma configuração do ambiente, as dos arquivos de configuração entrarão em vigor. Consulte o [Precedência](#) para obter detalhes.

### Sintaxe

A sintaxe padrão para configurações de opção é uma matriz de objetos, cada um com uma chave namespace, `option_name` e `value`.

```
option_settings:
 - namespace: namespace
 option_name: option name
 value: option value
 - namespace: namespace
 option_name: option name
 value: option value
```

A chave namespace é opcional. Se você não especificar um namespace, o padrão usado será `aws:elasticbeanstalk:application:environment`:

```
option_settings:
 - option_name: option name
 value: option value
 - option_name: option name
 value: option value
```

O Elastic Beanstalk também é compatível com uma sintaxe abreviada para configurações de opção que permite especificar opções como pares de chave/valor sob o namespace:

```
option_settings:
 namespace:
 option name: option value
```

```
option name: option value
```

## Exemplos

Os exemplos a seguir definem uma opção específica da plataforma Tomcat no namespace `aws:elasticbeanstalk:container:tomcat:jvmoptions` e uma propriedade de ambiente chamada `MYPARAMETER`.

No formato YAML padrão:

Example `.ebextensions/options.config`

```
option_settings:
 - namespace: aws:elasticbeanstalk:container:tomcat:jvmoptions
 option_name: Xmx
 value: 256m
 - option_name: MYPARAMETER
 value: parametervalue
```

No formato abreviado:

Example `.ebextensions/options.config`

```
option_settings:
 aws:elasticbeanstalk:container:tomcat:jvmoptions:
 Xmx: 256m
 aws:elasticbeanstalk:application:environment:
 MYPARAMETER: parametervalue
```

Em JSON:

Example `.ebextensions/options.config`

```
{
 "option_settings": [
 {
 "namespace": "aws:elasticbeanstalk:container:tomcat:jvmoptions",
 "option_name": "Xmx",
 "value": "256m"
 },
 {
 "option_name": "MYPARAMETER",
```

```
 "value": "parametervalue"
 }
]
}
```

## Personalizar software em servidores Linux

Você pode personalizar e configurar o software do qual seu aplicativo depende. É possível adicionar comandos a serem executados durante o provisionamento da instância, definir usuários e grupos do Linux e baixar ou criar arquivos diretamente nas instâncias do ambiente. Esses arquivos podem ser dependências exigidas pela aplicação, por exemplo, pacotes adicionais do repositório yum, ou podem ser arquivos de configuração, por exemplo, um arquivo de configuração de proxy para substituir configurações específicas usadas como padrão pelo Elastic Beanstalk.

Esta seção descreve o tipo de informações que você pode incluir em um arquivo de configuração para personalizar o software nas instâncias do EC2 que executam o Linux. Para obter informações gerais sobre como personalizar e configurar ambientes, consulte [Configurar ambientes do Elastic Beanstalk](#). Para obter informações sobre a personalização de software nas instâncias do EC2 que executam o Windows, consulte [Personalizar software em Windows Servers](#).

### Observações

- Em plataformas do Amazon Linux 2, em vez de fornecer arquivos e comandos em arquivos de configuração .ebextensions, é altamente recomendável usar Buildfile, Procfile, e hooks de plataforma sempre que possível para configurar e executar um código personalizado nas instâncias do ambiente durante o provisionamento da instância. Para obter detalhes sobre esses mecanismos, consulte [the section called “Estender plataformas Linux”](#).
- YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Os arquivos de configuração oferecem suporte às seguintes chaves que afetam o servidor Linux no qual seu aplicativo é executado.

### Chaves

- [Pacotes](#)
- [Grupos](#)

- [Usuários](#)
- [Origens](#)
- [Arquivos](#)
- [Comandos](#)
- [Serviços](#)
- [Comandos de contêiner](#)
- [Exemplo: como usar métricas personalizadas do Amazon CloudWatch](#)

As chaves são processadas na ordem em que aparecem listadas aqui.

Assista a seus [eventos](#) de ambiente enquanto desenvolve e testa os arquivos de configuração. O Elastic Beanstalk ignora um arquivo de configuração que contém erros de validação, como uma chave inválida, e não processa nenhuma das outras chaves no mesmo arquivo. Quando isso acontece, o Elastic Beanstalk adiciona um evento de aviso ao log de eventos.

## Pacotes

Use a chave `packages` para fazer download e instalar aplicativos e componentes predefinidos.

### Sintaxe

```
packages:
 name of package manager:
 package name: version
 ...
 name of package manager:
 package name: version
 ...
 ...
```

Você pode especificar vários pacotes sob cada chave do gerenciador de pacotes.

### Formatos de pacote com suporte

Atualmente, o Elastic Beanstalk é compatível com os seguintes gerenciadores de pacote: yum, rubygems, python e rpm. Os pacotes são processados na seguinte ordem: rpm, yum e, em seguida, rubygems e python. Não há ordem entre rubygems e python. Dentro de cada gerenciador de pacotes, a ordem de instalação do pacote não é garantida. Use um gerenciador de pacotes compatível com o seu sistema operacional.



**Note**

O Elastic Beanstalk é compatível com dois gerenciadores de pacotes subjacentes para Python, pip e easy\_install. No entanto, na sintaxe do arquivo de configuração, especifique o nome do gerenciador de pacotes como python. Quando você usa um arquivo de configuração para especificar um gerenciador de pacotes Python, o Elastic Beanstalk usa o Python 2.7. Se seu aplicativo conta com uma versão diferente do Python, especifique os pacotes a serem instalados em um arquivo `requirements.txt`. Para obter mais informações, consulte [Especificação de dependências usando um arquivo de requisitos](#).

## Como especificar versões

Dentro de cada gerenciador de pacotes, cada pacote é especificado como um nome de pacote e uma lista de versões. A versão pode ser uma sequência, uma lista de versões ou uma sequência ou lista vazia. Uma sequência ou lista vazia indica que você deseja a versão mais recente. Para o gerenciador de rpm, a versão é especificada como um caminho para um arquivo no disco ou um URL. Não há suporte para caminhos relativos.

Se você especificar uma versão de um pacote, o Elastic Beanstalk tentará instalar essa versão, mesmo que uma versão mais recente do pacote já esteja instalada na instância. Se uma versão mais recente já estiver instalada, a implantação não terá êxito. Alguns gerenciadores de pacotes oferecem suporte a várias versões, mas outros não. Verifique a documentação do seu gerenciador de pacotes para obter mais informações. Se você não especificar uma versão e uma versão do pacote já estiver instalada, o Elastic Beanstalk não instalará uma nova versão, pois vai considerar que você deseja manter e usar a versão existente.

## Trecho de exemplo

O trecho a seguir especifica um URL da versão para rpm, solicita a versão mais recente do yum e a versão 0.10.2 de chef do rubygems.

```
packages:
 yum:
 libmemcached: []
 ruby-devel: []
 gcc: []
 rpm:
 epel: http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm
 rubygems:
```

```
chef: '0.10.2'
```

## Grupos

Use a chave `groups` para criar grupos do Linux/UNIX e para atribuir IDs de grupos. Para criar um grupo, adicione um novo par de chave/valor que mapeie um novo nome de grupo para um ID de grupo opcional. A chave de grupos pode conter um ou mais nomes de grupos. A tabela a seguir lista as chaves disponíveis.

### Sintaxe

```
groups:
 name of group: {}
 name of group:
 gid: "group id"
```

### Opções

#### gid

Um número de ID de grupo.

Se um ID de grupo for especificado e o grupo já existir por nome, a criação do grupo não terá êxito. Se outro grupo tiver o ID de grupo especificado, o sistema operacional poderá rejeitar a respectiva criação.

### Trecho de exemplo

O trecho a seguir especifica um grupo chamado `groupOne`, sem atribuir um ID de grupo, e um grupo chamado `groupTwo` que especificou um valor de ID de grupo 45.

```
groups:
 groupOne: {}
 groupTwo:
 gid: "45"
```

## Usuários

Use a chave `users` para criar usuários Linux/UNIX na instância do EC2.

## Sintaxe

```
users:
 name of user:
 groups:
 - name of group
 uid: "id of the user"
 homeDir: "user's home directory"
```

## Opções

### uid

Um ID de usuário. Haverá falha no processo de criação se o nome do usuário existir com outro ID de usuário. Se o ID de usuário já estiver atribuído a um usuário existente, o sistema operacional poderá rejeitar a solicitação de criação.

### groups

Uma lista de nomes de grupos. O usuário é adicionado a cada grupo da lista.

### homeDir

O diretório base de usuário.

Os usuários são criados como usuários de sistema não interativos com um shell de `/sbin/nologin`. Este é o design e não pode ser modificado.

## Trecho de exemplo

```
users:
 myuser:
 groups:
 - group1
 - group2
 uid: "50"
 homeDir: "/tmp"
```

## Origens

É possível usar a chave `sources` para fazer download de um arquivo compactado de um URL público e descompactá-lo no diretório de destino da instância do EC2.

## Sintaxe

```
sources:
 target directory: location of archive file
```

### Formatos com suporte

Os formatos com suporte são tar, tar+gzip, tar+bz2 e zip. Você pode fazer referência a locais externos, como o Amazon Simple Storage Service (Amazon S3) (por exemplo, `https://mybucket.s3.amazonaws.com/myobject`), desde que o URL seja publicamente acessível.

### Trecho de exemplo

O exemplo a seguir faz download de um arquivo .zip público de um bucket do Amazon S3 e o descompacta em `/etc/myapp`:

```
sources:
 /etc/myapp: https://mybucket.s3.amazonaws.com/myobject
```

#### Note

Não reutilize o mesmo caminho de destino para várias extrações. A extração de outra origem para o mesmo caminho de destino substituirá em vez de anexar aos conteúdos.

## Arquivos

Você pode usar a chave `files` para criar arquivos na instância do EC2. O conteúdo pode estar embutido no arquivo de configuração ou o conteúdo pode ser obtido de um URL. Os arquivos são gravados em disco em ordem lexicográfica.

É possível usar a chave `files` para fazer download de arquivos privados do Amazon S3 ao fornecer o perfil da instância para autorização.

Se o caminho do arquivo especificado já existir na instância, o arquivo existente será mantido com a extensão `.bak` anexada ao nome.

## Sintaxe

```
files:
```

```
"target file location on disk":
 mode: "six-digit octal value"
 owner: name of owning user for file
 group: name of owning group for file
 source: URL
 authentication: authentication name:

"target file location on disk":
 mode: "six-digit octal value"
 owner: name of owning user for file
 group: name of owning group for file
 content: |
 # this is my
 # file content
 encoding: encoding format
 authentication: authentication name:
```

## Opções

### content

Conteúdo de sequência para adicionar ao arquivo. Especifique content ou source, mas não ambos.

### source

URL de um arquivo para fazer download: Especifique content ou source, mas não ambos.

### encoding

O formato de codificação da sequência especificada com a opção content.

Valores válidos: plain | base64

### group

Grupo do Linux que possui o arquivo.

### owner

Usuário do Linux que possui o arquivo.

### mode

Um valor octal de seis dígitos que representa o modo para este arquivo. Sem suporte para sistemas Windows. Use os primeiros três dígitos para symlinks e os três últimos dígitos para

configurar permissões. Para criar um symlink, especifique `120xxx`, em que `xxx` define as permissões do arquivo de destino. Para especificar permissões para um arquivo, use os últimos três dígitos, como `000644`.

## authentication

O nome de um [método de autenticação do AWS CloudFormation](#) para usar. É possível adicionar métodos de autenticação aos metadados do grupo de Auto Scaling com a chave `Resources`. Veja um exemplo a seguir.

### Trecho de exemplo

```
files:
 "/home/ec2-user/myfile" :
 mode: "000755"
 owner: root
 group: root
 source: http://foo.bar/myfile

 "/home/ec2-user/myfile2" :
 mode: "000755"
 owner: root
 group: root
 content: |
 this is my
 file content
```

Exemplo usando um symlink. Isso cria um link `/tmp/myfile2.txt` que aponta para o arquivo existente `/tmp/myfile1.txt`.

```
files:
 "/tmp/myfile2.txt" :
 mode: "120400"
 content: "/tmp/myfile1.txt"
```

O exemplo a seguir usa a chave `Resources` para adicionar um método de autenticação chamado `S3Auth` e o usa para fazer download de um arquivo privado de um bucket do Amazon S3:

```
Resources:
 AWSEBAutoScalingGroup:
 Metadata:
```

```
AWS::CloudFormation::Authentication:
 S3Auth:
 type: "s3"
 buckets: ["elasticbeanstalk-us-west-2-123456789012"]
 roleName:
 "Fn::GetOptionSetting":
 Namespace: "aws:autoscaling:launchconfiguration"
 OptionName: "IamInstanceProfile"
 DefaultValue: "aws-elasticbeanstalk-ec2-role"

files:
 "/tmp/data.json" :
 mode: "000755"
 owner: root
 group: root
 authentication: "S3Auth"
 source: https://elasticbeanstalk-us-west-2-123456789012.s3-us-west-2.amazonaws.com/
 data.json
```

## Comandos

É possível usar a chave `commands` para executar comandos na instância do EC2. Os comandos executados antes da configuração do aplicativo e do servidor web e da extração do arquivo de versão do aplicativo.

Os comandos especificados são executadas como o usuário raiz e processados em ordem alfabética por nome. Por padrão, os comandos são executados no diretório raiz. Para executar comandos de outro diretório, use a opção `cwd`.

Para solucionar problemas com seus comandos, você pode encontrar a saída deles nos [logs da instância](#).

## Sintaxe

```
commands:
 command name:
 command: command to run
 cwd: working directory
 env:
 variable name: variable value
 test: conditions for command
 ignoreErrors: true
```

## Opções

### command

Uma matriz ([coleção de sequência em bloco](#) em sintaxe de YAML) ou uma cadeia que especifica o comando a ser executado. Algumas observações importantes:

- Se você usar uma cadeia, você não precisa colocar a cadeia inteira entre aspas. Se você usar aspas, pule as ocorrências literais do mesmo tipo de citação.
- Se você usar uma matriz, não precisará de caracteres de espaço de escape ou de aspas para parâmetros de comandos. Cada elemento de matriz é um único argumento de comando. Não use uma matriz para especificar vários comandos.

Os exemplos a seguir são todos equivalentes:

```
commands:
 command1:
 command: git commit -m "This is a comment."
 command2:
 command: "git commit -m \"This is a comment.\""
 command3:
 command: 'git commit -m "This is a comment."'
 command4:
 command:
 - git
 - commit
 - -m
 - This is a comment.
```

Para especificar múltiplos comandos, use um [escalar de bloco literal](#), como exibido no exemplo a seguir.

```
commands:
 command block:
 command: |
 git commit -m "This is a comment."
 git push
```

### env

(Opcional) Define variáveis de ambiente para o comando. Essa propriedade substitui, em vez de anexar, o ambiente existente.



## `cwd`

(Opcional) O diretório de trabalho. Se não for especificado, os comandos serão executados a partir do diretório raiz (`/`).

## `test`

(Opcional) Um comando que deve retornar o valor `true` (código de saída 0) para que o Elastic Beanstalk processe o comando, por exemplo, um script de shell, contido na chave `command`.

## `ignoreErrors`

(Opcional) Um valor booleano que determina se outros comandos deverão ser executados se houver falha no comando incluído na chave `command` (retorna um valor diferente de zero). Defina esse valor como `true` para continuar executando os comandos mesmo se houver falha no comando. Defina-o como `false` para interromper a execução dos comandos em caso de falha no comando. O valor padrão é `false`.

## Trecho de exemplo

O exemplo a seguir executa um trecho de script do Python.

```
commands:
 python_install:
 command: myscript.py
 cwd: /home/ec2-user
 env:
 myvarname: myvarvalue
 test: "[-x /usr/bin/python]"
```

## Serviços

É possível usar a chave `services` para definir quais serviços deverão ser iniciados ou interrompidos quando a instância for executada. A chave `services` também permite especificar dependências de origens, pacotes e arquivos para que, se for necessária uma reinicialização devido à instalação de arquivos, o Elastic Beanstalk reinicie o serviço.

## Sintaxe

```
services:
 sysvinit:
```

```
name of service:
 enabled: "true"
 ensureRunning: "true"
 files:
 - "file name"
 sources:
 - "directory"
 packages:
 name of package manager:
 "package name[: version]"
 commands:
 - "name of command"
```

## Opções

### ensureRunning

Defina `true` para garantir que o serviço esteja em execução depois que o Elastic Beanstalk for concluído.

Defina `false` para garantir que o serviço não esteja em execução depois que o Elastic Beanstalk for concluído.

Omita essa chave para não fazer nenhuma alteração no estado do serviço.

### enabled

Defina como `true` para garantir que o serviço seja iniciado automaticamente na inicialização.

Defina como `false` para garantir que o serviço não seja iniciado automaticamente na inicialização.

Omita essa chave para não fazer nenhuma alteração nessa propriedade.

### files

Uma lista de arquivos. Se o Elastic Beanstalk alterar um diretamente por meio do bloco de arquivos, o serviço será reiniciado.

### sources

Uma lista de diretórios. Se o Elastic Beanstalk expandir um arquivo em um desses diretórios, o serviço será reiniciado.

## packages

Um mapa do gerenciador de pacotes para uma lista de nomes de pacotes. Se o Elastic Beanstalk instalar ou atualizar um desses pacotes, o serviço será reiniciado.

## commands

Uma lista de nomes de comandos. Se o Elastic Beanstalk executar o comando especificado, o serviço será reiniciado.

## Trecho de exemplo

Veja, a seguir, um trecho de exemplo:

```
services:
 sysvinit:
 myservice:
 enabled: true
 ensureRunning: true
```

## Comandos de contêiner

É possível usar a chave `container_commands` para executar comandos que afetam o código-fonte do seu aplicativo. Os comandos de contêiner são executados depois que o aplicativo e o servidor Web são configurados e o arquivo de versão do aplicativo é extraído, mas antes da implantação da versão do aplicativo. Os comandos que não são de contêiner e outras operações de personalização são executados antes da extração do código-fonte do aplicativo.

Os comandos especificados são executadas como o usuário raiz e processados em ordem alfabética por nome. Os comandos de contêiner são executados no diretório de preparação, onde seu código-fonte é extraído antes de ser implantado no servidor de aplicativo. Quaisquer alterações que você fizer em seu código-fonte no diretório de preparação com um comando de contêiner serão incluídas quando a fonte for implantada em seu local final.

### Note

A saída dos comandos de contêiner é registrada no log da instância `cfn-init-cmd.log`. Para obter mais informações sobre como recuperar e visualizar logs de instâncias, consulte [Visualizar logs de instâncias do Amazon EC2](#).

É possível usar `leader_only` para somente executar o comando em uma única instância ou configurar um `test` para somente executar o comando quando um comando de teste for avaliado como `true`. Os comandos de contêiner somente líderes são executados apenas durante a criação do ambiente e as implantações, enquanto outros comandos e operações de personalização de servidor são executados sempre que uma instância é provisionada ou atualizada. Os comandos de contêiner somente líderes não são executados devido a alterações na configuração de execução, como uma alteração no ID do AMI ou no tipo de instância.

## Sintaxe

```
container_commands:
 name of container_command:
 command: "command to run"
 leader_only: true
 name of container_command:
 command: "command to run"
```

## Opções

### command

Uma string ou matriz de strings para execução.

### env

(Opcional) Defina variáveis de ambiente antes de executar o comando, substituindo qualquer valor existente.

### cwd

(Opcional) O diretório de trabalho. Por padrão, este é o diretório de preparação do aplicativo descompactado.

### leader\_only

(Opcional) Somente execute o comando em uma única instância escolhida pelo Elastic Beanstalk. Os comandos de contêiner somente líderes são executados antes dos outros comandos de contêiner. Um comando pode ser somente líder ou ter um `test`, mas não as duas coisas (`leader_only` tem precedência).

## test

(Opcional) Execute um comando de teste que deve retornar `true` para executar o comando de contêiner. Um comando pode ser somente líder ou ter um `test`, mas não as duas coisas (`leader_only` tem precedência).

## ignoreErrors

(Opcional) Não falhar implantações se o comando de contêiner retornar um valor diferente de 0 (êxito). Defina como `true` para habilitar.

## Trecho de exemplo

Veja, a seguir, um trecho de exemplo.

```
container_commands:
 collectstatic:
 command: "django-admin.py collectstatic --noinput"
 01syncdb:
 command: "django-admin.py syncdb --noinput"
 leader_only: true
 02migrate:
 command: "django-admin.py migrate"
 leader_only: true
 99customize:
 command: "scripts/customize.sh"
```

## Exemplo: como usar métricas personalizadas do Amazon CloudWatch

O Amazon CloudWatch é um serviço web que permite monitorar, gerenciar e publicar várias métricas, bem como configurar ações de alarmes com base em dados de métricas. É possível definir métricas personalizadas para seu próprio uso, e o Elastic Beanstalk enviará essas métricas para o Amazon CloudWatch. Assim que o Amazon CloudWatch contiver suas métricas personalizadas, você poderá visualizar as que estão no console do Amazon CloudWatch.

### Important

Os scripts de monitoramento do Amazon CloudWatch estão obsoletos. O agente do CloudWatch agora substituiu os scripts de monitoramento do CloudWatch para coletar métricas e logs.

Se você ainda estiver migrando scripts de monitoramento obsoletos para o agente e precisa de informações sobre os scripts de monitoramento, consulte [Obsoleto: Coletar métricas usando scripts de monitoramento do CloudWatch](#), no Guia do usuário do Amazon EC2 para instâncias Linux.

## O agente do Amazon CloudWatch

O agente do Amazon CloudWatch permite a coleta de métricas e logs do CloudWatch de instâncias do Amazon EC2 e de servidores on-premises entre sistemas operacionais. O agente oferece suporte a métricas coletadas no nível do sistema. Ele também oferece suporte para a coleta de métricas personalizadas de aplicações ou serviços. Para obter mais informações sobre o agente do CloudWatch, consulte [Coletar métricas e logs com o agente do CloudWatch](#) no Guia do usuário do Amazon CloudWatch.

### Note

Os [Relatórios aprimorados de integridade](#) do Elastic Beanstalk têm suporte nativo para a publicação de uma ampla variedade de métricas de instância e ambiente do CloudWatch. Para mais detalhes, consulte [Publicar métricas personalizadas do Amazon CloudWatch para um ambiente](#).

## Tópicos

- [Arquivo de configuração .Ebextensions](#)
- [Permissões](#)
- [Visualizar métricas no console do CloudWatch](#)

## Arquivo de configuração .Ebextensions

Este exemplo usa arquivos e comandos em um arquivo de configuração .ebextensions para configurar e executar o agente do Amazon CloudWatch na plataforma Amazon Linux 2. O agente faz parte do pacote do Amazon Linux 2. Se você estiver usando um sistema operacional diferente, etapas adicionais para instalar o agente podem ser necessárias. Para obter mais informações, consulte [Instalar o agente do CloudWatch](#) no Guia do usuário do Amazon CloudWatch.

Para usar esse exemplo, salve-o em um arquivo chamado `ccloudwatch.config` em um diretório chamado `.ebextensions` no nível superior do diretório do projeto e implante a aplicação usando o

console do Elastic Beanstalk (inclua o diretório `.ebextensions` em seu [pacote de origem](#)) ou na [CLI do EB](#).

Para obter mais informações sobre esses arquivos de configuração, consulte [Personalização avançada de ambiente com arquivos de configuração \(.ebextensions\)](#).

`.ebextensions/cloudwatch.config`

```
files:
 "/opt/aws/amazon-cloudwatch-agent/bin/config.json":
 mode: "000600"
 owner: root
 group: root
 content: |
 {
 "agent": {
 "metrics_collection_interval": 60,
 "run_as_user": "root"
 },
 "metrics": {
 "namespace": "System/Linux",
 "append_dimensions": {
 "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
 },
 "metrics_collected": {
 "mem": {
 "measurement": [
 "mem_used_percent"
]
 }
 }
 }
 }
container_commands:
 start_cloudwatch_agent:
 command: /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json
```

Este arquivo tem duas seções:

- `files`: essa seção adiciona o arquivo de configuração do agente. Ela indica quais métricas e logs o agente deve enviar para o Amazon CloudWatch. Neste exemplo, estamos enviando apenas a métrica `mem_used_percent`. Para obter uma lista completa das métricas no nível do sistema que

são compatíveis com o agente do Amazon CloudWatch, consulte [Métricas coletadas pelo agente do CloudWatch](#) no Guia do usuário do Amazon CloudWatch.

- `container_commands`: essa seção contém o comando que inicia o agente, passando o arquivo de configuração como um parâmetro. Para obter mais detalhes sobre o `container_commands`, consulte [Comandos de contêiner](#).

## Permissões

As instâncias do ambiente precisam das permissões adequadas do IAM para publicar métricas personalizadas do Amazon CloudWatch, usando o agente do CloudWatch. Você pode conceder permissões às instâncias do ambiente adicionando-as ao [perfil da instância](#) do ambiente. Você pode adicionar permissões ao perfil da instância antes ou depois da implantação de seu aplicativo.

Para conceder permissões para publicar métricas do CloudWatch

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, selecione Roles (Funções).
3. Escolha a função de perfil da instância de seu ambiente. Por padrão, ao criar um ambiente com o console do Elastic Beanstalk ou com a [CLI do EB](#), ela é `aws-elasticbeanstalk-ec2-role`.
4. Escolha a guia Permissions (Permissões).
5. Em Permissions Policies (Políticas de permissões), na seção Permissions (Permissões) escolha Attach policies (Anexar políticas).
6. Em Attach Permissions (Anexar políticas), escolha a política gerenciada pela AWS CloudWatchAgentServerPolicy. Clique em Attach Policy (Anexar política).

Para obter mais informações sobre o gerenciamento de políticas, consulte [Trabalhar com políticas](#) no Guia do usuário do IAM.

## Visualizar métricas no console do CloudWatch

Após a implantação do arquivo de configuração do CloudWatch no ambiente, consulte o [console do Amazon CloudWatch](#) para visualizar suas métricas. As métricas personalizadas estarão localizadas no namespace CWAgent.

Para obter mais informações, consulte [Visualização de métricas disponíveis](#) no Manual do usuário do Amazon CloudWatch.



## Personalizar software em Windows Servers

Você pode personalizar e configurar o software do qual seu aplicativo depende. Estes arquivos podem ser dependências exigidas pelo aplicativo, por exemplo, pacotes adicionais ou serviços que precisam ser executados. Para obter informações gerais sobre como personalizar e configurar ambientes do Elastic Beanstalk, consulte [Configurar ambientes do Elastic Beanstalk](#).

### Note

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Os arquivos de configuração oferecem suporte às seguintes chaves que afetam o servidor Windows no qual seu aplicativo é executado.

### Chaves

- [Pacotes](#)
- [Origens](#)
- [Arquivos](#)
- [Comandos](#)
- [Serviços](#)
- [Comandos de contêiner](#)

As chaves são processadas na ordem em que aparecem listadas aqui.

### Note

Versões mais antigas (sem versionamento) da plataforma .NET não processam arquivos de configuração na ordem correta. Saiba mais em [Migrar entre versões principais da plataforma do Windows Server do Elastic Beanstalk](#).

Assista a seus [eventos](#) de ambiente enquanto desenvolve e testa os arquivos de configuração. O Elastic Beanstalk ignora um arquivo de configuração que contém erros de validação, como uma

chave inválida, e não processa nenhuma das outras chaves no mesmo arquivo. Quando isso acontece, o Elastic Beanstalk adiciona um evento de aviso ao log de eventos.

## Pacotes

Use a chave `packages` para fazer download e instalar aplicativos e componentes predefinidos.

Em ambientes Windows, o Elastic Beanstalk é compatível com o download e a instalação de pacotes MSI. (Os ambientes do Linux são compatíveis com outros gerenciadores de pacotes. Para obter detalhes, consulte [Pacotes](#) na página Personalizar software em servidores Linux).

É possível fazer referência a locais externos, como o Amazon Simple Storage Service (Amazon S3), desde que o URL seja publicamente acessível.

Se você especificar vários pacotes `msi:`, sua ordem de instalação não será garantida.

### Sintaxe

Especifique um nome de sua preferência para o pacote e, como valor, um URL para um arquivo MSI local. Você pode especificar vários pacotes sob a chave `msi:`.

```
packages:
 msi:
 package name: package url
 ...
```

### Exemplos

O exemplo a seguir especifica um URL para fazer download do mysql em `https://dev.mysql.com/`.

```
packages:
 msi:
 mysql: https://dev.mysql.com/get/Downloads/Connector-Net/mysql-connector-
net-8.0.11.msi
```

O exemplo a seguir especifica um objeto do Amazon S3 como o local do arquivo MSI.

```
packages:
 msi:
```

```
mymsi: https://mybucket.s3.amazonaws.com/myobject.msi
```

## Origens

Use a chave `sources` para fazer download de um arquivo compactado de um URL público e descompacte-o no diretório de destino na instância do EC2.

### Sintaxe

```
sources:
 target directory: location of archive file
```

### Formatos com suporte

Em ambientes Windows, o Elastic Beanstalk é compatível com o formato `.zip`. (Os ambientes do Linux são compatíveis com outros formatos. Para obter detalhes, consulte [Origens](#) na página Personalizar software em servidores Linux.)

É possível fazer referência a locais externos, como o Amazon Simple Storage Service (Amazon S3), desde que o URL seja publicamente acessível.

### Exemplo

O exemplo a seguir faz download de um arquivo `.zip` público de um bucket do Amazon S3 e o descompacta em `c:/myproject/myapp`.

```
sources:
 "c:/myproject/myapp": https://mybucket.s3.amazonaws.com/myobject.zip
```

## Arquivos

Use a chave `files` para criar arquivos na instância do EC2. O conteúdo pode ser incorporado no arquivo de configuração ou obtido de um URL. Os arquivos são gravados em disco em ordem lexicográfica. Para fazer download de arquivos privados do Amazon S3, forneça o perfil da instância para autorização.

### Sintaxe

```
files:
 "target file location on disk":
```

```
source: URL
authentication: authentication name:

"target file location on disk":
 content: |
 this is my content
 encoding: encoding format
```

## Opções

### content

(Opcional) Uma string.

### source

(Opcional) O URL de onde o arquivo é carregado. Essa opção não pode ser especificada com a chave de conteúdo.

### encoding

(Opcional) O formato de codificação. Essa opção só é usada para um valor de chave de conteúdo fornecido. O valor padrão é `plain`.

Valores válidos: `plain` | `base64`

### authentication

(Opcional) O nome de um [método de autenticação do AWS CloudFormation](#) a ser usado. É possível adicionar métodos de autenticação aos metadados do grupo de Auto Scaling com a chave `Resources`.

## Exemplos

O exemplo a seguir mostra as duas maneiras de fornecer conteúdo do arquivo: por meio de um URL ou em linha no arquivo de configuração.

```
files:
 "c:\\targetdirectory\\targetfile.txt":
 source: http://foo.bar/myfile

 "c:/targetdirectory/targetfile.txt":
 content: |
```

```
this is my file
with content
```

### Note

Se você usar uma barra invertida (\) no caminho do arquivo, deverá precedê-la com outra barra invertida (caractere de escape), como mostrado no exemplo anterior.

O exemplo a seguir usa a chave `Resources` para adicionar um método de autenticação chamado `S3Auth` e o usa para fazer download de um arquivo privado de um bucket do Amazon S3:

```
files:
 "c:\\targetdirectory\\targetfile.zip":
 source: https://elasticbeanstalk-us-east-2-123456789012.s3.amazonaws.com/prefix/
myfile.zip
 authentication: S3Auth

Resources:
 AWSEBAutoScalingGroup:
 Metadata:
 AWS::CloudFormation::Authentication:
 S3Auth:
 type: "s3"
 buckets: ["elasticbeanstalk-us-east-2-123456789012"]
 roleName:
 "Fn::GetOptionSetting":
 Namespace: "aws:autoscaling:launchconfiguration"
 OptionName: "IamInstanceProfile"
 DefaultValue: "aws-elasticbeanstalk-ec2-role"
```

## Comandos

Use a chave `commands` para executar comandos na instância do EC2. Os comandos são processados em ordem alfabética por nome e executados antes que o aplicativo e o servidor Web sejam configurados e o arquivo de versão do aplicativo seja extraído.

Os comandos especificados são executados como usuário administrador.

Para solucionar problemas com seus comandos, você pode encontrar a saída deles nos [logs da instância](#).

## Sintaxe

```
commands:
 command name:
 command: command to run
```

### Opções

#### command

Uma matriz ou uma string que especifica o comando a ser executado. Se você usar uma matriz, não precisará de caracteres de espaço de escape ou de aspas para parâmetros de comandos.

#### cwd

(Opcional) O diretório de trabalho. Por padrão, o Elastic Beanstalk tenta encontrar o local do diretório do projeto. Se não for encontrado, ele usará `c:\Windows\System32` como padrão.

#### env

(Opcional) Define variáveis de ambiente para o comando. Essa propriedade substitui, em vez de anexar, o ambiente existente.

#### ignoreErrors

(Opcional) Um valor booleano que determina se outros comandos deverão ser executados se houver falha no comando incluído na chave `command` (retorna um valor diferente de zero). Defina esse valor como `true` para continuar executando os comandos mesmo se houver falha no comando. Defina-o como `false` para interromper a execução dos comandos em caso de falha no comando. O valor padrão é `false`.

#### test

(Opcional) Um comando que deve retornar o valor `true` (código de saída 0) para que o Elastic Beanstalk processe o comando incluído na chave `command`.

#### waitAfterCompletion

(Opcional) Quantos segundos aguardar para executar o próximo comando depois que o comando é concluído. Se o sistema exigir reinicialização após a conclusão do comando, ele será reinicializado após o número de segundos especificado. Se o sistema for reinicializado como resultado de um comando, o Elastic Beanstalk recuperará até o ponto depois do comando no arquivo de configuração. O valor padrão é **60** segundos. Você também pode especificar **forever**, mas o sistema deve ser reinicializado antes de você executar outro comando.

## Exemplo

O exemplo a seguir salva a saída do comando `set` no arquivo especificado. Se houver um comando subsequente, o Elastic Beanstalk o executará logo após o término deste comando. Se esse comando exigir reinicialização, o Elastic Beanstalk reinicializará a instância logo após o término do comando.

```
commands:
 test:
 command: set > c:\\myapp\\set.txt
 waitAfterCompletion: 0
```

## Serviços

Use a chave `services` para definir quais serviços deverão ser iniciados ou interrompidos quando a instância for executada. A chave `services` também permite especificar dependências em fontes, pacotes e arquivos para que, se for necessária uma reinicialização devido à instalação de arquivos, o Elastic Beanstalk reinicie o serviço.

## Sintaxe

```
services:
 windows:
 name of service:
 files:
 - "file name"
 sources:
 - "directory"
 packages:
 name of package manager:
 "package name[: version]"
 commands:
 - "name of command"
```

## Opções

### ensureRunning

(Opcional) Defina `true` para garantir que o serviço esteja sendo executado após o Elastic Beanstalk ser concluído.

Defina `false` para garantir que o serviço não esteja em execução depois que o Elastic Beanstalk for concluído.

Omita essa chave para não fazer nenhuma alteração no estado do serviço.

### enabled

(Opcional) Defina como `true` para garantir que o serviço seja iniciado automaticamente na inicialização.

Defina como `false` para garantir que o serviço não seja iniciado automaticamente na inicialização.

Omita essa chave para não fazer nenhuma alteração nessa propriedade.

### files

Uma lista de arquivos. Se o Elastic Beanstalk alterar um diretamente por meio do bloco de arquivos, o serviço será reiniciado.

### sources

Uma lista de diretórios. Se o Elastic Beanstalk expandir um arquivo em um desses diretórios, o serviço será reiniciado.

### packages

Um mapa do gerenciador de pacotes para uma lista de nomes de pacotes. Se o Elastic Beanstalk instalar ou atualizar um desses pacotes, o serviço será reiniciado.

### commands

Uma lista de nomes de comandos. Se o Elastic Beanstalk executar o comando especificado, o serviço será reiniciado.

## Exemplo

```
services:
 windows:
 myservice:
 enabled: true
 ensureRunning: true
```

## Comandos de contêiner

Use a chave `container_commands` para executar comandos que afetam o código-fonte do seu aplicativo. Os comandos de contêiner são executados depois que o aplicativo e o servidor Web são



configurados e o arquivo de versão do aplicativo é extraído, mas antes da implantação da versão do aplicativo. Os comandos que não são de contêiner e outras operações de personalização são executados antes da extração do código-fonte do aplicativo.

Os comandos de contêiner são executados no diretório de preparação, onde seu código-fonte é extraído antes de ser implantado no servidor de aplicativo. Quaisquer alterações que você fizer em seu código-fonte no diretório de preparação com um comando de contêiner serão incluídas quando a fonte for implantada em seu local final.

Para solucionar problemas com seus comandos de contêiner, você pode encontrar a saída deles nos [logs da instância](#).

Use a opção `leader_only` para somente executar o comando em uma única instância ou configure `test` para somente executar o comando quando um comando de teste é avaliado como `true`. Os comandos de contêiner somente líderes são executados apenas durante a criação do ambiente e as implantações, enquanto outros comandos e operações de personalização de servidor são executados sempre que uma instância é provisionada ou atualizada. Os comandos de contêiner somente líderes não são executados devido a alterações na configuração de execução, como uma alteração no ID do AMI ou no tipo de instância.

## Sintaxe

```
container_commands:
 name of container_command:
 command: command to run
```

## Opções

### command

Uma string ou matriz de strings para execução.

### env

(Opcional) Defina variáveis de ambiente antes de executar o comando, substituindo qualquer valor existente.

### cwd

(Opcional) O diretório de trabalho. Por padrão, este é o diretório de preparação do aplicativo descompactado.

## leader\_only

(Opcional) Somente execute o comando em uma única instância escolhida pelo Elastic Beanstalk. Os comandos de contêiner somente líderes são executados antes dos outros comandos de contêiner. Um comando pode ser somente líder ou ter um `test`, mas não as duas coisas (`leader_only` tem precedência).

## test

(Opcional) Execute um comando de teste que deve retornar `true` para executar o comando de contêiner. Um comando pode ser somente líder ou ter um `test`, mas não as duas coisas (`leader_only` tem precedência).

## ignoreErrors

(Opcional) Não falhar implantações se o comando de contêiner retornar um valor diferente de 0 (êxito). Defina como `true` para habilitar.

## waitAfterCompletion

(Opcional) Quantos segundos aguardar para executar o próximo comando depois que o comando é concluído. Se o sistema exigir reinicialização após a conclusão do comando, ele será reinicializado após o número de segundos especificado. Se o sistema for reinicializado como resultado de um comando, o Elastic Beanstalk recuperará até o ponto depois do comando no arquivo de configuração. O valor padrão é **60** segundos. Você também pode especificar **forever**, mas o sistema deve ser reinicializado antes de você executar outro comando.

## Exemplo

O exemplo a seguir salva a saída do comando `set` no arquivo especificado. O Elastic Beanstalk executa o comando em uma instância e reinicializa a instância logo após o término do comando.

```
container_commands:
 foo:
 command: set > c:\\myapp\\set.txt
 leader_only: true
 waitAfterCompletion: 0
```

## Adicionar e personalizar recursos do ambiente do Elastic Beanstalk

Você também pode personalizar seus recursos de ambiente que são parte do ambiente do Elastic Beanstalk. Por exemplo, é possível adicionar uma fila do Amazon SQS e um alarme sobre o

comprimento da fila, ou adicionar um cluster do Amazon ElastiCache. Você pode personalizar facilmente seu ambiente ao mesmo tempo que implanta a versão do aplicativo incluindo um arquivo de configuração com o pacote de origem.

Você pode usar a chave `Resources` em um [arquivo de configuração](#) para criar e personalizar recursos da AWS em seu ambiente. Os recursos definidos em arquivos de configuração são adicionados ao modelo do AWS CloudFormation usado para iniciar seu ambiente. Todos os AWS CloudFormation tipos de recursos [do](#) são suportados.

### Note

Sempre que você adicionar um recurso que não seja gerenciado pelo Elastic Beanstalk, certifique-se de adicionar uma política de usuário com as permissões apropriadas aos usuários do AWS Identity and Access Management (IAM). As [políticas de usuário gerenciadas](#) que o Elastic Beanstalk fornece apenas permissões de cobertura para recursos gerenciados pelo Elastic Beanstalk.

Por exemplo, o arquivo de configuração a seguir adiciona um gancho de ciclo de vida do Auto Scaling ao grupo de Auto Scaling padrão criado pelo Elastic Beanstalk:

### **~/my-app/.ebextensions/as-hook.config**

```
Resources:
 hookrole:
 Type: AWS::IAM::Role
 Properties:
 AssumeRolePolicyDocument: {
 "Version" : "2012-10-17",
 "Statement": [{
 "Effect": "Allow",
 "Principal": {
 "Service": ["autoscaling.amazonaws.com"]
 },
 "Action": ["sts:AssumeRole"]
 }]
 }
 Policies: [{
 "PolicyName": "SNS",
 "PolicyDocument": {
 "Version": "2012-10-17",
```

```

 "Statement": [{
 "Effect": "Allow",
 "Resource": "*",
 "Action": [
 "sqs:SendMessage",
 "sqs:GetQueueUrl",
 "sns:Publish"
]
 }
]
}
}]

```

hooktopic:

Type: AWS::SNS::Topic

Properties:

Subscription:

- Endpoint: *"my-email@example.com"*

Protocol: email

lifecyclehook:

Type: AWS::AutoScaling::LifecycleHook

Properties:

AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }

LifecycleTransition: autoscaling:EC2\_INSTANCE\_TERMINATING

NotificationTargetARN: { "Ref" : "hooktopic" }

RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn" ] }

Este exemplo define três recursos: hookrole, hooktopic e lifecyclehook. Os dois primeiros recursos são uma função do IAM, que concede ao Amazon EC2 Auto Scaling permissão para publicar mensagens no Amazon SNS e um tópico SNS, que retransmite mensagens do grupo de Auto Scaling para um endereço de e-mail. O Elastic Beanstalk cria esses recursos com as propriedades e os tipos especificados.

O recurso final, lifecyclehook, é o ciclo de vida do gancho propriamente dito:

lifecyclehook:

Type: AWS::AutoScaling::LifecycleHook

Properties:

AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }

LifecycleTransition: autoscaling:EC2\_INSTANCE\_TERMINATING

NotificationTargetARN: { "Ref" : "hooktopic" }

RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn" ] }

A definição de gancho do ciclo de vida usa duas [funções](#) para preencher valores para as propriedades do gancho. `{ "Ref" : "AWSEBAutoScalingGroup" }` recupera o nome do grupo de Auto Scaling criado pelo Elastic Beanstalk para o ambiente. `AWSEBAutoScalingGroup` é um dos [nomes de recurso](#) padrão fornecidos pelo Elastic Beanstalk.

Para [AWS::IAM::Role](#), `Ref` retorna somente o nome da função, não o nome de região da Amazon (ARN). Para obter o ARN para o parâmetro `RoleARN`, use outra função intrínseca, `Fn::GetAtt`, que pode obter qualquer atributo de um recurso. `RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn" ] }` obtém o atributo `Arn` do recurso `hookrole`.

`{ "Ref" : "hooktopic" }` obtém o ARN do tópico do Amazon SNS criado anteriormente no arquivo de configuração. O valor retornado por `Ref` varia por tipo de recurso e pode ser encontrado no [AWS CloudFormation tópico sobre o tipo de recurso AWS::SNS::Topic](#) [do](#) User Guide.

## Modificar os recursos criados pelo Elastic Beanstalk para o seu ambiente

Os recursos que o Elastic Beanstalk cria para o seu ambiente têm nomes. Você pode usar esses nomes para obter informações sobre os recursos com uma [função](#) ou modificar as propriedades nos recursos para personalizar seu comportamento. Este tópico descreve os recursos da AWS utilizados pelo Elastic Beanstalk nos diferentes tipos de ambientes.

### Note

O tópico anterior, [Recursos personalizados](#), fornece alguns casos de uso e exemplos para personalizar recursos de ambientes. Você também pode encontrar mais exemplos de arquivos de configuração no tópico posterior: [Exemplos de recursos personalizados](#).

Os ambientes de servidor da web tem os recursos a seguir.

### Ambientes de servidor da web

- `AWSEBAutoScalingGroup` ([AWS::AutoScaling::AutoScalingGroup](#)): o grupo de Auto Scaling anexado ao seu ambiente.
- Um dos dois recursos a seguir.
  - `AWSEBAutoScalingLaunchConfiguration` ([AWS::AutoScaling::LaunchConfiguration](#)): a configuração de execução anexada ao grupo de Auto Scaling do seu ambiente.
  - `AWSEBEC2LaunchTemplate` ([AWS::EC2::LaunchTemplate](#)): o modelo de execução do Amazon EC2 usado pelo grupo de Auto Scaling do seu ambiente.

**Note**

Se o seu ambiente utilizar funcionalidades que requerem modelos de execução do Amazon EC2, e a sua política de usuário não tiver as permissões necessárias, a criação ou atualização do ambiente poderá falhar. Use a [política de usuário gerenciada AdministratorAccess-AWSElasticBeanstalk](#) ou adicione as permissões necessárias à sua [política personalizada](#).

- `AWSEBEnvironmentName` ([AWS::ElasticBeanstalk::Environment](#)): o ambiente.
- `AWSEBSecurityGroup` ([AWS::EC2::SecurityGroup](#)): o grupo de segurança anexado ao seu grupo de Auto Scaling.
- `AWSEBRDSDatabase` ([AWS::RDS::DBInstance](#)): a instância de banco de dados do Amazon RDS associada ao ambiente (se aplicável).

Em um ambiente com balanceamento de carga, é possível acessar recursos adicionais relacionados ao load balancer. Os load balancers clássicos têm um recurso para o load balancer e outro para o grupo de segurança anexado a ele. Os load balancers de rede e de aplicativos têm recursos adicionais para listener padrão do load balancer para a regra do listener e para o grupo de destino.

#### Ambientes de carga equilibrada

- `AWSEBLoadBalancer` ([AWS::ElasticLoadBalancing::LoadBalancer](#)): o balanceador de carga clássico do ambiente.
- `AWSEBV2LoadBalancer` ([AWS::ElasticLoadBalancingV2::LoadBalancer](#)): o balanceador de carga de aplicação ou de rede do ambiente.
- `AWSEBLoadBalancerSecurityGroup` ([AWS::EC2::SecurityGroup](#)): apenas em uma [Amazon Virtual Private Cloud](#) (Amazon VPC) personalizada, o nome do grupo de segurança criado pelo Elastic Beanstalk para o balanceador de carga. Em uma VPC padrão ou no EC2 Classic, o Elastic Load Balancing atribui um grupo de segurança padrão ao balanceador de carga.
- `AWSEBV2LoadBalancerListener` ([AWS::ElasticLoadBalancingV2::Listener](#)): um listener que permite que o balanceador de carga verifique solicitações de conexão e as encaminhe para um ou mais grupos de destino.
- `AWSEBV2LoadBalancerListenerRule` ([AWS::ElasticLoadBalancingV2::ListenerRule](#)): define em quais solicitações um listener do Elastic Load Balancing age e a ação que ele executa.

- `AWSEBV2LoadBalancerTargetGroup` ([AWS::ElasticLoadBalancingV2::TargetGroup](#)): um grupo de destino do Elastic Load Balancing que encaminha solicitações para um ou mais destinos registrados, como instâncias do Amazon EC2.

Ambientes de operador têm recursos para a fila do SQS que armazena as solicitações de entrada, e uma tabela do Amazon DynamoDB que as instâncias usam para escolher o líder.

#### Ambientes de operador

- `AWSEBWorkerQueue` ([AWS::SQS::Queue](#)): a fila do Amazon SQS da qual o daemon extrai solicitações que precisam ser processadas.
- `AWSEBWorkerDeadLetterQueue` ([AWS::SQS::Queue](#)): a fila do Amazon SQS que armazena mensagens que não podem ser entregues ou que não foram processadas com êxito pelo daemon.
- `AWSEBWorkerCronLeaderRegistry` ([AWS::DynamoDB::Table](#)): a tabela do Amazon DynamoDB que é o registro interno usado pelo daemon para tarefas periódicas.

## Outras chaves de modelos de AWS CloudFormation

Nós já introduzimos chaves de arquivos de configuração do AWS CloudFormation como `Resources`, `files`, e `packages`. O Elastic Beanstalk adiciona o conteúdo de arquivos de configurações ao modelo do AWS CloudFormation que oferece suporte ao ambiente, para que você possa usar outras seções do AWS CloudFormation para realizar tarefas avançadas nos arquivos de configuração.

### Chaves

- [Parâmetros](#)
- [Outputs](#)
- [Mapeamentos](#)

### Parâmetros

Os parâmetros são uma alternativa para as [opções personalizadas](#) próprias do Elastic Beanstalk que podem ser usadas para definir valores utilizados em outros lugares nos arquivos de configuração. Do mesmo modo que as opções personalizadas, você pode usar parâmetros para reunir valores configuráveis pelo usuário em um único lugar. Diferente das opções personalizadas, não é possível

usar a API do Elastic Beanstalk para definir valores de parâmetros; o número de parâmetros que podem ser definidos em um modelo é limitado pelo AWS CloudFormation.

Um dos motivos pelos quais você pode querer usar parâmetros é duplicar seus arquivos de configuração como modelos do AWS CloudFormation. Se você usar parâmetros em vez de opções personalizadas, poderá usar o arquivo de configuração para criar o mesmo recurso no AWS CloudFormation como sua própria pilha. Por exemplo, você pode ter um arquivo de configuração que adiciona um sistema de arquivos do Amazon EFS ao seu ambiente para fins de teste, e depois usar o mesmo arquivo para criar um sistema de arquivos independentes que não estejam vinculados ao ciclo de vida do seu ambiente para ser usado em produção.

O exemplo a seguir mostra a utilização de parâmetros para reunir valores configuráveis pelo usuário no início de um arquivo de configuração.

Example [Loadbalancer-accesslogs-existingbucket.config](#): parâmetros

```
Parameters:
 bucket:
 Type: String
 Description: "Name of the Amazon S3 bucket in which to store load balancer logs"
 Default: "DOC-EXAMPLE-BUCKET"
 bucketprefix:
 Type: String
 Description: "Optional prefix. Can't start or end with a /, or contain the word
AWSLogs"
 Default: ""
```

## Outputs

Você pode usar um bloco de Outputs para exportar informações sobre os recursos criados para o AWS CloudFormation. Depois, você pode usar a função `Fn::ImportValue` para enviar o valor para um modelo do AWS CloudFormation do Elastic Beanstalk.

O exemplo a seguir cria um tópico do Amazon SNS e exporta o ARN para o AWS CloudFormation com o nome de `NotificationTopicArn`.

Example [sns-topic.config](#)

```
Resources:
 NotificationTopic:
```



```
Type: AWS::SNS::Topic
```

#### Outputs:

```
NotificationTopicArn:
 Description: Notification topic ARN
 Value: { "Ref" : "NotificationTopic" }
Export:
 Name: NotificationTopicArn
```

Em um arquivo de configuração para um ambiente diferente ou um modelo do AWS CloudFormation fora do Elastic Beanstalk, é possível usar a função `Fn::ImportValue` para obter o ARN exportado. Este exemplo atribui o valor exportado a uma propriedade do ambiente chamada `TOPIC_ARN`.

#### Example env.config

```
option_settings:
 aws:elasticbeanstalk:application:environment:
 TOPIC_ARN: ``{ "Fn::ImportValue" : "NotificationTopicArn" }``
```

#### Mapeamentos

Você pode usar um mapeamento para armazenar pares de chave-valor organizados por namespace. Um mapeamento pode ajudar você a organizar os valores que utiliza em toda as configurações, ou alterar o valor de um parâmetro dependendo de outro valor. Por exemplo, a configuração a seguir define o valor de um parâmetro de ID de conta com base na região atual.

Example [Loadbalancer-accesslogs-newbucket.config](#): mapeamentos

```
Mappings:
 Region2ELBAccountId:
 us-east-1:
 AccountId: "111122223333"
 us-west-2:
 AccountId: "444455556666"
 us-west-1:
 AccountId: "123456789012"
 eu-west-1:
 AccountId: "777788889999"
 ...
 Principal:
 AWS:
 ? "Fn::FindInMap"
```

```
 :
 - Region2ELBAccountId
 -
 Ref: "AWS::Region"
 - AccountId
```

## Funções

É possível usar funções nos arquivos de configuração para preencher valores de propriedades de recursos com informações de outros recursos ou das definições de opção de configuração do Elastic Beanstalk. O Elastic Beanstalk é compatível com funções do AWS CloudFormation (Ref, Fn::GetAtt, Fn::Join) e uma função específica do Elastic Beanstalk, Fn::GetOptionSetting.

### Funções

- [Ref.](#)
- [Fn::GetAtt](#)
- [Fn::Join](#)
- [Fn::GetOptionSetting](#)

### Ref.

Use Ref para recuperar a representação de string padrão de um recurso da AWS. O valor retornado por Ref depende do tipo de recurso e, às vezes, depende também de outros fatores. Por exemplo, um grupo de segurança ([AWS::EC2::SecurityGroup](#)) retorna o nome ou o ID do grupo de segurança, dependendo de o grupo de segurança estar em uma [Amazon Virtual Private Cloud](#) (Amazon VPC) padrão, um EC2 Classic ou uma VPC personalizada.

```
{ "Ref" : "resource name" }
```

#### Note

Para obter detalhes sobre cada tipo de recurso, incluindo os valores de retorno de Ref, consulte [Referência de tipos de recursos da AWS](#) no Guia do usuário do AWS CloudFormation.

A partir do [gancho de ciclo de vida do Auto Scaling](#) demonstrativo:

```
Resources:
 lifecyclehook:
 Type: AWS::AutoScaling::LifecycleHook
 Properties:
 AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
```

Você também pode usar o Ref para recuperar o valor de um parâmetro do AWS CloudFormation; definido em outro lugar no mesmo arquivo ou em um arquivo de configuração diferente.

## Fn::GetAtt

Use Fn::GetAtt para recuperar o valor de um atributo em um recurso da AWS.

```
{ "Fn::GetAtt" : ["resource name", "attribute name"] }
```

A partir do [gancho de ciclo de vida do Auto Scaling](#) demonstrativo:

```
Resources:
 lifecyclehook:
 Type: AWS::AutoScaling::LifecycleHook
 Properties:
 RoleARN: { "Fn::GetAtt" : ["hookrole", "Arn"] }
```

Consulte [Fn::GetAtt](#) para obter mais informações.

## Fn::Join

Use Fn::Join para combinar strings com um delimitador. As strings podem ser codificadas ou usar a saída de Fn::GetAtt ou Ref.

```
{ "Fn::Join" : ["delimiter", ["string1", "string2"]] }
```

Consulte [Fn::Join](#) para obter mais informações.

## Fn::GetOptionSetting

Use Fn::GetOptionSetting para recuperar o valor de uma definição de [opção de configuração](#) aplicada ao ambiente.

```
"Fn::GetOptionSetting":
```

```

Namespace: "namespace"
OptionName: "option name"
DefaultValue: "default value"

```

No exemplo de [armazenamento de chaves privadas](#):

```

Resources:
 AWSEBAutoScalingGroup:
 Metadata:
 AWS::CloudFormation::Authentication:
 S3Auth:
 type: "s3"
 buckets: ["elasticbeanstalk-us-west-2-123456789012"]
 roleName:
 "Fn::GetOptionSetting":
 Namespace: "aws:autoscaling:launchconfiguration"
 OptionName: "IamInstanceProfile"
 DefaultValue: "aws-elasticbeanstalk-ec2-role"

```

## Exemplos de recursos personalizados

Veja a seguir uma lista de arquivos de configuração demonstrativos que podem ser usados para personalizar os ambientes do Elastic Beanstalk:

- [DynamoDB, CloudWatch e SNS](#)
- [Elastic Load Balancing e CloudWatch](#)
- [ElastiCache](#)
- [RDS e CloudWatch](#)
- [SQS, SNS e CloudWatch](#)

Os subtópicos desta página fornecem alguns exemplos estendidos para adicionar e configurar recursos personalizados em um ambiente do Elastic Beanstalk.

### Exemplos

- [Exemplo: ElastiCache](#)
- [Exemplo: SQS, CloudWatch e SNS](#)
- [Exemplo: DynamoDB, CloudWatch e SNS](#)

## Exemplo: ElastiCache

Os exemplos a seguir adicionam um cluster do Amazon ElastiCache às plataformas EC2-Classic e EC2-VPC (padrão e personalizado [Amazon Virtual Private Cloud](#) (Amazon VPC)). Para obter mais informações sobre essas plataformas e saber como determinar quais delas são compatíveis com o EC2 para a sua região e conta da AWS, consulte <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-supported-platforms.html>. Depois, consulte a seção neste tópico que se aplica à sua plataforma.

- [Plataformas EC2-Classic](#)
- [EC2-VPC \(padrão\)](#)
- [EC2-VPC \(personalizado\)](#)

### Plataformas EC2-Classic

Este exemplo adiciona um cluster do Amazon ElastiCache a um ambiente com instâncias iniciadas na plataforma EC2-Classic. Todas as propriedades listadas neste exemplo são as propriedades mínimas necessárias que devem ser definidas para cada tipo de recurso. É possível fazer download do exemplo em [Exemplo do ElastiCache](#).

#### Note

Este exemplo cria recursos da AWS, pelos quais você poderia ser cobrado. Para obter mais informações sobre a definição de preços do AWS, consulte <https://aws.amazon.com/pricing/>. Alguns serviços fazem parte do nível de uso gratuito da AWS. Se for um cliente novo, você pode testar esses serviços gratuitamente. Consulte <https://aws.amazon.com/free/> para obter mais informações.

Para usar esse exemplo, faça o seguinte:

1. Crie um diretório [.ebextensions](#) no diretório de nível superior do pacote de origem.
2. Crie dois arquivos de configuração com a extensão `.config` e os coloque no diretório `.ebextensions`. Um arquivo de configuração define os recursos, e o outro arquivo de configuração define as opções.
3. Implante a aplicação no Elastic Beanstalk.

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Crie um arquivo de configuração (por exemplo, `elasticache.config`) que define os recursos. Neste exemplo, criamos o cluster do ElastiCache ao especificar o nome do recurso dele (`MyElastiCache`), declarar seu tipo e, em seguida, configurar as propriedades para o cluster. O exemplo faz referência ao nome do recurso de security group do ElastiCache, que é criado e definido neste arquivo de configuração. Depois, criamos um security group do ElastiCache. Definimos o nome deste recurso, declaramos seu tipo e adicionamos uma descrição para o security group. Por fim, vamos definir as regras de entrada para o security group do ElastiCache. Dessa forma, o acesso é permitido somente a partir de instâncias dentro do security group do ElastiCache (`MyCacheSecurityGroup`) e o security group do Elastic Beanstalk (`AWSEBSecurityGroup`). O nome do parâmetro, `AWSEBSecurityGroup`, é o nome do recurso fixo fornecido pelo Elastic Beanstalk. É necessário adicionar o `AWSEBSecurityGroup` às regras de entrada do grupo de segurança do ElastiCache para que a aplicação do Elastic Beanstalk se conecte às instâncias do cluster do ElastiCache.

```
#This sample requires you to create a separate configuration file that defines the
 custom option settings for CacheCluster properties.
```

```
Resources:
```

```
 MyElastiCache:
```

```
 Type: AWS::ElastiCache::CacheCluster
```

```
 Properties:
```

```
 CacheNodeType:
```

```
 Fn::GetOptionSetting:
```

```
 OptionName : CacheNodeType
```

```
 DefaultValue: cache.m1.small
```

```
 NumCacheNodes:
```

```
 Fn::GetOptionSetting:
```

```
 OptionName : NumCacheNodes
```

```
 DefaultValue: 1
```

```
 Engine:
```

```
 Fn::GetOptionSetting:
```

```
 OptionName : Engine
```

```
 DefaultValue: memcached
```

```
 CacheSecurityGroupNames:
```

```
 - Ref: MyCacheSecurityGroup
```

```
MyCacheSecurityGroup:
 Type: AWS::ElastiCache::SecurityGroup
 Properties:
 Description: "Lock cache down to webserver access only"
MyCacheSecurityGroupIngress:
 Type: AWS::ElastiCache::SecurityGroupIngress
 Properties:
 CacheSecurityGroupName:
 Ref: MyCacheSecurityGroup
 EC2SecurityGroupName:
 Ref: AWSEBSecurityGroup
```

Para obter mais informações sobre os recursos usados neste exemplo de arquivo de configuração, consulte as seguintes referências:

- [AWS::ElastiCache::CacheCluster](#)
- [AWS::ElastiCache::SecurityGroup](#)
- [AWS::ElastiCache::SecurityGroupIngress](#)

Crie um arquivo de configuração separado chamado `options.config` e defina as opções de configurações personalizadas.

```
option_settings:
 "aws:elasticbeanstalk:customoption":
 CacheNodeType : cache.m1.small
 NumCacheNodes : 1
 Engine : memcached
```

Essas linhas instruem o Elastic Beanstalk a obter os valores para as propriedades `CacheNodeType`, `NumCacheNodes` e `Engine`, a partir dos valores `CacheNodeType`, `NumCacheNodes` e `Engine`, em um arquivo de configuração (`options.config` em nosso exemplo). Esse arquivo contém uma seção de `option_settings` com outra de `aws:elasticbeanstalk: customoption` que tem um par de nome e valor com o valor real a ser usado. No exemplo acima, isso significa que `cache.m1.small`, `1` e `Memcached` seriam usados para os valores. Para obter mais informações sobre o `Fn::GetOptionSetting`, consulte [Funções](#).

## EC2-VPC (padrão)

Este exemplo adiciona um cluster do Amazon ElastiCache a um ambiente com instâncias iniciadas na plataforma EC2-VPC. A informação nessa seção se aplica especificamente a um cenário em

que o EC2 inicia instâncias na VPC padrão. Todas as propriedades neste exemplo são as mínimas necessárias que devem ser definidas para cada tipo de recurso. Para obter mais informações sobre as VPCs padrão, consulte [Sua VPC padrão e sub-redes](#).

### Note

Este exemplo cria recursos da AWS, pelos quais você poderia ser cobrado. Para obter mais informações sobre a definição de preços do AWS, consulte <https://aws.amazon.com/pricing/>. Alguns serviços fazem parte do nível de uso gratuito da AWS. Se for um cliente novo, você pode testar esses serviços gratuitamente. Consulte <https://aws.amazon.com/free/> para obter mais informações.

Para usar esse exemplo, faça o seguinte:

1. Crie um diretório `.ebextensions` no diretório de nível superior do pacote de origem.
2. Crie dois arquivos de configuração com a extensão `.config` e os coloque no diretório `.ebextensions`. Um arquivo de configuração define os recursos, e o outro arquivo de configuração define as opções.
3. Implante a aplicação no Elastic Beanstalk.

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Agora nomeie o arquivo de configuração de recursos `elasticache.config`. Para criar o cluster do ElastiCache, esse exemplo especifica o nome do recurso do cluster do ElastiCache (`MyElasticCache`), declara o tipo e então configura as propriedades do cluster. O exemplo faz referência ao ID do recurso do security group que criamos e definimos neste arquivo de configuração.

Depois, criamos um security group do EC2. Definimos o nome deste recurso, declaramos o tipo, adicionamos uma descrição e definimos as regras de entrada no security group para permitir o acesso somente a partir de instâncias dentro do security group do Elastic Beanstalk (`AWSEBSecurityGroup`). (O nome do parâmetro, `AWSEBSecurityGroup`, é um nome de recurso fixo fornecido pelo Elastic Beanstalk. É necessário adicionar o `AWSEBSecurityGroup` às regras



de ingresso do grupo de segurança do ElastiCache para que a aplicação do Elastic Beanstalk se conecte às instâncias do cluster do ElastiCache.)

As regras de entrada para o security group do EC2 também definem o protocolo de IP e os números de portas em que os nós de cache podem aceitar conexões. Para Redis, o número padrão da porta é 6379.

```
#This sample requires you to create a separate configuration file that defines the
 custom option settings for CacheCluster properties.
```

```
Resources:
```

```
 MyCacheSecurityGroup:
```

```
 Type: "AWS::EC2::SecurityGroup"
```

```
 Properties:
```

```
 GroupDescription: "Lock cache down to webserver access only"
```

```
 SecurityGroupIngress :
```

```
 - IpProtocol : "tcp"
```

```
 FromPort :
```

```
 Fn::GetOptionSetting:
```

```
 OptionName : "CachePort"
```

```
 DefaultValue: "6379"
```

```
 ToPort :
```

```
 Fn::GetOptionSetting:
```

```
 OptionName : "CachePort"
```

```
 DefaultValue: "6379"
```

```
 SourceSecurityGroupName:
```

```
 Ref: "AWSEBSecurityGroup"
```

```
 MyElastiCache:
```

```
 Type: "AWS::ElastiCache::CacheCluster"
```

```
 Properties:
```

```
 CacheNodeType:
```

```
 Fn::GetOptionSetting:
```

```
 OptionName : "CacheNodeType"
```

```
 DefaultValue : "cache.t2.micro"
```

```
 NumCacheNodes:
```

```
 Fn::GetOptionSetting:
```

```
 OptionName : "NumCacheNodes"
```

```
 DefaultValue : "1"
```

```
 Engine:
```

```
 Fn::GetOptionSetting:
```

```
 OptionName : "Engine"
```

```
 DefaultValue : "redis"
```

```
 VpcSecurityGroupIds:
```

```
-
 Fn::GetAtt:
 - MyCacheSecurityGroup
 - GroupId

Outputs:
 ElastiCache:
 Description : "ID of ElastiCache Cache Cluster with Redis Engine"
 Value :
 Ref : "MyElastiCache"
```

Para obter mais informações sobre os recursos usados neste exemplo de arquivo de configuração, consulte as seguintes referências:

- [AWS::ElastiCache::CacheCluster](#)
- [AWS::EC2::SecurityGroup](#)

Depois, nomeie o arquivo de configuração de opções `options.config` e defina as opções de configurações personalizadas.

```
option_settings:
 "aws:elasticbeanstalk:customoption":
 CacheNodeType : cache.t2.micro
 NumCacheNodes : 1
 Engine : redis
 CachePort : 6379
```

Essas linhas instruem o Elastic Beanstalk para obter os valores para as propriedades `CacheNodeType`, `NumCacheNodes`, `Engine` e `CachePort` dos valores `CacheNodeType`, `NumCacheNodes`, `Engine` e `CachePort` em um arquivo de configuração (`options.config` em nosso exemplo). Esse arquivo inclui uma seção de `aws:elasticbeanstalk:customoption` (em `option_settings`) que contém pares de nome e valor com os valores reais para serem usados. No exemplo anterior, `cache.t2.micro`, `1`, `redis` e `6379` seriam usados para os valores. Para obter mais informações sobre o `Fn::GetOptionSetting`, consulte [Funções](#).

## EC2-VPC (personalizado)

Se você criar uma VPC personalizada na plataforma EC2-VPC e especificá-la como o local onde o EC2 inicia as instâncias, o processo de adicionar um cluster do Amazon ElastiCache ao seu ambiente será diferente do de uma VPC padrão. A principal diferença é que você precisa criar um

grupo de sub-rede para o cluster do ElastiCache. Todas as propriedades neste exemplo são as mínimas necessárias que devem ser definidas para cada tipo de recurso.

### Note

Este exemplo cria recursos da AWS, pelos quais você poderia ser cobrado. Para obter mais informações sobre a definição de preços do AWS, consulte <https://aws.amazon.com/pricing/>. Alguns serviços fazem parte do nível de uso gratuito da AWS. Se for um cliente novo, você pode testar esses serviços gratuitamente. Consulte <https://aws.amazon.com/free/> para obter mais informações.

Para usar esse exemplo, faça o seguinte:

1. Crie um diretório `.ebextensions` no diretório de nível superior do pacote de origem.
2. Crie dois arquivos de configuração com a extensão `.config` e os coloque no diretório `.ebextensions`. Um arquivo de configuração define os recursos, e o outro arquivo de configuração define as opções.
3. Implante a aplicação no Elastic Beanstalk.

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Agora nomeie o arquivo de configuração de recursos `elasticache.config`. Para criar o cluster do ElastiCache, esse exemplo especifica o nome do recurso do cluster do ElastiCache (`MyElasticache`), declara o tipo e então configura as propriedades do cluster. As propriedades no exemplo fazem referência ao nome do grupo de sub-rede para o cluster do ElastiCache, bem como para o ID do recurso do security group que criamos e definimos neste arquivo de configuração.

Depois, criamos um security group do EC2. Definimos o nome deste recurso, declaramos o tipo, adicionamos uma descrição, o ID da VPC, e definimos as regras de entrada no security group para permitir o acesso somente a partir das instâncias dentro do security group do Elastic Beanstalk (`AWSEBSecurityGroup`). (O nome do parâmetro, `AWSEBSecurityGroup`, é um nome de recurso fixo fornecido pelo Elastic Beanstalk. É necessário adicionar o `AWSEBSecurityGroup` às regras de ingresso do grupo de segurança do ElastiCache para que a aplicação do Elastic Beanstalk se conecte às instâncias do cluster do ElastiCache.)

As regras de entrada para o security group do EC2 também definem o protocolo de IP e os números de portas em que os nós de cache podem aceitar conexões. Para Redis, o número padrão da porta é 6379. Por fim, esse exemplo cria um grupo de sub-rede para o cluster do ElastiCache. Definimos o nome desse recurso, declaramos seu tipo e adicionamos uma descrição e um ID da sub-rede no grupo de sub-rede.

### Note

Recomendamos que você use sub-redes privadas para o cluster do ElastiCache. Para obter mais informações sobre uma VPC com uma sub-rede privada, consulte [https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_Scenario2.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Scenario2.html).

```
#This sample requires you to create a separate configuration file that defines the
 custom option settings for CacheCluster properties.
```

#### Resources:

##### MyElastiCache:

```
Type: "AWS::ElastiCache::CacheCluster"
```

##### Properties:

###### CacheNodeType:

###### Fn::GetOptionSetting:

```
OptionName : "CacheNodeType"
```

```
DefaultValue : "cache.t2.micro"
```

###### NumCacheNodes:

###### Fn::GetOptionSetting:

```
OptionName : "NumCacheNodes"
```

```
DefaultValue : "1"
```

###### Engine:

###### Fn::GetOptionSetting:

```
OptionName : "Engine"
```

```
DefaultValue : "redis"
```

###### CacheSubnetGroupName:

```
Ref: "MyCacheSubnets"
```

###### VpcSecurityGroupIds:

```
- Ref: "MyCacheSecurityGroup"
```

##### MyCacheSecurityGroup:

```
Type: "AWS::EC2::SecurityGroup"
```

##### Properties:

```
GroupDescription: "Lock cache down to webserver access only"
```

```
VpcId:
```

```
 Fn::GetOptionSetting:
 OptionName : "VpcId"
SecurityGroupIngress :
 - IpProtocol : "tcp"
 FromPort :
 Fn::GetOptionSetting:
 OptionName : "CachePort"
 DefaultValue: "6379"
 ToPort :
 Fn::GetOptionSetting:
 OptionName : "CachePort"
 DefaultValue: "6379"
 SourceSecurityGroupId:
 Ref: "AWSEBSecurityGroup"
MyCacheSubnets:
 Type: "AWS::ElastiCache::SubnetGroup"
 Properties:
 Description: "Subnets for ElastiCache"
 SubnetIds:
 Fn::GetOptionSetting:
 OptionName : "CacheSubnets"
Outputs:
 ElastiCache:
 Description : "ID of ElastiCache Cache Cluster with Redis Engine"
 Value :
 Ref : "MyElastiCache"
```

Para obter mais informações sobre os recursos usados neste exemplo de arquivo de configuração, consulte as seguintes referências:

- [AWS::ElastiCache::CacheCluster](#)
- [AWS::EC2::SecurityGroup](#)
- [AWS::ElastiCache::SubnetGroup](#)

Depois, nomeie o arquivo de configuração de opções `options.config` e defina as opções de configurações personalizadas.

**Note**

No exemplo a seguir, substitua o exemplo `CacheSubnets` e os valores de `VpcId` com suas próprias sub-redes e VPC.

```
option_settings:
 "aws:elasticbeanstalk:customoption":
 CacheNodeType : cache.t2.micro
 NumCacheNodes : 1
 Engine : redis
 CachePort : 6379
 CacheSubnets:
 - subnet-1a1a1a1a
 - subnet-2b2b2b2b
 - subnet-3c3c3c3c
 VpcId: vpc-4d4d4d4d
```

Essas linhas instruem o Elastic Beanstalk para obter os valores para as propriedades `CacheNodeType`, `NumCacheNodes`, `Engine`, `CachePort`, `CacheSubnets` e `VpcId` dos valores `CacheNodeType`, `NumCacheNodes`, `Engine`, `CachePort`, `CacheSubnets` e `VpcId` em um arquivo de configuração (`options.config` em nosso exemplo). Esse arquivo inclui uma seção de `aws:elasticbeanstalk:customoption` (em `option_settings`) que contém pares de nome e valor com os valores de exemplo. No exemplo acima, `cache.t2.micro`, `1`, `redis`, `6379`, `subnet-1a1a1a1a`, `subnet-2b2b2b2b`, `subnet-3c3c3c3c` e `vpc-4d4d4d4d` seriam usados para os valores. Para obter mais informações sobre o `Fn::GetOptionSetting`, consulte [Funções](#).

Exemplo: SQS, CloudWatch e SNS

Este exemplo adiciona uma fila do Amazon SQS e um alarme no comprimento da fila para o ambiente. As propriedades que você vê neste exemplo são as mínimas necessárias a serem definidas para cada um desses recursos. É possível fazer download do exemplo em [SQS, SNS e CloudWatch](#).

**Note**

Este exemplo cria recursos da AWS, pelos quais você poderia ser cobrado. Para obter mais informações sobre a definição de preços do AWS, consulte <https://aws.amazon.com/pricing/>. Alguns serviços fazem parte do nível de uso gratuito da AWS. Se for um cliente novo, você

pode testar esses serviços gratuitamente. Consulte <https://aws.amazon.com/free/> para obter mais informações.

Para usar esse exemplo, faça o seguinte:

1. Crie um diretório `.ebextensions` no diretório de nível superior do pacote de origem.
2. Crie dois arquivos de configuração com a extensão `.config` e os coloque no diretório `.ebextensions`. Um arquivo de configuração define os recursos, e o outro arquivo de configuração define as opções.
3. Implante a aplicação no Elastic Beanstalk.

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Crie um arquivo de configuração (por exemplo, `sqs.config`) que define os recursos. Neste exemplo, criamos uma fila SQS e definimos a propriedade `VisibilityTimeout` no recurso `MySQSQueue`. Em seguida, criamos um `Topic SNS` e especificamos que o e-mail seja enviado para `someone@example.com` quando o alarme for acionado. Por fim, criamos um alarme do `CloudWatch` se a fila crescer além de 10 mensagens. Na propriedade `Dimensions`, especificamos o nome da dimensão e o valor que representa a medida da dimensão. Usamos `Fn::GetAtt` para retornar o valor de `QueueName` do `MySQSQueue`.

```
#This sample requires you to create a separate configuration file to define the custom
options for the SNS topic and SQS queue.
Resources:
 MySQSQueue:
 Type: AWS::SQS::Queue
 Properties:
 VisibilityTimeout:
 Fn::GetOptionSetting:
 OptionName: VisibilityTimeout
 DefaultValue: 30
 AlarmTopic:
 Type: AWS::SNS::Topic
 Properties:
 Subscription:
 - Endpoint:
```

```

 Fn::GetOptionSetting:
 OptionName: AlarmEmail
 DefaultValue: "nobody@amazon.com"
 Protocol: email
QueueDepthAlarm:
 Type: AWS::CloudWatch::Alarm
 Properties:
 AlarmDescription: "Alarm if queue depth grows beyond 10 messages"
 Namespace: "AWS/SQS"
 MetricName: ApproximateNumberOfMessagesVisible
 Dimensions:
 - Name: QueueName
 Value : { "Fn::GetAtt" : ["MySQSQueue", "QueueName"] }
 Statistic: Sum
 Period: 300
 EvaluationPeriods: 1
 Threshold: 10
 ComparisonOperator: GreaterThanThreshold
 AlarmActions:
 - Ref: AlarmTopic
 InsufficientDataActions:
 - Ref: AlarmTopic

Outputs :
 QueueURL:
 Description : "URL of newly created SQS Queue"
 Value : { Ref : "MySQSQueue" }
 QueueARN :
 Description : "ARN of newly created SQS Queue"
 Value : { "Fn::GetAtt" : ["MySQSQueue", "Arn"] }
 QueueName :
 Description : "Name newly created SQS Queue"
 Value : { "Fn::GetAtt" : ["MySQSQueue", "QueueName"] }

```

Para obter mais informações sobre os recursos usados neste exemplo de arquivo de configuração, consulte as seguintes referências:

- [AWS::SQS::Queue](#)
- [AWS::SNS::Topic](#)
- [AWS::CloudWatch::Alarm](#)



Crie um arquivo de configuração separado chamado `options.config` e defina as opções de configurações personalizadas.

```
option_settings:
 "aws:elasticbeanstalk:customoption":
 VisibilityTimeout : 30
 AlarmEmail : "nobody@example.com"
```

Essas linhas instruem o Elastic Beanstalk a obter os valores para as propriedades `VisibilityTimeout` (Tempo limite de visibilidade) e `Subscription Endpoint` (Endpoint de assinatura) a partir dos valores de `VisibilityTimeout` (Tempo limite de visibilidade) e `Subscription Endpoint` (Endpoint de assinatura) em um arquivo de configuração (`options.config` em nosso exemplo) que contém uma seção de `option_settings` com uma seção `aws:elasticbeanstalk:customoption` contendo um par de nome/valor que contém o valor real a ser usado. No exemplo acima, isso significa que `30` e `"nobody@amazon.com"` seriam usados para os valores. Para obter mais informações sobre o `Fn::GetOptionSetting`, consulte [the section called "Funções"](#).

Exemplo: DynamoDB, CloudWatch e SNS

Este arquivo de configuração define a tabela do DynamoDB como um manipulador de sessão para uma aplicação baseada em PHP usando o AWS SDK for PHP 2. Para usar este exemplo, você deve ter um perfil de instância do IAM, que é adicionado às instâncias em seu ambiente e usado para acessar a tabela do DynamoDB.

Você poderá fazer download do exemplo que usaremos nesta etapa em [Suporte de sessão do DynamoDB demonstrativo](#). A amostra contém os seguintes arquivos:

- O aplicativo de exemplo, `index.php`
- Um arquivo de configuração, `dynamodb.config`, para criar e configurar uma tabela do DynamoDB e outros recursos da AWS e instalar software nas instâncias do EC2 que hospedam a aplicação em um ambiente do Elastic Beanstalk
- Um arquivo de configuração, `options.config`, que substitui os valores padrão em `dynamodb.config` por configurações específicas para essa determinada instalação

## **index.php**

```
<?php

// Include the SDK using the Composer autoloader
```

```

require '../vendor/autoload.php';

use Aws\DynamoDb\DynamoDbClient;

// Grab the session table name and region from the configuration file
list($tableName, $region) = file(__DIR__ . '/../sessiontable');
$tableName = rtrim($tableName);
$region = rtrim($region);

// Create a DynamoDB client and register the table as the session handler
$dynamodb = DynamoDbClient::factory(array('region' => $region));
$handler = $dynamodb->registerSessionHandler(array('table_name' => $tableName,
 'hash_key' => 'username'));

// Grab the instance ID so we can display the EC2 instance that services the request
$instanceId = file_get_contents("http://169.254.169.254/latest/meta-data/instance-id");
?>
<h1>Elastic Beanstalk PHP Sessions Sample</h1>
<p>This sample application shows the integration of the Elastic Beanstalk PHP
container and the session support for DynamoDB from the AWS SDK for PHP 2.
Using DynamoDB session support, the application can be scaled out across
multiple web servers. For more details, see the
PHP Developer Center.</p>

<form id="SimpleForm" name="SimpleForm" method="post" action="index.php">
<?php
echo 'Request serviced from instance ' . $instanceId . '
';
echo '
';

if (isset($_POST['continue'])) {
 session_start();
 $_SESSION['visits'] = $_SESSION['visits'] + 1;
 echo 'Welcome back ' . $_SESSION['username'] . '
';
 echo 'This is visit number ' . $_SESSION['visits'] . '
';
 session_write_close();
 echo '
';
 echo '<input type="Submit" value="Refresh" name="continue" id="continue"/>';
 echo '<input type="Submit" value="Delete Session" name="killsession"
id="killsession"/>';
} elseif (isset($_POST['killsession'])) {
 session_start();
 echo 'Goodbye ' . $_SESSION['username'] . '
';
 session_destroy();
 echo 'Username: <input type="text" name="username" id="username" size="30"/>
';

```

```

echo '
';
echo '<input type="Submit" value="New Session" name="newsession" id="newsession"/>';
} elseif (isset($_POST['newsession'])) {
 session_start();
 $_SESSION['username'] = $_POST['username'];
 $_SESSION['visits'] = 1;
 echo 'Welcome to a new session ' . $_SESSION['username'] . '
';
 session_write_close();
 echo '
';
 echo '<input type="Submit" value="Refresh" name="continue" id="continue"/>';
 echo '<input type="Submit" value="Delete Session" name="killsession"
id="killsession"/>';
} else {
 echo 'To get started, enter a username.
';
 echo '
';
 echo 'Username: <input type="text" name="username" id="username" size="30"/>
';
 echo '<input type="Submit" value="New Session" name="newsession" id="newsession"/>';
}
?>
</form>

```

## .ebextensions/dynamodb.config

### Resources:

#### SessionTable:

Type: AWS::DynamoDB::Table

#### Properties:

##### KeySchema:

##### HashKeyElement:

##### AttributeName:

Fn::GetOptionSetting:

OptionName : SessionHashKeyName

DefaultValue: "username"

##### AttributeType:

Fn::GetOptionSetting:

OptionName : SessionHashKeyType

DefaultValue: "S"

##### ProvisionedThroughput:

##### ReadCapacityUnits:

Fn::GetOptionSetting:

OptionName : SessionReadCapacityUnits

DefaultValue: 1

##### WriteCapacityUnits:

```
 Fn::GetOptionSetting:
 OptionName : SessionWriteCapacityUnits
 DefaultValue: 1

SessionWriteCapacityUnitsLimit:
 Type: AWS::CloudWatch::Alarm
 Properties:
 AlarmDescription: { "Fn::Join" : ["", [{ "Ref" : "AWSEBEnvironmentName" }], "
write capacity limit on the session table."]] }
 Namespace: "AWS/DynamoDB"
 MetricName: ConsumedWriteCapacityUnits
 Dimensions:
 - Name: TableName
 Value: { "Ref" : "SessionTable" }
 Statistic: Sum
 Period: 300
 EvaluationPeriods: 12
 Threshold:
 Fn::GetOptionSetting:
 OptionName : SessionWriteCapacityUnitsAlarmThreshold
 DefaultValue: 240
 ComparisonOperator: GreaterThanThreshold
 AlarmActions:
 - Ref: SessionAlarmTopic
 InsufficientDataActions:
 - Ref: SessionAlarmTopic

SessionReadCapacityUnitsLimit:
 Type: AWS::CloudWatch::Alarm
 Properties:
 AlarmDescription: { "Fn::Join" : ["", [{ "Ref" : "AWSEBEnvironmentName" }], " read
capacity limit on the session table."]] }
 Namespace: "AWS/DynamoDB"
 MetricName: ConsumedReadCapacityUnits
 Dimensions:
 - Name: TableName
 Value: { "Ref" : "SessionTable" }
 Statistic: Sum
 Period: 300
 EvaluationPeriods: 12
 Threshold:
 Fn::GetOptionSetting:
 OptionName : SessionReadCapacityUnitsAlarmThreshold
 DefaultValue: 240
```

```

 ComparisonOperator: GreaterThanThreshold
 AlarmActions:
 - Ref: SessionAlarmTopic
 InsufficientDataActions:
 - Ref: SessionAlarmTopic

SessionThrottledRequestsAlarm:
 Type: AWS::CloudWatch::Alarm
 Properties:
 AlarmDescription: { "Fn::Join" : ["", [{ "Ref" : "AWSEBEnvironmentName" }, " :
requests are being throttled."]] }
 Namespace: AWS/DynamoDB
 MetricName: ThrottledRequests
 Dimensions:
 - Name: TableName
 Value: { "Ref" : "SessionTable" }
 Statistic: Sum
 Period: 300
 EvaluationPeriods: 1
 Threshold:
 Fn::GetOptionSetting:
 OptionName: SessionThrottledRequestsThreshold
 DefaultValue: 1
 ComparisonOperator: GreaterThanThreshold
 AlarmActions:
 - Ref: SessionAlarmTopic
 InsufficientDataActions:
 - Ref: SessionAlarmTopic

SessionAlarmTopic:
 Type: AWS::SNS::Topic
 Properties:
 Subscription:
 - Endpoint:
 Fn::GetOptionSetting:
 OptionName: SessionAlarmEmail
 DefaultValue: "nobody@amazon.com"
 Protocol: email

files:
 "/var/app/sessiontable":
 mode: "000444"
 content: |
 `{"Ref" : "SessionTable"}`

```

```

 {"Ref" : "AWS::Region"}`

"/var/app/composer.json":
 mode: "000744"
 content:
 {
 "require": {
 "aws/aws-sdk-php": "*"
 }
 }

container_commands:
 "1-install-composer":
 command: "cd /var/app; curl -s http://getcomposer.org/installer | php"
 "2-install-dependencies":
 command: "cd /var/app; php composer.phar install"
 "3-cleanup-composer":
 command: "rm -Rf /var/app/composer.*"

```

No arquivo de configuração de exemplo, primeiro crie a tabela do DynamoDB e configure a chave primária da tabela e as unidades de capacidade para alocar recursos suficientes para fornecer a taxa de transferência solicitada. Em seguida, criamos alarmes do CloudWatch para WriteCapacity e ReadCapacity. Criamos um tópico do SNS que envia um e-mail para "nobody@amazon.com" caso os limites de alarme sejam violados.

Depois de criar e configurar os recursos da AWS para o ambiente, precisamos personalizar as instâncias do EC2. Usamos a chave `files` para passar os detalhes da tabela do DynamoDB para as instâncias do EC2 do ambiente e para adicionar uma "exigência" ao arquivo `composer.json` para o AWS SDK for PHP 2. Por fim, executamos comandos de contêiner para instalar o Composer, as dependências necessárias e, em seguida, remover o instalador.

## **.ebextensions/options.config**

```

option_settings:
 "aws:elasticbeanstalk:customoption":
 SessionHashKeyName : username
 SessionHashKeyType : S
 SessionReadCapacityUnits : 1
 SessionReadCapacityUnitsAlarmThreshold : 240
 SessionWriteCapacityUnits : 1
 SessionWriteCapacityUnitsAlarmThreshold : 240
 SessionThrottledRequestsThreshold : 1

```

`SessionAlarmEmail``: me@example.com`

Substitua o valor `SessionAlarmEmail` pelo e-mail para o qual deseja que as notificações de alarme sejam enviadas. O arquivo `options.config` contém os valores usados para algumas das variáveis definidas em `dynamodb.config`. Por exemplo, `dynamodb.config` contém as seguintes linhas:

```
Subscription:
- Endpoint:
 Fn::GetOptionSetting:
 OptionName: SessionAlarmEmail
 DefaultValue: "nobody@amazon.com"
```

Essas linhas dizem ao Elastic Beanstalk para obter o valor da propriedade `Endpoint` do valor de `SessionAlarmEmail` em um arquivo de configuração (`options.config` no exemplo de aplicação) que contém uma seção `option_settings` com uma seção `aws:elasticbeanstalk:customoption` que contém um par nome/valor que contém o valor real a ser usado. No exemplo acima, isso significa que `SessionAlarmEmail` será atribuído o valor `nobody@amazon.com`.

Para obter mais informações sobre os recursos do CloudFormation usados neste exemplo, consulte as seguintes referências:

- [AWS::DynamoDB::Table](#)
- [AWS::CloudWatch::Alarm](#)
- [AWS::SNS::Topic](#)

## Usar configurações salvas do Elastic Beanstalk

Você pode salvar a configuração de seu ambiente como um objeto no Amazon Simple Storage Service (Amazon S3) para aplicar a outros ambientes durante a criação ou para aplicar a um ambiente em execução. Configurações salvas são modelos formatados em YAML que definem a [versão da plataforma](#), o [nível](#), as definições das [opções de configuração](#) e as tags de um ambiente.

Você pode aplicar tags a uma configuração salva ao criá-la e editar tags de configurações salvas existentes. As tags aplicadas a uma configuração salva não estão relacionadas às tags especificadas em uma configuração salva que usa a chave `Tags`:. Os últimos são aplicados a um ambiente quando você aplica a configuração salva ao ambiente. Para obter mais detalhes, consulte [Marcação de configurações salvas](#).

**Note**

Você pode criar e aplicar configurações salvas em seus ambientes do Elastic Beanstalk usando vários métodos. Isso inclui o console do Elastic Beanstalk, a CLI do EB e o AWS CLI. Consulte os seguintes tópicos para ver exemplos de métodos alternativos de criação e aplicação de configurações salvas:

- [Definir as opções de configuração antes da criação do ambiente](#)
- [Definir as opções de configuração durante a criação do ambiente](#)
- [Definir as opções de configuração depois da criação do ambiente](#)

Crie uma configuração salva com base no estado atual do ambiente no console de gerenciamento do Elastic Beanstalk.

Para salvar a configuração de um ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

**Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Save Configuration (Salvar configuração).
4. Use o formulário na tela para nomear a configuração salva. Opcionalmente, forneça uma breve descrição e adicione chaves e valores de tag.
5. Escolha Save (Salvar).



Elastic Beanstalk > Environments > GettingStartedApp-env

## Save Configuration

Save this environment's current configuration.

Environment:  
GettingStartedApp-env

Configuration name:

Description:

### Tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

Key	Value	
<input type="text" value="mytag1"/>	<input type="text" value="value1"/>	<input type="button" value="Remove tag"/>

49 remaining


A configuração salva inclui todas as configurações aplicadas ao ambiente com o console ou qualquer outro cliente que usa a API do Elastic Beanstalk. Em seguida, você pode aplicar a configuração salva ao ambiente em uma data futura para restaurá-lo ao seu estado anterior ou aplicá-la a um novo ambiente durante a [criação dele](#).

Você pode fazer download de uma configuração usando o comando [the section called “eb config”](#) da CLI do EB, como mostrado no exemplo a seguir. *NAME* é o nome da configuração salva.

```
eb config get NAME
```

Como aplicar uma configuração salva durante a criação do ambiente (console do Elastic Beanstalk)

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

 Note

Se você tiver muitos aplicativos, use a barra de pesquisa para filtrar a lista de aplicativos.

3. No painel de navegação, encontre o nome do aplicativo e escolha Saved configurations (Configurações salvas).
4. Selecione a configuração salva que deseja aplicar e escolha Launch environment (Iniciar ambiente).
5. Continue no assistente para criar seu ambiente.

As configurações salvas não incluem as definições aplicadas com os [arquivos de configuração](#) no código-fonte de seu aplicativo. Se a mesma configuração for aplicada tanto em um arquivo de configuração quanto em uma configuração salva, a da configuração salva terá precedência. Da mesma forma, as opções especificadas no console do Elastic Beanstalk substituem as das configurações salvas. Para obter mais informações, consulte [Precedência](#).

As configurações salvas são armazenadas no bucket do S3 do Elastic Beanstalk em uma pasta com o mesmo nome da sua aplicação. Por exemplo, as configurações de uma aplicação chamada my-app na região us-west-2 da conta número 123456789012 podem ser encontradas em `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app/`.

Visualize o conteúdo de uma configuração salva abrindo-a em um editor de texto. O exemplo a seguir mostra a configuração de um ambiente de servidor Web iniciado com o console de gerenciamento do Elastic Beanstalk.

```
EnvironmentConfigurationMetadata:
```

```
Description: Saved configuration from a multicontainer Docker environment created
with the Elastic Beanstalk Management Console
DateCreated: '1520633151000'
DateModified: '1520633151000'
Platform:
PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit
Amazon Linux/2.5.0
OptionSettings:
aws:elasticbeanstalk:command:
 BatchSize: '30'
 BatchSizeType: Percentage
aws:elasticbeanstalk:sns:topics:
 Notification Endpoint: me@example.com
aws:elb:policies:
 ConnectionDrainingEnabled: true
 ConnectionDrainingTimeout: '20'
aws:elb:loadbalancer:
 CrossZone: true
aws:elasticbeanstalk:environment:
 ServiceRole: aws-elasticbeanstalk-service-role
aws:elasticbeanstalk:application:
 Application Healthcheck URL: /
aws:elasticbeanstalk:healthreporting:system:
 SystemType: enhanced
aws:autoscaling:launchconfiguration:
 IamInstanceProfile: aws-elasticbeanstalk-ec2-role
 InstanceType: t2.micro
 EC2KeyName: workstation-uswest2
aws:autoscaling:updatepolicy:rollingupdate:
 RollingUpdateType: Health
 RollingUpdateEnabled: true
EnvironmentTier:
 Type: Standard
 Name: WebServer
AWSConfigurationTemplateVersion: 1.1.0.0
Tags:
 Cost Center: WebApp Dev
```

Você pode modificar o conteúdo de uma configuração salva e salvá-lo no mesmo local no Amazon S3. Qualquer configuração salva formatada adequadamente armazenada no local certo pode ser aplicada a um ambiente usando o console de gerenciamento do Elastic Beanstalk.

As seguintes chaves são compatíveis.

- **AWSConfigurationTemplateVersion** (obrigatória): a versão do modelo de configuração (1.1.0.0).

```
AWSConfigurationTemplateVersion: 1.1.0.0
```

- **Plataforma**: o nome de recurso da Amazon (ARN) da versão da plataforma do ambiente. Você pode especificar a plataforma pelo ARN ou pelo nome da pilha de soluções.

```
Platform:
```

```
PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit Amazon Linux/2.5.0
```

- **SolutionStack**: o nome completo da [pilha de soluções](#) usada para criar o ambiente.

```
SolutionStack: 64bit Amazon Linux 2017.03 v2.5.0 running Java 8
```

- **OptionSettings**: definições da [opção de Configuração \(Configuração\)](#) para aplicar ao ambiente. Por exemplo, a seguinte entrada define o tipo de instância como t2.micro.

```
OptionSettings:
```

```
aws:autoscaling:launchconfiguration:
InstanceType: t2.micro
```

- **Tags**: até 47 tags a serem aplicadas aos recursos criados no ambiente.

```
Tags:
```

```
Cost Center: WebApp Dev
```

- **EnvironmentTier**: o tipo de ambiente a ser criado. Para um ambiente de servidor Web, você pode excluir esta seção (servidor Web é o padrão). Para um ambiente de operador, use o seguinte.

```
EnvironmentTier:
```

```
Name: Worker
Type: SQS/HTTP
```

### Note

Você pode criar e aplicar configurações salvas em seus ambientes do Elastic Beanstalk usando vários métodos. Isso inclui o console do Elastic Beanstalk, a CLI do EB e o AWS CLI. Consulte os seguintes tópicos para ver exemplos de métodos alternativos de criação e aplicação de configurações salvas:

- [Definir as opções de configuração antes da criação do ambiente](#)
- [Definir as opções de configuração durante a criação do ambiente](#)
- [Definir as opções de configuração depois da criação do ambiente](#)

## Marcação de configurações salvas

Você pode aplicar tags às suas configurações salvas do AWS Elastic Beanstalk. As tags são pares de chave-valor associados a recursos da AWS. Para obter informações sobre a atribuição de tags do recurso do Elastic Beanstalk, casos de uso, restrições de chave e valor de tag, além de tipos de recursos compatíveis, consulte [Marcar recursos da aplicação do Elastic Beanstalk](#).

Você pode especificar tags quando criar uma configuração salva. Em uma configuração salva existente, você pode adicionar ou remover tags e atualizar os valores de tags existentes. Você pode adicionar até 50 tags para cada configuração salva.

### Adicionar tags durante a criação de configurações salvas

Ao usar o console do Elastic Beanstalk para [salvar uma configuração](#), é possível especificar chaves e valores de tags na página Save Configuration (Salvar configuração).

Se você usar a CLI do EB para salvar uma configuração, use a opção `--tags` com [eb config](#) para adicionar tags.

```
~/workspace/my-app$ eb config --tags mytag1=value1,mytag2=value2
```

Com a AWS CLI ou outros clientes baseados em API, adicione tags usando o parâmetro `--tags` no comando [create-configuration-template](#).

```
$ aws elasticbeanstalk create-configuration-template \
 --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
 --application-name my-app --template-name my-template --solution-stack-
name solution-stack
```

### Gerenciar tags de uma configuração salva existente

É possível adicionar, atualizar e excluir tags em uma configuração salva existente do Elastic Beanstalk.

## Como gerenciar as tags de uma configuração salva usando o console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

### Note

Se você tiver muitos aplicativos, use a barra de pesquisa para filtrar a lista de aplicativos.

3. No painel de navegação, encontre o nome do aplicativo e escolha Saved configurations (Configurações salvas).
4. Selecione a configuração salva que deseja gerenciar.
5. Escolha Actions (Ações) e Manage tags (Gerenciar tags).
6. Use o formulário na tela para adicionar, atualizar ou excluir tags.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Se você usar a CLI do EB para atualizar a configuração salva, use [eb tags](#) para adicionar, atualizar, excluir ou listar tags.

Por exemplo, o comando a seguir lista as tags em uma configuração salva.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-template"
```

O comando a seguir atualiza a tag mytag1 e exclui a tag mytag2.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
 --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-template"
```

Para obter uma lista de opções e mais exemplos, consulte [eb tags](#).

Com a AWS CLI ou outros clientes baseados em API, use o comando [list-tags-for-resource](#) para listar as tags de uma configuração salva.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
"arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-
template"
```

Use o comando [update-tags-for-resource](#) para adicionar, atualizar ou excluir tags em uma configuração salva.

```
$ aws elasticbeanstalk update-tags-for-resource \
--tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
--resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-
id:configurationtemplate/my-app/my-template"
```

Especifique as tags a serem adicionadas e atualizadas no parâmetro `--tags-to-add` do `update-tags-for-resource`. Uma tag não existente é adicionada, e o valor de uma tag existente é atualizado.

#### Note

Para usar alguns dos comandos da EB CLI e da AWS CLI com uma configuração salva do Elastic Beanstalk, você precisa do ARN da configuração salva. Para construir o ARN, primeiro use o comando a seguir para recuperar o nome da configuração salva.

```
$ aws elasticbeanstalk describe-applications --application-names my-app
```

Procure a chave `ConfigurationTemplates` na saída do comando. Esse elemento mostra o nome da configuração salva. Use esse nome, em que `my-template` é especificado nos comandos mencionados nessa página.

## Manifesto de ambiente (`env.yaml`)

Você pode incluir um manifesto de ambiente formatado com YAML na raiz do seu pacote de origem do aplicativo para configurar o nome do ambiente, a pilha de soluções e os [links do ambiente](#) para usar ao criá-lo.

Este formato de arquivo inclui suporte para grupos do ambiente. Para usar grupos, especifique o nome do ambiente no manifesto com um símbolo `+` no final. Ao criar ou atualizar o ambiente, especifique o nome do grupo com `--group-name` (AWS CLI) ou `--env-group-suffix` (EB CLI).

Para obter mais informações sobre grupos, consulte [Criar e atualizar grupos de ambientes do Elastic Beanstalk](#).

O manifesto de exemplo a seguir define um ambiente de servidor Web com um link para um componente do ambiente de operador do qual ele depende. O manifesto usa grupos para permitir a criação de vários ambientes com o mesmo pacote de origem:

### **~/myapp/frontend/env.yaml**

```
AWSConfigurationTemplateVersion: 1.1.0.0
SolutionStack: 64bit Amazon Linux 2015.09 v2.0.6 running Multi-container Docker 1.7.1
(Generic)
OptionSettings:
 aws:elasticbeanstalk:command:
 BatchSize: '30'
 BatchSizeType: Percentage
 aws:elasticbeanstalk:sns:topics:
 Notification Endpoint: me@example.com
 aws:elb:policies:
 ConnectionDrainingEnabled: true
 ConnectionDrainingTimeout: '20'
 aws:elb:loadbalancer:
 CrossZone: true
 aws:elasticbeanstalk:environment:
 ServiceRole: aws-elasticbeanstalk-service-role
 aws:elasticbeanstalk:application:
 Application Healthcheck URL: /
 aws:elasticbeanstalk:healthreporting:system:
 SystemType: enhanced
 aws:autoscaling:launchconfiguration:
 IamInstanceProfile: aws-elasticbeanstalk-ec2-role
 InstanceType: t2.micro
 EC2KeyName: workstation-uswest2
 aws:autoscaling:updatepolicy:rollingupdate:
 RollingUpdateType: Health
 RollingUpdateEnabled: true
Tags:
 Cost Center: WebApp Dev
CName: front-A08G28LG+
EnvironmentName: front+
EnvironmentLinks:
 "WORKERQUEUE" : "worker+"
```



As seguintes chaves são compatíveis.

- `AWSConfigurationTemplateVersion` (obrigatória): a versão do modelo de configuração (1.1.0.0).

```
AWSConfigurationTemplateVersion: 1.1.0.0
```

- `Platform`: o nome de recurso da Amazon (ARN) da versão da plataforma do ambiente. Você pode especificar a plataforma pelo ARN ou pelo nome da pilha de soluções.

```
Platform:
```

```
PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit Amazon Linux/2.5.0
```

- `SolutionStack`: o nome completo da [pilha de soluções](#) usada para criar o ambiente.

```
SolutionStack: 64bit Amazon Linux 2017.03 v2.5.0 running Java 8
```

- `OptionSettings`: definições da [opção de Configuration \(Configuração\)](#) para aplicar ao ambiente. Por exemplo, a seguinte entrada define o tipo de instância como t2.micro.

```
OptionSettings:
```

```
aws:autoscaling:launchconfiguration:
InstanceType: t2.micro
```

- `Tags`: até 47 tags a serem aplicadas aos recursos criados no ambiente.

```
Tags:
```

```
Cost Center: WebApp Dev
```

- `EnvironmentTier`: o tipo de ambiente a ser criado. Para um ambiente de servidor Web, você pode excluir esta seção (servidor Web é o padrão). Para um ambiente de operador, use o seguinte.

```
EnvironmentTier:
```

```
Name: Worker
Type: SQS/HTTP
```

- `CName`: o CNAME do ambiente. Incluir um caractere + no fim do nome para habilitar grupos.

```
CName: front-A08G28LG+
```

- `EnvironmentName`: o nome do ambiente a ser criado. Incluir um caractere + no fim do nome para habilitar grupos.

```
EnvironmentName: front+
```

Com os grupos habilitados, você deve especificar o nome do grupo ao criar os ambientes. O Elastic Beanstalk anexa o nome do grupo ao nome do ambiente com um hífen. Por exemplo, com o nome do ambiente `front+` e o nome do grupo `dev`, o Elastic Beanstalk criará o ambiente com o nome `front-dev`.

- `EnvironmentLinks`: um mapa de nomes de variáveis e de ambientes das dependências. O exemplo a seguir torna o ambiente `worker+` uma dependência e instrui o Elastic Beanstalk a salvar as informações do link em uma variável denominada `WORKERQUEUE`.

```
EnvironmentLinks:
 "WORKERQUEUE" : "worker+"
```

O valor da variável de link muda de acordo com o tipo de ambiente vinculado. Para um ambiente de servidor Web, o link é o CNAME do ambiente. Para um ambiente de operador, o link é o nome da fila do Amazon Simple Queue Service (Amazon SQS) do ambiente.

É possível usar as chaves `CName`, `EnvironmentName` e `EnvironmentLinks` para criar [grupos do ambiente](#) e [links para outros ambientes](#). Atualmente, esses recursos são suportados ao usar a EB CLI, a AWS CLI ou um SDK.

## Usar uma imagem de máquina da Amazon (AMI) personalizada

Quando você cria um ambiente do AWS Elastic Beanstalk, pode especificar uma imagem de máquina da Amazon (AMI) em vez de usar a AMI do Elastic Beanstalk padrão incluída na versão da plataforma. Uma AMI personalizada pode melhorar o tempo de provisionamento quando as instâncias são iniciadas em seu ambiente e você precisa instalar muitos softwares que não são incluídos nas AMIs padrão.

O uso de [arquivos de configuração](#) é ideal para configurar e personalizar seu ambiente de maneira rápida e consistente. A aplicação de configurações, entretanto, pode demorar muito tempo durante a criação do ambiente e atualizações. Se você especificar muitas configurações de servidor em arquivos de configuração, poderá reduzir esse tempo ao criar uma AMI personalizada que já tenha o software e a configuração de que precisa.

Uma AMI personalizada também permite fazer alterações em componentes de baixo nível, como o kernel do Linux, que são difíceis de implantar ou que demoram muito tempo para aplicar em arquivos de configuração. Para criar uma AMI personalizada, inicie uma AMI da plataforma do Elastic Beanstalk no Amazon EC2, personalize o software e a configuração de acordo com suas necessidades e, depois, interrompa a instância e salve uma AMI a partir dela.

## Criar uma AMI do personalizada

Como identificar a AMI base do Elastic Beanstalk

1. Em uma janela de comando, execute um comando semelhante ao seguinte. Para obter mais informações, consulte [describe-platform-version](#) na Referência de comando da AWS CLI.

Especifique a região da AWS em que você deseja usar a AMI personalizada e substitua o ARN da plataforma e o número de versão pela plataforma do Elastic Beanstalk na qual a aplicação está baseada.

Example - Mac OS/SO Linux

```
$ aws elasticbeanstalk describe-platform-version --region us-east-2 \
 --platform-arn "arn:aws:elasticbeanstalk:us-east-2::platform/Tomcat 8.5 with
Java 8 running on 64bit Amazon Linux/3.1.6" \
 --query PlatformDescription.CustomAmiList

[
 {
 "VirtualizationType": "pv",
 "ImageId": ""
 },
 {
 "VirtualizationType": "hvm",
 "ImageId": "ami-020ae06fdda6a0f66"
 }
]
```

Example - SO Windows

```
C:\> aws elasticbeanstalk describe-platform-version --region us-east-2 --platform-
arn"arn:aws:elasticbeanstalk:us-east-2::platform/IIS 10.0 running on 64bit Windows
Server 2019/2.6.4" --query PlatformDescription.CustomAmiList

[
```

```
{
 "VirtualizationType": "pv",
 "ImageId": ""
},
{
 "VirtualizationType": "hvm",
 "ImageId": "ami-020ae06fdda6a0f66"
}
]
```

2. Anote o valor ImageId, que será semelhante a `ami-020ae06fdda6a0f66` no resultado.

O valor é a AMI do Elastic Beanstalk de estoque para a versão da plataforma, a arquitetura da instância do EC2 e a região da AWS relevantes para a aplicação. Se você precisa criar AMIs para várias plataformas, arquiteturas ou regiões da AWS, repita o processo para identificar a AMI básica correta para cada combinação.

#### Observações

- Não crie uma AMI a partir de uma instância que foi executada em um ambiente do Elastic Beanstalk. O Elastic Beanstalk faz alterações nas instâncias durante o provisionamento que podem causar problemas na AMI salva. Salvar uma imagem de uma instância em um ambiente do Elastic Beanstalk também faz com que a versão da aplicação que foi implantada na instância seja uma parte fixa da imagem.
- Recomendamos usar sempre a versão mais recente da plataforma. Ao atualizar para uma nova versão da plataforma, também recomendamos que você restabeleça a base da sua AMI personalizada na AMI da nova versão da plataforma. Isso minimiza falhas de implantação devido a versões incompatíveis de pacotes ou bibliotecas.

Para o Linux, também é possível criar uma AMI personalizada a partir de uma AMI de comunidade que não foi publicada pelo Elastic Beanstalk. É possível usar a AMI mais recente do [Amazon Linux](#) como ponto de partida. Quando você executa um ambiente com uma AMI do Linux que não é gerenciada pelo Elastic Beanstalk, o Elastic Beanstalk tenta instalar o software da plataforma (idioma, estrutura, servidor de proxy, etc.) e os componentes adicionais para oferecer suporte a recursos como [relatórios de integridade aprimorada](#).

**Note**

AMIs personalizadas baseadas no Windows Server exigem a AMI do Elastic Beanstalk de estoque retornada de `describe-platform-version`, como mostrado anteriormente na Etapa 1.

Embora o Elastic Beanstalk possa usar uma AMI não gerenciada pelo Elastic Beanstalk, o aumento no tempo de provisionamento que resulta da instalação pelo Elastic Beanstalk de componentes ausentes pode reduzir ou eliminar os benefícios da criação de uma AMI personalizada. Outras distribuições do Linux talvez funcionem com a solução de alguns problemas, mas não são oficialmente compatíveis. Se a aplicação exigir uma distribuição específica do Linux, uma alternativa será criar uma imagem de Docker e executá-la na [plataforma Docker](#) ou na [plataforma Docker de vários contêineres](#) do Elastic Beanstalk.

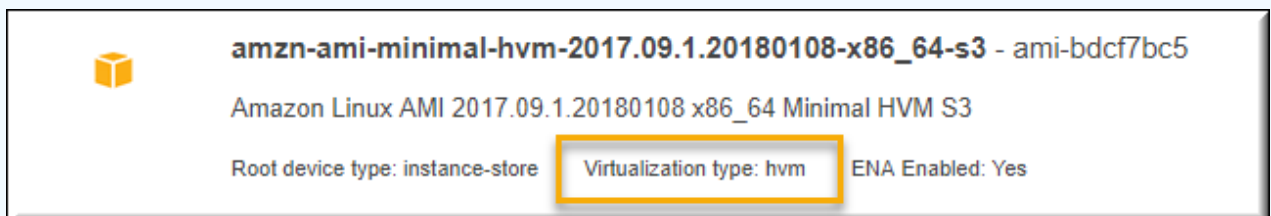
Para criar uma AMI personalizada

1. Abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
2. Escolha Launch Instance (Executar instância).
3. Escolha Community AMIs (AMIs da comunidade).
4. Se você identificou uma AMI base do Elastic Beanstalk (usando `describe-platform-version`) ou uma AMI do Amazon Linux, insira o ID da AMI na caixa de pesquisa. Em seguida, pressione Enter.

Você também pode pesquisar a lista para outra comunidade AMI que atenda às suas necessidades.

**Note**

Recomendamos que você escolha uma AMI que use a virtualização de HVM. Essas AMIs mostram um tipo de virtualização: `hvm` em sua descrição.



Para obter detalhes sobre os tipos de virtualização de instâncias, consulte [Tipos de virtualização de AMI do Linux](#) no Guia do usuário do Amazon EC2 para instâncias do Linux ou [tipos de virtualização de AMI do Windows](#) no Guia do usuário do Amazon EC2 para instâncias do Windows.

- Escolha Select para selecionar a AMI.
- Selecione um tipo de instância e, a seguir, escolha Próximo: Configurar detalhes da instância.
- (Para plataformas Linux) Expanda a seção Detalhes avançados e cole o texto a seguir no campo Dados do usuário.

```
#cloud-config
repo_releasever: repository version number
repo_upgrade: none
```

O número de versão do repositório é a versão de ano e mês no nome da AMI. Por exemplo, AMIs com base na versão de março de 2015 do Amazon Linux têm um número de versão de repositório 2015.03. Para uma imagem do Elastic Beanstalk, isso corresponde à data mostrada no nome da pilha de soluções para a [versão da plataforma](#) com base na AMI do Amazon Linux (anterior ao Amazon Linux 2).

#### Note

A configuração `repo_releasever` define o recurso de bloqueio no lançamento para uma AMI do Amazon Linux. Isso faz com que a AMI use uma versão fixa e específica do repositório quando é iniciada. Esse recurso não é compatível com o Amazon Linux 2, ou seja, não o especifique se seu ambiente usa uma ramificação de plataforma atual do Amazon Linux 2. A configuração será necessária se você estiver usando uma AMI personalizada com o Elastic Beanstalk somente nas ramificações da plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2).

A configuração `repo_upgrade` desativa a instalação automática de atualizações de segurança. É necessário usar uma AMI personalizada com o Elastic Beanstalk.

- Siga as instruções do assistente para [iniciar a instância do EC2](#). Quando solicitado, selecione um par de chaves ao qual tenha acesso para que você possa se conectar à instância para as próximas etapas.
- [Conecte-se à instância](#) com SSH ou RDP.
- Execute todas as personalizações desejadas.

11. (Plataformas Windows) Execute o Sysprep do serviço EC2Config. Para obter informações sobre o EC2Config, consulte [Configurar uma instância do Windows usando o serviço EC2Config](#). Certifique-se de que o Sysprep está configurado para gerar uma senha aleatória que pode ser recuperada no AWS Management Console.
12. No console do Amazon EC2, interrompa a instância do EC2. Em seguida, no menu Instance Actions (Ações da instância), escolha Create Image (EBS AMI) (Criar imagem (AMI do EBS)).
13. Para evitar encargos adicionais da AWS, [termine a instância do EC2](#).

## Como usar sua AMI personalizada em um ambiente do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Em AMI ID (ID de AMI), insira seu ID de AMI personalizado.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Quando você cria um ambiente com a AMI personalizada, deve usar a mesma versão da plataforma usada como base para criar a AMI. Se você aplicar posteriormente uma [atualização de plataforma](#) a um ambiente usando uma AMI personalizada, o Elastic Beanstalk tentará aplicar a biblioteca e as atualizações de configurações durante o processo de bootstrapping.

## Limpeza de uma AMI personalizada

Quando tiver concluído com uma AMI personalizada e não precisar mais dela para iniciar ambientes do Elastic Beanstalk, considere limpá-la para minimizar o custo de armazenamento. A limpeza de uma AMI personalizada envolve o cancelamento de seu registro no Amazon EC2 e a exclusão de outros recursos associados. Para obter detalhes, consulte [Cancelar o registro da AMI do Linux](#) ou [Cancelar o registro da AMI do Windows](#).

## Preservar o acesso a uma imagem de máquina da Amazon (AMI) para uma plataforma descontinuada

O Elastic Beanstalk define o status de uma ramificação da plataforma como descontinuada quando o sistema operacional ou componente principal usado pela ramificação atinge o fim da vida útil. Podemos também transformar a AMI básica do Elastic Beanstalk para a ramificação da plataforma em privada para evitar o uso dessa AMI desatualizada. Os ambientes que usam AMIs que foram alteradas para privadas não poderão mais iniciar instâncias.

Se você não conseguir migrar sua aplicação para um ambiente compatível antes de ser descontinuada, seu ambiente pode estar nessa situação. Pode ser necessário atualizar um ambiente para uma ramificação da plataforma Beanstalk, onde sua AMI básica do Elastic Beanstalk foi alterada para privada. Há uma abordagem alternativa. Você pode atualizar um ambiente existente com base em uma cópia da AMI básica do Elastic Beanstalk usada pelo seu ambiente.

Este tópico apresenta algumas etapas e um script independente para atualizar um ambiente existente com base em uma cópia da AMI básica do Elastic Beanstalk usada pelo seu ambiente. Depois de migrar sua aplicação para uma plataforma compatível, você pode continuar usando os procedimentos padrão para manter sua aplicação e ambientes compatíveis.

### Etapas manuais

Para atualizar um ambiente com base em uma cópia da AMI básica do Elastic Beanstalk

1. Determine qual AMI seu ambiente está usando. Este comando retorna a AMI usada pelo ambiente do Elastic Beanstalk que você fornece nos parâmetros. O valor retornado é usado como `source-ami-id` na próxima etapa.

Em uma janela de comando, execute um comando semelhante ao seguinte. Para obter mais informações, consulte [describe-configuration-settings](#) na Referência de comandos da AWS CLI.

Especifique a região da AWS que armazena a AMI de origem que você deseja copiar. Substitua o nome da aplicação e o nome do ambiente por aqueles baseados na AMI de origem. Insira o texto para o parâmetro de consulta, conforme mostrado.

#### Example

```
>aws elasticbeanstalk describe-configuration-settings \
--application-name my-application \
--region us-east-1 \
--environment-name my-environment \
--profile my-profile \
--query 'ConfigurationSettings[0].SourceAmiId'
```



```
--environment-name my-environment \
--region us-east-2 \
--query "ConfigurationSettings[0].OptionSettings[?OptionName=='ImageId'] |
[0].Value"
```

2. Copie a AMI em sua conta. Este comando retorna a nova AMI resultante da cópia do source-ami-id que foi retornada na etapa anterior.

#### Note

Não se esqueça de anotar o novo ID da AMI que é gerado por esse comando. Você precisará inseri-lo na próxima etapa, substituindo copied-ami-id no comando de exemplo.

Em uma janela de comando, execute um comando semelhante ao seguinte. Para obter mais informações, consulte [copy-image](#) na AWS CLI Command Reference.

Especifique a região da AWS da AMI de origem que você deseja copiar (--source-region) e a região em que você deseja usar sua nova AMI personalizada (--region). Substitua source-ami-id pela AMI da imagem que você está copiando. O source-ami-id foi retornado pelo comando na etapa anterior. Substitua new-ami-name por um nome para descrever a nova AMI na região de destino. O script que segue esse procedimento gera o novo nome da AMI anexando a string "Copy of" ao início do nome do source-ami-id.

```
>aws ec2 copy-image \
--region us-east-2 \
--source-image-id source-ami-id \
--source-region us-east-2 \
--name new-ami-name
```

3. Atualize o ambiente para usar a AMI copiada. Depois de executar o comando, ele retornará o status do ambiente.

Em uma janela de comando, execute um comando semelhante ao seguinte. Para obter mais informações, consulte [update-environment](#) na Referência de comandos da AWS CLI.

Especifique a região da AWS do ambiente e da aplicação que você precisa atualizar. Substitua o nome da aplicação e o nome do ambiente pelos que você precisa associar ao copied-ami-id da

etapa anterior. Para o parâmetro `--option-settings`, substitua `copied-ami-id` pelo ID da AMI que você anotou na saída do comando anterior.

```
>aws elasticbeanstalk update-environment \
 --application-name my-application \
 --environment-name my-environment \
 --region us-east-2 \
 --option-settings
 "Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId,Value=copied-ami-id"
```

### Note

Para minimizar o custo de armazenamento, considere limpar a AMI personalizada quando não precisar mais dela para iniciar ambientes do Elastic Beanstalk. Para obter mais informações, consulte [Limpeza de uma AMI personalizada](#).

## Script autônomo

O script a seguir fornece os mesmos resultados das etapas manuais anteriores. Faça o download do script selecionando este link: [copy\\_ami\\_and\\_update\\_env.zip](#).

Fonte do script: `copy_ami_and_update_env.sh`

```
#!/bin/bash

set -ue

USAGE="This script is used to copy an AMI used by your Elastic Beanstalk environment
 into your account to use in your environment.\n\n"
USAGE+="Usage:\n\n"
USAGE+="./$(basename $0) [OPTIONS]\n\n"
USAGE+="OPTIONS:\n\n"
USAGE+="\t--application-name <application-name>\tThe name of your Elastic Beanstalk
 application.\n\n"
USAGE+="\t--environment-name <environment-name>\tThe name of your Elastic Beanstalk
 environment.\n\n"
USAGE+="\t--region <region> \t\t\tThe AWS region your Elastic Beanstalk environment is
 deployed to.\n\n"
```

```
USAGE+="\n\n"
USAGE+="Script Usage Example(s):\n"
USAGE+="./$(basename $0) --application-name my-application --environment-name my-
environment --region us-east-1\n"

if [$# -eq 0]; then
 echo -e $USAGE
 exit
fi

while [[$# -gt 0]]; do
 case $1 in
 --application-name) APPLICATION_NAME="$2"; shift ;;
 --environment-name) ENVIRONMENT_NAME="$2"; shift ;;
 --region) REGION="$2"; shift ;;
 *) echo "Unknown option $1" ; echo -e $USAGE ; exit ;;
 esac
 shift
done

aws_cli_version="$(aws --version)"
if [$? -ne 0]; then
 echo "aws CLI not found. Please install it: https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html. Exiting."
 exit 1
fi
echo "Using aws CLI version: ${aws_cli_version}"

account=$(aws sts get-caller-identity --query "Account" --output text)
echo "Using account ${account}"

environment_ami_id=$(aws elasticbeanstalk describe-configuration-settings \
 --application-name "$APPLICATION_NAME" \
 --environment-name "$ENVIRONMENT_NAME" \
 --region "$REGION" \
 --query "ConfigurationSettings[0].OptionSettings[?OptionName=='ImageId'] | [0].Value" \
 --output text)
echo "Image associated with environment ${ENVIRONMENT_NAME} is ${environment_ami_id}"

owned_image=$(aws ec2 describe-images \
 --owners self \
 --image-ids "$environment_ami_id" \
 --region "$REGION" \
```

```
--query "Images[0]" \
--output text)
if ["$owned_image" != "None"]; then
 echo "${environment_ami_id} is already owned by account ${account}. Exiting."
 exit
fi

source_image_name=$(aws ec2 describe-images \
 --image-ids "$environment_ami_id" \
 --region "$REGION" \
 --query "Images[0].Name" \
 --output text)
if ["$source_image_name" = "None"]; then
 echo "Cannot find ${environment_ami_id}. Please contact AWS support if you need
 additional help: https://aws.amazon.com/support."
 exit 1
fi

copied_image_name="Copy of ${source_image_name}"
copied_ami_id=$(aws ec2 describe-images \
 --owners self \
 --filters Name=name,Values="${copied_image_name}" \
 --region "$REGION" \
 --query "Images[0].ImageId" \
 --output text)
if ["$copied_ami_id" != "None"]; then
 echo "Detected that ${environment_ami_id} has already been copied by account
 ${account}. Skipping image copy."
else
 echo "Copying ${environment_ami_id} to account ${account} with name
 ${copied_image_name}"
 copied_ami_id=$(aws ec2 copy-image \
 --source-image-id "$environment_ami_id" \
 --source-region "$REGION" \
 --name "$copied_image_name" \
 --region "$REGION" \
 --query "ImageId" \
 --output text)
 echo "New AMI ID is ${copied_ami_id}"

 echo "Waiting for ${copied_ami_id} to become available"
 aws ec2 wait image-available \
 --image-ids "$copied_ami_id" \
 --region "$REGION"
```

```
echo "${copied_ami_id} is now available"
fi

echo "Updating environment ${ENVIRONMENT_NAME} to use ${copied_ami_id}"
environment_status=$(aws elasticbeanstalk update-environment \
 --application-name "$APPLICATION_NAME" \
 --environment-name "$ENVIRONMENT_NAME" \
 --option-settings
"Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId,Value=
${copied_ami_id}" \
 --region "$REGION" \
 --query "Status" \
 --output text)
echo "Environment ${ENVIRONMENT_NAME} is now ${environment_status}"

echo "Waiting for environment ${ENVIRONMENT_NAME} update to complete"
aws elasticbeanstalk wait environment-updated \
 --application-name "$APPLICATION_NAME" \
 --environment-names "$ENVIRONMENT_NAME" \
 --region "$REGION"
echo "Environment ${ENVIRONMENT_NAME} update complete"
```

### Note

Você deve ter a AWS CLI instalada para executar o script. Para obter instruções de atualização, consulte [Instalar ou atualizar a versão mais recente da AWS CLI](#) no Guia do usuário do AWS Command Line Interface.

Depois de instalar a AWS CLI, você também deve configurá-la para usar a conta da AWS proprietária do ambiente. Para obter mais informações, consulte [Configurar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface. A conta também deve ter permissões para criar uma AMI e atualizar o ambiente do Elastic Beanstalk.

Estas etapas descrevem o processo que o script segue.

1. Imprima a conta em uso.
2. Determine qual AMI é usada pelo ambiente (AMI de origem).
3. Verifique se a AMI de origem já pertence à conta. Se sim, saia.
4. Determine o nome da AMI de origem para que ela possa ser usada no novo nome da AMI. Isso também serve para confirmar o acesso à AMI de origem.

5. Verifique se a AMI de origem já foi copiada para a conta. Isso é feito pesquisando AMIs com o nome da AMI copiada de propriedade da conta. Se o nome da AMI tiver sido alterado entre as execuções do script, ela copiará a imagem novamente.
6. Se a AMI de origem ainda não tiver sido copiada, copie a AMI de origem para a conta e aguarde até que a nova AMI esteja disponível.
7. Atualize a configuração do ambiente para usar a nova AMI.
8. Aguarde a conclusão da atualização do ambiente.

Depois de extrair o script do arquivo [copy\\_ami\\_and\\_update\\_env.zip](#), execute-o conforme o exemplo a seguir. Substitua o nome da aplicação e o nome do ambiente no exemplo por seus valores.

```
>sh copy_ami_and_update_env.sh \
 --application-name my-application \
 --environment-name my-environment \
 --region us-east-1
```

#### Note

Para minimizar o custo de armazenamento, considere limpar a AMI personalizada quando não precisar mais dela para iniciar ambientes do Elastic Beanstalk. Para obter mais informações, consulte [Limpeza de uma AMI personalizada](#).

## Fornecer arquivos estáticos

Para melhorar o desempenho, você pode configurar o servidor de proxy para fornecer arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro de seu aplicativo web. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

O Elastic Beanstalk é compatível com a configuração do proxy para atender a arquivos estáticos na maioria das ramificações da plataforma baseadas no Amazon Linux 2. A única exceção é o Docker.

#### Note

Nas plataformas Python e Ruby, o Elastic Beanstalk configura algumas pastas de arquivos estáticos por padrão. Para obter detalhes, consulte as seções de configuração de arquivo

estático para [Python](#) e [Ruby](#). É possível configurar pastas adicionais conforme explicado nesta página.

## Configurar arquivos estáticos usando o console

Para configurar o servidor de proxy para fornecer arquivos estáticos

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Vá até a seção Platform software (Software da plataforma) e localize o grupo Static files (Arquivos estáticos).
  - a. Para adicionar um mapeamento de arquivo estático, selecione Add static files (Adicionar arquivos estáticos). Na linha extra que aparece, você inserirá um caminho para servir arquivos estáticos e o diretório que contém os arquivos estáticos a serem servidos.
    - No campo Path (Caminho), inicie o nome do caminho com uma barra (/) (por exemplo, "/images").
    - No campo Directory (Diretório), especifique um nome de diretório localizado na raiz do código-fonte do seu aplicativo. Não comece com uma barra (por exemplo, "static/image-files").

### Note

Se a seção Static files (Arquivos estáticos) não for exibida, adicione pelo menos um mapeamento usando um [arquivo de configuração](#). Para obter detalhes, consulte [the](#)

[section called “Configurar arquivos estáticos usando opções de configuração”](#) nesta página.

- b. Para remover um mapeamento, selecione Remove (Remover).
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

## Configurar arquivos estáticos usando opções de configuração

É possível usar um [arquivo de configuração](#) para configurar caminhos de arquivos estáticos e locais de diretório usando opções de configuração. É possível adicionar um arquivo de configuração ao pacote de origem do aplicativo e implantá-lo durante a criação do ambiente ou em uma implantação posterior.

Se o seu ambiente usar uma ramificação da plataforma baseada no Amazon Linux 2, use o namespace [aws:elasticbeanstalk:environment:proxy:staticfiles](#).

O exemplo de arquivo de configuração a seguir informa ao servidor de proxy para servir arquivos na pasta `statichtml` no caminho `/html`, e arquivos na pasta `staticimages` no caminho `/images`.

Example `.ebextensions/static-files.config`

```
option_settings:
 aws:elasticbeanstalk:environment:proxy:staticfiles:
 /html: statichtml
 /images: staticimages
```

Se o seu ambiente do Elastic Beanstalk usar uma versão da plataforma da AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as seguintes informações adicionais:

Namespaces específicos da plataforma da AMI do Amazon Linux

Nas ramificações da plataforma da AMI do Amazon Linux, os namespaces de configuração de arquivos estáticos variam de acordo com a plataforma. Para obter detalhes, consulte uma das seguintes páginas:

- [Namespace de configuração do Go](#)
- [Namespace de configuração do Java SE](#)
- [Namespaces de configuração do Tomcat](#)
- [Namespace de configuração do Node.js](#)



- [Namespaces de configuração do Python](#)

## Configurar HTTPS para o seu ambiente do Elastic Beanstalk

Se você tiver adquirido e configurado um [nome de domínio personalizado](#) para o seu ambiente Elastic Beanstalk, poderá usar HTTPS para permitir que os usuários se conectem ao seu site com segurança. Se você não tiver um nome de domínio, ainda poderá usar HTTPS com um certificado autoassinado para fins de desenvolvimento e teste. O HTTPS é essencial para qualquer aplicativo que transmite dados de usuário ou informações de login.

A maneira mais simples de usar HTTPS com um ambiente Elastic Beanstalk é [atribuindo um certificado de servidor ao balanceador de carga do seu ambiente](#). Quando você configura o load balancer para encerrar o HTTPS, a conexão entre o cliente e o load balancer é segura. As conexões de back-end entre o load balancer e as instâncias do EC2 usam HTTP, portanto nenhuma configuração adicional das instâncias é necessária.

### Note

Com o [AWS Certificate Manager \(ACM\)](#), você pode criar um certificado confiável para seus nomes de domínio gratuitamente. Os certificados do ACM só podem ser usados com balanceadores de carga da AWS e distribuições do Amazon CloudFront, e o ACM está [disponível somente em determinadas regiões da AWS](#).

Para usar um certificado do ACM com o Elastic Beanstalk, consulte [Configurar o balanceador de carga do seu ambiente Elastic Beanstalk para encerrar o HTTPS](#).

Se você executar o seu aplicativo em um ambiente de uma única instância ou precisar proteger a conexão o caminho todo até as instâncias do EC2 atrás do load balancer, você pode [configurar o servidor de proxy que é executado na instância para encerrar o HTTPS](#). Configurar suas instâncias para encerrar as conexões HTTPS exige o uso de [arquivos de configuração](#) para modificar o software em execução nas instâncias e para modificar os grupos de segurança para permitir conexões seguras.

Para HTTPS de ponta a ponta em um ambiente com balanceamento de carga, é possível [combinar o encerramento da instância e do load balancer](#) para criptografar ambas as conexões. Por padrão, se você configurar o load balancer para encaminhar o tráfego usando HTTPS, ele confiará em qualquer certificado apresentado a ele pelas instâncias de back-end. Para obter o máximo de segurança,

you can attach policies to the load balancer that prevent it from connecting to instances that do not present a public certificate that it trusts.

#### Note

You can also configure the load balancer to [retransmit the HTTPS traffic without encrypting it](#). The disadvantage of this method is that the load balancer cannot see the requests and, therefore, cannot optimize the routing or report response metrics.

If ACM is not available in your region, you can purchase a trusted certificate from a third party. A third-party certificate can be used to encrypt HTTPS traffic on the load balancer, on the back-end instances, or on both.

For development and testing, you can [create and sign a certificate](#) on your own with open-source tools. Self-signed certificates are free and easy to create, but they cannot be used for front-end encryption on public sites. If you try to use a self-signed certificate for an HTTPS connection with a client, the user's browser will display an error message indicating that the site is not secure. However, you can use a self-signed certificate to protect back-end connections without problems.

ACM is the preferred tool for provisioning, managing, and deploying your certificates to the server by using programming or the AWS CLI. If ACM is not [available in your AWS region](#), you can [upload a self-signed or third-party certificate and a private key](#) to the AWS Identity and Access Management (IAM) using the AWS CLI. Certificates stored in IAM can be used with load balancers and CloudFront distributions.

#### Note

The example application [Does it have Snakes?](#) on GitHub includes configuration files and instructions for each configuration method for HTTPS with a Tomcat web application. Consult the [README file](#) and the [HTTPS instructions](#) for more details.

## Topics

- [Create and access an X.509 certificate](#)
- [Upload a certificate to IAM](#)
- [Configure the load balancer of your Elastic Beanstalk environment to terminate HTTPS](#)

- [Configurar o aplicativo para encerrar conexões HTTPS na instância](#)
- [Configurar a criptografia de ponta a ponta em um ambiente Elastic Beanstalk com balanceamento de carga](#)
- [Configurar o load balancer do ambiente para passagem de TCP](#)
- [Armazenar chaves privadas com segurança no Amazon S3](#)
- [Configurar o redirecionamento de HTTP para HTTPS](#)

## Criar e acessar um certificado X509

Você pode criar um certificado X509 para seu aplicativo com OpenSSL. OpenSSL é uma biblioteca de código aberto padrão que oferece suporte a uma grande variedade de funções de criptografia, incluindo a criação e a assinatura de certificados x509. Para obter mais informações sobre OpenSSL, visite [www.openssl.org](http://www.openssl.org).

### Note

Só será necessário criar um certificado localmente se você quiser [usar HTTPS em um ambiente de instância única](#) ou [criptografar novamente no back-end](#) com um certificado autoassinado. Se você possui um nome de domínio, pode criar um certificado na AWS e usá-lo com um ambiente com balanceamento de carga gratuitamente usando o AWS Certificate Manager (ACM). Consulte [Solicite um certificado](#) no AWS Certificate Manager Guia do usuário para obter instruções.

Execute `openssl version` na linha de comando para verificar se o OpenSSL já está instalado. Caso não esteja, você pode criar e instalar o código-fonte usando as instruções no [repositório GitHub público](#) ou usar o gerenciador de pacotes de sua preferência. O OpenSSL também é instalado em imagens Linux do Elastic Beanstalk, por isso, uma alternativa rápida é conectar-se a uma instância do EC2 em um ambiente em execução usando o comando `eb ssh` da [CLI do EB](#):

```
~/eb$ eb ssh
[ec2-user@ip-255-55-55-255 ~]$ openssl version
OpenSSL 1.0.1k-fips 8 Jan 2015
```

Você precisa criar uma chave privada RSA para criar sua solicitação de assinatura de certificado (CSR). Para criar sua chave privada, use o comando `openssl genrsa`:

```
[ec2-user@ip-255-55-55-255 ~]$ openssl genrsa 2048 > privatekey.pem
Generating RSA private key, 2048 bit long modulus
.....
+++
.....+++
e is 65537 (0x10001)
```

### *privatekey.pem*

O nome do arquivo onde deseja salvar a chave privada. Em geral, o comando `openssl genrsa` imprime o conteúdo da chave privada na tela, mas esse comando envia a saída para um arquivo. Escolha qualquer nome de arquivo e armazene o arquivo em um local seguro, de modo que possa recuperá-lo mais tarde. Se você perder a sua chave privada, não poderá usar seu certificado.

Um CSR é um arquivo que você envia a uma autoridade de certificação (CA) para solicitar um certificado de servidor digital. Para criar um CSR, use o comando `openssl req`:

```
$ openssl req -new -key privatekey.pem -out csr.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

Insira as informações solicitadas e pressione Enter. A tabela a seguir descreve e mostra exemplos para cada campo.

Name (Nome)	Descrição	Exemplo
Nome do país	A abreviação ISO de duas letras para seu país.	US = Estados Unidos
Estado	O nome do estado ou província onde sua organização está localizada. Você não pode abreviar esse nome.	Washington

Name (Nome)	Descrição	Exemplo
Nome da localidade e	O nome da cidade onde sua organização está localizada.	Seattle
Nome da organização	A razão social completa da sua organização. Não abrevie o nome de sua organização.	Corporação de exemplo
Unidade Organizacional	Opcional, para fornecer informações adicionais sobre sua organização.	Marketing
Nome comum	O nome de domínio totalmente qualificado para o seu site. Isso deve corresponder ao nome de domínio que os usuários veem quando visitam seu site; caso contrário, serão mostrados erros de certificado.	www.exemplo.com
Endereço de e-mail	O endereço de e-mail do administrador do site.	someone@example.com

Você pode enviar a solicitação de assinatura para terceiros assinarem ou assine-a você mesmo para desenvolvimento e teste. Os certificados autoassinados também podem ser usados para HTTPS back-end entre um load balancer e as instâncias do EC2.

Para assinar o certificado, use o comando `openssl x509`. O exemplo a seguir usa a chave privada da etapa anterior (*privatekey.pem*) e a solicitação de assinatura (*csr.pem*) para criar um certificado público denominado *public.crt* que é válido por 365 dias.

```
$ openssl x509 -req -days 365 -in csr.pem -signkey privatekey.pem -out public.crt
Signature ok
subject=/C=us/ST=Washington/L=Seattle/O=example corporation/OU=marketing/
CN=www.example.com/emailAddress=someone@example.com
Getting Private key
```

Guarde a chave privada e o certificado público para uso posterior. Você pode descartar a solicitação de assinatura. Sempre [armazene a chave privada em um local seguro](#) e evite adicioná-la ao seu código-fonte.

Para usar o certificado com a plataforma Windows Server, converta-o em um formato PFX. Use o comando a seguir para criar um certificado PFX da chave privada e dos arquivos públicos do certificado:

```
$ openssl pkcs12 -export -out example.com.pfx -inkey privatekey.pem -in public.crt
Enter Export Password: password
Verifying - Enter Export Password: password
```

Agora que você tem um certificado, pode [fazer upload para o IAM](#) para uso com um load balancer ou [configurar as instâncias em seu ambiente para encerrar o HTTPS](#).

## Upload de um certificado no IAM

Para usar seu certificado com o balanceador de carga do seu ambiente do Elastic Beanstalk, carregue o certificado e chave privada no AWS Identity and Access Management (IAM). É possível usar um certificado armazenado no IAM com balanceadores de carga Elastic Load Balancing e distribuições do Amazon CloudFront.

### Note

AWS Certificate Manager (ACM) é a ferramenta preferencial para provisionar, gerenciar e implantar seus certificados de servidor. Para obter mais informações sobre como solicitar um certificado do ACM, consulte [Solicitação de um certificado](#) no Manual do usuário do AWS Certificate Manager. Para obter mais informações sobre a importação de certificados de terceiros para o ACM, consulte [Importação de certificados](#) no Manual do usuário do AWS Certificate Manager. Somente use o IAM para carregar um certificado se o ACM não estiver [disponível em sua região da AWS](#).

Você pode usar a [AWS Command Line Interface](#) (AWS CLI) para carregar o seu certificado. O seguinte comando faz upload de um certificado autoassinado chamado *https-cert.crt* com uma chave privada chamada *private-key.pem*:

```
$ aws iam upload-server-certificate --server-certificate-name elastic-beanstalk-x509 --
certificate-body file://https-cert.crt --private-key file://private-key.pem
{
 "ServerCertificateMetadata": {
 "ServerCertificateId": "AS5YBEI0N02Q7CAIHKNGC",
 "ServerCertificateName": "elastic-beanstalk-x509",
```

```
"Expiration": "2017-01-31T23:06:22Z",
"Path": "/",
"Arn": "arn:aws:iam::123456789012:server-certificate/elastic-beanstalk-x509",
"UploadDate": "2016-02-01T23:10:34.167Z"
}
}
```

O prefixo `file://` informa a AWS CLI para carregar o conteúdo de um arquivo no diretório atual. `elastic-beanstalk-x509` especifica o nome para chamar o certificado no IAM.

Se você adquiriu um certificado de uma autoridade de certificação e recebeu um arquivo de cadeia de certificados, faça upload dele incluindo a opção `--certificate-chain`:

```
$ aws iam upload-server-certificate --server-certificate-name elastic-beanstalk-x509 --
certificate-chain file://certificate-chain.pem --certificate-body file://https-cert.crt
--private-key file://private-key.pem
```

Anote o Nome de recurso da Amazon (ARN) do seu certificado. Você o usará quando atualizar as definições de configuração do load balancer para usar HTTPS.

#### Note


Um certificado carregado para o IAM permanece armazenado mesmo depois de deixar de ser usado em qualquer balanceador de carga do ambiente. Ele contém dados confidenciais. Quando não precisar mais do certificado para qualquer ambiente, lembre-se de excluí-lo. Para obter detalhes sobre como excluir um certificado do IAM, consulte [https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_credentials\\_server-certs.html#delete-server-certificate](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_server-certs.html#delete-server-certificate).

Para obter mais informações sobre certificados de servidor no IAM, consulte [Trabalhar com certificados de servidor](#) no Guia do usuário do IAM.

## Configurar o balanceador de carga do seu ambiente Elastic Beanstalk para encerrar o HTTPS

Para atualizar seu AWS Elastic Beanstalk ambiente para usar HTTPS, você precisa configurar um ouvinte HTTPS para o balanceador de carga em seu ambiente. Dois tipos de balanceador de carga são compatíveis com um listener HTTPS: Classic Load Balancer e Application Load Balancer.

É possível usar o console do Elastic Beanstalk ou um arquivo de configuração para configurar um listener seguro e atribuir o certificado.


 Note

Ambientes de instância única não têm um load balancer e não são compatíveis com o encerramento do HTTPS no load balancer.

## Configurar um listener seguro usando o console do Elastic Beanstalk


Como atribuir um certificado ao load balancer do seu ambiente

1. Abra o console do [Elastic Beanstalk](#) e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Load balancer, escolha Edit (Editar).


 Note

Se a categoria de configuração Load balancer não tem um botão Edit (Editar), seu ambiente não tem um [load balancer](#).

5. Na página Modify load balancer (Modificar load balancer), o procedimento varia dependendo do tipo de load balancer associado ao seu ambiente.
  - Classic Load Balancer
    - a. Escolha Add listener.
    - b. Na caixa de diálogo Classic Load Balancer listener (Listener do Classic Load Balancer), defina as seguintes configurações:
      - Para Listener port, digite a porta de tráfego de entrada, normalmente 443.
      - Para Listener protocol, escolha HTTPS.



- Para Instance port, digite 80.
- Para Instance protocol, escolha HTTP.
- Para SSL certificate, choose your certificate.
- c. Escolha Adicionar.
- Application Load Balancer
  - a. Escolha Add listener.
  - b. Na caixa de diálogo Application Load Balancer listener (Listener do Application Load Balancer), defina as seguintes configurações:
    - Para Port, digite a porta de tráfego de entrada, normalmente 443.
    - Em Protocol, escolha HTTPS.
    - Para SSL certificate, choose your certificate.
  - c. Escolha Adicionar.

 Note

Para Classic Load Balancer e Application Load Balancer, se o menu suspenso não mostrar nenhum certificado, será necessário criar ou carregar um certificado para seu nome de [domínio personalizado](#) no [AWS Certificate Manager \(ACM\)](#) (preferencial). Como alternativa, faça upload de um certificado para o IAM com a AWS CLI.

- Network Load Balancer
  - a. Escolha Add listener.
  - b. Na caixa de diálogo Network Load Balancer listener (Listener do Network Load Balancer) em Port (Porta), digite o número da porta de tráfego de entrada que, normalmente, é 443.
  - c. Escolha Adicionar.
- 6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

## Configurar um listener seguro usando um arquivo de configuração

É possível configurar um listener seguro em seu load balancer com um dos seguintes [arquivos de configuração](#).

Example `.ebextensions/securelistener-clb.config`

Use este exemplo quando seu ambiente tiver um Classic Load Balancer. O exemplo usa as opções no namespace do `aws:elb:listener` para configurar um listener HTTPS na porta 443 com o certificado especificado e encaminhar o tráfego descryptografado para as instâncias em seu ambiente na porta 80.

```
option_settings:
 aws:elb:listener:443:
 SSLCertificateId: arn:aws:acm:us-east-2:1234567890123:certificate/
#####
 ListenerProtocol: HTTPS
 InstancePort: 80
```

Substitua o texto destacado pelo ARN do seu certificado. O certificado pode ser aquele que você criou ou carregou no AWS Certificate Manager (ACM) (preferencial) ou um que você enviou para o IAM com o AWS CLI

Para obter mais informações sobre as opções de configuração do Classic Load Balancer, consulte [Namespaces de configuração do Classic Load Balancer](#).

Example `.ebextensions/securelistener-alb.config`

Use este exemplo quando seu ambiente tiver um Application Load Balancer. O exemplo usa opções no namespace `aws:elbv2:listener` para configurar um listener HTTPS na porta 443 com o certificado especificado. O listener direciona o tráfego para o processo padrão.

```
option_settings:
 aws:elbv2:listener:443:
 ListenerEnabled: 'true'
 Protocol: HTTPS
 SSLCertificateArns: arn:aws:acm:us-east-2:1234567890123:certificate/
#####
```

## Example .ebextensions/securelistener-nlb.config

Use este exemplo quando seu ambiente tiver um Network Load Balancer. O exemplo usa as opções no namespace `aws:elbv2:listener` para configurar um listener na porta 443. O listener direciona o tráfego para o processo padrão.

```
option_settings:
 aws:elbv2:listener:443:
 ListenerEnabled: 'true'
```

## Configurar um grupo de segurança

Se você configurar o load balancer para encaminhar o tráfego para uma porta de instância diferente da porta 80, será necessário adicionar uma regra para o security group que permita o tráfego de entrada vindo de seu load balancer pela porta de instância configurada. Se você criar seu ambiente em uma VPC personalizada, o Elastic Beanstalk adicionará essa regra para você.

Você adiciona essa regra ao inserir uma chave `Resources` em um [arquivo de configuração](#) no diretório `.ebextensions` para seu aplicativo.

O arquivo de configuração de exemplo a seguir adiciona uma regra de entrada no security group `AWSEBSecurityGroup`. Isso permite tráfego na porta 1000 do security group do load balancer.

## Example .ebextensions/sg-ingressfromlb.config

```
Resources:
 sslSecurityGroupIngress:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp
 ToPort: 1000
 FromPort: 1000
 SourceSecurityGroupId: {"Fn::GetAtt" : ["AWSEBLoadBalancerSecurityGroup",
"GroupId"]}
```

## Configurar o aplicativo para encerrar conexões HTTPS na instância

Use [arquivos de configuração](#) para configurar o servidor de proxy que passa o tráfego para o aplicativo para encerrar conexões HTTPS. Isso é útil se você quer usar o HTTPS com um ambiente

de uma única instância, ou se configurar o load balancer para passar o tráfego através sem descriptografá-lo.

Para habilitar o HTTPS, é necessário permitir o tráfego de entrada na porta 443 para a instância do EC2 na qual a aplicação do Elastic Beanstalk estiver sendo executada. Faça isso usando a chave `Resources` do arquivo de configuração para adicionar uma regra da porta 443 às regras de ingresso para o security group `AWSEBSecurityGroup`.

O seguinte snippet adiciona uma regra de entrada para o security group `AWSEBSecurityGroup` que abre a porta 443 para todo o tráfego de um ambiente de uma única instância:

### **.ebextensions/https-instance-securitygroup.config**

```
Resources:
 sslSecurityGroupIngress:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp
 ToPort: 443
 FromPort: 443
 CidrIp: 0.0.0.0/0
```

Em um ambiente com balanceamento de carga em um [Amazon Virtual Private Cloud](#) (Amazon VPC) padrão, é possível modificar essa política para aceitar apenas o tráfego do balanceador de carga. Para ver um exemplo, consulte [Configurar a criptografia de ponta a ponta em um ambiente Elastic Beanstalk com balanceamento de carga](#).

### Plataformas

- [Encerrar o HTTPS nas instâncias do EC2 que executam Docker](#)
- [Encerrar o HTTPS em instâncias do EC2 que executam o Go](#)
- [Encerrar o HTTPS em instâncias do EC2 que executam Java SE](#)
- [Encerrar o HTTPS em instâncias do EC2 que executam Node.js](#)
- [Encerrar o HTTPS em instâncias do EC2 que executam o PHP](#)
- [Encerrar o HTTPS em instâncias do EC2 que executam Python](#)
- [Como encerrar o HTTPS em instâncias do EC2 que executam o Ruby](#)
- [Encerrar o HTTPS em instâncias do EC2 que executam Tomcat](#)

- [Encerrar o HTTPS em instâncias do Amazon EC2 executando .NET Core no Linux](#)
- [Encerrar o HTTPS em instâncias do Amazon EC2 que executam o .NET](#)

## Encerrar o HTTPS nas instâncias do EC2 que executam Docker

Para os contêineres do Docker, use um [arquivo de configuração](#) para habilitar o HTTPS.

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo o material de certificado e de chave privada como instruído, e salve-o no diretório `.ebextensions` do seu pacote de origem. O arquivo de configuração executa as seguintes tarefas:

- A chave `files` cria os seguintes arquivos na instância:

```
/etc/nginx/conf.d/https.conf
```

Configura o servidor nginx. Este arquivo é carregado quando o serviço do nginx é iniciado.

```
/etc/pki/tls/certs/server.crt
```

Cria o arquivo de certificado na instância. Substitua o *conteúdo do arquivo de certificado* pelo conteúdo do seu certificado.

### Note

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
-----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

## /etc/pki/tls/certs/server.key

Cria o arquivo de chave privada na instância. Substitua o *conteúdo da chave privada* pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

### Example .ebextensions/https-instance.config

```
files:
 /etc/nginx/conf.d/https.conf:
 mode: "000644"
 owner: root
 group: root
 content: |
 # HTTPS Server

 server {
 listen 443;
 server_name localhost;

 ssl on;
 ssl_certificate /etc/pki/tls/certs/server.crt;
 ssl_certificate_key /etc/pki/tls/certs/server.key;

 ssl_session_timeout 5m;

 ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
 ssl_prefer_server_ciphers on;

 location / {
 proxy_pass http://docker;
 proxy_http_version 1.1;

 proxy_set_header Connection "";
 proxy_set_header Host $host;
 proxy_set_header X-Real-IP $remote_addr;
 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 proxy_set_header X-Forwarded-Proto https;
 }
 }

 /etc/pki/tls/certs/server.crt:
```

```
mode: "000400"
owner: root
group: root
content: |
 -----BEGIN CERTIFICATE-----
 certificate file contents
 -----END CERTIFICATE-----

/etc/pki/tls/certs/server.key:
mode: "000400"
owner: root
group: root
content: |
 -----BEGIN RSA PRIVATE KEY-----
 private key contents # See note below.
 -----END RSA PRIVATE KEY-----
```

### Note

Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte [Armazenar chaves privadas com segurança no Amazon S3](#).

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O seguinte arquivo de configuração recupera a ID do grupo de segurança usando uma [função](#) AWS CloudFormation e adiciona uma regra a ela.

Example `.ebextensions/https-instance-single.config`

```
Resources:
 sslSecurityGroupIngress:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp
 ToPort: 443
 FromPort: 443
 CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, configure o load balancer para [passar o tráfego seguro inalterado](#) ou [descriptografar e criptografar novamente](#) para criptografia de ponta a ponta.

## Encerrar o HTTPS em instâncias do EC2 que executam o Go

Para os tipos de contêiner Go, você habilita o HTTPS com um [arquivo de configuração](#) e um arquivo de configuração nginx que configura o servidor nginx para usar HTTPS.

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo os espaços reservados do certificado e da chave privada como instruído, e salve-o no diretório `.ebextensions` do pacote de origem. O arquivo de configuração executa as seguintes tarefas:

- A chave `Resources` habilita a porta 443 no security group usado pela instância do seu ambiente.
- A chave `files` cria os seguintes arquivos na instância:

```
/etc/pki/tls/certs/server.crt
```

Cria o arquivo de certificado na instância. Substitua o *conteúdo do arquivo de certificado* pelo conteúdo do seu certificado.

### Note

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
-----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```



```
/etc/pki/tls/certs/server.key
```

Cria o arquivo de chave privada na instância. Substitua o *conteúdo da chave privada* pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

- A chave `container_commands` reinicia o servidor nginx depois que tudo é configurado para o servidor carregar o arquivo de configuração nginx.

Example `.ebextensions/https-instance.config`

```
files:
 /etc/pki/tls/certs/server.crt:
 content: |
 -----BEGIN CERTIFICATE-----
 certificate file contents
 -----END CERTIFICATE-----

 /etc/pki/tls/certs/server.key:
 content: |
 -----BEGIN RSA PRIVATE KEY-----
 private key contents # See note below.
 -----END RSA PRIVATE KEY-----

container_commands:
 01restart_nginx:
 command: "service nginx restart"
```

### Note

Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte [Armazenar chaves privadas com segurança no Amazon S3](#).

Insira o seguinte em um arquivo com a extensão `.conf` no diretório `.ebextensions/nginx/conf.d/` do seu pacote de origem (ex. `.ebextensions/nginx/conf.d/https.conf`). Substitua *app\_port* pelo número da porta na qual seu aplicativo escuta. Este exemplo configura o servidor

nginx para escutar na porta 443 usando SSL. Para obter mais informações sobre esses arquivos de configuração na plataforma Go, consulte [Configurar o proxy reverso](#).

Example `.ebextensions/nginx/conf.d/https.conf`

```
HTTPS server

server {
 listen 443;
 server_name localhost;

 ssl on;
 ssl_certificate /etc/pki/tls/certs/server.crt;
 ssl_certificate_key /etc/pki/tls/certs/server.key;

 ssl_session_timeout 5m;

 ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
 ssl_prefer_server_ciphers on;

 location / {
 proxy_pass http://localhost:app_port;
 proxy_set_header Connection "";
 proxy_http_version 1.1;
 proxy_set_header Host $host;
 proxy_set_header X-Real-IP $remote_addr;
 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 proxy_set_header X-Forwarded-Proto https;
 }
}
```

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O seguinte arquivo de configuração recupera a ID do grupo de segurança usando uma [função](#) AWS CloudFormation e adiciona uma regra a ela.

Example `.ebextensions/https-instance-single.config`

```
Resources:
 sslSecurityGroupIngress:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
```

```
IpProtocol: tcp
ToPort: 443
FromPort: 443
CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, configure o load balancer para [passar o tráfego seguro inalterado](#) ou [descriptografar e criptografar novamente](#) para criptografia de ponta a ponta.

## Encerrar o HTTPS em instâncias do EC2 que executam Java SE

Para os tipos de contêiner Java SE, você habilita o HTTPS com um [arquivo de configuração](#) .ebextensions e um arquivo de configuração nginx que configura o servidor nginx para usar HTTPS.

Todas as plataformas AL2023/AL2 oferecem suporte a um recurso de configuração de proxy uniforme. Para obter mais informações sobre a configuração do servidor proxy nas versões de plataforma que executam o AL2023/AL2, expanda a seção Configuração do proxy reverso em [the section called “Estender plataformas Linux”](#).

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo os espaços reservados do certificado e da chave privada como instruído, e salve-o no diretório .ebextensions. O arquivo de configuração executa as seguintes tarefas:

- A chave files cria os seguintes arquivos na instância:

```
/etc/pki/tls/certs/server.crt
```

Cria o arquivo de certificado na instância. Substitua o *conteúdo do arquivo de certificado* pelo conteúdo do seu certificado.

### Note

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
-----BEGIN CERTIFICATE-----
certificate file contents
```

```

-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----

```

/etc/pki/tls/certs/server.key

Cria o arquivo de chave privada na instância. Substitua o *conteúdo da chave privada* pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

- A chave `container_commands` reinicia o servidor nginx depois que tudo é configurado para o servidor carregar o arquivo de configuração nginx.

Example `.ebextensions/https-instance.config`

```

files:
 /etc/pki/tls/certs/server.crt:
 content: |
 -----BEGIN CERTIFICATE-----
 certificate file contents
 -----END CERTIFICATE-----

 /etc/pki/tls/certs/server.key:
 content: |
 -----BEGIN RSA PRIVATE KEY-----
 private key contents # See note below.
 -----END RSA PRIVATE KEY-----

container_commands:
 0restart_nginx:
 command: "service nginx restart"

```

### Note

Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para

fazer download dele durante a implantação. Para obter instruções, consulte [Armazenar chaves privadas com segurança no Amazon S3](#).

Insira o seguinte em um arquivo com a extensão `.conf` no diretório `.ebextensions/nginx/conf.d/` do seu pacote de origem (ex. `.ebextensions/nginx/conf.d/https.conf`). Substitua `app_port` pelo número da porta na qual seu aplicativo escuta. Este exemplo configura o servidor nginx para escutar na porta 443 usando SSL. Para obter mais informações sobre esses arquivos de configuração na plataforma Java SE, consulte [Configurar o proxy reverso](#).

Example `.ebextensions/nginx/conf.d/https.conf`

```
HTTPS server

server {
 listen 443;
 server_name localhost;

 ssl on;
 ssl_certificate /etc/pki/tls/certs/server.crt;
 ssl_certificate_key /etc/pki/tls/certs/server.key;

 ssl_session_timeout 5m;

 ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
 ssl_prefer_server_ciphers on;

 location / {
 proxy_pass http://localhost:app_port;
 proxy_set_header Connection "";
 proxy_http_version 1.1;
 proxy_set_header Host $host;
 proxy_set_header X-Real-IP $remote_addr;
 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 proxy_set_header X-Forwarded-Proto https;
 }
}
```

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O seguinte arquivo de configuração recupera a ID do grupo de segurança usando uma [função](#) AWS CloudFormation e adiciona uma regra a ela.

## Example .ebextensions/https-instance-single.config

```
Resources:
 sslSecurityGroupIngress:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp
 ToPort: 443
 FromPort: 443
 CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, configure o balanceador de carga para [passar o tráfego seguro inalterado](#) ou [descriptografar e criptografar novamente](#) para criptografia de ponta a ponta.

## Encerrar o HTTPS em instâncias do EC2 que executam Node.js

O arquivo de configuração de exemplo a seguir [amplia a configuração padrão do nginx](#) para escutar na porta 443 e encerrar conexões SSL/TLS com um certificado público e uma chave privada.

Se você configurou seu ambiente para [relatórios de saúde aprimorados](#), precisará configurar o nginx para gerar logs de acesso. Para fazer isso, apague a escrita do bloco de linhas sob o comentário `# For enhanced health...` removendo o `#` inicial.

## Example .ebextensions/https-instance.config

```
files:
 /etc/nginx/conf.d/https.conf:
 mode: "000644"
 owner: root
 group: root
 content: |
 # HTTPS server

 server {
 listen 443;
 server_name localhost;

 ssl on;
 ssl_certificate /etc/pki/tls/certs/server.crt;
 ssl_certificate_key /etc/pki/tls/certs/server.key;
```

```
ssl_session_timeout 5m;

ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;

For enhanced health reporting support, uncomment this block:

#if ($time_iso8601 ~ "^(\\d{4})-(\\d{2})-(\\d{2})T(\\d{2})") {
set $year $1;
set $month $2;
set $day $3;
set $hour $4;
#}
healthd;
#access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour
#access_log /var/log/nginx/access.log main;

location / {
 proxy_pass http://nodejs;
 proxy_set_header Connection "";
 proxy_http_version 1.1;
 proxy_set_header Host $host;
 proxy_set_header X-Real-IP $remote_addr;
 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 proxy_set_header X-Forwarded-Proto https;
}

/etc/pki/tls/certs/server.crt:
mode: "000400"
owner: root
group: root
content: |
 -----BEGIN CERTIFICATE-----
 certificate file contents
 -----END CERTIFICATE-----

/etc/pki/tls/certs/server.key:
mode: "000400"
owner: root
group: root
content: |
 -----BEGIN RSA PRIVATE KEY-----
 private key contents # See note below.
```

```
-----END RSA PRIVATE KEY-----
```


A chave `files` cria os seguintes arquivos na instância:

```
/etc/nginx/conf.d/https.conf
```

Configura o servidor nginx. Este arquivo é carregado quando o serviço do nginx é iniciado.

```
/etc/pki/tls/certs/server.crt
```

Cria o arquivo de certificado na instância. Substitua o *conteúdo do arquivo de certificado* pelo conteúdo do seu certificado.

 Note

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
-----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

```
/etc/pki/tls/certs/server.key
```

Cria o arquivo de chave privada na instância. Substitua o *conteúdo da chave privada* pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.



**Note**

Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte [Armazenar chaves privadas com segurança no Amazon S3](#).

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O seguinte arquivo de configuração recupera a ID do grupo de segurança usando uma [função](#) AWS CloudFormation e adiciona uma regra a ela.

Example `.ebextensions/https-instance-single.config`

```
Resources:
 sslSecurityGroupIngress:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp
 ToPort: 443
 FromPort: 443
 CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, configure o load balancer para [passar o tráfego seguro inalterado](#) ou [descriptografar e criptografar novamente](#) para criptografia de ponta a ponta.

## Encerrar o HTTPS em instâncias do EC2 que executam o PHP

Para tipos de contêiner PHP, você usa um [arquivo de configuração](#) para habilitar o Apache HTTP Server a usar HTTPS.

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo o material de certificado e de chave privada como instruído, e salve-o no diretório `.ebextensions` do seu pacote de origem.

O arquivo de configuração executa as seguintes tarefas:


- A chave `packages` usa o yum para instalar `mod24_ssl`.
- A chave `files` cria os seguintes arquivos na instância:

```
/etc/httpd/conf.d/ssl.conf
```

Configura o servidor Apache. Este arquivo é carregado quando o serviço do Apache é iniciado.

```
/etc/pki/tls/certs/server.crt
```

Cria o arquivo de certificado na instância. Substitua o *conteúdo do arquivo de certificado* pelo conteúdo do seu certificado.

 Note

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
-----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

```
/etc/pki/tls/certs/server.key
```

Cria o arquivo de chave privada na instância. Substitua o *conteúdo da chave privada* pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

Example `.ebextensions/https-instance.config`

```
packages:
 yum:
 mod24_ssl : []
```

```
files:
/etc/httpd/conf.d/ssl.conf:
mode: "000644"
owner: root
group: root
content: |
LoadModule ssl_module modules/mod_ssl.so
Listen 443
<VirtualHost *:443>
 <Proxy *>
 Order deny,allow
 Allow from all
 </Proxy>

 SSLEngine on
 SSLCertificateFile "/etc/pki/tls/certs/server.crt"
 SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"
 SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
 SSLProtocol All -SSLv2 -SSLv3
 SSLHonorCipherOrder On
 SSLSessionTickets Off

 Header always set Strict-Transport-Security "max-age=63072000;
includeSubdomains; preload"
 Header always set X-Frame-Options DENY
 Header always set X-Content-Type-Options nosniff

 ProxyPass / http://localhost:80/ retry=0
 ProxyPassReverse / http://localhost:80/
 ProxyPreserveHost on
 RequestHeader set X-Forwarded-Proto "https" early

</VirtualHost>

/etc/pki/tls/certs/server.crt:
mode: "000400"
owner: root
group: root
content: |
 -----BEGIN CERTIFICATE-----
 certificate file contents
 -----END CERTIFICATE-----
```

```
/etc/pki/tls/certs/server.key:
mode: "000400"
owner: root
group: root
content: |
 -----BEGIN RSA PRIVATE KEY-----
 private key contents # See note below.
 -----END RSA PRIVATE KEY-----
```

### Note

Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte [Armazenar chaves privadas com segurança no Amazon S3](#).

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O seguinte arquivo de configuração recupera a ID do grupo de segurança usando uma [função](#) AWS CloudFormation e adiciona uma regra a ela.

Example `.ebextensions/https-instance-single.config`

```
Resources:
 sslSecurityGroupIngress:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp
 ToPort: 443
 FromPort: 443
 CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, configure o load balancer para [passar o tráfego seguro inalterado](#) ou [descriptografar e criptografar novamente](#) para criptografia de ponta a ponta.

## Encerrar o HTTPS em instâncias do EC2 que executam Python

Para os tipos de contêiner Python que usam Apache HTTP Server com a Web Server Gateway Interface (WSGI), você usa um [arquivo de configuração](#) para habilitar o Apache HTTP Server a usar o HTTPS.

Adicione o seguinte trecho ao seu [arquivo de configuração](#), substituindo o material de certificado e de chave privada como instruído, e salve-o no diretório `.ebextensions` do seu pacote de origem. O arquivo de configuração executa as seguintes tarefas:

- A chave `packages` usa o yum para instalar `mod24_ssl`.
- A chave `files` cria os seguintes arquivos na instância:

```
/etc/httpd/conf.d/ssl.conf
```

Configura o servidor Apache. Se o seu aplicativo não estiver nomeado como `application.py`, substitua o texto destacado no valor para `WSGIScriptAlias` pelo caminho local para seu aplicativo. Por exemplo, um aplicativo django pode estar em `django/wsgi.py`. A localização deve corresponder ao valor da opção `WSGIPath` que você definiu para seu ambiente.

De acordo com os requisitos do seu aplicativo, você também pode precisar adicionar outros diretórios ao parâmetro `python-path`.

```
/etc/pki/tls/certs/server.crt
```

Cria o arquivo de certificado na instância. Substitua o *conteúdo do arquivo de certificado* pelo conteúdo do seu certificado.

### Note

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
-----BEGIN CERTIFICATE-----
certificate file contents
```

```
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

/etc/pki/tls/certs/server.key

Cria o arquivo de chave privada na instância. Substitua o *conteúdo da chave privada* pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

- A chave `container_commands` interrompe o serviço `httpd` após tudo estar configurado, para que ele use o novo certificado e o arquivo `https.conf`.

#### Note

O exemplo funciona apenas em ambientes usando a Plataforma [Python](#).

Example `.ebextensions/https-instance.config`

```
packages:
 yum:
 mod24_ssl : []

files:
 /etc/httpd/conf.d/ssl.conf:
 mode: "000644"
 owner: root
 group: root
 content: |
 LoadModule wsgi_module modules/mod_wsgi.so
 WSGIPythonHome /opt/python/run/baselinenv
 WSGISocketPrefix run/wsgi
 WSGIRestrictEmbedded On
 Listen 443
 <VirtualHost *:443>
 SSLEngine on
 SSLCertificateFile "/etc/pki/tls/certs/server.crt"
```

```
 SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"

 Alias /static/ /opt/python/current/app/static/
 <Directory /opt/python/current/app/static>
 Order allow,deny
 Allow from all
 </Directory>

 WSGIScriptAlias / /opt/python/current/app/application.py

 <Directory /opt/python/current/app>
 Require all granted
 </Directory>

 WSGIDaemonProcess wsgi-ssl processes=1 threads=15 display-name=%{GROUP} \
 python-path=/opt/python/current/app \
 python-home=/opt/python/run/venv \
 home=/opt/python/current/app \
 user=wsgi \
 group=wsgi
 WSGIProcessGroup wsgi-ssl

</VirtualHost>

/etc/pki/tls/certs/server.crt:
mode: "000400"
owner: root
group: root
content: |
 -----BEGIN CERTIFICATE-----
 certificate file contents
 -----END CERTIFICATE-----

/etc/pki/tls/certs/server.key:
mode: "000400"
owner: root
group: root
content: |
 -----BEGIN RSA PRIVATE KEY-----
 private key contents # See note below.
 -----END RSA PRIVATE KEY-----

container_commands:
01killhttpd:
```

```
command: "killall httpd"
02waitforhttpdeath:
command: "sleep 3"
```

### Note

Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte [Armazenar chaves privadas com segurança no Amazon S3](#).

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O seguinte arquivo de configuração recupera a ID do grupo de segurança usando uma [função](#) AWS CloudFormation e adiciona uma regra a ela.

Example .ebextensions/https-instance-single.config

```
Resources:
 sslSecurityGroupIngress:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp
 ToPort: 443
 FromPort: 443
 CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, configure o load balancer para [passar o tráfego seguro inalterado](#) ou [descriptografar e criptografar novamente](#) para criptografia de ponta a ponta.

## Como encerrar o HTTPS em instâncias do EC2 que executam o Ruby

Para tipos de contêiner do Ruby, o modo como o HTTPS é habilitado depende do tipo de servidor de aplicativos utilizado.

### Tópicos

- [Configurar HTTPS para Ruby com Puma](#)



- [Configurar o HTTPS para Ruby com o Passenger](#)

## Configurar HTTPS para Ruby com Puma

Para tipos de contêiner do Ruby que usam o Puma como servidor de aplicativos, você pode usar um [arquivo de configuração](#) para ativar o HTTPS.

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo o material de certificado e de chave privada como instruído, e salve-o no diretório `.ebextensions` do seu pacote de origem. O arquivo de configuração executa as seguintes tarefas:

- A chave `files` cria os seguintes arquivos na instância:

```
/etc/nginx/conf.d/https.conf
```

Configura o servidor nginx. Este arquivo é carregado quando o serviço do nginx é iniciado.

```
/etc/pki/tls/certs/server.crt
```

Cria o arquivo de certificado na instância. Substitua o *conteúdo do arquivo de certificado* pelo conteúdo do seu certificado.

### Note

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
-----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

```
/etc/pki/tls/certs/server.key
```

Cria o arquivo de chave privada na instância. Substitua o *conteúdo da chave privada* pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

- A chave `container_commands` reiniciará o servidor nginx depois que tudo estiver configurado, assim o servidor usará o novo arquivo `https.conf`.

Example `.ebextensions/https-instance.config`

```
files:
 /etc/nginx/conf.d/https.conf:
 content: |
 # HTTPS server

 server {
 listen 443;
 server_name localhost;

 ssl on;
 ssl_certificate /etc/pki/tls/certs/server.crt;
 ssl_certificate_key /etc/pki/tls/certs/server.key;

 ssl_session_timeout 5m;

 ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
 ssl_prefer_server_ciphers on;

 location / {
 proxy_pass http://my_app;
 proxy_set_header Host $host;
 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 proxy_set_header X-Forwarded-Proto https;
 }

 location /assets {
 alias /var/app/current/public/assets;
 gzip_static on;
 gzip on;
 expires max;
 add_header Cache-Control public;
 }
 }

```

```
 }

 location /public {
 alias /var/app/current/public;
 gzip_static on;
 gzip on;
 expires max;
 add_header Cache-Control public;
 }
}

/etc/pki/tls/certs/server.crt:
content: |
 -----BEGIN CERTIFICATE-----
 certificate file contents
 -----END CERTIFICATE-----

/etc/pki/tls/certs/server.key:
content: |
 -----BEGIN RSA PRIVATE KEY-----
 private key contents # See note below.
 -----END RSA PRIVATE KEY-----

container_commands:
 01restart_nginx:
 command: "service nginx restart"
```

### Note

Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte [Armazenar chaves privadas com segurança no Amazon S3](#).

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O seguinte arquivo de configuração recupera a ID do grupo de segurança usando uma [função](#) AWS CloudFormation e adiciona uma regra a ela.

## Example .ebextensions/https-instance-single.config

```
Resources:
 sslSecurityGroupIngress:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp
 ToPort: 443
 FromPort: 443
 CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, configure o load balancer para [passar o tráfego seguro inalterado](#) ou [descriptografar e criptografar novamente](#) para criptografia de ponta a ponta.

### Configurar o HTTPS para Ruby com o Passenger

Para tipos de contêiner do Ruby que usam o Passenger como servidor de aplicativos, você pode usar um arquivo de configuração e um arquivo JSON para ativar o HTTPS.

### Para configurar o HTTPS para Ruby com o Passenger

1. Adicione o seguinte trecho ao seu arquivo de configuração, substituindo o material de certificado e de chave privada como instruído, e salve-o no diretório `.ebextensions` do seu pacote de origem. O arquivo de configuração executa as seguintes tarefas:

- A chave `files` cria os seguintes arquivos na instância:

```
/etc/pki/tls/certs/server.crt
```

Cria o arquivo de certificado na instância. Substitua o *conteúdo do arquivo de certificado* pelo conteúdo do seu certificado.

#### Note

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
-----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

```
/etc/pki/tls/certs/server.key
```

Cria o arquivo de chave privada na instância. Substitua o *conteúdo da chave privada* pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

Example Snippet `.ebextensions` para configuração do HTTPS para Ruby com Passenger

```
files:
 /etc/pki/tls/certs/server.crt:
 content: |
 -----BEGIN CERTIFICATE-----
 certificate file contents
 -----END CERTIFICATE-----

 /etc/pki/tls/certs/server.key:
 content: |
 -----BEGIN RSA PRIVATE KEY-----
 private key contents # See note below.
 -----END RSA PRIVATE KEY-----
```

#### Note

Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela

está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte [Armazenar chaves privadas com segurança no Amazon S3](#).

2. Crie um arquivo de texto e adicione o JSON a seguir ao arquivo. Salve-o no diretório raiz do pacote de origem com o nome `passenger-standalone.json`. Esse arquivo JSON configura o Passenger para usar HTTPS.

#### Important

Esse arquivo JSON não deve conter a marca de ordem de bytes (BOM). Se a resposta for sim, a biblioteca JSON do Passenger não lerá o arquivo corretamente e o serviço Passenger não será iniciado.

#### Example `passenger-standalone.json`

```
{
 "ssl" : true,
 "ssl_port" : 443,
 "ssl_certificate" : "/etc/pki/tls/certs/server.crt",
 "ssl_certificate_key" : "/etc/pki/tls/certs/server.key"
}
```

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O seguinte arquivo de configuração recupera a ID do grupo de segurança usando uma [função](#) AWS CloudFormation e adiciona uma regra a ela.

#### Example `.ebextensions/https-instance-single.config`

```
Resources:
 sslSecurityGroupIngress:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp
 ToPort: 443
 FromPort: 443
 CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, configure o load balancer para [passar o tráfego seguro inalterado](#) ou [descriptografar e criptografar novamente](#) para criptografia de ponta a ponta.

## Encerrar o HTTPS em instâncias do EC2 que executam Tomcat

Para os tipos de contêiner Tomcat, você usa um [arquivo de configuração](#) para habilitar o Apache HTTP Server a usar HTTPS ao atuar como proxy reverso para o Tomcat.

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo o material de certificado e de chave privada como instruído, e salve-o no diretório `.ebextensions` do seu pacote de origem. O arquivo de configuração executa as seguintes tarefas:

- A chave `files` cria os seguintes arquivos na instância:

```
/etc/pki/tls/certs/server.crt
```

Cria o arquivo de certificado na instância. Substitua o *conteúdo do arquivo de certificado* pelo conteúdo do seu certificado.

### Note

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

```
/etc/pki/tls/certs/server.key
```

Cria o arquivo de chave privada na instância. Substitua o *conteúdo da chave privada* pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

```
/opt/elasticbeanstalk/hooks/appdeploy/post/99_start_httpd.sh
```

Cria um gancho de implantação pós-script para reiniciar o serviço `httpd`.

## Example `.ebextensions/https-instance.config`

```
files:
 /etc/pki/tls/certs/server.crt:
 mode: "000400"
```

```
owner: root
group: root
content: |
 -----BEGIN CERTIFICATE-----
 certificate file contents
 -----END CERTIFICATE-----

/etc/pki/tls/certs/server.key:
mode: "000400"
owner: root
group: root
content: |
 -----BEGIN RSA PRIVATE KEY-----
 private key contents # See note below.
 -----END RSA PRIVATE KEY-----

/opt/elasticbeanstalk/hooks/appdeploy/post/99_start_httpd.sh:
mode: "000755"
owner: root
group: root
content: |
 #!/usr/bin/env bash
 sudo service httpd restart
```

Você também precisa configurar o servidor de proxy do ambiente para escutar na porta 443. A configuração do Apache 2.4 a seguir adiciona um ouvinte na porta 443. Para saber mais, consulte [Configurar servidor de proxy do seu ambiente Tomcat](#).

Example `.ebextensions/httpd/conf.d/ssl.conf`

```
Listen 443
<VirtualHost *:443>
 ServerName server-name
 SSLEngine on
 SSLCertificateFile "/etc/pki/tls/certs/server.crt"
 SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"

 <Proxy *>
 Require all granted
 </Proxy>
 ProxyPass / http://localhost:8080/ retry=0
 ProxyPassReverse / http://localhost:8080/
 ProxyPreserveHost on
```



```
ErrorLog /var/log/httpd/elasticbeanstalk-ssl-error_log

</VirtualHost>
```

O fornecedor do seu certificado pode incluir certificados intermediários que você pode instalar para melhorar a compatibilidade com clientes móveis. Configure o Apache com um pacote de certificate authority (CA – autoridade de certificação) intermediária adicionando o seguinte ao arquivo de configuração SSL (consulte [Estender e substituir a configuração padrão do Apache: Amazon Linux AMI \(AL1\)](#) para saber o local):

- No conteúdo do arquivo `ssl.conf`, especifique o arquivo de cadeia:

```
SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"
SSLCertificateChainFile "/etc/pki/tls/certs/gd_bundle.crt"
SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
```

- Adicione uma nova entrada à chave `files` com o conteúdo dos certificados intermediários:

```
files:
 /etc/pki/tls/certs/gd_bundle.crt:
 mode: "000400"
 owner: root
 group: root
 content: |
 -----BEGIN CERTIFICATE-----
 First intermediate certificate
 -----END CERTIFICATE-----
 -----BEGIN CERTIFICATE-----
 Second intermediate certificate
 -----END CERTIFICATE-----
```

### Note

Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para

fazer download dele durante a implantação. Para obter instruções, consulte [Armazenar chaves privadas com segurança no Amazon S3](#).

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O seguinte arquivo de configuração recupera a ID do grupo de segurança usando uma [função](#) AWS CloudFormation e adiciona uma regra a ela.

Example .ebextensions/https-instance-single.config

```
Resources:
 sslSecurityGroupIngress:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp
 ToPort: 443
 FromPort: 443
 CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, configure o load balancer para [passar o tráfego seguro inalterado](#) ou [descriptografar e criptografar novamente](#) para criptografia de ponta a ponta.

## Encerrar o HTTPS em instâncias do Amazon EC2 executando .NET Core no Linux

Para os tipos de contêiner .NET Core no Linux, habilite o HTTPS com um .ebextensions [arquivo de configuração](#) e um arquivo de configuração nginx que configura o servidor nginx para usar HTTPS.

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo os espaços reservados do certificado e da chave privada como instruído, e salve-o no diretório .ebextensions. O arquivo de configuração executa as seguintes tarefas:

- A chave files cria os seguintes arquivos na instância:

```
/etc/pki/tls/certs/server.crt
```

Cria o arquivo de certificado na instância. Substitua o *conteúdo do arquivo de certificado* pelo conteúdo do seu certificado.

**Note**

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```

-----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----

```

`/etc/pki/tls/certs/server.key`

Cria o arquivo de chave privada na instância. Substitua o *conteúdo da chave privada* pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

- A chave `container_commands` reinicia o servidor nginx depois que tudo é configurado para o servidor carregar o arquivo de configuração nginx.

Example `.ebextensions/https-instance.config`

```

files:
 /etc/pki/tls/certs/server.crt:
 content: |
 -----BEGIN CERTIFICATE-----
 certificate file contents
 -----END CERTIFICATE-----

 /etc/pki/tls/certs/server.key:
 content: |

```

```
-----BEGIN RSA PRIVATE KEY-----
private key contents # See note below.
-----END RSA PRIVATE KEY-----
```

```
container_commands:
 01restart_nginx:
 command: "systemctl restart nginx"
```

### Note

Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte [Armazenar chaves privadas com segurança no Amazon S3](#).

Insira o seguinte em um arquivo com a extensão `.conf` no diretório `.platform/nginx/conf.d/` do seu pacote de origem (por ex. `.platform/nginx/conf.d/https.conf`). Substitua `app_port` pelo número da porta na qual seu aplicativo escuta. Este exemplo configura o servidor nginx para escutar na porta 443 usando SSL. Para obter mais informações sobre esses arquivos de configuração na plataforma .NET Core no Linux, consulte [the section called “Servidor de proxy”](#).

Example `.platform/nginx/conf.d/https.conf`

```
HTTPS server

server {
 listen 443 ssl;
 server_name localhost;

 ssl_certificate /etc/pki/tls/certs/server.crt;
 ssl_certificate_key /etc/pki/tls/certs/server.key;

 ssl_session_timeout 5m;

 ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
 ssl_prefer_server_ciphers on;

 location / {
 proxy_pass http://localhost:app_port; }
}
```

```
proxy_set_header Connection "";
proxy_http_version 1.1;
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto https;
}
}
```

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O seguinte arquivo de configuração recupera a ID do grupo de segurança usando uma [função](#) AWS CloudFormation e adiciona uma regra a ela.

Example `.ebextensions/https-instance-single.config`

```
Resources:
 sslSecurityGroupIngress:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp
 ToPort: 443
 FromPort: 443
 CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, configure o load balancer para [passar o tráfego seguro inalterado](#) ou [descriptografar e criptografar novamente](#) para criptografia de ponta a ponta.

## Encerrar o HTTPS em instâncias do Amazon EC2 que executam o .NET


O [arquivo de configuração](#) a seguir cria e executa um PowerShell script do Windows que executa as seguintes tarefas:

- Verifica se há uma associação de certificado HTTPS existente à porta 443.
- Obtém o [certificado PFX](#) de um bucket do Amazon S3.

### Note


Adicione uma `AmazonS3ReadOnlyAccess` política ao `aws-elasticbeanstalk-ec2-role` para acessar o certificado SSL no bucket do Amazon S3.

- Obtém a senha de AWS Secrets Manager.

 Note

Adicione uma declaração `aws-elasticbeanstalk-ec2-role` que permita a `secretsmanager:GetSecretValue` ação do segredo que contém a senha do certificado

- Instala o certificado.
- Vincula o certificado à porta 443.

 Note

Para remover o endpoint HTTP (porta 80), inclua o comando `Remove-WebBinding` na seção `Remova a vinculação HTTP do exemplo`.

### Example `.ebextensions/ .config https-instance-dotnet`

```
files:
 "C:\\certs\\install-cert.ps1":
 content: |
 import-module webadministration
 ## Settings - replace the following values with your own
 $bucket = "DOC-EXAMPLE-BUCKET" ## S3 bucket name
 $certkey = "example.com.pfx" ## S3 object key for your PFX certificate
 $secretname = "example_secret" ## AWS Secrets Manager name for a secret that
 contains the certificate's password
 ##

 # Set variables
 $certfile = "C:\\cert.pfx"
 $pwd = Get-SECSecretValue -SecretId $secretname | select -expand SecretString

 # Clean up existing binding
 if (Get-WebBinding "Default Web Site" -Port 443) {
 Echo "Removing WebBinding"
 Remove-WebBinding -Name "Default Web Site" -BindingInformation *:443:
 }
 if (Get-Item -path IIS:\\SslBindings\\0.0.0.0!443) {
 Echo "Deregistering WebBinding from IIS"
```

```

 Remove-Item -path IIS:\SslBindings\0.0.0.0!443
 }

 # Download certificate from S3
 Read-S3Object -BucketName $bucket -Key $certkey -File $certfile

 # Install certificate
 Echo "Installing cert..."
 $securepwd = ConvertTo-SecureString -String $pwd -Force -AsPlainText
 $cert = Import-PfxCertificate -FilePath $certfile cert:\localMachine\my -Password
$securepwd

 # Create site binding
 Echo "Creating and registering WebBinding"
 New-WebBinding -Name "Default Web Site" -IP "*" -Port 443 -Protocol https
 New-Item -path IIS:\SslBindings\0.0.0.0!443 -value $cert -Force

 ## Remove the HTTP binding
 ## (optional) Uncomment the following line to unbind port 80
 # Remove-WebBinding -Name "Default Web Site" -BindingInformation *:80:
 ##

 # Update firewall
 netsh advfirewall firewall add rule name="Open port 443" protocol=TCP
localport=443 action=allow dir=OUT

commands:
 00_install_ssl:
 command: powershell -NoProfile -ExecutionPolicy Bypass -file C:\\certs\\install-
cert.ps1

```

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O seguinte arquivo de configuração recupera a ID do grupo de segurança usando uma [função](#) AWS CloudFormation e adiciona uma regra a ela.

Example .ebextensions/ .config https-instance-single

```

Resources:
 sslSecurityGroupIngress:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp

```

```
ToPort: 443
FromPort: 443
CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, você configura o balanceador de carga para [passar tráfego seguro intocado ou descriptografar](#) e recriptografar para [criptografia](#). end-to-end

## Configurar a criptografia de ponta a ponta em um ambiente Elastic Beanstalk com balanceamento de carga

O encerramento de conexões seguras no load balancer e o uso de HTTP no back-end podem ser suficientes para seu aplicativo. O tráfego de rede entre os recursos da AWS pode não ser escutado por instâncias que não façam parte da conexão, mesmo que elas estejam em execução na mesma conta.

No entanto, se você está desenvolvendo um aplicativo que precisa estar em conformidade com as rigorosas normas externas, pode ser necessário proteger todas as conexões de rede. É possível usar o console do Elastic Beanstalk ou os [arquivos de configuração](#) para fazer o balanceador de carga do ambiente do Elastic Beanstalk se conectar a instâncias de back-end com segurança a fim de atender a esses requisitos. O procedimento a seguir se concentra nos arquivos de configuração.

Primeiro, [adicione um listener seguro ao seu load balancer](#), caso ainda não o tenha feito.

Você também deve configurar as instâncias do ambiente para escutar na porta segura e encerrar conexões HTTPS. A configuração varia de acordo com a plataforma. Para obter instruções, consulte [Configurar o aplicativo para encerrar conexões HTTPS na instância](#). Use um [certificado autoassinado](#) para as instâncias do EC2 sem problema.

Em seguida, configure o listener para encaminhar o tráfego usando HTTPS na porta segura usada pelo seu aplicativo. Use um dos seguintes arquivos de configuração, com base no tipo de load balancer que seu ambiente utiliza.

### **.ebextensions/https-reencrypt-clb.config**

Use este arquivo de configuração com um Classic Load Balancer. Além de configurar o load balancer, o arquivo de configuração também altera a verificação de integridade padrão para usar a porta 443 e HTTPS, garantindo que o load balancer possa se conectar com segurança.

```
option_settings:
```



```
aws:elb:listener:443:
 InstancePort: 443
 InstanceProtocol: HTTPS
aws:elasticbeanstalk:application:
 Application Healthcheck URL: HTTPS:443/
```

### **.ebextensions/https-reencrypt-alb.config**

Use este arquivo de configuração com um Application Load Balancer.

```
option_settings:
 aws:elbv2:listener:443:
 DefaultProcess: https
 ListenerEnabled: 'true'
 Protocol: HTTPS
 aws:elasticbeanstalk:environment:process:https:
 Port: '443'
 Protocol: HTTPS
```

### **.ebextensions/https-reencrypt-nlb.config**

Use este arquivo de configuração com um Network Load Balancer.

```
option_settings:
 aws:elbv2:listener:443:
 DefaultProcess: https
 ListenerEnabled: 'true'
 aws:elasticbeanstalk:environment:process:https:
 Port: '443'
```

A opção `DefaultProcess` recebe esse nome em virtude do Application Load Balancers, que podem ter listeners não padrão na mesma porta para o tráfego de caminhos específicos (consulte [Application Load Balancer](#) para obter detalhes). Para um Network Load Balancer, a opção especifica o único processo de destino para esse listener.

Neste exemplo, chamamos o processo de `https` porque ele faz o listening do tráfego seguro (HTTPS). O listener envia o tráfego para o processo na porta designada usando o protocolo TCP, pois o Network Load Balancer funciona apenas com TCP. Isso está correto, porque o tráfego de rede para HTTP e HTTPS é implementado na parte superior do TCP.

**Note**

A CLI do EB e o console do Elastic Beanstalk aplicam valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Consulte o [Valores recomendados](#) para obter detalhes.

Na próxima tarefa, você precisa modificar o security group do load balancer para permitir tráfego. Dependendo da [Amazon Virtual Private Cloud](#) (Amazon VPC) em que você inicia seu ambiente, a VPC padrão ou uma VPC personalizada, o grupo de segurança do balanceador de carga variará. Em uma VPC padrão, o Elastic Load Balancing fornece um grupo de segurança padrão que todos os balanceadores de carga podem usar. Em uma Amazon VPC criada, o Elastic Beanstalk cria um grupo de segurança para o balanceador de carga usar.

Para oferecer suporte aos dois cenários, crie um grupo de segurança e instrua o Elastic Beanstalk a usá-lo. O seguinte arquivo de configuração cria um security group e o anexa ao load balancer.

**.ebextensions/https-lbsecuritygroup.config**

```
option_settings:
 # Use the custom security group for the load balancer
 aws:elb:loadbalancer:
 SecurityGroups: ``{ "Ref" : "loadbalancersg" }``
 ManagedSecurityGroup: ``{ "Ref" : "loadbalancersg" }``

Resources:
 loadbalancersg:
 Type: AWS::EC2::SecurityGroup
 Properties:
 GroupDescription: load balancer security group
 VpcId: vpc-#####
 SecurityGroupIngress:
 - IpProtocol: tcp
 FromPort: 443
 ToPort: 443
 CidrIp: 0.0.0.0/0
 - IpProtocol: tcp
 FromPort: 80
 ToPort: 80
 CidrIp: 0.0.0.0/0
 SecurityGroupEgress:
```

```
- IpProtocol: tcp
 FromPort: 80
 ToPort: 80
 CidrIp: 0.0.0.0/0
```

Substitua o texto destacado pelo seu ID de VPC padrão ou personalizada. O exemplo anterior inclui entrada e saída pela porta 80 para permitir conexões HTTP. Remova essas propriedades se você deseja somente permitir conexões seguras.

Por fim, adicione regras de entrada e saída que permitam a comunicação pela porta 443, entre o security group do load balancer e o security group das instâncias.

### **.ebextensions/https-backendsecurity.config**

```
Resources:
 # Add 443-inbound to instance security group (AWSEBSecurityGroup)
 httpsFromLoadBalancerSG:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp
 ToPort: 443
 FromPort: 443
 SourceSecurityGroupId: {"Fn::GetAtt" : ["loadbalancersg", "GroupId"]}
 # Add 443-outbound to load balancer security group (loadbalancersg)
 httpsToBackendInstances:
 Type: AWS::EC2::SecurityGroupEgress
 Properties:
 GroupId: {"Fn::GetAtt" : ["loadbalancersg", "GroupId"]}
 IpProtocol: tcp
 ToPort: 443
 FromPort: 443
 DestinationSecurityGroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
```

Faça isso separadamente da criação do security group para restringir os grupos de segurança de origem e destino, sem criar uma dependência circular.

Depois de concluir todas as tarefas anteriores, o load balancer se conecta a instâncias de back-end com segurança usando HTTPS. O load balancer não se importa se o certificado da instância é autoassinado ou emitido por uma autoridade de certificação confiável, e aceitará qualquer certificado apresentado.

Esse comportamento pode ser alterado adicionando políticas ao load balancer que o instrua a confiar apenas em um certificado específico. O arquivo de configuração a seguir cria duas políticas. Uma política especifica um certificado público, e o outro instrui o load balancer a confiar apenas naquele certificado para conexões com a porta 443 da instância.

### **.ebextensions/https-backendauth.config**

```
option_settings:
 # Backend Encryption Policy
 aws:elb:policies:backendencryption:
 PublicKeyPolicyNames: backendkey
 InstancePorts: 443
 # Public Key Policy
 aws:elb:policies:backendkey:
 PublicKey: |
 -----BEGIN CERTIFICATE-----
 #####
 #####
 #####
 #####
 #####
 -----END CERTIFICATE-----
```

Substitua o texto destacado pelo conteúdo do seu certificado público da instância do EC2.

## Configurar o load balancer do ambiente para passagem de TCP

Se você não deseja que o load balancer em seu ambiente AWS Elastic Beanstalk descriptografe o tráfego HTTPS, pode configurar o listener seguro para transmitir as solicitações para as instâncias de back-end da forma como estão.

Primeiramente, [configure as instâncias do EC2 do seu ambiente para encerrar o HTTPS](#). Teste a configuração em um ambiente de instância única para garantir que tudo esteja funcionando antes de adicionar um load balancer à combinação.

Adicione um [arquivo de configuração](#) ao seu projeto para configurar um listener na porta 443 que transmite pacotes TCP como estão para a porta 443 nas instâncias de back-end:

### **.ebextensions/https-lb-passthrough.config**

```
option_settings:
 aws:elb:listener:443:
```

```
ListenerProtocol: TCP
InstancePort: 443
InstanceProtocol: TCP
```

Em uma [Amazon Virtual Private Cloud](#) (Amazon VPC) padrão, também é necessário adicionar uma regra ao grupo de segurança das instâncias para permitir o tráfego de entrada na porta 443 do balanceador de carga:

### **.ebextensions/https-instance-securitygroup.config**

```
Resources:
 443inboundfromloadbalancer:
 Type: AWS::EC2::SecurityGroupIngress
 Properties:
 GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
 IpProtocol: tcp
 ToPort: 443
 FromPort: 443
 SourceSecurityGroupName: { "Fn::GetAtt": ["AWSEBLoadBalancer",
"SourceSecurityGroup.GroupName"] }
```

Em uma VPC personalizada, o Elastic Beanstalk atualiza a configuração do grupo de segurança para você.

## Armazenar chaves privadas com segurança no Amazon S3

A chave privada que você usa para assinar seu certificado público é privada e não deve ser submetida ao código-fonte. Evite armazenar chaves privadas em arquivos de configuração carregando-as no Amazon S3 e configurando o Elastic Beanstalk para fazer download do arquivo do Amazon S3 durante a implantação da aplicação.

O exemplo a seguir mostra as seções [Recursos](#) e [arquivos](#) de um [arquivo de configuração](#) que faz download do arquivo de chave privada de um bucket do Amazon S3.

### Example .ebextensions/privatekey.config

```
Resources:
 AWSEBAutoScalingGroup:
 Metadata:
 AWS::CloudFormation::Authentication:
 S3Auth:
```

```
type: "s3"
buckets: ["elasticbeanstalk-us-west-2-123456789012"]
roleName:
 "Fn::GetOptionSetting":
 Namespace: "aws:autoscaling:launchconfiguration"
 OptionName: "IamInstanceProfile"
 DefaultValue: "aws-elasticbeanstalk-ec2-role"
files:
 # Private key
 "/etc/pki/tls/certs/server.key":
 mode: "000400"
 owner: root
 group: root
 authentication: "S3Auth"
 source: https://elasticbeanstalk-us-west-2-123456789012.s3.us-west-2.amazonaws.com/
server.key
```

Substitua o nome do bucket e o URL no exemplo pelos seus. A primeira entrada desse arquivo adiciona um método de autenticação chamado S3Auth aos metadados do grupo de Auto Scaling do ambiente. Se você tiver configurado um [perfil de instância](#) personalizado para o ambiente a ser usado, caso contrário, será aplicado o valor padrão `aws-elasticbeanstalk-ec2-role`. O perfil de instância padrão tem permissão de leitura do bucket de armazenamento do Elastic Beanstalk. Se você usar um outro bucket, [adicione permissões ao perfil de instância](#).

A segunda entrada usa o método de autenticação S3Auth para fazer download da chave privada do URL especificado e salvá-la no `/etc/pki/tls/certs/server.key`. O servidor de proxy pode ler a chave privada a partir desse local para [encerrar conexões HTTPS na instância](#).

O perfil de instância atribuído às instâncias do EC2 do ambiente deve ter permissão para ler o objeto chave do bucket especificado. [Verifique se o perfil de instância tem permissão](#) para ler o objeto no IAM e se as permissões no bucket e objeto não proíbem o perfil de instância.

Para visualizar as permissões de um bucket

1. Abra o [Console de gerenciamento do Amazon S3](#).
2. Escolha um bucket.
3. Escolha Propriedades e, em seguida, Permissões.
4. Verifique se sua conta tem permissão de leitura no bucket.
5. Se uma política de bucket estiver anexada, escolha Bucket policy para visualizar as permissões atribuídas ao bucket.

## Configurar o redirecionamento de HTTP para HTTPS

No tópico [Configurar HTTPS para o seu ambiente do Elastic Beanstalk](#) e em seus subtópicos, discutiremos como configurar seu ambiente Elastic Beanstalk para usar HTTPS a fim de garantir a criptografia de tráfego em sua aplicação. Este tópico descreve como lidar de forma elegante com o tráfego HTTP para o seu aplicativo se os usuários finais ainda iniciá-lo. Isso é feito ao configurar o redirecionamento de HTTP para HTTPS, às vezes conhecido como forçando HTTPS.

Para configurar o redirecionamento, configure primeiro seu ambiente para lidar com o tráfego HTTPS. Em seguida, redirecione o tráfego HTTP para HTTPS. Essas duas etapas são discutidas nas subseções a seguir.

### Configurar o ambiente para lidar com o tráfego HTTPS

Dependendo da configuração do balanceamento de carga do ambiente, siga um destes procedimentos:

- Ambiente com balanceamento de carga: [configure o balanceador de carga para encerrar o HTTPS](#).
- Ambiente de única instância: [configure a aplicação para encerrar conexões HTTPS na instância](#). Esta configuração depende da Plataforma do seu ambiente.

### Redirecionar o tráfego HTTP para HTTPS

É possível configurar os servidores Web nas instâncias do ambiente ou o Application Load Balancer do ambiente para redirecionar o tráfego HTTP para HTTPS. Execute um destes procedimentos:

- Configure servidores Web de instância: este método funciona em qualquer ambiente de servidor Web. Configure servidores Web em suas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) para responder ao tráfego HTTP com um status de resposta de redirecionamento HTTP. Esta configuração depende da Plataforma do seu ambiente. Encontre a pasta para sua plataforma na coleção [https-redirect](#) do GitHub e use o arquivo de configuração demonstrativo na pasta.

Se o seu ambiente usar as [verificações de integridade do Elastic Load Balancing](#), o balanceador de carga esperará uma instância de integridade para responder a mensagens de verificação de integridade HTTP com respostas HTTP 200 (OK). Portanto, o servidor da web não deve redirecionar essas mensagens para HTTPS. Os arquivos de configuração demonstrativos em [https-redirect](#) lidam com esse requisito corretamente.

- Configurar balanceador de carga: esse método funciona se você tiver um ambiente com carga balanceada que use um [Application Load Balancer](#). O Application Load Balancer pode enviar respostas de redirecionamento à medida que o tráfego HTTP entra. Nesse caso, não é preciso configurar o redirecionamento nas instâncias do ambiente. Temos dois arquivos de configuração demonstrativos no GitHub que mostram como configurar o Application Load Balancer para redirecionamento. O arquivo de configuração [alb-http-to-https-redirection-full.config](#) cria um listener HTTPS na porta 443 e modifica o listener padrão da porta 80 para redirecionar o tráfego HTTP de entrada para HTTPS. O arquivo de configuração [alb-http-to-https-redirection.config](#) espera que o listener 443 seja definido (você pode usar namespaces de configuração padrão do Elastic Beanstalk ou o console do Elastic Beanstalk). Em seguida, ele cuida de modificar o listener da porta 80 para redirecionamento.



# Monitoramento de ambiente

Quando você executa um site de produção, é importante saber que seu aplicativo está disponível e respondendo às solicitações. Para ajudar a monitorar a capacidade de resposta da aplicação, o Elastic Beanstalk dispõe de recursos que monitoram as estatísticas sobre a aplicação e criam alertas que são acionados quando os limites são excedidos.

## Tópicos

- [Monitoramento da integridade do ambiente no Console de Gerenciamento da AWS](#)
- [Relatórios de integridade básica](#)
- [Monitoramento e relatório de integridade aprimorada](#)
- [Gerenciar alarmes](#)
- [Visualizar o histórico de alterações de um ambiente do Elastic Beanstalk](#)
- [Visualizar o stream de eventos de um ambiente do Elastic Beanstalk](#)
- [Listar e conectar com instâncias de servidor](#)
- [Visualizar logs de instâncias do Amazon EC2 no ambiente do Elastic Beanstalk](#)

## Monitoramento da integridade do ambiente no Console de Gerenciamento da AWS

É possível acessar informações operacionais sobre sua aplicação no console do Elastic Beanstalk. O console exibe o status do ambiente e a integridade do aplicativo rapidamente. Na página Environments (Ambientes) do console e na página de cada aplicativo, os ambientes na lista são codificados por cores para indicar o status.

Como monitorar um ambiente no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

### 3. No painel de navegação, escolha Monitoring (Monitoramento).

A página Monitoramento mostra as estatísticas gerais do ambiente, como a latência média e a utilização de CPU. Além das estatísticas gerais, é possível visualizar gráficos de monitoramento que mostram o uso de recursos ao longo do tempo. Você pode clicar em qualquer um dos gráficos para visualizar informações mais detalhadas.

#### Note

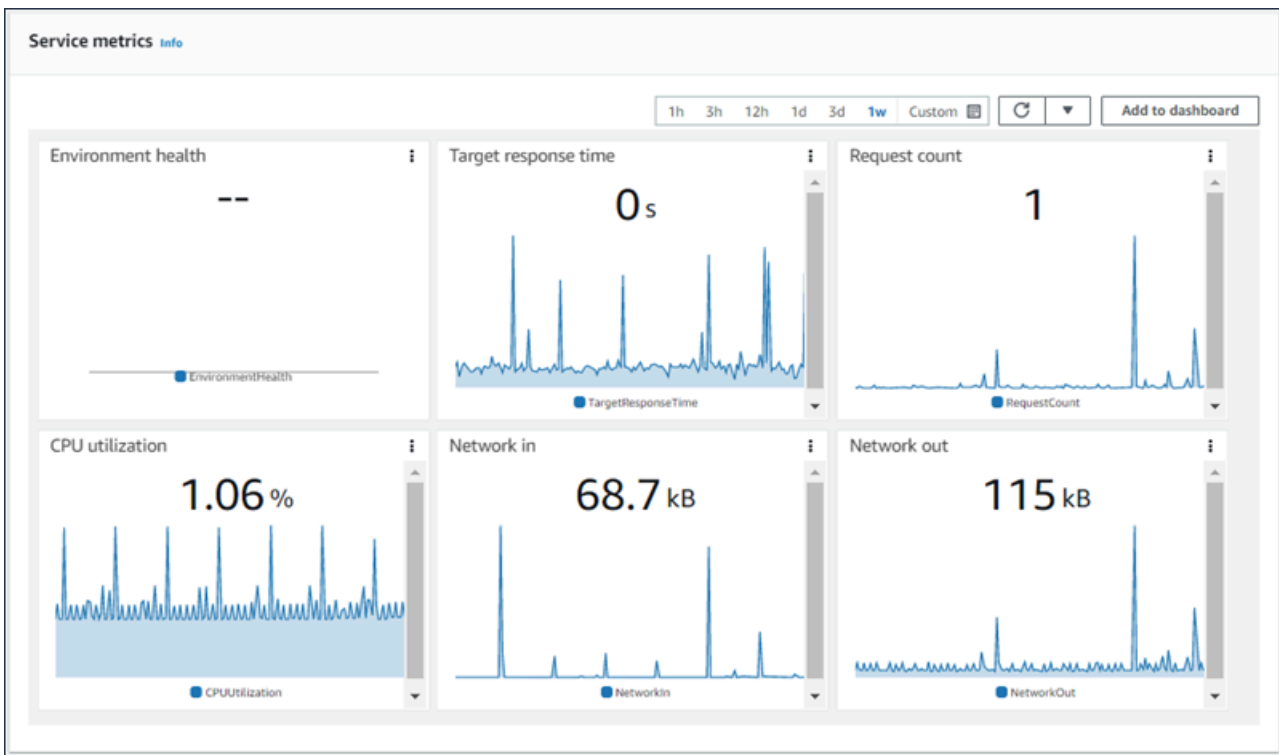
Por padrão, somente as métricas básicas do CloudWatch são habilitadas, que retornam dados em períodos de cinco minutos. É possível permitir métricas do CloudWatch de um minuto mais granulares editando as configurações do ambiente.

## Grafos de monitoramento

A página Monitoring (Monitoramento) exibe uma visão geral das métricas relacionadas à integridade do seu ambiente. Isso inclui o conjunto padrão de métricas fornecidas pelo Elastic Load Balancing e o Amazon EC2 e os gráficos que mostram como a integridade do ambiente mudou ao longo do tempo.

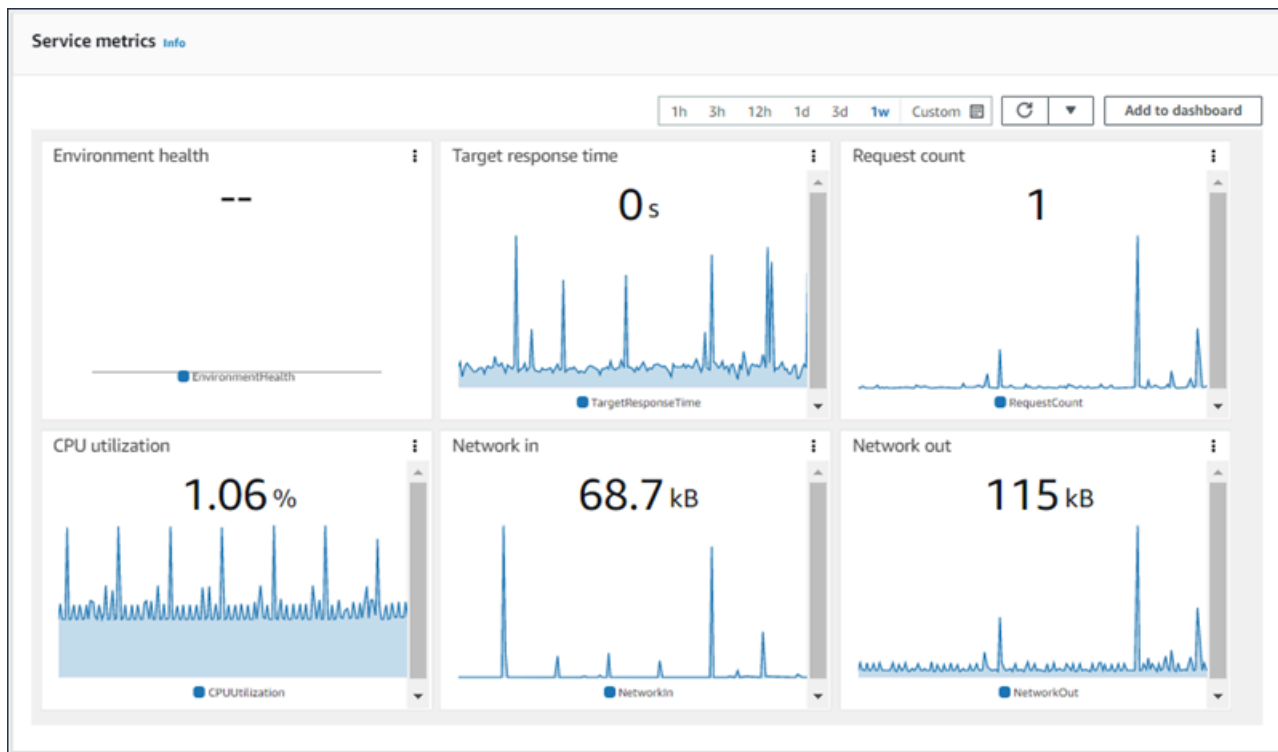
A barra acima dos gráficos fornece uma variedade de intervalos de tempo para você selecionar. Por exemplo, selecione 1w para exibir informações que abrangem a última semana. Ou selecione 3h para exibir informações que abrangem as últimas três horas.

Para uma maior variedade de seleções de intervalo de tempo, escolha Custom (Personalizar). A partir daqui, você tem duas opções de intervalo: Absoluto ou Relativo. A opção Absolute permite que você especifique um intervalo de datas específico, como 1º de janeiro de 2023 a 30 de junho de 2023. A opção Relativo permite selecionar um número inteiro com uma unidade de tempo específica: minutos, horas, dias, semanas ou meses. Os exemplos incluem 10 horas, 10 dias e 10 meses.



## Personalizar o console de monitoramento

Para criar e visualizar métricas personalizadas, você deve usar o Amazon CloudWatch. Com o CloudWatch você pode criar painéis para monitorar seus recursos em uma única exibição. Selecione **Add to dashboard** (Adicionar ao painel) para navegar até o console do Amazon CloudWatch a partir da página de Monitoring (Monitoramento). O Amazon CloudWatch oferece a opção de criar um novo painel ou selecionar um existente. Para obter mais informações, consulte [Usar painéis do Amazon CloudWatch](#) no Manual do usuário do Amazon CloudWatch.



As métricas do [Elastic Load Balancing](#) e do [Amazon EC2](#) estão habilitadas para todos os ambientes.

Com [integridade aprimorada](#), a métrica EnvironmentHealth é habilitada e um gráfico é adicionado automaticamente ao console de monitoramento. A integridade aprimorada também acrescenta a [página Integridade](#) ao console de gerenciamento. Para obter uma lista das métricas de integridade aprimorada, consulte [Publicar métricas personalizadas do Amazon CloudWatch para um ambiente](#).

## Relatórios de integridade básica

AWS Elastic Beanstalk usa informações de várias fontes para determinar se seu ambiente está disponível e processando solicitações da Internet. A integridade de um ambiente é representada por uma das quatro cores e é exibida na página de [visão geral do ambiente](#) do console do Elastic Beanstalk. Também está disponível na [DescribeEnvironmentsAPI](#) e por meio de chamadas eb status com a [CLI do EB](#).

Antes das versões da plataforma Linux versão 2, o único sistema de relatórios de integridade era o básico. O sistema básico de relatórios de integridade fornece informações sobre a integridade das instâncias em um ambiente do Elastic Beanstalk com base em verificações de integridade realizadas pelo Elastic Load Balancing para ambientes com carga balanceada ou pelo Amazon Elastic Compute Cloud para ambientes de instância única.

Além de verificar a integridade de suas instâncias do EC2, o Elastic Beanstalk também monitora outros recursos em seu ambiente e relata recursos ausentes ou configurados incorretamente que podem fazer com que seu ambiente se torne indisponível para os usuários.

As métricas coletadas pelos recursos em seu ambiente são publicadas CloudWatch na Amazon em intervalos de cinco minutos. Isso inclui as métricas de sistema operacional do EC2 e as métricas de solicitação do Elastic Load Balancing. Você pode visualizar gráficos com base nessas CloudWatch métricas na [página Monitoramento](#) do console do ambiente. Para integridade básica, essas métricas não são usadas para determinar a integridade de um ambiente.

## Tópicos

- [Cores de integridade](#)
- [Verificações de integridade do Elastic Load Balancing](#)
- [Verificações de integridade do ambiente de nível do operador e instância única](#)
- [Verificações adicionais](#)
- [CloudWatch Métricas da Amazon](#)

## Cores de integridade

O Elastic Beanstalk relata a integridade de um ambiente de servidor Web de acordo com o modo como a aplicação em execução responde à verificação de integridade. O Elastic Beanstalk usa uma de quatro cores para descrever o status, como mostrado na tabela a seguir:

Cor	Descrição
Cinza	O ambiente está sendo atualizado.
Verde	O ambiente passou na verificação de integridade mais recente. Pelo menos uma instância em seu ambiente está disponível e aceitando solicitações.
Amarelo	Seu ambiente falhou em uma ou mais verificações de integridade. Algumas solicitações a seu ambiente estão falhando.
Vermelha	Seu ambiente falhou em três ou mais verificações de integridade, ou um recurso do ambiente ficou indisponível. As solicitações falham consistentemente.

Essas descrições se aplicam somente a ambientes que usam relatórios de integridade básica. Consulte [Status e cores de integridade](#) para obter detalhes relacionados à integridade aprimorada.

## Verificações de integridade do Elastic Load Balancing

Em um ambiente com carga balanceada, o Elastic Load Balancing envia uma solicitação para cada instância em um ambiente a cada 10 segundos para confirmar que as instâncias estão íntegras. Por padrão, o load balancer é configurado para abrir uma conexão TCP na porta 80. Se a instância reconhece a conexão, ela é considerada íntegra.

Você pode optar por substituir essa configuração especificando um recurso existente no seu aplicativo. Se você especificar um caminho, como `/health`, o URL da verificação de integridade é definido como `HTTP:80/health`. O URL da verificação de integridade deve ser definido como um caminho que sempre é atendido pelo seu aplicativo. Se estiver definido como uma página estática que é atendida ou armazenada em cache pelo servidor web na frente de seu aplicativo, as verificações de integridade não revelarão problemas com o servidor de aplicativo ou o contêiner da Web. Para obter instruções sobre como modificar o URL de verificação de integridade, consulte [Verificação de integridade](#).

Se o URL de uma verificação de integridade está configurado, o Elastic Load Balancing espera uma solicitação GET que ele envia para retornar a resposta `200 OK`. O aplicativo na passará na verificação de integridade se não responder dentro de 5 segundos ou se responder com qualquer outro código de status HTTP. Após 5 falhas consecutivas na verificação de integridade, o Elastic Load Balancing tira a instância de serviço.

Para obter mais informações sobre verificações de integridade do Elastic Load Balancing, consulte [Verificação de integridade](#) no Guia do usuário do Elastic Load Balancing

### Note

A configuração de um URL de verificação de integridade não altera o comportamento de verificação de integridade de um grupo de Auto Scaling de um ambiente. Uma instância não íntegra é removida do balanceador de carga, mas não é substituída automaticamente pelo Amazon EC2 Auto Scaling, a menos que você o configure para usar a verificação de integridade do Elastic Load Balancing como base para a substituição de instâncias. Para configurar o Amazon EC2 Auto Scaling para substituir instâncias que falham em uma verificação de integridade do Elastic Load Balancing, consulte [Configuração da verificação de integridade do Auto Scaling](#).

## Verificações de integridade do ambiente de nível do operador e instância única

Em uma única instância ou ambiente de nível de operador, o Elastic Beanstalk determina a integridade da instância monitorando seu status de instância do Amazon EC2. As configurações de integridade do Elastic Load Balancing, incluindo URLs de verificação de integridade HTTP, não podem ser usadas nesses tipos de ambiente.

Para obter mais informações sobre verificações de status de instâncias do Amazon EC2, consulte [Monitorar instâncias com verificações de status](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

### Verificações adicionais

Além das verificações de integridade do Elastic Load Balancing, o Elastic Beanstalk monitora recursos no ambiente e altera o status de integridade para vermelho caso eles não sejam implantados, não estiverem configurados corretamente ou ficarem indisponíveis. Essas verificações confirmam que:

- O grupo de Auto Scaling do ambiente está disponível e tem no mínimo uma instância.
- O security group do ambiente está disponível e configurado para permitir o tráfego de entrada na porta 80.
- O CNAME do ambiente existe e está apontando para o load balancer correto.
- Em um ambiente de operador, a fila do Amazon Simple Queue Service (Amazon SQS) está sendo pesquisada pelo menos uma vez a cada três minutos.

## CloudWatch Métricas da Amazon

Com relatórios básicos de saúde, o serviço Elastic Beanstalk não publica nenhuma métrica na Amazon CloudWatch. As métricas usadas para produzir gráficos na [página Monitoramento](#) do console do ambiente são publicadas pelos recursos do seu ambiente.

Por exemplo, o EC2 publica as métricas a seguir para as instâncias no grupo de Auto Scaling de seu ambiente:

### CPUUtilization

Porcentagem de unidades de computação em uso no momento.

**DiskReadBytes, DiskReadOps, DiskWriteBytes, DiskWriteOps**

O número de bytes lidos e gravados e o número de operações de leitura e gravação.

**NetworkIn, NetworkOut**

O número de bytes enviados e recebidos.

O Elastic Load Balancing publica as métricas a seguir para o balanceador de carga do ambiente:

**BackendConnectionErrors**

Número de falhas de conexão entre o load balancer e as instâncias do ambiente.

**HTTPCode\_Backend\_2XX, HTTPCode\_Backend\_4XX**

Número de códigos de resposta de êxito (2XX) e erro do cliente (4XX) gerados pelas instâncias em seu ambiente.

**Latency**

O número de segundos entre o momento em que o load balancer retransmite uma solicitação para uma instância e o momento em que a resposta é recebida.

**RequestCount**

O número de solicitações concluídas.

Essas listas não são abrangentes. Para obter uma lista completa das métricas que podem ser relatadas para esses recursos, consulte os seguintes tópicos no Amazon CloudWatch Developer Guide:

**Metrics**

Namespace	Tópico
AWS::ElasticLoadBalancing::LoadBalancer	<a href="#">Métricas e recursos do Elastic Load Balancing</a>
AWS::AutoScaling::AutoScalingGrupo	<a href="#">Métricas e recursos do Amazon Elastic Compute Cloud</a>
AWS::SQS::Queue	<a href="#">Métricas e recursos do Amazon SQS</a>



Namespace	Tópico
AWS::RDS::DBInstance	<a href="#">Métricas e dimensões do Amazon RDS</a>

## Métrica de integridade do ambiente de operador

Somente para ambientes de trabalho, o daemon SQS publica uma métrica personalizada para a integridade do ambiente CloudWatch, em que um valor de 1 é verde. Você pode revisar os dados da métrica de CloudWatch saúde em sua conta usando o ElasticBeanstalk/SQSD namespace. A dimensão da métrica é EnvironmentName e o nome da métrica é Health. Todas as instâncias publicam suas métricas no mesmo namespace.

Para permitir que o daemon publique métricas, o perfil de instância do ambiente deve ter permissão para chamar `cloudwatch:PutMetricData`. Essa permissão está incluída no perfil de instância padrão. Para ter mais informações, consulte [Gerenciar perfis de instância do Elastic Beanstalk](#).

## Monitoramento e relatório de integridade aprimorada


Os relatórios de integridade aprimorada são um recurso que você pode habilitar no seu ambiente para permitir que o AWS Elastic Beanstalk obtenha informações adicionais sobre os recursos em seu ambiente. O Elastic Beanstalk analisa as informações reunidas para fornecer uma visão melhor da integridade geral do ambiente e ajudar na identificação de problemas que podem fazer com que a aplicação se torne indisponível.

Além das alterações na forma como as cores de integridade funcionam, a integridade aprimorada adiciona um descritor de status que fornece um indicador da gravidade dos problemas observados quando um ambiente está amarelo ou vermelho. Quando mais informações sobre o status atual estiverem disponíveis, você poderá escolher o botão Causes para ver informações detalhadas de integridade na [página de integridade](#).

Elastic Beanstalk > Environments > GettingStartedApp-env

**Elastic Beanstalk is updating your environment.**  
To cancel this operation select **Abort Current Operation** from the **Actions** dropdown.  
[View Events](#)

**GettingStartedApp-env** Refresh  
GettingStartedApp-env.bx7dx222kw.us-east-2.elasticbeanstalk.com

**Health**  
  
Info  
Causes

**Running version**  
Sample Application  
Upload and deploy

Tomcat (64b)

**Recent events**

Time	Type	Details
2020-01-28 15:16:51 UTC-0800	INFO	Deploying new version to instance(s).
2020-01-28 15:16:47 UTC-0800	INFO	Environment update is starting.
2020-01-28 12:11:17 UTC-0800	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 9

Para fornecer informações de integridade detalhadas sobre as instâncias do Amazon EC2 em execução no ambiente, o Elastic Beanstalk inclui um [agente de integridade](#) na Imagem de máquina da Amazon (AMI) para cada versão de plataforma compatível com a integridade aprimorada. O agente de integridade monitora os logs de servidor da Web e as métricas do sistema, e os transmite para o serviço do Elastic Beanstalk. O Elastic Beanstalk analisa essas métricas e dados do Elastic Load Balancing e do Amazon EC2 Auto Scaling para fornecer uma visão geral da integridade de um ambiente.

Além de coletar e apresentar informações sobre os recursos de ambiente, o Elastic Beanstalk monitora os recursos no ambiente em busca de várias condições de erro e fornece notificações para ajudar você a evitar falhas e resolver problemas de configuração. [Os fatores que influenciam a](#)

[integridade do seu ambiente](#) incluem os resultados de cada solicitação atendida pelo seu aplicativo, as métricas do sistema operacional de suas instâncias e o status da implantação mais recente.

É possível visualizar o status de integridade em tempo real usando a página [visão geral do ambiente](#) do console do Elastic Beanstalk ou o comando [eb health](#) na [Interface de Linhas de Comando do Elastic Beanstalk](#) (CLI do EB). Para registrar e controlar a integridade de ambientes e instâncias ao longo do tempo, é possível configurar seu ambiente para publicar as informações obtidas pelo Elastic Beanstalk para os relatórios de integridade aprimorada no Amazon CloudWatch como métricas personalizadas. As [cobranças](#) do CloudWatch pelas métricas personalizadas se aplicam a todas as métricas exceto a EnvironmentHealth, que é gratuita.

Os relatórios de integridade aprimorada exigem uma versão 2 ou mais recente da [versão de plataforma](#). Para monitorar os recursos e publicar as métricas, o ambiente deve ter um [perfil de instância e uma função de serviço](#). A plataforma do Docker de vários contêineres não inclui um servidor web por padrão, mas poderá ser usada com relatórios de integridade aprimorada se você configurar o servidor web para [fornecer logs no formato apropriado](#).

#### Notas da plataforma Windows

- Esse recurso não está disponível em [versões da plataforma Windows Server](#) anteriores à versão 2 (v2).
- Ao habilitar os relatórios de integridade aprimorada em um ambiente do Windows Server, não altere a [configuração de log do IIS](#). Para o monitoramento de integridade aprimorada funcionar corretamente, o registro do IIS deve ser configurado com o formato W3C e os destinos de evento de log ETW event only (Evento ETW somente) ou Both log file and ETW event (Arquivo de log e evento ETW).

Além disso, não desabilite nem interrompa o serviço do Windows do [agente de integridade do Elastic Beanstalk](#) em nenhuma das instâncias do ambiente. Para coletar e relatar informações de integridade aprimorada em uma instância, esse serviço deve estar habilitado e em execução.

A integridade aprimorada requer que o ambiente tenha um perfil de instância. O perfil de instância deve ter funções que forneçam permissões para que as instâncias do ambiente colem e relatem informações sobre a integridade aprimorada. Ao criar pela primeira vez um ambiente com uma versão de plataforma v2 no console do Elastic Beanstalk, o Elastic Beanstalk solicita que as funções necessárias sejam criadas e habilita os relatórios de integridade avançada por padrão. Continue

lendo para obter detalhes sobre como os relatórios de integridade aprimorada funcionam ou consulte [Habilitar relatórios de integridade aprimorada do Elastic Beanstalk](#) para começar a usá-los imediatamente.

As plataformas do Amazon Linux 2 exigem perfis de instância, para que possam dar suporte incondicional à integridade aprimorada. Quando é criado um ambiente usando uma plataforma do Amazon Linux 2, o Elastic Beanstalk sempre habilita a integridade aprimorada. Isso se aplica independentemente de como você cria o ambiente: usando o console do Elastic Beanstalk, a EB CLI, a AWS CLI, ou a API.

## Tópicos

- [Agente de integridade do Elastic Beanstalk](#)
- [Fatores que determinam a integridade de instâncias e ambientes](#)
- [Personalização de regra de verificação de integridade](#)
- [Funções de integridade aperfeiçoada](#)
- [Autorização de integridade aprimorada](#)
- [Eventos de integridade aprimorada](#)
- [Comportamento dos relatórios de integridade aprimorada durante as atualizações, as implantações e a escalabilidade](#)
- [Habilitar relatórios de integridade aprimorada do Elastic Beanstalk](#)
- [Monitoramento de integridade aprimorada com o console de gerenciamento do ambiente](#)
- [Status e cores de integridade](#)
- [Métricas de instância](#)
- [Configurar as regras de integridade aperfeiçoada de um ambiente](#)
- [Publicar métricas personalizadas do Amazon CloudWatch para um ambiente](#)
- [Usar relatórios de integridade aprimorada com a API do Elastic Beanstalk](#)
- [Formato de log de integridade aprimorado](#)
- [Notificações e solução de problemas](#)

## Agente de integridade do Elastic Beanstalk

O agente de integridade do Elastic Beanstalk é um processo daemon (ou serviço, em ambientes do Windows) que é executado em cada instância do Amazon EC2 no ambiente, monitorando as métricas de integridade em nível de sistema operacional e aplicação e relatando os problemas para o

Elastic Beanstalk. O agente de integridade está incluído em todas as versões da plataforma, a partir da versão 2.0 de cada plataforma.

O agente de saúde relata métricas semelhantes às [publicadas no CloudWatch](#) pelo Amazon EC2 Auto Scaling e Elastic Load Balancing como parte dos [relatórios básicos de integridade](#), incluindo carga de CPU, códigos HTTP e latência. O agente de integridade, no entanto, relata diretamente ao Elastic Beanstalk, com maior granularidade e frequência do que os relatórios de integridade básica.

Para a integridade básica, essas métricas são publicados a cada cinco minutos e podem ser monitoradas com gráficos no console de gerenciamento do ambiente. Com a integridade aprimorada, o agente de integridade do Elastic Beanstalk relata métricas para o Elastic Beanstalk a cada 10 segundos. O Elastic Beanstalk usa as métricas fornecidas pelo agente de integridade para determinar o status de integridade de cada instância no ambiente e, combinadas com outros [fatores](#), para determinar a integridade geral do ambiente.

A integridade geral do ambiente pode ser visualizada em tempo real na página de visão geral do ambiente do console do Elastic Beanstalk e é publicada no CloudWatch pelo Elastic Beanstalk a cada 60 segundos. É possível visualizar as métricas detalhadas relatadas pelo agente de integridade em tempo real com o comando [eb health](#) na [CLI do EB](#).

Mediante um custo adicional, você pode optar por publicar métricas individuais em nível de instância e ambiente no CloudWatch a cada 60 segundos. As métricas publicadas no CloudWatch podem ser usadas para criar [gráficos de monitoramento](#) no [console de gerenciamento de ambiente](#).

Os relatórios de integridade aprimorada gerarão cobrança apenas se você optar por publicar métricas de integridade aprimorada no CloudWatch. Quando você usa a integridade aprimorada, ainda tem as métricas de integridade básica publicadas gratuitamente, mesmo se optar por não publicar as métricas de integridade aprimorada.

Consulte [Métricas de instância](#) para obter detalhes sobre as métricas relatadas pelo agente de integridade. Para obter detalhes sobre a publicação de métricas de integridade aprimorada no CloudWatch, consulte [Publicar métricas personalizadas do Amazon CloudWatch para um ambiente](#).

## Fatores que determinam a integridade de instâncias e ambientes

Além das verificações do sistema de relatórios de integridade básica, incluindo o [Verificações de integridade do Elastic Load Balancing](#) e o [monitoramento de recursos](#), os relatórios de integridade aprimorada do Elastic Beanstalk coletam dados adicionais sobre o estado das instâncias no ambiente. Isso inclui métricas de sistema operacional, logs de servidor e o estado de operações

contínuas de ambiente, como implantações e atualizações. O serviço de relatório de integridade do Elastic Beanstalk combina informações de todas as origens disponíveis e as analisa para determinar a integridade geral do ambiente.

## Operações e comandos

Quando você executa uma operação no ambiente, como a implantação de uma nova versão de uma aplicação, o Elastic Beanstalk faz várias alterações que afetam o status de integridade do ambiente.

Por exemplo, quando você implanta uma nova versão de um aplicativo em um ambiente que está executando várias instâncias, você pode ver mensagens semelhantes à seguinte ao monitorar a integridade do ambiente [com a CLI do EB](#).

```

id status cause
Overall Info Command is executing on 3 out of 5 instances
i-bb65c145 Pending 91 % of CPU is in use. 24 % in I/O wait
 Pending Performing application deployment (running for 31 seconds)
i-ba65c144 Pending Performing initialization (running for 12 seconds)
i-f6a2d525 Ok Application deployment completed 23 seconds ago and took 26
seconds
i-e8a2d53b Pending 94 % of CPU is in use. 52 % in I/O wait
 Pending Performing application deployment (running for 33 seconds)
i-e81cca40 Ok

```

Neste exemplo, o status geral do ambiente é Ok e a causa desse status é que o comando está sendo executado em 3 de 5 instâncias. Três das instâncias no ambiente têm o status Pendente, indicando que há uma operação em andamento.

Quando uma operação é concluída, o Elastic Beanstalk relata as informações adicionais sobre ela. Por exemplo, o Elastic Beanstalk exibe as informações a seguir sobre uma instância que já foi atualizada com a nova versão da aplicação:

```

i-f6a2d525 Ok Application deployment completed 23 seconds ago and took 26
seconds

```

As informações de integridade da instância também incluem detalhes sobre a implantação mais recente a cada instância em seu ambiente. Cada instância relata um ID de implantação e o status. O ID de implantação é um número inteiro que aumenta em um a cada vez que você implanta uma nova versão do aplicativo ou altera as definições das opções de configuração na instância, como as

variáveis de ambiente. Você pode usar as informações de implantação para identificar as instâncias que estão executando a versão incorreta de seu aplicativo após uma falha [de implantação contínua](#).

Na coluna de causa, o Elastic Beanstalk inclui mensagens informativas sobre operações bem-sucedidas e outros estados de integridade em várias verificações de integridade, mas elas não são mantidas indefinidamente. As causas para status de não integridade no ambiente são mantidas até o ambiente retornar a um status de integridade.

## Tempo de espera do comando

O Elastic Beanstalk aplica um tempo de espera do comando a partir do momento que uma operação começa para permitir que uma instância faça a transição para um estado de integridade. Esse tempo limite do comando é definido na configuração de implantação e atualização do ambiente (no namespace [aws:elasticbeanstalk:command](#)), sendo 10 minutos o padrão.

Durante atualizações contínuas, o Elastic Beanstalk aplica um tempo de espera separado para cada lote na operação. Esse tempo de espera é definido como parte da configuração de atualização contínua do ambiente (no namespace [aws:autoscaling:updatepolicy:rollingupdate](#)). Se todas as instâncias no lote estiverem íntegras dentro do tempo limite de atualização contínua, a operação continuará no próximo lote. Se não, a operação falha.

### Note

Caso a aplicação não passe nas verificações de integridade com um status OK, mas esteja estável em outro nível, você poderá definir a opção `HealthCheckSuccessThreshold` no [aws:elasticbeanstalk:command namespace](#) para alterar o nível no qual o Elastic Beanstalk considera uma instância como íntegra.

Para que um ambiente de servidor web seja considerado íntegro, cada instância no ambiente ou no lote deve passar por 12 verificações de integridade consecutivas ao longo de dois minutos. Para um ambiente do nível de operador, cada instância deve passar por 18 verificações de integridade. Antes de o comando atingir o tempo limite, o Elastic Beanstalk não reduz o status de integridade de um ambiente quando a verificação de integridade falha. Se as instâncias no ambiente se tornarem íntegras dentro do tempo limite do comando, a operação será bem-sucedida.

## Solicitações HTTP

Quando nenhuma operação está em andamento em um ambiente, a origem principal de informações sobre a integridade do ambiente e da instância são os logs do servidor web de cada instância. Para

determinar a integridade de uma instância e a integridade geral do ambiente, o Elastic Beanstalk considera o número de solicitações, o resultado de cada uma e a velocidade com que cada solicitação foi resolvida.

Em plataformas Linux, o Elastic Beanstalk lê e analisa logs do servidor Web para obter informações sobre solicitações HTTP. Na plataforma Windows Server, o Elastic Beanstalk recebe essas informações [diretamente do servidor Web do IIS](#).

O ambiente pode não ter um servidor web ativo. Por exemplo, a plataforma do Docker de vários contêineres não inclui um servidor web. Outras plataformas incluem um servidor web e o aplicativo pode desabilitá-lo. Nesses casos, o ambiente exige configuração adicional para fornecer ao [agente de integridade do Elastic Beanstalk](#) os logs no formato necessário para que as informações de integridade sejam retransmitidas ao serviço do Elastic Beanstalk. Para mais detalhes, consulte [Formato de log de integridade aprimorado](#).

## Métricas do sistema operacional

O Elastic Beanstalk monitora as métricas do sistema operacional relatadas pelo agente de integridade para identificar as instâncias que são consistentemente baixas em recursos do sistema.

Consulte [Métricas de instância](#) para obter detalhes sobre as métricas relatadas pelo agente de integridade.

## Personalização de regra de verificação de integridade

Os relatórios de integridade aprimorada do Elastic Beanstalk dependem de um conjunto de regras para determinar a integridade do ambiente. Algumas dessas regras podem não ser adequadas para o seu aplicativo. Um caso comum é um aplicativo que retorna erros HTTP 4xx frequentes por padrão. O Elastic Beanstalk, usando uma de suas regras padrão, conclui que algo está errado e altera o status de integridade do ambiente de OK para Warning (Aviso), Degraded (Reduzido) ou Severe (Grave), de acordo com a taxa de erros. Para lidar com esse caso corretamente, o Elastic Beanstalk permite configurar essa regra e ignorar erros de HTTP 4xx da aplicação. Para obter mais detalhes, consulte [Configurar as regras de integridade aperfeiçoada de um ambiente](#).

## Funções de integridade aperfeiçoada

Os relatórios de integridade aperfeiçoada requerem duas funções: uma função de serviço para o Elastic Beanstalk e um perfil de instância para o ambiente. A função de serviço permite que o Elastic Beanstalk interaja com outros serviços da AWS em seu nome para coletar informações sobre os



recursos no ambiente. O perfil de instância permite que as instâncias no ambiente gravem logs no Amazon S3 e comuniquem informações de integridade aprimoradas ao serviço do Elastic Beanstalk.

Quando você cria um ambiente do Elastic Beanstalk usando o console do Elastic Beanstalk ou a CLI do EB, o Elastic Beanstalk cria uma função de serviço padrão e anexa as políticas gerenciadas necessárias a um perfil de instância padrão para o ambiente.

Se você usar a API, um SDK ou a AWS CLI para criar ambientes, deverá criar essas funções antecipadamente e especificá-las durante a criação do ambiente para usar integridade aprimorada. Para obter instruções sobre a criação de funções adequadas para seus ambientes, consulte [Funções de serviço, perfis de instância e políticas de usuário](#).

Recomendamos que você use políticas gerenciadas para o perfil de instância e a função de serviço. As políticas gerenciadas são políticas do AWS Identity and Access Management (IAM) mantidas pelo Elastic Beanstalk. O uso de políticas gerenciadas garante que o ambiente tenha todas as permissões necessárias para funcionar corretamente.

Para o perfil de instância, você pode usar a política gerenciada `AWSElasticBeanstalkWebTier` ou `AWSElasticBeanstalkWorkerTier`, para um ambiente de [nível de servidor Web](#) ou de [nível de operador](#), respectivamente. Para obter detalhes sobre essas duas políticas gerenciadas de perfil de instância, consulte [the section called “Perfis de instância”](#).

## Autorização de integridade aprimorada

As políticas gerenciadas do perfil da instância do Elastic Beanstalk contêm permissão para a ação `elasticbeanstalk:PutInstanceStatistics`. Essa ação não faz parte da API do Elastic Beanstalk. Ela faz parte de uma API diferente que as instâncias de ambiente usam internamente para transmitir informações de integridade aprimoradas para o serviço do Elastic Beanstalk. Você não chama essa API diretamente.

Quando você cria um novo ambiente, a autorização para a ação `elasticbeanstalk:PutInstanceStatistics` é habilitada por padrão. Para aumentar a segurança do ambiente e ajudar a impedir a falsificação de dados de integridade em seu nome, recomendamos manter a autorização para esta ação habilitada. Se você usar políticas gerenciadas para seu perfil da instância, esse recurso estará disponível para seu novo ambiente sem necessidade de configuração adicional. Se você usar um perfil da instância personalizado em vez de uma política gerenciada, o ambiente poderá mostrar o status de integridade Sem dados. Isso acontece porque as instâncias não estão autorizadas para a ação que comunica dados de integridade aprimorados ao serviço.

Para autorizar a ação, inclua a instrução a seguir no perfil de instância.

```
{
 "Sid": "ElasticBeanstalkHealthAccess",
 "Action": [
 "elasticbeanstalk:PutInstanceStatistics"
],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:*:*:application/*",
 "arn:aws:elasticbeanstalk:*:*:environment/*"
]
}
```

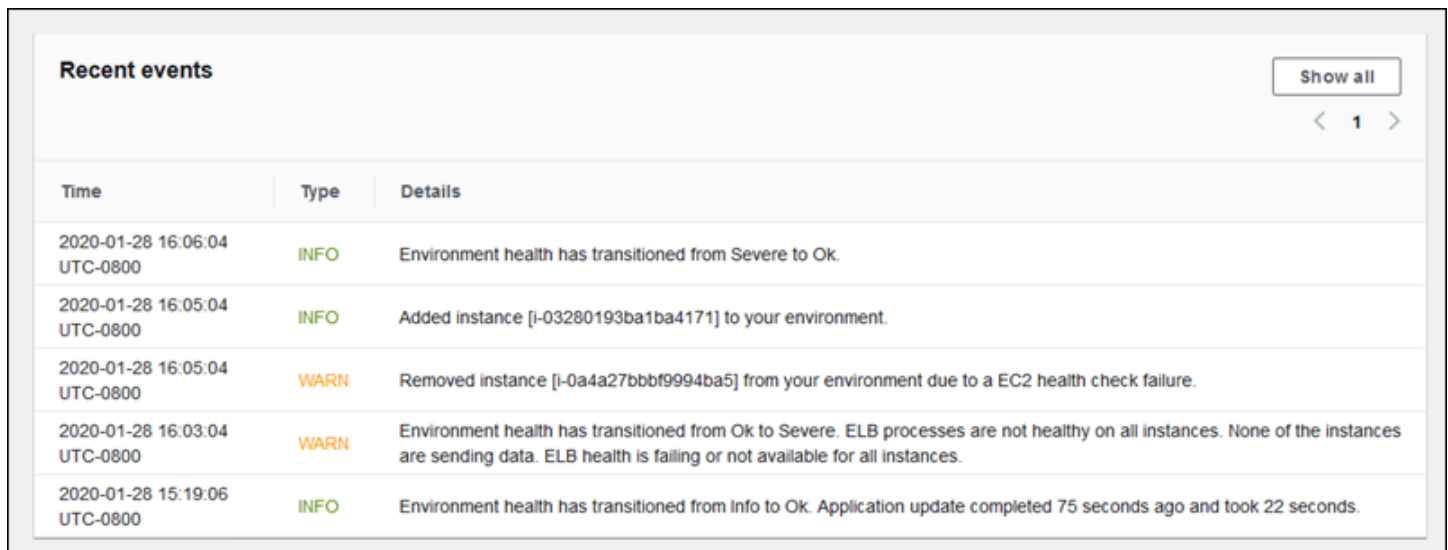
Se você não quiser usar a autorização de integridade aprimorada no momento, desabilite-a definindo a opção `EnhancedHealthAuthEnabled` no namespace [the section called “aws:elasticbeanstalk:healthreporting:system”](#) como `false`. Se a opção estiver desabilitada, as permissões descritas anteriormente não são necessárias. Você pode removê-las do perfil de instância para [acesso mínimo](#) a suas aplicações e ambientes.

#### Note

Anteriormente, a configuração padrão para `EnhancedHealthAuthEnabled` foi `false`, que resultou na autorização para a ação `elasticbeanstalk:PutInstanceStatistics` também ser desabilitada por padrão. Para habilitar essa ação para um ambiente existente, defina a opção `EnhancedHealthAuthEnabled` no [the section called “aws:elasticbeanstalk:healthreporting:system”](#) namespace to `true`. É possível configurar essa opção usando uma [configuração de opção](#) em um [arquivo de configuração](#).

## Eventos de integridade aprimorada

O sistema de integridade aprimorada gera eventos quando um ambiente alterna entre estados. O exemplo a seguir mostra eventos gerados pela transição de um ambiente entre os estados Info (Informações), OK e Severe (Grave).



Time	Type	Details
2020-01-28 16:06:04 UTC-0800	INFO	Environment health has transitioned from Severe to Ok.
2020-01-28 16:05:04 UTC-0800	INFO	Added instance [i-03280193ba1ba4171] to your environment.
2020-01-28 16:05:04 UTC-0800	WARN	Removed instance [i-0a4a27bbf9994ba5] from your environment due to a EC2 health check failure.
2020-01-28 16:03:04 UTC-0800	WARN	Environment health has transitioned from Ok to Severe. ELB processes are not healthy on all instances. None of the instances are sending data. ELB health is failing or not available for all instances.
2020-01-28 15:19:06 UTC-0800	INFO	Environment health has transitioned from Info to Ok. Application update completed 75 seconds ago and took 22 seconds.

Quando ocorre uma transição para um estado pior, o evento de integridade aprimorada inclui uma mensagem indicando a causa da transição.

Nem todas as alterações no status em um nível de instância fazem com que o Elastic Beanstalk gere um evento. Para evitar alarmes falsos, o Elastic Beanstalk gerará um evento relacionado à integridade somente se um problema persistir em várias verificações.

As informações de integridade em tempo real no nível do ambiente, incluindo status, cor e causa, estão disponíveis na página de [visão geral do ambiente](#) do console do Elastic Beanstalk e na [CLI do EB](#). Ao anexar a CLI do EB ao ambiente e executar o comando [eb health](#), você também poderá visualizar o status em tempo real de cada uma das instâncias no ambiente.

## Comportamento dos relatórios de integridade aprimorada durante as atualizações, as implantações e a escalabilidade

Habilitar os relatórios de integridade aprimorada pode afetar como seu ambiente se comporta durante atualizações e implantações de configuração. O Elastic Beanstalk não concluirá um lote de atualizações até que todas as instâncias passem nas verificações de integridade de forma consistente. Além disso, como os relatórios de integridade aprimorada aplicam um critério mais rigoroso de integridade e monitoram mais fatores, as instâncias que passam na [verificação de integridade do ELB](#) dos relatórios de integridade básica não passarão necessariamente com os relatórios de integridade aprimorada. Consulte os tópicos sobre [atualizações de configuração contínuas](#) e [implantações contínuas](#) para obter detalhes sobre como as verificações de integridade afetam o processo de atualização.

Os relatórios de integridade aprimorada também podem destacar a necessidade de definir um [URL de verificação de integridade](#) adequado para o Elastic Load Balancing. Quando seu ambiente expandir para atender à demanda, novas instâncias começarão a atender solicitações assim que passarem por verificações de integridade do ELB suficientes. Se não for configurado um URL de verificação de integridade, isso poderá levar apenas 20 segundos após uma nova instância ser capaz de aceitar uma conexão TCP.

Se o seu aplicativo não tiver concluído a inicialização até o momento em que o load balancer o declarar íntegro o suficiente para receber tráfego, você verá um afluxo de solicitações falhadas e seu ambiente começará a não passar nas verificações de integridade. Um URL de verificação de integridade que atinge um caminho fornecido pelo aplicativo pode evitar esse problema. As verificações de integridade do ELB não passarão até que uma solicitação GET para a URL de verificação de integridade retorne um código de status 200.

## Habilitar relatórios de integridade aprimorada do Elastic Beanstalk

Os novos ambientes criados com as [versões de Plataforma](#) mais recentes incluem o [agente de integridade do AWS Elastic Beanstalk](#), que é compatível com relatórios de integridade aprimorada. Se você criar o ambiente no console do Elastic Beanstalk ou com a CLI do EB, a integridade aprimorada será habilitada por padrão. Também é possível definir a preferência de relatórios de integridade no código-fonte do aplicativo usando [arquivos de configuração](#).

Os relatórios de integridade aprimorada exigem o [perfil da instância](#) e a [função de serviço](#) com o conjunto padrão de permissões. Quando você cria um ambiente no console do Elastic Beanstalk, o Elastic Beanstalk cria as funções necessárias automaticamente. Para obter instruções sobre como criar seu primeiro ambiente, consulte [Conceitos básicos do Elastic Beanstalk](#).

### Tópicos

- [Habilitar relatórios de integridade aprimorada usando o console do Elastic Beanstalk](#)
- [Habilitar relatórios de integridade aprimorada usando a CLI do EB](#)
- [Habilitar relatórios de integridade aprimorada usando um arquivo de configuração](#)

## Habilitar relatórios de integridade aprimorada usando o console do Elastic Beanstalk

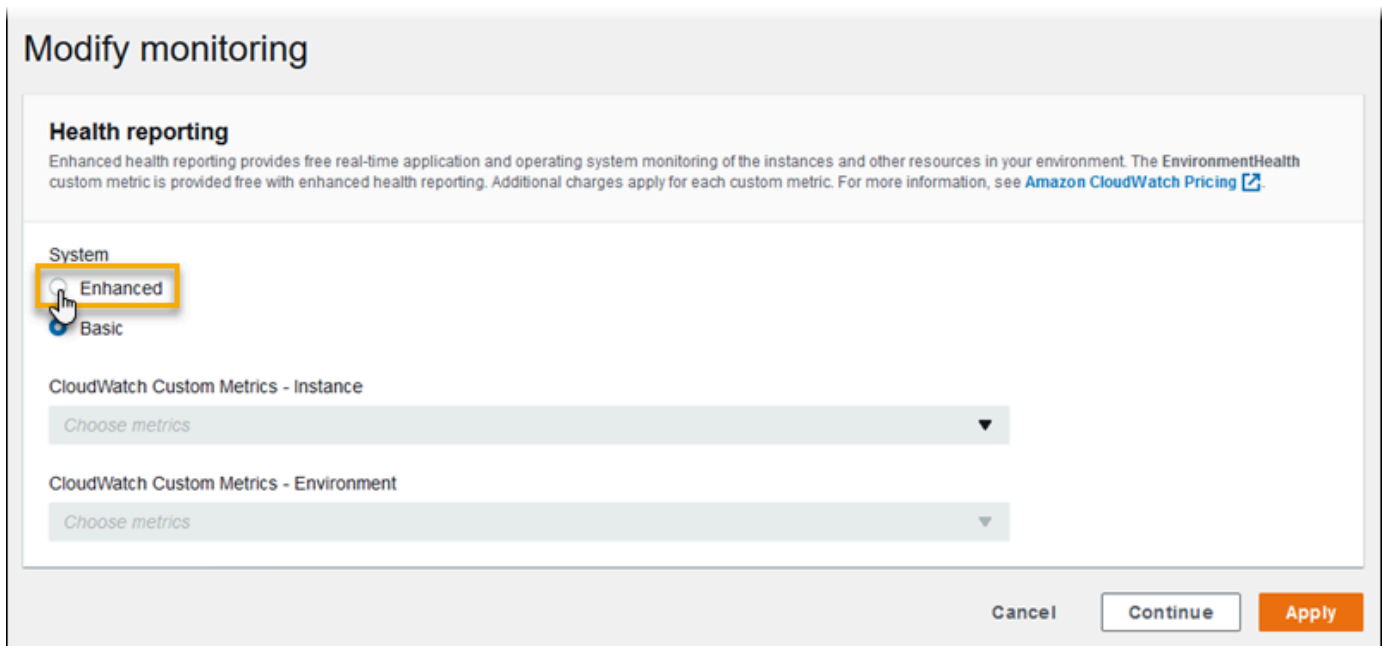
Como habilitar relatórios de integridade aprimorada em um ambiente em execução usando o console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Monitoring (Monitoramento), escolha Edit (Editar).
5. Em Health reporting (Relatórios de integridade), para System (Sistema), escolha Enhanced (Avançado).



### Note

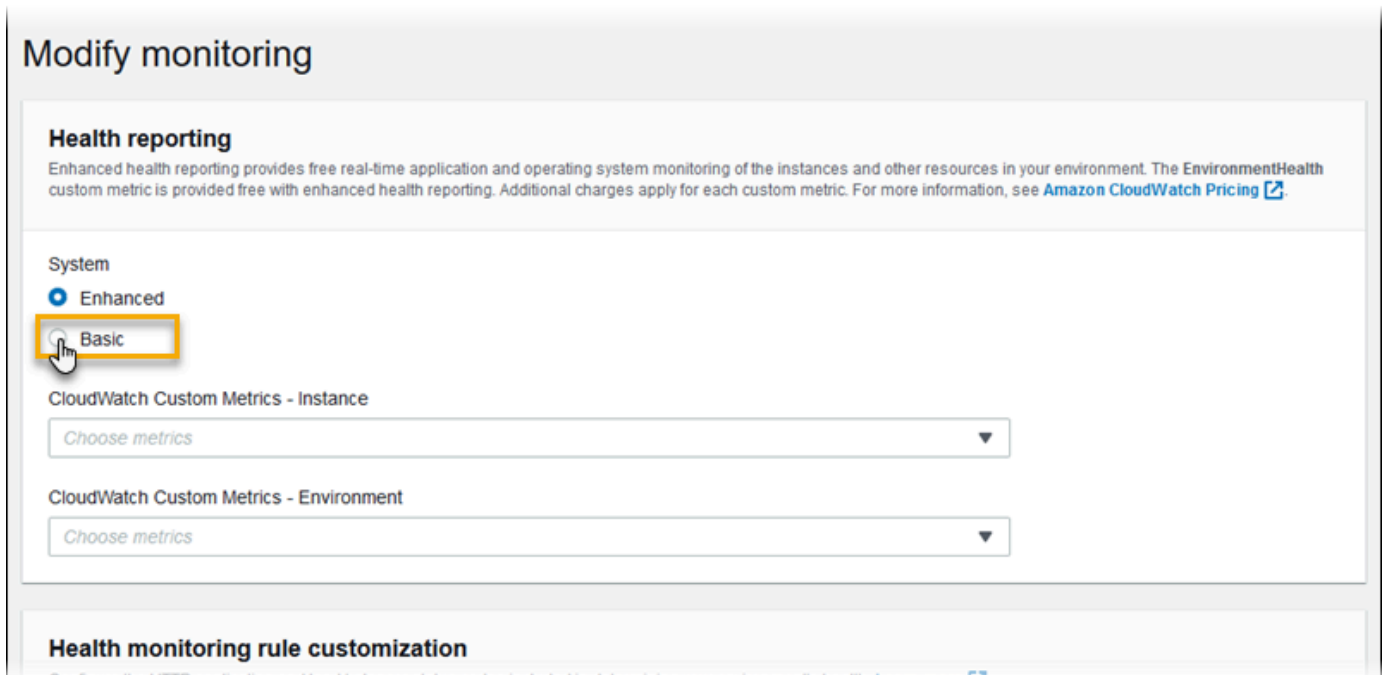
As opções de relatórios de integridade avançados não serão exibidas se você estiver usando uma [Plataforma ou versão incompatível](#).

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

O console do Elastic Beanstalk usa como padrão os relatórios de integridade aprimorada ao criar um novo ambiente com uma versão de plataforma de versão 2 (v2). Você pode desabilitar os relatórios de integridade aprimorada alterando a respectiva opção durante a criação do ambiente.

Como desabilitar os relatórios de integridade aprimorada durante a criação de um ambiente usando o console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. [Crie um aplicativo](#) ou selecione um que já exista.
3. [Crie um ambiente](#). Na página Create a new environment (Criar um novo ambiente), antes de escolher Create environment (Criar ambiente), escolha Configure more options (Configurar mais opções).
4. Na categoria de configuração Monitoring (Monitoramento), escolha Edit (Editar).
5. Em Health reporting (Relatórios de integridade), para System (Sistema), escolha Basic (Básico).



6. Escolha Save (Salvar).

## Habilitar relatórios de integridade aprimorada usando a CLI do EB

Quando você cria um novo ambiente com o comando `eb create`, a EB CLI habilita relatórios de integridade aprimorada, por padrão, e aplica a função de serviço e o perfil da instância padrão.

É possível especificar outra função de serviço por nome usando a opção `--service-role`.

Se você tem um ambiente em execução com relatórios de integridade básica da versão de plataforma v2 e quer mudar para a integridade aprimorada, siga estas etapas.

Para habilitar a integridade aprimorada no ambiente em execução usando a [EB CLI](#)

1. Use o comando `eb config` para abrir o arquivo de configuração no editor de texto padrão.

```
~/project$ eb config
```

2. Localize o namespace `aws:elasticbeanstalk:environment` na seção de configurações. Certifique-se de que o valor de `ServiceRole` não seja nulo e que corresponda ao nome da sua [função de serviço](#).

```
aws:elasticbeanstalk:environment:
 EnvironmentType: LoadBalanced
 ServiceRole: aws-elasticbeanstalk-service-role
```

3. No namespace `aws:elasticbeanstalk:healthreporting:system:`, altere o valor de `SystemType` para **enhanced**.

```
aws:elasticbeanstalk:healthreporting:system:
 SystemType: enhanced
```

4. Salve o arquivo de configuração e feche o editor de texto.
5. A EB CLI inicia uma atualização do ambiente para aplicar as alterações de configuração. Aguarde a conclusão da operação ou pressione `Ctrl+C` para sair com segurança.

```
~/project$ eb config
Printing Status:
INFO: Environment update is starting.
INFO: Health reporting type changed to ENHANCED.
INFO: Updating environment no-role-test's configuration settings.
```

## Habilitar relatórios de integridade aprimorada usando um arquivo de configuração

Habilite os relatórios de integridade aprimorada, incluindo um [arquivo de configuração](#) no pacote de origem. O exemplo a seguir mostra um arquivo de configuração que habilita relatórios de integridade aprimorada e atribui o perfil da instância e serviço padrão ao ambiente:

Example `.ebextensions/enhanced-health.config`

```
option_settings:
 aws:elasticbeanstalk:healthreporting:system:
 SystemType: enhanced
 aws:autoscaling:launchconfiguration:
 IamInstanceProfile: aws-elasticbeanstalk-ec2-role
 aws:elasticbeanstalk:environment:
 ServiceRole: aws-elasticbeanstalk-service-role
```

Se você criou seu próprio perfil de instância ou função de serviço, substitua o texto destacado pelos nomes dessas funções.

## Monitoramento de integridade aprimorada com o console de gerenciamento do ambiente

Ao habilitar os relatórios de integridade avançada no AWS Elastic Beanstalk, será possível monitorar a integridade do ambiente no [console de gerenciamento do ambiente](#).

### Tópicos

- [Visão geral do ambiente](#)
- [Página de integridade do ambiente](#)
- [Página de monitoramento](#)

### Visão geral do ambiente

A [visão geral do ambiente](#) exibe o [status da integridade](#) do ambiente e lista eventos que fornecem informações sobre alterações recentes no status da integridade.

### Como visualizar a visão geral do ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.

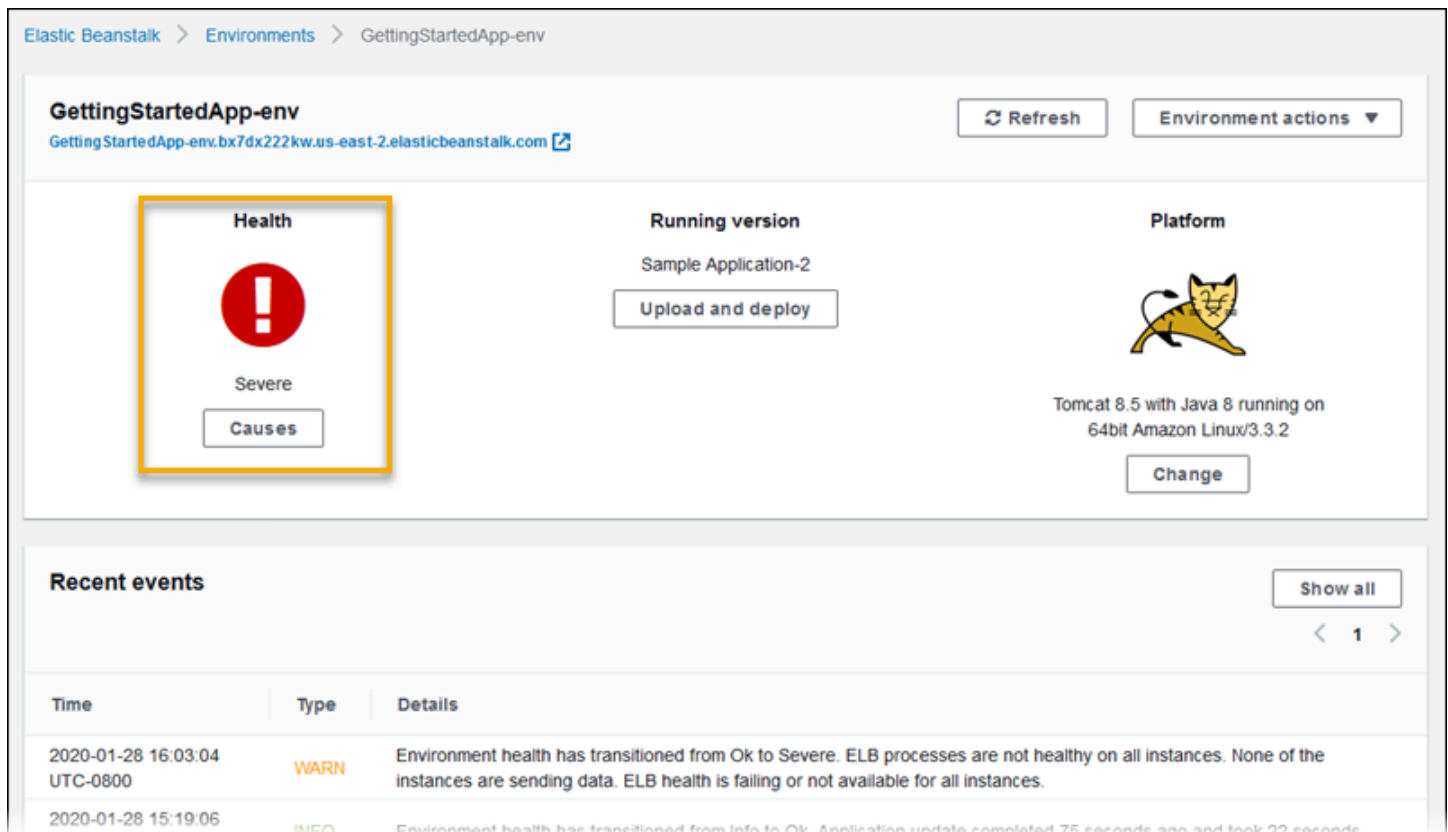


2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

**Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

Para obter informações detalhadas sobre a integridade do ambiente atual, abra a página Health (Integridade) selecionando Causes (Causas). Como alternativa, no painel de navegação, escolha Health (Integridade).



Elastic Beanstalk > Environments > GettingStartedApp-env

**GettingStartedApp-env**  
GettingStartedApp-env.bx7dx222kw.us-east-2.elasticbeanstalk.com

Refresh Environment actions

**Health**  
Severe  
Causes

**Running version**  
Sample Application-2  
Upload and deploy

**Platform**  
Tomcat 8.5 with Java 8 running on 64bit Amazon Linux/3.3.2  
Change

**Recent events** Show all

Time	Type	Details
2020-01-28 16:03:04 UTC-0800	WARN	Environment health has transitioned from Ok to Severe. ELB processes are not healthy on all instances. None of the instances are sending data. ELB health is failing or not available for all instances.
2020-01-28 15:19:06	INFO	Environment health has transitioned from Info to Ok. Application update completed 75 seconds ago and took 22 seconds.

## Página de integridade do ambiente

A página Health (Integridade) exibe o status da integridade, as métricas e as causas relativos ao ambiente e a cada instância do Amazon EC2 no ambiente.

**Note**

O Elastic Beanstalk exibirá a página Health (Integridade) somente se você tiver [habilitado o monitoramento de integridade aprimorada](#) para o ambiente.

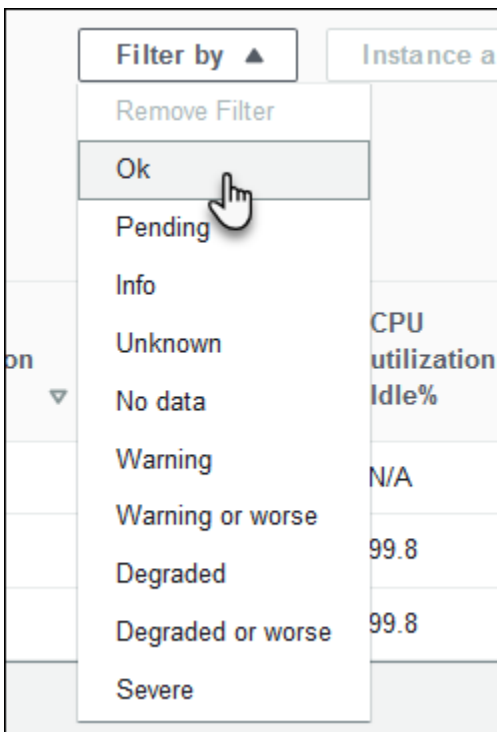
A imagem a seguir mostra a página Health (Integridade) para um ambiente Linux.

Instance ID	Status	Running	Deployment ID	Requests/sec	2xx Responses	3xx Responses	4xx Responses	5xx Responses	P99 Latency	P90 Latency	P75 Latency	P50 Latency	P10 Latency	Load1 average	Load5 average	CPU utilization User%	CPU utilization Sys%	CPU utilization Idle%	CPU utilization I/O wait%
Overall	Ok	N/A	N/A	0.4	100%	0.0%	0.0%	0.0%	0.002	0.002	0.002	0.002	0.001	N/A	N/A	N/A	N/A	N/A	N/A
i-00227807c4c4a1334	Ok	2 hours	3	0.2	2	0	0	0	0.002	0.002	0.002	0.002	0.002	0.00	0.00	0.0	0.0	99.9	0.0
i-03280193ba1ba4171	Ok	19 days	3	0.2	2	0	0	0	0.001	0.001	0.001	0.001	0.001	0.00	0.00	0.1	0.0	99.9	0.0

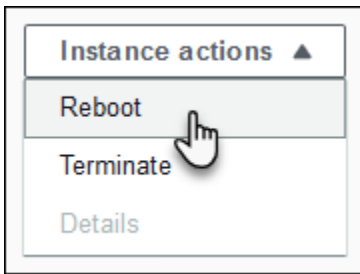
A imagem a seguir mostra a página Health (Integridade) para um ambiente Windows. Observe que as métricas de CPU são diferentes das contidas em um ambiente Linux.

Instance ID	Status	Running	Deployment ID	Requests/sec	2xx Responses	3xx Responses	4xx Responses	5xx Responses	P99 Latency	P90 Latency	P75 Latency	P50 Latency	P10 Latency	CPU utilization % User Time	CPU utilization % Privileged Time	CPU utilization % Idle Time
Overall	Ok	N/A	N/A	0.2	100%	0.0%	0.0%	0.0%	0.015	0.014	0.011	0.008	0.002	N/A	N/A	N/A
i-04b33b4c983018af	Ok	29 days	1	0.2	2	0	0	0	0.015	0.014	0.011	0.008	0.002	0.0	0.0	100

Na parte superior da página, é possível ver o número total de instâncias de ambiente, bem como o número de instâncias por status. Para exibir somente as instâncias que têm um status específico, escolha Filter By (Filtrar por) e selecione um [status](#).



Para reinicializar ou encerrar uma instância não íntegra, escolha Instance Actions e, em seguida, Reboot ou Terminate.



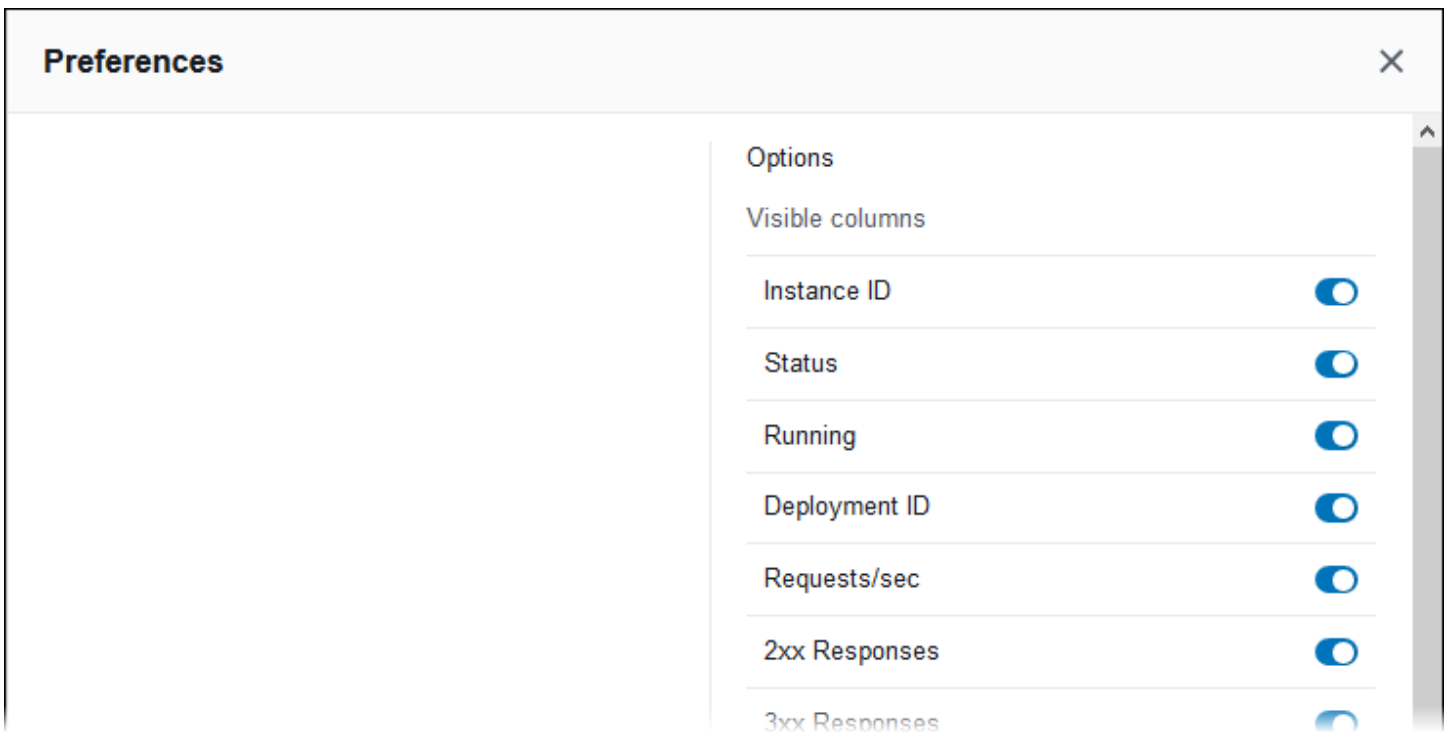
O Elastic Beanstalk atualiza a página Health (Integridade) a cada 10 segundos. Ela relata informações sobre o ambiente e a integridade da instância.

Para cada instância do Amazon EC2 no ambiente, a página exibe o ID e o [status](#) da instância, o tempo desde que a instância foi executada, o ID da implantação mais recente executada na instância, as respostas e a latência das solicitações atendidas pela instância e informações de carga e utilização da CPU. A linha Overall (Geral) exibe informações de médias de resposta e latência para todo o ambiente.

A página exibe muitos detalhes em uma tabela muito ampla. Para ocultar algumas das colunas, escolha



(Preferences (Preferências)). Selecione ou desmarque os nomes das colunas e escolha Confirm (Confirmar).



Escolha o Instance ID (ID da instância) de qualquer instância para exibir mais informações sobre a instância, incluindo a zona de disponibilidade e o tipo de instância.

	Instance ID ▾	Status ▲	Running ▾	Deployment ID ▾	Reque
●	Overall	Ok	N/A	N/A	0.2
○	i-00227807c4c4a1334	Ok	1 day	3	0.1
○	i-03280193ba1ba4171	Ok	20 days	3	0.1



**i-00227807c4c4a1334 details** ×

---

Instance ID: i-00227807c4c4a1334  
Instance type: t2.micro  
Availability zone: us-east-2b

Escolha o Deployment ID (ID de implantação) de qualquer instância para exibir informações sobre a última [implantação](#) na instância.

	Instance ID ▾	Status ▲	Running ▾	Deployment ID ▾	Reque
●	Overall	Ok	N/A	N/A	0.2
○	i-00227807c4c4a1334	Ok	1 day	3	0.1
○	i-03280193ba1ba4171	Ok	20 days	3	0.1



### Deployment details

Deployment ID 3  
Version: Sample Application-3  
**Deployed** 1 day ago

As informações de implantação incluem:

- ID da implantação: o identificador exclusivo da [implantação](#). Os IDs de implantação começam em 1 e são incrementados em uma unidade sempre que você implantar uma nova versão do aplicativo ou alterar as configurações que afetam o software ou o sistema operacional em execução nas instâncias em seu ambiente.
- Versão: o rótulo da versão do código-fonte da aplicação usada na implantação.
- Status: o status da implantação, que pode ser In Progress, Deployed ou Failed.
- Horário: para implantações em andamento, o horário em que a implantação começou. Para implantações concluídas, o horário em que a implantação terminou.

Se você [habilitar a integração com o X-Ray](#) no ambiente e equipar sua aplicação com o SDK do AWS X-Ray, a página Integridade adicionará links para o console do AWS X-Ray na linha de visão geral.

Requests/sec ▾	2xx Responses ▾	3xx Responses ▾	4xx Responses ▾	5xx Responses ▾	P99 Latency ▾	P90 Latency ▾	P75 Latency ▾	P50 Latency ▾	P10 Latency ▾	Loss rate
100%	0.0%	0.0%	0.0%	0.0%	0.002	0.002	0.002	0.002	0.001	N/A
1	0	0	0	0	0.002	0.002	0.002	0.002	0.002	0.01
1	0	0	0	0	0.001	0.001	0.001	0.001	0.001	0.00

Escolha um link para visualizar rastreamentos relacionadas à estatística realçada no console do AWS X-Ray.

## Página de monitoramento

A página Monitoring (Monitoramento) exibe estatísticas resumidas e gráficos das métricas personalizadas do Amazon CloudWatch gerados pelo sistema de relatórios de integridade aprimorada. Consulte [Monitoramento da integridade do ambiente no Console de Gerenciamento da AWS](#) para obter instruções sobre como adicionar gráficos e estatísticas a essa página.

## Status e cores de integridade

Relatórios de integridade aprimorada representam a integridade geral do ambiente e da instância usando quatro cores, semelhantes aos [relatórios de integridade básica](#). Os relatórios de integridade aprimorada também fornecem sete status de integridade, que são descritores de palavra única que fornecem uma melhor indicação do estado do ambiente.

## Status da instância e status do ambiente

Toda vez que o Elastic Beanstalk executa uma verificação de integridade no ambiente, os relatórios de integridade aprimorada verificam a integridade de cada instância do ambiente, analisando todos [os dados](#) disponíveis. Se alguma verificação de nível inferior falhar, o Elastic Beanstalk fará downgrade da integridade da instância.

O Elastic Beanstalk exibe as informações de integridade do ambiente geral (cor, status e causa) no [console de gerenciamento do ambiente](#). Essas informações também estão disponíveis na EB CLI. As mensagens de causa e status de integridade de instâncias individuais são atualizadas a cada dez segundos e estão disponíveis a partir da [CLI do EB](#) quando você visualiza o status de integridade com [eb health](#).

O Elastic Beanstalk usa as alterações de integridade da instância para avaliar a integridade do ambiente, mas não altera imediatamente o status de integridade do ambiente. Quando uma instância não passa nas verificações de integridade pelo menos três vezes no período de um minuto, o Elastic

Beanstalk pode fazer downgrade da integridade do ambiente. Dependendo do número de instâncias no ambiente e do problema identificado, uma instância não íntegra pode fazer com que o Elastic Beanstalk exiba uma mensagem informativa ou altere o status de integridade do ambiente de verde (OK) para amarelo (Warning (Aviso)) ou vermelho (Degraded (Reduzido) ou Severe (Grave)).

## OK (verde)

Esse status é exibido quando:

- Uma instância está passando nas verificações de integridade e o agente de integridade não está encontrando nenhum problema.
- A maioria das instâncias do ambiente está passando nas verificações de integridade e o agente de integridade não está encontrando problemas importantes.
- Uma instância está passando nas verificações de integridade e está concluindo as solicitações normalmente.

Exemplo: o ambiente foi implantado recentemente e está recebendo solicitações normalmente. Cinco por cento das solicitações estão retornando erros da série 400. Implantação concluída normalmente em cada instância.

Mensagem (instância): implantação do aplicativo concluída há 23 segundos e levou 26 segundos.

## Aviso (amarelo)

Esse status é exibido quando:

- O agente de integridade está encontrando um número moderado de falhas de solicitação ou outros problemas em uma instância ou ambiente.
- Uma operação está em andamento em uma instância e está demorando muito.

Exemplo: uma instância no ambiente tem um status de Severe (Grave).

Mensagem (ambiente): serviços com problemas em 1 de 5 instâncias.

## Reduzido (vermelho)

Esse status é exibido quando o agente de integridade está encontrando um número alto de falhas de solicitação ou outros problemas em uma instância ou ambiente.

Exemplo: o ambiente está no processo de aumentar para até 5 instâncias.

Mensagem (ambiente): 4 instâncias ativas estão abaixo do tamanho mínimo 5 do grupo de Auto Scaling.

## Grave (vermelho)

Esse status é exibido quando o agente de integridade está encontrando um número muito alto de falhas de solicitação ou outros problemas em uma instância ou ambiente.

Exemplo: o Elastic Beanstalk não consegue entrar em contato com o balanceador de carga para obter a integridade da instância.

Mensagem (ambiente): a integridade do ELB está falhando ou não está disponível para todas as instâncias. Nenhuma das instâncias está enviando dados. Não foi possível assumir a função "arn:aws:iam::123456789012:role/aws-elasticbeanstalk-service-role". Verifique se a função existe e se está configurada corretamente.

Mensagem (instâncias): a integridade do ELB da instância não está disponível há 37 minutos. Não há dados. Visto pela última vez 37 minutos atrás.

## Informações (verde)

Esse status é exibido quando:

- Uma operação está em andamento em uma instância.
- Uma operação está em andamento em várias instâncias de um ambiente.

Exemplo: uma nova versão do aplicativo está sendo implantada nas instâncias em execução.

Mensagem (ambiente): o comando está sendo executado em 3 de 5 instâncias.

Mensagem (instância): realizando a implantação do aplicativo (em execução por 3 segundos).

## Pendente (cinza)

Esse status é exibido quando uma operação está em andamento em uma instância dentro do [tempo limite do comando](#).

Exemplo: você criou recentemente o ambiente e as instâncias estão sendo inicializadas.

Mensagem: realizando a inicialização (em execução por 12 segundos).



## Desconhecido (cinza)

Esse status é exibido quando o Elastic Beanstalk e o agente de integridade estão relatando uma quantidade insuficiente de dados em uma instância.

Exemplo: não há dados sendo recebidos.

## Suspenso (cinza)

Esse status é exibido quando o Elastic Beanstalk interrompe o monitoramento da integridade do ambiente. O ambiente pode não funcionar corretamente. Se algumas condições de integridade graves durarem muito tempo, o Elastic Beanstalk fará a transição do ambiente para o status Suspended (Suspenso).

Exemplo: o Elastic Beanstalk não pode acessar a [função de serviço](#) do ambiente.

Exemplo: o [grupo de Auto Scaling](#) criado pelo Elastic Beanstalk para o ambiente foi excluído.

Mensagem: a integridade do ambiente mudou de OK para Severe (Grave). Não há instâncias listadas. A capacidade desejada do grupo do Auto Scaling está definida como 1.

## Métricas de instância

As métricas de instância fornecem informações sobre a integridade das instâncias em seu ambiente. O [agente de integridade do Elastic Beanstalk](#) é executado em cada instância. Ele reúne e transmite novamente métricas sobre instâncias para o Elastic Beanstalk, que as analisa para determinar a integridade das instâncias em seus ambientes.

O agente de integridade do Elastic Beanstalk na instância reúne métricas sobre instâncias de servidores Web e o sistema operacional. Para obter informações sobre o servidor Web em plataformas Linux, o Elastic Beanstalk lê e analisa os logs do servidor Web. Na plataforma Windows Server, o Elastic Beanstalk recebe essas informações diretamente do servidor Web do IIS. Os servidores web fornecem informações sobre solicitações HTTP recebidas: quantas solicitações chegaram, quantas resultaram em erros e quanto tempo levaram para serem resolvidas. O sistema operacional fornece informações do snapshot sobre o estado dos recursos das instâncias: a carga de CPU e a distribuição de tempo gasto em cada tipo de processo.

O agente de integridade reúne métricas de sistema operacional e servidor Web e as transmite novamente para o Elastic Beanstalk a cada 10 segundos. O Elastic Beanstalk analisa os dados e usa os resultados para atualizar o status de integridade de cada instância e o ambiente.

## Tópicos

- [Métricas do servidor web](#)
- [Métricas do sistema operacional](#)
- [Métricas do servidor web capturadas no IIS no Windows Server](#)

## Métricas do servidor web

Em plataformas baseadas no Linux, o agente de integridade do Elastic Beanstalk lê métricas do servidor Web nos logs gerados pelo contêiner da Web ou pelo servidor que processa solicitações em cada instância no ambiente. As plataformas do Elastic Beanstalk são configuradas para gerar dois logs: um no formato legível por humanos e um no formato legível por máquina. O agente de integridade transmite logs legíveis por máquina para o Elastic Beanstalk a cada 10 segundos.

Para obter mais informações sobre o formato de log usado pelo Elastic Beanstalk, consulte [Formato de log de integridade aprimorado](#).

Na plataforma do Windows Server, o Elastic Beanstalk adiciona um módulo ao pipeline de solicitações do servidor Web do IIS e captura as métricas sobre os tempos de solicitação HTTP e os códigos de resposta. O módulo envia essas métricas para o agente de integridade na instância usando um canal IPC (comunicação entre processos) de alta performance. Para obter mais detalhes sobre a implementação, visite [Métricas do servidor web capturadas no IIS no Windows Server](#).

## Métricas do servidor web relatadas

### RequestCount

Número de solicitações atendidas pelo servidor Web por segundo nos últimos 10 segundos. Mostrado como uma média  $r/sec$  (solicitações por segundo) na CLI do EB e [Página de integridade do ambiente](#).

### Status2xx, Status3xx, Status4xx, Status5xx

Número de solicitações que resultaram em cada tipo de código de status nos últimos 10 segundos. Por exemplo, as solicitações bem-sucedidas retornam um 200 OK, redirecionamentos são 301, e se o URL inserido não corresponder a nenhum recurso no aplicativo, retornará 404.

A EB CLI e [Página de integridade do ambiente](#) mostram essas métricas como um número bruto de solicitações de instâncias e como uma porcentagem do total de solicitações de ambientes.

p99.9, p99, p95, p90, p85, p75, p50, p10

A latência média para x % mais lento de solicitações nos últimos 10 segundos, em que x é a diferença entre o número e 100. Por exemplo, p99 1.403 indica que 1% das solicitações mais lentas nos últimos 10 segundos tiveram uma latência média de 1,403 segundos.

## Métricas do sistema operacional

O agente de integridade do Elastic Beanstalk relata as métricas a seguir do sistema operacional. O Elastic Beanstalk usa essas métricas para identificar instâncias que estão sob carga pesada constante. As métricas diferem por sistema operacional.

### Métricas do sistema operacional relatadas — Linux

#### Running

O tempo que passou desde que a instância foi iniciada.

Load 1, Load 5

Média de carga nos últimos períodos de um minuto e cinco minutos. Mostrado como um valor decimal indicando o número médio de processos em execução durante esse período. Se o número mostrado for maior que o número de vCPUs (threads) disponíveis, o restante será o número médio de processos que estavam em espera.

Por exemplo, se o tipo de instância tem quatro vCPUs e a carga é de 4,5, houve uma média de 0,5 processos em espera durante aquele período, o que equivale a um processo que aguarda 50% do tempo.

User %, Nice %, System %, Idle %, I/O Wait %

Porcentagem de tempo que a CPU passou em cada estado nos últimos 10 segundos.

### Métricas do sistema operacional relatadas — Windows

#### Running

O tempo que passou desde que a instância foi iniciada.

% User Time, % Privileged Time, % Idle Time

Porcentagem de tempo que a CPU passou em cada estado nos últimos 10 segundos.

## Métricas do servidor web capturadas no IIS no Windows Server

Na plataforma do Windows Server, o Elastic Beanstalk adiciona um módulo ao pipeline de solicitações do servidor Web do IIS e captura as métricas sobre os tempos de solicitação HTTP e os códigos de resposta. O módulo envia essas métricas para o agente de integridade na instância usando um canal IPC (comunicação entre processos) de alta performance. O agente de integridade agrega essas métricas, combina-as com as métricas do sistema operacional e as envia para o serviço do Elastic Beanstalk.

### Detalhes da implantação

Para capturar as métricas do IIS, o Elastic Beanstalk implementa um [IHttpModule](#) gerenciado e assina os eventos [BeginRequest](#) e [EndRequest](#). Isso permite que o módulo relate a latência das solicitações HTTP e os códigos de resposta de todas as solicitações da web atendidas pelo IIS. Para adicionar o módulo ao pipeline de solicitações do IIS, o Elastic Beanstalk registra o módulo na seção [<modules>](#) do arquivo de configuração do IIS, `%windir%\System32\inetsrv\config\applicationHost.config`.

O módulo do Elastic Beanstalk no IIS envia as métricas capturadas da solicitação da Web ao agente de integridade na instância, que é um serviço Windows chamado HealthD. Para enviar esses dados, o módulo usa [NetNamedPipeBinding](#), que fornece uma vinculação segura e confiável que é otimizada para a comunicação na máquina.


## Configurar as regras de integridade aperfeiçoada de um ambiente

Os relatórios de integridade avançada do AWS Elastic Beanstalk dependem de um conjunto de regras para determinar a integridade do ambiente. Algumas dessas regras podem não ser adequadas para o seu aplicativo. Veja a seguir alguns exemplos comuns:

- Você usa ferramentas de teste no lado do cliente. Nesse caso, são esperados erros frequentes de cliente HTTP (4xx).
- Use o [AWS WAF](#) junto com o Application Load Balancer do ambiente para bloquear tráfego de entrada indesejado. Nesse caso, o Application Load Balancer retorna HTTP 403 para cada mensagem recebida rejeitada.

Por padrão, o Elastic Beanstalk inclui todos os erros HTTP 4xx da aplicação ao determinar a integridade do ambiente. Ele altera o status de integridade do ambiente de OK para Warning (Aviso), Degraded (Degradado) ou Severe (Grave), dependendo da taxa de erros. Para lidar corretamente

com casos como os exemplos que mencionamos, o Elastic Beanstalk permite configurar algumas regras de integridade aprimorada. É possível optar por ignorar os erros HTTP 4xx do aplicativo nas instâncias do ambiente ou ignorar os erros HTTP 4xx retornados pelo load balancer do ambiente. Este tópico descreve como fazer essas alterações de configuração.

 Note


No momento, essas são as únicas personalizações de regras de integridade avançada disponíveis. Não é possível configurar a integridade avançada para ignorar outros erros HTTP além do 4xx.

## Configurar regras de integridade aprimorada usando o console do Elastic Beanstalk

É possível usar o console do Elastic Beanstalk para configurar regras de integridade aprimorada no ambiente.

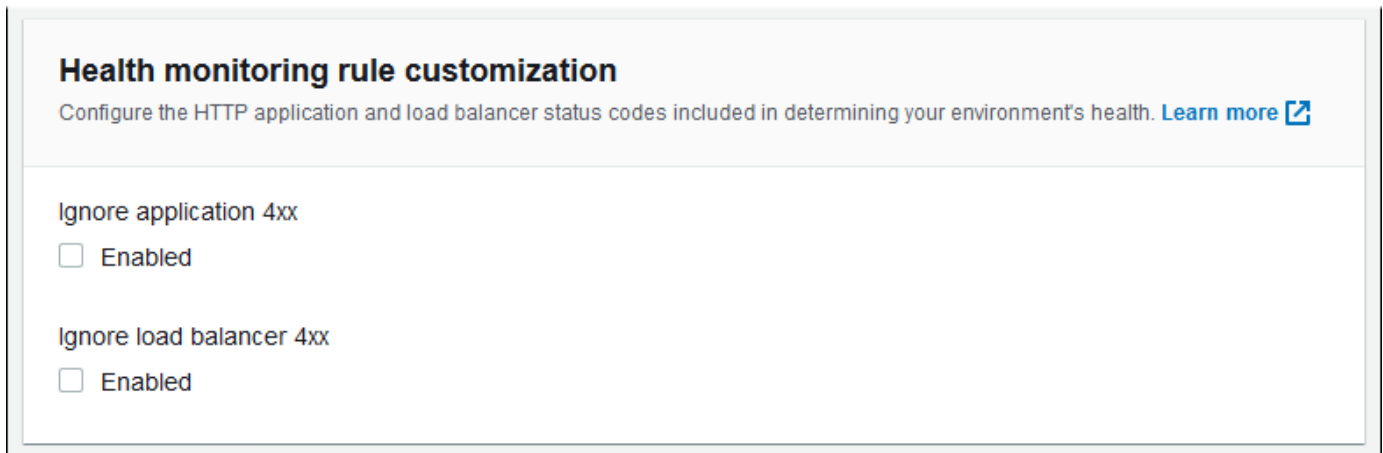
Como configurar a verificação de código de status HTTP 4xx usando o console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Monitoring (Monitoramento), escolha Edit (Editar).
5. Em Health monitoring rule customization (Personalização da regra de monitoramento de integridade), habilite ou desabilite as opções Ignore (Ignorar) desejadas.



6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

## Configurar regras de integridade aprimorada usando a EB CLI

Use a CLI do EB para configurar regras de integridade aprimorada, salvando a configuração do ambiente localmente, adicionando uma entrada que defina essas regras e fazendo upload da configuração no Elastic Beanstalk. Você pode aplicar a configuração salva a um ambiente durante ou após a criação.

Como configurar a verificação de código de status HTTP 4xx usando a CLI do EB e configurações salvas

1. Inicialize a pasta do projeto com [eb init](#).
2. Crie um ambiente executando o comando [eb create](#).
3. Salve um modelo de configuração localmente, executando o comando `eb config save`. O exemplo a seguir usa a opção `--cfg` para especificar o nome da configuração.

```
$ eb config save --cfg 01-base-state
Configuration saved at: ~/project/.elasticbeanstalk/saved_configs/01-base-state.cfg.yml
```

4. Abra o arquivo de configuração salvo em um editor de texto.
5. Em `OptionSettings > aws:elasticbeanstalk:healthreporting:system:`, adicione uma chave `ConfigDocument` para listar cada regra de integridade aprimorada a ser configurada. O `ConfigDocument` a seguir desabilita a verificação dos códigos de status HTTP 4xx do aplicativo, mantendo a verificação do código HTTP 4xx do load balancer habilitada.

```
OptionSettings:
 ...
 aws:elasticbeanstalk:healthreporting:system:
 ConfigDocument:
 Rules:
 Environment:
 Application:
 ApplicationRequests4xx:
 Enabled: false
 ELB:
 ELBRequests4xx:
 Enabled: true
 Version: 1
 SystemType: enhanced
 ...
```

#### Note

É possível combinar Rules e CloudWatchMetrics na mesma configuração da opção ConfigDocument. CloudWatchMetrics são descritas em [Publicar métricas personalizadas do Amazon CloudWatch para um ambiente](#).

Se você tiver habilitado CloudWatchMetrics anteriormente, o arquivo de configuração recuperado usando o comando `eb config save` já terá uma chave ConfigDocument com uma seção CloudWatchMetrics. Não o exclua, em vez disso, adicione uma seção Rules no mesmo valor da opção ConfigDocument.

6. Salve o arquivo de configuração e feche o editor de texto. Neste exemplo, o arquivo de configuração atualizado é salvo com um nome (`02-cloudwatch-enabled.cfg.yml`) diferente do arquivo de configuração obtido por download. Isso cria uma configuração salva separada quando o arquivo é carregado. Você pode usar o mesmo nome que o arquivo baixado para substituir a configuração existente, sem criar um novo.
7. Use o comando `eb config put` para fazer upload do arquivo de configuração atualizado no Elastic Beanstalk.

```
$ eb config put 02-cloudwatch-enabled
```

Ao usar os comandos `eb config`, `get` e `put` com configurações salvas, não inclua a extensão do nome do arquivo.

## 8. Aplique a configuração salva ao ambiente em execução.

```
$ eb config --cfg 02-cloudwatch-enabled
```

A opção `--cfg` especifica um arquivo de configuração nomeado que é aplicado ao ambiente. É possível salvar o arquivo de configuração localmente ou no Elastic Beanstalk. Se um arquivo de configuração com o nome especificado já existir em ambos os locais, a EB CLI usará o arquivo local.

## Configurar regras de integridade aprimorada usando um documento de configuração

O documento de configuração (config) para regras de integridade aprimorada é um documento JSON que lista as regras a serem configuradas.

O exemplo a seguir mostra um documento de configuração que desabilita a verificação de códigos de status HTTP 4xx do aplicativo e habilita a verificação de códigos de status HTTP 4xx do load balancer.

```
{
 "Rules": {
 "Environment": {
 "Application": {
 "ApplicationRequests4xx": {
 "Enabled": false
 }
 },
 "ELB": {
 "ELBRequests4xx": {
 "Enabled": true
 }
 }
 }
 },
 "Version": 1
}
```

Na AWS CLI, você passa o documento como um valor da chave `Value` em um argumento de configurações de opção, que é um objeto JSON. Nesse caso, você deve escapar as aspas no documento incorporado. O comando a seguir verifica se as definições de configuração são válidas.



```
$ aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings '[
 {
 "Namespace": "aws:elasticbeanstalk:healthreporting:system",
 "OptionName": "ConfigDocument",
 "Value": "{\\"Rules\\": { \\"Environment\\": { \\"Application\\":
{ \\"ApplicationRequests4xx\\": { \\"Enabled\\": false } }, \\"ELB\\": { \\"ELBRequests4xx\\":
{ \\"Enabled\\": true } } } }, \\"Version\\": 1 }"
 }
]'
```

Para um arquivo de configuração `.ebextensions` em YAML, você pode fornecer o documento JSON no estado em que está.

```
option_settings:
- namespace: aws:elasticbeanstalk:healthreporting:system
 option_name: ConfigDocument
 value: {
"Rules": {
 "Environment": {
 "Application": {
 "ApplicationRequests4xx": {
 "Enabled": false
 }
 },
 "ELB": {
 "ELBRequests4xx": {
 "Enabled": true
 }
 }
 }
},
"Version": 1
}
```

## Publicar métricas personalizadas do Amazon CloudWatch para um ambiente

Publique os dados reunidos pelo relatório de integridade aprimorada do AWS Elastic Beanstalk no Amazon CloudWatch como métricas personalizadas. A publicação de métricas do CloudWatch

permite monitorar alterações na performance da aplicação ao longo do tempo e identificar possíveis problemas rastreando como o uso de recursos e a latência de solicitação aumentam com carga.

Com a publicação de métricas no CloudWatch, você pode disponibilizá-las para uso em [gráficos de monitoramento](#) e [alarmes](#). Uma métrica gratuita, EnvironmentHealth, é habilitada automaticamente quando você usa relatórios de integridade aprimorada. As métricas personalizadas que não sejam a EnvironmentHealth geram [cobranças do CloudWatch](#) padrão.

Para publicar métricas personalizadas do CloudWatch de um ambiente, primeiro habilite os relatórios de integridade aprimorada no ambiente. Para obter instruções, consulte [Habilitar relatórios de integridade aprimorada do Elastic Beanstalk](#).

## Tópicos

- [Métricas de relatórios de integridade aprimorada](#)
- [Configurar métricas do CloudWatch usando o console do Elastic Beanstalk](#)
- [Configurar métricas personalizadas do CloudWatch usando a CLI do EB](#)
- [Fornecer documentos Config de métrica personalizada](#)

## Métricas de relatórios de integridade aprimorada

Quando você habilita os relatórios de integridade aprimorada no ambiente, esse sistema de relatórios publica automaticamente uma [métrica personalizada do CloudWatch](#), a EnvironmentHealth. Para publicar métricas adicionais no CloudWatch, configure seu ambiente com essas métricas usando o [console do Elastic Beanstalk](#), a [CLI da EB](#) ou [.ebextensions](#).

É possível publicar estas métricas de integridade aprimorada do seu ambiente para o CloudWatch.

Métricas disponíveis — todas as plataformas

### EnvironmentHealth

Ambiente apenas. Essa é a única métrica do CloudWatch publicada pelo sistema de relatórios de integridade aprimorada, a menos que você configure mais métricas. A integridade do ambiente é representada por um dos sete [status](#). No console do CloudWatch, esses status são mapeados para os seguintes valores:

- 0: OK
- 1: Informações

- 5: Desconhecido
- 10: Sem dados
- 15: Aviso
- 20: Reduzido
- 25: Grave

`InstancesSevere`, `InstancesDegraded`, `InstancesWarning`, `InstancesInfo`,  
`InstancesOk`, `InstancesPending`, `InstancesUnknown`, `InstancesNoData`

Ambiente apenas. Essas métricas indicam o número de instâncias no ambiente com cada status de integridade. `InstancesNoData` indica o número de instâncias para as quais não há dados recebidos.

`ApplicationRequestsTotal`, `ApplicationRequests5xx`, `ApplicationRequests4xx`,  
`ApplicationRequests3xx`, `ApplicationRequests2xx`

Instância e ambiente. Indica o número total de solicitações concluídas pela instância ou ambiente, e o número de solicitações que foram concluídas com cada categoria de código de status.

`ApplicationLatencyP10`, `ApplicationLatencyP50`, `ApplicationLatencyP75`,  
`ApplicationLatencyP85`, `ApplicationLatencyP90`, `ApplicationLatencyP95`,  
`ApplicationLatencyP99`, `ApplicationLatencyP99.9`

Instância e ambiente. Indica o tempo médio, em segundos, que leva para concluir com mais rapidez x % de solicitações.

`InstanceHealth`

Instância apenas. Indica o status da integridade atual da instância. A integridade da instância é representada por um dos sete [status](#). No console do CloudWatch, esses status são mapeados para os seguintes valores:

- 0: OK
- 1: Informações
- 5: Desconhecido
- 10: Sem dados
- 15: Aviso
- 20: Reduzido

- 25: Grave

## Métricas disponíveis — Linux

CPUirq, CPUIdle, CPUUser, CPUSystem, CPUsoftirq, CPUiowait, CPUNice

Instância apenas. Indica a porcentagem de tempo que a CPU passou em cada estado no último minuto.

LoadAverage1min

Instância apenas. A carga média de CPU da instância durante o último minuto.

RootFilesystemUtil

Instância apenas. Indica a porcentagem do espaço em disco em uso.

## Métricas disponíveis — Windows

CPUIdle, CPUUser, CPUPrivileged

Instância apenas. Indica a porcentagem de tempo que a CPU passou em cada estado no último minuto.

## Configurar métricas do CloudWatch usando o console do Elastic Beanstalk

É possível usar o console do Elastic Beanstalk para configurar o ambiente de modo a publicar métricas de relatórios de integridade aprimorada no CloudWatch e disponibilizá-las para uso em gráficos de monitoramento e alarmes.

Como configurar métricas personalizadas do CloudWatch no console do Elastic Beanstalk

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Monitoring (Monitoramento), escolha Edit (Editar).
5. Em Health reporting (Relatórios de integridade), selecione as métricas de instância e de ambiente a serem publicadas no CloudWatch. Para selecionar várias métricas, mantenha pressionada a tecla Ctrl enquanto escolhe.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Habilitar métricas personalizadas do CloudWatch as adiciona à lista de métricas disponíveis na página [Monitoring \(Monitoramento\)](#).

## Configurar métricas personalizadas do CloudWatch usando a CLI do EB

Use a CLI do EB para configurar métricas personalizadas, salvando a configuração do ambiente localmente, adicionando uma entrada que define as métricas para publicação e fazendo upload da configuração no Elastic Beanstalk. Você pode aplicar a configuração salva a um ambiente durante ou após a criação.

Como configurar métricas personalizadas do CloudWatch com a CLI do EB e configurações salvas

1. Inicialize a pasta do projeto com [eb init](#).
2. Crie um ambiente executando o comando [eb create](#).
3. Salve um modelo de configuração localmente, executando o comando `eb config save`. O exemplo a seguir usa a opção `--cfg` para especificar o nome da configuração.

```
$ eb config save --cfg 01-base-state
Configuration saved at: ~/project/.elasticbeanstalk/saved_configs/01-base-state.cfg.yml
```

4. Abra o arquivo de configuração salvo em um editor de texto.
5. Em `OptionSettings > aws:elasticbeanstalk:healthreporting:system:`, adicione uma chave `ConfigDocument` para habilitar cada uma das métricas do CloudWatch desejadas. Por exemplo, `ConfigDocument` publica as métricas `ApplicationRequests5xx` e `ApplicationRequests4xx` no nível de ambiente e as métricas `ApplicationRequestsTotal` no nível de instância.

```
OptionSettings:
 ...
 aws:elasticbeanstalk:healthreporting:system:
```

```
ConfigDocument:
 CloudWatchMetrics:
 Environment:
 ApplicationRequests5xx: 60
 ApplicationRequests4xx: 60
 Instance:
 ApplicationRequestsTotal: 60
 Version: 1
 SystemType: enhanced
...
```

No exemplo, 60 indica o número de segundos entre as medidas. No momento, este é o único valor aceito.

#### Note

É possível combinar `CloudWatchMetrics` e `Rules` na mesma configuração da opção `ConfigDocument`. `Rules` são descritas em [Configurar as regras de integridade aperfeiçoada de um ambiente](#).

Se você tiver usado `Rules` anteriormente para configurar as regras de integridade aprimorada, o arquivo de configuração recuperado por meio do comando `eb config save` já terá uma chave `ConfigDocument` com uma seção `Rules`. Não o exclua, em vez disso, adicione uma seção `CloudWatchMetrics` no mesmo valor da opção `ConfigDocument`.

6. Salve o arquivo de configuração e feche o editor de texto. Neste exemplo, o arquivo de configuração atualizado é salvo com um nome (`02-cloudwatch-enabled.cfg.yml`) que seja diferente do arquivo de configuração que foi baixado. Isso cria uma configuração salva separada quando o arquivo é carregado. Você pode usar o mesmo nome que o arquivo baixado para substituir a configuração existente, sem criar um novo.
7. Use o comando `eb config put` para fazer upload do arquivo de configuração atualizado no Elastic Beanstalk.

```
$ eb config put 02-cloudwatch-enabled
```

Ao usar os comandos `eb config get` e `eb config put` com configurações salvas, não inclua a extensão de arquivo.

8. Aplique a configuração salva ao ambiente em execução.

```
$ eb config --cfg 02-cloudwatch-enabled
```

A opção `--cfg` especifica um arquivo de configuração nomeado que é aplicado ao ambiente. É possível salvar o arquivo de configuração localmente ou no Elastic Beanstalk. Se um arquivo de configuração com o nome especificado já existir em ambos os locais, a EB CLI usará o arquivo local.

## Fornecer documentos Config de métrica personalizada

O documento de configuração (config) de métricas personalizadas do Amazon CloudWatch é um documento JSON que lista as métricas a serem publicadas nos níveis do ambiente e da instância. O exemplo a seguir mostra um documento config que habilita todas as métricas disponíveis no Linux.

```
{
 "CloudWatchMetrics": {
 "Environment": {
 "ApplicationLatencyP99.9": 60,
 "InstancesSevere": 60,
 "ApplicationLatencyP90": 60,
 "ApplicationLatencyP99": 60,
 "ApplicationLatencyP95": 60,
 "InstancesUnknown": 60,
 "ApplicationLatencyP85": 60,
 "InstancesInfo": 60,
 "ApplicationRequests2xx": 60,
 "InstancesDegraded": 60,
 "InstancesWarning": 60,
 "ApplicationLatencyP50": 60,
 "ApplicationRequestsTotal": 60,
 "InstancesNoData": 60,
 "InstancesPending": 60,
 "ApplicationLatencyP10": 60,
 "ApplicationRequests5xx": 60,
 "ApplicationLatencyP75": 60,
 "InstancesOk": 60,
 "ApplicationRequests3xx": 60,
 "ApplicationRequests4xx": 60
 },
 "Instance": {
```

```

 "ApplicationLatencyP99.9": 60,
 "ApplicationLatencyP90": 60,
 "ApplicationLatencyP99": 60,
 "ApplicationLatencyP95": 60,
 "ApplicationLatencyP85": 60,
 "CPUUser": 60,
 "ApplicationRequests2xx": 60,
 "CPUIdle": 60,
 "ApplicationLatencyP50": 60,
 "ApplicationRequestsTotal": 60,
 "RootFilesystemUtil": 60,
 "LoadAverage1min": 60,
 "CPUirq": 60,
 "CPUNice": 60,
 "CPUiowait": 60,
 "ApplicationLatencyP10": 60,
 "LoadAverage5min": 60,
 "ApplicationRequests5xx": 60,
 "ApplicationLatencyP75": 60,
 "CPUSystem": 60,
 "ApplicationRequests3xx": 60,
 "ApplicationRequests4xx": 60,
 "InstanceHealth": 60,
 "CPUSoftirq": 60
 }
},
"Version": 1
}

```

Na AWS CLI, você passa o documento como um valor da chave `Value` em um argumento de configurações de opção, que é um objeto JSON. Nesse caso, você deve escapar as aspas no documento incorporado.

```

$ aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings '[
 {
 "Namespace": "aws:elasticbeanstalk:healthreporting:system",
 "OptionName": "ConfigDocument",
 "Value": "{\"CloudWatchMetrics\": {\"Environment\":
 {\"ApplicationLatencyP99.9\": 60,\"InstancesSevere\": 60,\"ApplicationLatencyP90\":
 60,\"ApplicationLatencyP99\": 60,\"ApplicationLatencyP95\": 60,\"InstancesUnknown
\": 60,\"ApplicationLatencyP85\": 60,\"InstancesInfo\": 60,\"ApplicationRequests2xx
\": 60,\"InstancesDegraded\": 60,\"InstancesWarning\": 60,\"ApplicationLatencyP50\":

```



```

60,\"ApplicationRequestsTotal\": 60,\"InstancesNoData\": 60,\"InstancesPending
\": 60,\"ApplicationLatencyP10\": 60,\"ApplicationRequests5xx\": 60,
\"ApplicationLatencyP75\": 60,\"InstancesOk\": 60,\"ApplicationRequests3xx\": 60,
\"ApplicationRequests4xx\": 60},\"Instance\": {\"ApplicationLatencyP99.9\": 60,
\"ApplicationLatencyP90\": 60,\"ApplicationLatencyP99\": 60,\"ApplicationLatencyP95\":
60,\"ApplicationLatencyP85\": 60,\"CPUUser\": 60,\"ApplicationRequests2xx\":
60,\"CPUIdle\": 60,\"ApplicationLatencyP50\": 60,\"ApplicationRequestsTotal\":
60,\"RootFilesystemUtil\": 60,\"LoadAverage1min\": 60,\"CPUIrq\": 60,\"CPUNice
\": 60,\"CPUiowait\": 60,\"ApplicationLatencyP10\": 60,\"LoadAverage5min\": 60,
\"ApplicationRequests5xx\": 60,\"ApplicationLatencyP75\": 60,\"CPUSystem\": 60,
\"ApplicationRequests3xx\": 60,\"ApplicationRequests4xx\": 60,\"InstanceHealth\": 60,
\"CPUSoftirq\": 60}},\"Version\": 1}"
}
]'

```

Para um arquivo de configuração `.ebextensions` em YAML, você pode fornecer o documento JSON no estado em que está.

```

option_settings:
 - namespace: aws:elasticbeanstalk:healthreporting:system
 option_name: ConfigDocument
 value: {
"CloudWatchMetrics": {
 "Environment": {
 "ApplicationLatencyP99.9": 60,
 "InstancesSevere": 60,
 "ApplicationLatencyP90": 60,
 "ApplicationLatencyP99": 60,
 "ApplicationLatencyP95": 60,
 "InstancesUnknown": 60,
 "ApplicationLatencyP85": 60,
 "InstancesInfo": 60,
 "ApplicationRequests2xx": 60,
 "InstancesDegraded": 60,
 "InstancesWarning": 60,
 "ApplicationLatencyP50": 60,
 "ApplicationRequestsTotal": 60,
 "InstancesNoData": 60,
 "InstancesPending": 60,
 "ApplicationLatencyP10": 60,
 "ApplicationRequests5xx": 60,
 "ApplicationLatencyP75": 60,
 "InstancesOk": 60,

```

```
 "ApplicationRequests3xx": 60,
 "ApplicationRequests4xx": 60
 },
 "Instance": {
 "ApplicationLatencyP99.9": 60,
 "ApplicationLatencyP90": 60,
 "ApplicationLatencyP99": 60,
 "ApplicationLatencyP95": 60,
 "ApplicationLatencyP85": 60,
 "CPUUser": 60,
 "ApplicationRequests2xx": 60,
 "CPUIdle": 60,
 "ApplicationLatencyP50": 60,
 "ApplicationRequestsTotal": 60,
 "RootFilesystemUtil": 60,
 "LoadAverage1min": 60,
 "CPUIrq": 60,
 "CPUNice": 60,
 "CPUiowait": 60,
 "ApplicationLatencyP10": 60,
 "LoadAverage5min": 60,
 "ApplicationRequests5xx": 60,
 "ApplicationLatencyP75": 60,
 "CPUSystem": 60,
 "ApplicationRequests3xx": 60,
 "ApplicationRequests4xx": 60,
 "InstanceHealth": 60,
 "CPUSoftirq": 60
 }
},
"Version": 1
}
```

## Usar relatórios de integridade aprimorada com a API do Elastic Beanstalk

Como os relatórios de integridade aprimorada do AWS Elastic Beanstalk têm requisitos de pilha de função e solução, você deve atualizar os scripts e o código que usou antes do lançamento deles para poder usá-los. Para manter a compatibilidade com versões anteriores, os relatórios de integridade aprimorada não são habilitados por padrão quando você cria um ambiente usando a API do Elastic Beanstalk.

Configure os relatórios de integridade aprimorada ao definir as opções de função de serviço, perfil da instância e configuração do Amazon CloudWatch no ambiente. Você pode fazer isso de três formas: definindo as opções de configuração na pasta `.ebextensions`, com as configurações salvas, ou configurando-as diretamente no parâmetro `create-environment` da chamada `option-settings`.

Para usar a API, os SDKs ou a interface de linha de comando AWS (CLI) para criar um ambiente que ofereça suporte à integridade aprimorada, você deve:

- Criar uma função de serviço e um perfil da instância com as [permissões](#) apropriadas
- Crie um ambiente com uma nova [versão da plataforma](#)
- Definir as [opções de configuração](#) de tipo de sistema, perfil da instância e função de serviço

Use as seguintes opções de configuração nos namespaces `aws:elasticbeanstalk:healthreporting:system`, `aws:autoscaling:launchconfiguration` e `aws:elasticbeanstalk:environment` para configurar seu ambiente para os relatórios de integridade aprimorada.

## Opções de configuração de integridade aprimorada

### SystemType

Namespace: `aws:elasticbeanstalk:healthreporting:system`

Para habilitar os relatórios de integridade aprimorada, defina como **enhanced**.

### IamInstanceProfile

Namespace: `aws:autoscaling:launchconfiguration`

Defina como o nome de um perfil da instância configurado para uso com o Elastic Beanstalk.

### ServiceRole

Namespace: `aws:elasticbeanstalk:environment`

Defina como o nome de uma função de serviço configurada para uso com o Elastic Beanstalk.

### ConfigDocument (opcional)

Namespace: `aws:elasticbeanstalk:healthreporting:system`

Um documento JSON que define as métricas da instância e do ambiente para publicar no CloudWatch. Por exemplo:

```
{
 "CloudWatchMetrics":
 {
 "Environment":
 {
 "ApplicationLatencyP99.9":60,
 "InstancesSevere":60
 }
 "Instance":
 {
 "ApplicationLatencyP85":60,
 "CPUUser": 60
 }
 }
 "Version":1
}
```

#### Note

Os documentos de configuração podem exigir formatação especial, como aspas de escape, dependendo de como você os disponibiliza no Elastic Beanstalk. Consulte [Fornecer documentos Config de métrica personalizada](#) para ver exemplos.

## Formato de log de integridade aprimorado

AWS Elastic BeanstalkAs plataformas do usam um formato personalizado de log de servidor web para retransmitir informações sobre solicitações HTTP com eficiência ao sistema de relatórios de integridade aprimorada. O sistema analisa os logs, identifica os problemas e define a integridade da instância e do ambiente adequadamente. Se você desabilitar o servidor de proxy da Web no ambiente e atender às solicitações diretamente do contêiner da Web, ainda poderá usar totalmente os relatórios de integridade aprimorada configurando o servidor para enviar logs para o local e no formato usado pelo [agente de integridade do Elastic Beanstalk](#).

**Note**

As informações desta página são relevantes apenas para plataformas Linux. Na plataforma Windows Server, o Elastic Beanstalk recebe informações sobre solicitações HTTP diretamente do servidor Web do IIS. Para obter mais detalhes, consulte [Métricas do servidor web capturadas no IIS no Windows Server](#).

## Configuração do log de servidor web

As plataformas do Elastic Beanstalk são configuradas para gerar dois logs com informações sobre solicitações HTTP. O primeiro está no formato verboso e fornece informações detalhadas sobre a solicitação, incluindo informações sobre o agente do usuário do solicitante e um carimbo de data e hora legível.

```
/var/log/nginx/access.log
```

O exemplo a seguir é de um proxy nginx em execução em um ambiente de servidor da web Ruby, mas o formato é semelhante ao do Apache.

```
172.31.24.3 - - [23/Jul/2015:00:21:20 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
(x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:21 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
(x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:22 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
(x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:22 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
(x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:22 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
(x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
librtmp/2.3" "177.72.242.17"
```

O segundo log está no formato conciso. Ele inclui informações relevantes apenas para os relatórios de integridade aprimorados. Esse log é gerado em uma subpasta chamada `healthd` e muda de hora em hora. Os logs antigos são excluídos imediatamente após o rodízio.

```
/var/log/nginx/healthd/application.log.2015-07-23-00
```

O exemplo a seguir mostra um log no formato legível por máquina.

```
1437609879.311"/"200"0.083"0.083"177.72.242.17
1437609879.874"/"200"0.347"0.347"177.72.242.17
1437609880.006"/bad/path"404"0.001"0.001"177.72.242.17
1437609880.058"/"200"0.530"0.530"177.72.242.17
1437609880.928"/bad/path"404"0.001"0.001"177.72.242.17
```

O formato do log de integridade aprimorado inclui as seguintes informações:

- A hora da solicitação, no horário do Unix.
- O caminho da solicitação
- O código de status HTTP para o resultado
- A hora da solicitação
- A hora de upstream
- O cabeçalho HTTP X-Forwarded-For

Para os proxies nginx, os horários são impressos em segundos de ponto flutuante, com três casas decimais. Para o Apache, são usados microssegundos inteiros.

#### Note

Se você vir um aviso semelhante ao seguinte em um arquivo de log, em que DATE-TIME é uma data e hora, e estiver usando um proxy personalizado, como em um ambiente do Docker de vários contêineres, você deverá usar um .ebextension para configurar o ambiente a fim que de healthd possa ler os arquivos de log:

```
W, [DATE-TIME #1922] WARN -- : log file "/var/log/nginx/healthd/
application.log.DATE-TIME" does not exist
```

É possível começar com .ebextension no [Docker de vários contêineres demonstrativo](#).

```
/etc/nginx/conf.d/webapp_healthd.conf
```

O exemplo a seguir mostra a configuração do log para nginx com o formato de log healthd destacado.

```
upstream my_app {
 server unix:///var/run/puma/my_app.sock;
}

log_format healthd '$msec"$uri"'
 '$status"$request_time"$upstream_response_time"'
 '$http_x_forwarded_for';

server {
 listen 80;
 server_name _ localhost; # need to listen to localhost for worker tier

 if ($time_iso8601 ~ "^(\\d{4})-(\\d{2})-(\\d{2})T(\\d{2})") {
 set $year $1;
 set $month $2;
 set $day $3;
 set $hour $4;
 }

 access_log /var/log/nginx/access.log main;
 access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour healthd;

 location / {
 proxy_pass http://my_app; # match the name of upstream directive which is defined
 proxy_set_header Host $host;
 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 }

 location /assets {
 alias /var/app/current/public/assets;
 gzip_static on;
 gzip on;
 expires max;
 add_header Cache-Control public;
 }

 location /public {
 alias /var/app/current/public;
 gzip_static on;
 }
}
```

```
gzip on;
expires max;
add_header Cache-Control public;
}
}
```

/etc/httpd/conf.d/healthd.conf

O exemplo a seguir mostra a configuração do log para Apache.

```
LogFormat "%{s}t\"%U\"%s\"%D\"%D\"%{X-Forwarded-For}i" healthd
CustomLog "|/usr/sbin/rotatelogs /var/log/httpd/healthd/application.log.%Y-%m-%d-%H
3600" healthd
```

## Gerar logs para a geração de relatórios de integridade aprimorados

Para fornecer logs para o agente de integridade, faça o seguinte:

- Gere os logs no formato correto, como mostrado na seção anterior
- Gere logs em /var/log/nginx/healthd/
- Nomeie os logs usando o seguinte formato: application.log.\$year-\$month-\$day-\$hour
- Faça rodízio dos logs uma vez por hora
- Não trunque os logs

## Notificações e solução de problemas

Esta página lista exemplos de mensagens da causa de problemas comuns e links para mais informações. As mensagens da causa são exibidas na página de [visão geral do ambiente](#) no console do Elastic Beanstalk e são registradas nos [eventos](#) quando problemas de integridade persistem em várias verificações.

## Implantações

O Elastic Beanstalk monitora a consistência do ambiente após as implantações. Se uma implantação contínua falhar, a versão do aplicativo em execução nas instâncias de seu ambiente pode variar. Isso poderá ocorrer se uma implantação for bem-sucedida em um ou mais lotes, mas falhar antes da conclusão de todos os lotes.



Versão incorreta do aplicativo encontrada em 2 de 5 instâncias. Versão esperada "v1" (implantação 1).

Versão incorreta do aplicativo nas instâncias do ambiente. Versão esperada "v1" (implantação 1).

A versão esperada do aplicativo não está em execução em algumas ou todas as instâncias de um ambiente.

Versão incorreta do aplicativo "v2" (implantação 2). Versão esperada "v1" (implantação 1).

O aplicativo implantado em uma instância não tem a versão esperada. Se uma implantação falhar, a versão esperada será redefinida para a versão da implantação bem-sucedida mais recente. No exemplo acima, a primeira implantação (versão "v1") foi bem-sucedida, mas a segunda falhou (versão "v2"). Todas as instâncias que executam "v2" não são consideradas íntegras.

Para resolver esse problema, inicie outra implantação. Você pode [reimplantar uma versão anterior](#) que você sabe que funciona, ou configurar seu ambiente para [ignorar verificações de integridade](#) durante a implantação e reimplantar a nova versão para forçar a conclusão da implantação.

Você também pode identificar e encerrar as instâncias que estão executando a versão incorreta do aplicativo. O Elastic Beanstalk iniciará instâncias com a versão correta para substituir as instâncias encerradas. Use o [comando de integridade da EB CLI](#) para identificar as instâncias que estão executando a versão incorreta do aplicativo.

## Servidor de aplicativos

15% das solicitações estão com erro de HTTP 4xx

20% das solicitações para o ELB estão com erro de HTTP 4xx.

Uma alta porcentagem de solicitações HTTP para uma instância ou um ambiente está falhando com erros 4xx.

Um código de status da série 400 indica que o usuário fez uma solicitação incorreta, como solicitar uma página que não existe (404 Arquivo não encontrado) ou que o usuário não tem acesso (403 Proibido). Um pequeno número de 404s não é incomum, mas um grande número pode significar que existem links internos ou externos para páginas indisponíveis. Esses problemas podem ser resolvidos ao corrigir os links internos inválidos e ao adicionar redirecionamentos de links externos inválidos.

5% das solicitações estão falhando com HTTP 5xx

3% das solicitações para o ELB estão falhando com HTTP 5xx.

Uma alta porcentagem de solicitações HTTP para uma instância ou um ambiente está falhando com códigos de status da série 500.

Um código de status da série 500 indica que o servidor de aplicativos detectou um erro interno. Esses problemas indicam que há um erro no código do aplicativo que deve ser identificado e corrigido rapidamente.

95% da CPU está em uso

Em uma instância, o agente de integridade está relatando uma porcentagem extremamente alta de uso da CPU e define a integridade da instância como Aviso ou Reduzido.

Escale o ambiente para retirar carga das instâncias.

## Instância do operador

20 mensagens em espera na fila (25 segundos atrás)

As solicitações estão sendo adicionadas à fila do ambiente do operador com mais rapidez do que podem ser processadas. Escale o ambiente para aumentar a capacidade.

5 mensagens em Dead Letter Queue (15 segundos atrás)

As solicitações de operador estão falhando repetidamente e sendo adicionadas à [the section called “Filas de mensagens mortas”](#). Verifique o motivo da falha das solicitações na fila de mensagens mortas.

## Outros recursos

4 instâncias ativas estão abaixo do tamanho mínimo 5 do grupo de Auto Scaling 5

O número de instâncias em execução no ambiente é menor do que o mínimo configurado para o grupo de Auto Scaling.

As notificações do grupo de Auto Scaling (groupname) foram excluídas ou modificadas

As notificações configuradas para o seu grupo de Auto Scaling foram modificadas fora do Elastic Beanstalk.

## Gerenciar alarmes

É possível criar alarmes para as métricas que você está monitorando usando o console do Elastic Beanstalk. Os alarmes ajudam a monitorar alterações no ambiente do AWS Elastic Beanstalk, para que seja possível identificar facilmente e mitigar os problemas antes que eles ocorram. Por exemplo, defina um alarme que notifique quando o uso da CPU no ambiente exceder um determinado limite, garantindo que você seja notificado antes que ocorra um problema em potencial. Para obter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon CloudWatch](#).

### Note

O Elastic Beanstalk usa o CloudWatch para monitoramento e alarmes, o que significa que os custos do CloudWatch são aplicados à sua conta da AWS para quaisquer alarmes utilizados.

Para obter mais informações sobre monitoramento de métricas específicas, consulte [Relatórios de integridade básica](#).

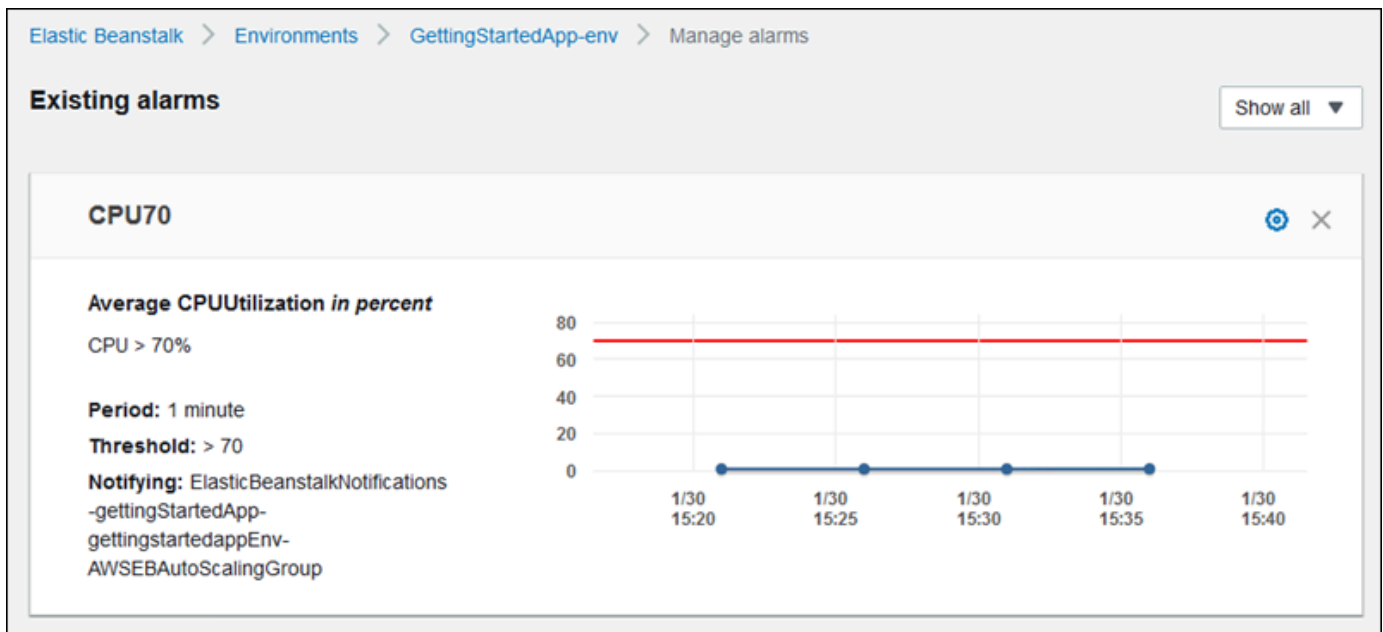
Para verificar o estado dos alarmes

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, selecione Alarmes.



A página exibe uma lista de alarmes existentes. Se algum alarme estiver no estado de alarme, será marcado com



(aviso).

4. Para filtrar alarmes, escolha o menu suspenso e selecione um filtro.
5. Para editar ou apagar um alarme, escolha



(editar) ou



(excluir), respectivamente.

Para criar um alarme

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

#### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Monitoring (Monitoramento).
4. Localize a métrica para a qual você deseja criar um alarme e escolha



(alarme). A página Add alarm (Adicionar alarme) é exibida.

The screenshot shows the 'Add Alarm' interface in the AWS console. The breadcrumb trail is 'Elastic Beanstalk > Environments > GettingStartedApp-env > Add alarm'. The main heading is 'Add Alarm'. The metric selected is 'Average CPUUtilization in percent'. The form includes fields for Name, Description (optional), Period (set to 1 minute), Threshold (Average CPUUtilization), Change state after, Notify (set to 'A new SNS topic...'), Topic name (ElasticBeanstalkNotifications-gettingStartedApp-gettingsta), and E-mail address. There are checkboxes for 'Notify when state changes to' with options 'OK', 'Alarm', and 'Insufficient data'. At the bottom, there are 'Cancel' and 'Add' buttons. To the right, there is a line graph showing the metric's value over time (1/30 15:35 to 15:55) and a section titled 'Other Alarms For This Metric' with a link for 'CPU70'.

Alarm Name
CPU70

5. Insira detalhes sobre o alarme:
  - Name: o nome desse alarme.
  - Description (opcional): uma breve descrição do alarme.
  - Period: o intervalo de tempo entre as leituras.

- **Threshold:** descreve o comportamento e o valor que a métrica deve exceder para acionar um alarme.
  - **Change state after:** o tempo após o limite ter sido excedido para acionar uma alteração no estado do alarme.
  - **Notify (Notificar):** o tópico do Amazon SNS que é notificado quando um alarme muda de estado.
  - **Notify when state changes to:**
    - **OK:** a métrica está dentro do limite definido.
    - **Alarm:** a métrica excedeu o limite definido.
    - **Insufficient data:** o alarme acabou de ser acionado, a métrica não está disponível ou não há dados suficientes para a métrica determinar o estado do alarme.
6. Escolha **Add (Adicionar)**. O status do ambiente muda para cinza enquanto o ambiente é atualizado. É possível visualizar o alarme criado escolhendo **Alarms (Alarmes)** no painel de navegação.

## Visualizar o histórico de alterações de um ambiente do Elastic Beanstalk

Você pode usar o Console de gerenciamento da AWS para visualizar um histórico de alterações de configuração feitas nos seus ambientes do Elastic Beanstalk. O Elastic Beanstalk busca seu histórico de alterações em eventos registrados no [AWS CloudTrail](#) e os exibe em uma lista em que você pode navegar e filtrar facilmente.

O painel **Change History (Histórico de alterações)** exibe as seguintes informações sobre alterações feitas nos seus ambientes:

- A data e a hora em que uma alteração foi feita
- O usuário do IAM que foi responsável por uma alteração feita
- A ferramenta de origem (interface de linhas de comando do Elastic Beanstalk (CLI do EB) ou console) que foi usada para fazer a alteração
- O parâmetro de configuração e os novos valores que foram definidos

Quaisquer dados confidenciais que fazem parte da alteração, como os nomes dos usuários do banco de dados afetados pela alteração, não são exibidos no painel.

## Para ver o histórico de alterações

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Change history (Histórico de alterações).

The screenshot shows the AWS Elastic Beanstalk console interface. On the left, there is a navigation sidebar with options like Environments, Applications, and Change history. The main area displays the 'Change history' page, which includes a search bar and a table of configuration changes. The table has the following columns: Date, IAM user, Event source, Environment, and Configuration changes. The Configuration changes column is expanded to show details for several events, including aws:autoscaling:launchconfiguration, aws:elasticbeanstalk:healthreporting:system, and aws:elasticbeanstalk:environment:proxy:staticfiles.

Date	IAM user	Event source	Environment	Configuration changes
2020-10-16T02:14:15Z	AIDACKCEVSQ6C2EXAMPLE	eb-cli/3.19.0 Python/3.8.5 Windows/10	GettingStartedApp-env	<ul style="list-style-type: none"> <li>Changes made</li> <li>aws:autoscaling:launchconfiguration <ul style="list-style-type: none"> <li>EC2KeyName : example-keyname</li> </ul> </li> </ul>
2020-10-16T02:10:59Z	AIDACKCEVSQ6C2EXAMPLE2	console.amazonaws.com	GettingStartedApp-env	<ul style="list-style-type: none"> <li>Changes made</li> <li>aws:elasticbeanstalk:healthreporting:system <ul style="list-style-type: none"> <li>ConfigDocument : -</li> </ul> </li> <li>aws:elasticbeanstalk:sns:topics <ul style="list-style-type: none"> <li>Notification Endpoint : jane@example.com</li> </ul> </li> </ul>
2020-10-16T02:02:50Z	AIDACKCEVSQ6C2EXAMPLE	eb-cli/3.19.0 Python/3.8.5 Windows/10	GettingStartedApp-env	-
2020-10-09T21:20:07Z	AIDACKCEVSQ6C2EXAMPLE2	console.amazonaws.com	Ruby-example-dev	<ul style="list-style-type: none"> <li>Changes made</li> <li>aws:elasticbeanstalk:environment:proxy:staticfiles <ul style="list-style-type: none"> <li>/public : Sensitive data removed</li> </ul> </li> </ul>
2020-10-09T21:17:01Z	AIDACKCEVSQ6C2EXAMPLE3	console.amazonaws.com	Ruby-example-dev	Changes made
2020-10-09T19:05:21Z	AIDACKCEVSQ6C2EXAMPLE3	console.amazonaws.com	Ruby-example-dev	Changes made
2020-10-09T19:03:04Z	AIDACKCEVSQ6C2EXAMPLE3	console.amazonaws.com	Ruby-example-dev	Changes made
2020-10-09T19:00:04Z	AIDACKCEVSQ6C2EXAMPLE3	eb-cli/3.19.0 Python/3.8.5 Windows/10	Ruby-example-dev	-
2020-10-09T18:55:42Z	AIDACKCEVSQ6C2EXAMPLE3	eb-cli/3.19.0 Python/3.8.5 Windows/10	Ruby-example-dev	-

A página Change history (Histórico de alterações) mostra uma lista das alterações de configuração que foram feitas nos seus ambientes do Elastic Beanstalk. Você pode pesquisar a lista escolhendo <(previous) (anterior) ou, > (next) (próximo), ao selecionar um número de página específico. Na coluna Configuration changes (Alterações de configuração), selecione o ícone de seta para alternar entre expandir e recolher a lista de alterações no cabeçalho Changes made (Alterações feitas). Use a barra de pesquisa para filtrar os resultados da lista de histórico de alterações. Você pode inserir qualquer string para refinar a lista de alterações que são exibidas.

Observe o seguinte sobre como filtrar os resultados exibidos:

- O filtro de pesquisa não diferencia maiúsculas de minúsculas.
- Você pode filtrar as alterações exibidas com base nas informações da coluna Configuration changes (Alterações de configuração), mesmo quando ela não estiver visível por estar recolhida dentro das alterações feitas.

- Você só pode filtrar os resultados exibidos. No entanto, o filtro permanece no lugar mesmo se você optar por ir para outra página para exibir mais resultados. Seus resultados filtrados também são anexados ao conjunto de resultados da página seguinte.

Os exemplos a seguir demonstram como os dados mostrados na tela anterior podem ser filtrados:

- Digite **GettingStartedApp-env** na caixa de pesquisa para refinar os resultados para incluir apenas as alterações feitas no ambiente denominado GettingStartedApp-env.
- Digite **example3** na caixa de pesquisa para refinar os resultados para incluir apenas as alterações feitas por usuários do IAM cujo nome de usuário contém a string example3.
- Digite **2020-10** na caixa de pesquisa para refinar os resultados para incluir apenas as alterações feitas durante o mês de outubro de 2020. Altere o valor da pesquisa **2020-10-16** para filtrar ainda mais os resultados exibidos para incluir apenas as alterações feitas no dia 16 de outubro de 2020.
- Digite **proxy:staticfiles** na caixa de pesquisa para refinar os resultados para incluir apenas as alterações feitas no namespace denominado aws:elasticbeanstalk:environment:proxy:staticfiles. As linhas exibidas são o resultado do filtro. Isso é verdade mesmo para resultados que são recolhidos em Changes made (Alterações feitas).

## Visualizar o stream de eventos de um ambiente do Elastic Beanstalk

Você pode usar o Console de Gerenciamento da AWS para acessar eventos e notificações associados a sua aplicação.

Para visualizar eventos

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Events.



Elastic Beanstalk > Environments > GettingStartedApp-env > Events

Click the link to be routed to the previous Beanstalk Console  
Switch to the previous console

### Events

Severity  < 1 2 3 4 5 6 7 ... 18 >

Time	Type	Details
2020-03-09 17:14:06 UTC-0700	INFO	createConfigurationTemplate completed successfully.
2020-03-09 17:14:06 UTC-0700	INFO	createConfigurationTemplate is starting.
2020-03-03 04:16:55 UTC-0800	INFO	Environment health has transitioned from Info to Ok. Configuration update completed 85 seconds ago and took 15 minutes.
2020-03-03 04:16:07 UTC-0800	INFO	Environment update completed successfully.
2020-03-03 04:16:07 UTC-0800	INFO	Successfully deployed new configuration to environment.

A página de eventos exibe uma lista de todos os eventos que foram registrados para o ambiente. É possível percorrer a lista escolhendo < (anterior), > (próximo) ou números de página. É possível filtrar por tipo de evento usando a lista suspensa Severity (Gravidade).

A [EB CLI](#) e a [AWS CLI](#) fornecem comandos para recuperar eventos. Se você estiver gerenciando seu ambiente usando a EB CLI, use [eb events](#) para imprimir uma lista de eventos. Este comando também tem uma opção `--follow` que continua a mostrar novos eventos até que você pressione Ctrl+C para interromper a saída.

Para extrair eventos usando a AWS CLI, use o comando `describe-events` e especifique o ambiente por nome ou ID:

```
$ aws elasticbeanstalk describe-events --environment-id e-gbjzqcra3
{
 "Events": [
 {
 "ApplicationName": "elastic-beanstalk-example",
```

```
 "EnvironmentName": "elasticBeanstalkExa-env",
 "Severity": "INFO",
 "RequestId": "a4c7bfd6-2043-11e5-91e2-9114455c358a",
 "Message": "Environment update completed successfully.",
 "EventDate": "2015-07-01T22:52:12.639Z"
 },
 ...
```

Para obter mais informações sobre as ferramentas da linha de comando, consulte [Ferramentas](#).

## Listar e conectar com instâncias de servidor

É possível visualizar uma lista de instâncias do Amazon EC2 executando o ambiente de aplicações do AWS Elastic Beanstalk por meio do console do Elastic Beanstalk. Você pode se conectar a instâncias usando qualquer cliente SSH. Você pode se conectar a instâncias que executam o Windows usando a Área de Trabalho Remota.

Algumas notas sobre ambientes específicos de desenvolvimento:

- Para obter mais informações sobre como listar e conectar-se a instâncias de servidor usando o AWS Toolkit for Eclipse, consulte [Listar e conectar com instâncias de servidor](#).
- Para obter mais informações sobre como listar e conectar-se a instâncias de servidor usando o AWS Toolkit for Visual Studio, consulte [Listar e conectar com instâncias de servidor](#).

### Important

Para poder acessar as instâncias do Amazon EC2 provisionadas pelo Elastic Beanstalk, é necessário criar um par de chaves do Amazon EC2 e configurar as instâncias do Amazon EC2 provisionadas pelo Elastic Beanstalk para usar o par de chaves do Amazon EC2. Você pode configurar os pares de chaves do Amazon EC2 usando o [Console de Gerenciamento da AWS](#). Para obter instruções sobre como criar um par de chaves do Amazon EC2, consulte o Guia de conceitos básicos do Amazon EC2. Para obter mais informações sobre como configurar suas instâncias do Amazon EC2 para usar um par de chaves do Amazon EC2, consulte [EC2 key pair](#).

Por padrão, o Elastic Beanstalk não habilita conexões remotas com instâncias do EC2 em um contêiner do Windows, exceto os contêineres legados do Windows. (O Elastic Beanstalk configura as instâncias do EC2 nos contêineres legados do Windows para usar a porta 3389 para conexões RDP.) Você pode habilitar conexões remotas com suas instâncias do EC2

que executam o Windows adicionando uma regra a um security group que autoriza o tráfego de entrada para as instâncias. Recomendamos enfaticamente que você remova a regra quando encerrar sua conexão remota. Você pode adicioná-la novamente na próxima vez que precisar fazer login remotamente. Para obter mais informações, consulte [Adicionar uma regra para tráfego de entrada RDP para uma instância do Windows](#) e [Conectar-se à instância do Windows](#) no Guia do usuário do Amazon Elastic Compute Cloud para Microsoft Windows.

Para visualizar e conectar-se a instâncias do Amazon EC2 em um ambiente

1. Abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
2. No painel de navegação do console, escolha Load Balancers.

EC2 Dashboard

- Events
- Tags
- Reports
- Limits
- INSTANCES
- IMAGES
- ELASTIC BLOCK STORE
- NETWORK & SECURITY
- LOAD BALANCING
  - Load Balancers**

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

2 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
2 Volumes	2 Load Balancers
0 Key Pairs	6 Security Groups
0 Placement Groups	

Just need a simple virtual private server? Get everything you need to jumpstart your project - compute, storage, and networking – for a low, predictable price. [Try Amazon Lightsail for free.](#)

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

3. Os balanceadores de carga criados pelo Elastic Beanstalk têm awseb no nome. Encontre e clique no load balancer do seu ambiente.

EC2 Dashboard

- Events
- Tags
- Reports
- Limits
- INSTANCES
- IMAGES

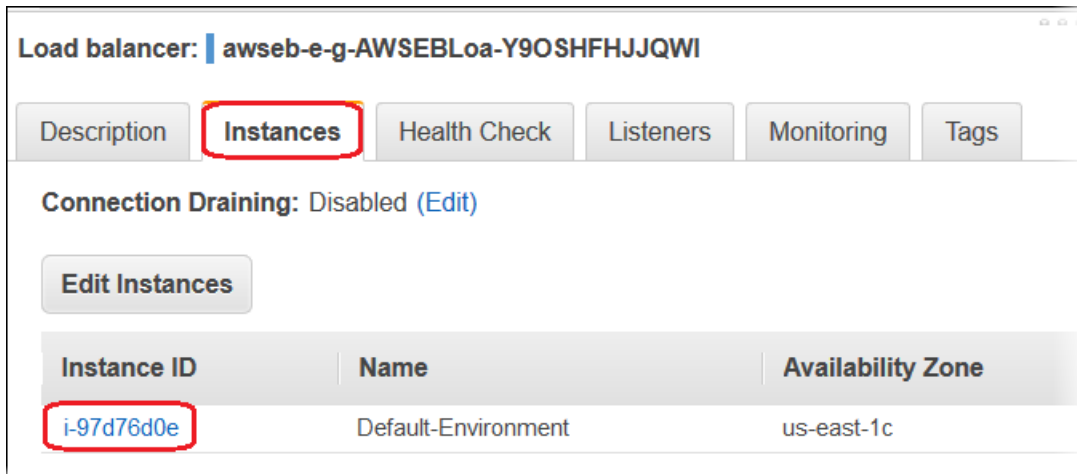
Create Load Balancer

Actions

Filter: Search

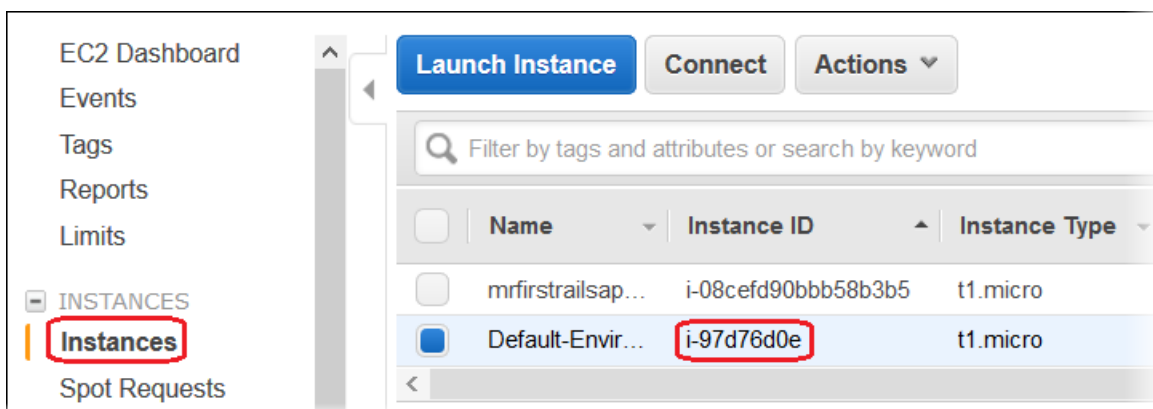
Name	DNS name	State
<b>awseb-e-g-AWSEBLoa-Y9O...</b>	awseb-e-g-AWSEBLoa-Y9O...	
awseb-e-w-AWSEBLoa-2XLI...	awseb-e-w-AWSEBLoa-2XLI...	

4. Escolha a guia Instances (Instâncias) no painel inferior do console.



Uma lista de instâncias que o balanceador de carga do seu ambiente do Elastic Beanstalk usa é exibida. Anote o ID da instância à qual você deseja se conectar.

- No painel de navegação do console do Amazon EC2, escolha Instances (Instâncias) e encontre o ID de sua instância na lista.



- Clique com o botão direito do mouse no ID da instância do Amazon EC2 que está em execução no balanceador de carga do seu ambiente e, depois, selecione Connect (Conectar) no menu de contexto.
- Anote o endereço DNS público da instância na guia Description.
- Conecte-se a uma instância que executa o Linux usando o cliente SSH de sua escolha e, em seguida, digite `ssh -i .ec2/mykeypair.pem ec2-user@<public-DNS-of-the-instance>`.

Para obter mais informações sobre como se conectar a uma instância Linux do Amazon EC2, consulte [Conceitos básicos de instâncias Linux do Amazon EC2](#) no Guia do usuário do Amazon EC2 para instâncias Linux.

Se o ambiente do Elastic Beanstalk usa a [plataforma .NET no Windows Server](#), consulte [Conceitos básicos com instâncias do Windows do Amazon EC2](#) no Guia do usuário do Amazon EC2 para instâncias do Windows.

## Visualizar logs de instâncias do Amazon EC2 no ambiente do Elastic Beanstalk

As instâncias do Amazon EC2 no ambiente do Elastic Beanstalk geram logs que podem ser visualizados para solucionar problemas com sua aplicação ou arquivos de configuração. Os logs criados pelo servidor Web, pelo servidor de aplicações, pelos scripts da plataforma do Elastic Beanstalk e pelo AWS CloudFormation são armazenados localmente em instâncias individuais. Você pode recuperá-los facilmente usando o [console de gerenciamento do ambiente](#) ou a CLI do EB. Também é possível configurar seu ambiente para fazer streaming de logs para o Amazon CloudWatch Logs em tempo real.

Logs finais são as últimas 100 linhas dos arquivos de log usados com mais frequência, ou seja, logs operacionais do Elastic Beanstalk e logs do servidor Web ou do servidor de aplicações. Quando logs finais são solicitados no console de gerenciamento do ambiente ou com `eb logs`, uma instância no ambiente concatena as entradas de log mais recentes em um único arquivo de texto e o carrega no Amazon S3.

Logs em pacote são logs completos para uma maior variedade de arquivos de log, incluindo logs do yum e do cron e vários logs do AWS CloudFormation. Quando logs em pacote são solicitados, uma instância no ambiente empacota todos os arquivos de log em um arquivo ZIP e o carrega no Amazon S3.

### Note

As plataformas do Elastic Beanstalk Windows Server não são compatíveis com logs em pacote.

Para fazer upload de logs alternados para o Amazon S3, as instâncias no ambiente precisam ter um [perfil da instância](#) com permissão para gravar no bucket do Amazon S3 do Elastic Beanstalk. Essas permissões são incluídas no perfil de instância padrão cuja criação o Elastic Beanstalk solicita ao iniciar um ambiente no console do Elastic Beanstalk pela primeira vez.

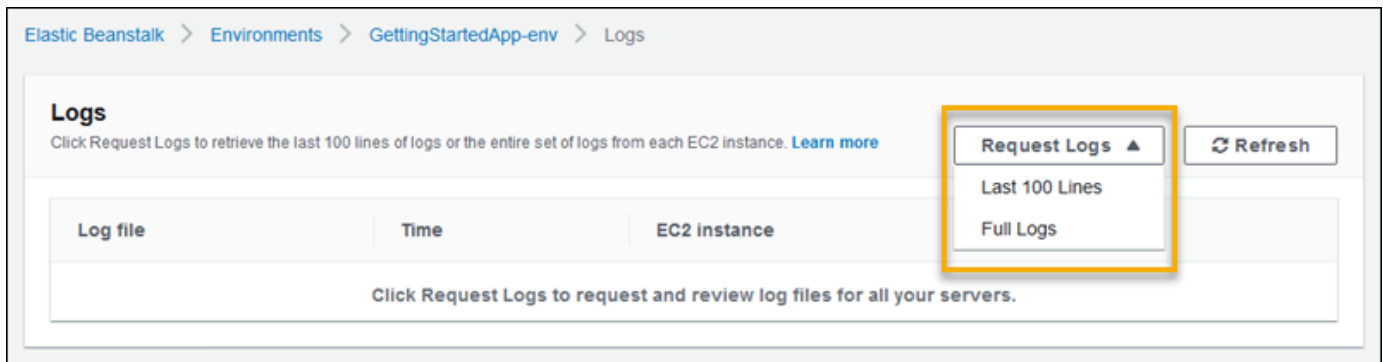
## Como recuperar logs de instâncias

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, selecione Logs.
4. Escolha Request Logs (Logs de solicitação) e escolha o tipo de logs a serem recuperados. Para obter logs finais, escolha Últimas 100 linhas. Para obter logs em pacote, selecione Logs completos.



5. Quando o Elastic Beanstalk terminar de recuperar seus logs, escolha Download.

O Elastic Beanstalk armazena logs finais e em pacote em um bucket do Amazon S3 e gera um URL pré-assinado do Amazon S3 que pode ser usado para acessar os logs. O Elastic Beanstalk exclui os arquivos do Amazon S3 após 15 minutos.

### Warning

Qualquer pessoa que tenha o URL do Amazon S3 pré-assinado poderá acessar os arquivos antes que eles sejam excluídos. Disponibilize o URL somente para partes confiáveis.

**Note**

Sua política de usuário deve ter a permissão `s3:DeleteObject`. O Elastic Beanstalk usa suas permissões de usuário para excluir os logs do Amazon S3.

Para mantê-los, configure o ambiente para publicá-los no Amazon S3 automaticamente depois que forem alternados. Para habilitar a alternância de logs para o Amazon S3, siga o procedimento em [Configurar visualização de logs de instância](#). As instâncias em seu ambiente tentarão fazer upload dos logs que foram girados uma vez por hora.

Se o aplicativo gera logs em um local que não faz parte da configuração padrão da plataforma do ambiente, você pode estender a configuração padrão usando arquivos de configuração ([.ebextensions](#)). É possível adicionar arquivos de log de seu aplicativo aos logs finais, aos logs em pacote e à rotação de logs.

Para streaming de logs em tempo real e armazenamento no longo prazo, configure o ambiente para [fazer streaming de logs para o Amazon CloudWatch Logs](#).

## Seções

- [Local de log em instâncias do Amazon EC2](#)
- [Local de log no Amazon S3](#)
- [Configurações de rotação de log no Linux](#)
- [Extensão da configuração padrão da tarefa de log](#)
- [Fazer streaming de arquivos de log para o Amazon CloudWatch Logs](#)

## Local de log em instâncias do Amazon EC2

Os logs são armazenados em locais padrão nas instâncias do Amazon EC2 no ambiente. O Elastic Beanstalk gera os logs a seguir.

### Amazon Linux 2

- `/var/log/eb-engine.log`

### Amazon Linux AMI (AL1)

**Note**

Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

- `/var/log/eb-activity.log`
- `/var/log/eb-commandprocessor.log`

### Windows Server

- `C:\Program Files\Amazon\ElasticBeanstalk\logs\`
- `C:\cfn\log\cfn-init.log`

Esses logs contêm mensagens sobre atividades de implantação, incluindo mensagens relacionadas a arquivos de configuração ([.ebextensions](#)).

Cada aplicativo e servidor web armazena logs em uma pasta própria:

- Apache: `/var/log/httpd/`
- IIS: `C:\inetpub\wwwroot\`
- Node.js: `/var/log/nodejs/`
- nginx: `/var/log/nginx/`
- Passenger: `/var/app/support/logs/`
- Puma: `/var/log/puma/`
- Python: `/opt/python/log/`
- Tomcat: `/var/log/tomcat/`

## Local de log no Amazon S3

Ao solicitar logs finais ou em pacote de seu ambiente, ou quando as instâncias fazem upload de logs alternados, eles são armazenados em seu bucket do Elastic Bucket no Amazon S3. O Elastic



Beanstalk cria um bucket denominado `elasticbeanstalk-region-account-id` para cada região da AWS em que ambientes são criados. Dentro desse bucket, os logs são armazenados no caminho `resources/environments/logs/logtype/environment-id/instance-id`.

Por exemplo, os logs da instância `i-0a1fd158`, no ambiente do Elastic Beanstalk `-mpcwnwheky` na região da AWS `us-west-2` na conta `123456789012` são armazenados nos seguintes locais:

- Logs finais:

```
s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/tail/e-mpcwnwheky/i-0a1fd158
```

- Logs em pacote:

```
s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/bundle/e-mpcwnwheky/i-0a1fd158
```

- Logs alternados:

```
s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/publish/e-mpcwnwheky/i-0a1fd158
```

#### Note

Você pode localizar o ID do ambiente no console de gerenciamento do ambiente.

O Elastic Beanstalk exclui os logs finais e em pacote do Amazon S3 automaticamente 15 minutos após eles serem criados. Os logs alternados são mantidos até você excluí-los ou movê-los para o S3 Glacier.

## Configurações de rotação de log no Linux

Em plataformas Linux, o Elastic Beanstalk usa `logrotate` para alternar logs periodicamente. Se configurado, depois que um log é alternado localmente, a tarefa de alternância de logs o seleciona e faz upload dele no Amazon S3. Por padrão, os logs que são alternados localmente não são exibidos em logs finais ou em pacote.

É possível encontrar arquivos de configuração do Elastic Beanstalk para `logrotate` em `/etc/logrotate.elasticbeanstalk.hourly/`. Essas configurações de rotação são específicas

da plataforma e podem mudar em futuras versões dela. Para obter mais informações sobre as definições disponíveis e configurações de exemplo, execute `man logrotate`.

Os arquivos de configuração são chamados de trabalhos cron em `/etc/cron.hourly/`. Para obter mais informações sobre cron, execute `man cron`.

## Extensão da configuração padrão da tarefa de log

O Elastic Beanstalk usa arquivos em subpastas de `/opt/elasticbeanstalk/tasks` (Linux) ou `C:\Program Files\Amazon\ElasticBeanstalk\config` (Windows Server) na instância do Amazon EC2 para configurar tarefas para logs finais, logs em pacote e alternância de logs.

No Amazon Linux:

- Logs finais:

```
/opt/elasticbeanstalk/tasks/taillogs.d/
```

- Logs em pacote:

```
/opt/elasticbeanstalk/tasks/bundlelogs.d/
```

- Logs alternados:

```
/opt/elasticbeanstalk/tasks/publishlogs.d/
```

No Windows Server:

- Logs finais:

```
c:\Program Files\Amazon\ElasticBeanstalk\config\taillogs.d\
```

- Logs alternados:

```
c:\Program Files\Amazon\ElasticBeanstalk\config\publogs.d\
```

Por exemplo, o arquivo `eb-activity.conf` no Linux adiciona dois arquivos de log à tarefa de logs finais.

### **`/opt/elasticbeanstalk/tasks/taillogs.d/eb-activity.conf`**

```
/var/log/eb-commandprocessor.log
```

```
/var/log/eb-activity.log
```

Você pode usar arquivos de configuração do ambiente ([.ebextensions](#)) para adicionar seus próprios arquivos `.conf` a essas pastas. Um arquivo `.conf` lista os arquivos de log específicos para a aplicação, que o Elastic Beanstalk adiciona às tarefas de arquivo de log.

Use a seção [files](#) para adicionar os arquivos de configuração às tarefas que você deseja modificar. Por exemplo, o texto de configuração a seguir adiciona um arquivo de configuração de log a cada instância em seu ambiente: Este arquivo de configuração de log, `cloud-init.conf`, adiciona `/var/log/cloud-init.log` a logs finais.

```
files:
 "/opt/elasticbeanstalk/tasks/taillogs.d/cloud-init.conf" :
 mode: "000755"
 owner: root
 group: root
 content: |
 /var/log/cloud-init.log
```

Adicione esse texto a um arquivo com a extensão de nome de arquivo `.config` em seu pacote de origem em uma pasta chamada `.ebextensions`.

```
~/workspace/my-app
|-- .ebextensions
| |-- tail-logs.config
|-- index.php
`-- styles.css
```

Em plataformas Linux, você também pode usar caracteres curinga em configurações de tarefas de log. Este arquivo de configuração adiciona todos os arquivos com a extensão de nome de arquivo `.log` da pasta `log` na raiz do aplicativo aos logs em pacote.

```
files:
 "/opt/elasticbeanstalk/tasks/bundlelogs.d/applogs.conf" :
 mode: "000755"
 owner: root
 group: root
 content: |
 /var/app/current/log/*.log
```

As configurações de tarefas de log não oferecem suporte a caracteres curinga em plataformas Windows.

### Note

Para ajudar a se familiarizar com os procedimentos de personalização de log, é possível implantar uma aplicação de exemplo usando a [CLI do EB](#). Para isso, a CLI do EB cria um diretório de aplicação local que contém um subdiretório `.ebextensions` com uma configuração de exemplo. Também é possível usar os arquivos de log da aplicação de exemplo para explorar o recurso de recuperação de log descrito neste tópico. Para obter mais informações sobre como criar uma aplicação de exemplo com a CLI do EB, consulte as [Noções básicas da CLI do EB](#).

Para obter mais informações sobre o uso de arquivos de configuração, consulte [Personalização avançada de ambiente com arquivos de configuração \(.ebextensions\)](#).

Assim como é possível estender logs finais e logs de pacote, você pode estender a rotação de log usando um arquivo de configuração. Sempre que o Elastic Beanstalk alterna seus próprios logs e os envia para o Amazon S3, ele também alterna e faz upload de seus logs adicionais. A extensão de rotação de log se comporta de forma diferente dependendo da plataforma do sistema operacional. As seções a seguir descrevem os dois casos.

## Extensão de rotação de log no Linux

Conforme explicado em [Configurações de rotação de log no Linux](#), o Elastic Beanstalk usa `logrotate` para alternar logs em plataformas Linux. Quando você configura os arquivos de log do seu aplicativo para a rotação de log, o aplicativo não precisa criar cópias de arquivos de log. O Elastic Beanstalk configura `logrotate` para criar uma cópia dos arquivos de log da aplicação para cada alternância. Portanto, o aplicativo deve manter os arquivos de log desbloqueados quando não estiver gravando nesses arquivos.

## Extensão da rotação de log no Windows Server

No Windows Server, quando você configura os arquivos de log do seu aplicativo para a rotação de log, o aplicativo deve girar os arquivos de log periodicamente. O Elastic Beanstalk procura arquivos com nomes iniciados pelo padrão que você configurou e seleciona-os para fazer upload para o Amazon S3. Além disso, pontos no nome de arquivo são ignorados e o Elastic Beanstalk considera o nome até o ponto como a base do nome do arquivo de log.

O Elastic Beanstalk faz upload de todas as versões de um arquivo de log base, exceto da mais recente, porque considera essa a versão do arquivo de log da aplicação ativa, que pode ser bloqueada. O aplicativo pode, portanto, manter o arquivo de log ativo bloqueado entre alternâncias.

Por exemplo: o aplicativo grava em um arquivo de log chamado `my_log.log` e você especifica esse nome no arquivo `.conf`. O aplicativo periodicamente gira o arquivo. Durante o ciclo de alternância do Elastic Beanstalk, ele localiza os seguintes arquivos na pasta do arquivo de log: `my_log.log`, `my_log.0800.log`, `my_log.0830.log`. O Elastic Beanstalk considera todos eles como versões do nome base `my_log`. O arquivo `my_log.log` tem a hora de modificação mais recente, então o Elastic Beanstalk faz upload apenas dos outros dois arquivos, `my_log.0800.log` e `my_log.0830.log`.

## Fazer streaming de arquivos de log para o Amazon CloudWatch Logs

É possível configurar seu ambiente para fazer streaming de logs para o Amazon CloudWatch Logs no console do Elastic Beanstalk ou usando [opções de configuração](#). Com o CloudWatch Logs, cada instância no ambiente faz o streaming de logs para grupos de logs que você pode configurar para serem retidos por semanas ou anos, mesmo depois que seu ambiente for encerrado.

O conjunto de logs transmitido varia por ambiente, mas sempre inclui o `eb-engine.log` e os logs de acesso do servidor de proxy nginx ou Apache executado à frente do aplicativo.

É possível configurar o streaming de logs no console do Elastic Beanstalk [durante a criação do ambiente](#) ou [em um ambiente existente](#). No exemplo a seguir, os logs são salvos por até sete dias, mesmo quando o ambiente é encerrado.

### Instance log streaming to CloudWatch Logs

Configure the instances in your environment to stream logs to CloudWatch Logs. You can set the retention to up to ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

**Log groups**  
[/aws/elasticbeanstalk/GettingStartedApp-env](#)

**Log streaming**  
(Standard CloudWatch charges apply.)  
 Enabled

**Retention**  
7 days

**Lifecycle**  
Keep logs after terminating environment

O [arquivo de configuração](#) a seguir permite o streaming de logs com retenção de 180 dias, mesmo se o ambiente estiver encerrado.

Example `.ebextensions/log-streaming.config`

```
option_settings:
 aws:elasticbeanstalk:cloudwatch:logs:
 StreamLogs: true
 DeleteOnTerminate: false
 RetentionInDays: 180
```

# Usar o Elastic Beanstalk com outros serviços da AWS

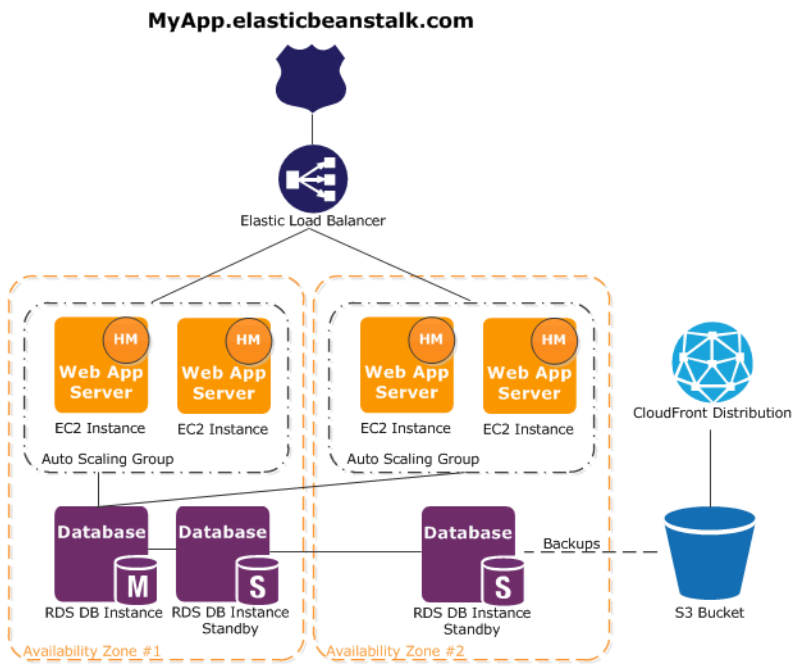
Para implementar os ambientes da sua aplicação, o Elastic Beanstalk gerencia recursos de outros serviços da AWS ou usa sua funcionalidade. Além disso, o Elastic Beanstalk integra-se aos serviços da AWS que ela não use diretamente como parte de seus ambientes. Os tópicos nesta seção descrevem muitas maneiras de usar esses serviços adicionais com sua aplicação do Elastic Beanstalk.

## Tópicos

- [Visão geral da arquitetura](#)
- [Usar o Elastic Beanstalk com o Amazon CloudFront](#)
- [Registro em log de chamadas da API do Elastic Beanstalk com AWS CloudTrail](#)
- [Usar o Elastic Beanstalk com o Amazon CloudWatch](#)
- [Usar o Elastic Beanstalk com o Amazon CloudWatch Logs](#)
- [Usar o Elastic Beanstalk com o Amazon EventBridge](#)
- [Localizar e acompanhar recursos do Elastic Beanstalk com o AWS Config](#)
- [Usar o Elastic Beanstalk com o Amazon DynamoDB](#)
- [Usar o Elastic Beanstalk com o Amazon ElastiCache](#)
- [Usar o Elastic Beanstalk com o Amazon Elastic File System](#)
- [Usar o Elastic Beanstalk com o AWS Identity and Access Management](#)
- [Usar o Elastic Beanstalk com o Amazon RDS](#)
- [Usar o Elastic Beanstalk com o Amazon S3](#)
- [Usar o Elastic Beanstalk com Amazon VPC](#)

## Visão geral da arquitetura

O diagrama a seguir ilustra uma arquitetura demonstrativa do Elastic Beanstalk em várias zonas de disponibilidade que trabalham com outros produtos da AWS, como Amazon CloudFront, Amazon Simple Storage Service (Amazon S3) e Amazon Relational Database Service (Amazon RDS).



Para planejar a tolerância a falhas, é recomendável ter N+1 instâncias do Amazon EC2 e distribuí-las em várias zonas de disponibilidade. No caso improvável de uma zona de disponibilidade ficar inativa, você ainda terá outras instâncias do Amazon EC2 em execução em outra zona de disponibilidade. É possível ajustar o Amazon EC2 Auto Scaling para permitir um número mínimo de instâncias, bem como várias zonas de disponibilidade. Para obter instruções sobre como fazer isso, consulte [Grupo do Auto Scaling para o seu ambiente do Elastic Beanstalk](#). Para obter mais informações sobre a construção de aplicações tolerantes a falhas, consulte [Construção de aplicações tolerantes a falhas na AWS](#).

As seções a seguir discutem mais detalhadamente a integração com o Amazon CloudFront, Amazon CloudWatch, Amazon DynamoDB Amazon ElastiCache, Amazon RDS, Amazon Route 53, Amazon Simple Storage Service, Amazon VPC e IAM.

## Usar o Elastic Beanstalk com o Amazon CloudFront

O Amazon CloudFront é um serviço da web que agiliza a distribuição do seu conteúdo web estático e dinâmico, por exemplo arquivos .html, .css, .php e arquivos de imagem e mídia, para usuários finais. O CloudFront distribui seu conteúdo através de uma rede global de pontos de presença. Quando um usuário final solicita um conteúdo que você está servindo com o CloudFront, o usuário é direcionado ao ponto de presença com a menor latência, de forma que o conteúdo é entregue com o melhor desempenho possível. Se o conteúdo já está naquele ponto de presença, o CloudFront o entrega imediatamente. Se o conteúdo ainda não está naquele ponto de presença, o CloudFront o



recupera em um bucket do Amazon S3 ou um servidor HTTP (um servidor web, por exemplo) que você identificou como fonte da versão definitiva do seu conteúdo.

Depois de criar e implantar a aplicação do Elastic Beanstalk, você pode se cadastrar no CloudFront e começar a usar o CloudFront para distribuir o conteúdo. Saiba mais sobre o CloudFront no [Guia do desenvolvedor do Amazon CloudFront](#).

## Registro em log de chamadas da API do Elastic Beanstalk com AWS CloudTrail

O Elastic Beanstalk é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, uma função ou um produto da AWS no Elastic Beanstalk. O CloudTrail captura todas as chamadas de API para o Elastic Beanstalk como eventos, incluindo chamadas do console do Elastic Beanstalk, da CLI do EB e do seu código para as APIs do Elastic Beanstalk. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para o Elastic Beanstalk. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history (Histórico de eventos). Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita para o Elastic Beanstalk, o endereço IP no qual foi feita a solicitação, quem a fez e quando ela foi feita, além de detalhes adicionais.

Para saber mais sobre o CloudTrail, consulte o [Guia do usuário do AWS CloudTrail](#).

### Informações do Elastic Beanstalk no CloudTrail

O CloudTrail é habilitado em sua conta da AWS quando ela é criada. Quando ocorre atividade no Elastic Beanstalk, essa atividade é registrada em um evento do CloudTrail junto com outros eventos de serviços da AWS em Histórico de eventos. Você pode visualizar, pesquisar e fazer download de eventos recentes em sua conta da AWS. Para mais informações, consulte [Visualizar eventos com o histórico de eventos do CloudTrail](#).

Para obter um registro de eventos em andamento na sua conta da AWS, incluindo eventos do Elastic Beanstalk, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões. A trilha registra em log eventos de todas as regiões na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, você pode configurar outros produtos da AWS para analisar mais profundamente e agir sobre os dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- [Visão geral da criação de uma trilha](#)
- [Serviços e integrações compatíveis com o CloudTrail](#)
- [Configuração de notificações do Amazon SNS para o CloudTrail](#)
- [Recebimento de arquivos de log do CloudTrail de várias regiões](#) e [Recebimento de arquivos de log do CloudTrail de várias contas](#)

Todas as ações do Elastic Beanstalk são registradas pelo CloudTrail e estão documentadas na [Referência da API do AWS Elastic Beanstalk](#). Por exemplo, as chamadas para as APIs `DescribeApplications`, `UpdateEnvironment` e `ListTagsForResource` geram entradas nos arquivos de log do CloudTrail.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário raiz ou do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro produto da AWS.

Para mais informações, consulte o [Elemento `userIdentity` do CloudTrail](#).

## Noções básicas sobre entradas de arquivo de log do Elastic Beanstalk

Uma trilha é uma configuração que permite a entrega de eventos como registros de log a um bucket do Amazon S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada das chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada do log do CloudTrail que demonstra a ação `UpdateEnvironment` chamada por um usuário do IAM denominado `intern`, para o ambiente `sample-env` na aplicação `sample-app`.

```
{
 "Records": [{
 "eventVersion": "1.05",
```

```
"userIdentity": {
 "type": "IAMUser",
 "principalId": "AIXDAYQEXAMPLEUMLYNGL",
 "arn": "arn:aws:iam::123456789012:user/intern",
 "accountId": "123456789012",
 "accessKeyId": "ASXIAGXEXAMPLEQULKNXV",
 "userName": "intern",
 "sessionContext": {
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2016-04-22T00:23:24Z"
 }
 },
 "invokedBy": "signin.amazonaws.com"
},
"eventTime": "2016-04-22T00:24:14Z",
"eventSource": "elasticbeanstalk.amazonaws.com",
"eventName": "UpdateEnvironment",
"awsRegion": "us-west-2",
"sourceIPAddress": "255.255.255.54",
"userAgent": "signin.amazonaws.com",
"requestParameters": {
 "applicationName": "sample-app",
 "environmentName": "sample-env",
 "optionSettings": []
},
"responseElements": null,
"requestID": "84ae9ecf-0280-17ce-8612-705c7b132321",
"eventID": "e48b6a08-c6be-4a22-99e1-c53139cbfb18",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}]
}
```

## Usar o Elastic Beanstalk com o Amazon CloudWatch

O Amazon CloudWatch permite monitorar, gerenciar e publicar várias métricas, bem como configurar ações de alarme com base em dados de métricas. O monitoramento do Amazon CloudWatch permite coletar, analisar e visualizar métricas de sistema e aplicação, para que você possa tomar decisões operacionais e comerciais mais rapidamente e com maior confiança.

Você pode usar o Amazon CloudWatch para coletar métricas sobre seus recursos da Amazon Web Services (AWS), como a performance de suas instâncias do Amazon EC2. Você também pode publicar suas próprias métricas diretamente no Amazon CloudWatch. Os alarmes do Amazon CloudWatch ajudam você a implementar decisões mais facilmente, permitindo enviar notificações ou fazer alterações automaticamente nos recursos que está monitorando, com base nas regras definidas. Por exemplo, é possível criar alarmes que iniciam ações do Auto Scaling do Amazon EC2 e do Amazon Simple Notification Service (Amazon SNS) em seu nome.

O Elastic Beanstalk usa automaticamente o Amazon CloudWatch para ajudar você a monitorar o status da aplicação e do ambiente. É possível navegar até o console do Amazon CloudWatch para ver seu painel e obter uma visão geral de todos os seus recursos, bem como de seus alarmes. Você também pode escolher visualizar mais métricas ou adicionar métricas personalizadas.

Para obter mais informações sobre o Amazon CloudWatch, consulte o [Guia do desenvolvedor do Amazon CloudWatch](#). Para obter um exemplo de como usar o Amazon CloudWatch com o Elastic Beanstalk, consulte [the section called “Exemplo: como usar métricas personalizadas do Amazon CloudWatch”](#).

## Usar o Elastic Beanstalk com o Amazon CloudWatch Logs

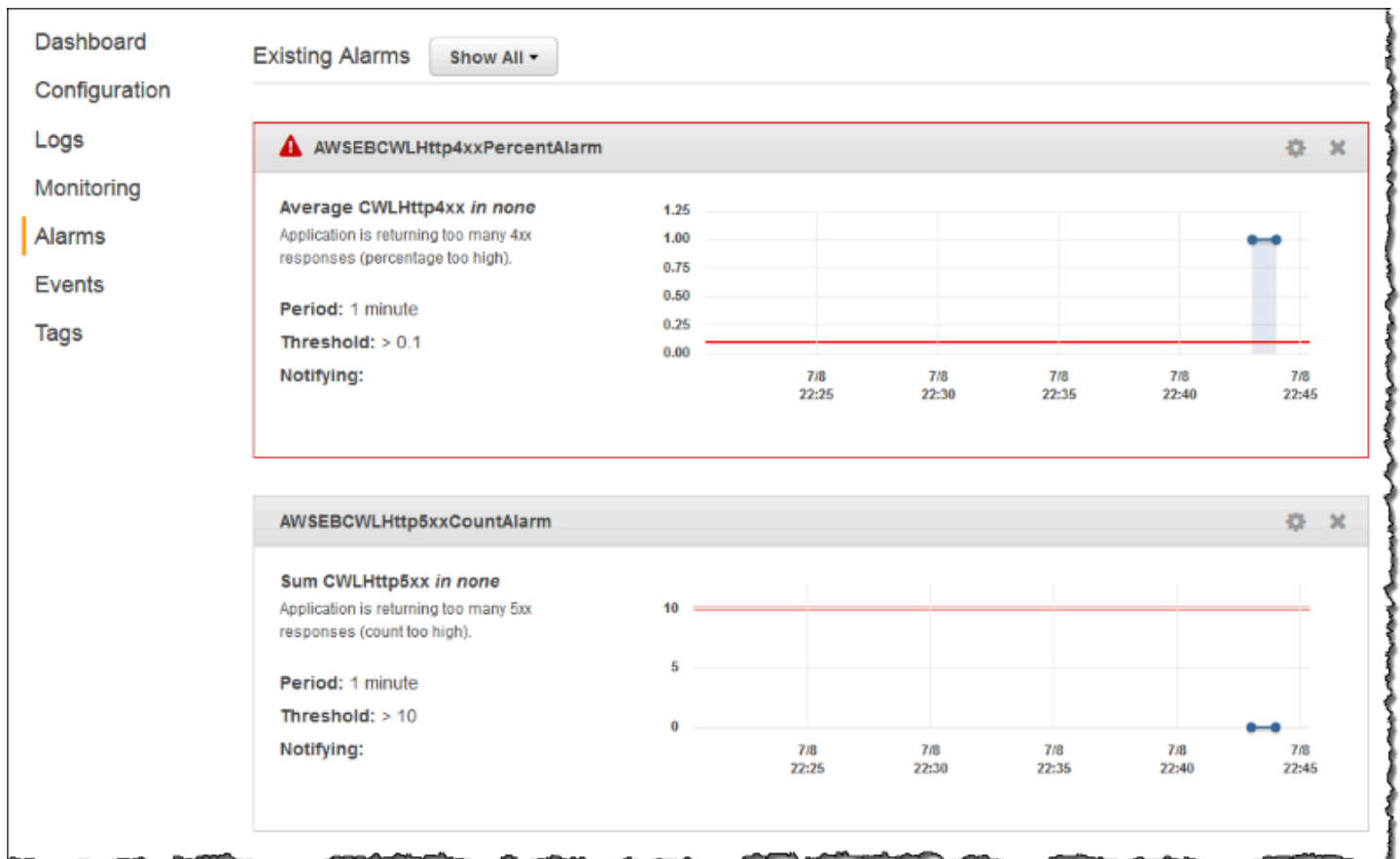
Com o CloudWatch Logs, é possível monitorar e arquivar seus arquivos de log personalizados, do sistema e da aplicação do Elastic Beanstalk a partir de instâncias do Amazon EC2 de seus ambientes. Além disso, você pode configurar alarmes para que reaja mais facilmente a eventos específicos de stream de logs que os filtros de métrica extraem. O agente do CloudWatch Logs instalado em cada instância do Amazon EC2 no ambiente publica pontos de dados de métrica para o serviço CloudWatch para cada grupo de logs configurado. Cada grupo de logs aplica seus próprios padrões de filtro para determinar quais eventos de stream de log serão enviados ao CloudWatch como pontos de dados. Os streams de log que pertencem ao mesmo grupo de log compartilham as mesmas configurações de retenção, monitoramento e controle de acesso. É possível configurar o Elastic Beanstalk para fazer streaming automaticamente de logs para o serviço CloudWatch, conforme descrito em [Fazer streaming de logs de instâncias para o CloudWatch Logs](#). Para obter mais informações sobre o CloudWatch Logs, incluindo terminologia e conceitos, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).

Além de logs da instância, se você habilitar a [integridade avançada](#) para seu ambiente, poderá configurar o ambiente para fazer streaming de informações de integridade para o CloudWatch Logs. Consulte [Fazer streaming de informações de integridade do ambiente do Elastic Beanstalk para o Amazon CloudWatch Logs](#).

A figura a seguir mostra a página Monitoring (Monitoramento) e gráficos de um ambiente configurado com a integração do CloudWatch Logs. As métricas de exemplo neste ambiente são chamadas CWLHttp4xx e CWLHttp5xx. Um dos gráficos mostra que a métrica CWLHttp4xx acionou um alarme com base nas condições especificadas nos arquivos de configuração.



A figura a seguir mostra a página Alarms (Alarmes) e gráficos para os exemplos de alarmes denominados AWSEBCWLHttp4xxPercentAlarm e AWSEBCWLHttp5xxCountAlarm, que correspondem às métricas CWLHttp4xx e CWLHttp5xx, respectivamente.



## Tópicos

- [Pré-requisitos para streaming de logs de instância para o CloudWatch Logs](#)
- [Como o Elastic Beanstalk configura o CloudWatch Logs](#)
- [Fazer streaming de logs de instâncias para o CloudWatch Logs](#)
- [Solucionar problemas de integração do CloudWatch Logs](#)
- [Fazer streaming de informações de integridade do ambiente do Elastic Beanstalk para o Amazon CloudWatch Logs](#)

## Pré-requisitos para streaming de logs de instância para o CloudWatch Logs

Para habilitar o streaming de logs de instâncias do Amazon EC2 do ambiente para o CloudWatch Logs, é necessário atender às condições a seguir.

- Plataforma: esse recurso só está disponível em versões de plataforma lançadas a partir [desta versão](#), portanto, se você estiver usando uma versão de plataforma anterior, atualize o ambiente para um atual.

- Caso não tenha a política gerenciada `AWSElasticBeanstalkWebTier` ou `AWSElasticBeanstalkWorkerTier` do Elastic Beanstalk em seu [perfil de instância do Elastic Beanstalk](#), adicione o seguinte ao seu perfil para habilitar esse recurso.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "logs:PutLogEvents",
 "logs:CreateLogStream"
],
 "Resource": [
 "*"
]
 }
]
}
```

## Como o Elastic Beanstalk configura o CloudWatch Logs

O Elastic Beanstalk instala um agente de log do CloudWatch com as configurações padrão em cada instância criada. Saiba mais na [Referência do agente do CloudWatch Logs](#).

Quando o streaming de log de instâncias é habilitado para o CloudWatch Logs, o Elastic Beanstalk envia arquivos de log das instâncias do seu ambiente para o CloudWatch Logs. Plataformas diferentes transmitem logs diferentes. A tabela a seguir lista os logs por plataforma.

Plataforma/ramificação da plataforma	Logs
Docker/ ramificação da plataforma: a: Docker em execução no Amazon Linux 2 de 64 bits	<ul style="list-style-type: none"><li>• <code>/var/log/eb-engine.log</code></li><li>• <code>/var/log/eb-hooks.log</code></li><li>• <code>/var/log/docker</code></li><li>• <code>/var/log/docker-events.log</code></li><li>• <code>/var/log/eb-docker/containers/eb-current-app/stdouterr.log</code></li><li>• <code>/var/log/nginx/access.log</code></li></ul>

Plataforma/ramificação da plataforma	Logs
	<ul style="list-style-type: none"> <li>• /var/log/nginx/error.log</li> </ul>
Docker/  ramificação da plataforma: ECS em execução no Amazon Linux 2 de 64 bits	<ul style="list-style-type: none"> <li>• /var/log/docker-events.log</li> <li>• /var/log/eb-ecs-mgr.log</li> <li>• /var/log/eb-engine.log</li> <li>• /var/log/eb-hooks.log</li> <li>• /var/log/ecs/ecs-agent.log</li> <li>• /var/log/ecs/ecs-init.log</li> </ul>
Go  .NET Core no Linux  Java/ramificação da plataforma: Corretto em execução no Amazon Linux 2 de 64 bits	<ul style="list-style-type: none"> <li>• /var/log/eb-engine.log</li> <li>• /var/log/eb-hooks.log</li> <li>• /var/log/web.stdout.log</li> <li>• /var/log/nginx/access.log</li> <li>• /var/log/nginx/error.log</li> </ul>
Node.js  Python	<ul style="list-style-type: none"> <li>• /var/log/eb-engine.log</li> <li>• /var/log/eb-hooks.log</li> <li>• /var/log/web.stdout.log</li> <li>• /var/log/httpd/access_log</li> <li>• /var/log/httpd/error_log</li> <li>• /var/log/nginx/access.log</li> <li>• /var/log/nginx/error.log</li> </ul>
Tomcat  PHP	<ul style="list-style-type: none"> <li>• /var/log/eb-engine.log</li> <li>• /var/log/eb-hooks.log</li> <li>• /var/log/httpd/access_log</li> <li>• /var/log/httpd/error_log</li> <li>• /var/log/nginx/access.log</li> <li>• /var/log/nginx/error.log</li> </ul>



Plataforma/ramificação da plataforma	Logs
.NET no Windows Server	<ul style="list-style-type: none"> <li>• C:\inetpub\logs\LogFiles\W3SVC1\u_ex*.log</li> <li>• C:\Program Files\Amazon\ElasticBeanstalk\logs\AWSDeployment.log</li> <li>• C:\Program Files\Amazon\ElasticBeanstalk\logs\Hooks.log</li> </ul>
Ruby	<ul style="list-style-type: none"> <li>• /var/log/eb-engine.log</li> <li>• /var/log/eb-hooks.log</li> <li>• /var/log/puma/puma.log</li> <li>• /var/log/web.stdout.log</li> <li>• /var/log/nginx/access.log</li> <li>• /var/log/nginx/error.log</li> </ul>

## Arquivos de log em plataformas da AMI do Amazon Linux

### Note

Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte [Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2](#).

A tabela a seguir lista os arquivos de log transmitidos de instâncias em ramificações da plataforma com base na AMI do Amazon Linux (anterior ao Amazon Linux 2), por plataforma.

Plataforma/ramificação da plataforma	Logs
Docker/	<ul style="list-style-type: none"> <li>• /var/log/eb-activity.log</li> <li>• /var/log/nginx/error.log</li> <li>• /var/log/docker-events.log</li> </ul>

Plataforma/ramificação da plataforma	Logs
ramificação da plataforma: Docker em execução no Amazon Linux de 64 bits	<ul style="list-style-type: none"> <li>• /var/log/docker</li> <li>• /var/log/nginx/access.log</li> <li>• /var/log/eb-docker/containers/eb-current-app/stdouterr.log</li> </ul>
Docker/ ramificação da plataforma: Docker de vários contêineres em execução no Amazon Linux de 64 bits	<ul style="list-style-type: none"> <li>• /var/log/eb-activity.log</li> <li>• /var/log/ecs/ecs-init.log</li> <li>• /var/log/eb-ecs-mgr.log</li> <li>• /var/log/ecs/ecs-agent.log</li> <li>• /var/log/docker-events.log</li> </ul>
Glassfish (Docker pré-configurado)	<ul style="list-style-type: none"> <li>• /var/log/eb-activity.log</li> <li>• /var/log/nginx/error.log</li> <li>• /var/log/docker-events.log</li> <li>• /var/log/docker</li> <li>• /var/log/nginx/access.log</li> </ul>
Go	<ul style="list-style-type: none"> <li>• /var/log/eb-activity.log</li> <li>• /var/log/nginx/error.log</li> <li>• /var/log/nginx/access.log</li> </ul>
Java/ ramificação da plataforma: Java 8 em execução no Amazon Linux de 64 bits  Ramificação da plataforma: Java 7 em execução no Amazon Linux de 64 bits	<ul style="list-style-type: none"> <li>• /var/log/eb-activity.log</li> <li>• /var/log/nginx/access.log</li> <li>• /var/log/nginx/error.log</li> <li>• /var/log/web-1.error.log</li> <li>• /var/log/web-1.log</li> </ul>

Plataforma/ramificação da plataforma	Logs
Tomcat	<ul style="list-style-type: none"><li>• /var/log/eb-activity.log</li><li>• /var/log/httpd/error_log</li><li>• /var/log/httpd/access_log</li><li>• /var/log/nginx/error_log</li><li>• /var/log/nginx/access_log</li></ul>
Node.js	<ul style="list-style-type: none"><li>• /var/log/eb-activity.log</li><li>• /var/log/nodejs/nodejs.log</li><li>• /var/log/nginx/error.log</li><li>• /var/log/nginx/access.log</li><li>• /var/log/httpd/error.log</li><li>• /var/log/httpd/access.log</li></ul>
PHP	<ul style="list-style-type: none"><li>• /var/log/eb-activity.log</li><li>• /var/log/httpd/error_log</li><li>• /var/log/httpd/access_log</li></ul>
Python	<ul style="list-style-type: none"><li>• /var/log/eb-activity.log</li><li>• /var/log/httpd/error_log</li><li>• /var/log/httpd/access_log</li><li>• /opt/python/log/supervisord.log</li></ul>
Ruby/ ramificação da plataforma: Puma com Ruby em execução no Amazon Linux de 64 bits	<ul style="list-style-type: none"><li>• /var/log/eb-activity.log</li><li>• /var/log/nginx/error.log</li><li>• /var/log/puma/puma.log</li><li>• /var/log/nginx/access.log</li></ul>

Plataforma/ramificação da plataforma	Logs
Ruby/ ramificação da plataforma: Passenger com Ruby em execução no Amazon Linux de 64 bits	<ul style="list-style-type: none"> <li>• /var/log/eb-activity.log</li> <li>• /var/app/support/logs/passenger.log</li> <li>• /var/app/support/logs/access.log</li> <li>• /var/app/support/logs/error.log</li> </ul>

O Elastic Beanstalk configura grupos de logs no CloudWatch Logs para os vários arquivos de log dos quais ele faz streaming. Para recuperar arquivos de log específicos do CloudWatch Logs, é necessário saber o nome do grupo de logs correspondente. O esquema de nomenclatura do grupo de logs depende do sistema operacional da plataforma.

Para plataformas Linux, prefixe o local do arquivo de log na instância com `/aws/elasticbeanstalk/environment_name` para obter o nome do grupo de logs. Por exemplo, para recuperar o arquivo `/var/log/nginx/error.log`, especifique o grupo de logs `/aws/elasticbeanstalk/environment_name/var/log/nginx/error.log`.

Para plataformas Windows, consulte a tabela a seguir referente ao grupo de logs correspondente a cada arquivo de log.

Arquivo de log na instância	Grupo de logs
C:\Program Files\Amazon\ElasticBeanstalk\logs\AWSDeployent.log	/aws/elasticbeanstalk/<environment-name>/EBDeploy-Log
C:\Program Files\Amazon\ElasticBeanstalk\logs\Hooks.log	/aws/elasticbeanstalk/<environment-name>/EBHooks-Log
C:\inetpub\logs\LogFiles (todo o diretório)	/aws/elasticbeanstalk/<environment-name>/IIS-Log

## Fazer streaming de logs de instâncias para o CloudWatch Logs

É possível habilitar o streaming de logs de instâncias para o CloudWatch Logs usando o console do Elastic Beanstalk, a CLI do EB ou as opções de configuração.

Antes de habilitá-lo, configure as permissões do IAM a serem usadas com o agente do CloudWatch Logs. Você pode anexar a seguinte política personalizada ao [perfil da instância](#) atribuído a seu ambiente.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "logs:CreateLogStream",
 "logs:PutLogEvents",
 "logs:DescribeLogGroups",
 "logs:DescribeLogStreams"
],
 "Resource": [
 "*"
]
 }
]
}
```

## Streaming de logs de instâncias usando o console do Elastic Beanstalk

Como fazer streaming de logs de instâncias para o CloudWatch Logs

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).

4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Em Instance log streaming to CloudWatch Logs (Streaming de logs de instância para o CloudWatch Logs):
  - Ative Log streaming (Streaming de logs).
  - Defina Retention (Retenção) com o número de dias para salvar os logs.
  - Selecione a configuração Lifecycle (Ciclo de vida) que determina se os logs devem ser salvos depois que o ambiente é encerrado.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Depois de ativar o streaming de logs, você poderá voltar para a categoria ou a página de configuração Software para encontrar o link Log Groups (Grupos de logs). Clique nesse link para ver logs no console do CloudWatch.

## Streaming de logs de instância usando a EB CLI

Para habilitar o streaming de logs de instância para o CloudWatch Logs usando a CLI do EB, use o comando [eb logs](#).

```
$ eb logs --cloudwatch-logs enable
```

Você também pode usar `eb logs` para recuperar logs do CloudWatch Logs. Você pode recuperar todos os logs de instância do ambiente ou usar as várias opções do comando para especificar subconjuntos de logs que devem ser recuperados. Por exemplo, o comando a seguir recupera o conjunto completo de logs de instância para seu ambiente e salva todos eles em um diretório em `.elasticbeanstalk/logs`.

```
$ eb logs --all
```

Especificamente, a opção `--log-group` permite que você recupere logs da instância de um grupo de logs específico, correspondente a um arquivo de log específico na instância. Para fazer isso, você precisa saber o nome do grupo de logs que corresponde ao arquivo de log que você deseja recuperar. Você pode encontrar essas informações em [Como o Elastic Beanstalk configura o CloudWatch Logs](#).

## Streaming de logs de instância usando arquivos de configuração

Ao criar ou atualizar um ambiente, você pode usar um arquivo de configuração para definir e configurar o streaming de logs de instância para o CloudWatch Logs. O exemplo de arquivo de configuração a seguir ativa o streaming de logs de instância padrão. O Elastic Beanstalk faz streaming do conjunto padrão de arquivos de log para a plataforma do ambiente. Para usar o exemplo, copie o texto em um arquivo com a extensão `.config` no diretório `.ebextensions`, no nível superior do pacote de origem do aplicativo.

```
option_settings:
 - namespace: aws:elasticbeanstalk:cloudwatch:logs
 option_name: StreamLogs
 value: true
```

## Streaming do arquivo de log personalizado

A integração do Elastic Beanstalk ao CloudWatch Logs não é diretamente compatível com o streaming de arquivos de log personalizados gerados pela aplicação. Para fazer streaming de logs personalizados, use um arquivo de configuração para instalar diretamente o agente do CloudWatch Logs e configurar os arquivos a serem enviados. Para obter um arquivo de configuração demonstrativo, consulte [logs-streamtocloudwatch-linux.config](#).

### Note

O exemplo não funciona na plataforma Windows.

Para obter mais informações sobre como configurar o CloudWatch Logs, consulte a [Referência do agente do CloudWatch Logs](#) no Guia do usuário do Amazon CloudWatch Logs.

## Solucionar problemas de integração do CloudWatch Logs

Se não conseguir encontrar alguns dos logs de instância do ambiente que você espera encontrar no CloudWatch Logs, investigue os problemas comuns a seguir:

- Sua função do IAM não tem as permissões do IAM necessárias.
- Você iniciou o ambiente em uma região da Região da AWS que não é compatível com o CloudWatch Logs.
- Um de seus arquivos de log personalizados não existe no caminho especificado.

## Fazer streaming de informações de integridade do ambiente do Elastic Beanstalk para o Amazon CloudWatch Logs

Se você habilitar o relatório de [integridade avançada](#) para seu ambiente, poderá configurar o ambiente para fazer streaming de informações de integridade para o CloudWatch Logs. Esse streaming é independente do streaming de logs de instância do Amazon EC2. Este tópico descreve o processo de streaming de informações de integridade do ambiente. Para obter mais informações sobre streaming de logs de instância, consulte [Usar o Elastic Beanstalk com o Amazon CloudWatch Logs](#).

Quando você configura o streaming de integridade do ambiente, o Elastic Beanstalk cria um grupo de logs do CloudWatch Logs para a integridade do ambiente. O nome do grupo de logs é `/aws/elasticbeanstalk/environment-name/environment-health.log`. Dentro desse grupo de logs, o Elastic Beanstalk cria streams de logs denominados `YYYY-MM-DD#<hash-suffix>` (pode haver mais de um stream de logs por data).

Quando o status de integridade do ambiente é alterado, o Elastic Beanstalk adiciona um registro ao stream de logs de integridade. O registro representa a transição do status de integridade – o novo status e uma descrição do motivo da alteração. Por exemplo, o status de um ambiente pode se alterar para Grave porque o load balancer está falhando. Para obter uma descrição do status de integridade avançada, consulte [Status e cores de integridade](#).

### Pré-requisitos para fazer streaming da integridade do ambiente para o CloudWatch Logs

Para habilitar o streaming de integridade do ambiente para o CloudWatch Logs, é necessário atender às seguintes condições:

- **Plataforma:** você precisa usar uma versão de plataforma compatível com os relatórios de integridade aprimorada.
- **Permissões:** é necessário conceder determinadas permissões relacionadas ao registro em log ao Elastic Beanstalk para que ele possa atuar em seu nome com o objetivo de fazer streaming de informações de integridade do ambiente. Se o ambiente não estiver usando uma função de serviço criada pelo Elastic Beanstalk para ele, `aws-elasticbeanstalk-service-role` ou a função vinculada ao serviço de sua conta, `AWSServiceRoleForElasticBeanstalk`, lembre-se de adicionar as permissões a seguir à sua função de serviço personalizada.

```
{
```



```
"Effect": "Allow",
"Action": [
 "logs:DescribeLogStreams",
 "logs:CreateLogStream",
 "logs:PutLogEvents"
],
"Resource": "arn:aws:logs:*:*:log-group:/aws/elasticbeanstalk/*:log-stream:*"
}
```

## Fazer streaming de logs de integridade do ambiente para o CloudWatch Logs

É possível habilitar o streaming de integridade do ambiente para o CloudWatch Logs usando o console do Elastic Beanstalk, a CLI do EB ou as opções de configuração.

Fazer streaming de logs de integridade do ambiente usando o console do Elastic Beanstalk

Como fazer streaming de logs de integridade do ambiente para o CloudWatch Logs

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Monitoring (Monitoramento), escolha Edit (Editar).
5. Em Health reporting (Relatório de integridade), verifique se System (Sistema) está definido como Enhanced (Avançado).
6. Em Health event streaming to CloudWatch Logs (Streaming de eventos de integridade para o CloudWatch Logs)
  - Ative Log streaming (Streaming de logs).
  - Defina Retention (Retenção) com o número de dias para salvar os logs.
  - Selecione a configuração Lifecycle (Ciclo de vida) que determina se os logs devem ser salvos depois que o ambiente é encerrado.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Depois de ativar o streaming de logs, você poderá voltar para a categoria ou a página de configuração Monitoring (Monitoramento) para encontrar o link Log Group (Grupo de logs). Clique neste link para ver os logs de integridade do ambiente no console do CloudWatch.

## Streaming de logs de integridade do ambiente usando a EB CLI

Para habilitar o streaming de logs de integridade do ambiente para o CloudWatch Logs usando a CLI do EB, use o comando [eb logs](#).

```
$ eb logs --cloudwatch-logs enable --cloudwatch-log-source environment-health
```

Você também pode usar `eb logs` para recuperar logs do CloudWatch Logs. Por exemplo, o comando a seguir recupera todos os logs de integridade para seu ambiente e os salva em um diretório em `.elasticbeanstalk/logs`.

```
$ eb logs --all --cloudwatch-log-source environment-health
```

## Streaming de logs de integridade do ambiente usando arquivos de configuração

Ao criar ou atualizar um ambiente, você pode usar um arquivo de configuração para definir e configurar o streaming de logs de integridade do ambiente para o CloudWatch Logs. Para usar o exemplo abaixo, copie o texto em um arquivo com a extensão `.config` no diretório `.ebextensions`, no nível superior do pacote de origem de seu aplicativo. O exemplo configura o Elastic Beanstalk para habilitar o streaming de logs de integridade do ambiente, manter os logs depois de encerrar o ambiente e salvá-los por 30 dias.

### Example [Arquivo de configuração de streaming de integridade](#)

```

Sets up Elastic Beanstalk to stream environment health information
to Amazon CloudWatch Logs.
Works only for environments that have enhanced health reporting enabled.

option_settings:
 aws:elasticbeanstalk:cloudwatch:logs:health:
 HealthStreamingEnabled: true
 ### Settings below this line are optional.
 # DeleteOnTerminate: Delete the log group when the environment is
 # terminated. Default is false. If false, the health data is kept
```

```
RetentionInDays days.
DeleteOnTerminate: false
RetentionInDays: The number of days to keep the archived health data
before it expires, if DeleteOnTerminate isn't set. Default is 7 days.
RetentionInDays: 30
```

Sobre opções padrão e valores válidos, consulte

[aws:elasticbeanstalk:cloudwatch:logs:health](#).

## Usar o Elastic Beanstalk com o Amazon EventBridge

Usando o Amazon EventBridge, você pode configurar regras orientadas por eventos que monitoram os recursos do Elastic Beanstalk e iniciam ações de destino que usam outros produtos da AWS. Por exemplo, você pode definir uma regra para enviar notificações por e-mail sinalizando um tópico do Amazon SNS sempre que a integridade de um ambiente de produção mudar para um status de Warning (Aviso). Ou você pode definir uma função do Lambda para passar uma notificação ao Slack sempre que a integridade do seu ambiente mudar para um status Degradated (Degradado) ou Severe (Grave).

Você pode criar regras no Amazon EventBridge para atuar em qualquer um dos seguintes eventos do Elastic Beanstalk:

- Alterações de estado para operações de ambiente (incluindo operações de criação, atualização e encerramento). O evento especificará se a alteração de estado foi iniciada, bem-sucedida ou falhou.
- Alterações de estado para outros recursos. Além dos ambientes, outros recursos monitorados incluem balanceadores de carga, grupos de escalabilidade automática e instâncias.
- Transição de integridade para ambientes. O evento indica onde a integridade do ambiente passou de um determinado estado para outro.
- Alteração de estado para atualizações gerenciadas. O evento especificará se a alteração de estado foi iniciada, bem-sucedida ou falhou.

Para capturar eventos específicos do Elastic Beanstalk de seu interesse, defina padrões específicos do evento que o EventBridge possa usar para detectar os eventos. Os padrões de eventos têm a mesma estrutura que os eventos aos quais correspondem. O padrão menciona os campos com os quais você deseja fazer a correspondência e fornece os valores que você está procurando. Os eventos são emitidos com base no melhor esforço. Eles são entregues do Elastic Beanstalk para o

EventBridge quase em tempo real em circunstâncias operacionais normais. No entanto, podem surgir situações que podem atrasar ou impedir a entrega de um evento.

Para obter uma lista de campos incluídos nos eventos do Elastic Beanstalk e seus possíveis valores de string, consulte [Mapeamento de campo de evento do Elastic Beanstalk](#). Para obter informações sobre como as regras do EventBridge funcionam com padrões de eventos, consulte [Padrões de eventos e eventos no EventBridge](#).

## Monitorar um recurso Elastic Beanstalk com o EventBridge

Com o EventBridge, você pode criar regras que definem ações a serem executadas quando o Elastic Beanstalk emitir eventos para seus recursos. Por exemplo, você pode criar uma regra que envia uma mensagem de e-mail sempre que o status de um ambiente for alterado.


O console do EventBridge tem uma opção de Pre-defined pattern (Padrão predefinido) para criar padrões de eventos do Elastic Beanstalk. Se você selecionar essa opção no console do EventBridge ao criar uma regra, poderá criar um padrão de evento do Elastic Beanstalk rapidamente. Você só precisa selecionar os campos e os valores do evento. Ao fazer seleções, o console constrói e exibe o padrão do evento. Como alternativa, você pode editar manualmente o padrão do evento que você criar e pode salvá-lo como um padrão personalizado. O console também fornece a opção para exibir um Sample Event (Exemplo de evento) detalhado que você pode copiar e colar no padrão de evento que está criando.

Se você preferir digitar ou copiar e colar um padrão de evento no console do EventBridge, poderá optar por usar a opção Padrão personalizado (Padrão personalizado) no console. Ao fazer isso, você não precisará passar pelas etapas de seleção de campos e valores descritos anteriormente. Este tópico oferece exemplos de [padrões de correspondência de eventos](#) e [eventos do Elastic Beanstalk](#) que você pode usar.

Para criar uma regra para um evento de recurso

1. Registre-se na AWS usando uma conta que tenha permissões para usar o EventBridge e o Elastic Beanstalk.
2. Abra o console do Amazon EventBridge em <https://console.aws.amazon.com/events/>.
3. No painel de navegação, escolha Rules (Regras).
4. Escolha Create rule (Criar regra).
5. Informe um Name (Nome) para a regra e, opcionalmente, uma descrição.

6. Em Event Bus (Barramento de eventos), escolha default (padrão). Quando um serviço da AWS em sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
7. Em Rule type (Tipo de regra), escolha Rule with an event pattern (Regra com um padrão de evento).
8. Escolha Next (Próximo).
9. Em Event source (Origem do evento), escolha Eventos da AWS ou eventos de parceiro do EventBridge.
10. (Opcional) Para Sample event (Exemplo de evento), selecione AWS events (Eventos da AWS). Insira Elastic Beanstalk no campo de pesquisa. Isso fornecerá uma lista de exemplos de eventos do Elastic Beanstalk que é possível escolher para exibição. Essa etapa simplesmente exibe um exemplo de evento que você pode referenciar. Isso não afeta o resultado da criação da regra. A seção [Eventos demonstrativos do Elastic Beanstalk](#) mais adiante neste tópico fornecerá exemplos do mesmo tipo de evento.
11. Na seção Event pattern (Padrão de evento), selecione Event pattern form (Formulário de padrão de evento).

 Note

Se você já tiver texto para um padrão de evento e não precisar do console do EventBridge para criá-lo, selecione Custom pattern (JSON editor) (Padrão personalizado - Editor JSON). Você pode inserir manualmente ou copiar e colar texto na caixa Event pattern (Padrão de evento). Selecione Next (Seguinte) e acesse a etapa sobre como inserir um destino.

12. Em Event source (Origem do evento), escolha AWS services (Serviços da ).
13. Em AWS service (Serviço da AWS), selecione Elastic Beanstalk.
14. Em Event type (Tipo de evento), selecione Status Change (Alteração de status).
15. Essa etapa aborda como você pode trabalhar com os campos de evento detail type (tipo de detalhe), status severity (gravidade) do Elastic Beanstalk. À medida que você escolhe esses campos e os valores que deseja corresponder, o console cria e exibe o padrão do evento.
  - Se você selecionar somente um valor para Specific detail type(s) (Tipos de detalhes específicos), poderá escolher um ou mais valores para o próximo campo na hierarquia.

- Se você escolher mais de um valor para Specific detail type(s) (Tipos de detalhes específicos), não escolha valores específicos para os próximos campos na hierarquia. Isso evita uma lógica de correspondência ambígua entre campos no seu padrão de evento.

O campo de evento environment (ambiente) não é afetado por essa hierarquia. Portanto, ele é exibido conforme descrito na próxima etapa.

16. Para o ambiente, selecione Any environment (Qualquer ambiente) ou Specific environment(s) (Ambientes específicos).
  - Se você selecionar Specific environment(s) (Ambientes específicos), poderá escolher um ou mais ambientes na lista suspensa. O EventBridge adiciona todos os ambientes que você seleciona dentro da lista EnvironmentName[ ] na seção detail (detalhe) do padrão de evento. Em seguida, sua regra filtra todos os eventos para incluir apenas os ambientes específicos que você escolher.
  - Se você selecionar Any environment (Qualquer ambiente), nenhum ambiente será adicionado ao seu padrão de evento. Por isso, sua regra não filtra qualquer dos eventos do Elastic Beanstalk com base no ambiente.
17. Escolha Next (Próximo).
18. Em Target types (Tipos de destino), escolha AWS service (Serviço da AWS).
19. Em Select a target (Selecionar um destino), escolha a ação de destino a ser executada quando um evento de alteração de estado de recurso for recebido do Elastic Beanstalk.

Por exemplo, é possível usar um tópico do Amazon Simple Notification Service (SNS) para enviar um e-mail ou mensagem de texto quando ocorrer um evento. Para fazer isso, você precisa criar um tópico do Amazon SNS usando o console do Amazon SNS. Para saber mais, consulte [Usar o Amazon SNS para notificações de usuários](#).

#### Important

Algumas ações de destino podem exigir o uso de outros serviços e gerar cobranças adicionais, como o Amazon SNS ou o serviço Lambda. Para obter mais informações sobre os preços da AWS, consulte <https://aws.amazon.com/pricing/>. Alguns serviços fazem parte do nível de uso gratuito da AWS. Se for um cliente novo, você pode testar esses serviços gratuitamente. Consulte <https://aws.amazon.com/free/> para obter mais informações.

20. (Opcional) Escolha Add another target (Adicionar outro destino) para especificar uma ação de destino adicional para a regra de evento.
21. Escolha Next (Próximo).
22. (Opcional) Insira uma ou mais tags para a regra. Para obter mais informações, consulte [Etiquetas do Amazon EventBridge](#) no Guia do usuário do Amazon EventBridge.
23. Escolha Next (Próximo).
24. Analise os detalhes da regra e escolha Create rule (Criar regra).

## Eventos demonstrativos de padrões de eventos do Elastic Beanstalk

Os padrões de eventos têm a mesma estrutura que os eventos aos quais correspondem. O padrão menciona os campos com os quais você deseja fazer a correspondência e fornece os valores que você está procurando.

- Health status change (Alteração de estado de integridade) para todos os ambientes

```
{
 "source": [
 "aws.elasticbeanstalk"
],
 "detail-type": [
 "Health status change"
]
}
```

- Health status change (Alteração do estado de integridade) para os seguintes ambientes: myEnvironment1 e myEnvironment2. Esse padrão de evento filtra para esses dois ambientes específicos, enquanto o exemplo anterior de Health status change (Alteração do estado de integridade) que não filtra envia eventos para todos os ambientes.

```
{"source": [
 "aws.elasticbeanstalk"
],
"detail-type": [
 "Health status change"
],
"detail": {
 "EnvironmentName": [
 "myEnvironment1",
```

```

 "myEnvironment2"
]
 }
 }
}

```

- Elastic Beanstalk resource status change (Alteração de status do recurso do Elastic Beanstalk) para todos os ambientes

```

{
 "source": [
 "aws.elasticbeanstalk"
],
 "detail-type": [
 "Elastic Beanstalk resource status change"
]
}

```

- Elastic Beanstalk resource status change (Alteração de status do recurso do Elastic Beanstalk) com Status a atualização do ambiente falhou e Severity ERRO para os seguintes ambientes: myEnvironment1 e myEnvironment2

```

{"source": [
 "aws.elasticbeanstalk"
],
"detail-type": [
 "Elastic Beanstalk resource status change"
],
"detail": {
 "Status": [
 "Environment update failed"
],
 "Severity": [
 "ERROR"
],
 "EnvironmentName": [
 "myEnvironment1",
 "myEnvironment2"
]
}
}

```

- Other resource status change (Alteração de status de outros recursos) para balanceadores de carga, grupos de Auto Scaling e instâncias



```
{
 "source": [
 "aws.elasticbeanstalk"
],
 "detail-type": [
 "Other resource status change"
]
}
```

- Managed update status change (Alteração de status de atualização gerenciada) para ambientes

```
{
 "source": [
 "aws.elasticbeanstalk"
],
 "detail-type": [
 "Managed update status change"
]
}
```

- Para capturar all events (todos os eventos) do Elastic Beanstalk (exclua a seção detail-type)

```
{
 "source": [
 "aws.elasticbeanstalk"
]
}
```

## Eventos demonstrativos do Elastic Beanstalk

Veja a seguir um evento demonstrativo do Elastic Beanstalk para uma alteração de status de recurso:

```
{
 "version":"0",
 "id":"1234a678-1b23-c123-12fd3f456e78",
 "detail-type":"Elastic Beanstalk resource status change",
 "source":"aws.elasticbeanstalk",
 "account":"111122223333",
 "time":"2020-11-03T00:31:54Z",
```

```
"region":"us-east-1",
"resources":[
 "arn:aws:elasticbeanstalk:us-east-1:111122223333:environment/myApplication/
myEnvironment"
],
"detail":{
 "Status":"Environment creation started",
 "EventDate":1604363513951,
 "ApplicationName":"myApplication",
 "Message":"createEnvironment is starting.",
 "EnvironmentName":"myEnvironment",
 "Severity":"INFO"
}
}
```

Veja a seguir um evento demonstrativo do Elastic Beanstalk para uma alteração de status de integridade:

```
{
 "version":"0",
 "id":"1234a678-1b23-c123-12fd3f456e78",
 "detail-type":"Health status change",
 "source":"aws.elasticbeanstalk",
 "account":"111122223333",
 "time":"2020-11-03T00:34:48Z",
 "region":"us-east-1",
 "resources":[
 "arn:aws:elasticbeanstalk:us-east-1:111122223333:environment/myApplication/
myEnvironment"
],
 "detail":{
 "Status":"Environment health changed",
 "EventDate":1604363687870,
 "ApplicationName":"myApplication",
 "Message":"Environment health has transitioned from Pending to Ok. Initialization
completed 1 second ago and took 2 minutes.",
 "EnvironmentName":"myEnvironment",
 "Severity":"INFO"
 }
}
```

## Mapeamento de campo de evento do Elastic Beanstalk

A tabela a seguir mapeia os campos de evento do Elastic Beanstalk e seus possíveis valores de string para o campo `detail-type` do EventBridge. Para obter mais informações sobre como o EventBridge funciona com padrões de eventos para um serviço, consulte [Padrões de eventos e eventos no EventBridge](#).

Campo do EventBridge <code>detail-type</code>	Campo do Elastic Beanstalk Status	Campo do Elastic Beanstalk Severity (Gravidade)	Campo do Elastic Beanstalk Message (Mensagem)
Alteração de status do recurso do Elastic Beanstalk	Criação de ambiente iniciada	INFO	<code>createEnvironment</code> está sendo iniciada.
	Criação de ambiente bem-sucedida	INFO	<code>createEnvironment</code> concluída com êxito.
	Criação de ambiente bem-sucedida	INFO	Ambiente executado: <Environment Name>. No entanto, houve problemas durante a execução. Consulte o log de eventos para obter detalhes.
	Assistent e de criação de ambiente	ERROR	Falha ao executar o ambiente.
	Atualização do ambiente iniciada	INFO	A atualização do ambiente está sendo iniciada.

Campo do EventBridge detail-type	Campo do Elastic Beanstalk Status	Campo do Elastic Beanstalk Severity (Gravidade)	Campo do Elastic Beanstalk Message (Mensagem)
	Atualização do ambiente bem-sucedida	INFO	Atualização do ambiente concluída com êxito.
	Falha na atualização do ambiente	ERROR	Falha ao implantar a configuração.
	Encerramento do ambiente iniciado	INFO	terminateEnvironment está sendo iniciado.
	Encerramento do ambiente bem-sucedido	INFO	terminateEnvironment concluído com êxito.
	Falha no encerramento do ambiente	INFO	A etapa de encerramento do ambiente falhou porque pelo menos um dos fluxos de trabalho de encerramento do ambiente falhou.
Alteração de status de outros recursos	Grupo de Auto Scaling criado	INFO	createEnvironment está sendo iniciada.

Campo do EventBridge detail-type	Campo do Elastic Beanstalk Status	Campo do Elastic Beanstalk Severity (Gravidade)	Campo do Elastic Beanstalk Message (Mensagem)
	Grupo de Auto Scaling excluído	INFO	createEnvironment está sendo iniciada.
	Instância adicionada	INFO	Instância adicionada [i-123456789a12b1234] ao ambiente.
	Instância removida	INFO	Instância removida [i-123456789a12b1234] do ambiente.
	Balanceador de carga criado	INFO	Balanceador de carga criado chamado: <LB Name>
	Balanceador de carga excluído	INFO	Balanceador de carga excluído chamado: <LB Name>
Alteração do estado de integridade	A integridade do ambiente mudou	INFO/AVISO	A integridade do ambiente passou para <healthStatus>.
	A integridade do ambiente mudou	INFO/AVISO	A integridade do ambiente passou de <healthStatus> para <healthStatus>.

Campo do EventBridge detail-type	Campo do Elastic Beanstalk Status	Campo do Elastic Beanstalk Severity (Gravidade)	Campo do Elastic Beanstalk Message (Mensagem)
Alteração de status de atualização gerenciada	Atualização gerenciada iniciada	INFO	A atualização da plataforma gerenciada está em andamento.
	Falha na atualização gerenciada	INFO	Falha na atualização gerenciada, tentando novamente em %s minutos.

## Localizar e acompanhar recursos do Elastic Beanstalk com o AWS Config

O [AWS Config](#) fornece uma visão detalhada da configuração dos recursos da AWS em sua AWS. Você pode ver como os recursos estão relacionados, obter um histórico de alterações de configuração e ver como os relacionamentos e as configurações foram alterados ao longo do tempo. O AWS Config pode ser usado para definir regras de avaliação das configurações de recurso para conformidade de dados.

Vários tipos de recursos do Elastic Beanstalk estão integrados ao AWS Config:

- Aplicativos
- Versões do aplicativo
- Ambientes do

A seção a seguir mostra como configurar o AWS Config para registrar recursos desses tipos.

Para obter mais informações sobre o AWS Config, consulte o [Guia do desenvolvedor do AWS Config](#). Para obter informações sobre a definição de preço, consulte a [página de informações sobre a definição de preço do AWS Config](#).

## Configurar o AWS Config

Para fazer a configuração inicial do AWS Config, consulte os tópicos a seguir no [Guia do desenvolvedor do AWS Config](#).

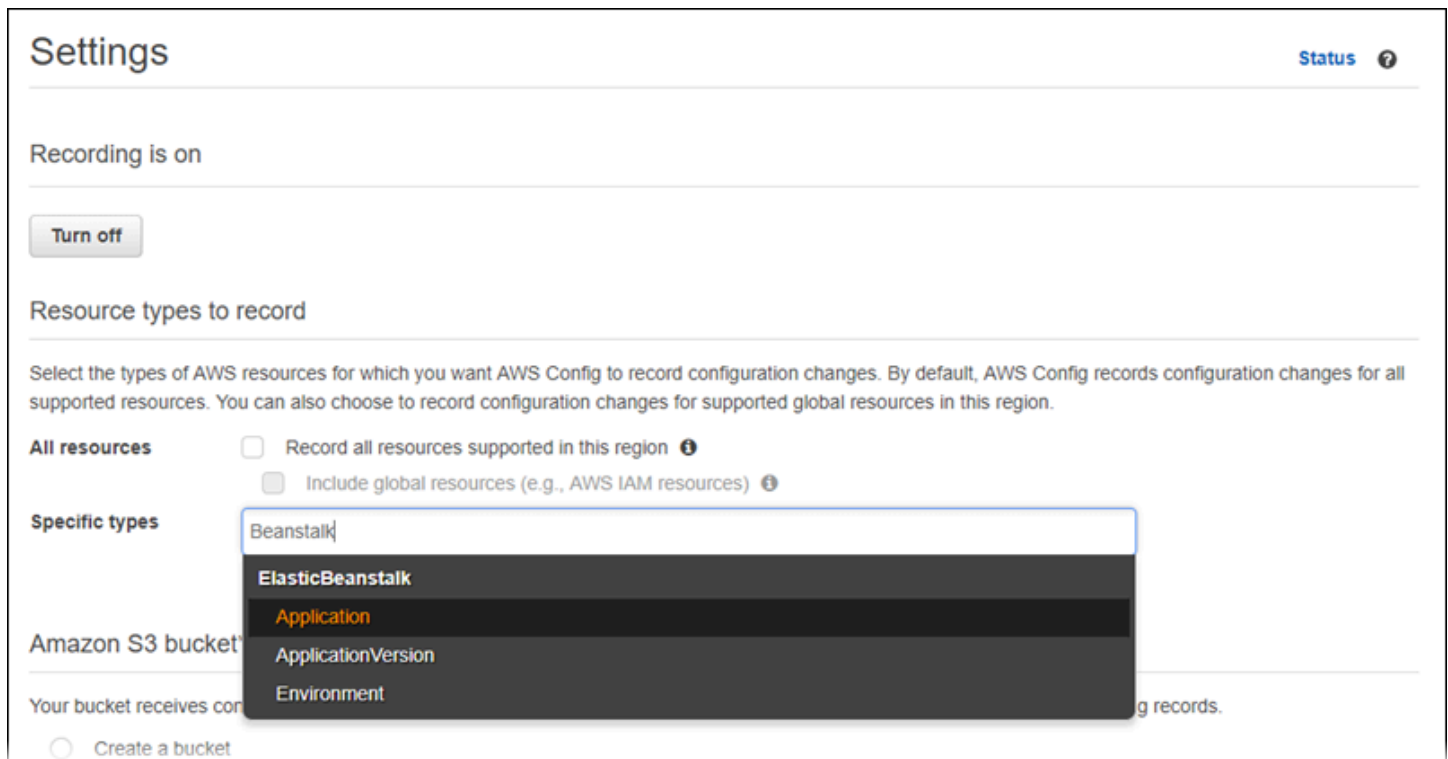
- [Configuração do AWS Config com o console](#)
- [Configuração do AWS Config com a AWS CLI](#)

## Configuração do AWS Config para registrar recursos do Elastic Beanstalk

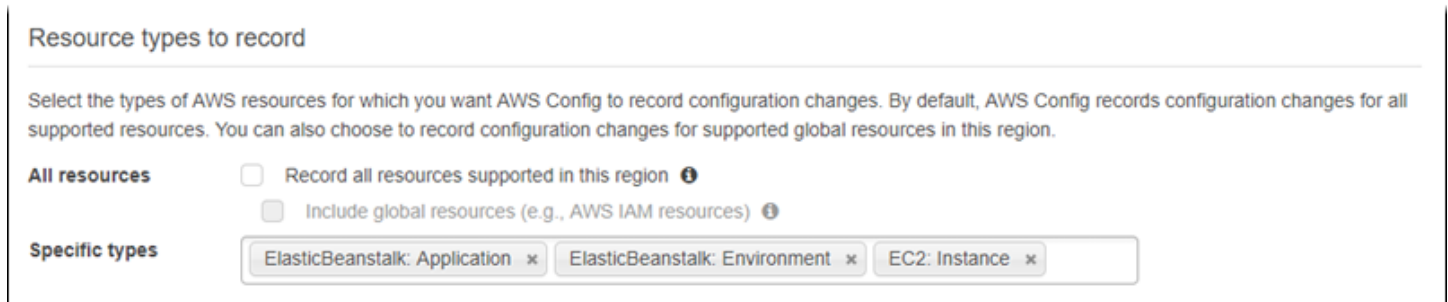
Por padrão, o AWS Config registra as alterações de configuração para todos os tipos de recursos regionais com suporte, que ele descobre na região em que seu ambiente está sendo executado. Você pode personalizar o AWS Config para registrar alterações somente para tipos de recursos específicos ou alterações em recursos globais.

Por exemplo, você pode configurar o AWS Config para registrar alterações nos recursos do Elastic Beanstalk e um subconjunto de outros recursos da AWS que o Elastic Beanstalk inicia para você. No [Console do AWS Config](#), você pode selecionar Elastic Beanstalk como um recurso na página Settings (Configurações) do AWS Config no campo Specific Types (Tipos específicos). A partir daí, você pode escolher registrar qualquer um dos tipos de recursos do Elastic Beanstalk: Application (Aplicação), Application Version (Versão da aplicação) e Environment (Ambiente).

A figura a seguir mostra a página AWS ConfigSettings (Configurações), com os tipos de recursos do Elastic Beanstalk que você pode escolher registrar: Application (Aplicação), ApplicationVersion (Versão da aplicação) e Environment (Ambiente)..



Depois de selecionar alguns tipos de recursos, esta é a forma que a lista dos Specific types (Tipos específicos) é exibida.



Para saber mais sobre os recursos regionais versus globais e para o procedimento de personalização completo, consulte [Selecionar quais recursos o AWS Config registra](#).

## Visualizar os detalhes da configuração do Elastic Beanstalk no console do AWS Config

É possível usar o console do AWS Config para procurar recursos do Elastic Beanstalk atuais e históricos e obter detalhes sobre suas configurações. O exemplo a seguir mostra como encontrar informações sobre um ambiente do Elastic Beanstalk.



## Como encontrar um ambiente do Elastic Beanstalk no console do AWS Config

1. Abra o [console do AWS Config](#).
2. Escolha Resources (Recursos).
3. Na página de inventário Resource (Recurso), selecione Resources (Recursos).
4. Abra o menu Resource type (Tipo de recurso), role a tela até ElasticBeanstalk e escolha um ou mais tipos de recurso do Elastic Beanstalk.

### Note

Para visualizar detalhes de configuração de outros recursos que o Elastic Beanstalk criou para a aplicação, escolha outros tipos de recursos. Por exemplo, você pode escolher Instance (Instância) em EC2.

5. Escolha Look up. Veja 2 na figura a seguir.

The screenshot shows the 'Resource inventory' page in the AWS Config console. At the top, there is a 'Status' link with a help icon. Below the title, a brief description explains that the page shows existing and deleted resources recorded by AWS Config, with options to view compliance details or configuration timelines. There are three radio buttons: 'Resources' (selected), 'Tag', and 'Compliance status'. A search bar labeled 'Resource identifier (optional)' is present. A 'Look up' button is highlighted with a yellow box and a callout bubble containing the number '2'. To the left, a dropdown menu is open, showing a list of resource types. The 'ElasticBeanstalk' category is highlighted with a yellow box and a callout bubble containing the number '1'. Under 'ElasticBeanstalk', the 'Application' and 'Environment' options are checked. Below the dropdown, a table lists resources with columns for 'Config timeline', 'Compliance', and 'Manage resource'. The first row shows a resource with ID 'i-0abae959f6fb4b133' and a 'Compliant' status. The second row shows an application configuration 'am:aws:elasticbeanstalk:us-east-1:270205402845:application/config-demo'. The third row shows an environment 'e-yaumygtbwr'.

Config timeline	Compliance	Manage resource
i-0abae959f6fb4b133	Compliant	<a href="#">Manage resource</a>
am:aws:elasticbeanstalk:us-east-1:270205402845:application/config-demo	--	
e-yaumygtbwr	--	

6. Escolha um ID de recurso na lista de recursos que o AWS Config exibe.

## Resource inventory Status ?


Look up existing and deleted resources recorded by AWS Config. View compliance details for each resource or choose the Config timeline icon to see how a particular resource's configuration has changed over time.



Resources  Tag  Compliance status

EC2: Instance, ElasticBeanstalk: ...

Include deleted resources

[Look up](#)

Choose Config timeline  to view a history of configuration details for the resource.

Resource type	Config timeline 	Compliance	Manage resource
▶ EC2 Instance	i-0abae959f6fb4b133	Compliant	
▶ ElasticBeanstalk Application	arn:aws:elasticbeanstalk:us-east-1:270205402845:application/config-demo	--	
▶ ElasticBeanstalk Environment	<b>e-yaumygtbwr</b>	--	

AWS ConfigO exibe detalhes de configuração e outras informações sobre o recurso que você selecionou.

## ElasticBeanstalk Environment e-yaumygtbwr

on February 09, 2018 4:03:54 PM Pacific Standard Time (UTC-08:00)

Manage resource ⓘ

← [ ] [ ] [ 05<sup>th</sup> February 2018 4:34:35 PM ] [ 06<sup>th</sup> February 2018 3:43:45 PM ] [ 07<sup>th</sup> February 2018 11:43:44 PM ] →

Now [ ] [ ]

1 Change 7 Changes

▼ Configuration Details [View Details](#)

**Amazon Resource Name** am:aws:elasticbeanstalk:us-east-1:270205402845:environment/config-demo/ConfigDemo-env

**Resource type** AWS::ElasticBeanstalk::Environment

**Resource ID** e-yaumygtbwr

**Resource name** ConfigDemo-env

**Availability zone** Not Applicable

**Created on** February 05, 2018 3:45:05 PM

**Tags (3)**

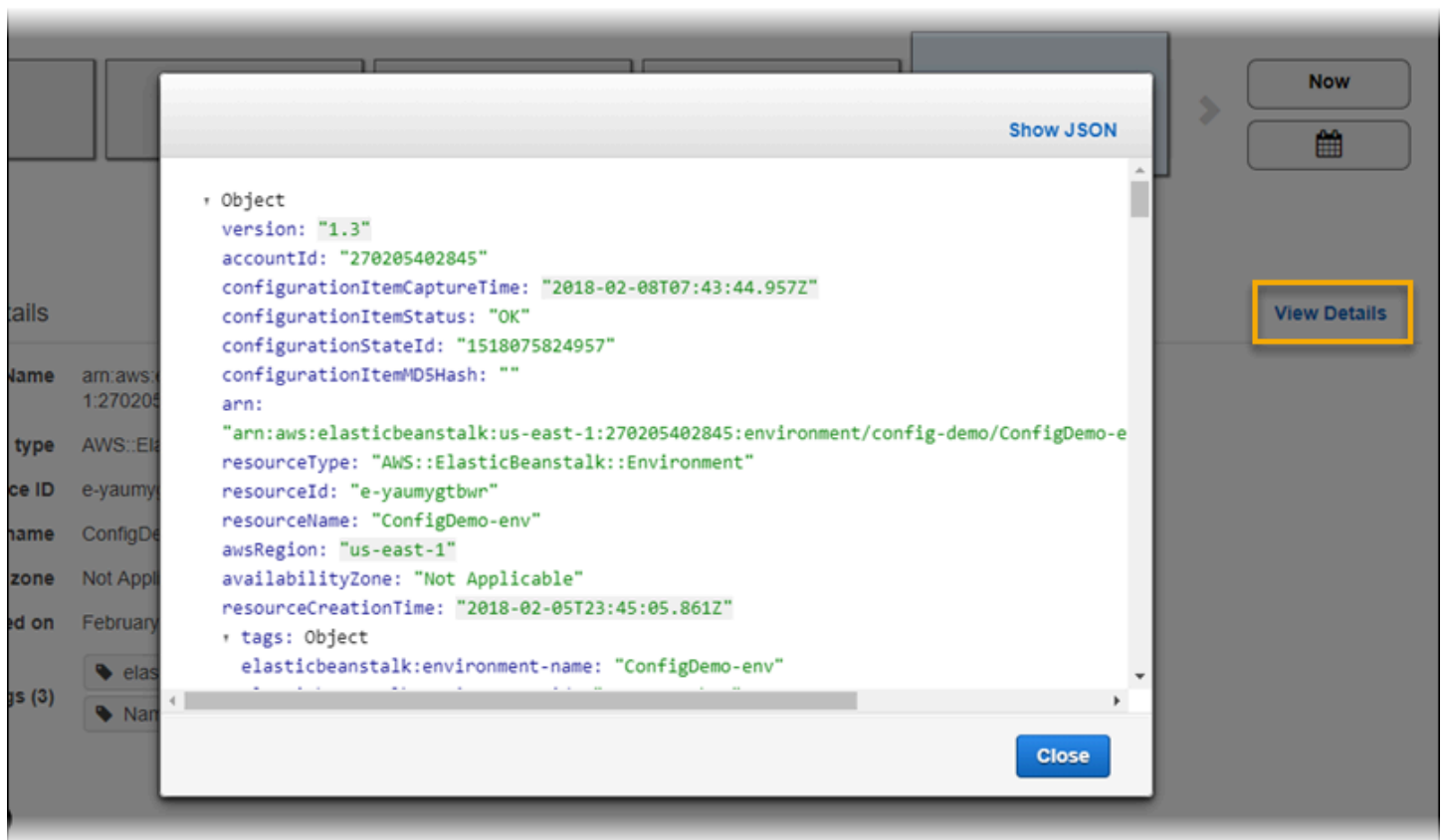
- elasticbeanstalk:envi...
- elasticbeanstalk:envi...
- Name: ConfigDemo-...

► Relationships 5

► Changes 7

► CloudTrail Events 6

Para ver os detalhes completos da configuração registrada, selecione View Details (Exibir detalhes).



Para saber mais maneiras de localizar um recurso e exibir informações nesta página, consulte [Exibindo configurações de recursos e histórico da AWS](#) no Guia do desenvolvedor do AWS Config.

## Avaliar recursos do Elastic Beanstalk usando regras do AWS Config

É possível criar regras do AWS Config que representam as definições de configuração ideais para seus recursos do Elastic Beanstalk. Você pode usar Regras de configuração gerenciadas pela AWS predefinidas ou definir regras personalizadas. O AWS Config monitora de forma contínua as alterações na configuração dos seus recursos para determinar se essas alterações violam alguma condição em suas regras. O console AWS Config mostra o status de compatibilidade de suas regras e recursos.

Se um recurso violar uma regra e for sinalizado como não compatível, o AWS Config poderá alertá-lo usando um tópico do [Amazon Simple Notification Service \(Amazon SNS\)](#). Para consumir programaticamente os dados nesses alertas do AWS Config, use uma fila do [Amazon Simple Queue Service \(Amazon SQS\)](#) como o endpoint de notificação para o tópico do Amazon SNS. Por exemplo, convém gravar um código que inicia um fluxo de trabalho quando alguém modifica a configuração do grupo de Auto Scaling do seu ambiente.

Para saber mais sobre como configurar e usar regras, consulte [Avaliando recursos com regras do AWS Config](#) no Guia do desenvolvedor do AWS Config.

## Usar o Elastic Beanstalk com o Amazon DynamoDB

O Amazon DynamoDB é um serviço de banco de dados NoSQL totalmente gerenciado que fornece uma performance rápida e previsível com escalabilidade integrada. Se você for um desenvolvedor, poderá usar o DynamoDB para criar uma tabela de banco de dados para armazenar e recuperar qualquer quantidade de dados, bem como atender a todos os níveis de tráfego de solicitação. O DynamoDB distribui automaticamente dados e tráfego para a tabela entre um número suficiente de servidores para controlar a capacidade de solicitação especificada pelo cliente e a quantidade de dados armazenados, enquanto mantém uma performance consistente e rápida. Todos os itens de dados são armazenados em discos de estado sólido (SSDs) e automaticamente replicados através de várias Zonas de disponibilidade em uma Região da AWS para fornecer maior durabilidade de dados e disponibilidade de integração.

Se você usa [tarefas periódicas](#) em um ambiente de operador, o Elastic Beanstalk cria uma tabela do DynamoDB e a usa para executar a eleição de líder e armazenar informações sobre a tarefa. Cada instância no ambiente tenta gravar na tabela em intervalos de poucos segundos para se tornar líder e realizar a tarefa quando estiver programada.

É possível usar [arquivos de configuração](#) para criar uma tabela do DynamoDB para sua aplicação. Consulte [eb-node-express-sample](#) no GitHub para obter uma aplicação Node.js demonstrativa que cria uma tabela com um arquivo de configuração e se conecta a ele com o AWS SDK for JavaScript in Node.js. Para obter uma demonstração usando o DynamoDB com PHP, consulte [Exemplo: DynamoDB, CloudWatch e SNS](#). Para obter um exemplo que usa o AWS SDK for Java, consulte [Gerenciar estado de sessão Tomcat com o DynamoDB](#) na documentação do AWS SDK for Java.

Ao criar uma tabela do DynamoDB usando os arquivos de configuração, a tabela não se vincula ao ciclo de vida de seu ambiente e não é excluída quando seu ambiente é encerrado. Para que as informações pessoais não sejam retidas desnecessariamente, exclua todos os registros que você não precisa mais ou exclua a tabela.

Para obter mais informações sobre o DynamoDB, consulte o [Guia do desenvolvedor do DynamoDB](#).

## Usar o Elastic Beanstalk com o Amazon ElastiCache

O Amazon ElastiCache é um serviço da Web que permite configurar, gerenciar e dimensionar ambientes de cache de memória distribuídos na nuvem. Ele fornece um cache de memória

econômico, de alta performance e escalável, ao mesmo tempo que remove a complexidade associada à implantação e ao gerenciamento de um ambiente de cache distribuído. O ElastiCache apresenta protocolo em conformidade com o Redis e o Memcached. Portanto, os códigos, aplicações e ferramentas mais populares que você usa hoje com os ambientes existentes do Redis e do Memcached funcionarão perfeitamente com o serviço. Para obter mais informações sobre o ElastiCache, acesse a página do produto [Amazon ElastiCache](#).

Como usar o Elastic Beanstalk com o Amazon ElastiCache

1. Crie um cluster do ElastiCache.
  - Para obter instruções sobre como criar um cluster do ElastiCache com o Redis, acesse [Conceitos básicos do Amazon ElastiCache for Redis](#) no Guia do usuário do ElastiCache for Redis.
  - Para obter instruções sobre como criar um cluster do ElastiCache com o Memcached, acesse [Conceitos básicos do Amazon ElastiCache for Memcached](#) no Guia do usuário do ElastiCache for Memcached.
2. Configure o grupo de segurança do ElastiCache para permitir o acesso do grupo de segurança do Amazon EC2 usado pela aplicação do Elastic Beanstalk. Para obter instruções sobre como encontrar o nome do grupo de segurança do EC2 usando o Console de Gerenciamento da AWS, consulte na página do documento [Grupos de segurança](#) Instâncias do EC2.
  - Para obter mais informações sobre o Redis, acesse [Autorizar o acesso](#) no Guia do usuário do ElastiCache para Redis.
  - Para obter mais informações sobre o Memcached, acesse [Autorizar o acesso](#) no Guia do usuário do ElastiCache para Memcached.

Arquivos de configuração podem ser usados para personalizar o ambiente do ElastiCache. Para ver arquivos de configuração demonstrativos que integram o ElastiCache ao Elastic Beanstalk, consulte [Exemplo: ElastiCache](#).

## Usar o Elastic Beanstalk com o Amazon Elastic File System

Com o Amazon Elastic File System (Amazon EFS), você pode criar sistemas de arquivos de rede que podem ser montados por instâncias em várias zonas de disponibilidade. Um sistema de arquivos do Amazon EFS é um recurso da AWS que usa grupos de segurança para controlar o acesso pela rede que está na VPC padrão ou personalizada.

Em um ambiente do Elastic Beanstalk, você pode usar o Amazon EFS para criar um diretório compartilhado que armazena arquivos para a sua aplicação que os usuários carregam e modificam. Sua aplicação pode tratar um volume montado do Amazon EFS como armazenamento local. Assim, você não precisa alterar o código da aplicação para aumentar a escala vertical para várias instâncias.

Para obter mais informações sobre o Amazon EFS, consulte o [Guia do usuário do Amazon Elastic File System](#).

#### Note

O Elastic Beanstalk cria um usuário webapp que você pode configurar como proprietário de diretórios de aplicações em instâncias do Amazon EC2. Para obter mais informações, consulte [Armazenamento persistente](#) no tópico Considerações sobre design deste guia.

## Seções

- [Arquivos de configuração](#)
- [Sistemas de arquivos criptografados](#)
- [Aplicativos de exemplo](#)
- [Limpeza dos sistemas de arquivos](#)

## Arquivos de configuração

O Elastic Beanstalk fornece [arquivos de configuração](#) que pode ser usado para criar e montar sistemas de arquivos do Amazon EFS. Crie um volume do Amazon EFS como parte de seu ambiente ou monte um volume do Amazon EFS que você criou independentemente do Elastic Beanstalk.

- [storage-efs-createfilesystem.config](#): usa a chave `Resources` para criar um novo sistema de arquivos e pontos de montagem no Amazon EFS. Todas as instâncias do ambiente podem se conectar ao mesmo sistema de arquivos para armazenamento escalável compartilhado. Use `storage-efs-mountfilesystem.config` para montar o sistema de arquivos em cada instância.

### Recursos internos

Todos os recursos que você cria com os arquivos de configuração são vinculados ao ciclo de vida do ambiente. Se você terminar o ambiente ou remover o arquivo de configuração, esses recursos serão perdidos.

- [storage-efs-mountfilesystem.config](#): montar um sistema de arquivos do Amazon EFS em um caminho local nas instâncias do ambiente. Você pode criar o volume como parte do ambiente com `storage-efs-createfilesystem.config`. Ou você pode montá-lo no ambiente usando o console do Amazon EFS, a AWS CLI ou o AWS SDK.

Para usar os arquivos de configuração, comece criando o sistema de arquivos do Amazon EFS com `storage-efs-createfilesystem.config`. Siga as instruções do arquivo de configuração e o adicione ao diretório [.ebextensions](#) em seu código-fonte para criar o sistema de arquivos na VPC.

Implante o código-fonte atualizado no ambiente do Elastic Beanstalk. Isso é para confirmar se o sistema de arquivos foi criado com êxito. Em seguida, adicione o `storage-efs-mountfilesystem.config` para montar o sistema de arquivos nas instâncias do ambiente. Faça isso em duas implantações separadas para garantir que, se a operação de montagem falhar, o sistema de arquivos permaneça intacto. Se você fizer as duas coisas na mesma implantação, um problema em uma das etapas fará com que o sistema de arquivos seja encerrado quando a implantação falhar.

## Sistemas de arquivos criptografados

O Amazon EFS é compatível com sistemas de arquivos criptografados. O arquivo de configuração [storage-efs-createfilesystem.config](#) discutido neste tópico define duas opções personalizadas. Você pode usar essas opções para criar um sistema de arquivos criptografado do Amazon EFS. Para obter mais informações, consulte as instruções no arquivo de configuração.

## Aplicativos de exemplo

O Elastic Beanstalk também fornece aplicações demonstrativas que usam o Amazon EFS para armazenamento compartilhado. Os dois projetos têm arquivos de configuração que você pode usar com um instalador padrão do WordPress ou Drupal para executar um blog ou outro sistema de gerenciamento de conteúdo em um ambiente de balanceamento de carga. Quando um usuário carrega fotos ou outras mídias, o arquivo é armazenado em um sistema de arquivos do Amazon



EFS. Isso evita ter que usar a alternativa, que é usar um plugin para armazenar arquivos carregados no Amazon S3.

- [WordPress com carga balanceada](#): arquivos de configuração para instalar com segurança o WordPress e executá-lo em um ambiente do Elastic Beanstalk com carga balanceada.
- [Drupal com carga balanceada](#): arquivos de configuração e instruções para instalar com segurança o Drupal e executá-lo em um ambiente do Elastic Beanstalk com carga balanceada.

## Limpeza dos sistemas de arquivos

Se você criou um sistema de arquivos do Amazon EFS usando um arquivo de configuração como parte do seu ambiente do Elastic Beanstalk, o Elastic Beanstalk removerá o sistema de arquivos quando o ambiente for terminado. Para minimizar os custos de armazenamento de uma aplicação em execução, exclua rotineiramente os objetos dos quais a aplicação não precisa. Ou certifique-se de que o código da aplicação mantenha o ciclo de vida do arquivo corretamente.

### Important

Se você criou um sistema de arquivos do Amazon EFS fora de um ambiente do Elastic Beanstalk e montou-o nas instâncias do ambiente, lembre-se de que o Elastic Beanstalk não removerá o sistema de arquivos quando o ambiente for terminado. Para garantir que suas informações pessoais não sejam retidas desnecessariamente e evitar custos de armazenamento, exclua os arquivos que sua aplicação armazenou se não precisar mais deles. Como alternativa, você pode remover todo o sistema de arquivos.

## Usar o Elastic Beanstalk com o AWS Identity and Access Management

O AWS Identity and Access Management (IAM) ajuda você a controlar com segurança o acesso aos seus recursos da AWS. Esta seção inclui materiais de referência para trabalhar com políticas do IAM, perfis de instância e funções de serviço.

Para obter uma visão geral de permissões, consulte [Funções de serviço, perfis de instância e políticas de usuário](#). Na maioria dos ambientes, a função de serviço e o perfil de instância cuja criação é solicitada pelo console do Elastic Beanstalk ao iniciar seu primeiro ambiente têm todas as permissões necessárias. Da mesma maneira, as [políticas gerenciadas](#) fornecidas pelo Elastic

Beanstalk para acesso total e acesso somente leitura contêm todas as permissões de usuário necessárias ao uso diário.

O [Guia do usuário do IAM](#) oferece cobertura detalhada de permissões da AWS.

## Tópicos

- [Gerenciar perfis de instância do Elastic Beanstalk](#)
- [Gerenciar funções de serviço do Elastic Beanstalk](#)
- [Usar funções vinculadas ao serviço para o Elastic Beanstalk](#)
- [Gerenciar políticas de usuário do Elastic Beanstalk](#)
- [Formato de nome de recurso da Amazon para o Elastic Beanstalk](#)
- [Recursos e condições para ações do Elastic Beanstalk](#)
- [Usar tags para controlar o acesso aos recursos do Elastic Beanstalk](#)
- [Exemplo de políticas baseadas em políticas gerenciadas](#)
- [Políticas de exemplo baseadas em permissões de recursos](#)

## Gerenciar perfis de instância do Elastic Beanstalk

Um perfil da instância é um contêiner para uma função do AWS Identity and Access Management (IAM) que pode ser usada para passar informações da função para uma instância do Amazon EC2 quando a instância é iniciada.

Se a sua conta da AWS não tiver um perfil de instância do EC2, você deverá criar um usando o serviço IAM. Depois, você poderá atribuir o perfil de instância do EC2 aos novos ambientes que criar. O assistente Criar ambiente fornece informações que guiam você pelo serviço IAM para que possa criar um perfil de instância do EC2 com as permissões necessárias. Depois de criar o perfil de instância, você pode retornar ao console para selecioná-lo como o perfil de instância do EC2 e continuar as etapas para criar seu ambiente.

### Note

Anteriormente, o Elastic Beanstalk criava um perfil de instância padrão do EC2 denominado `aws-elasticbeanstalk-ec2-role` a primeira vez que uma conta da AWS criava um ambiente. Esse perfil de instância incluía as políticas gerenciadas padrão. Se sua conta já tiver esse perfil de instância, ele permanecerá disponível para você atribuí-lo aos seus ambientes.

Porém, as diretrizes de segurança recentes da AWS não permitem que um serviço da AWS crie automaticamente perfis com políticas de confiança para outros serviços da AWS, neste caso o EC2. Por causa dessas diretrizes de segurança, o Elastic Beanstalk não cria mais um perfil de instância padrão `aws-elasticbeanstalk-ec2-role`.

## Políticas gerenciadas

O Elastic Beanstalk fornece várias políticas gerenciadas para permitir que seu ambiente atenda a diferentes casos de uso. Para atender aos casos de uso padrão de um ambiente, essas políticas devem ser anexadas ao perfil de instância do EC2.

- `AWSElasticBeanstalkWebTier`: concede permissões para a aplicação carregar os logs no Amazon S3 e informações de depuração no AWS X-Ray. Para visualizar o conteúdo da política gerenciada, consulte a página [AWSElasticBeanstalkWebTier](#) no Guia de referência de políticas gerenciadas da AWS.
- `AWSElasticBeanstalkWorkerTier`: concede permissões para tarefas de uploads de logs, depuração, publicação de métricas e instância de operador, incluindo gerenciamento de filas, eleição de líder e tarefas periódicas. Para visualizar o conteúdo da política gerenciada, consulte a página [AWSElasticBeanstalkWorkerTier](#) no Guia de referência de políticas gerenciadas da AWS.
- `AWSElasticBeanstalkMulticontainerDocker`: concede permissões para o Amazon Elastic Container Service coordenar tarefas de cluster em ambientes do Docker. Para visualizar o conteúdo da política gerenciada, consulte a página [AWSElasticBeanstalkMulticontainerDocker](#) no Guia de referência de políticas gerenciadas da AWS.

## Política de relacionamento de confiança para o EC2

Para permitir que as instâncias do EC2 do ambiente assumam o perfil necessário, o perfil de instância especifica o Amazon EC2 como uma entidade confiável na política de relacionamento de confiança, como se segue.

```
{
 "Version": "2008-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
```

```
 "Service": "ec2.amazonaws.com"
 },
 "Action": "sts:AssumeRole"
}
]
```

Para personalizar as permissões, você pode adicionar políticas à função anexada ao perfil da instância padrão ou criar seu próprio perfil de instância com um conjunto restrito de permissões.

## Seções

- [Criar um perfil da instância](#)
- [Verificar as permissões atribuídas ao perfil de instância](#)
- [Atualizar um perfil da instância padrão desatualizado](#)
- [Adicionar permissões ao perfil da instância padrão](#)

## Criar um perfil da instância

Perfil da instância é um wrapper em torno de uma função padrão do IAM que permite a uma instância do EC2 assumir a função. Você pode criar perfis de instância adicionais para personalizar permissões para diferentes aplicações. Ou você pode criar um perfil de instância que não conceda permissões para o nível de operador ou para ambientes do Docker gerenciados pelo ECS, caso não use esses recursos.

Para criar um perfil da instância

1. Abra a página [Roles](#) (Funções) no console do IAM.
2. Selecione Create role (Criar função).
3. Em Tipo de entidade confiável, selecione Serviço da AWS.
4. Em Use case (Caso de uso), selecione EC2.
5. Escolha Next (Próximo).
6. Associe as políticas gerenciadas adequadas disponibilizadas pelo Elastic Beanstalk e quaisquer políticas adicionais que fornecem as permissões necessárias à sua aplicação.
7. Escolha Next (Próximo).
8. Insira um nome para a função.
9. (Opcional) Adicione tags à função.

## 10. Selecione Create role (Criar função).

### Verificar as permissões atribuídas ao perfil de instância

As permissões atribuídas ao seu perfil da instância padrão pode variar de acordo com a data de criação, a última vez em que você iniciou um ambiente e qual cliente você usou. É possível verificar as permissões do perfil da instância padrão no console do IAM.

Para verificar as permissões do perfil da instância padrão

1. Abra a página [Roles](#) (Funções) no console do IAM.
2. Escolha a função atribuída como seu perfil de instância do EC2.
3. Na guia Permissions, (Permissões) revise a lista de políticas associadas à função.
4. Para ver as permissões que uma política concede, selecione a política.

### Atualizar um perfil da instância padrão desatualizado

Se o perfil de instância padrão não tiver as permissões necessárias, você poderá adicionar manualmente as políticas ao perfil atribuído como seu perfil de instância do EC2.

Para adicionar políticas gerenciadas à função anexadas ao perfil de instância padrão

1. Abra a página [Roles](#) (Funções) no console do IAM.
2. Escolha a função atribuída como seu perfil de instância do EC2.
3. Na guia Permissions (Permissões), escolha Attach policies (Anexar políticas).
4. Digite **AWSElasticBeanstalk** para filtrar as políticas.
5. Selecione as seguintes políticas e selecione Attach policy (Associar política):
  - AWSElasticBeanstalkWebTier
  - AWSElasticBeanstalkWorkerTier
  - AWSElasticBeanstalkMulticontainerDocker

### Adicionar permissões ao perfil da instância padrão

Se a sua aplicação acessa APIs ou recursos da AWS aos quais não foram concedidas permissões no perfil da instância padrão, adicione políticas que concedem permissões no console do IAM.

Para adicionar políticas à função anexadas ao perfil de instância padrão

1. Abra a página [Roles \(Funções\)](#) no console do IAM.
2. Escolha a função atribuída como seu perfil de instância do EC2.
3. Na guia Permissions (Permissões), escolha Attach policies (Anexar políticas).
4. Selecione a política gerenciada para os serviços adicionais que o seu aplicativo utiliza. Por exemplo, o AmazonS3FullAccess ou o AmazonDynamoDBFullAccess.
5. Escolha Attach policy (Anexar política).

## Gerenciar funções de serviço do Elastic Beanstalk

Para gerenciar e monitorar o seu ambiente, o AWS Elastic Beanstalk realiza ações em recursos do ambiente em seu nome. O Elastic Beanstalk precisa de certas permissões para realizar essas ações e assume funções de serviço do AWS Identity and Access Management (IAM) para obtê-las.

O Elastic Beanstalk precisa usar credenciais de segurança temporárias sempre que assumir uma função de serviço. Para obter essas credenciais, o Elastic Beanstalk envia uma solicitação para o AWS Security Token Service (AWS STS) em um endpoint específico da região. Para obter mais informações, consulte [Credenciais de segurança temporárias](#) no Manual do usuário do IAM.

### Note

Se o endpoint do AWS STS da região em que o seu ambiente está localizado estiver desativado, o Elastic Beanstalk envia a solicitação em um endpoint alternativo que não pode ser desativado. Este endpoint está associado a uma região diferente. Assim, trata-se de uma solicitação entre regiões. Para obter mais informações, consulte [Ativar e desativar o AWS STS em uma região da AWS](#) no Manual do usuário do IAM.

## Gerenciar funções de serviço usando o console do Elastic Beanstalk e a CLI do EB

Você pode usar o console do Elastic Beanstalk e a CLI do EB para configurar funções de serviço para o seu ambiente com um conjunto suficiente de permissões. Eles criam uma função de serviço padrão e usam políticas gerenciadas nela.

## Políticas gerenciadas de função de serviços

O Elastic Beanstalk fornece uma política gerenciada para o [monitoramento avançado de integridade](#) e outra com permissões adicionais necessárias para [atualizações de plataforma gerenciada](#). O console e a CLI do EB atribuem as políticas à função de serviço padrão criada para você. Essas políticas devem ser usadas somente para esta função de serviço padrão. Elas não devem ser usadas com outros usuários ou funções em suas contas.

### **AWS**ElasticBeanstalkEnhancedHealth

Esta política concede permissões ao Elastic Beanstalk para monitorar a integridade do ambiente e da instância.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticloadbalancing:DescribeInstanceHealth",
 "elasticloadbalancing:DescribeLoadBalancers",
 "elasticloadbalancing:DescribeTargetHealth",
 "ec2:DescribeInstances",
 "ec2:DescribeInstanceStatus",
 "ec2:GetConsoleOutput",
 "ec2:AssociateAddress",
 "ec2:DescribeAddresses",
 "ec2:DescribeSecurityGroups",
 "sqs:GetQueueAttributes",
 "sqs:GetQueueUrl",
 "autoscaling:DescribeAutoScalingGroups",
 "autoscaling:DescribeAutoScalingInstances",
 "autoscaling:DescribeScalingActivities",
 "autoscaling:DescribeNotificationConfigurations",
 "sns:Publish"
],
 "Resource": [
 "*"
]
 }
]
}
```

## AWS Elastic Beanstalk Managed Updates Customer Role Policy

Esta política concede permissões ao Elastic Beanstalk para atualizar ambientes em seu nome a fim de realizar atualizações gerenciadas de plataforma.

Agrupamentos de permissão no nível de serviço

Esta política é agrupada em declarações com base no conjunto de permissões fornecidas.

- *ElasticBeanstalkPermissions*: esse grupo de permissões destina-se a acionar as ações de serviço do Elastic Beanstalk (APIs do Elastic Beanstalk).
- *AllowPassRoleToElasticBeanstalkAndDownstreamServices* - Esse grupo de permissões permite que qualquer função seja transferida para o Elastic Beanstalk e para outros serviços downstream, como o AWS CloudFormation.
- *ReadOnlyPermissions*: esse grupo de permissões destina-se a coletar informações sobre o ambiente em execução.
- *\*OperationPermissions*: os grupos com este padrão de nomenclatura destinam-se a acionar as operações necessárias para executar atualizações na plataforma.
- *\*BroadOperationPermissions*: os grupos com este padrão de nomenclatura destinam-se a acionar as operações necessárias para executar atualizações na plataforma. Eles também incluem permissões amplas para oferecer suporte a ambientes herdados.
- *\*TagResource*— Grupos com esse padrão de nomenclatura são para chamadas que usam as APIs tag-on-create para anexar tags em recursos que estão sendo criados em um ambiente Elastic Beanstalk.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ElasticBeanstalkPermissions",
 "Effect": "Allow",
 "Action": [
 "elasticbeanstalk:*"
],
 "Resource": "*"
 },
 {
 "Sid": "AllowPassRoleToElasticBeanstalkAndDownstreamServices",
 "Effect": "Allow",
```



```
"Action": "iam:PassRole",
"Resource": "arn:aws:iam::*:role/*",
"Condition": {
 "StringEquals": {
 "iam:PassedToService": [
 "elasticbeanstalk.amazonaws.com",
 "ec2.amazonaws.com",
 "ec2.amazonaws.com.cn",
 "autoscaling.amazonaws.com",
 "elasticloadbalancing.amazonaws.com",
 "ecs.amazonaws.com",
 "cloudformation.amazonaws.com"
]
 }
},
{
 "Sid": "ReadOnlyPermissions",
 "Effect": "Allow",
 "Action": [
 "autoscaling:DescribeAccountLimits",
 "autoscaling:DescribeAutoScalingGroups",
 "autoscaling:DescribeAutoScalingInstances",
 "autoscaling:DescribeLaunchConfigurations",
 "autoscaling:DescribeLoadBalancers",
 "autoscaling:DescribeNotificationConfigurations",
 "autoscaling:DescribeScalingActivities",
 "autoscaling:DescribeScheduledActions",
 "ec2:DescribeAccountAttributes",
 "ec2:DescribeAddresses",
 "ec2:DescribeAvailabilityZones",
 "ec2:DescribeImages",
 "ec2:DescribeInstanceAttribute",
 "ec2:DescribeInstances",
 "ec2:DescribeKeyPairs",
 "ec2:DescribeLaunchTemplates",
 "ec2:DescribeLaunchTemplateVersions",
 "ec2:DescribeSecurityGroups",
 "ec2:DescribeSnapshots",
 "ec2:DescribeSpotInstanceRequests",
 "ec2:DescribeSubnets",
 "ec2:DescribeVpcClassicLink",
 "ec2:DescribeVpcs",
 "elasticloadbalancing:DescribeInstanceHealth",
```

```

 "elasticloadbalancing:DescribeLoadBalancers",
 "elasticloadbalancing:DescribeTargetGroups",
 "elasticloadbalancing:DescribeTargetHealth",
 "logs:DescribeLogGroups",
 "rds:DescribeDBEngineVersions",
 "rds:DescribeDBInstances",
 "rds:DescribeOrderableDBInstanceOptions",
 "sns:ListSubscriptionsByTopic"
],
 "Resource": [
 "*"
]
},
{
 "Sid": "EC2BroadOperationPermissions",
 "Effect": "Allow",
 "Action": [
 "ec2:AllocateAddress",
 "ec2:AssociateAddress",
 "ec2:AuthorizeSecurityGroupEgress",
 "ec2:AuthorizeSecurityGroupIngress",
 "ec2:CreateLaunchTemplate",
 "ec2:CreateLaunchTemplateVersion",
 "ec2:CreateSecurityGroup",
 "ec2>DeleteLaunchTemplate",
 "ec2>DeleteLaunchTemplateVersions",
 "ec2>DeleteSecurityGroup",
 "ec2:DisassociateAddress",
 "ec2:ReleaseAddress",
 "ec2:RevokeSecurityGroupEgress",
 "ec2:RevokeSecurityGroupIngress"
],
 "Resource": "*"
},
{
 "Sid": "EC2RunInstancesOperationPermissions",
 "Effect": "Allow",
 "Action": "ec2:RunInstances",
 "Resource": "*",
 "Condition": {
 "ArnLike": {
 "ec2:LaunchTemplate": "arn:aws:ec2:*:*:launch-template/*"
 }
 }
}

```

```

 },
 {
 "Sid": "EC2TerminateInstancesOperationPermissions",
 "Effect": "Allow",
 "Action": [
 "ec2:TerminateInstances"
],
 "Resource": "arn:aws:ec2:*:*:instance/*",
 "Condition": {
 "StringLike": {
 "ec2:ResourceTag/aws:cloudformation:stack-id": [
 "arn:aws:cloudformation:*:*:stack/awseb-e-*",
 "arn:aws:cloudformation:*:*:stack/eb-*"
]
 }
 }
 }
],
 {
 "Sid": "ECSBroadOperationPermissions",
 "Effect": "Allow",
 "Action": [
 "ecs:CreateCluster",
 "ecs:DescribeClusters",
 "ecs:RegisterTaskDefinition"
],
 "Resource": "*"
 },
 {
 "Sid": "ECSDeleteClusterOperationPermissions",
 "Effect": "Allow",
 "Action": "ecs>DeleteCluster",
 "Resource": "arn:aws:ecs:*:*:cluster/awseb-*"
 },
 {
 "Sid": "ASGOperationPermissions",
 "Effect": "Allow",
 "Action": [
 "autoscaling:AttachInstances",
 "autoscaling:CreateAutoScalingGroup",
 "autoscaling:CreateLaunchConfiguration",
 "autoscaling:CreateOrUpdateTags",
 "autoscaling>DeleteLaunchConfiguration",
 "autoscaling>DeleteAutoScalingGroup",
 "autoscaling>DeleteScheduledAction",

```

```

 "autoscaling:DetachInstances",
 "autoscaling>DeletePolicy",
 "autoscaling:PutScalingPolicy",
 "autoscaling:PutScheduledUpdateGroupAction",
 "autoscaling:PutNotificationConfiguration",
 "autoscaling:ResumeProcesses",
 "autoscaling:SetDesiredCapacity",
 "autoscaling:SuspendProcesses",
 "autoscaling:TerminateInstanceInAutoScalingGroup",
 "autoscaling:UpdateAutoScalingGroup"
],
 "Resource": [
 "arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/awseb-e-*",
 "arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/eb-*",
 "arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/awseb-e-*",
 "arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/eb-*"
]
},
{
 "Sid": "CFNOperationPermissions",
 "Effect": "Allow",
 "Action": [
 "cloudformation:*"
],
 "Resource": [
 "arn:aws:cloudformation:*:*:stack/awseb-*",
 "arn:aws:cloudformation:*:*:stack/eb-*"
]
},
{
 "Sid": "ELBOperationPermissions",
 "Effect": "Allow",
 "Action": [
 "elasticloadbalancing:AddTags",
 "elasticloadbalancing:ApplySecurityGroupsToLoadBalancer",
 "elasticloadbalancing:ConfigureHealthCheck",
 "elasticloadbalancing>CreateLoadBalancer",
 "elasticloadbalancing>DeleteLoadBalancer",
 "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
 "elasticloadbalancing:DeregisterTargets",
 "elasticloadbalancing:RegisterInstancesWithLoadBalancer",

```

```

 "elasticloadbalancing:RegisterTargets"
],
 "Resource": [
 "arn:aws:elasticloadbalancing:*:*:targetgroup/awseb-*",
 "arn:aws:elasticloadbalancing:*:*:targetgroup/eb-*",
 "arn:aws:elasticloadbalancing:*:*:loadbalancer/awseb-*",
 "arn:aws:elasticloadbalancing:*:*:loadbalancer/eb-*",
 "arn:aws:elasticloadbalancing:*:*:loadbalancer/*/awseb-*/**",
 "arn:aws:elasticloadbalancing:*:*:loadbalancer/*/eb-*/**"
]
},
{
 "Sid": "CWLogsOperationPermissions",
 "Effect": "Allow",
 "Action": [
 "logs:CreateLogGroup",
 "logs>DeleteLogGroup",
 "logs:PutRetentionPolicy"
],
 "Resource": "arn:aws:logs:*:*:log-group:/aws/elasticbeanstalk/*"
},
{
 "Sid": "S3ObjectOperationPermissions",
 "Effect": "Allow",
 "Action": [
 "s3:DeleteObject",
 "s3:GetObject",
 "s3:GetObjectAcl",
 "s3:GetObjectVersion",
 "s3:GetObjectVersionAcl",
 "s3:PutObject",
 "s3:PutObjectAcl",
 "s3:PutObjectVersionAcl"
],
 "Resource": "arn:aws:s3:::elasticbeanstalk-*/*"
},
{
 "Sid": "S3BucketOperationPermissions",
 "Effect": "Allow",
 "Action": [
 "s3:GetBucketLocation",
 "s3:GetBucketPolicy",
 "s3:ListBucket",
 "s3:PutBucketPolicy"
]
}

```

```

],
 "Resource": "arn:aws:s3:::elasticbeanstalk-*"
 },
 {
 "Sid": "SNSOperationPermissions",
 "Effect": "Allow",
 "Action": [
 "sns:CreateTopic",
 "sns:GetTopicAttributes",
 "sns:SetTopicAttributes",
 "sns:Subscribe"
],
 "Resource": "arn:aws:sns:*:*:ElasticBeanstalkNotifications-*"
 },
 {
 "Sid": "SQSOperationPermissions",
 "Effect": "Allow",
 "Action": [
 "sqs:GetQueueAttributes",
 "sqs:GetQueueUrl"
],
 "Resource": [
 "arn:aws:sqs:*:*:awseb-e-*",
 "arn:aws:sqs:*:*:eb-*"
]
 },
 {
 "Sid": "CWPutMetricAlarmOperationPermissions",
 "Effect": "Allow",
 "Action": [
 "cloudwatch:PutMetricAlarm"
],
 "Resource": [
 "arn:aws:cloudwatch:*:*:alarm:awseb-*",
 "arn:aws:cloudwatch:*:*:alarm:eb-*"
]
 },
 {
 "Sid": "AllowECSTagResource",
 "Effect": "Allow",
 "Action": [
 "ecs:TagResource"
],
 "Resource": "*"
 }

```

```
 "Condition": {
 "StringEquals": {
 "ecs:CreateAction": [
 "CreateCluster",
 "RegisterTaskDefinition"
]
 }
 }
]
}
```

Para visualizar o conteúdo de uma política gerenciada, você também pode usar a página [Políticas](#) no console do IAM.

### Note

No passado, o Elastic Beanstalk oferecia suporte à política gerenciada de função de serviços `AWSElasticBeanstalkService`. Esta política foi substituída por `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`. Talvez você ainda consiga ver e usar a política anterior no console do IAM.

Para visualizar o conteúdo da política gerenciada, consulte a página [AWSElasticBeanstalkService](#) no Guia de referência de políticas gerenciadas da AWS. Contudo, recomendamos que você faça a transição para usar a nova política gerenciada (`AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`). Adicione políticas personalizadas para conceder permissões a recursos personalizados, se tiver alguma.

## Usar o console do Elastic Beanstalk

Ao executar um ambiente no console do Elastic Beanstalk, o console cria uma função de serviço padrão chamada `aws-elasticbeanstalk-service-role` e anexa políticas gerenciadas com permissões padrão a essa função de serviço.

Para permitir que o Elastic Beanstalk assumira a função `aws-elasticbeanstalk-service-role`, a função de serviço especifica o Elastic Beanstalk como uma entidade confiável na política de relacionamento de confiança.

```
{
 "Version": "2012-10-17",
```

```
"Statement": [
 {
 "Sid": "",
 "Effect": "Allow",
 "Principal": {
 "Service": "elasticbeanstalk.amazonaws.com"
 },
 "Action": "sts:AssumeRole",
 "Condition": {
 "StringEquals": {
 "sts:ExternalId": "elasticbeanstalk"
 }
 }
 }
]
}
```

Ao habilitar [atualizações gerenciadas de plataforma](#) para o ambiente, o Elastic Beanstalk assume uma função de serviço separada para executar atualizações gerenciadas. Por padrão, o console do Elastic Beanstalk usa a mesma função de serviço gerada, `aws-elasticbeanstalk-service-role`, para a função de serviço de atualizações gerenciadas. Se você alterar a função de serviço padrão, o console definirá a função de serviço de atualizações gerenciadas para usar a função vinculada ao serviço de atualizações gerenciadas, `AWSServiceRoleForElasticBeanstalkManagedUpdates`. Para obter mais informações sobre funções vinculadas ao serviço, consulte [the section called “Usar funções vinculadas ao serviço”](#).

#### Note

Em função de problemas de permissão, o serviço do Elastic Beanstalk nem sempre consegue criar essa função vinculada ao serviço para você com sucesso. Portanto, o console tenta criá-lo explicitamente. Para garantir que sua conta tenha essa função vinculada ao serviço, crie um ambiente pelo menos uma vez usando o console e configure as atualizações gerenciadas para serem habilitadas antes da criação do ambiente.

## Uso da EB CLI

Se você executar um ambiente usando o comando [the section called “eb create”](#) da Interface de linhas de comando do Elastic Beanstalk (CLI do EB) e não especificar uma função de serviço por meio da opção `--service-role`, o Elastic Beanstalk cria uma função de serviço padrão `aws-`



`elasticbeanstalk-service-role`. Se a função de serviço padrão já existir, o Elastic Beanstalk a usará para o novo ambiente. Nessas situações, o console do Elastic Beanstalk também executa ações semelhantes.

Diferente de como é no console, você não pode especificar uma função de serviço de atualizações gerenciadas usando uma opção de comando da CLI do EB. Se você habilitar atualizações gerenciadas para o seu ambiente, deverá definir a função de serviço de atualizações gerenciadas nas opções de configuração. O exemplo a seguir habilita atualizações gerenciadas e usa a função de serviço padrão como uma função de serviço de atualizações gerenciadas.

Example `.ebextensions/managed-platform-update.config`

```
option_settings:
 aws:elasticbeanstalk:managedactions:
 ManagedActionsEnabled: true
 PreferredStartTime: "Tue:09:00"
 ServiceRoleForManagedUpdates: "aws-elasticbeanstalk-service-role"
 aws:elasticbeanstalk:managedactions:platformupdate:
 UpdateLevel: patch
 InstanceRefreshEnabled: true
```

## Gerenciar funções de serviço usando a API do Elastic Beanstalk

Ao usar a ação `CreateEnvironment` da API do Elastic Beanstalk para criar um ambiente, especifique uma função de serviço usando a opção de configuração `ServiceRole` no namespace [aws:elasticbeanstalk:environment](#). Para obter mais informações sobre como usar o monitoramento avançado de integridade com a API do Elastic Beanstalk, consulte [Usar relatórios de integridade aprimorada com a API do Elastic Beanstalk](#).

Além disso, se você habilitar as [atualizações gerenciadas de plataforma](#) para o ambiente, poderá especificar uma função de serviço de atualizações gerenciadas usando a opção `ServiceRoleForManagedUpdates` do namespace [aws:elasticbeanstalk:managedactions](#).

## Usar funções vinculadas ao serviço

Uma função vinculada ao serviço é um tipo exclusivo de função de serviço predefinida pelo Elastic Beanstalk para incluir todas as permissões que o serviço exige para acionar outros produtos da AWS em seu nome. A função vinculada a serviço é associada à sua conta. O Elastic Beanstalk cria essa função uma única vez e depois a reutiliza ao criar ambientes adicionais. Para obter mais informações

sobre o uso de funções vinculadas a serviços com ambientes do Elastic Beanstalk, consulte [Usar funções vinculadas ao serviço para o Elastic Beanstalk](#).

Se você criar um ambiente usando a API do Elastic Beanstalk e não especificar uma função de serviço, o Elastic Beanstalk cria uma [função vinculada ao serviço de monitoramento](#) para a sua conta, se ainda não houver uma. O Elastic Beanstalk usa essa função para o novo ambiente. Você também pode usar o IAM para criar uma função vinculada ao serviço de monitoramento para a sua conta com antecedência. Depois que a sua conta tiver essa função, você pode usá-la para criar um ambiente usando a API do Elastic Beanstalk, o console do Elastic Beanstalk ou a CLI do EB.

Se você habilitar [atualizações de plataforma gerenciadas](#) para o ambiente e especificar `AWSServiceRoleForElasticBeanstalkManagedUpdates` como o valor da opção `ServiceRoleForManagedUpdates` do namespace [aws:elasticbeanstalk:managedactions](#), o Elastic Beanstalk criará uma [função vinculada ao serviço de atualizações gerenciadas](#) para a conta, se ainda não houver uma. O Elastic Beanstalk usa a função para executar atualizações gerenciadas para o novo ambiente.

#### Note

Quando o Elastic Beanstalk tenta criar funções vinculadas ao serviço de monitoramento e atualizações gerenciadas para sua conta quando um ambiente é criado, é necessário ter a permissão `iam:CreateServiceLinkedRole`. Se você não tiver essa permissão, haverá falhas na criação do ambiente e uma mensagem explicando o problema será exibida. Como alternativa, outro usuário com permissão para criar funções vinculadas ao serviço pode usar o IAM para criá-las com antecedência. Com esse método, você não precisa da permissão `iam:CreateServiceLinkedRole` para criar o seu ambiente.

## Como verificar as permissões de funções de serviço padrão

As permissões concedidas pela sua função de serviço padrão variam com base na sua data de criação, na última vez que você executou um ambiente e em qual cliente você usou. No console do IAM, você pode verificar as permissões concedidas pela função de serviço padrão.

Para verificar as permissões da função de serviço padrão

1. No console do IAM, abra a página [Roles](#) (Funções).
2. Escolha `aws-elasticbeanstalk-service-role`.

3. Na guia Permissions, (Permissões) revise a lista de políticas associadas à função.
4. Para visualizar as permissões que uma política concede, selecione a política.

## Atualizar uma função de serviço padrão desatualizado

Se a função de serviço padrão não tiver as permissões necessárias, você poderá atualizá-la [criando um novo ambiente](#) no console de gerenciamento de ambiente do Elastic Beanstalk.

Como alternativa, você pode adicionar manualmente as políticas gerenciadas à função de serviço padrão.

Para adicionar políticas gerenciadas à função de serviço padrão

1. No console do IAM, abra a página [Roles](#) (Funções).
2. Escolha `aws-elasticbeanstalk-service-role`.
3. Na guia Permissions (Permissões), escolha Attach policies (Anexar políticas).
4. Insira **AWSElasticBeanstalk** para filtrar as políticas.
5. Selecione as seguintes políticas e selecione Attach policy (Associar política):
  - `AWSElasticBeanstalkEnhancedHealth`
  - `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`

## Adicionar permissões à função de serviço padrão

Se a aplicação incluir arquivos de configuração que se referem a recursos da AWS para os quais as permissões não estão incluídas na função de serviço padrão, pode ser que o Elastic Beanstalk precise de permissões adicionais. Tais permissões extras são necessárias para resolver essas referências quando elas processam os arquivos de configuração durante uma atualização gerenciada. Em caso de ausência das permissões, a atualização falha e o Elastic Beanstalk retorna uma mensagem indicando quais permissões ele precisa. Siga estas etapas para adicionar permissões para serviços adicionais à função de serviço padrão no console do IAM.

Para adicionar outras políticas à função de serviço padrão

1. No console do IAM, abra a página [Roles](#) (Funções).
2. Escolha `aws-elasticbeanstalk-service-role`.
3. Na guia Permissions (Permissões), escolha Attach policies (Anexar políticas).

4. Selecione a política gerenciada para os serviços adicionais que o seu aplicativo utiliza. Por exemplo, o `AmazonAPIGatewayAdministrator` ou o `AmazonElasticFileSystemFullAccess`.
5. Escolha Attach policy (Anexar política).

## Criar uma função de serviço

Se você não pode usar a função de serviço padrão, crie uma função de serviço.

Para criar uma função de serviço

1. No console do IAM, abra a página [Roles](#) (Funções).
2. Selecione Create role (Criar função).
3. Em Produto da AWS, escolha AWS Elastic Beanstalk, e selecione o caso de uso.
4. Escolha Next: Permissions (Próximo: permissões).
5. Associe as políticas gerenciadas `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` e `AWSElasticBeanstalkEnhancedHealth` e quaisquer políticas adicionais que fornecem as permissões das quais seu aplicativo precisa.
6. Escolha Next: Tags (Próximo: tags).
7. (Opcional) Adicione tags à função.
8. Escolha Next: Review (Próximo: revisar).
9. Insira um nome para a função.
10. Selecione Create role (Criar função).

Aplique a sua função de serviço personalizada ao criar um ambiente usando o [assistente de criação de ambiente](#) ou com a opção `--service-role` para o comando [eb create](#).

## Usar funções vinculadas ao serviço para o Elastic Beanstalk

AWS Elastic Beanstalk usa funções [vinculadas ao serviço AWS Identity and Access Management](#) (IAM). A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao Elastic Beanstalk. As funções vinculadas ao serviço são predefinidas pelo Elastic Beanstalk e incluem todas as permissões que o serviço exige para chamar outros serviços em seu nome. AWS

O Elastic Beanstalk define alguns tipos de funções vinculadas ao serviço:

- Monitoramento da função vinculada ao serviço: permite que o Elastic Beanstalk monitore a integridade dos ambientes em execução e publique notificações de eventos de integridade.
- Função vinculada ao serviço de manutenção: permite que o Elastic Beanstalk realize atividades regulares de manutenção para seus ambientes em execução.
- Função vinculada ao serviço de atualizações gerenciadas: permite que o Elastic Beanstalk realize atualizações agendadas da plataforma dos ambientes em execução.

## Tópicos

- [A função de função vinculada ao serviço de monitoramento](#)
- [A função vinculada ao serviço de manutenção](#)
- [A função vinculada ao serviço de atualizações gerenciadas](#)

## A função de função vinculada ao serviço de monitoramento

AWS Elastic Beanstalk usa funções [vinculadas ao serviço AWS Identity and Access Management \(IAM\)](#). A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao Elastic Beanstalk. As funções vinculadas ao serviço são predefinidas pelo Elastic Beanstalk e incluem todas as permissões que o serviço exige para chamar outros serviços em seu nome. AWS

Uma função vinculada ao serviço facilita a configuração do Elastic Beanstalk, já que não é preciso adicionar as permissões necessárias manualmente. O Elastic Beanstalk define as permissões de suas funções vinculadas ao serviço e, a menos que seja definido em contrário, somente o Elastic Beanstalk pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus recursos do Elastic Beanstalk porque não é possível remover por engano a permissão para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com funções vinculadas a serviços, consulte [Serviços da AWS compatíveis com o IAM](#) e procure os serviços que contenham Sim na coluna Funções vinculadas ao serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado a esse serviço.

## Permissões de função vinculada ao serviço para o Elastic Beanstalk

O Elastic Beanstalk usa a função vinculada ao serviço `AWSServiceRoleForElasticBeanstalk` chamada — Permite que o Elastic Beanstalk monitore a integridade dos ambientes em execução e publique notificações de eventos de saúde.

A função `AWSServiceRoleForElasticBeanstalk` vinculada ao serviço confia nos seguintes serviços para assumir a função:

- `elasticbeanstalk.amazonaws.com`

A política de permissões da função `AWSServiceRoleForElasticBeanstalk` vinculada ao serviço contém todas as permissões que o Elastic Beanstalk precisa para concluir ações em seu nome:

### AllowCloudformationReadOperationsOnElasticBeanstalkStacks

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "AllowCloudformationReadOperationsOnElasticBeanstalkStacks",
 "Effect": "Allow",
 "Action": [
 "cloudformation:DescribeStackResource",
 "cloudformation:DescribeStackResources",
 "cloudformation:DescribeStacks"
],
 "Resource": [
 "arn:aws:cloudformation:*:*:stack/awseb-*",
 "arn:aws:cloudformation:*:*:stack/eb-*"
]
 },
 {
 "Sid": "AllowOperations",
 "Effect": "Allow",
 "Action": [
 "autoscaling:DescribeAutoScalingGroups",
 "autoscaling:DescribeAutoScalingInstances",
 "autoscaling:DescribeNotificationConfigurations",
 "autoscaling:DescribeScalingActivities",
 "autoscaling:PutNotificationConfiguration",
 "ec2:DescribeInstanceStatus",

```

```

 "ec2:AssociateAddress",
 "ec2:DescribeAddresses",
 "ec2:DescribeInstances",
 "ec2:DescribeSecurityGroups",
 "elasticloadbalancing:DescribeInstanceHealth",
 "elasticloadbalancing:DescribeLoadBalancers",
 "elasticloadbalancing:DescribeTargetHealth",
 "elasticloadbalancing:DescribeTargetGroups",
 "sqs:GetQueueAttributes",
 "sqs:GetQueueUrl",
 "sns:Publish"
],
 "Resource": [
 "*"
]
}
]
}

```

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada a serviço. Para mais informações, consulte [Permissões de perfil vinculado ao serviço](#) no Guia do usuário do IAM.

Como alternativa, você pode usar uma política AWS gerenciada para [fornecer acesso total](#) ao Elastic Beanstalk.

Criar uma função vinculada ao serviço para o Elastic Beanstalk

Não é necessário criar manualmente uma função vinculada a serviço. Quando um ambiente do Elastic Beanstalk é criado usando a API do Elastic Beanstalk e nenhuma função de serviço é especificada, o Elastic Beanstalk cria a função vinculada ao serviço para você.

#### Important

Se você estava usando o serviço Elastic Beanstalk antes de 27 de setembro de 2017, quando ele começou a `AWSServiceRoleForElasticBeanstalk` oferecer suporte à função vinculada ao serviço, e sua conta precisava dele, o Elastic Beanstalk criou a função em sua conta. `AWSServiceRoleForElasticBeanstalk` Para saber mais, consulte [A New Role Appeared in My IAM Account](#).

Quando o Elastic Beanstalk tenta `AWSServiceRoleForElasticBeanstalk` criar a função vinculada ao serviço para sua conta ao criar um ambiente, você deve ter a permissão.

`iam:CreateServiceLinkedRole` Se você não tiver essa permissão, ocorrerá falha na criação do ambiente e você verá uma mensagem explicando o problema.

Como alternativa, outro usuário com permissão para criar funções vinculadas ao serviço pode usar o IAM para pré-criá-las com antecedência. Você poderá criar seu ambiente, mesmo sem ter a permissão `iam:CreateServiceLinkedRole`.

Você (ou outro usuário) pode usar o console do IAM para criar uma função vinculada ao serviço com o caso de uso do Elastic Beanstalk. Na CLI ou na API do IAM, crie uma função vinculada ao serviço com o nome de serviço `elasticbeanstalk.amazonaws.com`. Para obter mais informações, consulte [Criar uma função vinculada ao serviço](#) no Guia do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, você poderá usar esse mesmo processo para recriar o perfil em sua conta. Quando um ambiente do Elastic Beanstalk é criado usando a API do Elastic Beanstalk e nenhuma função de serviço é especificada, o Elastic Beanstalk cria novamente a função vinculada ao serviço para você.

### Editar uma função vinculada ao serviço para o Elastic Beanstalk

O Elastic Beanstalk não permite que você edite a função vinculada ao serviço.

`AWSServiceRoleForElasticBeanstalk` Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte [Editar uma função vinculada a serviço](#) no Guia do usuário do IAM.

### Excluir uma função vinculada a serviço para o Elastic Beanstalk

Se você não precisar mais usar um recurso ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de sua função vinculada ao serviço antes de excluí-la manualmente.

### Limpar um perfil vinculado ao serviço

Para poder usar o IAM para excluir uma função vinculada ao serviço, primeiro é necessário confirmar se todos os ambientes do Elastic Beanstalk estão usando uma função de serviço diferente ou se foram encerrados.



**Note**

Se o serviço Elastic Beanstalk estiver usando a função vinculada ao serviço quando você tentar encerrar os ambientes, o encerramento poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para encerrar um ambiente do Elastic Beanstalk que usa o (console)  
AWSServiceRoleForElasticBeanstalk

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

**Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Consulte [eb terminate](#) para obter detalhes sobre como encerrar um ambiente do Elastic Beanstalk usando a CLI do EB.

Consulte [TerminateEnvironment](#) para obter detalhes sobre como encerrar um ambiente do Elastic Beanstalk usando a API.

Excluir manualmente o perfil vinculado ao serviço

Use o console do IAM, a CLI do IAM ou a API do IAM para excluir a função vinculada ao AWSServiceRoleForElasticBeanstalk serviço. Para mais informações, consulte [Excluir um perfil vinculado ao serviço](#) no Guia do usuário do IAM.

Regiões compatíveis com funções vinculadas ao serviço do Elastic Beanstalk

O Elastic Beanstalk é compatível com funções vinculadas ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte [Endpoints e cotas do AWS Elastic Beanstalk](#).

## A função vinculada ao serviço de manutenção

AWS Elastic Beanstalk usa funções [vinculadas ao serviço AWS Identity and Access Management \(IAM\)](#). A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao Elastic Beanstalk. As funções vinculadas ao serviço são predefinidas pelo Elastic Beanstalk e incluem todas as permissões que o serviço exige para chamar outros serviços em seu nome. AWS

Uma função vinculada ao serviço facilita a configuração do Elastic Beanstalk, já que não é preciso adicionar as permissões necessárias manualmente. O Elastic Beanstalk define as permissões de suas funções vinculadas ao serviço e, a menos que seja definido em contrário, somente o Elastic Beanstalk pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus recursos do Elastic Beanstalk porque não é possível remover por engano a permissão para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com funções vinculadas a serviços, consulte [Serviços da AWS compatíveis com o IAM](#) e procure os serviços que contenham Sim na coluna Funções vinculadas ao serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado a esse serviço.

### Permissões de função vinculada ao serviço para o Elastic Beanstalk

O Elastic Beanstalk usa a função vinculada ao serviço `AWSServiceRoleForElasticBeanstalkMaintenance` chamada — Permite que o Elastic Beanstalk realize atividades regulares de manutenção em seus ambientes em execução.

A função `AWSServiceRoleForElasticBeanstalkMaintenance` vinculada ao serviço confia nos seguintes serviços para assumir a função:

- `maintenance.elasticbeanstalk.amazonaws.com`

A política de permissões da função `AWSServiceRoleForElasticBeanstalkMaintenance` vinculada ao serviço contém todas as permissões que o Elastic Beanstalk precisa para concluir ações em seu nome:

```
{
 "Version": "2012-10-17",
 "Statement":
```

```
{
 "Sid": "AllowCloudformationChangeSetOperationsOnElasticBeanstalkStacks",
 "Effect": "Allow",
 "Action": [
 "cloudformation:CreateChangeSet",
 "cloudformation:DescribeChangeSet",
 "cloudformation:ExecuteChangeSet",
 "cloudformation>DeleteChangeSet",
 "cloudformation:ListChangeSets",
 "cloudformation:DescribeStacks"
],
 "Resource": [
 "arn:aws:cloudformation:*:*:stack/awseb-*",
 "arn:aws:cloudformation:*:*:stack/eb-*"
]
}
```

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada a serviço. Para mais informações, consulte [Permissões de perfil vinculado ao serviço](#) no Guia do usuário do IAM.

Como alternativa, você pode usar uma política AWS gerenciada para [fornecer acesso total](#) ao Elastic Beanstalk.

### Criar uma função vinculada ao serviço para o Elastic Beanstalk

Não é necessário criar manualmente uma função vinculada a serviço. Quando um ambiente do Elastic Beanstalk é criado usando a API do Elastic Beanstalk e nenhum perfil de instância é especificado, o Elastic Beanstalk cria a função vinculada ao serviço para você.

#### Important

Esse perfil vinculado ao serviço pode aparecer em sua conta se você concluiu uma ação em outro serviço que usa os atributos compatíveis com esse perfil. Se você estava usando o serviço Elastic Beanstalk antes de 18 de abril de 2019, quando ele começou a `AWSServiceRoleForElasticBeanstalkMaintenance` oferecer suporte à função vinculada ao serviço, e sua conta precisava dele, o Elastic Beanstalk criou a função em sua conta. `AWSServiceRoleForElasticBeanstalkMaintenance` Para saber mais, consulte [Uma Nova Função Apareceu na minha Conta do IAM](#).

Se excluir essa função vinculada ao serviço e precisar criá-la novamente, você pode usar esse mesmo processo para recriar a função na sua conta. Quando um ambiente do Elastic Beanstalk é criado usando a API do Elastic Beanstalk e nenhum perfil de instância é especificado, o Elastic Beanstalk cria a função novamente vinculada ao serviço para você.

### Editar uma função vinculada ao serviço para o Elastic Beanstalk

O Elastic Beanstalk não permite que você edite a função vinculada ao serviço.

**AWSServiceRoleForElasticBeanstalkMaintenance** Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte [Editar uma função vinculada a serviço](#) no Guia do usuário do IAM.

### Excluir uma função vinculada a serviço para o Elastic Beanstalk

Se você não precisar mais usar um recurso ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de sua função vinculada ao serviço antes de excluí-la manualmente.

### Limpar um perfil vinculado ao serviço

Antes de usar o IAM para excluir uma função vinculada ao serviço, primeiro é necessário encerrar todos os ambientes do Elastic Beanstalk que usam a função.

#### Note

Se o serviço Elastic Beanstalk estiver usando a função vinculada ao serviço quando você tentar encerrar os ambientes, o encerramento poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para encerrar um ambiente do Elastic Beanstalk que usa o (console)

**AWSServiceRoleForElasticBeanstalkMaintenance**

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Consulte [eb terminate](#) para obter detalhes sobre como encerrar um ambiente do Elastic Beanstalk usando a CLI do EB.

Consulte [TerminateEnvironment](#) para obter detalhes sobre como encerrar um ambiente do Elastic Beanstalk usando a API.

### Excluir manualmente o perfil vinculado ao serviço

Use o console do IAM, a CLI do IAM ou a API do IAM para excluir a função vinculada ao AWSServiceRoleForElasticBeanstalkMaintenance serviço. Para mais informações, consulte [Excluir um perfil vinculado ao serviço](#) no Guia do usuário do IAM.

### Regiões compatíveis com funções vinculadas ao serviço do Elastic Beanstalk

O Elastic Beanstalk é compatível com funções vinculadas ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte [Endpoints e cotas do AWS Elastic Beanstalk](#).

### A função vinculada ao serviço de atualizações gerenciadas

AWS Elastic Beanstalk usa funções [vinculadas ao serviço AWS Identity and Access Management](#) (IAM). A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao Elastic Beanstalk. As funções vinculadas ao serviço são predefinidas pelo Elastic Beanstalk e incluem todas as permissões que o serviço exige para chamar outros serviços em seu nome. AWS

Uma função vinculada ao serviço facilita a configuração do Elastic Beanstalk, já que não é preciso adicionar as permissões necessárias manualmente. O Elastic Beanstalk define as permissões de suas funções vinculadas ao serviço e, a menos que seja definido em contrário, somente o Elastic Beanstalk pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus recursos do Elastic Beanstalk porque não é possível remover por engano a permissão para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com funções vinculadas a serviços, consulte [Serviços da AWS compatíveis com o IAM](#) e procure os serviços que contenham Sim na coluna Funções vinculadas ao serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado a esse serviço.

### Permissões de função vinculada ao serviço para o Elastic Beanstalk

O Elastic Beanstalk usa a função vinculada ao serviço `AWSServiceRoleForElasticBeanstalkManagedUpdates` chamada — Permite que o Elastic Beanstalk realize atualizações programadas da plataforma de seus ambientes em execução.

A função `AWSServiceRoleForElasticBeanstalkManagedUpdates` vinculada ao serviço confia nos seguintes serviços para assumir a função:

- `managedupdates.elasticbeanstalk.amazonaws.com`

A política gerenciada `AWSElasticBeanstalkManagedUpdatesServiceRolePolicy` concede `AWSServiceRoleForElasticBeanstalkManagedUpdates` à função vinculada ao serviço todas as permissões que o Elastic Beanstalk precisa para concluir as ações de atualização gerenciada em seu nome. Para ver o conteúdo da política gerenciada, consulte a [AWSElasticBeanstalkManagedUpdatesServiceRolePolicy](#) página no Guia de referência de políticas AWS gerenciadas.

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada a serviço. Para mais informações, consulte [Permissões de perfil vinculado ao serviço](#) no Guia do usuário do IAM.

Como alternativa, você pode usar uma política AWS gerenciada para [fornecer acesso total](#) ao Elastic Beanstalk.

### Criar uma função vinculada ao serviço para o Elastic Beanstalk

Não é necessário criar manualmente uma função vinculada a serviço. Quando você cria um ambiente do Elastic Beanstalk com a sua respectiva API, habilita atualizações gerenciadas e especifica `AWSServiceRoleForElasticBeanstalkManagedUpdates` como o valor para a opção `ServiceRoleForManagedUpdates` do namespace

[aws:elasticbeanstalk:managedactions](#), o Elastic Beanstalk cria a função novamente vinculada ao serviço para você.

Quando o Elastic Beanstalk tenta `AWSServiceRoleForElasticBeanstalkManagedUpdates` criar a função vinculada ao serviço para sua conta ao criar um ambiente, você deve ter a permissão `iam:CreateServiceLinkedRole`. Se você não tiver essa permissão, ocorrerá falha na criação do ambiente e você verá uma mensagem explicando o problema.

Como alternativa, outro usuário com permissão para criar funções vinculadas ao serviço pode usar o IAM para pré-criá-las com antecedência. Você poderá criar seu ambiente, mesmo sem ter a permissão `iam:CreateServiceLinkedRole`.

Você (ou outro usuário) pode usar o console do IAM para criar uma função vinculada ao serviço com o caso de uso Elastic Beanstalk Managed Updates (Atualizações gerenciadas do Elastic Beanstalk). Na CLI ou na API do IAM, crie uma função vinculada ao serviço com o nome de serviço `managedupdates.elasticbeanstalk.amazonaws.com`. Para obter mais informações, consulte [Criar uma função vinculada ao serviço](#) no Guia do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, você poderá usar esse mesmo processo para recriar o perfil em sua conta. Quando você cria um ambiente do Elastic Beanstalk com a sua respectiva API, habilita atualizações gerenciadas e especifica `AWSServiceRoleForElasticBeanstalkManagedUpdates` como o valor para a opção `ServiceRoleForManagedUpdates` do namespace [aws:elasticbeanstalk:managedactions](#), o Elastic Beanstalk cria novamente a função novamente vinculada ao serviço para você.

### Editar uma função vinculada ao serviço para o Elastic Beanstalk

O Elastic Beanstalk não permite que você edite a função vinculada ao serviço. `AWSServiceRoleForElasticBeanstalkManagedUpdates` Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte [Editar uma função vinculada a serviço](#) no Guia do usuário do IAM.


### Excluir uma função vinculada a serviço para o Elastic Beanstalk

Se você não precisar mais usar um recurso ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não tem uma entidade não utilizada que não seja

monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de sua função vinculada ao serviço antes de excluí-la manualmente.

Limpar um perfil vinculado ao serviço


Para poder usar o IAM para excluir uma função vinculada ao serviço, primeiro é necessário confirmar se os ambientes com atualizações gerenciadas habilitadas do Elastic Beanstalk estão usando uma função de serviço diferente ou se foram encerrados.

 Note

Se o serviço Elastic Beanstalk estiver usando a função vinculada ao serviço quando você tentar encerrar os ambientes, o encerramento poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para encerrar um ambiente do Elastic Beanstalk que usa o (console)  
`AWSServiceRoleForElasticBeanstalkManagedUpdates`

1. Abra o console do [Elastic](#) Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions e, em seguida, Terminate Environment.
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Consulte [eb terminate](#) para obter detalhes sobre como encerrar um ambiente do Elastic Beanstalk usando a CLI do EB.

Consulte [TerminateEnvironment](#) para obter detalhes sobre como encerrar um ambiente do Elastic Beanstalk usando a API.



## Excluir manualmente o perfil vinculado ao serviço

Use o console do IAM, a CLI do IAM ou a API do IAM para excluir a função vinculada ao `AWSServiceRoleForElasticBeanstalkManagedUpdates` serviço. Para mais informações, consulte [Excluir um perfil vinculado ao serviço](#) no Guia do usuário do IAM.

## Regiões compatíveis com funções vinculadas ao serviço do Elastic Beanstalk

O Elastic Beanstalk é compatível com funções vinculadas ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte [Endpoints e cotas do AWS Elastic Beanstalk](#).

## Gerenciar políticas de usuário do Elastic Beanstalk

AWS Elastic Beanstalk fornece duas políticas gerenciadas que permitem que você atribua acesso total ou acesso somente de leitura a todos os recursos gerenciados pelo Elastic Beanstalk. Você pode anexar as políticas a usuários ou grupos AWS Identity and Access Management (IAM) ou a funções assumidas por seus usuários.

### Important

As políticas gerenciadas do Elastic Beanstalk não fornecem permissões granulares, elas concedem todas as permissões que são potencialmente necessárias para trabalhar com aplicações Elastic Beanstalk. Nossas políticas gerenciadas também não abrangem permissões para recursos personalizados que você pode adicionar à sua solução e que não são gerenciados pelo Elastic Beanstalk. Para implementar permissões mais granulares, permissões mínimas necessárias ou permissões de recursos personalizadas, use [políticas personalizadas](#).

## Políticas de usuário gerenciadas

- `AdministratorAccess-AWSElasticBeanstalk` — Concede ao usuário permissões administrativas completas para criar, modificar e excluir aplicativos, versões de aplicativos, definições de configuração, ambientes e seus recursos subjacentes do Elastic Beanstalk. Para ver o conteúdo da política gerenciada, consulte a `AWSElasticBeanstalk` página [AdministratorAccess-](#) no Guia de referência da política AWS gerenciada.
- `AWSElasticBeanstalkReadOnly`— Permite que o usuário visualize aplicativos e ambientes, mas não realize operações que os modifiquem. Ele fornece acesso somente de leitura a

todos os recursos do Elastic Beanstalk e a outros recursos AWS que o console do Elastic Beanstalk recupera. Observe que o acesso somente leitura não permite ações como fazer download de logs do Elastic Beanstalk para que você possa lê-los. Isso ocorre porque os logs são preparados no bucket do Amazon S3, em que o Elastic Beanstalk exige permissão para gravação. Consulte o exemplo no fim deste tópico para obter informações sobre como habilitar o acesso aos logs do Elastic Beanstalk. Para ver o conteúdo da política gerenciada, consulte a [AWSElasticBeanstalkReadOnly](#) página no Guia de referência de políticas AWS gerenciadas.

### Note

Anteriormente, o Elastic Beanstalk dava suporte a duas outras políticas de usuários gerenciados, e. `AWSElasticBeanstalkFullAccess` e `AWSElasticBeanstalkReadOnlyAccess`. Planejamos aposentar essas políticas anteriores. Talvez você ainda consiga vê-las e usá-las no console do IAM. No entanto, recomendamos que você faça a transição para o uso das novas políticas de usuário gerenciadas e adicione políticas personalizadas para conceder permissões a recursos personalizados, se você tiver algum.

## Políticas de integração com outros serviços

Também fornecemos políticas mais granulares que permitem integrar seu ambiente a outros serviços, se você preferir usá-los.

- `AWSElasticBeanstalkRoleCWL`— Permite que um ambiente gerencie grupos de CloudWatch logs do Amazon Logs.
- `AWSElasticBeanstalkRoleRDS`— Permite que um ambiente integre uma instância do Amazon RDS.
- `AWSElasticBeanstalkRoleWorkerTier`— Permite que um nível de ambiente de trabalho crie uma tabela do Amazon DynamoDB e uma fila do Amazon SQS.
- `AWSElasticBeanstalkRoleECS`— Permite que um ambiente Docker de vários contêineres gerencie clusters do Amazon ECS.
- `AWSElasticBeanstalkRoleCore`— Permite as operações principais de um ambiente de serviços web.
- `AWSElasticBeanstalkRoleSNS`— Permite que um ambiente habilite a integração de tópicos do Amazon SNS.

Para ver a fonte JSON de uma política gerenciada específica, consulte o [Guia de referência de políticas AWS gerenciadas](#).

## Controlar o acesso com políticas gerenciadas

É possível usar as políticas gerenciadas para conceder acesso total ou somente leitura ao Elastic Beanstalk. O Elastic Beanstalk atualiza essas políticas automaticamente quando outras permissões são necessárias para acessar novos recursos.

Como aplicar uma política gerenciada a grupos ou usuários do IAM

1. Abra a página [Policies \(Políticas\)](#) no console do IAM.
2. Na caixa de pesquisa, digite **AWSElasticBeanstalk** para filtrar as políticas.
3. Na lista de políticas, marque a caixa de seleção ao lado de **AWSElasticBeanstalkReadOnlyAdministratorAccess- AWSElasticBeanstalk**.
4. Selecione Ações da política e escolha Anexar.
5. Selecione um ou mais usuários e grupos aos quais associar a política. Você pode usar o menu Filter (Filtro) e a caixa de pesquisa para filtrar a lista de entidades principais.
6. Escolha Anexar política.

## Criar uma política de usuário personalizada

Você pode criar sua própria política do IAM para permitir ou negar ações específicas da API do Elastic Beanstalk em recursos específicos do Elastic Beanstalk e controlar o acesso a recursos personalizados que não são gerenciados pelo Elastic Beanstalk. Para obter mais informações sobre como anexar uma política a um usuário ou grupo, consulte [Trabalhar com políticas](#) no Guia do usuário do IAM. Para obter detalhes sobre a criação de uma política personalizada, consulte [Criar políticas do IAM](#) no Guia do usuário do IAM.

### Note

Embora seja possível restringir como um usuário interage com as APIs do Elastic Beanstalk, atualmente não existe uma forma eficiente para impedir que os usuários com permissão para criar os recursos subjacentes necessários criem outros recursos no Amazon EC2 e em outros serviços.

Considere essas políticas como uma forma eficiente para distribuir as responsabilidades do Elastic Beanstalk, e não como uma maneira de proteger todos os recursos subjacentes.

Em novembro de 2019, o Elastic Beanstalk lançou suporte para [modelos de execução do Amazon EC2](#). Trata-se de um novo tipo de recurso que o grupo de Auto Scaling do ambiente pode usar para executar instâncias do Amazon EC2 e requer novas permissões. A maioria dos clientes não deve ser afetada, pois os ambientes ainda podem usar o recurso legado e as configurações de execução, se sua política de usuário não tiver as permissões necessárias. No entanto, se você estiver tentando usar um novo recurso que requer modelos de execução do Amazon EC2 e tiver uma política personalizada, a criação ou a atualização do ambiente poderá falhar. Nesse caso, verifique se sua política personalizada tem as seguintes permissões.

Permissões necessárias para modelos de execução do Amazon EC2

- `EC2:CreateLaunchTemplate`
- `EC2:CreateLaunchTemplateVersions`
- `EC2>DeleteLaunchTemplate`
- `EC2>DeleteLaunchTemplateVersions`
- `EC2:DescribeLaunchTemplate`
- `EC2:DescribeLaunchTemplateVersions`

Uma política do IAM contém declarações que descrevem as permissões a serem concedidas. Ao criar uma declaração de política para o Elastic Beanstalk, você precisa entender como usar as seguintes quatro partes:

- Efeito especifica se é para permitir ou negar as ações na declaração.
- Ação especifica as [operações da API](#) que você deseja controlar. Por exemplo, use `elasticbeanstalk:CreateEnvironment` para especificar a operação `CreateEnvironment`. Algumas operações, como criação de ambiente, exigem permissões adicionais para executar essas ações. Para ter mais informações, consulte [Recursos e condições para ações do Elastic Beanstalk](#).

#### Note

Para usar a operação de API [UpdateTagsForResource](#), especifique uma das duas ações virtuais a seguir (ou ambas) em vez do nome de operação da API:

```
elasticbeanstalk:AddTags
```

Controla a permissão para chamar `UpdateTagsForResource` e passar uma lista de tags a serem adicionadas ao parâmetro `TagsToAdd`.

## elasticbeanstalk:RemoveTags

Controla a permissão para chamar UpdateTagsForResource e passar uma lista de chaves de tags a serem removidas do parâmetro TagsToRemove.

- Recurso especifica os recursos aos quais você deseja controlar o acesso. Para especificar recursos do Elastic Beanstalk, liste o [Nome de recurso da Amazon](#) (ARN) de cada recurso.
- (opcional) Condição especifica restrições sobre a permissão concedida na declaração. Para ter mais informações, consulte [Recursos e condições para ações do Elastic Beanstalk](#).

As seções a seguir demonstram alguns casos em que você pode considerar uma política de usuário personalizada.

### Habilitar a criação limitada do ambiente do Elastic Beanstalk

A política no exemplo a seguir permite que um usuário chame a ação CreateEnvironment para criar um ambiente cujo nome começa com **Test** com o aplicativo especificado e a versão do aplicativo.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CreateEnvironmentPerm",
 "Action": [
 "elasticbeanstalk:CreateEnvironment"
],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My First Elastic Beanstalk Application/Test*"
],
 "Condition": {
 "StringEquals": {
 "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My First Elastic Beanstalk Application"],
 "elasticbeanstalk:FromApplicationVersion": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My First Elastic Beanstalk Application/First Release"]
 }
 }
 }
]
}
```

```
 },
 {
 "Sid": "AllNonResourceCalls",
 "Action": [
 "elasticbeanstalk:CheckDNSAvailability",
 "elasticbeanstalk:CreateStorageLocation"
],
 "Effect": "Allow",
 "Resource": [
 "*"
]
 }
]
}
```

A política acima mostra como conceder acesso limitado às operações do Elastic Beanstalk. Para realmente iniciar um ambiente, o usuário também deve ter permissão para criar os AWS recursos que alimentam o ambiente. Por exemplo, a seguinte política concede acesso ao conjunto padrão de recursos para um ambiente de servidor Web:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ec2:*",
 "ecs:*",
 "elasticloadbalancing:*",
 "autoscaling:*",
 "cloudwatch:*",
 "s3:*",
 "sns:*",
 "cloudformation:*",
 "sqs:*"
],
 "Resource": "*"
 }
]
}
```

## Habilitar o acesso aos logs do Elastic Beanstalk armazenados no Amazon S3

A política no exemplo a seguir permite que um usuário extraia logs do Elastic Beanstalk, prepare-os no Amazon S3 e os recupere.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "s3:DeleteObject",
 "s3:GetObjectAcl",
 "s3:PutObjectAcl"
],
 "Effect": "Allow",
 "Resource": "arn:aws:s3:::elasticbeanstalk-*"
 }
]
}
```

### Note

Para restringir essas permissões a somente caminho de logs, use o seguinte formato de recursos.

```
"arn:aws:s3:::elasticbeanstalk-us-east-2-123456789012/resources/environments/
logs/*"
```

## Permitir o gerenciamento de uma aplicação específica do Elastic Beanstalk

A política no exemplo a seguir permite que um usuário gerencie ambientes e outros recursos dentro de uma aplicação específica do Elastic Beanstalk. A política nega ações do Elastic Beanstalk nos recursos de outras aplicações e também nega a criação e exclusão de aplicações do Elastic Beanstalk.

**Note**

A política não nega acesso a nenhum recurso por meio de outros serviços. Ela se revela uma forma eficiente de distribuir responsabilidades para gerenciar aplicações do Elastic Beanstalk entre diferentes usuários, e não para proteger os recursos subjacentes.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": [
 "elasticbeanstalk:CreateApplication",
 "elasticbeanstalk>DeleteApplication"
],
 "Resource": [
 "*"
]
 },
 {
 "Effect": "Deny",
 "Action": [
 "elasticbeanstalk:CreateApplicationVersion",
 "elasticbeanstalk:CreateConfigurationTemplate",
 "elasticbeanstalk:CreateEnvironment",
 "elasticbeanstalk>DeleteApplicationVersion",
 "elasticbeanstalk>DeleteConfigurationTemplate",
 "elasticbeanstalk>DeleteEnvironmentConfiguration",
 "elasticbeanstalk:DescribeApplicationVersions",
 "elasticbeanstalk:DescribeConfigurationOptions",
 "elasticbeanstalk:DescribeConfigurationSettings",
 "elasticbeanstalk:DescribeEnvironmentResources",
 "elasticbeanstalk:DescribeEnvironments",
 "elasticbeanstalk:DescribeEvents",
 "elasticbeanstalk>DeleteEnvironmentConfiguration",
 "elasticbeanstalk:RebuildEnvironment",
 "elasticbeanstalk:RequestEnvironmentInfo",
 "elasticbeanstalk:RestartAppServer",
 "elasticbeanstalk:RetrieveEnvironmentInfo",
 "elasticbeanstalk:SwapEnvironmentCNAMEs",
 "elasticbeanstalk:TerminateEnvironment",
```



```
 "elasticbeanstalk:UpdateApplicationVersion",
 "elasticbeanstalk:UpdateConfigurationTemplate",
 "elasticbeanstalk:UpdateEnvironment",
 "elasticbeanstalk:RetrieveEnvironmentInfo",
 "elasticbeanstalk:ValidateConfigurationSettings"
],
 "Resource": [
 "*"
],
 "Condition": {
 "StringNotEquals": {
 "elasticbeanstalk:InApplication": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/myapplication"
]
 }
 }
}
```

## Formato de nome de recurso da Amazon para o Elastic Beanstalk

Especifique um recurso para uma política do IAM usando o Nome de recurso da Amazon (ARN) desse recurso. Para o Elastic Beanstalk, o ARN tem o formato a seguir.

```
arn:aws:elasticbeanstalk:region:account-id:resource-type/resource-path
```

Onde:

- *region* é a região na qual o recurso reside (por exemplo, **us-west-2**).
- *account-id* é o ID da conta da AWS, sem hifens (por exemplo, **123456789012**)
- *resource-type* identifica o tipo de recurso do Elastic Beanstalk, por exemplo, **environment**. Consulte a tabela a seguir para obter uma lista de todos os tipos de recursos do Elastic Beanstalk.
- *resource-path* é a parte que identifica o recurso específico. Um recurso do Elastic Beanstalk tem um caminho que identifica exclusivamente esse recurso. Veja na tabela abaixo o formato do caminho do recurso para cada tipo de recurso. Por exemplo, um ambiente está sempre associado a um aplicativo. O caminho do recurso para o ambiente **myEnvironment** no aplicativo **myApp** seria assim:

myApp/myEnvironment

O Elastic Beanstalk tem vários tipos de recursos que podem ser especificados em uma política. A tabela a seguir mostra o formato de ARN para cada tipo de recurso e um exemplo.

Tipo de recurso	Formato do ARN
application	<p>arn:aws:elasticbeanstalk: <i>region:account-id</i> :application/<i> application-name</i></p> <p>Exemplo: <b>arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App</b></p>
applicationversion	<p>arn:aws:elasticbeanstalk: <i>region:account-id</i> :applicationversion/<i> application-name /version-label</i></p> <p>Exemplo: <b>arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version</b></p>
configurationtemplate	<p>arn:aws:elasticbeanstalk: <i>region:account-id</i> :configurationtemplate/<i> application-name /template-name</i></p> <p>Exemplo: <b>arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template</b></p>
environment	<p>arn:aws:elasticbeanstalk: <i>region:account-id</i> :environment/<i> application-name /environment-name</i></p> <p>Exemplo: <b>arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/MyEnvironment</b></p>
platform	<p>arn:aws:elasticbeanstalk: <i>region:account-id</i> :platform/<i> platform-name /platform-version</i></p> <p>Exemplo: <b>arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/MyPlatform/1.0</b></p>

Tipo de recurso	Formato do ARN
solutions tack	arn:aws:elasticbeanstalk: <i>region</i> ::solutionstack/ <i>solutions tack-name</i>  Exemplo: <b>arn:aws:elasticbeanstalk:us-east-2::solutions tack/32bit Amazon Linux running Tomcat 7</b>

Um ambiente, versão do aplicativo e modelo de configuração estão sempre contidos em um aplicativo específico. Você verá que esses recursos têm um nome de aplicativo no caminho do recurso, de forma que eles sejam identificados exclusivamente pelo nome do recurso e pelo aplicativo que os contêm. Embora as pilhas de soluções sejam usadas por modelos de configuração e ambientes, elas não são específicas de uma aplicação ou conta da AWS e não têm a aplicação nem conta da AWS nos ARNs.

## Recursos e condições para ações do Elastic Beanstalk

Esta seção descreve os recursos e as condições que podem ser usadas em declarações de política a fim de conceder permissões para que ações específicas sejam executadas em recursos específicos do Elastic Beanstalk.

As condições permitem que você especifique permissões para recursos necessários para a conclusão da ação. Por exemplo, quando você chama a ação `CreateEnvironment`, também deve especificar a versão do aplicativo para implantar, bem como o aplicativo que contém o nome do aplicativo. Ao definir permissões para a ação `CreateEnvironment`, você especifica o aplicativo e a versão do aplicativo nos quais deseja que a ação tenha efeito usando as condições `InApplication` e `FromApplicationVersion`.

Além disso, você pode especificar a configuração do ambiente com um stack de solução (`FromSolutionStack`) ou com um modelo de configuração (`FromConfigurationTemplate`). A declaração de política a seguir permite que a ação `CreateEnvironment` crie um ambiente com o nome **myenv** (especificado por `Resource`) no aplicativo **My App** (especificado pela condição `InApplication`) usando a versão do aplicativo **My Version** (`FromApplicationVersion`) com uma configuração **32bit Amazon Linux running Tomcat 7** (`FromSolutionStack`):

```
{
```

```
"Version": "2012-10-17",
"Statement": [
 {
 "Action": [
 "elasticbeanstalk:CreateEnvironment"
],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"
],
 "Condition": {
 "StringEquals": {
 "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],
 "elasticbeanstalk:FromApplicationVersion": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:applicationversion/My App/My Version"],
 "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-
east-2::solutionstack/32bit Amazon Linux running Tomcat 7"]
 }
 }
 }
]
```

### Note

A maioria das chaves de condição mencionadas neste tópico é específica do Elastic Beanstalk, e seus nomes contêm o prefixo `elasticbeanstalk:`. Em resumo, nós omitimos este prefixo dos nomes de chave de condição quando nós os mencionamos nas seções a seguir. Por exemplo, mencionamos `InApplication` ao invés de seu nome completo `elasticbeanstalk:InApplication`.

Por outro lado, nós mencionamos algumas chaves de condição usadas em todos os serviços da AWS e incluímos seu prefixo `aws:` para destacar a exceção.

Os exemplos de política sempre mostram nomes de chave de condição completos, incluindo o prefixo.

## Seções

- [Informações de políticas para ações do Elastic Beanstalk](#)
- [Chaves de condição para ações do Elastic Beanstalk](#)

## Informações de políticas para ações do Elastic Beanstalk

A tabela a seguir lista todas as ações do Elastic Beanstalk, o recurso afetado por cada ação e as informações contextuais adicionais que podem ser fornecidas usando condições.

Informações sobre política para ações do Elastic Beanstalk, incluindo recursos, condições, exemplos e dependências

Recurso	Condições	Exemplo de declaração
---------	-----------	-----------------------

### Ação: [AbortEnvironmentUpdate](#)

application environment	aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que um usuário interrompa as operações de atualização de ambiente em ambientes de um aplicativo chamado My App.</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:AbortEnvironmentUpdate"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"       ]     }   ] }</pre>
----------------------------	---------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Ação: [CheckDNSAvailability](#)

"*"	N/D	<pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:CheckDNSAvailability"       ]     }   ] }</pre>
-----	-----	------------------------------------------------------------------------------------------------------------------------------------------------------

Recurso	Condições	Exemplo de declaração
		<pre> ], "Effect": "Allow", "Resource": "*" } ] } </pre>

Ação: [ComposeEnvironments](#)

application	<p>aws:ResourceTag/ <i>key-name</i> (opcional)</p> <p>aws:TagKeys (opcional)</p>	<p>A política a seguir permite que um usuário componha ambientes que pertencem a um aplicativo chamado My App.</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:ComposeEnvironments"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App"       ]     }   ] } </pre>
-------------	----------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Ação: [CreateApplication](#)

Recurso	Condições	Exemplo de declaração
application	<p>aws:RequestTag/ <i>key-name</i> (opcional)</p> <p>aws:TagKeys (opcional)</p>	<p>Este exemplo permite que a ação <code>CreateApplication</code> crie aplicativos cujos nomes começam com <b>DivA</b>:</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:CreateApplication"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/DivA*"       ]     }   ] }</pre>

Ação: [CreateApplicationVersion](#)

Recurso	Condições	Exemplo de declaração
applicationversion	<p>InApplication</p> <p>aws:RequestTag/ <i>key-name</i> (opcional)</p> <p>aws:TagKeys (opcional)</p>	<p>Este exemplo permite que a ação <code>CreateApplicationVersion</code> crie versões do aplicativo com qualquer nome (*) no aplicativo <b>My App</b>:</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:CreateApplicationVersion"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/MyApp/*"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication":             ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"]         }       }     }   ] }</pre>

Ação: [CreateConfigurationTemplate](#)



Recurso	Condições	Exemplo de declaração
configurationtemplate	InApplication FromApplication FromApplicationVersion FromConfigurationTemplate FromEnvironment FromSolutionStack aws:RequestTag/ <i>key-name</i> (opcional) aws:TagKeys (opcional)	<p>A política a seguir permite que a ação <code>CreateConfigurationTemplate</code> crie modelos de configuração cujos nomes comecem com <b>My Template</b> (<code>My Template*</code>) no aplicativo <b>My App</b>:</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:CreateConfigurationTemplate"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template*"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication": [             "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App",             "elasticbeanstalk:FromSolutionStack": [               "arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7"             ]           ]         }       }     }   ] } </pre>

Ação: [CreateEnvironment](#)

Recurso	Condições	Exemplo de declaração
environment	InApplication  FromApplicationVersion  FromConfigurationTemplate  FromSolutionStack  aws:RequestTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação <code>CreateEnvironment</code> crie um ambiente cujo nome é <b>myenv</b> no aplicativo <b>My App</b> usando o stack de solução <b>32bit Amazon Linux running Tomcat 7</b>:</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:CreateEnvironment"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication": [             "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App",             "elasticbeanstalk:FromApplicationVersion": [               "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version",               "elasticbeanstalk:FromSolutionStack": [                 "arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7"               ]             ]           ]         }       }     }   ] } </pre>

Ação: [CreatePlatformVersion](#)

Recurso	Condições	Exemplo de declaração
platform	aws:RequestTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>Esse exemplo permite que a ação CreatePlatformVersion crie versões de plataforma direcionando a região us-east-2, cujos nomes começam com <b>us-east-2_</b>:</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:CreatePlatformVersion"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*"       ]     }   ] }</pre>

Ação: [CreateStorageLocation](#)

"*"	N/D	<pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:CreateStorageLocation"       ],       "Effect": "Allow",       "Resource": "*"     }   ] }</pre>
-----	-----	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Recurso	Condições	Exemplo de declaração
Ação: <a href="#">DeleteApplication</a>		
application	aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação DeleteApplication exclua o aplicativo <b>My App</b>:</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:DeleteApplication"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"       ]     }   ] }</pre>

Ação: [DeleteApplicationVersion](#)

Recurso	Condições	Exemplo de declaração
applicationversion	<p>InApplication</p> <p>aws:ResourceTag/ <i>key-name</i> (opcional)</p> <p>aws:TagKeys (opcional)</p>	<p>A política a seguir permite que a ação DeleteApplicationVersion exclua uma versão do aplicativo cujo nome é <b>My Version</b> no aplicativo <b>My App</b>:</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:DeleteApplicationVersion"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/MyApp/My Version"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication":             ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"]         }       }     }   ] } </pre>

Ação: [DeleteConfigurationTemplate](#)

Recurso	Condições	Exemplo de declaração
configurationtemplate	InApplication (opcional)  aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação DeleteConfigurationTemplate exclua um modelo de configuração cujo nome é <b>My Template</b> no aplicativo <b>My App</b>. A especificação do nome do aplicativo como uma condição é opcional.</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:DeleteConfigurationTemplate"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template"       ]     }   ] }</pre>

Ação: [DeleteEnvironmentConfiguration](#)

Recurso	Condições	Exemplo de declaração
environment	InApplication (opcional)	<p>A política a seguir permite que a ação <code>DeleteEnvironmentConfiguration</code> exclua um modelo de configuração de rascunho do ambiente <b>myenv</b> no aplicativo <b>My App</b>. A especificação do nome do aplicativo como uma condição é opcional.</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:DeleteEnvironmentConfiguration"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"       ]     }   ] }</pre>

Ação: [DeletePlatformVersion](#)

Recurso	Condições	Exemplo de declaração
platform	<code>aws:ResourceTag/ <i>key-name</i></code> (opcional)  <code>aws:TagKeys</code> (opcional)	<p>A política a seguir permite que a ação <code>DeletePlatformVersion</code> exclua versões de plataforma direcionando a região <code>us-east-2</code>, cujos nomes começam com <b>us-east-2_</b>:</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:DeletePlatformVersion"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*"       ]     }   ] }</pre>

Ação: [DescribeApplications](#)



Recurso	Condições	Exemplo de declaração
application	aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação DescribeApplications descreva o aplicativo My App.</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:DescribeApplications"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"       ]     }   ] }</pre>

Ação: [DescribeApplicationVersions](#)

Recurso	Condições	Exemplo de declaração
applicationversion	InApplication (opcional)  aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação DescribeApplicationVersions descreva a versão do aplicativo <b>My Version</b> no aplicativo <b>My App</b>. A especificação do nome do aplicativo como uma condição é opcional.</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:DescribeApplicationVersions"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/MyApp/My Version"       ]     }   ] } </pre>

Ação: [DescribeConfigurationOptions](#)

Recurso	Condições	Exemplo de declaração
environment configurationtemplate solutions tack	InApplication (opcional)  aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação DescribeConfigurationOptions descreva as opções de configuração do ambiente <b>myenv</b> no aplicativo <b>My App</b>. A especificação do nome do aplicativo como uma condição é opcional.</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": "elasticbeanstalk:DescribeConfigurationOptions",       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"       ]     }   ] } </pre>

Ação: [DescribeConfigurationSettings](#)

Recurso	Condições	Exemplo de declaração
environment configurationtemplate	InApplication (opcional)  aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação DescribeConfigurationSettings descreva as definições de configuração do ambiente <b>myenv</b> no aplicativo <b>My App</b>. A especificação do nome do aplicativo como uma condição é opcional.</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": "elasticbeanstalk:DescribeConfigurationSettings",       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"       ]     }   ] } </pre>

Ação: [DescribeEnvironmentHealth](#)

Recurso	Condições	Exemplo de declaração
environment	<p>aws:ResourceTag/ <i>key-name</i> (opcional)</p> <p>aws:TagKeys (opcional)</p>	<p>A política a seguir permite o uso de DescribeEnvironmentHealth para recuperar informações de integridade de um ambiente chamado <b>myenv</b>.</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": "elasticbeanstalk:DescribeEnvironmentHealth",       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"       ]     }   ] }</pre>

Ação: [DescribeEnvironmentResources](#)

Recurso	Condições	Exemplo de declaração
environment	InApplica tion (opcional)  aws:Resou rceTag/ <i>key-</i> <i>name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação DescribeEnvironmentResources retorne a lista dos recursos da AWS para o ambiente <b>myenv</b> na aplicação <b>My App</b>. A especificação do nome do aplicativo como uma condição é opcional.</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": "elasticbeanstalk:DescribeE nvironmentResources",       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us- east-2:123456789012:environment/My App/ myenv"       ]     }   ] }</pre>

Ação: [DescribeEnvironments](#)

Recurso	Condições	Exemplo de declaração
environment	InApplica tion (opcional)  aws:Resou rceTag/ <i>key-</i> <i>name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação <code>DescribeEnvironments</code> descreva os ambientes <b>myenv</b> e <b>myotherenv</b> no aplicativo <b>My App</b>. A especificação do nome do aplicativo como uma condição é opcional.</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": "elasticbeanstalk:DescribeEnvironments",       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv",         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App2/myotherenv"       ]     }   ] }</pre>

Ação: [DescribeEvents](#)

Recurso	Condições	Exemplo de declaração
application applicationversion configurationtemplate environment	InApplication  aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação DescribeEvents liste descrições de evento para o ambiente <b>myenv</b> e a versão do aplicativo <b>My Version</b> no aplicativo <b>My App</b>.</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": "elasticbeanstalk:DescribeEvents",       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv",         "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App/My Version"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication":             ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"]         }       }     }   ] } </pre>

Ação: [DescribeInstancesHealth](#)



Recurso	Condições	Exemplo de declaração
environment	N/D	<p>A política a seguir permite o uso de DescribeInstancesHealth para recuperar informações de integridade de instâncias em um ambiente chamado <b>myenv</b>.</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": "elasticbeanstalk:DescribeInstancesHealth",       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"       ]     }   ] }</pre>

Ação: [DescribePlatformVersion](#)

Recurso	Condições	Exemplo de declaração
platform	aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação DescribePlatformVersion descreva as versões de plataforma direcionando a região us-east-2 , cujos nomes começam com <b>us-east-2_</b> :</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:DescribePlatformVersion"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*"       ]     }   ] }</pre>

Ação: [ListAvailableSolutionStacks](#)

Recurso	Condições	Exemplo de declaração
solutions tack	N/D	<p>A política a seguir permite que a ação <code>ListAvailableSolutionStacks</code> retorne somente a pilha de solução <b>32bit Amazon Linux running Tomcat 7</b>.</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:ListAvailableSolutionStacks"       ],       "Effect": "Allow",       "Resource": "arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7"     }   ] }</pre>

Ação: [ListPlatformVersions](#)

Recurso	Condições	Exemplo de declaração
platform	<code>aws:RequestTag/ <i>key-name</i></code> (opcional)  <code>aws:TagKeys</code> (opcional)	<p>Esse exemplo permite que a ação <code>CreatePlatformVersion</code> crie versões de plataforma direcionando a região <code>us-east-2</code>, cujos nomes começam com <b>us-east-2_</b>:</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:ListPlatformVersions"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*"       ]     }   ] }</pre>

Ação: [ListTagsForResource](#)

Recurso	Condições	Exemplo de declaração
application applicationversion configurationtemplate environment platform	aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação ListTagsForResource liste as tags de recursos existentes somente se elas tiverem uma tag chamada stage com o valor test:</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:ListTagsForResource"       ],       "Effect": "Allow",       "Resource": "*",       "Condition": {         "StringEquals": {           "aws:ResourceTag/stage": ["test"]         }       }     }   ] } </pre>

Ação: [RebuildEnvironment](#)

Recurso	Condições	Exemplo de declaração
environment	<p>InApplication</p> <p>aws:ResourceTag/ <i>key-name</i> (opcional)</p> <p>aws:TagKeys (opcional)</p>	<p>A política a seguir permite que a ação <code>RebuildEnvironment</code> recrie o ambiente <b>myenv</b> no aplicativo <b>My App</b>.</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:RebuildEnvironment"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication": [             "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"           ]         }       }     }   ] } </pre>

Ação: [RequestEnvironmentInfo](#)

Recurso	Condições	Exemplo de declaração
environment	<p>InApplication</p> <p>aws:ResourceTag/ <i>key-name</i> (opcional)</p> <p>aws:TagKeys (opcional)</p>	<p>A política a seguir permite que a ação RequestEnvironmentInfo compile informações sobre o ambiente <b>myenv</b> no aplicativo <b>My App</b>.</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:RequestEnvironmentInfo"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication":             ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"]         }       }     }   ] } </pre>

Ação: [RestartAppServer](#)

Recurso	Condições	Exemplo de declaração
environment	InApplication	<p>A política a seguir permite que a ação <code>RestartAppServer</code> reinicie o servidor do contêiner do aplicativo para o ambiente <code>myenv</code> no aplicativo <b>My App</b>.</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:RestartAppServer"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication": [             "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"           ]         }       }     }   ] }</pre>

Ação: [RetrieveEnvironmentInfo](#)



Recurso	Condições	Exemplo de declaração
environment	InApplication  aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação RetrieveEnvironmentInfo recupere as informações compiladas sobre o ambiente <b>myenv</b> no aplicativo <b>My App</b>.</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:RetrieveEnvironmentInfo"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication":             ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"]         }       }     }   ] } </pre>

Ação: [SwapEnvironmentCNAMEs](#)

Recurso	Condições	Exemplo de declaração
environment	InApplication (opcional)  FromEnvironment (opcional)	<p>A política a seguir permite que a ação <code>SwapEnvironmentCNAMEs</code> troque os CNAMEs dos ambientes <code>mysrcenv</code> e <code>mydestenv</code>.</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:SwapEnvironmentCNAMEs"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/mysrcenv",         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/mydestenv"       ]     }   ] } </pre>

Ação: [TerminateEnvironment](#)

Recurso	Condições	Exemplo de declaração
environment	<p>InApplication</p> <p>aws:ResourceTag/ <i>key-name</i> (opcional)</p> <p>aws:TagKeys (opcional)</p>	<p>A política a seguir permite que a ação <code>Terminate Environment</code> termine o ambiente <b>myenv</b> no aplicativo <b>My App</b>.</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:Terminate Environment"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us- east-2:123456789012:environment/My App/ myenv"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2 :123456789012:application/My App"]         }       }     }   ] } </pre>

Ação: [UpdateApplication](#)

Recurso	Condições	Exemplo de declaração
application	<p>aws:ResourceTag/ <i>key-name</i> (opcional)</p> <p>aws:TagKeys (opcional)</p>	<p>A política a seguir permite que a ação UpdateApplication atualize as propriedades do aplicativo <b>My App</b>.</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:UpdateApplication"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"       ]     }   ] }</pre>

Ação: [UpdateApplicationResourceLifecycle](#)

Recurso	Condições	Exemplo de declaração
application	<p>aws:ResourceTag/ <i>key-name</i> (opcional)</p> <p>aws:TagKeys (opcional)</p>	<p>A política a seguir permite que a ação <code>UpdateApplicationResourceLifecycle</code> atualize as configurações de ciclo de vida do aplicativo <b>My App</b>.</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:UpdateApplicationResourceLifecycle"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"       ]     }   ] }</pre>

Ação: [UpdateApplicationVersion](#)

Recurso	Condições	Exemplo de declaração
applicationversion	<p>InApplication</p> <p>aws:ResourceTag/ <i>key-name</i> (opcional)</p> <p>aws:TagKeys (opcional)</p>	<p>A política a seguir permite que a ação UpdateApplicationVersion atualize as propriedades da versão do aplicativo <b>My Version</b> no aplicativo <b>My App</b>.</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:UpdateApplicationVersion"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/MyApp/My Version"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication":             ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"]         }       }     }   ] } </pre>

Ação: [UpdateConfigurationTemplate](#)

Recurso	Condições	Exemplo de declaração
configurationtemplate	<p>InApplication</p> <p>aws:ResourceTag/ <i>key-name</i> (opcional)</p> <p>aws:TagKeys (opcional)</p>	<p>A política a seguir permite que a ação UpdateConfigurationTemplate atualize as propriedades ou as opções do modelo de configuração <b>My Template</b> no aplicativo <b>My App</b>.</p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:UpdateConfigurationTemplate"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication":             ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"]         }       }     }   ] }</pre>

Ação: [UpdateEnvironment](#)

Recurso	Condições	Exemplo de declaração
environment	InApplication  FromApplicationVersion  FromConfigurationTemplate  aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação UpdateEnvironment atualize o ambiente <b>myenv</b> no aplicativo <b>My App</b> implantando a versão do aplicativo <b>My Version</b>.</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:UpdateEnvironment"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication": [             "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App",             "elasticbeanstalk:FromApplicationVersion": [               "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"             ]           ]         }       }     }   ] } </pre>

Ação: [UpdateTagsForResource](#) – AddTags



Recurso	Condições	Exemplo de declaração
application applicationversion configurationtemplate environment platform	aws:ResourceTag/ <i>key-name</i> (opcional)  aws:RequestTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A ação AddTags é uma das duas ações virtuais associadas à API <a href="#">UpdateTagsForResource</a> .</p> <p>A política a seguir permite que a ação AddTags modifique as tags de recursos existentes somente se elas tiverem uma tag chamada stage com o valor test:</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:AddTags"       ],       "Effect": "Allow",       "Resource": "*",       "Condition": {         "StringEquals": {           "aws:ResourceTag/stage": ["test"]         }       }     }   ] } </pre>

Ação: [UpdateTagsForResource](#) – RemoveTags

Recurso	Condições	Exemplo de declaração
application applicationversion configurationtemplate environment platform	aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A ação <code>RemoveTags</code> é uma das duas ações virtuais associadas à API <a href="#">UpdateTagsForResource</a>.</p> <p>A política a seguir nega a ação <code>RemoveTags</code> para solicitar a remoção de uma tag chamada <code>stage</code> dos recursos existentes:</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:RemoveTags"       ],       "Effect": "Deny",       "Resource": "*",       "Condition": {         "ForAnyValue:StringEquals": {           "aws:TagKeys": ["stage"]         }       }     }   ] } </pre>

Ação: [ValidateConfigurationSettings](#)

Recurso	Condições	Exemplo de declaração
template environment	InApplication  aws:ResourceTag/ <i>key-name</i> (opcional)  aws:TagKeys (opcional)	<p>A política a seguir permite que a ação <code>ValidateConfigurationSettings</code> valide as configurações no ambiente <b>myenv</b> no aplicativo <b>My App</b>.</p> <pre> {   "Version": "2012-10-17",   "Statement": [     {       "Action": [         "elasticbeanstalk:ValidateConfigurationSettings"       ],       "Effect": "Allow",       "Resource": [         "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"       ],       "Condition": {         "StringEquals": {           "elasticbeanstalk:InApplication":             ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"]         }       }     }   ] } </pre>

## Chaves de condição para ações do Elastic Beanstalk

As chaves permitem que você especifique condições que expressam dependências, restringem permissões ou especificam restrições aos parâmetros de entrada de uma ação. O Elastic Beanstalk é compatível com as chaves a seguir.

### InApplication

Especifica o aplicativo que contém o recurso afetado pela ação.

O exemplo a seguir permite que a ação `UpdateApplicationVersion` atualize as propriedades da versão do aplicativo **My Version**. A condição `InApplication` especifica **My App** como o contêiner de **My Version**.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "elasticbeanstalk:UpdateApplicationVersion"
],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"
],
 "Condition": {
 "StringEquals": {
 "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"]
 }
 }
 }
]
}
```

## FromApplicationVersion

Especifica uma versão do aplicativo como uma dependência ou uma restrição a um parâmetro de entrada.

O exemplo a seguir permite que a ação `UpdateEnvironment` atualize o ambiente **myenv** no aplicativo **My App**. A condição `FromApplicationVersion` restringe o parâmetro `VersionLabel` para permitir que somente a versão do aplicativo **My Version** atualize o ambiente.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "elasticbeanstalk:UpdateEnvironment"
],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"
],
 "Condition": {
 "StringEquals": {
 "elasticbeanstalk:FromApplicationVersion": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App/My Version"]
 }
 }
 }
]
}
```

```

],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"
],
 "Condition": {
 "StringEquals": {
 "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],
 "elasticbeanstalk:FromApplicationVersion": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:applicationversion/My App/My Version"]
 }
 }
 }
]
}

```

## FromConfigurationTemplate

Especifica um modelo de configuração como uma dependência ou uma restrição a um parâmetro de entrada.

O exemplo a seguir permite que a ação `UpdateEnvironment` atualize o ambiente **myenv** no aplicativo **My App**. A condição `FromConfigurationTemplate` restringe o parâmetro `TemplateName` para permitir que somente o modelo de configuração **My Template** atualize o ambiente.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "elasticbeanstalk:UpdateEnvironment"
],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"
],
 "Condition": {
 "StringEquals": {
 "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],

```

```

 "elasticbeanstalk:FromConfigurationTemplate":
 ["arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My
 Template"]
 }
 }
}
]
}

```

## FromEnvironment

Especifica um ambiente como uma dependência ou uma restrição a um parâmetro de entrada.

O exemplo a seguir permite que a ação `SwapEnvironmentCNAMEs` troque os CNAMEs em **My App** para todos os ambientes cujos nomes começam com **mysrcenv** e **mydestenv**, mas não naqueles cujos nomes começam com **mysrcenvPROD\*** e **mydestenvPROD\***.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "elasticbeanstalk:SwapEnvironmentCNAMEs"
],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
 mysrcenv*",
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
 mydestenv*"
],
 "Condition": {
 "StringNotLike": {
 "elasticbeanstalk:FromEnvironment": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
 mysrcenvPROD*",
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
 mydestenvPROD*"
]
 }
 }
 }
]
}

```

```
}
```

## FromSolutionStack

Especifica um stack de solução como uma dependência ou uma restrição a um parâmetro de entrada.

A política a seguir permite que a ação `CreateConfigurationTemplate` crie modelos de configuração cujos nomes comecem com **My Template** (`My Template*`) no aplicativo **My App**. A condição `FromSolutionStack` restringe que o parâmetro `solutionstack` permita apenas a pilha de solução **32bit Amazon Linux running Tomcat 7** como o valor de entrada do parâmetro.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "elasticbeanstalk:CreateConfigurationTemplate"
],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My
App/My Template*"
],
 "Condition": {
 "StringEquals": {
 "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],
 "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-
east-2::solutionstack/32bit Amazon Linux running Tomcat 7"]
 }
 }
 }
]
}
```

`aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, `aws:TagKeys`

Especifique as condições com base na tag. Para obter mais detalhes, consulte [Usar tags para controlar o acesso aos recursos do Elastic Beanstalk](#).

## Usar tags para controlar o acesso aos recursos do Elastic Beanstalk

As condições em instruções de política do usuário do AWS Identity and Access Management (IAM) fazem parte da sintaxe que você usa para especificar permissões para os recursos de que as ações do Elastic Beanstalk precisam para serem realizadas. Para obter detalhes sobre como especificar condições de declaração de política, consulte [Recursos e condições para ações do Elastic Beanstalk](#). O uso de tags em condições é uma forma de controlar o acesso a recursos e solicitações. Para obter informações sobre como marcar recursos do Elastic Beanstalk, consulte [Marcar recursos da aplicação do Elastic Beanstalk](#). Este tópico discute o controle de acesso com base em tags.

Ao criar políticas do IAM, você pode definir permissões granulares concedendo acesso a recursos específicos. À medida que o número de recursos que você gerencia cresce, essa tarefa se torna mais difícil. Atribuir tags a recursos e usar tags em condições de declaração de política pode facilitar essa tarefa. Você concede acesso em massa a qualquer recurso com uma determinada tag. Depois, você aplica essa tag repetidamente a recursos relevantes, durante a criação ou posteriormente.

As tags podem ser anexadas ao recurso ou passadas na solicitação para serviços que ofereçam suporte a tags. No Elastic Beanstalk, os recursos podem ter tags, e algumas ações podem incluir tags. Ao criar uma política do IAM, você poderá usar chaves de condição de tag para controlar:

- Quais usuários podem executar ações em um ambiente, com base nas tags que ele já tem.
- Quais tags podem ser transmitidas na solicitação de uma ação.
- Se chaves de tags específicas podem ser usadas em uma solicitação.

Para obter a sintaxe e a semântica completas das chaves de condição de tag, consulte [Controlar o acesso usando tags](#) no Guia do usuário do IAM.

Os exemplos a seguir demonstram como especificar condições de tag em políticas para usuários do Elastic Beanstalk.

### Example 1: Limitar ações com base em tags na solicitação

A política de usuário gerenciada `AdministratorAccess-AWSElasticBeanstalk` do Elastic Beanstalk oferece aos usuários permissão ilimitada para executar qualquer ação do Elastic Beanstalk em qualquer recurso gerenciado pelo Elastic Beanstalk.

A política a seguir limita esse poder e nega a usuários não autorizados permissão para criar ambientes de produção do Elastic Beanstalk. Para fazer isso, ela negará a ação



CreateEnvironment se a solicitação especificar uma tag chamada stage com um dos valores gamma ou prod. Além disso, a política impede que esses usuários não autorizados interfiram nas fase de ambientes de produção, impedindo que ações de modificação de tags incluam esses mesmos valores de tag ou removam completamente a tag stage. O administrador de um cliente deve anexar essa política do IAM a usuários não autorizados do IAM, além da política de usuário gerenciada.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": [
 "elasticbeanstalk:CreateEnvironment",
 "elasticbeanstalk:AddTags"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "aws:RequestTag/stage": ["gamma", "prod"]
 }
 }
 },
 {
 "Effect": "Deny",
 "Action": [
 "elasticbeanstalk:RemoveTags"
],
 "Resource": "*",
 "Condition": {
 "ForAnyValue:StringEquals": {
 "aws:TagKeys": ["stage"]
 }
 }
 }
]
}
```

## Example 2: Limitar ações com base em tags de recursos

A política de usuário gerenciada `AdministratorAccess-AWSElasticBeanstalk` do Elastic Beanstalk oferece aos usuários permissão ilimitada para executar qualquer ação do Elastic Beanstalk em qualquer recurso gerenciado pelo Elastic Beanstalk.

A política a seguir limita esse poder e nega a usuários não autorizados permissão para realizar ações em ambientes de produção do Elastic Beanstalk. Para fazer isso, ela negará ações específicas se o ambiente tiver uma tag chamada `stage` com um dos valores `gamma` ou `prod`. O administrador de um cliente deve anexar essa política do IAM a usuários não autorizados do IAM, além da política de usuário gerenciada.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": [
 "elasticbeanstalk:AddTags",
 "elasticbeanstalk:RemoveTags",
 "elasticbeanstalk:DescribeEnvironments",
 "elasticbeanstalk:TerminateEnvironment",
 "elasticbeanstalk:UpdateEnvironment",
 "elasticbeanstalk:ListTagsForResource"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "aws:ResourceTag/stage": ["gamma", "prod"]
 }
 }
 }
]
}
```

## Example 3: Permitir ações com base em tags na solicitação

A política a seguir concede aos usuários permissão para criar aplicações de desenvolvimento do Elastic Beanstalk.

Para fazer isso, ela permitirá as ações `CreateApplication` e `AddTags` se a solicitação especificar uma tag chamada `stage` com o valor `development`. A condição `aws:TagKeys` garante que o

usuário não possa adicionar outras chaves de tag. Especificamente, ele garante a diferenciação entre maiúsculas e minúsculas da chave da tag `stage`. Observe que essa política é útil para os usuários do IAM que não tenham a política de usuário gerenciada `AdministratorAccess-AWSElasticBeanstalk` do Elastic Beanstalk anexada. A política gerenciada oferece aos usuários permissão ilimitada para executar qualquer ação do Elastic Beanstalk em qualquer recurso gerenciado pelo Elastic Beanstalk.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticbeanstalk:CreateApplication",
 "elasticbeanstalk:AddTags"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "aws:RequestTag/stage": "development"
 },
 "ForAllValues:StringEquals": {
 "aws:TagKeys": ["stage"]
 }
 }
 }
]
}
```

#### Example 4: Permitir ações com base em tags de recursos

A política a seguir concede aos usuários permissão para executar ações e obter informações sobre as aplicações de desenvolvimento do Elastic Beanstalk.

Para fazer isso, ela permitirá ações específicas se o aplicativo tiver uma tag chamada `stage` com o valor `development`. A condição `aws:TagKeys` garante que o usuário não possa adicionar outras chaves de tag. Especificamente, ele garante a diferenciação entre maiúsculas e minúsculas da chave da tag `stage`. Observe que essa política é útil para os usuários do IAM que não tenham a política de usuário gerenciada `AdministratorAccess-AWSElasticBeanstalk` do Elastic Beanstalk anexada. A política gerenciada oferece aos usuários permissão ilimitada para executar qualquer ação do Elastic Beanstalk em qualquer recurso gerenciado pelo Elastic Beanstalk.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticbeanstalk:UpdateApplication",
 "elasticbeanstalk>DeleteApplication",
 "elasticbeanstalk:DescribeApplications"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "aws:ResourceTag/stage": "development"
 },
 "ForAllValues:StringEquals": {
 "aws:TagKeys": ["stage"]
 }
 }
 }
]
}
```

## Exemplo de políticas baseadas em políticas gerenciadas

Esta seção demonstra como controlar o acesso do usuário ao AWS Elastic Beanstalk e inclui exemplos de políticas que fornecem o acesso necessário para cenários comuns. Essas políticas são derivadas das políticas gerenciadas do Elastic Beanstalk. Para obter informações sobre como anexar políticas gerenciadas a usuários e grupos, consulte [Gerenciar políticas de usuário do Elastic Beanstalk](#).

Neste cenário, Exemplo Corp. é uma empresa de software com três equipes responsáveis pelo site da empresa: os administradores que gerenciam a infraestrutura, os desenvolvedores que criam o software para o site e uma equipe de controle de qualidade que testa o site. Para ajudar a gerenciar permissões para os recursos do Elastic Beanstalk, a Exemplo Corp. cria três grupos aos quais os membros de cada equipe pertencem: administradores, desenvolvedores e testadores. A Exemplo Corp. deseja que o grupo de administradores tenha acesso total a todas as aplicações, ambientes e seus recursos subjacentes, para que possam criar, solucionar problemas e excluir todos os ativos do Elastic Beanstalk. Os desenvolvedores precisam de permissões para visualizar todos os ativos do Elastic Beanstalk e para criar e implantar versões da aplicação. Os desenvolvedores não devem

ter permissão para criar novos aplicativos ou ambientes ou para encerrar ambientes em execução. Os testadores precisam visualizar todos os recursos do Elastic Beanstalk para monitorar e testar aplicações. Os testadores não devem ser capazes de fazer alterações em nenhum recurso do Elastic Beanstalk.

O seguinte exemplo de políticas fornece as permissões necessárias para cada grupo.

### Exemplo 1: Grupo de administradores: todas as APIs do Elastic Beanstalk e APIs de serviço relacionadas

A política a seguir oferece aos usuários permissões para todas as ações necessárias para usar o Elastic Beanstalk. Ela também permite que o Elastic Beanstalk provisione e gerencie recursos em seu nome nos serviços a seguir. O Elastic Beanstalk depende desses serviços adicionais para provisionar recursos subjacentes ao criar um ambiente.

- Amazon Elastic Compute Cloud
- Elastic Load Balancing
- Auto Scaling
- Amazon CloudWatch
- Amazon Simple Storage Service
- Amazon Simple Notification Service
- Amazon Relational Database Service
- AWS CloudFormation

Observe que esta política é um exemplo. Ela fornece um amplo conjunto de permissões aos serviços da AWS que o Elastic Beanstalk usa para gerenciar aplicações e ambientes. Por exemplo, `ec2:*` permite que um usuário do AWS Identity and Access Management (IAM) realize qualquer ação em qualquer recurso do Amazon EC2 na conta da AWS. Essas permissões não estão limitadas aos recursos utilizados com o Elastic Beanstalk. Como uma das melhores práticas, você deve conceder a indivíduos apenas as permissões de que precisam para executar suas tarefas.

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

```

 "elasticbeanstalk:*",
 "ec2:*",
 "elasticloadbalancing:*",
 "autoscaling:*",
 "cloudwatch:*",
 "s3:*",
 "sns:*",
 "rds:*",
 "cloudformation:*"
],
 "Resource" : "*"
}
]
}

```

## Exemplo 2: Grupo de desenvolvedores: todas as operações, exceto altamente privilegiadas

A política a seguir nega permissão para criar aplicações e ambientes e permite todas as demais ações do Elastic Beanstalk.

Observe que esta política é um exemplo. Ela fornece um amplo conjunto de permissões para produtos da AWS que o Elastic Beanstalk usa para gerenciar aplicações e ambientes. Por exemplo, `ec2:*` permite que um usuário do IAM realize qualquer ação em qualquer recurso do Amazon EC2 na conta da AWS. Essas permissões não estão limitadas aos recursos utilizados com o Elastic Beanstalk. Como uma das melhores práticas, você deve conceder a indivíduos apenas as permissões de que precisam para executar suas tarefas.

```

{
 "Version" : "2012-10-17",
 "Statement" : [
 {
 "Action" : [
 "elasticbeanstalk:CreateApplication",
 "elasticbeanstalk:CreateEnvironment",
 "elasticbeanstalk>DeleteApplication",
 "elasticbeanstalk:RebuildEnvironment",
 "elasticbeanstalk:SwapEnvironmentCNAMEs",
 "elasticbeanstalk:TerminateEnvironment"],
 "Effect" : "Deny",
 "Resource" : "*"
 },

```

```
{
 "Action" : [
 "elasticbeanstalk:*",
 "ec2:*",
 "elasticloadbalancing:*",
 "autoscaling:*",
 "cloudwatch:*",
 "s3:*",
 "sns:*",
 "rds:*",
 "cloudformation:*"],
 "Effect" : "Allow",
 "Resource" : "*"
}
]
```

### Exemplo 3: Testadores: somente visualização

A seguinte política permite acesso somente leitura a todos os aplicativos, versões de aplicativos, eventos e ambientes. Não permite realizar nenhuma ação.

```
{
 "Version" : "2012-10-17",
 "Statement" : [
 {
 "Effect" : "Allow",
 "Action" : [
 "elasticbeanstalk:Check*",
 "elasticbeanstalk:Describe*",
 "elasticbeanstalk:List*",
 "elasticbeanstalk:RequestEnvironmentInfo",
 "elasticbeanstalk:RetrieveEnvironmentInfo",
 "ec2:Describe*",
 "elasticloadbalancing:Describe*",
 "autoscaling:Describe*",
 "cloudwatch:Describe*",
 "cloudwatch:List*",
 "cloudwatch:Get*",
 "s3:Get*",
 "s3:List*",
 "sns:Get*",
```

```
 "sns:List*",
 "rds:Describe*",
 "cloudformation:Describe*",
 "cloudformation:Get*",
 "cloudformation:List*",
 "cloudformation:Validate*",
 "cloudformation:Estimate*"
],
 "Resource" : "*"
}
]
```

## Políticas de exemplo baseadas em permissões de recursos

Esta seção aborda um caso de uso para controlar as permissões de usuário para ações do Elastic Beanstalk que acessam recursos específicos do Elastic Beanstalk. Apresentaremos as políticas de exemplo compatíveis com o caso de uso. Para obter mais informações sobre os recursos do Elastic Beanstalk, consulte [Criar uma política de usuário personalizada](#). Para obter informações sobre como anexar políticas a usuários e grupos, acesse [Gerenciamento de políticas do IAM](#) em Uso do AWS Identity and Access Management.

No nosso caso de uso, Exemplo Corp. é uma pequena empresa de consultoria que desenvolve aplicativos para dois clientes diferentes. John é o gerente de desenvolvimento que supervisiona o desenvolvimento das duas aplicações do Elastic Beanstalk, app1 e app2. John faz o desenvolvimento e alguns testes nos dois aplicativos, e apenas ele pode atualizar o ambiente de produção dos dois aplicativos. Estas são as permissões de que ele precisa para o app1 e o app2:

- Visualização do aplicativo, versões de aplicativos, ambientes e modelos de configuração
- Criação das versões do aplicativo e implantação no ambiente de preparação
- Atualização do ambiente de produção
- Criação e encerramento de ambientes

Jill é uma testadora que precisa de acesso para visualizar os seguintes recursos a fim de monitorar e testar os dois aplicativos: aplicativos, versões de aplicativos, ambientes e modelos de configuração. No entanto, ela não deve ser capaz de fazer alterações em nenhum recurso do Elastic Beanstalk.

Jack é o desenvolvedor do app1 que precisa de acesso para visualizar todos os recursos do app1 e também precisa criar versões de aplicativos para o app1 e implantá-las no ambiente de preparação.



Judy é a administradora da conta da AWS para a Exemplo Corp. Ela já criou usuários do IAM para John, Jill e Jack, e anexa as políticas a seguir a esses usuários para conceder as permissões apropriadas para os aplicativos app1 e app2.

### Exemplo 1: John: gerente de desenvolvimento para app1, app2

Dividimos a política de John em três políticas separadas para facilitar a leitura e o gerenciamento. Juntas, elas oferecem a John as permissões de que precisa para executar as ações de desenvolvimento, teste e implantação nos dois aplicativos.

A primeira política especifica ações para Auto Scaling, Amazon S3, Amazon EC2, CloudWatch, Amazon SNS, Elastic Load Balancing, Amazon RDS e AWS CloudFormation. O Elastic Beanstalk depende desses serviços adicionais para provisionar recursos subjacentes ao criar um ambiente.

Observe que esta política é um exemplo. Ela fornece um amplo conjunto de permissões para produtos da AWS que o Elastic Beanstalk usa para gerenciar aplicações e ambientes. Por exemplo, `ec2:*` permite que um usuário do IAM realize qualquer ação em qualquer recurso do Amazon EC2 na conta da AWS. Essas permissões não estão limitadas aos recursos utilizados com o Elastic Beanstalk. Como uma das melhores práticas, você deve conceder a indivíduos apenas as permissões de que precisam para executar suas tarefas.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ec2:*",
 "ecs:*",
 "ecr:*",
 "elasticloadbalancing:*",
 "autoscaling:*",
 "cloudwatch:*",
 "s3:*",
 "sns:*",
 "cloudformation:*",
 "dynamodb:*",
 "rds:*",
 "sqs:*",
 "logs:*",
 "iam:GetPolicyVersion",
 "iam:GetRole",

```

```

 "iam:PassRole",
 "iam:ListRolePolicies",
 "iam:ListAttachedRolePolicies",
 "iam:ListInstanceProfiles",
 "iam:ListRoles",
 "iam:ListServerCertificates",
 "acm:DescribeCertificate",
 "acm:ListCertificates",
 "codebuild:CreateProject",
 "codebuild>DeleteProject",
 "codebuild:BatchGetBuilds",
 "codebuild:StartBuild"
],
 "Resource": "*"
}
]
}

```

A segunda política especifica as ações do Elastic Beanstalk que John tem permissão para executar nos recursos do app1 e do app2. A declaração `AllCallsInApplications` permite que todas as ações do Elastic Beanstalk (`"elasticbeanstalk:*"`) sejam realizadas em todos os recursos no app1 e no app2 (por exemplo, `elasticbeanstalk:CreateEnvironment`). A declaração `AllCallsOnApplications` permite todas as ações do Elastic Beanstalk (`"elasticbeanstalk:*"`) nos recursos do app1 e do app2 (por exemplo, `elasticbeanstalk:DescribeApplications`, `elasticbeanstalk:UpdateApplication`, etc.). A declaração `AllCallsOnSolutionStacks` permite todas as ações do Elastic Beanstalk (`"elasticbeanstalk:*"`) para os recursos de pilha de solução (por exemplo, `elasticbeanstalk:ListAvailableSolutionStacks`).

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "AllCallsInApplications",
 "Action": [
 "elasticbeanstalk:*"
],
 "Effect": "Allow",
 "Resource": [
 "*"
],
 "Condition": {

```

```

 "StringEquals":{
 "elasticbeanstalk:InApplication":[
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1",
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app2"
]
 }
 },
 {
 "Sid":"AllCallsOnApplications",
 "Action":[
 "elasticbeanstalk:*"
],
 "Effect":"Allow",
 "Resource":[
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1",
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app2"
]
 },
 {
 "Sid":"AllCallsOnSolutionStacks",
 "Action":[
 "elasticbeanstalk:*"
],
 "Effect":"Allow",
 "Resource":[
 "arn:aws:elasticbeanstalk:us-east-2::solutionstack/*"
]
 }
]
}

```

A terceira política especifica as ações do Elastic Beanstalk para as quais a segunda política precisa de permissões para concluir essas ações do Elastic Beanstalk. A instrução `AllNonResourceCalls` permite a ação `elasticbeanstalk:CheckDNSAvailability`, que é necessária para chamar `elasticbeanstalk:CreateEnvironment` e outras ações. Ela também permite a ação `elasticbeanstalk:CreateStorageLocation`, que é necessária para `elasticbeanstalk:CreateApplication`, `elasticbeanstalk:CreateEnvironment` e outras ações.

```

{
 "Version": "2012-10-17",

```

```

"Statement":[
 {
 "Sid":"AllNonResourceCalls",
 "Action":[
 "elasticbeanstalk:CheckDNSAvailability",
 "elasticbeanstalk:CreateStorageLocation"
],
 "Effect":"Allow",
 "Resource":[
 "*"
]
 }
]
}

```

## Exemplo 2: Jill: testadora do app1, app2

Dividimos a política de Jill em três separadas para facilitar a leitura e o gerenciamento. Juntas, elas oferecem a Jill as permissões de que precisa para executar as ações de teste e monitoramento nos dois aplicativos.

A primeira política especifica as ações `Describe*`, `List*` e `Get*` no Auto Scaling, Amazon S3, Amazon EC2, CloudWatch, Amazon SNS, Elastic Load Balancing, Amazon RDS e AWS CloudFormation (para tipos de contêiner não herdados) para que as ações do Elastic Beanstalk possam recuperar as informações sobre os recursos subjacentes das aplicações app1 e app2.

```

{
 "Version": "2012-10-17",
 "Statement":[
 {
 "Effect":"Allow",
 "Action":[
 "ec2:Describe*",
 "elasticloadbalancing:Describe*",
 "autoscaling:Describe*",
 "cloudwatch:Describe*",
 "cloudwatch>List*",
 "cloudwatch:Get*",
 "s3:Get*",
 "s3>List*",
 "sns:Get*",
 "sns>List*",

```

```

 "rds:Describe*",
 "cloudformation:Describe*",
 "cloudformation:Get*",
 "cloudformation:List*",
 "cloudformation:Validate*",
 "cloudformation:Estimate*"
],
 "Resource": "*"
}
]
}

```

A segunda política especifica as ações do Elastic Beanstalk que Jill tem permissão para executar nos recursos do app1 e do app2. A instrução `AllReadCallsInApplications` permite que ela chame as ações `Describe*` e as ações de informações do ambiente. A instrução `AllReadCallsOnApplications` permite que ela chame as ações `DescribeApplications` and `DescribeEvents` nos recursos de aplicativos do app1 e do app2. A instrução `AllReadCallsOnSolutionStacks` permite visualizar as ações que envolvem recursos de pilha de soluções (`ListAvailableSolutionStacks`, `DescribeConfigurationOptions` e `ValidateConfigurationSettings`).

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "AllReadCallsInApplications",
 "Action": [
 "elasticbeanstalk:Describe*",
 "elasticbeanstalk:RequestEnvironmentInfo",
 "elasticbeanstalk:RetrieveEnvironmentInfo"
],
 "Effect": "Allow",
 "Resource": [
 "*"
],
 "Condition": {
 "StringEquals": {
 "elasticbeanstalk:InApplication": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1",
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app2"
]
 }
 }
 }
]
}

```

```

 }
 },
 {
 "Sid": "AllReadCallsOnApplications",
 "Action": [
 "elasticbeanstalk:DescribeApplications",
 "elasticbeanstalk:DescribeEvents"
],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1",
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app2"
]
 },
 {
 "Sid": "AllReadCallsOnSolutionStacks",
 "Action": [
 "elasticbeanstalk:ListAvailableSolutionStacks",
 "elasticbeanstalk:DescribeConfigurationOptions",
 "elasticbeanstalk:ValidateConfigurationSettings"
],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:us-east-2::solutionstack/*"
]
 }
]
}

```

A terceira política especifica as ações do Elastic Beanstalk para as quais a segunda política precisa de permissões para concluir essas ações do Elastic Beanstalk. A instrução `AllNonResourceCalls` permite a ação `elasticbeanstalk:CheckDNSAvailability`, que é necessária para chamar ações de visualização.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "AllNonResourceCalls",
 "Action": [
 "elasticbeanstalk:CheckDNSAvailability"
],
 "Effect": "Allow",

```

```

 "Resource": [
 "*"
]
 }
]
}

```

### Exemplo 3: Jack: desenvolvedor do app1

Dividimos a política de Jack em três separadas para facilitar a leitura e o gerenciamento. Juntas, elas oferecem a Jack as permissões de que precisa para executar as ações de teste, monitoramento e implantação no recurso do app1.

A primeira política especifica as ações em Auto Scaling, Amazon S3, Amazon EC2, CloudWatch, Amazon SNS, Elastic Load Balancing, Amazon RDS e AWS CloudFormation (para tipos de contêiner não herdados) para que as ações do Elastic Beanstalk possam visualizar e trabalhar com os recursos subjacentes da app1. Para obter uma lista de tipos de contêiner não legados compatíveis, consulte [. the section called “Por que algumas versões de plataforma são marcadas como legadas?”](#)

Observe que esta política é um exemplo. Ela fornece um amplo conjunto de permissões para produtos da AWS que o Elastic Beanstalk usa para gerenciar aplicações e ambientes. Por exemplo, `ec2:*` permite que um usuário do IAM realize qualquer ação em qualquer recurso do Amazon EC2 na conta da AWS. Essas permissões não estão limitadas aos recursos utilizados com o Elastic Beanstalk. Como uma das melhores práticas, você deve conceder a indivíduos apenas as permissões de que precisam para executar suas tarefas.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ec2:*",
 "elasticloadbalancing:*",
 "autoscaling:*",
 "cloudwatch:*",
 "s3:*",
 "sns:*",
 "rds:*",
 "cloudformation:*"
]
 }
],
}

```

```

 "Resource": "*"
 }
]
}

```

A segunda política especifica as ações do Elastic Beanstalk que Jack tem permissão para executar no recurso do app1.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "AllReadCallsAndAllVersionCallsInApplications",
 "Action": [
 "elasticbeanstalk:Describe*",
 "elasticbeanstalk:RequestEnvironmentInfo",
 "elasticbeanstalk:RetrieveEnvironmentInfo",
 "elasticbeanstalk:CreateApplicationVersion",
 "elasticbeanstalk>DeleteApplicationVersion",
 "elasticbeanstalk:UpdateApplicationVersion"
],
 "Effect": "Allow",
 "Resource": [
 "*"
],
 "Condition": {
 "StringEquals": {
 "elasticbeanstalk:InApplication": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1"
]
 }
 }
 },
 {
 "Sid": "AllReadCallsOnApplications",
 "Action": [
 "elasticbeanstalk:DescribeApplications",
 "elasticbeanstalk:DescribeEvents"
],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1"
]
 }
]
}

```



```

 },
 {
 "Sid": "UpdateEnvironmentInApplications",
 "Action": [
 "elasticbeanstalk:UpdateEnvironment"
],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/app1/app1-
staging*"
],
 "Condition": {
 "StringEquals": {
 "elasticbeanstalk:InApplication": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1"
]
 },
 "StringLike": {
 "elasticbeanstalk:FromApplicationVersion": [
 "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/
app1/*"
]
 }
 }
 }
],
 {
 "Sid": "AllReadCallsOnSolutionStacks",
 "Action": [
 "elasticbeanstalk:ListAvailableSolutionStacks",
 "elasticbeanstalk:DescribeConfigurationOptions",
 "elasticbeanstalk:ValidateConfigurationSettings"
],
 "Effect": "Allow",
 "Resource": [
 "arn:aws:elasticbeanstalk:us-east-2::solutionstack/*"
]
 }
]
}

```

A terceira política especifica as ações do Elastic Beanstalk para as quais a segunda política precisa de permissões para concluir essas ações do Elastic Beanstalk. A instrução `AllNonResourceCalls` permite a ação `elasticbeanstalk:CheckDNSAvailability`, que

é necessária para chamar `elasticbeanstalk:CreateEnvironment` e outras ações. Ela também permite a ação `elasticbeanstalk:CreateStorageLocation`, que é necessária para `elasticbeanstalk:CreateEnvironment` e outras ações.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "AllNonResourceCalls",
 "Action": [
 "elasticbeanstalk:CheckDNSAvailability",
 "elasticbeanstalk:CreateStorageLocation"
],
 "Effect": "Allow",
 "Resource": [
 "*"
]
 }
]
}
```

## Usar o Elastic Beanstalk com o Amazon RDS

É possível usar o Elastic Beanstalk com o Amazon Relational Database Service (Amazon RDS) para configurar, operar e escalar um banco de dados relacional. Há duas opções para começar, que são as seguintes:

- Criar um novo banco de dados do Amazon RDS.
- Comece com um banco de dados que foi [criado pelo Elastic Beanstalk](#) anteriormente e que depois foi [desacoplado](#) de um ambiente do Beanstalk. Para obter mais informações, consulte [the section called “Banco de dados”](#).

Você pode usar qualquer uma das duas abordagens para executar uma instância de banco de dados no Amazon RDS e configurar a sua aplicação para conectar-se a ela no lançamento. Você pode conectar vários ambientes a um banco de dados e também executar atualizações contínuas com implantações azul-verde.

**Note**

Se você nunca usou uma instância de banco de dados com sua aplicação, recomendamos que primeiro adicione um banco de dados a um ambiente de teste com o console do Elastic Beanstalk. Fazendo isso, você pode verificar se a aplicação pode ler as propriedades do ambiente, criar uma sequência de conexão e conectar-se a uma instância de banco de dados, sem o trabalho de configuração adicional necessário para um banco de dados autônomo. Para obter mais informações, consulte [Adição de um banco de dados ao seu ambiente do Elastic Beanstalk](#).

Para permitir que as instâncias do Amazon EC2 no seu ambiente se conectem a um banco de dados externo, configure um grupo de segurança adicional para o grupo do Auto Scaling associado ao seu ambiente. Você pode anexar o mesmo grupo de segurança anexado à sua instância de banco de dados. Você também pode usar um grupo de segurança separado. Se você anexar um grupo de segurança diferente, deve configurar o grupo de segurança anexado ao banco de dados para permitir o acesso de entrada a partir desse grupo de segurança.

**Note**

É possível conectar seu ambiente a um banco de dados adicionando uma regra ao grupo de segurança anexado ao banco de dados. Essa regra deve permitir o acesso de entrada a partir do grupo de segurança gerado automaticamente que o Elastic Beanstalk anexa ao grupo do Auto Scaling do seu ambiente. Mas saiba que, ao criar essa regra, você também cria uma dependência entre os dois grupos de segurança. Subsequentemente, quando você tentar terminar o ambiente, o Elastic Beanstalk não poderá excluir o grupo de segurança do ambiente, porque o grupo de segurança do banco de dados é dependente dele.

Após iniciar sua instância de banco de dados e configurar grupos de segurança, você pode passar as informações de conexão (endpoint, senha, etc.) para sua aplicação usando as propriedades do ambiente. Este é o mesmo mecanismo que o Elastic Beanstalk usa em segundo plano quando você executa uma instância de banco de dados no seu ambiente.

Para ter uma camada de segurança adicional, é possível armazenar suas informações de conexão no Amazon S3 e configurar o Elastic Beanstalk para recuperá-las durante a implantação. Com os [arquivos de configuração \(.ebextensions\)](#), é possível configurar as instâncias no ambiente para recuperar arquivos do Amazon S3 com segurança ao implantar a aplicação.

## Tópicos

- [Iniciar e conectar-se a uma instância externa do Amazon RDS em uma VPC padrão](#)
- [Iniciar e conectar-se a uma instância externa do Amazon RDS no EC2 Classic](#)
- [Armazenando as credenciais do Amazon RDS em AWS Secrets Manager](#)
- [Limpar uma instância externa do Amazon RDS](#)

## Iniciar e conectar-se a uma instância externa do Amazon RDS em uma VPC padrão

Para usar um banco de dados externo com uma aplicação em execução no Elastic Beanstalk, você tem duas opções. Você pode iniciar uma instância de banco de dados com o Amazon RDS. Qualquer instância executada com o Amazon RDS é completamente independente do Elastic Beanstalk e dos seus ambientes do Elastic Beanstalk. Isso significa que você pode usar qualquer mecanismo de banco de dados e tipo de instância suportados pelo Amazon RDS, mesmo os que não são usados pelo Elastic Beanstalk.

Ou, em vez de iniciar uma nova instância de banco de dados, você pode começar com um banco de dados que foi [criado pelo Elastic Beanstalk](#) anteriormente e que depois foi [desacoplado](#) de um ambiente do Beanstalk. Para obter mais informações, consulte [the section called “Banco de dados”](#). Com essa opção, você não precisa realizar o procedimento para iniciar um novo banco de dados. Mas, você precisa realizar os procedimentos subsequentes descritos neste tópico.

Os procedimentos a seguir descrevem o processo para uma [VPC padrão](#). O processo é o mesmo se você estiver usando uma VPC personalizada. Os únicos requisitos adicionais são que o seu ambiente e instância de banco de dados estejam na mesma sub-rede ou em sub-redes que possam se comunicar umas com as outras. Para obter mais informações sobre como configurar uma VPC personalizada para usar com o Elastic Beanstalk, consulte [Usar o Elastic Beanstalk com Amazon VPC](#).

### Note

- Se estiver começando com um banco de dados que foi criado pelo Elastic Beanstalk e posteriormente desacoplado de um ambiente do Beanstalk, você pode pular o primeiro grupo de etapas e continuar com as etapas em Para modificar as regras de entrada no grupo de segurança da instância do RDS.

- Se planeja usar o banco de dados que você desacopla para um ambiente de produção, verifique se o tipo de armazenamento usado pelo banco de dados é adequado para sua workload. Para obter mais informações, consulte [DB Instance Storage](#) (Armazenamento de instâncias de banco de dados) e [Modifying a DB instance](#) (Modificar uma instância de banco de dados) no Amazon RDS User Guide (Guia do usuário do Amazon RDS).

Para executar uma instância de banco de dados do RDS em uma VPC padrão

1. Abra o [console do RDS](#).
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Escolha Create database (Criar banco de dados).
4. Escolha Standard Create (Criação padrão).

 Important

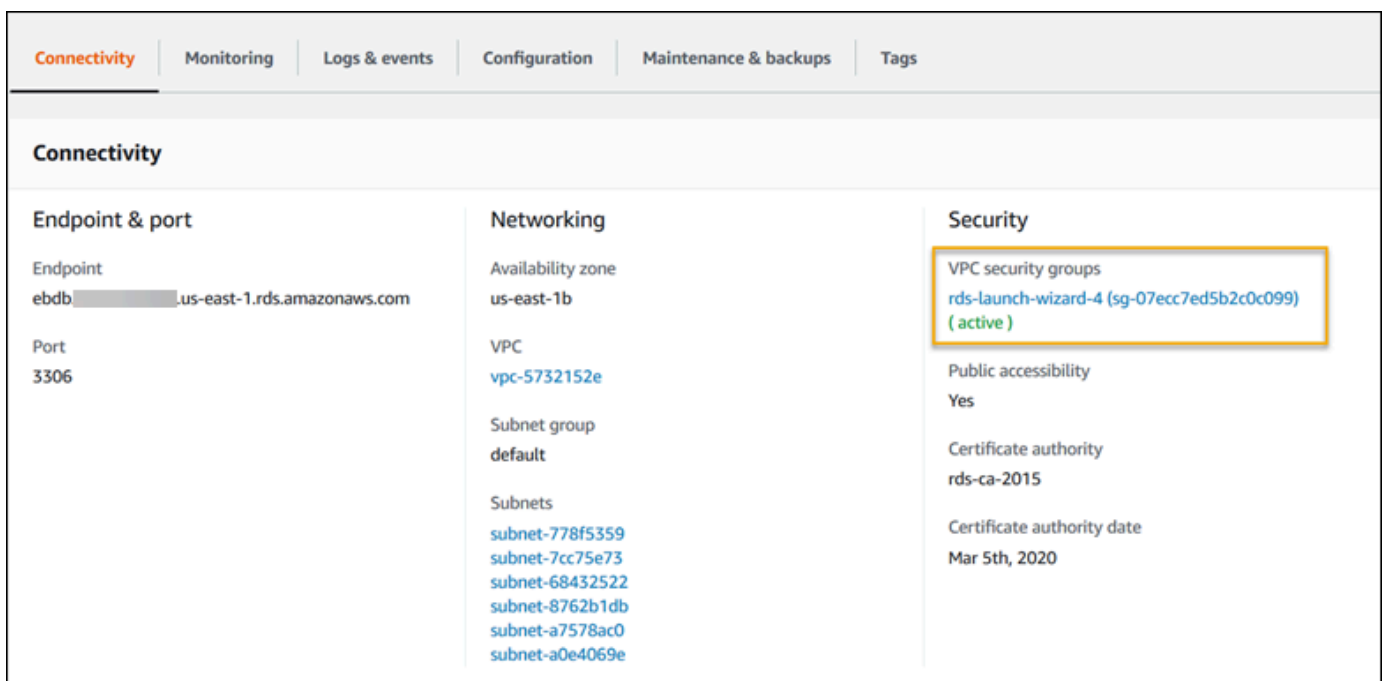
Não escolha Easy Create (Criação fácil). Se o escolher, você não poderá definir as configurações necessárias para iniciar esse banco de dados do RDS.

5. Em Additional configuration (Configuração adicional), para Initial database name (Nome do banco de dados inicial), digite **ebdb**.
6. Revise as configurações padrão e ajuste essas configurações de acordo com seus requisitos específicos. Preste atenção às seguintes opções:
  - DB instance class (Classe de instância de banco de dados): escolha um tamanho de instância que tem uma quantidade apropriada de memória e energia de CPU para sua workload.
  - Multi-AZ deployment (Implantação Multi-AZ): para obter alta disponibilidade, defina como Create an Aurora Replica/Reader node in a different AZ (Criar uma réplica Aurora/nó do Leitor em um AZ diferente).
  - Master username (Nome do usuário principal) e Master password (Senha principal): o nome de usuário do banco de dados e senha. Anote essas configurações porque você vai usá-las mais tarde.
7. Verifique as configurações padrão para as opções restantes e selecione Create database (Criar banco de dados).

Em seguida, modifique o grupo de segurança anexado à sua instância de banco de dados para permitir tráfego de entrada na porta apropriada. Este é o mesmo grupo de segurança que você vai anexar ao ambiente do Elastic Beanstalk mais tarde. Como resultado, a regra que você adiciona concederá permissão de acesso de entrada a outros recursos no mesmo grupo de segurança.

Para modificar as regras de entrada no grupo de segurança anexado à instância do RDS

1. Abra o [console do Amazon RDS](#).
2. Escolha Databases (Bancos de dados).
3. Escolha o nome da sua instância de banco de dados para visualizar detalhes.
4. Na seção Connectivity (Conectividade), anote as Subnets (Sub-redes), os Security groups (Grupos de segurança) e o Endpoint que são exibidos nesta página. Faça isso para poder usar essas informações mais tarde.
5. Em Security (Segurança), você pode ver o grupo de segurança associado à instância de banco de dados. Abra o link para visualizar o grupo de segurança no console do Amazon EC2.



6. Nos detalhes do grupo de segurança, escolha a guia Inbound (Entrada).
7. Escolha Editar.
8. Escolha Add Rule (Adicionar regra).
9. Para Tipo, escolha o mecanismo de banco de dados usado pelo aplicativo.
10. Em Source (Origem), digite **sg-** para exibir uma lista dos grupos de segurança disponíveis. Escolha o grupo de segurança associado ao grupo do Auto Scaling usado com seu ambiente

do Elastic Beanstalk. Isso é para que as instâncias do Amazon EC2 no ambiente possam ter acesso ao banco de dados.

Type	Protocol	Port Range	Source	Description
MySQL/Aurora	TCP	3306	Custom 72.21.198.67/32	e.g. SSH for Admin Desktop
MySQL/Aurora	TCP	3306	Custom sg-	e.g. SSH for Admin Desktop

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

11. Escolha Save (Salvar).

Em seguida, adicione o grupo de segurança da instância de banco de dados ao seu ambiente em execução. Nesse procedimento o Elastic Beanstalk provisiona novamente todas as instâncias do ambiente com o grupo de segurança adicional anexado.

Para adicionar um security group ao seu ambiente

- Faça um dos seguintes procedimentos:
  - Como adicionar um grupo de segurança usando o console do Elastic Beanstalk
    - a. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
    - b. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

**Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.


- c. No painel de navegação, escolha Configuration (Configuração).
- d. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar).

- e. Em EC2 security groups (Grupos de segurança do EC2), escolha o grupo de segurança a ser anexado às instâncias, além do grupo de segurança da instância que o Elastic Beanstalk criará.
  - f. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.
  - g. Leia o aviso e escolha Confirm.
- Para adicionar um grupo de segurança usando um [arquivo de configuração](#), use o exemplo de arquivo [securitygroup-addexisting.config](#).

Em seguida, passe as informações de conexão para o seu ambiente usando as propriedades de ambiente. Quando você [adiciona uma instância de banco de dados ao ambiente](#) com o console do Elastic Beanstalk, o Elastic Beanstalk usa propriedades de ambiente, como RDS\_HOSTNAME, para passar informações de conexão à aplicação. Você pode usar as mesmas propriedades. Fazendo isso, você usa o mesmo código de aplicação com as instâncias de banco de dados integradas e externas. Ou, alternativamente, você pode escolher seus próprios nomes de propriedade.

Como configurar as propriedades do ambiente para uma instância de banco de dados do Amazon RDS

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Na seção Environment properties, defina as variáveis que seu aplicativo lê para construir uma string de conexão. Para compatibilidade com ambientes que têm uma instância de banco de dados do RDS integrada, use os seguintes nomes e valores. É possível encontrar todos os valores, exceto sua senha, no [console do RDS](#).



Nome da propriedade	Descrição	Valor da propriedade
RDS_HOSTNAME	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
RDS_PORT	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
RDS_DB_NAME	O nome do banco de dado, <b>ebdb</b> .	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
RDS_USERNAME	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

### Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	<input type="text" value="ebdb"/> ✕
RDS_HOSTNAME	<input type="text" value="webapp-db.jxzc b5mpaniu.us-wes"/> ✕
RDS_PORT	<input type="text" value="5432"/> ✕
RDS_USERNAME	<input type="text" value="webapp-admin"/> ✕
<input type="text" value="RDS_PASSWORD"/>	<input type="text" value="kUj5uKxmWDMYc403"/> +

[Cancel](#) [Apply](#)

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.


Se você ainda não programou a aplicação para ler as propriedades de ambiente e criar uma sequência de conexão, consulte os seguintes tópicos específicos de cada linguagem para obter instruções:

- Java SE: [Conectar a um banco de dados \(plataformas Java SE\)](#)
- Java com Tomcat: [Conectar a um banco de dados \(plataformas Tomcat\)](#)
- Node.js: [Conectar-se a um banco de dados](#)
- .NET: [Conectar-se a um banco de dados](#)
- PHP: [Conexão com banco de dados com um PDO ou MySQLi](#)
- Python: [Conectar-se a um banco de dados](#)
- Ruby: [Conectar-se a um banco de dados](#)

Finalmente, dependendo de quando seu aplicativo lê as variáveis de ambiente, pode ser necessário reiniciar o servidor de aplicativo nas instâncias em seu ambiente.

Para reiniciar os servidores de aplicativos do ambiente


1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Restart app server(s) (Reiniciar servidores de aplicações).

## Iniciar e conectar-se a uma instância externa do Amazon RDS no EC2 Classic

 Important

O Amazon EC2-Classic atingirá ao fim do suporte padrão em 15 de agosto de 2022. Para evitar interrupções em suas workloads, recomendamos que você migre do Amazon EC2-Classic para uma VPC antes disso. Também solicitamos que você não inicie nenhum lançamento de recurso da AWS no Amazon EC2-Classic no futuro e, em vez disso, use a Amazon VPC. Para obter mais informações, consulte [Migrar do EC2-Classic para uma VPC](#) e a postagem do blog [EC2-Classic Networking is Retiring - Here's How to Prepare \(O EC2-Classic está se descontinuando - Veja como se preparar\)](#).

Se você usar o EC2-Classic (não a VPC) com o AWS Elastic Beanstalk, o procedimento será ligeiramente alterado devido às diferenças no funcionamento dos grupos de segurança. No EC2 Classic, as instâncias de banco de dados não podem usar grupos de segurança do EC2, por isso elas têm um grupo de segurança de banco de dados que funciona apenas com o Amazon RDS.

É possível adicionar regras a um grupo de segurança do banco de dados que permitam acesso de entrada a partir de grupos de segurança do EC2. Entretanto, você não pode anexar um grupo de segurança do banco de dados ao grupo do Auto Scaling associado ao seu ambiente. Para evitar a criação de uma dependência entre o grupo de segurança do banco de dados e seu ambiente, você deve criar um terceiro grupo de segurança no Amazon EC2. Depois, você precisa adicionar uma regra no grupo de segurança do banco de dados para conceder acesso de entrada ao novo grupo

de segurança. Por último, você deve atribuí-lo ao grupo do Auto Scaling no ambiente do Elastic Beanstalk.

### Note

- Se estiver começando com um banco de dados que foi criado pelo Elastic Beanstalk e posteriormente desacoplado de um ambiente do Beanstalk, você pode pular o primeiro grupo de etapas e continuar com as etapas em Para criar um no grupo de segurança da instância intermediário.
- Se planeja usar o banco de dados que você desacopla para um ambiente de produção, verifique se o tipo de armazenamento usado pelo banco de dados é adequado para sua workload. Para obter mais informações, consulte [DB Instance Storage](#) (Armazenamento de instâncias de banco de dados) e [Modifying a DB instance](#) (Modificar uma instância de banco de dados) no Amazon RDS User Guide (Guia do usuário do Amazon RDS).

Como executar uma instância do RDS no EC2-Classic (não na VPC)

1. Abra o [console de gerenciamento do RDS](#).
2. Escolha Criar banco de dados.
3. Continue no assistente. Preste atenção nos valores que você inserir para as seguintes opções:
  - Master Username
  - Senha mestre
4. Quando chegar em Configure advanced settings (Definir configurações avançadas), para as configurações de Network and Security (Rede e segurança), selecione o seguinte:
  - VPC: **Not in VPC**. Se essa opção não estiver disponível, sua conta talvez não suporte o [EC2-Classic](#), ou você pode ter escolhido um [tipo de instância que só está disponível na VPC](#).
  - Zona de disponibilidade: **No Preference**
  - Grupos de segurança de banco de dados: **Create new Security Group**
5. Configure as opções restantes e selecione Create database (Criar banco de dados). Preste atenção nos valores que você inserir para as seguintes opções:
  - Database Name
  - Database Port

No EC2-Classic, sua instância de banco de dados terá um grupo de segurança do banco de dados, em vez de um grupo de segurança da VPC. Você não pode anexar um grupo de segurança do banco de dados ao ambiente do Elastic Beanstalk. Em vez disso, você precisa criar um novo grupo de segurança ao qual você pode autorizar acesso à instância de banco de dados e que pode anexar ao seu ambiente. Vamos nos referir a ele como security group de ligação e o nomearemos como **webapp-bridge**.

Para criar um security group de ligação

1. Abra o [console do Amazon EC2](#).
2. Escolha Security Groups na seção Network & Security do painel de navegação à esquerda.
3. Escolha Create Security Group (Criar grupo de segurança).
4. Em Security group name (Nome do grupo de segurança), digite **webapp-bridge**.
5. Em Description (Descrição), digite **Provide access to DB instance from Elastic Beanstalk environment instances..**
6. Para VPC, deixe a padrão selecionada.
7. Escolha Create (Criar)

Em seguida, modifique o security group anexado à sua instância de banco de dados para permitir o tráfego de entrada do security group de ligação.


Como modificar as regras de ingresso no grupo de segurança da sua instância do RDS

1. Abra o [console do Amazon RDS](#).
2. Escolha Databases (Bancos de dados).
3. Escolha o nome da sua instância de banco de dados para visualizar detalhes.
4. Na seção Connectivity, (Conectividade) em Security (Segurança), o grupo de segurança associado à instância de banco de dados é exibido. Abra o link para visualizar o grupo de segurança no console do Amazon EC2.
5. Nos detalhes do security group, defina Connection Type como EC2 Security Group.
6. Defina EC2 Security Group Name com o nome do security group de ligação que você criou.
7. Escolha Authorize.

Em seguida, adicione o security group de ligação ao seu ambiente em execução. Esse procedimento exige que todas as instâncias em seu ambiente sejam reprovadas com o security group adicional anexado.

Para adicionar um security group ao seu ambiente

- Faça um dos seguintes procedimentos:
  - Como adicionar um grupo de segurança usando o console do Elastic Beanstalk
    - a. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
    - b. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 **Note**

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

  - c. No painel de navegação, escolha Configuration (Configuração).
  - d. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar).
  - e. Em EC2 security groups (Grupos de segurança do EC2), escolha o grupo de segurança a ser anexado às instâncias, além do grupo de segurança da instância que o Elastic Beanstalk criará.
  - f. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.
  - g. Leia o aviso e escolha Confirm.
- Para adicionar um grupo de segurança usando um [arquivo de configuração](#), use o exemplo de arquivo [securitygroup-addexisting.config](#).

Em seguida, passe as informações de conexão para o seu ambiente usando as propriedades de ambiente. Quando você [adiciona uma instância de banco de dados ao ambiente](#) com o console do Elastic Beanstalk, o Elastic Beanstalk usa propriedades de ambiente, como RDS\_HOSTNAME, para passar informações de conexão à aplicação. Você pode usar as mesmas propriedades para usar o mesmo código de aplicação com as instâncias de banco de dados integradas e externas. Ou, alternativamente, você pode escolher seus próprios nomes de propriedade.

## Para configurar as propriedades de ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

### Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Na seção Environment Properties, defina as variáveis que seu aplicativo lê para construir uma string de conexão. Para a compatibilidade com ambientes que têm uma instância RDS integrada, use o seguinte:
  - RDS\_DB\_NAME: o DB Name (Nome do banco de dados) que está no console do Amazon RDS.
  - RDS\_USERNAME: o Master Username (Nome do usuário principal) que você insere ao adicionar o banco de dados ao ambiente.
  - RDS\_PASSWORD: a Master Password (Senha principal) que você insere ao adicionar o banco de dados ao ambiente.
  - RDS\_HOSTNAME: o Endpoint da instância de banco de dados exibida no console do Amazon RDS.
  - RDS\_PORT: a Port (Porta) que está no console do Amazon RDS.

### Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	<input type="text" value="ebdb"/> ✕
RDS_HOSTNAME	<input type="text" value="webapp-db.jxzc b5mpaniu.us-wes"/> ✕
RDS_PORT	<input type="text" value="5432"/> ✕
RDS_USERNAME	<input type="text" value="webapp-admin"/> ✕
<input type="text" value="RDS_PASSWORD"/>	<input type="text" value="kUj5uKxmWDMYc403"/> +

[Cancel](#) [Apply](#)

## 6. Escolha Aplicar

Se você ainda não programou sua aplicação para ler as propriedades de ambiente e criar uma sequência de conexão, consulte os seguintes tópicos específicos de cada linguagem para obter instruções:


- Java SE: [Conectar a um banco de dados \(plataformas Java SE\)](#)
- Java com Tomcat: [Conectar a um banco de dados \(plataformas Tomcat\)](#)
- Node.js: [Conectar-se a um banco de dados](#)
- .NET: [Conectar-se a um banco de dados](#)
- PHP: [Conexão com banco de dados com um PDO ou MySQLi](#)
- Python: [Conectar-se a um banco de dados](#)
- Ruby: [Conectar-se a um banco de dados](#)

Finalmente, dependendo de quando seu aplicativo lê as variáveis de ambiente, pode ser necessário reiniciar o servidor de aplicativo nas instâncias em seu ambiente.



Para reiniciar os servidores de aplicações do seu ambiente

1. Abra o [console do Elastic Beanstalk](#) e, na lista Regions (Regiões), selecione a sua Região da AWS.
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

 Note

Se você tiver muitos ambientes, use a barra de pesquisa para filtrar a lista de ambientes.

3. Escolha Actions (Ações) e depois Restart app server(s) (Reiniciar servidores de aplicações).

## Armazenando as credenciais do Amazon RDS em AWS Secrets Manager

AWS Secrets Manager ajuda você a melhorar sua postura de segurança, fornecendo a capacidade de armazenar e recuperar credenciais criptografadas. Armazenar as credenciais no Secrets Manager evita um possível comprometimento por qualquer pessoa que possa inspecionar sua aplicação ou os componentes relacionados. Seu código pode fazer uma chamada em runtime para o serviço Secrets Manager para recuperar credenciais dinamicamente. O Secrets Manager também oferece atributos como componentes de cache secreto do lado do cliente para linguagens de runtime, que incluem Python, Go e Java.

Para obter mais informações, consulte os seguintes tópicos no Guia do usuário do AWS Secrets Manager:

- [Como o Amazon RDS usa o AWS Secrets Manager](#)
- [Criação de um segredo de banco de dados do AWS Secrets Manager](#)
- [Recuperar segredos do AWS Secrets Manager](#)

## Limpar uma instância externa do Amazon RDS

Quando você conecta uma instância externa do Amazon RDS ao ambiente do Elastic Beanstalk, a instância de banco de dados depende do ciclo de vida do ambiente e não é excluída quando o ambiente é terminado. Para garantir que as informações pessoais que você tenha armazenado na instância de banco de dados não sejam retidas desnecessariamente, exclua todos os registros que não deseja mais. Como alternativa, exclua a instância do banco de dados.

## Usar o Elastic Beanstalk com o Amazon S3

O Amazon Simple Storage Service (Amazon S3) fornece armazenamento de dados resiliente e tolerante a falhas.

O Elastic Beanstalk cria um bucket do Amazon S3 denominado `elasticbeanstalk-region-account-id` para cada região em que você cria ambientes. O Elastic Beanstalk usa esse bucket para armazenar objetos, por exemplo, arquivos de configuração temporários, necessários para o bom funcionamento da aplicação.

O Elastic Beanstalk não ativa a criptografia padrão para o bucket do Amazon S3 criado por ele. Isso significa que, por padrão, os objetos são armazenados descriptografados no bucket (e podem ser acessados somente por usuários autorizados). Algumas aplicações exigem que todos os objetos sejam criptografados quando são armazenados: em um disco rígido, em um banco de dados, etc. (também conhecida como criptografia em repouso). Se você tiver esse requisito, poderá configurar os buckets da sua conta para a criptografia padrão. Para obter mais detalhes, consulte [Criptografia padrão do Amazon S3 para buckets do S3](#) no Guia do usuário do Amazon Simple Storage Service.

### Conteúdo do bucket do Amazon S3 do Elastic Beanstalk

A tabela a seguir lista alguns objetos armazenados pelo Elastic Beanstalk no bucket do Amazon S3 `elasticbeanstalk-*`. A tabela também mostra quais objetos devem ser excluídos manualmente. Para evitar custos de armazenamento desnecessários, e para garantir que as informações pessoais não sejam retidas, exclua manualmente esses objetos quando não forem mais necessários.

Objeto	Quando é armazenado?	Quando é excluído?
<a href="#">Versões do aplicativo</a>	Quando você cria um ambiente ou implanta o código da aplicação em um ambiente existente, o Elastic Beanstalk armazena uma versão da aplicação no Amazon S3 e associa-a ao ambiente.	Durante a exclusão do aplicativo e de acordo com o <a href="#">Ciclo de vida da versão</a> .
<a href="#">Pacotes de origem</a>	Quando você faz upload de uma nova versão da aplicação usando o console do Elastic Beanstalk ou a CLI do EB, o Elastic Beanstalk armazena uma cópia dela no	Manualmente. Ao excluir uma versão do aplicativo, você pode escolher Delete versions from Amazon S3 (Excluir versões do Amazon S3) e também excluir o

Objeto	Quando é armazenado?	Quando é excluído?
	Amazon S3 e a define como o pacote de origem do ambiente.	pacote de origem relacionado. Para obter mais detalhes, consulte <a href="#">Gerenciar versões de aplicativos</a> .
<a href="#">Plataformas personalizadas</a>	Quando uma plataforma personalizada é criada, o Elastic Beanstalk armazena temporariamente dados relacionados no Amazon S3.	Após a conclusão bem-sucedida da criação da plataforma personalizada.
<a href="#">Arquivos de log</a>	É possível solicitar ao Elastic Beanstalk para recuperar arquivos de log de instâncias (logs finais ou em pacote) e armazená-los no Amazon S3. Você também pode habilitar a alternância de logs e configurar o ambiente para publicá-los no Amazon S3 automaticamente depois de alternados.	Logs finais e em pacote: 15 minutos depois de serem criados.  Logs alternados: manualmente.
<a href="#">Configurações salvas</a>	Manualmente.	Manualmente.

## Excluir objetos no bucket do Amazon S3 do Elastic Beanstalk

Quando você encerra um ambiente ou exclui uma aplicação, o Elastic Beanstalk exclui a maioria dos objetos relacionados do Amazon S3. Para minimizar os custos de armazenamento de um aplicativo em execução, exclua rotineiramente objetos dos quais seu aplicativo não precisa. Além disso, preste atenção aos objetos que você precisa excluir manualmente, conforme listado em [Conteúdo do bucket do Amazon S3 do Elastic Beanstalk](#). Para que as informações privadas não sejam retidas desnecessariamente, exclua esses objetos quando não precisar mais deles.

- Exclua as versões de aplicativos que você não pretende mais usar em seu aplicativo. Ao excluir uma versão da aplicação, você pode selecionar Delete versions from Amazon S3 (Excluir versões do Amazon S3) para também excluir o pacote de origem relacionado: uma cópia do código-fonte da aplicação e os arquivos de configurações dos quais o Elastic Beanstalk fez upload para o Amazon S3 quando você implantou uma aplicação ou fez upload de uma versão da aplicação. Para saber como excluir uma versão do aplicativo, consulte [Gerenciar versões de aplicativos](#).

- Excluir logs alternados dos quais não precisa. Como alternativa, baixe-os ou mova-os para o Amazon S3 Glacier para uma análise mais aprofundada.
- Exclua as configurações salvas que você não usará mais em nenhum ambiente.

## Excluir o bucket do Amazon S3 do Elastic Beanstalk

Quando o Elastic Beanstalk cria um bucket, ele também cria uma política de bucket que se aplica ao novo bucket. Esta política atende a dois propósitos:

- Permitir que os ambientes escrevam no bucket.
- Evitar a exclusão acidental do bucket.

Devido à política aplicada pelo Elastic Beanstalk aos buckets que cria para seus ambientes, você não tem permissão para excluir esses buckets, a menos que exclua deliberadamente primeiro a política do bucket. É possível a política do bucket na seção Permissões das propriedades do bucket no console do Amazon S3.

### Aviso

Se você excluir um bucket que o Elastic Beanstalk criou em sua conta e ainda tiver aplicações existentes e ambientes em execução na região correspondente, as aplicações poderão parar de funcionar corretamente. Por exemplo:

- Quando um ambiente é expandido, o Elastic Beanstalk deve ser capaz de encontrar a versão da aplicação do ambiente no bucket do Amazon S3 e usá-lo para iniciar novas instâncias do Amazon EC2.
- Quando uma plataforma personalizada é criada, o Elastic Beanstalk usa armazenamento temporário do Amazon S3 durante o processo de criação.

Recomendamos excluir objetos desnecessários específicos do bucket do Amazon S3 do Elastic Beanstalk em vez de excluir o bucket inteiro.

Como excluir um bucket de armazenamento do Elastic Beanstalk (console)

O procedimento geral para excluir um bucket do S3 também está descrito em [Para excluir um bucket do S3](#) no Guia do usuário do Amazon S3. Como estamos excluindo um bucket criado pelo Elastic

Beanstalk no procedimento a seguir, incluímos etapas adicionais para excluir primeiro a política do bucket.

1. Abra o [console do Amazon S3](#).
2. Abra a página de armazenamento do bucket do Elastic Beanstalk escolhendo o nome do bucket.
3. Escolha a guia Permissions (Permissões).
4. Escolha Bucket Policy.
5. Escolha Delete (Excluir).
6. Volte para a página principal do console do Amazon S3 e, depois, selecione o bucket de armazenamento do Elastic Beanstalk.
7. Escolha Excluir bucket.
8. Confirme se deseja excluir o bucket inserindo o nome do bucket no campo de texto e, em seguida, escolha Excluir bucket.

## Usar o Elastic Beanstalk com Amazon VPC

É possível usar uma [Amazon Virtual Private Cloud](#) (Amazon VPC) para criar uma rede segura para a aplicação do Elastic Beanstalk e recursos relacionados da AWS. Ao criar seu ambiente, você escolhe a VPC, as sub-redes e os security groups que serão usados nas suas instâncias de aplicativo e no load balancer. Você pode usar qualquer configuração de VPC que desejar, desde que ela atenda aos seguintes requisitos.

### Requisitos da VPC

- Acesso à Internet: as instâncias podem ter acesso à Internet por meio de um dos métodos a seguir.
- Sub-rede pública: as instâncias têm um endereço IP público e usam um gateway da Internet para acessar a Internet.
- Sub-rede privada: as instâncias usam um dispositivo NAT para acessar a Internet.

#### Note

Se você configurar [VPC endpoints](#) em sua VPC para se conectar aos serviços `elasticbeanstalk` e `elasticbeanstalk-health`, o acesso à Internet será opcional e será necessário somente se o aplicativo precisar especificamente dele. Sem VPC endpoints, sua VPC deve ter acesso à Internet.

A VPC padrão que o Elastic Beanstalk configura para você fornece acesso à internet.

O Elastic Beanstalk não é compatível com as configurações de proxy, como HTTPS\_PROXY para configurar um proxy da Web.

- NTP: as instâncias no ambiente do Elastic Beanstalk usam o Network Time Protocol (NTP) para sincronizar o relógio do sistema. Caso as instâncias não consigam se comunicar na porta UDP 123, o relógio pode estar fora de sincronia, causando problemas de relatórios de integridade do Elastic Beanstalk. Certifique-se de que os security groups da VPC e as ACLs de rede permitem o tráfego de entrada e saída de UDP na porta 123 para evitar esses problemas.

O repositório [elastic-beanstalk-samples](#) fornece modelos do AWS CloudFormation que podem ser usados para criar uma VPC para uso com seus ambientes do Elastic Beanstalk.

Para criar recursos com um modelo do AWS CloudFormation

1. Clone o repositório de exemplos ou faça download de um modelo usando os links no arquivo [README](#).
2. Abra o [console do AWS CloudFormation](#).
3. Selecione Create stack.
4. Escolha Upload a template to Amazon S3.
5. Escolha Upload file e faça upload do arquivo de modelo da sua máquina local.
6. Escolha Next e siga as instruções para criar uma pilha com os recursos no modelo.

Quando a criação da pilha for concluída, marque a guia Outputs (Saídas) para encontrar o ID da VPC e os IDs de sub-rede. Use-os para configurar a VPC no novo assistente de ambiente [categoria de configuração de rede](#).

Tópicos

- [VPC pública](#)
- [VPC pública/privada](#)
- [VPC privada](#)
- [Exemplo: Iniciar uma aplicação do Elastic Beanstalk em uma VPC com bastion hosts](#)
- [Exemplo: Iniciar um Elastic Beanstalk em uma VPC com o Amazon RDS](#)

- [Usar o Elastic Beanstalk com VPC endpoints](#)

## VPC pública

Modelo do AWS CloudFormation – [vpc-public.yaml](#)

Configurações (balanceamento de carga)

- Visibilidade do balanceador de carga: público
- Sub-redes do balanceador de carga: as duas sub-redes públicas
- IP público da instância: habilitado
- Sub-redes da instância: as duas sub-redes públicas
- Grupos de segurança de instância: adicione o grupo de segurança padrão

Configurações (instância única)

- Sub-redes da instância: uma das sub-redes públicas
- Grupos de segurança de instância: adicione o grupo de segurança padrão

Um layout somente público da VPC inclui uma ou mais sub-redes públicas, um gateway da Internet e um grupo de segurança padrão que permite o tráfego entre recursos na VPC. Quando você cria um ambiente na VPC, o Elastic Beanstalk cria recursos adicionais que variam dependendo do tipo de ambiente.

Recursos da VPC

- Instância única: o Elastic Beanstalk cria um grupo de segurança para a instância da aplicação que permite o tráfego na porta 80 da Internet e atribui à instância um IP elástico para fornecer um endereço IP público. O nome de domínio do ambiente é resolvido para o endereço IP público da instância.
- Balanceamento de carga: o Elastic Beanstalk cria um grupo de segurança para o balanceador de carga que permite o tráfego na porta 80 da Internet e um grupo de segurança para as instâncias da aplicação que permite o tráfego do grupo de segurança do balanceador de carga. O nome de domínio do ambiente é resolvido para o nome de domínio público do load balancer.

Isso é semelhante ao modo como o Elastic Beanstalk gerencia a rede quando você usa a VPC padrão. A segurança em uma sub-rede pública depende do balanceador de carga e dos grupos de segurança da instância criados pelo Elastic Beanstalk. É a configuração mais econômica, porque não requer um gateway NAT.

## VPC pública/privada

Modelo do AWS CloudFormation – [vpc-privatepublic.yaml](#)

Configurações (balanceamento de carga)

- Visibilidade do balanceador de carga: público
- Sub-redes do balanceador de carga: as duas sub-redes públicas
- IP público da instância: desabilitado
- Sub-redes da instância: as duas sub-redes privadas
- Grupos de segurança de instância: adicione o grupo de segurança padrão

Para mais segurança, adicione sub-redes privadas à sua VPC para criar um layout público-privado. Esse layout requer um load balancer e um gateway NAT nas sub-redes públicas e permite que você execute instâncias do aplicativo, banco de dados e quaisquer outros recursos nas sub-redes privadas. Instâncias em sub-redes privadas podem se comunicar apenas com a Internet por meio do load balancer e do gateway NAT.

## VPC privada

Modelo do AWS CloudFormation – [vpc-private.yaml](#)

Configurações (balanceamento de carga)

- Visibilidade do balanceador de carga: privado
- Sub-redes do balanceador de carga: as duas sub-redes privadas
- IP público da instância: desabilitado
- Sub-redes da instância: as duas sub-redes privadas
- Grupos de segurança de instância: adicione o grupo de segurança padrão

Para aplicativos internos que não devem ter acesso da Internet, é possível executar tudo em sub-redes privadas e configurar o load balancer para ser interno (altere Load balancer visibility



(Visibilidade do load balancer) para Internal (Interno)). Esse modelo cria uma VPC sem sub-redes públicas e sem um gateway da Internet. Use esse layout para aplicativos que só devem ser acessados a partir da mesma VPC ou de uma VPN conectada.

## Executar um ambiente do Elastic Beanstalk em uma VPC privada

Quando você cria o ambiente do Elastic Beanstalk em uma VPC privada, o ambiente não tem acesso à Internet. Sua aplicação pode precisar de acesso ao serviço Elastic Beanstalk ou a outros serviços. Seu ambiente pode usar relatórios de integridade avançada, e, nesse caso, as instâncias do ambiente enviam informações de integridade para o serviço de integridade avançada. E o código do Elastic Beanstalk em instâncias do ambiente envia tráfego para outros produtos da AWS e outro tráfego para endpoints que não são da AWS (por exemplo, para baixar pacotes de dependência para a aplicação). Veja a seguir algumas etapas que podem ser necessárias para execução nesse caso, a fim de garantir que seu ambiente funcione corretamente.

- Configurar VPC endpoints para o Elastic Beanstalk: o Elastic Beanstalk e seu serviço de integridade aprimorada são compatíveis com VPC endpoints, que garantem que o tráfego para esses serviços permaneça dentro da rede da Amazon e não exija acesso à Internet. Para obter mais informações, consulte [the section called “VPC endpoints”](#).
- Configurar endpoints da VPC para serviços adicionais - As instâncias do Elastic Beanstalk enviam tráfego para vários outros produtos da AWS em seu nome: Amazon Simple Storage Service (Amazon S3), Amazon Simple Queue Service (Amazon SQS), AWS CloudFormation e Amazon CloudWatch Logs. É necessário configurar VPC endpoints para esses serviços também. Para obter informações detalhadas sobre VPC endpoints, incluindo links por serviço, consulte [VPC endpoints](#) no Guia do usuário da Amazon VPC.

### Note

Alguns produtos da AWS, incluindo o Elastic Beanstalk, são compatíveis com endpoints da VPC em um número limitado de regiões da AWS. Ao criar sua solução de VPC privada, verifique se o Elastic Beanstalk e os outros serviços dependentes mencionados aqui são compatíveis com endpoints da VPC na região da AWS escolhida.

- Fornecer uma imagem privada do Docker: em um ambiente [Docker](#), o código nas instâncias do ambiente pode tentar extrair sua imagem configurada do Docker da Internet durante a criação do ambiente e falhar. Para evitar essa falha, [crie uma imagem personalizada do Docker](#) no ambiente ou use uma imagem do Docker armazenada no [Amazon Elastic Container Registry](#) (Amazon ECR) e [configure um VPC endpoint para o serviço Amazon ECR](#).

- Habilitar nomes DNS - O código do Elastic Beanstalk em instâncias do ambiente envia tráfego para todos os produtos da AWS usando seus endpoints públicos. Para garantir que esse tráfego passe, escolha a opção Enable DNS name (Habilitar nome DNS) ao configurar todos os VPC endpoints de interface. Isso adiciona uma entrada DNS em sua VPC, que mapeia o endpoint público de serviço para o VPC endpoint de interface.

#### Important

Se sua VPC não for privada e tiver acesso público à Internet e se Enable DNS name (Habilitar nome DNS) estiver desabilitado para qualquer VPC endpoint, o tráfego para o respectivo serviço será enviado pela Internet pública. Isso provavelmente não é o que você pretende. É fácil detectar esse problema com uma VPC privada, pois impede que esse tráfego passe e você receba erros. No entanto, com uma VPC voltada para o público, você não recebe nenhuma indicação.

- Incluir dependências de aplicações: se sua aplicação tiver dependências como pacotes de tempo de execução de linguagem, ele poderá tentar fazer download deles e instalá-los da Internet durante a criação do ambiente e falhar. Para evitar essa falha, inclua todos os pacotes de dependência no pacote de origem do aplicativo.
- Usar uma versão da plataforma atual: certifique-se de que seu ambiente use uma versão da plataforma lançada em 24 de fevereiro de 2020 ou posterior. Especificamente, use uma versão da plataforma que foi lançada a partir de uma destas duas atualizações: [Linux Update 2020-02-28](#), [Windows Update 2020-02-24](#).

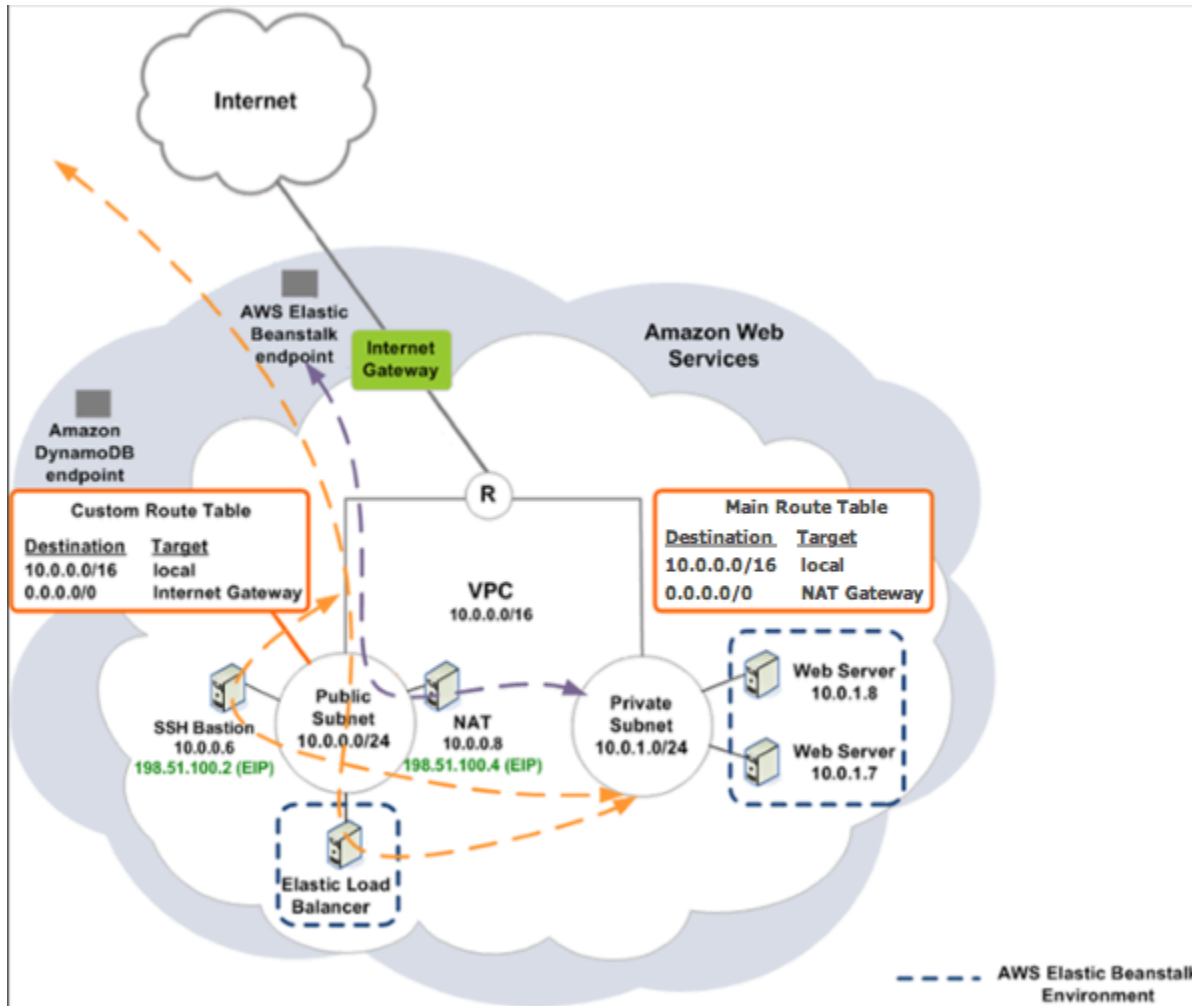
#### Note

O motivo da necessidade de uma versão atualizada da plataforma é que as versões mais antigas tinham um problema que impedia que as entradas DNS criadas pela opção Enable DNS name (Habilitar nome DNS) funcionassem corretamente para o Amazon SQS.

## Exemplo: Iniciar uma aplicação do Elastic Beanstalk em uma VPC com bastion hosts

Se as instâncias do Amazon EC2 estiverem localizadas em uma sub-rede privada, você não poderá se conectar a elas remotamente. Para se conectar às suas instâncias, configure os servidores bastion na sub-rede pública para atuarem como proxies. Por exemplo, configure encaminhadores

da porta SSH ou gateways RDP na sub-rede pública para atuar como proxy do tráfego que vai para os servidores de banco de dados de sua própria rede. Esta seção fornece um exemplo de como criar uma VPC com uma sub-rede pública e privada. As instâncias estão localizadas na sub-rede privada, e o bastion host, o gateway NAT e o load balancer estão localizados na sub-rede pública. A infraestrutura terá aparência semelhante ao diagrama a seguir.



Para implantar uma aplicação do Elastic Beanstalk em uma VPC usando um bastion host, conclua as etapas descritas nas subseções a seguir.

## Etapas

- [Criar uma VPC com uma sub-rede privada e pública](#)
- [Criar e configurar o grupo de segurança do bastion host](#)
- [Atualizar o grupo de segurança da instância](#)
- [Criar um bastion host](#)

## Criar uma VPC com uma sub-rede privada e pública

Complete todos os procedimentos em [VPC pública/privada](#). Ao implantar a aplicação, especifique um par de chaves do Amazon EC2 para as instâncias para que você possa se conectar a elas remotamente. Para obter mais informações sobre como especificar o par de chaves da instância, consulte [Instâncias do Amazon EC2 para seu ambiente do Elastic Beanstalk](#).

## Criar e configurar o grupo de segurança do bastion host

Crie um grupo de segurança para o bastion host e adicione regras que permitam tráfego SSH de entrada da Internet e de saída para a sub-rede privada que contém as instâncias do Amazon EC2.

Para criar o security group do bastion host

1. Abra o console da Amazon VPC em <https://console.aws.amazon.com/vpc/>.
2. No painel de navegação, selecione Grupos de segurança.
3. Escolha Create Security Group.
4. Na caixa de diálogo Create Security Group (Criar grupo de segurança), informe o seguinte e escolha Yes, Create (Sim, criar).

Name tag (Tag do nome) (opcional)

Informe uma tag de nome para o grupo de segurança.

Group name

Informe o nome do grupo de segurança.

Descrição

Informe uma descrição para o grupo de segurança.

VPC

Selecione a VPC.

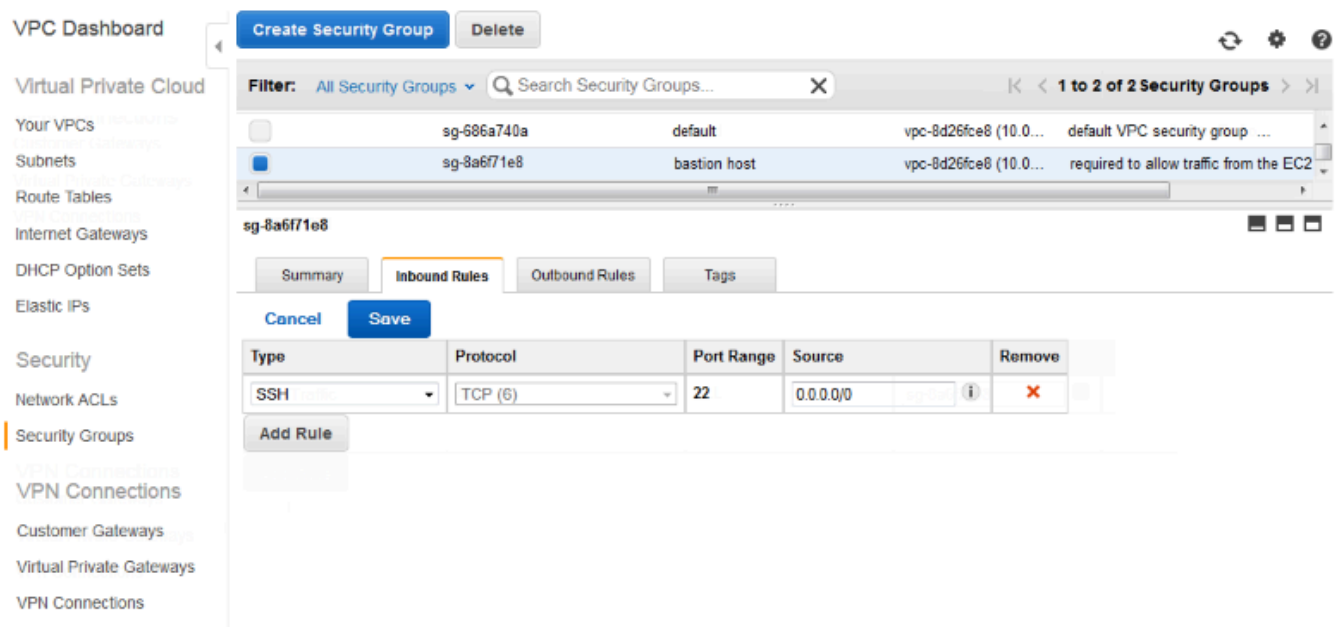
O grupo de segurança é criado e exibido na página Security Groups (Grupos de segurança). Ele tem uma ID (por exemplo, sg-xxxxxxx). Talvez você precise ativar a coluna de Group ID (ID de grupo) clicando em Show/Hide (Mostrar/ocultar) no canto superior direito da página.

Para configurar o security group do bastion host.

1. Na lista de security groups, marque a caixa de seleção do security group que acabou de criar para o bastion host.
2. Na guia Inbound Rules, escolha Edit.
3. Se necessário, escolha Add another rule.
4. Se o bastion host é uma instância do Linux, em Type, selecione SSH.

Se o bastion host é uma instância do Windows, em Tipo, selecione RDP.

5. Insira o intervalo de CIDR de origem desejado no campo Source e escolha Save.



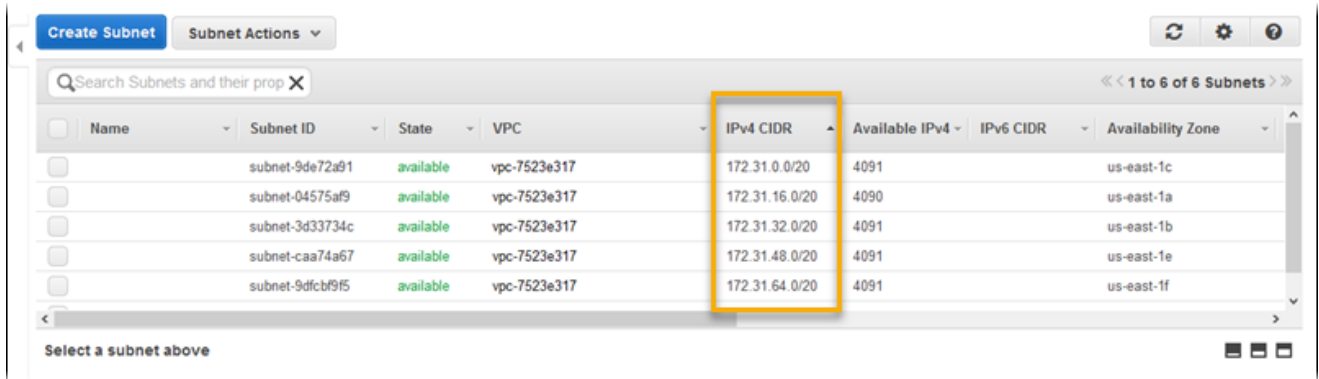
6. Na guia Inbound Rules (Regras de saída), escolha Edit (Editar).
7. Se necessário, escolha Add another rule.
8. Em Type (Tipo), selecione o tipo de dados que você especificou para a regra de entrada.
9. No campo Source (Origem), insira a variação de CIDR da sub-rede dos hosts na sub-rede privada do VPC.

Para encontrá-lo:

- a. Abra o console da Amazon VPC em <https://console.aws.amazon.com/vpc/>.
- b. No painel de navegação, escolha Sub-redes.
- c. Observe o valor em IPv4 CIDR para cada Zona de disponibilidade na qual você tem hosts que você deseja que o bastion host seja uma ponte para.

**Note**

Se você tiver hosts em várias zonas de disponibilidade, crie uma regra de saída para cada uma dessas zonas de disponibilidade.



Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
	subnet-9de72a91	available	vpc-7523e317	172.31.0.0/20	4091		us-east-1c
	subnet-04575af9	available	vpc-7523e317	172.31.16.0/20	4090		us-east-1a
	subnet-3d33734c	available	vpc-7523e317	172.31.32.0/20	4091		us-east-1b
	subnet-caa74a67	available	vpc-7523e317	172.31.48.0/20	4091		us-east-1e
	subnet-9dfcb9f5	available	vpc-7523e317	172.31.64.0/20	4091		us-east-1f

10. Escolha Save (Salvar).

## Atualizar o grupo de segurança da instância

Por padrão, o security group que você criou para suas instâncias não permite o tráfego de entrada. Embora o Elastic Beanstalk modifique o grupo padrão das instâncias para permitir o tráfego SSH, é necessário modificar o grupo de segurança da instância personalizada para permitir o tráfego RDP, caso suas instâncias sejam do Windows.

Para atualizar o security group da instância para RDP

1. Na lista de security groups, marque a caixa de seleção do security group da instância.
2. Na guia Entrada, escolha Editar.
3. Se necessário, escolha Add another rule.
4. Insira os valores a seguir e escolha Save.

Type

RDP

Protocolo

TCP

## Port Range (Intervalo de portas)

3389

## Origem

Insira o ID do security group do bastion host (por exemplo, sg-8a6f71e8) e escolha Save.

## Criar um bastion host

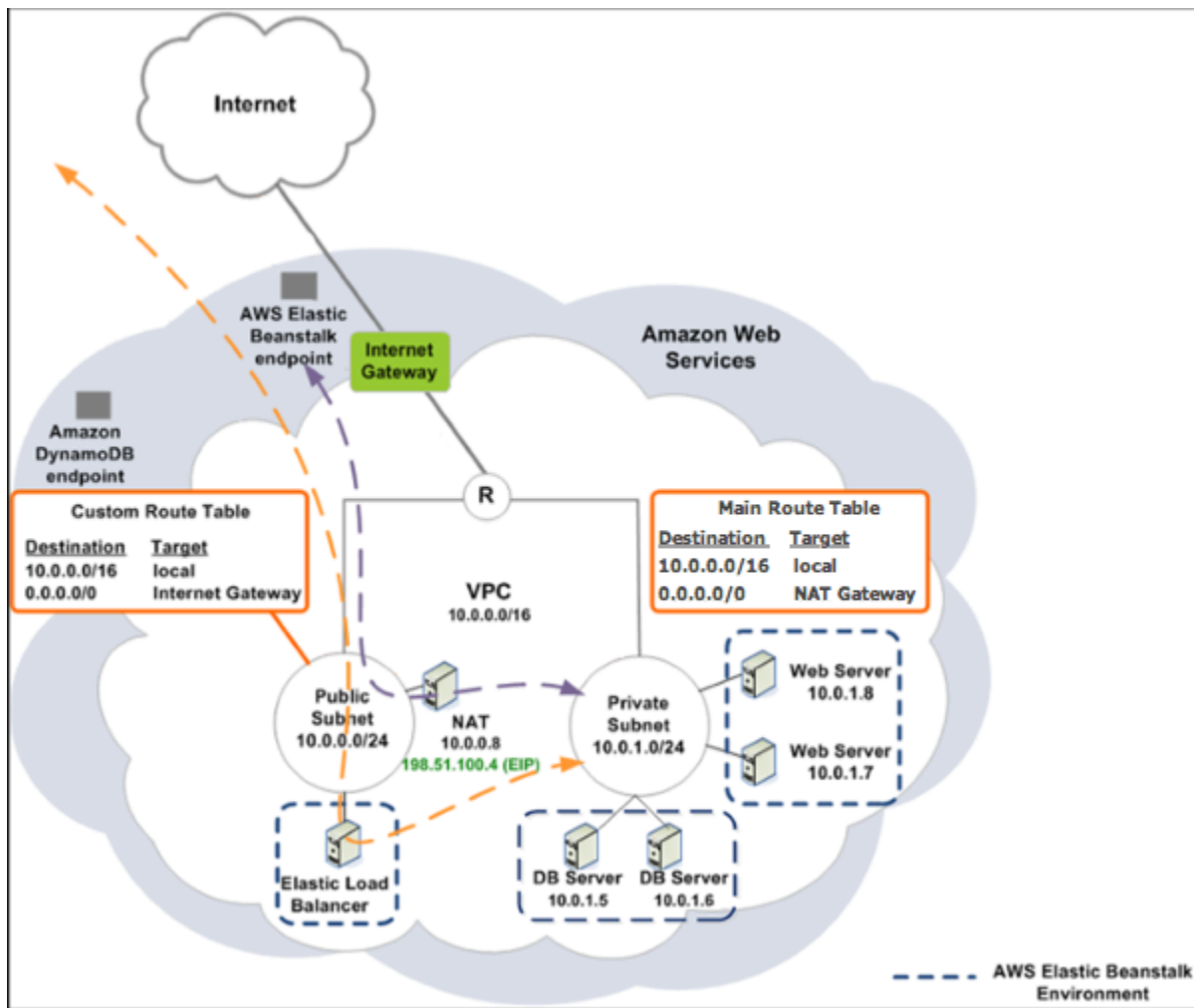
Para criar um bastion host, execute uma instância do Amazon EC2 na sua sub-rede pública que atua como um bastion host.

Para obter mais informações sobre como configurar um bastion host para instâncias do Windows na sub-rede privada, consulte [Controlar o acesso de rede para instâncias do EC2 usando um servidor bastion](#).

Para obter mais informações sobre como configurar um bastion host para instâncias Linux na sub-rede privada, consulte [Conexão segura com instâncias Linux executadas em uma VPC privada da Amazon](#).

## Exemplo: Iniciar um Elastic Beanstalk em uma VPC com o Amazon RDS

Esta seção mostra como implantar uma aplicação do Elastic Beanstalk com o Amazon RDS em uma VPC usando um gateway NAT. A infraestrutura terá aparência semelhante ao diagrama a seguir.



### Note

Se você não usou uma instância de banco de dados com seu aplicativo antes, tente [adicionar uma a um ambiente de teste](#) e [conectar-se a uma instância de banco de dados externa](#) antes de adicionar uma configuração de VPC à combinação.

Criar uma VPC com uma sub-rede privada e pública

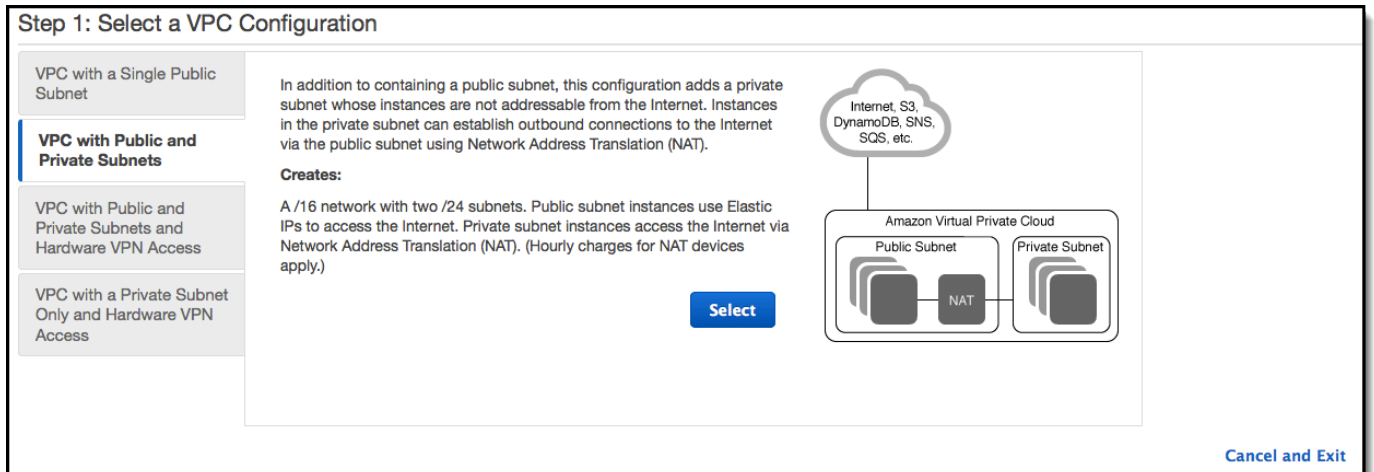
É possível usar o [console da Amazon VPC](#) para criar uma VPC.

Para criar uma VPC

1. Faça login no [console da Amazon VPC](#).



2. No painel de navegação, escolha VPC Dashboard (Painel da VPC). Em seguida, selecione Create VPC (Criar VPC).
3. Escolha VPC with Public and Private Subnets (VPC com sub-redes públicas e privadas) e selecione Select (Selecione).



4. O balanceador de carga do Elastic Load Balancing e as instâncias do Amazon EC2 devem estar na mesma zona de disponibilidade, para que se comuniquem entre si. Escolha mesma zona de disponibilidade em cada lista Availability Zone (Zona de disponibilidade).

**Step 2: VPC with Public and Private Subnets**

IPv4 CIDR block: 10.0.0.0/16 (65531 IP addresses available)

IPv6 CIDR block:  No IPv6 CIDR Block  
 Amazon provided IPv6 CIDR block

VPC name:

Public subnet's IPv4 CIDR: 10.0.0.0/24 (251 IP addresses available)

Availability Zone:

Public subnet name:

Private subnet's IPv4 CIDR: 10.0.1.0/24 (251 IP addresses available)

Availability Zone:

Private subnet name:

You can add more subnets after AWS creates the VPC.

Specify the details of your NAT gateway (NAT gateway rates apply). [Use a NAT instance instead](#)

Elastic IP Allocation ID:

Service endpoints

Enable DNS hostnames:  Yes  No

Hardware tenancy:

Enable ClassicLink:  Yes  No

Cancel and Exit

5. Escolha um endereço IP elástico no gateway NAT.
6. Escolha Create VPC (Criar VPC).

O assistente começa a criar a VPC, as sub-redes e o gateway da Internet. Ele também atualiza a tabela de rotas principal e cria uma tabela de rotas personalizada. Por fim, o assistente cria um gateway NAT na sub-rede pública.

### Note

Você pode optar por ativar uma instância NAT na sub-rede pública, em vez de um gateway NAT. Para obter mais informações, consulte [Cenário 2: VPC com sub-redes privadas e públicas \(NAT\)](#) no Guia do usuário da Amazon VPC.

7. Depois que a VPC for criada com êxito, você receberá uma ID de VPC. Você precisa desse valor na próxima etapa. Para visualizar o ID da VPC, escolha Your VPCs (Suas VPCs) no painel à esquerda do [console da Amazon VPC](#).



## Criar um grupo de sub-redes de banco de dados

Um grupo de sub-redes de banco de dados de uma VPC é um conjunto de sub-redes (geralmente privadas) que você pode designar para suas instâncias de Banco de Dados de backend do RDS. Cada grupo de sub-redes de banco de dados deve ter no mínimo uma sub-rede para cada zona de disponibilidade de uma determinada região da AWS. Para saber mais, consulte [Criar uma sub-rede na VPC](#).

### Criar um grupo de sub-redes de banco de dados

1. Abra o [console do Amazon RDS](#).
2. No painel de navegação, escolha Subnet groups (Grupos de sub-redes).
3. Escolha Create DB Subnet Group (Criar grupo de sub-redes de banco de dados).
4. Clique em Name (Nome) e digite o nome do seu grupo de sub-redes de banco de dados.

5. Clique em Description (Descrição) e descreva o seu grupo de sub-redes de banco de dados.
6. Em VPC, selecione o ID da VPC criada por você.
7. Na seção Add subnets (Adicionar sub-redes), clique em Add all the subnets related to this VPC (Adicionar todas as sub-redes relacionadas a essa VPC).

**Add subnets**  
Add subnet(s) to this subnet group. You may add subnets one at a time below or add all the subnets related to this VPC. You may make additions/edits after this group is created. A minimum of 2 subnets is required.

**Add all the subnets related to this VPC**

Availability zone  
Choose an availability zone

Subnet  
Choose a subnet **Add subnet**

**Subnets in this subnet group (4)**

Availability zone	Subnet ID	CIDR block	Action
us-east-2c	subnet-da3408ae	10.0.1.0/24	<b>Remove</b>
us-east-2c	subnet-db3408af	10.0.0.0/24	<b>Remove</b>
us-east-2b	subnet-4f195024	10.0.2.0/24	<b>Remove</b>
us-east-2a	subnet-fe064f95	10.0.3.0/24	<b>Remove</b>

**Cancel** **Create**

8. Quando terminar, escolha Create (Criar imagem).

Seu novo grupo aparece na lista de grupos de sub-redes de banco de dados do console do Amazon RDS. Clique nele para ver detalhes, por exemplo, todas as sub-redes associadas a esse grupo, no painel de detalhes, na parte inferior da página.

## Implantar no Elastic Beanstalk

Depois de configurar a VPC, é possível criar o ambiente dentro dela e implantar a aplicação no Elastic Beanstalk. Você pode fazer isso usando o console do Elastic Beanstalk ou pode usar a AWS toolkits, a AWS CLI, a EB CLI ou a API do Elastic Beanstalk. Se usar o console do Elastic Beanstalk, você só precisará fazer upload do arquivo `.war` ou `.zip` e selecionar as configurações da VPC dentro do assistente. O Elastic Beanstalk acaba criando o ambiente dentro da VPC e implantando a aplicação. Como alternativa, você pode usar a AWS toolkits, a AWS CLI, a EB CLI ou a API do Elastic Beanstalk para implantar a aplicação. Para isso, você precisa definir as configurações de

opção da VPC em um arquivo de configuração e implantar esse arquivo com o pacote de origem. Este tópico apresenta instruções para ambos os métodos.

### Implantar com o console do Elastic Beanstalk

O console do Elastic Beanstalk orienta durante a criação do novo ambiente dentro da VPC. Você precisa fornecer um arquivo `.war` (para aplicações Java) ou um arquivo `.zip` (para todas as outras aplicações). Na página VPC Configuration (Configuração de VPC) do assistente de ambiente do Elastic Beanstalk, você deve selecionar o seguinte:

#### VPC

Selecione a VPC.

#### VPC security group (Grupo de segurança da VPC)

Selecione o security group da instância que você criou acima.

#### ELB visibility (Visibilidade do ELB)

Selecione `External` se o load balancer precisar estar disponível publicamente, ou selecione `Internal` se o load balancer precisar estar disponível apenas dentro da VPC.

Selecione as sub-redes do load balancer e as instâncias EC2. Não se esqueça de selecionar a sub-rede pública para o balanceador de carga e a sub-rede privada para as instâncias do Amazon EC2. Por padrão, o assistente de criação da VPC cria a sub-rede pública na sub-rede privada `10.0.0.0/24` em `10.0.1.0/24`.

É possível visualizar os IDs escolhendo Subnets (Sub-redes) no [console da Amazon VPC](#).

The screenshot shows the AWS VPC Dashboard. On the left, the 'Subnets' link is highlighted with a red box. The main area displays a table of subnets:

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	Availability Zone
	subnet-3ba3c75e	available	vpc-f56cff91	172.31.64.0/20	4091	us-east-1a
	subnet-ec18feb4	available	vpc-f56cff91	172.31.16.0/20	4089	us-east-1d

Below the table, the details for 'subnet-ec18feb4' are shown under the 'Summary' tab:

<b>Subnet ID:</b>	subnet-ec18feb4	<b>Availability Zone:</b>	us-east-1d
<b>IPv4 CIDR:</b>	172.31.16.0/20	<b>Route table:</b>	rtb-4f0f472b
<b>IPv6 CIDR:</b>		<b>Network ACL:</b>	acl-ca059fae
<b>State:</b>	available	<b>Default subnet:</b>	yes
<b>VPC:</b>	vpc-f56cff91	<b>Auto-assign Public IP:</b>	yes
<b>Available IPs:</b>	4089	<b>Auto-assign IPv6 address:</b>	no

Implantação com a AWS toolkits, a EB CLI, AWS CLI ou API.

Ao implementar a aplicação no Elastic Beanstalk usando a AWS toolkits, a EB CLI, a AWS CLI ou a API, você pode especificar as configurações de opção da VPC em um arquivo e implantá-lo com o pacote de fonte. Consulte [Personalização avançada de ambiente com arquivos de configuração \(.ebextensions\)](#) para obter mais informações.

Quando você atualiza as configurações de opção, é necessário especificar pelo menos:

- VPCId: contém o ID da VPC.
- Subnets: contém o ID da sub-rede de grupos de Auto Scaling. Neste exemplo, é o ID da sub-rede privada.
- ELBSubnets: contém o ID da sub-rede do balanceador de carga. Neste exemplo, é o ID da sub-rede pública.
- SecurityGroups: contém o ID dos grupos de segurança.
- DBSubnets: contém o ID das sub-redes de banco de dados.

#### Note

Ao usar sub-redes de banco de dados, você precisa criar sub-redes adicionais na sua VPC para incluir todas as zonas de disponibilidade na região da AWS.

Como opção, especifique também as seguintes informações:

- **ELBScheme**: especifique `internal` para criar um balanceador de carga interno dentro da sua VPC para que a aplicação do Elastic Beanstalk não seja acessada de fora dela.

Veja a seguir um exemplo das configurações de opção que podem ser usadas ao implantar a aplicação do Elastic Beanstalk em uma VPC. Para obter mais informações sobre configurações de opção de VPC (inclusive exemplos de como especificá-las, valores padrão e valores válidos), consulte a tabela de namespace `aws:ec2:vpc` em [Opções de configuração](#).

```
option_settings:
 - namespace: aws:autoscaling:launchconfiguration
 option_name: EC2KeyName
 value: ec2keypair

 - namespace: aws:ec2:vpc
 option_name: VPCId
 value: vpc-170647c

 - namespace: aws:ec2:vpc
 option_name: Subnets
 value: subnet-4f195024

 - namespace: aws:ec2:vpc
 option_name: ELBSubnets
 value: subnet-fe064f95

 - namespace: aws:ec2:vpc
 option_name: DBSubnets
 value: subnet-fg148g78

 - namespace: aws:autoscaling:launchconfiguration
 option_name: InstanceType
 value: m1.small

 - namespace: aws:autoscaling:launchconfiguration
 option_name: SecurityGroups
 value: sg-7f1ef110
```

**Note**

Ao usar sub-redes de bancos de dados, certifique-se de ter sub-redes na sua VPC para incluir todas as zonas de disponibilidade na região da AWS.

## Usar o Elastic Beanstalk com VPC endpoints

Um VPC endpoint permite conectar de forma privada sua VPC aos serviços compatíveis da AWS e aos serviços de VPC endpoint habilitados pelo AWS PrivateLink, sem exigir um gateway da Internet, um dispositivo NAT, uma conexão VPN ou uma conexão do AWS Direct Connect.

As instâncias na sua VPC não exigem que endereços IP públicos se comuniquem com recursos no serviço. O tráfego entre a sua VPC e os outros serviços não sai da rede da Amazon. Para obter mais informações sobre VPC endpoints, consulte [VPC endpoints](#) no Guia do usuário da Amazon VPC.

O AWS Elastic Beanstalk oferece suporte ao AWS PrivateLink, que fornece conectividade privada ao serviço do Elastic Beanstalk e elimina a exposição do tráfego à Internet pública. Para habilitar que a aplicação envie solicitações para o Elastic Beanstalk usando o AWS PrivateLink, configure um tipo de VPC endpoint conhecido como endpoint da VPC de interface (endpoint de interface). Para obter mais informações, consulte [Endpoints da VPC da interface \(AWS PrivateLink\)](#) no Guia do usuário da Amazon VPC.

**Note**

O AWS PrivateLink oferece suporte ao Elastic Beanstalk e endpoints da VPC de interface em um número limitado de regiões da AWS. Estamos trabalhando para estender o suporte a mais regiões da AWS no futuro próximo.

## Configurar um VPC endpoint para o Elastic Beanstalk

Para criar o VPC endpoint de interface para o serviço do Elastic Beanstalk na VPC, siga o procedimento [Criar um endpoint de interface](#). Em Service Name (Nome do serviço), escolha `com.amazonaws.region.elasticbeanstalk`.

Se a VPC estiver configurada com acesso público à Internet, a aplicação ainda poderá acessar o Elastic Beanstalk pela Internet usando o endpoint público `elasticbeanstalk.region.amazonaws.com`. É possível impedir isso habilitando Enable DNS

name (Habilitar nome DNS) durante a criação do endpoint (true por padrão). Isso adiciona uma entrada DNS em sua VPC, que mapeia o endpoint público de serviço para o VPC endpoint de interface.

## Configurar um VPC endpoint para integridade avançada

Se você habilitou os [relatórios de integridade avançada](#) para seu ambiente, será possível configurar as informações de integridade avançada para serem enviadas pelo AWS PrivateLink também. As informações de integridade aprimorada são enviadas pelo daemon `healthd`, um componente do Elastic Beanstalk nas instâncias do seu ambiente, para um serviço de integridade aprimorada separado do Elastic Beanstalk. Para criar um VPC endpoint de interface para esse serviço na VPC, siga o procedimento [Criar um endpoint de interface](#). Em Service Name (Nome do serviço), escolha `com.amazonaws.region.elasticbeanstalk-health`.

### Important

O daemon `healthd` envia informações de integridade avançada para o endpoint público, `elasticbeanstalk-health.region.amazonaws.com`. Se a VPC estiver configurada com acesso público à Internet e `Enable DNS name` (Habilitar nome DNS) estiver desabilitado para o VPC endpoint, as informações de integridade avançada serão enviadas pela Internet pública. Isso provavelmente não é sua intenção ao configurar um VPC endpoint de integridade avançada. Certifique-se de que `Enable DNS name` (Habilitar nome DNS) esteja habilitado (true por padrão).

## Usar VPC endpoints em uma VPC privada

Uma VPC privada ou uma sub-rede privada em uma VPC não tem acesso público à Internet. Convém executar o ambiente do Elastic Beanstalk em uma [VPC privada](#) e configurar VPC endpoints de interface para segurança aprimorada. Nesse caso, lembre-se de que o ambiente pode tentar se conectar à Internet por outros motivos, além de entrar em contato com o serviço do Elastic Beanstalk. Para saber mais sobre como executar um ambiente em uma VPC privada, consulte [the section called “Executar um ambiente do Elastic Beanstalk em uma VPC privada”](#).

## Usar políticas de endpoint para controlar o acesso com VPC endpoints

Por padrão, um VPC endpoint permite acesso total ao serviço ao qual está associado. Ao criar ou modificar um endpoint, é possível anexar uma política de endpoint a ele.



Uma política de endpoint é uma política de recurso do AWS Identity and Access Management (IAM) que controla o acesso do endpoint ao serviço especificado. A política de endpoint é específica ao endpoint. Ela é separada de outras políticas do IAM de usuário ou instância que seu ambiente possa ter e não as substitui. Para obter detalhes sobre como criar e usar políticas de VPC endpoint, consulte [Controlar o acesso aos serviços com VPC endpoints](#) no Guia do usuário da Amazon VPC.

O exemplo a seguir nega a todos os usuários a permissão para encerrar um ambiente por meio do VPC endpoint e permite acesso total a todas as outras ações.

```
{
 "Statement": [
 {
 "Action": "*",
 "Effect": "Allow",
 "Resource": "*",
 "Principal": "*"
 },
 {
 "Action": "elasticbeanstalk:TerminateEnvironment",
 "Effect": "Deny",
 "Resource": "*",
 "Principal": "*"
 }
]
}
```

#### Note

No momento, somente o serviço do Elastic Beanstalk principal oferece suporte para anexar uma política de endpoint ao VPC endpoint. O serviço de integridade avançada não oferece suporte a políticas de endpoint.

# Configurar sua máquina de desenvolvimento para uso com o Elastic Beanstalk

Esta página mostra como configurar sua máquina local para o desenvolvimento de uma aplicação do AWS Elastic Beanstalk. Ela abrange a estrutura de pastas, o controle da origem e as ferramentas da CLI.

## Tópicos

- [Criar uma pasta de projeto](#)
- [Configurar o controle da origem](#)
- [Configurar um repositório remoto](#)
- [Instalar a EB CLI](#)
- [Instalar o AWS CLI](#)

## Criar uma pasta de projeto

Crie uma pasta para o seu projeto. Você pode armazenar a pasta em qualquer lugar no disco local, desde que tenha permissão para lê-lo e gravar nele. A criação de uma pasta em sua pasta de usuário é aceitável. Se você pretende trabalhar em vários aplicativos, crie suas pastas de projeto dentro de outra pasta com um nome como `workspace` ou `projects` para manter tudo organizado:

```
workspace/
|-- my-first-app
`-- my-second-app
```

O conteúdo da pasta do projeto varia de acordo com o contêiner da web ou a estrutura que o seu aplicativo utiliza.

### Note

Evite pastas e caminhos com as aspas simples (') ou duplas (") no nome da pasta ou em qualquer elemento do caminho. Alguns comandos do Elastic Beanstalk falham quando são executados em uma pasta com um desses caracteres no nome.

## Configurar o controle da origem

Configure o controle da origem para se proteger contra a exclusão acidental de arquivos ou código na pasta do projeto e para ter uma maneira de reverter as alterações que quebram seu projeto.

Se você não tiver um sistema de controle de origem, considere o Git, uma opção gratuita e fácil de usar que se integra bem à Interface de Linhas de Comando (CLI) do Elastic Beanstalk. Visite a [página inicial do Git](#) para instalá-lo.

Siga as instruções no site do Git para instalar e configurar o Git e, em seguida, execute `git init` na pasta do projeto para configurar um repositório local:

```
~/workspace/my-first-app$ git init
Initialized empty Git repository in /home/local/username/workspace/my-first-app/.git/
```

Na medida em que for adicionando conteúdo à pasta do projeto e atualizando o conteúdo, confirme as alterações no repositório Git:

```
~/workspace/my-first-app$ git add default.jsp
~/workspace/my-first-app$ git commit -m "add default JSP"
```

A cada vez que confirma, você cria um snapshot do seu projeto que você pode restaurar posteriormente se algo der errado. Para obter mais informações sobre comandos e fluxos de trabalho do Git, consulte a [documentação do Git](#).

## Configurar um repositório remoto

E se o seu disco rígido falhar ou se você desejar trabalhar em seu projeto em um computador diferente? Para fazer backup de seu código-fonte online e acessá-lo de qualquer computador, configure um repositório remoto para o qual você possa enviar as suas confirmações.

O AWS CodeCommit permite criar um repositório privado na nuvem AWS. O CodeCommit é o [nível gratuito da AWS](#) para até cinco usuários do AWS Identity and Access Management (IAM) em sua conta. Para obter mais detalhes de preço, consulte [Preços do AWS CodeCommit](#).

Consulte o [Guia do usuário do AWS CodeCommit](#) para obter instruções sobre como configurar.

O GitHub é outra opção popular para armazenar o código do seu projeto online. Ele permite que você crie um repositório online público gratuitamente e também é compatível com repositórios privados mediante uma cobrança mensal. Cadastre-se no GitHub em [github.com](#).

Depois de criar um repositório remoto para o seu projeto, anexe-o ao seu repositório local com `git remote add`:

```
~/workspace/my-first-app$ git remote add origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/my-repo
```

## Instalar a EB CLI

Use a [CLI do EB](#) para gerenciar seus ambientes do Elastic Beanstalk e monitorar a integridade a partir da linha de comando. Consulte [Instalar a EB CLI](#) para obter instruções de instalação.

Por padrão, a CLI do EB empacota tudo na pasta do projeto e faz upload no Elastic Beanstalk como um pacote de origem. Quando você usa o Git e a EB CLI juntos, pode evitar que os arquivos de classe criados se comprometam com a origem com `.gitignore` e impedir que os arquivos de origem sejam implantados com `.ebignore`.

Você também pode [configurar a EB CLI para implantar um artefato de criação](#) (um arquivo WAR ou ZIP), em vez do conteúdo da pasta do projeto.

## Instalar o AWS CLI

O AWS Command Line Interface (AWS CLI) é um cliente unificado para produtos AWS que fornece comandos para todas as operações públicas de API. Esses comandos são de nível mais baixo do que os fornecidos pela EB CLI, por isso costumam ser necessários mais comandos para realizar uma operação com a AWS CLI. Por outro lado, a AWS Command Line Interface permite que você trabalhe com qualquer aplicação ou ambiente em execução na sua conta sem configurar um repositório em sua máquina local. Use a AWS CLI para criar scripts que simplificam ou automatizam tarefas operacionais.

Para obter mais informações sobre serviços suportados e para fazer download da AWS Command Line Interface, consulte [AWS Command Line Interface](#).

# Usar a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB)

A EB CLI é uma interface da linha de comando do AWS Elastic Beanstalk que fornece comandos interativos que simplificam a criação, a atualização e o monitoramento de ambientes em um repositório local. Use a CLI do EB como parte do ciclo diário de desenvolvimento e testes como uma alternativa ao console do Elastic Beanstalk.

## Note

A versão atual da EB CLI tem um conjunto básico diferente de comandos que as versões anteriores à versão 3.0. Se você estiver usando uma versão mais antiga, consulte [Migrar para o CLI 3 do EB](#) e o [CodeCommit](#) para obter informações de migração.

Depois de [instalar a CLI do EB](#) e configurar um diretório de projeto, é possível criar ambientes com um único comando:

```
~/my-app$ eb create my-env
```

O código-fonte da CLI do EB é um projeto de código aberto. Ele reside no repositório do [aws/aws-elastic-beanstalk-cli](#) GitHub. É possível participar relatando problemas, fazendo sugestões e enviando solicitações pull. Valorizamos as suas contribuições. Para um ambiente em que você só pretende usar a CLI do EB como está, recomendamos instalar usando um dos scripts de configuração da CLI do EB, conforme detalhado em [the section called “Instalar a CLI do EB usando scripts de configuração”](#).

Anteriormente, o Elastic Beanstalk era compatível com uma CLI separada que permitia acesso direto às operações de API chamadas [CLI da API do Elastic Beanstalk](#). Isso foi substituído pela [AWS CLI](#), que fornece a mesma funcionalidade, mas para todas as APIs de produtos da AWS.

Com a AWS CLI, você tem acesso direto à API do Elastic Beanstalk. A AWS CLI é excelente para desenvolvimento de scripts, mas não é tão fácil de usar na linha de comando em virtude do número de comandos necessários para executar e do número de parâmetros em cada comando. Por exemplo, a criação de um ambiente requer uma série de comandos:

```
~$ aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
```

```
~$ aws elasticbeanstalk create-application-version --application-name my-application --
version-label v1 --source-bundle S3Bucket=DOC-EXAMPLE-BUCKET,S3Key=php-proxy-sample.zip
~$ aws elasticbeanstalk create-environment --cname-prefix my-cname --application-name
my-app --version-label v1 --environment-name my-env --solution-stack-name "64bit
Amazon Linux 2015.03 v2.0.0 running Ruby 2.2 (Passenger Standalone)"
```

Para informações sobre instalação da EB CLI, configuração de um repositório e como trabalhar com ambientes, consulte os seguintes tópicos:

## Tópicos

- [Instalar a EB CLI](#)
- [Configurar a EB CLI](#)
- [Gerenciar ambientes do Elastic Beanstalk com a CLI do EB](#)
- [Usar o EB CLI com o AWS CodeBuild](#)
- [Usar a EB CLI com o Git](#)
- [Usar o EB CLI com o AWS CodeCommit](#)
- [Usar a EB CLI para monitorar a integridade do ambiente](#)
- [Gerenciar vários ambientes do Elastic Beanstalk como um grupo com a CLI do EB](#)
- [Solução de problemas com a EB CLI](#)
- [Referência de comandos da EB CLI](#)
- [EB CLI 2.6 \(desativada\)](#)
- [Interface de Linhas de Comando da API do Elastic Beanstalk \(retirada\)](#)

## Instalar a EB CLI

A AWS Elastic Beanstalk Command Line Interface (EB CLI) é um cliente de linha de comando que você pode usar para criar, configurar e gerenciar ambientes do Elastic Beanstalk. Para obter mais informações sobre a CLI do EB, consulte [EB CLI](#).

## Tópicos

- [Instalar a CLI do EB usando scripts de configuração](#)
- [Instalar a CLI do EB manualmente](#)

## Instalar a CLI do EB usando scripts de configuração

A maneira mais fácil e recomendada de instalar a CLI do EB é usar os [scripts de configuração da CLI do EB](#) disponíveis no GitHub. Use os scripts para instalar a CLI do EB no Linux, macOS ou Windows. Os scripts instalam a CLI do EB e suas dependências, incluindo o Python e o `pip`. Os scripts também criam um ambiente virtual para a CLI do EB. Para obter instruções de instalação, consulte o repositório [aws/aws-elastic-beanstalk-cli-setup](#) no GitHub.

## Instalar a CLI do EB manualmente

Para instalar a CLI do EB, recomendamos o uso dos [scripts de configuração da CLI do EB](#). Se os scripts de configuração não forem compatíveis com o ambiente de desenvolvimento, instale manualmente a CLI do EB.

O método de distribuição principal da CLI do EB no Linux, macOS e Windows é o `pip`. Esse é um gerenciador de pacote para o Python que fornece uma maneira fácil de instalar, atualizar e remover pacotes Python e suas dependências. Para o macOS, você também pode obter a versão mais recente da CLI do EB com Homebrew.

## Observações sobre compatibilidade

A CLI do EB foi desenvolvida em Python e requer o Python versão 3.11 ou posterior.

Recomendamos o uso dos [scripts de configuração da CLI do EB](#) para instalar a CLI do EB e suas dependências. Se você instalar manualmente a CLI do EB, pode ser difícil gerenciar conflitos de dependência em seu ambiente de desenvolvimento.

A EB CLI e a [AWS Command Line Interface](#) (AWS CLI) compartilham uma dependência no pacote [botocore](#) Python. Devido a uma alteração no `botocore`, diferentes versões dessas duas ferramentas da CLI dependem de versões diferentes de `botocore`.

As versões mais recentes das duas CLIs são compatíveis. Se você precisar usar uma versão anterior, consulte a tabela a seguir para verificar uma versão compatível a ser usada.

Versão da CLI do EB	Versão compatível da AWS CLI
3.14.5 ou anterior	1.16.9 ou anterior
3.14.6 ou posterior	1.16.11 ou posterior

## Instalar a EB CLI

Se você já tem `pip` e uma versão compatível do Python, use o seguinte procedimento para instalar o EB CLI:

Se você não tem Python nem `pip`, use o procedimento para o sistema operacional que está utilizando.

- [Instalar Python, pip e EB CLI no Linux](#)
- [Instalar a EB CLI no macOS](#)
- [Instalar Python, pip e EB CLI no Windows](#)

Para instalar o EB CLI

1. Execute o comando a seguir da .

```
$ pip install awsebcli --upgrade --user
```

A opção `--upgrade` informa ao `pip` para atualizar os requisitos que já estão instalados. A opção `--user` informa ao `pip` para instalar o programa em um subdiretório do diretório de usuário para evitar a modificação de bibliotecas usadas pelo seu sistema operacional.

### Note

Caso haja problemas ao tentar instalar a CLI do EB com o `pip`, você poderá [instalar a CLI do EB em um ambiente virtual](#) para isolar a ferramenta e suas dependências, ou usar uma versão do Python diferente da que você normalmente usa.

2. Adicione o caminho para o arquivo executável da sua variável `PATH`:

- No Linux e macOS:

Linux: `~/.local/bin`

macOS: `~/Library/Python/3.7/bin`

Para modificar a variável `PATH` (Linux, Unix ou macOS):

- a. Encontre o script de perfil do shell em sua pasta de usuário. Se você não tiver certeza de qual shell você tem, execute `echo $SHELL`.



```
$ ls -a ~
. .. .bash_logout .bash_profile .bashrc Desktop Documents Downloads
```

- Bash: `.bash_profile`, `.profile` ou `.bash_login`.
  - Zsh: `.zshrc`
  - Tcsh: `.tcshrc`, `.cshrc` ou `.login`.
- b. Adicione um comando de exportação ao script de perfil. O seguinte exemplo adiciona o caminho representado por `LOCAL_PATH` à variável `PATH` atual.

```
export PATH=LOCAL_PATH:$PATH
```

- c. Carregue o script de perfil descrito na primeira etapa para a sessão atual. O seguinte exemplo carrega o script de perfil representado por `PROFILE_SCRIPT`.

```
$ source ~/PROFILE_SCRIPT
```

- No Windows:

Python 3.7: `%USERPROFILE%\AppData\Roaming\Python\Python37\Scripts`

Versões anteriores do Python: `%USERPROFILE%\AppData\Roaming\Python\Scripts`

Para modificar a variável `PATH` (Windows):

- a. Pressione a tecla Windows e insira **environment variables**.
  - b. Escolha Edit environment variables for your account.
  - c. Selecione `PATH` e, em seguida, Edit (Editar).
  - d. Adicione caminhos ao campo Variable value, separados por ponto e vírgula. Por exemplo: `C:\item1\path;C:\item2\path`
  - e. Selecione OK duas vezes para aplicar as novas configurações.
  - f. Feche todas as janelas do prompt de comando em execução e abra novamente uma janela do prompt de comando.
3. Verifique se a EB CLI foi instalada corretamente executando `eb --version`.

```
$ eb --version
EB CLI 3.14.8 (Python 3.7)
```

A CLI do EB é atualizada regularmente para adicionar as funcionalidades compatíveis com [os recursos mais recentes do Elastic Beanstalk](#). Para atualizar para a versão mais recente da EB CLI, execute o comando de instalação novamente.

```
$ pip install awsebcli --upgrade --user
```

Se você precisar desinstalar a EB CLI, use `pip uninstall`.

```
$ pip uninstall awsebcli
```

## Instalar Python, pip e EB CLI no Linux

A CLI do EB requer o Python 2.7, 3.4 ou posterior. Se sua distribuição não foi fornecida com o Python, ou foi fornecida com uma versão anterior, instale o Python antes de instalar o pip e a CLI do EB.

Para instalar o Python 3.7 no Linux

1. Determine se o Python já está instalado.

```
$ python --version
```

### Note

Se sua distribuição do Linux for fornecida com o Python, poderá ser necessário instalar o pacote do desenvolvedor Python para ter os cabeçalhos e as bibliotecas necessários para compilar extensões e instalar a EB CLI. Use o gerenciador de pacote para instalar o pacote do desenvolvedor (geralmente chamado `python-dev` ou `python-devel`).

2. Se o Python 2.7 ou posterior não estiver instalado, instale o Python 3.7 usando o gerenciador do pacote de distribuição. O comando e o nome do pacote variam:

- Nos derivados do Debian, como o Ubuntu, use APT.

```
$ sudo apt-get install python3.7
```

- No Red Hat e derivados, use yum.

```
$ sudo yum install python37
```

- No SUSE e derivados, use o `zypper`.

```
$ sudo zypper install python3-3.7
```

3. Para verificar se o Python foi instalado corretamente, abra um terminal ou shell e execute o comando a seguir.

```
$ python3 --version
Python 3.7.3
```

Instale o `pip` usando o script fornecido pelo Python Packaging Authority e, em seguida, instale a EB CLI.

Para instalar o `pip` e a CLI do EB

1. Faça download do script de instalação do [pypa.io](https://bootstrap.pypa.io).

```
$ curl -O https://bootstrap.pypa.io/get-pip.py
```

O script faz o download e instala a versão mais recente do pacote chamado `pip` e outro pacote necessário de `setuptools`.

2. Execute o script com Python.

```
$ python3 get-pip.py --user
Collecting pip
 Downloading pip-8.1.2-py2.py3-none-any.whl (1.2MB)
Collecting setuptools
 Downloading setuptools-26.1.1-py2.py3-none-any.whl (464kB)
Collecting wheel
 Downloading wheel-0.29.0-py2.py3-none-any.whl (66kB)
Installing collected packages: pip, setuptools, wheel
Successfully installed pip setuptools wheel
```

Invocar o Python versão 3 diretamente usando o comando `python3` em vez de `python` garante que o `pip` seja instalado no local apropriado, mesmo que uma versão anterior do Python esteja presente em seu sistema.

3. Adicione o caminho do executável, `~/local/bin`, à variável `PATH`.

Para modificar a variável PATH (Linux, Unix ou macOS):

- a. Encontre o script de perfil do shell em sua pasta de usuário. Se você não tiver certeza de qual shell você tem, execute `echo $SHELL`.

```
$ ls -a ~
. .. .bash_logout .bash_profile .bashrc Desktop Documents Downloads
```

- Bash: `.bash_profile`, `.profile` ou `.bash_login`.
  - Zsh: `.zshrc`
  - Tcsh: `.tcshrc`, `.cshrc` ou `.login`.
- b. Adicione um comando de exportação ao script de perfil. O seguinte exemplo adiciona o caminho representado por `LOCAL_PATH` à variável PATH atual.

```
export PATH=LOCAL_PATH:$PATH
```

- c. Carregue o script de perfil descrito na primeira etapa para a sessão atual. O seguinte exemplo carrega o script de perfil representado por `PROFILE_SCRIPT`.

```
$ source ~/PROFILE_SCRIPT
```

4. Verifique se o pip está instalado corretamente.

```
$ pip --version
pip 8.1.2 from ~/.local/lib/python3.7/site-packages (python 3.7)
```

5. Use o pip para instalar a EB CLI.

```
$ pip install awsebcli --upgrade --user
```

6. Verifique se a EB CLI está instalada corretamente.

```
$ eb --version
EB CLI 3.14.8 (Python 3.7)
```

Para atualizar para a versão mais recente, execute o comando de instalação novamente.

```
$ pip install awsebcli --upgrade --user
```

## Instalar a EB CLI no macOS

Se usar o gerenciador de pacotes Homebrew, você poderá instalar a CLI do EB usando o comando `brew`. Também é possível instalar o Python e o `pip` e, em seguida, usar o `pip` para instalar a EB CLI.

### Instalar a EB CLI com o homebrew

A versão mais recente da CLI do EB geralmente fica disponível no Homebrew alguns dias depois de aparecer no `pip`.

Para instalar a CLI do EB com o **Homebrew**

1. Verifique se você tem a versão mais recente do Homebrew.

```
$ brew update
```

2. Execute `brew install awsebcli`.

```
$ brew install awsebcli
```

3. Verifique se a EB CLI está instalada corretamente.

```
$ eb --version
EB CLI 3.14.8 (Python 3.7)
```

## Instalar Python, pip, e EB CLI no macOS

Instale a versão mais recente do Python e do `pip` e, em seguida, use-os para instalar a EB CLI.

Para instalar a EB CLI no macOS

1. Faça download e instale o Python da [página de download](#) em [Python.org](#). Usamos a versão 3.7 para demonstrar.

**Note**

A CLI do EB requer o Python 2 versão 2.7 ou o Python 3 no intervalo das versões 3.4 a 3.7.

2. Instale o pip com o script fornecido pela Python Packaging Authority.

```
$ curl -O https://bootstrap.pypa.io/get-pip.py
$ python3 get-pip.py --user
```

3. Use o pip para instalar a EB CLI.

```
$ pip3 install awsebcli --upgrade --user
```

4. Adicione o caminho do executável, ~/Library/Python/3.7/bin, à variável PATH.

Para modificar a variável PATH (Linux, Unix ou macOS):

- a. Encontre o script de perfil do shell em sua pasta de usuário. Se você não tiver certeza de qual shell você tem, execute `echo $SHELL`.

```
$ ls -a ~
. .. .bash_logout .bash_profile .bashrc Desktop Documents Downloads
```

- Bash: `.bash_profile`, `.profile` ou `.bash_login`.
  - Zsh: `.zshrc`
  - Tcsh: `.tcshrc`, `.cshrc` ou `.login`.
- b. Adicione um comando de exportação ao script de perfil. O seguinte exemplo adiciona o caminho representado por `LOCAL_PATH` à variável PATH atual.

```
export PATH=LOCAL_PATH:$PATH
```

- c. Carregue o script de perfil descrito na primeira etapa para a sessão atual. O seguinte exemplo carrega o script de perfil representado por `PROFILE_SCRIPT`.

```
$ source ~/PROFILE_SCRIPT
```

5. Verifique se a EB CLI está instalada corretamente.

```
$ eb --version
EB CLI 3.14.8 (Python 3.7)
```

Para atualizar para a versão mais recente, execute o comando de instalação novamente.

```
$ pip3 install awsebcli --upgrade --user
```

## Instalar Python, pip e EB CLI no Windows

O Python Software Foundation fornece instaladores para Windows que incluem pip.

Para instalar o Python e o **pip** (Windows)

1. Faça download do instalador executável do Python Windows mais recente x86-64 da [página de downloads](#) do [Python.org](#).
2. Execute o executável do instalador do Python que você baixou na etapa anterior.

Selecione as seguintes opções na janela do instalador do Python para configurar as etapas de instalação do EB CLI a seguir.

- a. Escolha adicionar o executável Python ao seu caminho.
- b. Escolha Instalar agora.

### Note

Para obter mais informações sobre as opções de instalação, consulte a página [Usando Python no Windows](#) no site do Python.

O site da documentação fornece um menu suspenso na parte superior da página onde você pode selecionar a versão do Python para a documentação.

O instalador instala o Python em sua pasta de usuário e adiciona seus diretórios executáveis ao caminho do usuário.

Como instalar a AWS CLI com o **pip** (Windows)

1. No menu Iniciar, abra uma janela do prompt de comando.

2. Verifique se o Python e o pip estão instalados corretamente usando os seguintes comandos.

```
C:\Users\myname> python --version
Python 3.11.4
C:\Users\myname> pip --version
pip 23.1.2 from C:\Users\myname\AppData\Local\Programs\Python\Python311\Lib\site-
packages\pip (python 3.11)
```

3. Instale a EB CLI usando o pip.

```
C:\Users\myname> pip install awsebcli --upgrade --user
```

4. Adicione o seguinte caminho executável à variável de ambiente Path em sua conta de usuário do Windows. A localização pode ser diferente dependendo se você instala o Python para um usuário ou para todos os usuários.

```
%USERPROFILE%\AppData\Roaming\Python\Python311\Scripts
```

5. Reinicie um novo shell de comando para que a nova variável Path entre em vigor.
6. Verifique se a EB CLI está instalada corretamente.

```
C:\Users\myname> eb --version
EB CLI 3.14.8 (Python 3.11)
```

Para atualizar para a versão mais recente, execute o comando de instalação novamente.

```
C:\Users\myname> pip install awsebcli --upgrade --user
```

## Instalar a EB CLI em um ambiente virtual

Você pode evitar conflitos de requisitos de versão com outros pacotes pip instalando a EB CLI em um ambiente virtual.

Para instalar a EB CLI em um ambiente virtual

1. Instale o virtualenv com o pip.

```
$ pip install --user virtualenv
```

2. Crie um ambiente virtual.



```
$ virtualenv ~/eb-ve
```

Para usar um executável Python diferente da padrão, use a opção `-p`.

```
$ virtualenv -p /usr/bin/python3.7 ~/eb-ve
```

3. Ative o ambiente virtual.

Linux, Unix ou macOS

```
$ source ~/eb-ve/bin/activate
```

Windows

```
$ %USERPROFILE%\eb-ve\Scripts\activate
```

4. Instale a EB CLI.

```
(eb-ve)~$ pip install awsebcli --upgrade
```

5. Verifique se a EB CLI está instalada corretamente.

```
$ eb --version
EB CLI 3.14.8 (Python 3.7)
```

Use o comando `deactivate` para sair do ambiente virtual. Sempre que iniciar uma nova sessão, execute o comando de ativação novamente.

Para atualizar para a versão mais recente, execute o comando de instalação novamente.

```
(eb-ve)~$ pip install awsebcli --upgrade
```

## Configurar a EB CLI

Após a [instalação da CLI do EB](#), você está pronto para configurar o diretório do projeto e a CLI do EB executando `eb init`.

O exemplo a seguir mostra as etapas de configuração na execução de `eb init` pela primeira vez em uma pasta do projeto chamada `eb`.

Para inicializar um projeto da EB CLI

1. Primeiro, a EB CLI solicita que seja selecionada uma região. Digite o número correspondente à região que você deseja usar e pressione Enter.

```
~/eb $ eb init
Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : Europe (Ireland)
5) eu-central-1 : Europe (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
...
(default is 3): 3
```

2. Em seguida, forneça sua chave de acesso e chave secreta para que a EB CLI possa gerenciar os recursos para você. As chaves de acesso são criadas no console do AWS Identity and Access Management. Se você não tiver chaves, consulte [Como obter credenciais de segurança?](#) na Referência geral da Amazon Web Services.

```
You have not yet set up your credentials or your credentials are incorrect.
You must provide your credentials.
(aws-access-id): AKIAJOUAASEXAMPLE
(aws-secret-key): 5ZRIrtTM4ciIAvd4EXAMPLEDtm+PiPSzpoK
```

3. Uma aplicação no Elastic Beanstalk é um recurso que contém um conjunto de versões da aplicação (origem), ambientes e configurações salvas associadas a uma única aplicação Web. Cada vez que você implanta o seu código-fonte no Elastic Beanstalk usando a CLI do EB, uma nova versão da aplicação é criada e adicionada à lista.

```
Select an application to use
1) [Create new Application]
(default is 1): 1
```

4. O nome do aplicativo padrão é o nome da pasta na qual você executa `eb init`. Insira um nome que descreva o projeto.

```
Enter Application Name
(default is "eb"): eb
Application eb has been created.
```

5. Selecione uma plataforma que corresponda à linguagem ou framework usados para desenvolver seu aplicativo Web. Se você ainda não começou a desenvolver um aplicativo, escolha uma plataforma de seu interesse. Você verá como iniciar um aplicativo de exemplo rapidamente e poderá alterar essa configuração mais tarde.

```
Select a platform.
1) Node.js
2) PHP
3) Python
4) Ruby
5) Tomcat
6) IIS
7) Docker
8) Multi-container Docker
9) GlassFish
10) Go
11) Java
(default is 1): 1
```

6. Escolha yes (sim) para atribuir um par de chaves SSH para as instâncias no ambiente do Elastic Beanstalk. Isso permite que você se conecte diretamente a elas e solucione problemas.

```
Do you want to set up SSH for your instances?
(y/n): y
```

7. Escolha um par de chaves existente ou crie um novo. Para usar `eb init` para criar um novo par de chaves, instale `ssh-keygen` em sua máquina local e disponibilize-o na linha de comando. A CLI do EB registra o novo par de chaves no Amazon EC2 e armazena a chave privada localmente em uma pasta chamada `.ssh` no diretório do usuário.

```
Select a keypair.
1) [Create new KeyPair]
(default is 1): 1
```

Agora, a instalação da EB CLI está configurada e pronta para uso. Consulte [Gerenciar ambientes do Elastic Beanstalk com a CLI do EB](#) para obter instruções sobre como criar e trabalhar com um ambiente do Elastic Beanstalk.

### Configuração avançada

- [Ignorar arquivos que usam .ebignore](#)
- [Usar perfis nomeados](#)
- [Implantar um artefato, em vez da pasta do projeto](#)
- [Definições de configuração e precedência](#)
- [Metadados da instância](#)

## Ignorar arquivos que usam .ebignore

Você pode instruir a EB CLI a ignorar determinados arquivos no diretório do projeto por meio da adição do arquivo `.ebignore` ao diretório. Esse arquivo funciona como um `.gitignore`. Quando você implanta o diretório do projeto no Elastic Beanstalk e cria uma versão da aplicação, a CLI do EB não inclui arquivos especificados pelo `.ebignore` no pacote de origem criado.

Se não houver nenhum `.ebignore`, mas houver um `.gitignore`, a EB CLI ignorará os arquivos especificados no `.gitignore`. Se houver um `.ebignore`, a EB CLI não fará a leitura do `.gitignore`.

Quando `.ebignore` estiver presente, a EB CLI não usará comandos `git` para criar o pacote de origem. Isso significa que a EB CLI ignora os arquivos especificados em `.ebignore` e inclui todos os outros. Especificamente, ela inclui os arquivos de origem não confirmados.

#### Note

No Windows, a adição de um `.ebignore` faz com que a EB CLI siga os links simbólicos e inclua o arquivo vinculado durante a criação de um pacote de origem. Esse é um problema conhecido e será corrigido em uma atualização futura.

## Usar perfis nomeados

Se você armazenar suas credenciais como um perfil nomeado em um arquivo `credentials` ou `config`, poderá usar a opção `--profile` para especificar explicitamente um perfil. Por exemplo, o seguinte comando cria um novo aplicativo usando o perfil `user2`.

```
$ eb init --profile user2
```

Você também pode alterar o perfil padrão, definindo a variável de ambiente `AWS_EB_PROFILE`. Quando essa variável está definida, a EB CLI faz a leitura das credenciais do perfil especificado, em vez de `default` ou `eb-cli`.

Linux, macOS ou Unix

```
$ export AWS_EB_PROFILE=user2
```

Windows

```
> set AWS_EB_PROFILE=user2
```

## Implantar um artefato, em vez da pasta do projeto

Instrua a EB CLI a implantar um arquivo ZIP ou WAR que você gera como parte de um processo de construção separado, adicionando as linhas a seguir ao `.elasticbeanstalk/config.yml` na pasta do projeto.

```
deploy:
 artifact: path/to/buildartifact.zip
```

Se você configurar a CLI do EB no [repositório Git](#) e não confirmar o artefato na fonte, use a opção `--staged` para implantar a compilação mais recente.

```
~/eb$ eb deploy --staged
```

## Definições de configuração e precedência

A EB CLI usa uma cadeia de provedor para procurar credenciais da AWS em vários lugares diferentes, inclusive variáveis de ambiente do sistema ou do usuário e arquivos de configuração locais da AWS.

A EB CLI procura credenciais e definições de configuração na seguinte ordem:

1. Opções da linha de comando: especifica um perfil nomeado usando `--profile` para substituir as configurações padrão.
2. Variáveis de ambiente: `AWS_ACCESS_KEY_ID` e `AWS_SECRET_ACCESS_KEY`.
3. O arquivo de credenciais da AWS: localizado em `~/.aws/credentials` em sistemas Linux e OS X ou em `C:\Users\USERNAME\.aws\credentials` em sistemas Windows. Este arquivo pode conter vários perfis nomeados, além de um perfil padrão.
4. O arquivo de configuração [da AWS CLI](#): localizado `~/.aws/config` em sistemas Linux e OS X ou `C:\Users\USERNAME\.aws\config` em sistemas Windows. O arquivo pode conter um perfil padrão, [perfis nomeados](#) e parâmetros de configuração específicos da AWS CLI para cada um deles.
5. Arquivo de configuração da CLI do EB legado: localizado em `~/.elasticbeanstalk/config` em sistemas Linux e OS X ou `C:\Users\USERNAME\.elasticbeanstalk\config` em sistemas Windows.
6. Credenciais de perfil de instância: essas credenciais podem ser usadas em instâncias do Amazon EC2 com uma função de instância atribuída, e são fornecidas por meio do serviço de metadados do Amazon EC2. O [perfil da instância](#) deve ter permissão para usar o Elastic Beanstalk.

Se o arquivo de credenciais contém um perfil nomeado com o nome "eb-cli", a EB CLI preferirá esse perfil em vez do perfil padrão. Se não for encontrado nenhum perfil, ou se for encontrado um perfil sem permissão para usar o Elastic Beanstalk, a CLI do EB solicitará a entrada de chaves.

## Metadados da instância

Para usar a CLI do EB em uma instância do Amazon EC2, crie uma função que tenha acesso aos recursos necessários e atribua essa função à instância quando ela for executada. Inicie a instância e instale a EB CLI usando o `pip`.

```
~$ sudo pip install awsebcli
```

O `pip` é fornecido pré-instalado no Amazon Linux.

A EB CLI lê as credenciais dos metadados da instância. Para obter mais informações, consulte [Conceder acesso a aplicações executadas em instâncias da Amazon EC2 para recursos da AWS](#) no Guia do usuário do IAM.

# Gerenciar ambientes do Elastic Beanstalk com a CLI do EB

Após [instalar a CLI do EB](#) e [configurar o diretório do projeto](#), você estará pronto para criar um ambiente do Elastic Beanstalk usando a CLI do EB, implantar atualizações de origem e de configuração e extrair logs e eventos.

## Note

A criação de ambientes com a CLI do EB exige uma [função de serviço](#). É possível criar uma função de serviço criando um ambiente no console do Elastic Beanstalk. Se você não tiver uma função de serviço, a CLI do EB tentará criar uma ao executar `eb create`.

O EB CLI retorna um código de saída zero (0) para todos os comandos bem-sucedidos e um código de saída diferente de zero quando encontra algum erro.

Os exemplos a seguir usam uma pasta de projeto vazia denominada `eb` que foi inicializada com a EB CLI para uso com um aplicativo Docker de exemplo.

## Comandos básicos

- [Eb create](#)
- [Eb status](#)
- [Eb health](#)
- [Eb events](#)
- [Eb logs](#)
- [Eb open](#)
- [Eb deploy](#)
- [Eb config](#)
- [Eb terminate](#)

## Eb create

Para criar seu primeiro ambiente, execute [eb create](#) e siga os prompts. Se o diretório do projeto tiver código-fonte, a EB CLI vai empacotá-lo e implantá-lo em seu ambiente. Caso contrário, um aplicativo de exemplo será usado.

```
~/eb$ eb create
Enter Environment Name
(default is eb-dev): eb-dev
Enter DNS CNAME prefix
(default is eb-dev): eb-dev
WARNING: The current directory does not contain any source code. Elastic Beanstalk is
launching the sample application instead.
Environment details for: elasticBeanstalkExa-env
 Application name: elastic-beanstalk-example
 Region: us-west-2
 Deployed Version: Sample Application
 Environment ID: e-j3pmc8tscn
 Platform: 64bit Amazon Linux 2015.03 v1.4.3 running Docker 1.6.2
 Tier: WebServer-Standard
 CNAME: eb-dev.elasticbeanstalk.com
 Updated: 2015-06-27 01:02:24.813000+00:00
Printing Status:
INFO: createEnvironment is starting.
-- Events -- (safe to Ctrl+C) Use "eb abort" to cancel the command.
```

O ambiente pode levar vários minutos para ficar pronto. Pressione Ctrl+C para retornar à linha de comando enquanto o ambiente é criado.

## Eb status

Execute `eb status` para ver o status atual do seu ambiente. Quando o status é `ready`, o aplicativo de exemplo está disponível em `elasticbeanstalk.com`, e o ambiente está pronto para ser atualizado.

```
~/eb$ eb status
Environment details for: elasticBeanstalkExa-env
 Application name: elastic-beanstalk-example
 Region: us-west-2
 Deployed Version: Sample Application
 Environment ID: e-gbzqc3jcra
 Platform: 64bit Amazon Linux 2015.03 v1.4.3 running Docker 1.6.2
 Tier: WebServer-Standard
 CNAME: elasticbeanstalkexa-env.elasticbeanstalk.com
 Updated: 2015-06-30 01:47:45.589000+00:00
 Status: Ready
 Health: Green
```



## Eb health

Use o comando `eb health` para visualizar as [informações de integridade](#) sobre as instâncias em seu ambiente e o estado geral dele. Use a opção `--refresh` para visualizar a integridade em uma exibição interativa que é atualizada a cada 10 segundos.

```
~/eb$ eb health
api Ok 2016-09-15 18:39:04
WebServer Java 8
total ok warning degraded severe info pending unknown
 3 3 0 0 0 0 0 0

instance-id status cause health
Overall Ok
i-0ef05ec54918bf567 Ok
i-001880c1187493460 Ok
i-04703409d90d7c353 Ok

instance-id r/sec %2xx %3xx %4xx %5xx p99 p90 p75
p50 p10
Overall 8.6 100.0 0.0 0.0 0.0 0.083* 0.065 0.053
0.040 0.019
i-0ef05ec54918bf567 2.9 29 0 0 0 0.069* 0.066 0.057
0.050 0.023
i-001880c1187493460 2.9 29 0 0 0 0.087* 0.069 0.056
0.050 0.034
i-04703409d90d7c353 2.8 28 0 0 0 0.051* 0.027 0.024
0.021 0.015

instance-id type az running load 1 load 5 user% nice%
system% idle% iowait%
i-0ef05ec54918bf567 t2.micro 1c 23 mins 0.19 0.05 3.0 0.0
0.3 96.7 0.0
i-001880c1187493460 t2.micro 1a 23 mins 0.0 0.0 3.2 0.0
0.3 96.5 0.0
i-04703409d90d7c353 t2.micro 1b 1 day 0.0 0.0 3.6 0.0
0.2 96.2 0.0

instance-id status id version ago
deployments
i-0ef05ec54918bf567 Deployed 28 app-bc1b-160915_181041 20 mins
i-001880c1187493460 Deployed 28 app-bc1b-160915_181041 20 mins
```

```
i-04703409d90d7c353 Deployed 28 app-bc1b-160915_181041 27 mins
```

## Eb events

Use `eb events` para ver uma lista de eventos emitidos pelo Elastic Beanstalk.

```
~/eb$ eb events
2015-06-29 23:21:09 INFO createEnvironment is starting.
2015-06-29 23:21:10 INFO Using elasticbeanstalk-us-east-2-EXAMPLE as Amazon S3
storage bucket for environment data.
2015-06-29 23:21:23 INFO Created load balancer named: awseb-e-g-AWSEBLoa-EXAMPLE
2015-06-29 23:21:42 INFO Created security group named: awseb-e-gbzqc3jcra-stack-
AWSEBSecurityGroup-EXAMPLE
...
```

## Eb logs

Use `eb logs` para extrair logs de uma instância em seu ambiente. Por padrão, o `eb logs` extrai os logs da primeira instância executada e exibe-os na saída padrão. Você pode especificar o ID de uma instância com a opção `--instance` para obter logs de uma instância específica.

A opção `--all` extrai os logs de todas as instâncias e salva-os nos subdiretórios em `.elasticbeanstalk/logs`.

```
~/eb$ eb logs --all
Retrieving logs...
Logs were saved to /home/local/ANT/mwunderl/ebcli/environments/test/.elasticbeanstalk/
logs/150630_201410
Updated symlink at /home/local/ANT/mwunderl/ebcli/environments/test/.elasticbeanstalk/
logs/latest
```

## Eb open

Para abrir o site do ambiente em um navegador, use `eb open`:

```
~/eb$ eb open
```

Em um ambiente em janelas, seu navegador padrão é aberto em uma nova janela. Em um ambiente de terminal, um navegador de linha de comando (por exemplo, `w3m`) será usado, se disponível.

## Eb deploy

Assim que o ambiente estiver pronto, você poderá atualizá-lo usando `eb deploy`.

Esse comando funciona melhor com algum código-fonte para empacotamento e implantação, portanto, para este exemplo, criamos um `Dockerfile` no diretório do projeto com o seguinte conteúdo:

`~/eb/Dockerfile`

```
FROM ubuntu:12.04

RUN apt-get update
RUN apt-get install -y nginx zip curl

RUN echo "daemon off;" >> /etc/nginx/nginx.conf
RUN curl -o /usr/share/nginx/www/master.zip -L https://codeload.github.com/gabrielecirulli/2048/zip/master
RUN cd /usr/share/nginx/www/ && unzip master.zip && mv 2048-master/* . && rm -rf 2048-master master.zip

EXPOSE 80

CMD ["/usr/sbin/nginx", "-c", "/etc/nginx/nginx.conf"]
```

Esse `Dockerfile` implanta uma imagem do Ubuntu 12.04 e instala o jogo 2048. Execute `eb deploy` para fazer upload do aplicativo em seu ambiente:

```
~/eb$ eb deploy
Creating application version archive "app-150630_014338".
Uploading elastic-beanstalk-example/app-150630_014338.zip to S3. This may take a while.
Upload Complete.
INFO: Environment update is starting.
-- Events -- (safe to Ctrl+C) Use "eb abort" to cancel the command.
```

Quando você executa `eb deploy`, a EB CLI empacota o conteúdo do diretório do projeto e implanta-o em seu ambiente.

**Note**

Se você inicializou um repositório git na pasta do projeto, a EB CLI sempre implanta a confirmação mais recente, mesmo se você tem alterações pendentes. Confirme suas alterações antes de executar `eb deploy` para implantá-las em seu ambiente.

## Eb config

Veja as opções de configuração disponíveis para o seu ambiente em execução com o comando `eb config`:

```
~/eb$ eb config
ApplicationName: elastic-beanstalk-example
DateUpdated: 2015-06-30 02:12:03+00:00
EnvironmentName: elasticBeanstalkExa-env
SolutionStackName: 64bit Amazon Linux 2015.03 v1.4.3 running Docker 1.6.2
settings:
 AWSEBAutoScalingScaleDownPolicy.aws:autoscaling:trigger:
 LowerBreachScaleIncrement: '-1'
 AWSEBAutoScalingScaleUpPolicy.aws:autoscaling:trigger:
 UpperBreachScaleIncrement: '1'
 AWSEBCloudwatchAlarmHigh.aws:autoscaling:trigger:
 UpperThreshold: '6000000'
...
```

Esse comando preenche uma lista de opções de configuração disponíveis em um editor de texto. Muitas das opções mostradas têm um valor `null`. Elas não são definidas por padrão, mas podem ser modificadas para atualizar os recursos em seu ambiente. Para obter mais informações sobre essas opções, consulte [Opções de configuração](#).

## Eb terminate

Se você terminou de usar o ambiente por enquanto, execute `eb terminate` para encerrá-lo.

```
~/eb$ eb terminate
The environment "eb-dev" and all associated instances will be terminated.
To confirm, type the environment name: eb-dev
INFO: terminateEnvironment is starting.
INFO: Deleted CloudWatch alarm named: awseb-e-jc8t3pmscn-stack-
AWSEBCloudwatchAlarmHigh-1XLMU7DNCBV6Y
```

```
INFO: Deleted CloudWatch alarm named: awseb-e-jc8t3pmscn-stack-
AWSEBCloudwatchAlarmLow-8IVI04W2SCXS
INFO: Deleted Auto Scaling group policy named: arn:aws:autoscaling:us-
east-2:123456789012:scalingPolicy:1753d43e-ae87-4df6-
a405-11d31f4c8f97:autoScalingGroupName/awseb-e-jc8t3pmscn-stack-
AWSEBAutoScalingGroup-90TTS2ZL4MXV:policyName/awseb-e-jc8t3pmscn-stack-
AWSEBAutoScalingScaleUpPolicy-A070H1BMUQAJ
INFO: Deleted Auto Scaling group policy named: arn:aws:autoscaling:us-
east-2:123456789012:scalingPolicy:1fd24ea4-3d6f-4373-
affc-4912012092ba:autoScalingGroupName/awseb-e-jc8t3pmscn-stack-
AWSEBAutoScalingGroup-90TTS2ZL4MXV:policyName/awseb-e-jc8t3pmscn-stack-
AWSEBAutoScalingScaleDownPolicy-LSWFUMZ46H1V
INFO: Waiting for EC2 instances to terminate. This may take a few minutes.
-- Events -- (safe to Ctrl+C)
```

Para obter uma lista completa dos comandos da CLI do EB disponíveis, confira [Referência de comandos da EB CLI](#).

#### Important

Se você encerrar um ambiente, também deverá excluir quaisquer mapeamentos CNAME criados, pois outros clientes poderão reutilizar um nome de host disponível. Certifique-se de excluir os registros DNS que apontam para seu ambiente encerrado para evitar uma entrada de DNS pendente. Uma entrada de DNS pendente pode expor o tráfego da Internet destinado ao seu domínio a vulnerabilidades de segurança. Também pode apresentar outros riscos.

Para obter mais informações, consulte [Proteção contra um registro de delegação pendente no Route 53](#) no Guia do desenvolvedor do Amazon Route 53. Você também pode aprender mais sobre entradas de DNS pendentes em [Proteções de domínio aprimoradas para solicitações do Amazon CloudFront](#) no blog de segurança da AWS.

## Usar o EB CLI com o AWS CodeBuild

O [AWS CodeBuild](#) compila o código-fonte, roda testes de unidade e produz artefatos prontos para implantação. É possível usar CodeBuild com a CLI do EB para automatizar a criação da aplicação a partir de seu código-fonte. A criação do ambiente e cada implantação subsequente iniciam com uma etapa de criação e com a implantação do aplicativo resultante.

**Note**

Algumas regiões não oferecem o CodeBuild. A integração entre o Elastic Beanstalk e o CodeBuild não funciona nessas regiões.

Para obter informações sobre os produtos da AWS oferecidos em cada região, consulte a [Tabela de regiões](#).

## Criar um aplicativo

Como criar uma aplicação do Elastic Beanstalk que use CodeBuild

1. Inclua um arquivo de especificação de criação do CodeBuild, [buildspec.yml](#), na pasta da aplicação.
2. Adicione uma entrada `eb_codebuild_settings` com opções específicas do Elastic Beanstalk ao arquivo.
3. Execute `eb init` na pasta.

**Note**

Não use os caracteres ponto (.) ou espaço ( ) em Application name (Nome da aplicação) quando usar a CLI do EB com CodeBuild.

O Elastic Beanstalk estende o [formato de arquivo de especificação de compilação do CodeBuild](#) para incluir as seguintes configurações adicionais:

```
eb_codebuild_settings:
 CodeBuildServiceRole: role-name
 ComputeType: size
 Image: image
 Timeout: minutes
```

### CodeBuildServiceRole

O ARN ou nome da função de serviço do AWS Identity and Access Management (IAM) que o CodeBuild pode usar para interagir com serviços dependentes da AWS em seu nome. Este

valor é obrigatório. Se você omiti-lo, qualquer comando `eb create` ou `eb deploy` subsequente apresentará falha.

Para saber mais sobre como criar uma função de serviço para o CodeBuild, consulte [Construir uma função de serviço do CodeBuild](#) no Guia do usuário do AWS CodeBuild.

#### Note

Você também precisa de permissões para executar ações no próprio CodeBuild. A política de usuário gerenciada do Elastic Beanstalk `AdministratorAccess-AWSElasticBeanstalk` inclui todas as permissões de ação necessárias do CodeBuild. Se você não estiver usando a política gerenciada, certifique-se de conceder as permissões a seguir na política de usuário.

```
"codebuild:CreateProject",
"codebuild>DeleteProject",
"codebuild:BatchGetBuilds",
"codebuild:StartBuild"
```

Para obter mais detalhes, consulte [Gerenciar políticas de usuário do Elastic Beanstalk](#).

## ComputeType

A quantidade de recursos usados pelo contêiner do Docker no ambiente de criação do CodeBuild. Os valores válidos são `BUILD_GENERAL1_SMALL`, `BUILD_GENERAL1_MEDIUM` e `BUILD_GENERAL1_LARGE`.

## Image

O nome da imagem do Docker Hub ou do Amazon ECR que o CodeBuild usa para o ambiente de compilação. Essa imagem do Docker deve conter bibliotecas de tempo de execução e ferramentas necessárias para criar o código e deve corresponder à plataforma de destino da aplicação. O CodeBuild gerencia e mantém um conjunto de imagens especificamente destinadas ao uso com o Elastic Beanstalk. É recomendado que você use uma delas. Para obter detalhes, consulte [Imagens do Docker fornecidas pelo CodeBuild](#) no Guia do usuário do AWS CodeBuild.

O valor `Image` é opcional. Se você omiti-lo, o comando `eb init` tentará escolher uma imagem que melhor corresponde à plataforma de destino. Além disso, se você executar `eb init` no modo interativo e ele falhar na escolha de uma imagem para você, será solicitado que você escolha

uma. Ao final de uma inicialização bem-sucedida, `eb init` grava a imagem escolhida no arquivo `buildspec.yml`.

## Timeout

A duração em minutos em que a compilação do CodeBuild é executado antes de atingir o tempo limite. Este valor é opcional. Para obter detalhes sobre valores padrão e válidos, consulte [Criar um projeto de compilação no CodeBuild](#).

### Note

Esse tempo limite controla a duração máxima para uma execução do CodeBuild, e a CLI do EB também o respeita como parte da primeira etapa para criar uma versão da aplicação. É diferente do valor que você pode especificar na opção `--timeout` dos comandos [eb create](#) ou [eb deploy](#). O último valor controla a duração máxima que a CLI do EB aguarda para a criação ou atualização do ambiente.

## Criar e implantar o código do aplicativo

Sempre que o código da aplicação precisar ser implantado, a CLI do EB usará o CodeBuild para executar uma compilação e, depois, implantará os artefatos de compilação resultantes no ambiente. Isso ocorre ao criar um ambiente do Elastic Beanstalk para a aplicação usando o comando [eb create](#) e sempre que você implantar alterações no código no ambiente usando o comando [eb deploy](#).

Se a etapa do CodeBuild falhar, a criação ou a implantação do ambiente não será iniciada.

## Usar a EB CLI com o Git

A EB CLI fornece integração com o Git. Esta seção apresenta uma visão geral de como usar o Git com a EB CLI.

Para instalar o Git e inicializar o repositório Git

1. Faça download da versão mais recente do Git acessando <http://git-scm.com>
2. Inicialize seu repositório Git digitando o seguinte:

```
~/eb$ git init
```



A EB CLI agora reconhece que seu aplicativo está configurado com o Git.

3. Se você ainda não executou `eb init`, faça isso agora:

```
~/eb$ eb init
```

## Associar ambientes do Elastic Beanstalk a ramificações Git

Você pode associar um ambiente diferente com cada ramificação do seu código. Quando você faz o check-out de uma ramificação, as alterações são implantadas no ambiente associado. Por exemplo, você pode digitar o seguinte para associar seu ambiente de produção à sua ramificação mestre e um ambiente de desenvolvimento separado à sua ramificação de desenvolvimento:

```
~/eb$ git checkout mainline
~/eb$ eb use prod
~/eb$ git checkout develop
~/eb$ eb use dev
```

## Implantação de alterações

Por padrão, a EB CLI implanta a última confirmação na ramificação atual, usando a mensagem e o ID de confirmação como a descrição e o rótulo da versão do aplicativo, respectivamente. Se você deseja implantar em seu ambiente sem confirmar, pode usar a opção `--staged` para implantar as alterações que foram adicionadas à área de preparação.

Para implantar alterações sem confirmação

1. Adicione arquivos novos e alterados à área de preparação:

```
~/eb$ git add .
```

2. Implante as alterações preparadas com `eb deploy`:

```
~/eb$ eb deploy --staged
```

Se você tiver configurado a CLI do EB para [implantar um artefato](#) e não confirmar o artefato no repositório Git, use a opção `--staged` para implantar a última compilação.

## Uso dos submódulos do Git

Alguns projetos de código-fonte se beneficiam dos submódulos do Git: repositórios no repositório de nível superior. Quando você implementa seu código-fonte usando `eb create` ou `eb deploy`, o EB CLI pode incluir submódulos no arquivo zip da versão do aplicativo e fazer upload deles com o resto do código-fonte.

Você pode controlar a inclusão de submódulos usando a `include_git_submodules` opção na seção `global` do arquivo de configuração EB CLI, `.elasticbeanstalk/config.yml`, na pasta do projeto.

Para incluir submódulos, defina essa opção como `true`:

```
global:
 include_git_submodules: true
```

Quando a opção `include_git_submodules` está ausente ou é definida como `false`, a EB CLI não inclui submódulos no arquivo zip carregado.

Consulte [Ferramentas do Git – Submódulos](#) para obter mais detalhes sobre os submódulos do Git.

### Comportamento padrão

Quando você executar `eb init` para configurar o seu projeto, a EB CLI adiciona a opção `include_git_submodules` e a define como `true`. Isso garante que todos os submódulos que você possui em seu projeto sejam incluídos em suas implantações.

A EB CLI nem sempre ofereceu suporte à inclusão de submódulos. Para evitar uma alteração acidental e indesejável em projetos que já existiam antes de adicionarmos o suporte ao submódulo, a EB CLI não inclui submódulos quando a opção `include_git_submodules` estiver ausente. Se você tiver um desses projetos existentes e deseja incluir submódulos em suas implantações, adicione a opção e defina-a como `true` conforme explicado nesta seção.

### Comportamento do CodeCommit

A integração do Elastic Beanstalk com o [CodeCommit](#) não é compatível com submódulos no momento. Se você ativou o ambiente para integrar-se ao CodeCommit, os submódulos não serão incluídos nas implantações.

## Para atribuir tags do Git à versão do aplicativo

Você pode usar uma tag do Git como rótulo de versão para identificar qual versão do aplicativo está em execução em seu ambiente. Por exemplo, digite o seguinte:

```
~/eb$ git tag -a v1.0 -m "My version 1.0"
```

## Usar o EB CLI com o AWS CodeCommit

Você pode usar a EB CLI para implantar sua aplicação diretamente de seu repositório AWS CodeCommit. Com o CodeCommit, você pode fazer upload somente de suas alterações no repositório ao implantar, em vez de fazer upload do projeto inteiro. Isso pode poupar tempo e largura de banda se você tem um projeto grande ou conectividade limitada com a Internet. A EB CLI envia confirmações locais e as usa para criar versões da aplicação quando você usa `eb appversion`, `eb create` ou `eb deploy`.

Para implantar as alterações, a integração com o CodeCommit requer a confirmação das alterações primeiro. No entanto, enquanto você desenvolve ou depura, talvez não seja conveniente enviar alterações que ainda não tem certeza de estarem funcionando. Você pode evitar confirmar as alterações feitas preparando-as e usando `eb deploy --staged` (que executa uma implantação padrão). Ou confirmar suas alterações em uma ramificação de desenvolvimento ou teste primeiro e mesclar a ramificação principal somente quando seu código estiver pronto. Com `eb use`, você pode configurar a EB CLI para implantar em um ambiente a partir da ramificação de desenvolvimento e em um ambiente diferente da ramificação principal.

### Note

Algumas regiões não oferecem CodeCommit. A integração entre o Elastic Beanstalk e o CodeCommit não funciona nessas regiões.

Para obter informações sobre os produtos da AWS oferecidos em cada região, consulte a [Tabela de regiões](#).

## Seções

- [Pré-requisitos](#)
- [Criar um repositório CodeCommit com a CLI do EB](#)
- [Implantar a partir do repositório CodeCommit](#)
- [Configurar ramificações e ambientes adicionais](#)
- [Usar um repositório CodeCommit existente](#)

## Pré-requisitos

Para usar o CodeCommit com o AWS Elastic Beanstalk, é necessário ter um repositório Git local (um que você já possua ou um novo criado) com pelo menos uma confirmação, [permissão para usar o CodeCommit](#), e um ambiente do Elastic Beanstalk em uma região suportada com o CodeCommit. O ambiente e o repositório devem estar na mesma região.

Para inicializar um repositório Git

1. Execute `git init` na pasta do projeto.

```
~/my-app$ git init
```

2. Prepare os arquivos do projeto com `git add`.

```
~/my-app$ git add .
```

3. Confirme as alterações com `git commit`.

```
~/my-app$ git commit -m "Elastic Beanstalk application"
```

## Criar um repositório CodeCommit com a CLI do EB

Para começar a usar o CodeCommit, execute [eb init](#). Durante a configuração do repositório, a CLI do EB solicita que você use o CodeCommit para armazenar seu código e acelerar as implantações.

Mesmo se você configurou previamente o projeto com `eb init`, poderá executá-lo novamente para configurar o CodeCommit.

Como criar um repositório CodeCommit com a CLI do EB

1. Execute `eb init` na pasta do projeto. Durante a configuração do repositório, a CLI do EB perguntará se você quer usar o CodeCommit para armazenar o código e acelerar as implantações. Se você configurou previamente o projeto com `eb init`, ainda poderá executá-lo novamente para configurar o CodeCommit. Digite **y** no prompt para configurar o CodeCommit.

```
~/my-app$ eb init
Note: Elastic Beanstalk now supports AWS CodeCommit; a fully-managed source control
service. To learn more, see Docs: https://aws.amazon.com/codecommit/
Do you wish to continue with CodeCommit? (y/n)(default is n): y
```

2. Escolha Create new Repository.

```
Select a repository
1) my-repo
2) [Create new Repository]
(default is 2): 2
```

3. Digite um nome de repositório ou pressione Enter para aceitar o nome padrão.

```
Enter Repository Name
(default is "codecommit-origin"): my-app
Successfully created repository: my-app
```

4. Escolha uma ramificação existente para suas confirmações ou use a EB CLI para criar uma nova ramificação.

```
Enter Branch Name
***** Must have at least one commit to create a new branch with CodeCommit *****
(default is "mainline"): ENTER
Successfully created branch: mainline
```

## Implantar a partir do repositório CodeCommit

Quando você configura o CodeCommit com o repositório da CLI do EB, esta usa o conteúdo do repositório para criar pacotes de origem. Quando você executa `eb deploy` ou `eb create`, a EB CLI

envia novas confirmações e usa a revisão HEAD da sua ramificação para criar o arquivo que ele implanta nas instâncias do EC2 em seu ambiente.

Como usar a integração do CodeCommit com a CLI do EB

1. Crie um novo ambiente com `eb create`.

```
~/my-app$ eb create my-app-env
Starting environment deployment via CodeCommit
--- Waiting for application versions to be pre-processed ---
Finished processing application version app-ac1ea-161010_201918
Setting up default branch
Environment details for: my-app-env
 Application name: my-app
 Region: us-east-2
 Deployed Version: app-ac1ea-161010_201918
 Environment ID: e-pm5mvvkfnd
 Platform: 64bit Amazon Linux 2016.03 v2.1.6 running Java 8
 Tier: WebServer-Standard
 CNAME: UNKNOWN
 Updated: 2016-10-10 20:20:29.725000+00:00
Printing Status:
INFO: createEnvironment is starting.
...
```

A EB CLI usa a confirmação mais recente na ramificação rastreada para criar a versão do aplicativo que é implantada no ambiente.

2. Quando você tiver novas confirmações locais, use `eb deploy` para enviar as confirmações e implantar em seu ambiente.

```
~/my-app$ eb deploy
Starting environment deployment via CodeCommit
INFO: Environment update is starting.
INFO: Deploying new version to instance(s).
INFO: New application version was deployed to running EC2 instances.
INFO: Environment update completed successfully.
```

3. Para testar as alterações antes de confirmá-las, use a opção `--staged` para implantar as alterações que você adicionou à área de preparação com `git add`.

```
~/my-app$ git add new-file
```

```
~/my-app$ eb deploy --staged
```

A implantação com a opção `--staged` executa uma implantação padrão, ignorando o CodeCommit.

## Configurar ramificações e ambientes adicionais

A configuração do CodeCommit se aplica a uma única ramificação. Você pode usar `eb use` e `eb codesource` para configurar ramificações adicionais ou modificar a configuração da ramificação atual.

Como configurar a integração do CodeCommit com a CLI do EB

1. Para alterar a ramificação remota, use a opção `--source` do comando [eb use](#).

```
~/my-app$ eb use test-env --source my-app/test
```

2. Para criar uma nova ramificação e um novo ambiente, faça o check-out de uma nova ramificação, envie-a para o CodeCommit, crie o ambiente e, depois, use `eb use` para conectar a ramificação local, a ramificação remota e o ambiente.

```
~/my-app$ git checkout -b production
~/my-app$ git push --set-upstream production
~/my-app$ eb create production-env
~/my-app$ eb use --source my-app/production production-env
```

3. Para configurar o CodeCommit interativamente, use [eb codesource codecommit](#).

```
~/my-app$ eb codesource codecommit
Current CodeCommit setup:
 Repository: my-app
 Branch: test
Do you wish to continue (y/n): y

Select a repository
1) my-repo
2) my-app
3) [Create new Repository]
(default is 2): 2

Select a branch
1) mainline
```

```
2) test
3) [Create new Branch with local HEAD]
(default is 1): 1
```

4. Para desabilitar a integração ao CodeCommit, use [eb codesource local](#).

```
~/my-app$ eb codesource local
Current CodeCommit setup:
 Repository: my-app
 Branch: mainline
Default set to use local sources
```

## Usar um repositório CodeCommit existente

Se você já tiver um repositório CodeCommit e quiser usá-lo com o Elastic Beanstalk, execute `eb init` na raiz do repositório Git local.

Como usar um repositório CodeCommit existente com a CLI do EB

1. Clone o repositório CodeCommit.

```
~$ git clone ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/my-app
```

2. Confira e envie uma ramificação para usar no seu ambiente do Elastic Beanstalk.

```
~/my-app$ git checkout -b dev-env
~/my-app$ git push --set-upstream origin dev-env
```

3. Executar `eb init`. Escolha os mesmos região, repositório e nome da ramificação que está usando no momento.

```
~/my-app$ eb init
Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : Europe (Ireland)
5) eu-central-1 : Europe (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
...
```



```
(default is 3): 1
...
Note: Elastic Beanstalk now supports AWS CodeCommit; a fully-managed source control
 service. To learn more, see Docs: https://aws.amazon.com/codecommit/
Do you wish to continue with CodeCommit? (y/n)(default is n): y

Select a repository
1) my-app
2) [Create new Repository]
(default is 1): 1

Select a branch
1) mainline
2) dev-env
3) [Create new Branch with local HEAD]
(default is 2): 2
```

Para obter mais informações sobre o uso de `eb init`, consulte [Configurar a EB CLI](#).

## Usar a EB CLI para monitorar a integridade do ambiente

A [Elastic Beanstalk Command Line Interface](#) (EB CLI) é uma ferramenta da linha de comando de gerenciamento de ambientes do AWS Elastic Beanstalk. Também é possível usar a CLI do EB para monitorar a integridade do ambiente em tempo real e com mais granularidade do que está disponível atualmente no console do Elastic Beanstalk

Depois de [instalar](#) e [configurar](#) a EB CLI, você pode [iniciar um novo ambiente](#) e implantar seu código com o comando `eb create`. Se você já tem um ambiente criado no console do Elastic Beanstalk, é possível anexar a CLI do EB a ele, executando `eb init` em uma pasta do projeto e seguindo as solicitações (a pasta do projeto pode estar vazia).

### Important

Certifique-se de que está usando a versão mais recente da EB CLI executando `pip install` com a opção `--upgrade`:

```
$ sudo pip install --upgrade awsebcli
```

Para concluir as instruções de instalação da CLI do EB, consulte [Instalar a EB CLI](#).

Para usar a EB CLI para monitorar a integridade do ambiente, configure primeiro uma pasta de projeto local executando `eb init` e seguindo as solicitações. Para obter instruções completas, consulte [Configurar a EB CLI](#).

Se você já tem um ambiente em execução no Elastic Beanstalk e deseja usar a CLI do EB para monitorar a integridade, siga estas etapas para anexá-lo ao ambiente existente.

Para anexar a EB CLI a um ambiente existente

1. Abra um terminal de linha de comando e navegue até a pasta do usuário.
2. Crie e abra uma nova pasta para o seu ambiente.
3. Execute o comando `eb init` e escolha o aplicativo e o ambiente cuja integridade que você quer monitorar. Se você tiver apenas um ambiente executando o aplicativo escolhido, a CLI do EB o selecionará automaticamente e não será necessário escolher o ambiente, como mostra o exemplo a seguir.

```
~/project$ eb init
Select an application to use
1) elastic-beanstalk-example
2) [Create new Application]
(default is 2): 1
Select the default environment.
You can change this later by typing "eb use [environment_name]".
1) elasticBeanstalkEx2-env
2) elasticBeanstalkExa-env
(default is 1): 1
```

Para monitorar a integridade usando a EB CLI

1. Abra uma linha de comando e navegue até a pasta do projeto.
2. Execute o comando `eb health` para exibir o status de integridade das instâncias em seu ambiente. Neste exemplo, há cinco instâncias em execução em um ambiente Linux.

```
~/project $ eb health
elasticBeanstalkExa-env Ok
2015-07-08 23:13:20
WebServer
 Ruby 2.1 (Puma)
total ok warning degraded severe info pending unknown
```

5	5	0	0	0	0	0	0	
instance-id	status	cause		health				
Overall	Ok							
i-d581497d	Ok							
i-d481497c	Ok							
i-136e00c0	Ok							
i-126e00c1	Ok							
i-8b2cf575	Ok							
instance-id	r/sec	%2xx	%3xx	%4xx	%5xx	p99	p90	p75
p50	p10	requests						
Overall	671.8	100.0	0.0	0.0	0.0	0.003	0.002	0.001
0.001	0.000							
i-d581497d	143.0	1430	0	0	0	0.003	0.002	0.001
0.001	0.000							
i-d481497c	128.8	1288	0	0	0	0.003	0.002	0.001
0.001	0.000							
i-136e00c0	125.4	1254	0	0	0	0.004	0.002	0.001
0.001	0.000							
i-126e00c1	133.4	1334	0	0	0	0.003	0.002	0.001
0.001	0.000							
i-8b2cf575	141.2	1412	0	0	0	0.003	0.002	0.001
0.001	0.000							
instance-id	type	az	running	load 1	load 5	user%	nice%	
system%	idle%	iowait%			cpu			
i-d581497d	t2.micro	1a	12 mins	0.0	0.04	6.2	0.0	
1.0	92.5	0.1						
i-d481497c	t2.micro	1a	12 mins	0.01	0.09	5.9	0.0	
1.6	92.4	0.1						
i-136e00c0	t2.micro	1b	12 mins	0.15	0.07	5.5	0.0	
0.9	93.2	0.0						
i-126e00c1	t2.micro	1b	12 mins	0.17	0.14	5.7	0.0	
1.4	92.7	0.1						
i-8b2cf575	t2.micro	1c	1 hour	0.19	0.08	6.5	0.0	
1.2	92.1	0.1						
instance-id	status	id	version	ago				
				deployments				
i-d581497d	Deployed	1	Sample Application	12 mins				
i-d481497c	Deployed	1	Sample Application	12 mins				
i-136e00c0	Deployed	1	Sample Application	12 mins				

```
i-126e00c1 Deployed 1 Sample Application 12 mins
i-8b2cf575 Deployed 1 Sample Application 1 hour
```

Neste exemplo, há uma única instância em execução em um ambiente Windows.

```
~/project $ eb health
WindowsSampleApp-env Ok
 2018-05-22 17:33:19
WebServer IIS 10.0 running on 64bit
Windows Server 2016/2.2.0
total ok warning degraded severe info pending unknown
 1 1 0 0 0 0 0 0

instance-id status cause
Overall Ok
i-065716fba0e08a351 Ok

instance-id r/sec %2xx %3xx %4xx %5xx p99 p90
p75 p50 p10 requests
Overall 13.7 100.0 0.0 0.0 0.0 1.403 0.970
0.710 0.413 0.079
i-065716fba0e08a351 2.4 100.0 0.0 0.0 0.0 1.102* 0.865
0.601 0.413 0.091

instance-id type az running % user time % privileged
time % idle time cpu
i-065716fba0e08a351 t2.large 1b 4 hours 0.2
0.1 99.7

instance-id status id version ago
deployments
i-065716fba0e08a351 Deployed 2 Sample Application 4 hours
```

## Leitura da saída

A saída exibe o nome do ambiente, a integridade geral do ambiente e a data atual na parte superior da tela.

```
elasticBeanstalkExa-env Ok
 2015-07-08 23:13:20
```

As próximas três linhas exibem o tipo de ambiente ("WebServer", neste caso), a configuração (Ruby 2.1 com Puma) e um detalhamento de quantas instâncias estão em cada um dos sete estados.

```
WebServer
Ruby 2.1 (Puma)
total ok warning degraded severe info pending unknown
 5 5 0 0 0 0 0 0
```

O resto da saída é dividido em quatro seções. A primeira exibe o status e a causa do status do ambiente geral e, em seguida, de cada instância. O exemplo a seguir mostra duas instâncias no ambiente com o status Info e a causa indicando que uma implantação foi iniciada.

```
instance-id status cause
 health
Overall Ok
i-d581497d Info Performing application deployment (running for 3 seconds)
i-d481497c Info Performing application deployment (running for 3 seconds)
i-136e00c0 Ok
i-126e00c1 Ok
i-8b2cf575 Ok
```

Para informações sobre o status de integridade e cores, consulte [Status e cores de integridade](#).

A seção de solicitações exibe informações de logs do servidor Web sobre cada instância. Neste exemplo, cada instância está fazendo solicitações normalmente e não há erros.

```
instance-id r/sec %2xx %3xx %4xx %5xx p99 p90 p75 p50
p10
Overall 13.7 100.0 0.0 0.0 0.0 1.403 0.970 0.710 0.413
0.079
i-d581497d 2.4 100.0 0.0 0.0 0.0 1.102* 0.865 0.601 0.413
0.091
i-d481497c 2.7 100.0 0.0 0.0 0.0 0.842* 0.788 0.480 0.305
0.062
i-136e00c0 4.1 100.0 0.0 0.0 0.0 1.520* 1.088 0.883 0.524
0.104
i-126e00c1 2.2 100.0 0.0 0.0 0.0 1.334* 0.791 0.760 0.344
0.197
i-8b2cf575 2.3 100.0 0.0 0.0 0.0 1.162* 0.867 0.698 0.477
0.076
```

A seção de cpu mostra as métricas do sistema operacional de cada instância. A saída difere de acordo com o sistema operacional. Veja aqui a saída para ambientes Linux.

```

instance-id type az running load 1 load 5 user% nice% system%
idle% iowait%
 cpu
i-d581497d t2.micro 1a 12 mins 0.0 0.03 0.2 0.0 0.0
99.7 0.1
i-d481497c t2.micro 1a 12 mins 0.0 0.03 0.3 0.0 0.0
99.7 0.0
i-136e00c0 t2.micro 1b 12 mins 0.0 0.04 0.1 0.0 0.0
99.9 0.0
i-126e00c1 t2.micro 1b 12 mins 0.01 0.04 0.2 0.0 0.0
99.7 0.1
i-8b2cf575 t2.micro 1c 1 hour 0.0 0.01 0.2 0.0 0.1
99.6 0.1

```

Veja aqui a saída para ambientes Windows.

```

instance-id type az running % user time % privileged time %
idle time
i-065716fba0e08a351 t2.large 1b 4 hours 0.2 0.0
99.8

```

Para obter informações sobre as métricas de servidor e de sistema operacional mostradas, consulte [Métricas de instância](#).

A seção final, de implantações, mostra o status da implantação de cada instância. Se uma implantação contínua falhar, use o ID da implantação, o status e o rótulo da versão mostrados para identificar as instâncias no ambiente que estão executando a versão incorreta.

```

instance-id status id version ago
 deployments
i-d581497d Deployed 1 Sample Application 12 mins
i-d481497c Deployed 1 Sample Application 12 mins
i-136e00c0 Deployed 1 Sample Application 12 mins
i-126e00c1 Deployed 1 Sample Application 12 mins
i-8b2cf575 Deployed 1 Sample Application 1 hour

```

## Visualização de integridade interativa

O comando `eb health` exibe um snapshot da integridade do ambiente. Para atualizar as informações exibidas a cada 10 segundos, use a opção `--refresh`.

```
$ eb health --refresh
elasticBeanstalkExa-env Ok
2015-07-09 22:10:04 (1 secs)
WebServer
 Ruby 2.1 (Puma)
total ok warning degraded severe info pending unknown
 5 5 0 0 0 0 0 0

instance-id status cause
health
Overall Ok
i-bb65c145 Ok Application deployment completed 35 seconds ago and took 26
seconds
i-ba65c144 Ok Application deployment completed 17 seconds ago and took 25
seconds
i-f6a2d525 Ok Application deployment completed 53 seconds ago and took 26
seconds
i-e8a2d53b Ok Application deployment completed 32 seconds ago and took 31
seconds
i-e81cca40 Ok

instance-id r/sec %2xx %3xx %4xx %5xx p99 p90 p75 p50
requests
p10
Overall 671.8 100.0 0.0 0.0 0.0 0.003 0.002 0.001 0.001
0.000
i-bb65c145 143.0 1430 0 0 0 0.003 0.002 0.001 0.001
0.000
i-ba65c144 128.8 1288 0 0 0 0.003 0.002 0.001 0.001
0.000
i-f6a2d525 125.4 1254 0 0 0 0.004 0.002 0.001 0.001
0.000
i-e8a2d53b 133.4 1334 0 0 0 0.003 0.002 0.001 0.001
0.000
i-e81cca40 141.2 1412 0 0 0 0.003 0.002 0.001 0.001
0.000

instance-id type az running load 1 load 5 user% nice% system%
idle% iowait%
cpu
```

```

i-bb65c145 t2.micro 1a 12 mins 0.0 0.03 0.2 0.0 0.0
99.7 0.1
i-ba65c144 t2.micro 1a 12 mins 0.0 0.03 0.3 0.0 0.0
99.7 0.0
i-f6a2d525 t2.micro 1b 12 mins 0.0 0.04 0.1 0.0 0.0
99.9 0.0
i-e8a2d53b t2.micro 1b 12 mins 0.01 0.04 0.2 0.0 0.0
99.7 0.1
i-e81cca40 t2.micro 1c 1 hour 0.0 0.01 0.2 0.0 0.1
99.6 0.1

```

```

instance-id status id version ago
 deployments
i-bb65c145 Deployed 1 Sample Application 12 mins
i-ba65c144 Deployed 1 Sample Application 12 mins
i-f6a2d525 Deployed 1 Sample Application 12 mins
i-e8a2d53b Deployed 1 Sample Application 12 mins
i-e81cca40 Deployed 1 Sample Application 1 hour

```

(Commands: **H**elp,**Q**uit, # # # #)

Este exemplo mostra um ambiente que foi recentemente ampliado de uma para cinco instâncias. A operação de escalabilidade foi bem-sucedida e agora todas as instâncias passam pelas verificações de integridade e estão prontas para fazer solicitações. No modo interativo, o status de integridade é atualizado a cada 10 segundos. No canto superior direito, um temporizador faz a contagem regressiva para a próxima atualização.

No canto inferior esquerdo, o relatório exibe uma lista de opções. Para sair do modo interativo, pressione Q. Para rolar, pressione as teclas de seta. Para ver uma lista de comandos adicionais, pressione H.

## Opções de visualização de integridade interativa

Na visualização interativa da integridade do ambiente, use o teclado para ajustar a visualização e informar ao Elastic Beanstalk para substituir ou reinicializar instâncias individuais. Para ver uma lista de comandos disponíveis na visualização do relatório de integridade no modo interativo, pressione H.

```

up,down,home,end Scroll vertically
left,right Scroll horizontally
F Freeze/unfreeze data
X Replace instance
B Reboot instance

```



```
<, > Move sort column left/right
-, + Sort order descending/ascending
P Save health snapshot data file
Z Toggle color/mono mode
Q Quit this program
```

#### Views

```
1 All tables/split view
2 Status Table
3 Request Summary Table
4 CPU%/Load Table
H This help menu
```

(press Q or ESC to return)

## Gerenciar vários ambientes do Elastic Beanstalk como um grupo com a CLI do EB

É possível usar a EB CLI para criar grupos de ambientes do AWS Elastic Beanstalk, cada um executando um componente separado de uma aplicação de arquitetura orientada a serviços. A CLI do EB gerencia esses grupos usando a API [ComposeEnvironments](#).

### Note

Grupos de ambientes não é a mesma coisa que vários contêineres em um ambiente do Docker de vários contêineres. Com grupos de ambientes, cada componente da aplicação é executada em um ambiente separado do Elastic Beanstalk com seu próprio conjunto dedicado de instâncias do Amazon EC2. Cada componente pode ser escalado separadamente. Com o Docker de vários contêineres, você combina vários componentes de um aplicativo em um único ambiente. Todos os componentes compartilham o mesmo conjunto de instâncias do Amazon EC2 com cada instância executando vários contêineres do Docker. Escolha uma dessas arquiteturas de acordo com as necessidades do seu aplicativo.

Para obter detalhes sobre o Docker de vários contêineres, consulte [Usar a ramificação da plataforma Amazon ECS](#).

Organize os componentes do aplicativo na seguinte estrutura de pastas:

```
~/project-name
|-- component-a
| `-- env.yaml
`-- component-b
 `-- env.yaml
```

Cada subpasta contém o código-fonte de um componente independente de um aplicativo que será executado em seu próprio ambiente e um arquivo de definição do ambiente chamado `env.yaml`. Para ver detalhes sobre o formato do `env.yaml`, consulte [Manifesto de ambiente \(env.yaml\)](#).

Para usar a API `Compose Environments`, primeiro execute `eb init` da pasta do projeto, especificando cada componente pelo nome da pasta que o contém com a opção `--modules`:

```
~/workspace/project-name$ eb init --modules component-a component-b
```

A CLI do EB solicita a [configuração de cada componente](#) e criação do diretório `.elasticbeanstalk` em cada componente da pasta. A EB CLI não cria arquivos de configuração no diretório `pai`.

```
~/project-name
|-- component-a
| |-- .elasticbeanstalk
| `-- env.yaml
`-- component-b
 |-- .elasticbeanstalk
 `-- env.yaml
```

Em seguida, execute o comando `eb create` com uma lista de ambientes para criar, um para cada componente:

```
~/workspace/project-name$ eb create --modules component-a component-b --env-group-suffix group-name
```

Esse comando cria um ambiente para cada componente. Os nomes dos ambientes são criados concatenando o `EnvironmentName` especificado no arquivo `env.yaml` com o nome do grupo, separados por um hífen. O tamanho total dessas duas opções e o hífen não deve exceder o máximo permitido de 23 caracteres do nome do ambiente.

Para atualizar o ambiente, use o comando `eb deploy`:

```
~/workspace/project-name$ eb deploy --modules component-a component-b
```

Você pode atualizar os componentes individualmente ou como um grupo. Especifique os componentes que deseja atualizar com a opção `--modules`.

A EB CLI armazena o nome do grupo que você usou com `eb create` na seção `branch-defaults` do arquivo de configuração da EB CLI, em `/.elasticbeanstalk/config.yml`. Para implantar o aplicativo em um grupo diferente, use a opção `--env-group-suffix` quando executar `eb deploy`. Se o grupo ainda não existe, a EB CLI cria um novo grupo de ambientes:

```
~/workspace/project-name$ eb deploy --modules component-a component-b --env-group-suffix group-2-name
```

Para encerrar ambientes, execute `eb terminate` na pasta de cada módulo. Por padrão, a EB CLI mostrará um erro se você tentar encerrar um ambiente do qual um outro ambiente em execução depende. Encerre o ambiente dependente primeiro ou use a opção `--ignore-links` para substituir o comportamento padrão:

```
~/workspace/project-name/component-b$ eb terminate --ignore-links
```

## Solução de problemas com a EB CLI

Este tópico lista as mensagens de erro comuns encontradas ao usar a EB CLI e as possíveis soluções. Se você encontrar uma mensagem de erro que não aparece aqui, use os links [Feedback](#) para relatá-la.

**ERRO:** Ocorreu um erro ao usar o comando `git`. Código do erro: 128 Erro: fatal: Não é um nome de objeto válido HEAD

**Causa:** essa mensagem de erro aparece quando você inicializou um repositório Git, mas ainda não confirmou. A EB CLI procura a revisão HEAD quando a pasta do seu projeto contém um repositório Git.

**Solução:** adicione os arquivos na pasta do seu projeto à área de preparação e confirme:

```
~/my-app$ git add .
~/my-app$ git commit -m "First commit"
```

ERRO: Esta ramificação não tem um ambiente padrão. Você deve especificar um ambiente digitando "eb status my-env-name" ou definir um ambiente padrão digitando "eb use my-env-name".

Causa: ao criar uma nova ramificação em git, ela não é anexada a um ambiente Elastic Beanstalk por padrão.

Solução: execute `eb list` para ver uma lista dos ambientes disponíveis. Em seguida, execute `eb use env-name` para usar um dos ambientes disponíveis.

ERRO: Plataformas 2.0+ exigem uma função de serviço. Você pode fornecer uma com a opção `--service-role`

Causa: se você especificar um nome de ambiente com `eb create` (por exemplo, `eb create my-env`), a EB CLI não tentará criar uma função de serviço para você. Se você não tem a função de serviço padrão, o erro acima é mostrado.

Solução: execute `eb create` sem um nome de ambiente e siga os prompts para criar a função de serviço padrão.

## Solução de problemas de implantações

Se a implantação do Elastic Beanstalk não for concluída sem nenhum problema conforme o planejado, você poderá receber uma resposta 404 (se a aplicação não iniciar) ou 500 (se a aplicação falhar durante o tempo de execução), em vez de ver seu site. Para solucionar muitos problemas comuns, você pode usar a EB CLI para conferir o status da implantação, visualizar os logs, obter acesso à sua instância do EC2 com SSH ou abrir a página do Console de Gerenciamento da AWS para seu ambiente da aplicação.

Para usar a EB CLI para ajudar a solucionar problemas de implantação

1. Execute `eb status` para ver o status da implantação atual e a integridade dos hosts do EC2. Por exemplo:

```
$ eb status --verbose
```

```
Environment details for: python_eb_app
Application name: python_eb_app
Region: us-west-2
Deployed Version: app-150206_035343
Environment ID: e-wa8u6rrmqy
```

```

Platform: 64bit Amazon Linux 2014.09 v1.1.0 running Python 2.7
Tier: WebServer-Standard-
CNAME: python_eb_app.elasticbeanstalk.com
Updated: 2015-02-06 12:00:08.557000+00:00
Status: Ready
Health: Green
Running instances: 1
 i-8000528c: InService

```

### Note

O uso do switch `--verbose` fornece informações sobre o status das suas instâncias em execução. Sem ele, o `eb status` imprime somente informações gerais sobre seu ambiente.

## 2. Execute `eb health` para visualizar informações de integridade do seu ambiente:

```

$ eb health --refresh
elasticBeanstalkExa-env Degraded
2016-03-28 23:13:20
WebServer
 Ruby 2.1 (Puma)
total ok warning degraded severe info pending unknown
5 2 0 2 1 0 0 0

instance-id status cause
Overall Degraded Incorrect application version found on 3 out of 5
instances. Expected version "Sample Application" (deployment 1).
i-d581497d Degraded Incorrect application version "v2" (deployment 2).
Expected version "Sample Application" (deployment 1).
i-d481497c Degraded Incorrect application version "v2" (deployment 2).
Expected version "Sample Application" (deployment 1).
i-136e00c0 Severe Instance ELB health has not been available for 5 minutes.
i-126e00c1 Ok
i-8b2cf575 Ok

instance-id r/sec %2xx %3xx %4xx %5xx p99 p90 p75
p50 p10
Overall 646.7 100.0 0.0 0.0 0.0 0.003 0.002 0.001
0.001 0.000
i-dac3f859 167.5 1675 0 0 0 0.003 0.002 0.001
0.001 0.000

```

```

i-05013a81 161.2 1612 0 0 0 0.003 0.002 0.001
0.001 0.000
i-04013a80 0.0 - - - - - - -
- -
i-3ab524a1 155.9 1559 0 0 0 0.003 0.002 0.001
0.001 0.000
i-bf300d3c 162.1 1621 0 0 0 0.003 0.002 0.001
0.001 0.000

```

```

instance-id type az running load 1 load 5 user% nice%
system% idle% iowait%
i-d581497d t2.micro 1a 25 mins 0.16 0.1 7.0 0.0
1.7 91.0 0.1
i-d481497c t2.micro 1a 25 mins 0.14 0.1 7.2 0.0
1.6 91.1 0.0
i-136e00c0 t2.micro 1b 25 mins 0.0 0.01 0.0 0.0
0.0 99.9 0.1
i-126e00c1 t2.micro 1b 25 mins 0.03 0.08 6.9 0.0
2.1 90.7 0.1
i-8b2cf575 t2.micro 1c 1 hour 0.05 0.41 6.9 0.0
2.0 90.9 0.0

```

```

instance-id status id version ago
deployments
i-d581497d Deployed 2 v2 9 mins
i-d481497c Deployed 2 v2 7 mins
i-136e00c0 Failed 2 v2 5 mins
i-126e00c1 Deployed 1 Sample Application 25 mins
i-8b2cf575 Deployed 1 Sample Application 1 hour

```

O exemplo acima mostra um ambiente com cinco instâncias em que a implantação da versão "v2" falhou na terceira instância. Após uma implantação com falha, a versão esperada será redefinida para a última versão bem-sucedida que, neste caso, é o "Aplicativo de exemplo" da primeira implantação. Consulte [Usar a EB CLI para monitorar a integridade do ambiente](#) para obter mais informações.

3. Execute `eb logs` para fazer download e visualizar os logs associados à implantação do aplicativo.

```
$ eb logs
```

4. Execute `eb ssh` para conectar-se à instância do EC2 que está executando seu aplicativo e examiná-la diretamente. Na instância, o aplicativo implantado está no diretório `/opt/python/current/app`, e seu ambiente Python está em `/opt/python/run/venv/`.
5. Execute `eb console` para visualizar o ambiente da aplicação no [Console de Gerenciamento da AWS](#). Você pode usar a interface da Web para examinar facilmente diversos aspectos da implantação, incluindo configuração, status, eventos e logs do aplicativo. Você também pode fazer download das versões atuais ou anteriores do aplicativo que você implantou no servidor.

## Referência de comandos da EB CLI

É possível usar a Interface de Linhas de Comando (CLI do EB) do Elastic Beanstalk para executar uma variedade de operações para implantar e gerenciar as aplicações e os ambientes do Elastic Beanstalk. A EB CLI integra-se ao Git para implantar o código-fonte do aplicativo que está sob controle de código-fonte Git. Para obter mais informações, consulte [Usar a Interface de Linhas de Comando do Elastic Beanstalk \(CLI do EB\)](#) e [Usar a EB CLI com o Git](#).

### Comandos

- [eb abort](#)
- [eb appversion](#)
- [eb clone](#)
- [eb codesource](#)
- [eb config](#)
- [eb console](#)
- [eb create](#)
- [eb deploy](#)
- [eb events](#)
- [eb health](#)
- [eb init](#)
- [eb labs](#)
- [eb list](#)
- [eb local](#)
- [eb logs](#)

- [eb open](#)
- [eb platform](#)
- [eb printenv](#)
- [eb restore](#)
- [eb scale](#)
- [eb setenv](#)
- [eb ssh](#)
- [eb status](#)
- [eb swap](#)
- [eb tags](#)
- [eb terminate](#)
- [eb upgrade](#)
- [eb use](#)
- [Opções comuns](#)

## eb abort

### Descrição

Cancela uma atualização quando as alterações de configuração do ambiente nas instâncias ainda estão em andamento.

#### Note

Se você tiver mais de dois ambientes passando por uma atualização, será solicitado a selecionar o nome do ambiente no qual deseja reverter as alterações.

### Sintaxe

eb abort

eb abort ***environment-name***



## Opções

Name (Nome)	Descrição
<a href="#">Opções comuns</a>	

## Resultado

O comando mostra uma lista de ambientes que estão sendo atualizados e solicita que você escolha a atualização que deseja anular. Se apenas um ambiente está sendo atualizado, você não precisa especificar o nome dele. Se bem-sucedido, o comando reverte as alterações de configuração do ambiente. O processo de reversão continua até que todas as instâncias no ambiente tenham a configuração de ambiente anterior ou até que o processo de reversão falhe.

## Exemplo

O seguinte exemplo cancela a atualização da plataforma.

```
$ eb abort
Aborting update to environment "tmp-dev".
<list of events>
```

## eb appversion

### Descrição

O comando `appversion` da EB CLI gerencia as [versões da aplicação](#) do Elastic Beanstalk. Você pode criar uma nova versão de aplicação sem implantar, excluir uma versão da aplicação ou criar a [política de ciclo de vida da versão da aplicação](#). Se você chamar o comando sem opções, ele entrará no [modo interativo](#).

Use a opção `--create` para criar uma nova versão da aplicação.

Use a opção `--delete` para excluir uma versão do aplicativo.

Use a opção `lifecycle` para exibir ou criar a política de ciclo de vida de versão do aplicativo. Para obter mais informações, consulte [the section called "Ciclo de vida da versão"](#).

## Sintaxe

```
eb appversion
```

```
eb appversion [-c | --create]
```

```
eb appversion [-d | --delete] version-label
```

```
eb appversion lifecycle [-p | --print]
```

## Opções

Name (Nome)	Descrição
	Tipo: sequência
<i>application_name</i> ou —application_name <i>nome da aplicação</i>	O nome do aplicativo. Se uma aplicação com o nome especificado não for encontrada, a EB CLI criará uma versão da aplicação para uma nova aplicação.  Aplicável somente com a opção --create.  Tipo: sequência
-c ou --create	Crie uma <a href="#">nova versão</a> da aplicação.
-d <i>rótulo-da-versão</i> ou --delete <i>rótulo-da-versão</i>	Exclua a versão da aplicação rotulada como <i>rótulo de versão</i> .
-l <i>version_label</i> ou --label <i>version_label</i>	Especifique um rótulo a ser usado para a versão criada pela EB CLI. Se você não usar essa opção, a EB CLI gerará um novo rótulo exclusivo. Se você fornecer um rótulo de versão, verifique se ele é exclusivo.

Name (Nome)	Descrição
	<p>Tipo: sequência</p> <p>Aplicável somente com a opção <code>--create</code>.</p> <p>Tipo: sequência</p>
ciclo de vida	<p>Chamar o editor padrão para criar uma nova política de ciclo de vida de versão do aplicativo. Use esta política para evitar atingir a <a href="#">cota da versão da aplicação</a>.</p>
<p>lifecycle -p</p> <p>ou</p> <p>lifecycle --print</p>	<p>Exibir a política de ciclo de vida do aplicativo atual.</p>
<p>-m "<i>version_description</i> "</p> <p>ou</p> <p>--message "<i>version_description</i> "</p>	<p>A descrição da versão da aplicação. Está incluído entre aspas duplas.</p> <p>Aplicável somente com a opção <code>--create</code>.</p> <p>Tipo: sequência</p>
<p>-p</p> <p>ou</p> <p>--process</p>	<p>Pré-processa e valida o manifesto de ambiente e os arquivos de configuração no pacote de origem. A validação de arquivos de configuração pode identificar problemas. Recomendamos que você faça isso antes de implantar a versão da aplicação no ambiente.</p> <p>Aplicável somente com a opção <code>--create</code>.</p>
<p>--source codecommit/<i>repository-name/branch-name</i></p>	<p>Repositório do CodeCommit e ramificação. Para obter mais informações, consulte <a href="#">Usar o EB CLI com o AWS CodeCommit</a>.</p> <p>Aplicável somente com a opção <code>--create</code>.</p>

Name (Nome)	Descrição
	Tipo: sequência
<code>--staged</code>	Use os arquivos preparados no índice git em vez da confirmação HEAD para criar a versão da aplicação.  Aplicável somente com a opção <code>--create</code> .
<code>--timeout</code> <i>minutos</i>	O número de minutos antes de esgotar o tempo limite do comando.  Aplicável somente com a opção <code>--create</code> .
<a href="#">Opções comuns</a>	

## Usar o comando interativamente

Se você usar o comando sem argumentos, a saída exibirá as versões da aplicação. Eles estão listados em ordem cronológica inversa, com a última versão listada primeiro. Consulte a seção [Examples \(Exemplos\)](#) para obter exemplos da aparência da tela. Observe que a linha de status é exibida na parte inferior. A linha de status exibe informações sensíveis ao contexto.

Pressione `d` para excluir uma versão do aplicativo, pressione `l` para gerenciar a política de ciclo de vida de seu aplicativo ou pressione `q` para sair sem fazer alterações.

### Note

Se a versão estiver implantada em qualquer ambiente, você não poderá excluí-la.

## Resultado

O comando com a opção `--create` exibe uma mensagem confirmando que a versão da aplicação foi excluída.

O comando com a opção `--delete` *rótulo-da-versão* exibe uma mensagem confirmando que a versão do aplicativo foi excluída.

## Exemplos

O exemplo a seguir mostra a janela interativa de um aplicativo sem implantações.

```

No Environment Specified Application Name: versions
Environment Status: Unknown Health Unknown
Current version # deployed: None

Version Label Date Created Age Description
3 v4 2016/12/22 13:28 56 secs new features
2 v3 2016/12/22 13:27 1 min important update
1 v1 2016/12/15 23:51 6 days wow

(Commands: Quit, Delete, Lifecycle, ▼▲◀▶)

```

O exemplo a seguir mostra a janela interativa de um aplicativo com a quarta versão, com o rótulo da versão Sample Application (Aplicativo de exemplo), implantada.

```

Sample-env Application Name: versions
Environment Status: Launching Health Green
Current version # deployed: 4

Version Label Date Created Age Description
4 Sample Application 2016/12/22 13:30 2 mins -
3 v4 2016/12/22 13:28 4 mins new features
2 v3 2016/12/22 13:27 5 mins important update
1 v1 2016/12/15 23:51 6 days wow

(Commands: Quit, Delete, Lifecycle, ▼▲◀▶)

```

O exemplo a seguir mostra a saída de um comando `eb appversion lifecycle -p`, em que *ACCOUNT-ID* é o ID da conta do usuário:

```

Application details for: lifecycle
Region: sa-east-1
Description: Application created from the EB CLI using "eb init"
Date Created: 2016/12/20 02:48 UTC
Date Updated: 2016/12/20 02:48 UTC
Application Versions: ['Sample Application']
Resource Lifecycle Config(s):
 VersionLifecycleConfig:
 MaxCountRule:
 DeleteSourceFromS3: False
 Enabled: False
 MaxCount: 200
 MaxAgeRule:
 DeleteSourceFromS3: False
 Enabled: False

```

```
MaxAgeInDays: 180
ServiceRole: arn:aws:iam::ACCOUNT-ID:role/aws-elasticbeanstalk-service-role
```

## eb clone

### Descrição

Clona um ambiente para um novo ambiente de modo que ambos têm configurações de ambiente idênticas.

#### Note

Por padrão, independentemente da versão da stack de soluções do ambiente do qual você criou o clone, o comando `eb clone` cria o ambiente clone com a stack de soluções mais recente. Você pode suprimir isso incluindo a opção `--exact` ao executar o comando.

### Sintaxe

```
eb clone
```

```
eb clone environment-name
```

### Opções

Name (Nome)	Descrição
<code>-n <i>string</i></code> ou <code>--clone_name <i>string</i></code>	Nome desejado para o ambiente clonado.
<code>-c <i>string</i></code> ou <code>--cname <i>string</i></code>	Prefixo CNAME desejado para o ambiente clonado.

Name (Nome)	Descrição
<code>--envvars</code>	<p>Propriedades do ambiente em uma lista separada por vírgula com o formato <i>nome=valor</i>.</p> <p>Tipo: sequência</p> <p>Restrições:</p> <ul style="list-style-type: none"><li>• Os pares de chave e valor devem ser separados por vírgulas.</li><li>• As chaves e valores podem conter qualquer caractere alfabético em qualquer idioma, qualquer caractere numérico, espaço em branco, separador invisível e os seguintes símbolos: <code>_ . : / + \ - @</code></li><li>• As chaves podem conter até 128 caracteres. Os valores podem conter até 256 caracteres.</li><li>• As chaves e os valores diferenciam letras maiúsculas de minúsculas.</li><li>• Os valores não podem coincidir com o nome do ambiente.</li><li>• Os valores não podem incluir <code>aws:</code> ou <code>elasticbeanstalk:</code>.</li><li>• O tamanho total de todas as propriedades do ambiente não pode exceder 4096 bytes.</li></ul>
<code>--exact</code>	<p>Impede que o Elastic Beanstalk atualize a versão da pilha de soluções do novo ambiente clone para a versão mais recente disponível (para a plataforma do ambiente original).</p>
<code>--scale <i>número</i></code>	<p>O número de instâncias a serem executadas no ambiente clone quando ele é iniciado.</p>
<code>--tags <i>nome=valor</i></code>	<p>As <a href="#">tags</a> para os recursos em seu ambiente em uma lista separada por vírgulas com o formato <i>nome=valor</i>.</p>
<code>--timeout</code>	<p>O número de minutos antes de esgotar o tempo limite do comando.</p>

Name (Nome)	Descrição
<a href="#">Opções comuns</a>	

## Resultado

Se tiver êxito, o comando cria um ambiente que tem as mesmas configurações que o ambiente original ou com modificações no ambiente, como especificado por qualquer opção `eb clone`.

## Exemplo

O exemplo a seguir clona o ambiente especificado.

```
$ eb clone
Enter name for Environment Clone
(default is tmp-dev-clone):
Enter DNS CNAME prefix
(default is tmp-dev-clone):
Environment details for: tmp-dev-clone
 Application name: tmp
 Region: us-west-2
 Deployed Version: app-141029_144740
 Environment ID: e-vjvrqnn5pv
 Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
 Tier: WebServer-Standard-1.0
 CNAME: tmp-dev-clone.elasticbeanstalk.com
 Updated: 2014-10-29 22:00:23.008000+00:00
Printing Status:
2018-07-11 21:04:20 INFO: createEnvironment is starting.
2018-07-11 21:04:21 INFO: Using elasticbeanstalk-us-west-2-888888888888 as Amazon S3
 storage bucket for environment data.
...
2018-07-11 21:07:10 INFO: Successfully launched environment: tmp-dev-clone
```

## eb codesource

### Descrição

Configura a CLI do EB a ser [implantada de um repositório do CodeCommit](#) ou desabilita a integração do CodeCommit e faz upload do pacote de origem de sua máquina local.



**Note**

Algumas regiões da AWS não oferecem CodeCommit. A integração entre o Elastic Beanstalk e o CodeCommit não funciona nessas regiões.

Para obter informações sobre os serviços da AWS oferecidos em cada região, consulte a [Tabela de regiões](#).

## Sintaxe

```
eb codesource
```

```
eb codesource codecommit
```

```
eb codesource local
```

## Opções

Name (Nome)	Descrição
<a href="#">Opções comuns</a>	

## Resultado

`eb codesource` solicita que você escolha entre integração do CodeCommit e implantações padrão.

`eb codesource codecommit` inicia a configuração do repositório interativo para integração do CodeCommit.

`eb codesource local` mostra a configuração original e desabilita a integração do CodeCommit.

## Exemplos

Use `eb codesource codecommit` para configurar a integração do CodeCommit para a ramificação atual.

```
~/my-app$ eb codesource codecommit
Select a repository
1) my-repo
2) my-app
```

```
3) [Create new Repository]
(default is 1): 1

Select a branch
1) mainline
2) test
3) [Create new Branch with local HEAD]
(default is 1): 1
```

Use `eb codesource local` para desabilitar a integração do CodeCommit para a ramificação atual.

```
~/my-app$ eb codesource local
Current CodeCommit setup:
 Repository: my-app
 Branch: mainline
Default set to use local sources
```

## eb config

### Descrição

Gerencia as [configurações](#) ativas e as [configurações salvas](#) do seu ambiente. Você pode usar esse comando para fazer upload, fazer download ou listar as configurações salvas do seu ambiente. Você também pode usá-lo para fazer download, exibir ou atualizar suas configurações ativas.

Se o diretório raiz contém um arquivo `platform.yaml` especificando uma plataforma personalizada, este comando também altera as configurações do construtor. Isso é feito com base nos valores definidos em `platform.yaml`.


#### Note

`eb config` não exibe as propriedades do ambiente. Para definir as propriedades do ambiente que você pode ler a partir da sua aplicação, use [eb setenv](#).

### Sintaxe

A seguir estão partes da sintaxe que é usada para que o comando `eb config` funcione com as [configurações](#) ativas do seu ambiente. Para obter exemplo específicos, consulte a seção [Exemplos](#) mais adiante neste tópico.

- `eb config`: exibe as configurações ativas do seu ambiente em um editor de texto que você configurou como a variável de ambiente `EDITOR`. Quando você salvar as alterações no arquivo e fechar o editor, o ambiente será atualizado com as configurações da opção no arquivo.


 Note

Se você não configurou uma variável de ambiente `EDITOR`, a EB CLI exibirá suas configurações de opção no editor padrão para arquivos YAML.

- `eb config environment-name`: exibe e atualiza a configuração para o ambiente nomeado. A configuração é exibida em um editor de texto que você configurou ou em seus arquivos YAML do editor padrão.
- `eb config save`: salva as configurações ativas para o ambiente atual `.elasticbeanstalk/saved_configs/` com o nome do arquivo `[configuration-name].cfg.yml`. Por padrão, a CLI do EB salva as configurações com um *nome-de-configuração* baseado no nome do ambiente. Você pode especificar um nome de configuração diferente incluindo a opção `--cfg` com o nome de configuração desejado quando executar o comando.

Você pode marcar a configuração salva usando a opção `--tags`.

- `eb config --display`: grava as configurações ativas de um ambiente em `stdout` em vez de um arquivo. Por padrão, isso exibe as configurações para o final.
- `eb config --update configuration_string | file_path`: atualiza as configurações ativas para o ambiente atual com as informações especificadas em *configuration\_string* ou dentro do arquivo identificado por *file\_path*.

 Note

As opções `--display` e `--update` oferecem flexibilidade para ler e revisar as configurações de um ambiente programaticamente.

A tabela a seguir descreve a sintaxe para usar o comando `eb config` para funcionar com [configurações salvas](#). Para obter exemplos, consulte a seção [Exemplos](#) mais adiante neste tópico.

- `eb config get config-name`: faz o download da configuração salva nomeada do Amazon S3.

- `eb config delete config-name` : exclui a configuração salva nomeada do Amazon S3. Também o exclui localmente, se você já fez o download.
- `eb config list`: lista as configurações salvas que você tem no Amazon S3.
- `eb config put filename`: faz upload da configuração salva nomeada em um bucket do Amazon S3. O **nome do arquivo** deve ter a extensão de arquivo `.cfg.yml`. Para especificar o nome de arquivo sem um caminho, você pode salvar o arquivo na pasta `.elasticbeanstalk` ou `.elasticbeanstalk/saved_configs/` antes de executar o comando. Como alternativa, você pode especificar o **nome de arquivo** fornecendo o caminho completo.

## Opções

Name (Nome)	Descrição
<code>--cfg <b>config-name</b></code>	<p>O nome a ser usado para uma configuração salva.</p> <p>Essa opção funciona apenas com <code>eb config save</code>.</p>
<code>-d</code> ou <code>--display</code>	<p>Exibe as configurações do ambiente atual (gravações em <code>stdout</code>).</p> <p>Use com a opção <code>--format</code> para especificar a saída para estar em JSON ou YAML. Se você não especificar, a saída estará no formato YAML.</p> <p>Essa opção só funciona se você usar o comando <code>eb config</code> sem nenhum dos outros subcomandos.</p>
<code>-f <b>format_type</b></code> ou <code>--format <b>format_type</b></code>	<p>Especifica o formato de exibição. Os valores válidos são JSON ou YAML.</p> <p>O padrão é YAML.</p> <p>Essa opção funciona somente com a opção <code>--display</code>.</p>
<code>--tags <b>key1=value1[,ke</b></code>	<p>As tags a serem adicionadas à sua configuração salva. Ao especificar tags na lista, especifique-as como pares chave=valor e separe cada uma com uma vírgula.</p>

Name (Nome)	Descrição
	<p>Para obter mais informações, consulte <a href="#">Marcação de configurações salvas</a>.</p> <p>Essa opção funciona apenas com eb config save.</p>
<code>--timeout <i>timeout</i></code>	<p>O número de minutos antes de esgotar o tempo limite do comando.</p>

Name (Nome)	Descrição
<pre>-u <i>configuration_string</i>   <i>file_path</i> ou --update <i>configuration_string</i>   <i>file_path</i></pre>	<p>Atualiza as configurações ativas para o ambiente atual.</p> <p>Essa opção só funciona se você usar o comando <code>eb config</code> sem nenhum dos outros subcomandos.</p> <p>O parâmetro <i>configuration_string</i>   <i>file_path</i> é do tipo de string. A string fornece a lista de namespaces e opções correspondentes para adicionar, atualizar ou remover das configurações do seu ambiente. Como alternativa, a string de entrada pode representar um arquivo que contém as mesmas informações.</p> <p>Para especificar um nome de arquivo, a string de entrada deve seguir o formato "<code>file://&lt; <i>path</i>&gt;&lt; <i>filename</i>&gt;</code>". Para especificar o nome do arquivo sem um <i>path</i>, salve o arquivo na pasta em que você executa o comando. Como alternativa, você pode especificar o nome de arquivo fornecendo o caminho completo.</p> <p>As informações de configuração devem atender às condições a seguir. Pelo menos uma das seções, <code>OptionSettings</code> ou <code>OptionsToRemove</code>, é necessária. Use <code>OptionSettings</code> para adicionar ou alterar opções. Use <code>OptionsToRemove</code> para remover opções de um namespace. Para obter exemplo específicos, consulte a seção <a href="#">Exemplos</a> mais adiante neste tópico.</p> <p>Example</p> <p>Formato YAML</p> <pre>OptionSettings:   namespace1:     option-name-1: <i>option-value-1</i>     option-name-2: <i>option-value-2</i>     ... OptionsToRemove:   namespace1:</pre>

Name (Nome)	Descrição
	<pre>option-name-1 option-name-2 ...</pre> <p>Example</p> <p>Formato JSON</p> <pre>{   "OptionSettings": {     "namespace1": {       "option-name-1": " <i>option-value-1</i> ",       "option-name-2": " <i>option-value-2</i> ",       ...     }   },   "OptionsToRemove": {     "namespace1": {       "option-name-1",       "option-name-2",       ...     }   } }</pre>
<a href="#">Opções comuns</a>	

## Resultado

Se o comando `eb config` ou `eb config environment-name` for executado com êxito sem subcomandos ou opções adicionadas, o comando exibirá as configurações de opção atuais no editor de texto que você configurou como a variável de ambiente `EDITOR`. Se você não configurou uma variável de ambiente `EDITOR`, a EB CLI exibirá suas configurações de opção no editor padrão para arquivos YAML.

Quando você salvar as alterações no arquivo e fechar o editor, o ambiente será atualizado com as configurações da opção no arquivo. A saída a seguir é exibida para confirmar a atualização de configuração.

```
$ eb config myApp-dev
 Printing Status:
 2021-05-19 18:09:45 INFO Environment update is starting.
 2021-05-19 18:09:55 INFO Updating environment myApp-dev's configuration
settings.
 2021-05-19 18:11:20 INFO Successfully deployed new configuration to
environment.
```

Se o comando for executado com êxito com a opção `--display`, ele exibirá as configurações do ambiente atual (gravações em `stdout`).

Se o comando for executado com êxito com o parâmetro `get`, o comando exibirá a localização da cópia local que você baixou.

Se o comando for executado com êxito com o parâmetro `save`, o comando exibirá a localização do arquivo salvo.

## Exemplos

Esta seção descreve como alterar o editor de texto usado para visualizar e editar o arquivo de configurações de opção.

Para Linux e UNIX, o seguinte exemplo altera o editor para vim:

```
$ export EDITOR=vim
```

Para Linux e UNIX, o exemplo a seguir altera o editor para aquele que estiver instalado em `/usr/bin/kate`.

```
$ export EDITOR=/usr/bin/kate
```

Para Windows, o exemplo a seguir altera o editor para Notepad++.

```
> set EDITOR="C:\Program Files\Notepad++\Notepad++.exe
```

Esta seção fornece exemplos do comando `eb config` quando ele é executado com os parâmetros.

O exemplo a seguir exclui a configuração salva chamada `app-tmp`.



```
$ eb config delete app-tmp
```

O exemplo a seguir faz download da configuração salva com o nome `app-tmp` do bucket do Amazon S3.

```
$ eb config get app-tmp
```

O exemplo a seguir lista os nomes das configurações salvas que são armazenadas no bucket do Amazon S3.

```
$ eb config list
```

O exemplo a seguir faz upload da cópia local da configuração salva `app-tmp` no bucket do Amazon S3.

```
$ eb config put app-tmp
```

O exemplo a seguir salva as configurações do ambiente em execução no momento. Se você não fornecer um nome a ser usado para a configuração salva, o Elastic Beanstalk fornecerá um nome ao arquivo de configuração de acordo com o nome do ambiente. Por exemplo, um ambiente chamado `tmp-dev` seria chamado `tmp-dev.cfg.yml`. O Elastic Beanstalk salva o arquivo na pasta `/.elasticbeanstalk/saved_configs/`.

```
$ eb config save
```

O exemplo a seguir mostra como usar a opção `--cfg` para salvar as configurações do ambiente `tmp-dev` em um arquivo chamado `v1-app-tmp.cfg.yml`. O Elastic Beanstalk salva o arquivo na pasta `/.elasticbeanstalk/saved_configs/`. Se você não especificar um nome de ambiente, o Elastic Beanstalk salvará as configurações do ambiente em execução no momento.

```
$ eb config save tmp-dev --cfg v1-app-tmp
```

Esta seção fornece exemplos para o comando `eb config` quando ele é executado sem subcomandos.

O comando a seguir exibe as configurações de opção do ambiente atual em um editor de texto.

```
$ eb config
```

O comando a seguir exibe as configurações de opção para o ambiente my-env em um editor de texto.

```
$ eb config my-env
```

O exemplo a seguir exibe as configurações de opções para o ambiente atual. Ele produz no formato YAML porque nenhum formato específico foi especificado com a opção `--format`.

```
$ eb config --display
```

O exemplo a seguir atualiza as configurações de opções para seu ambiente atual com as especificações no arquivo nomeado `example.txt`. O arquivo está no formato YAML ou JSON. A EB CLI detecta automaticamente o formato do arquivo.

- A opção `Minsize` é definida como 1 para o namespace `aws:autoscaling:asg`.
- O tamanho do lote para o namespace `aws:elasticbeanstalk:command` é definido como 30%.
- Ele remove a configuração da opção `IdleTimeout: None` (Tempo limite de inatividade) do namespace `AWSEBV2LoadBalancer.aws:elbv2:loadbalancer`.

```
$ eb config --update "file://example.txt"
```

Example - nome do arquivo: **example.txt** - formato YAML

```
OptionSettings:
 'aws:elasticbeanstalk:command':
 BatchSize: '30'
 BatchSizeType: Percentage
 'aws:autoscaling:asg':
 MinSize: '1'
OptionsToRemove:
 'AWSEBV2LoadBalancer.aws:elbv2:loadbalancer':
 IdleTimeout
```

Example - nome do arquivo: **example.txt** - formato JSON

```
{
 "OptionSettings": {
 "aws:elasticbeanstalk:command": {
```

```

 "BatchSize": "30",
 "BatchSizeType": "Percentage"
 },
 "aws:autoscaling:asg": {
 "MinSize": "1"
 }
},
"OptionsToRemove": {
 "AWSEBV2LoadBalancer.aws:elbv2:loadbalancer": {
 "IdleTimeout"
 }
}
}
}

```

Os exemplos a seguir atualizam as configurações de opções para seu ambiente atual. O comando define a opção MinSize como 1 para o namespace `aws:autoscaling:asg`.

#### Note

Esses exemplos são específicos do Windows PowerShell. Eles escapam das ocorrências literais do caractere de aspas duplas (`"`), precedendo-o com um caractere barra (`\`). Diferentes sistemas operacionais e ambientes de linha de comando podem ter sequências de escape diferentes. Por esse motivo, recomendamos usar a opção de arquivo mostrada nos exemplos anteriores. Especificar as opções de configuração em um arquivo não requer caracteres de escape e é consistente em diferentes sistemas operacionais.

O exemplo a seguir está no formato JSON. A EB CLI detecta se o formato está em JSON ou YAML.

```
PS C:\Users\myUser\EB_apps\myApp-env>eb config --update '{"OptionSettings\": {"aws:autoscaling:asg\":{"MaxSize\":"1\"}}}'
```

O exemplo a seguir está no formato YAML. Para inserir a string YAML no formato correto, o comando inclui espaçamento e retornos de fim de linha que são necessários em um arquivo YAML.

- Termine cada linha com a chave “enter” ou “return”.
- Inicie a segunda linha com dois espaços e inicie a terceira linha com quatro espaços.

```
PS C:\Users\myUser\EB_apps\myApp-env>eb config --update 'OptionSettings:
```

```
>> aws:autoscaling:asg:
>> MinSize: \"1\"
```

## eb console

### Descrição

Abre um navegador para exibir o painel de configuração do ambiente no console de gerenciamento do Elastic Beanstalk.

Se o diretório raiz contiver um arquivo `platform.yaml` especificando uma plataforma personalizada, esse comando também exibirá a configuração do ambiente do compilador, conforme especificado no `platform.yaml`, no console de gerenciamento do Elastic Beanstalk.

### Sintaxe

eb console

eb console ***environment-name***

### Opções

Name (Nome)	Descrição
<a href="#">Opções comuns</a>	

## eb create

### Descrição

Cria um novo ambiente e implanta uma versão do aplicativo nele.

#### Note

- Para usar o `eb create` em uma aplicação .NET, é necessário criar um pacote de implantação conforme descrito em [Criar pacote de origem para um aplicativo .NET](#) e definir a configuração da CLI para implantar o pacote como um artefato conforme descrito em [Implantar um artefato, em vez da pasta do projeto](#).

- A criação de ambientes com a CLI do EB exige uma [função de serviço](#). É possível criar uma função de serviço criando um ambiente no console do Elastic Beanstalk. Se você não tiver uma função de serviço, a CLI do EB tentará criar uma ao executar `eb create`.

Você pode implantar a versão do aplicativo de algumas origens:

- Por padrão: a partir do código-fonte da aplicação no diretório local do projeto.
- Usando a opção `--version`: a partir de uma versão da aplicação que já existe em sua aplicação.
- Quando o diretório do projeto não tem código de aplicação, ou ao usar a opção `--sample`: a partir de uma aplicação de amostra específica para a plataforma do seu ambiente.

## Sintaxe

```
eb create
```

```
eb create environment-name
```

O nome de um ambiente deve ter entre 4 e 40 caracteres de comprimento. Ele pode conter somente letras, números e hifens (-). Um nome de ambiente não pode começar nem terminar com um hífen.

Se você incluir um nome de ambiente no comando, a EB CLI não pedirá que você faça seleções ou crie uma função de serviço.

Se você executar o comando sem um argumento de nome de ambiente, ele será executado em um fluxo interativo e solicitará que você insira ou selecione valores para algumas configurações. Neste fluxo interativo, caso você esteja implementando um aplicativo de amostra, a EB CLI também pergunta se você deseja fazer download desse aplicativo de amostra no seu diretório local do projeto. Ao fazer download, você pode usar a EB CLI mais tarde com o novo ambiente para executar as operações que exigem o código da aplicação, como [eb deploy](#).

Alguns prompts de fluxo interativo são exibidos somente sob determinadas condições. Por exemplo, se você optar por usar um Application Load Balancer e sua conta tiver pelo menos um Application Load Balancer compartilhável, o Elastic Beanstalk exibirá um prompt perguntando se você deseja usar um balanceador de carga compartilhado. Esse prompt não será exibido se não houver nenhum balanceador de carga da aplicação compartilhável em sua conta.

## Opções

Nenhuma dessas opções é obrigatória. Se você executar `eb create` sem nenhuma opção, a EB CLI solicita que você insira ou selecione um valor para cada configuração.

Nome	Descrição
-d ou --branch_default	Definir o ambiente como o ambiente padrão para o repositório atual.
--cfg <i>config-name</i>	<a href="#">Use as configurações de plataforma de uma configuração salva</a> em <code>.elasticbeanstalk/saved_configs/</code> ou no bucket do Amazon S3. Especifique o nome do arquivo apenas, sem a extensão <code>.cfg.yml</code> .
-c <i>subdomain-name</i> ou --cname <i>subdomain-name</i>	O nome do subdomínio para prefixar a entrada de CNAME DNS que é roteada para seu website.  Tipo: sequência  Padrão: o nome do ambiente
-db ou --database	Anexa um banco de dados ao ambiente. Se você executar <code>eb create</code> com a opção <code>--database</code> , mas sem as opções <code>--database.username</code> e <code>--database.password</code> , a EB CLI solicitará o nome de usuário principal e a senha do banco de dados.
-db.engine <i>engine</i> ou --database.engine <i>engine</i>	O tipo de mecanismo de banco de dados. Se você executar <code>eb create</code> com essa opção, a EB CLI iniciará o ambiente com um banco de dados anexado. Esse será o caso mesmo que você não tenha executado o comando com a opção <code>--database</code> .  Tipo: sequência

Nome	Descrição
	Valores válidos: <code>mysql</code> , <code>oracle-se1</code> , <code>postgres</code> , <code>sqlserver-ex</code> , <code>sqlserver-web</code> , <code>sqlserver-se</code>
<p><code>-db.i <i>instance_type</i></code></p> <p>ou</p> <p><code>--database.instance <i>instance_type</i></code></p>	<p>O tipo de instância do Amazon EC2 para usar no banco de dados. Se você executar <code>eb create</code> com essa opção, a EB CLI iniciará o ambiente com um banco de dados anexado. Esse será o caso mesmo que você não tenha executado o comando com a opção <code>--database</code> .</p> <p>Tipo: sequência</p> <p>Valores válidos:</p> <p>O Amazon RDS é compatível com um conjunto padrão de instâncias de banco de dados. Para selecionar uma instância de banco de dados apropriada para seu mecanismo de banco de dados, você deve levar em consideração algumas considerações específicas. Para obter mais informações, consulte <a href="#">Classes de instâncias de banco de dados</a> no Guia do usuário do Amazon RDS.</p>
<p><code>-db.pass <i>password</i></code></p> <p>ou</p> <p><code>--database.password <i>password</i></code></p>	<p>A senha do banco de dados. Se você executar <code>eb create</code> com essa opção, a EB CLI iniciará o ambiente com um banco de dados anexado. Esse será o caso mesmo que você não tenha executado o comando com a opção <code>--database</code> .</p>

Nome	Descrição
<p><code>-db.size</code> <i>number_of_gigabytes</i></p> <p>ou</p> <p><code>--database.size</code> <i>number_of_gigabytes</i></p>	<p>O número de gigabytes (GB) para alocar ao armazenamento de banco de dados. Se você executar <code>eb create</code> com essa opção, a EB CLI iniciará o ambiente com um banco de dados anexado. Esse será o caso mesmo que você não tenha executado o comando com a opção <code>--database</code> .</p> <p>Tipo: número</p> <p>Valores válidos:</p> <ul style="list-style-type: none"> <li>• MySQL: 5 a 1024. O padrão é 5.</li> <li>• Postgres: 5 a 1024. O padrão é 5.</li> <li>• Oracle: 10 a 1024. O padrão é 10.</li> <li>• Microsoft SQL Server Express Edition – 30.</li> <li>• Microsoft SQL Server Web Edition – 30.</li> <li>• Microsoft SQL Server Standard Edition – 200.</li> </ul>
<p><code>-db.user</code> <i>username</i></p> <p>ou</p> <p><code>--database.username</code> <i>username</i></p>	<p>O nome de usuário para o banco de dados. Se você executar <code>eb create</code> com essa opção, a EB CLI iniciará o ambiente com um banco de dados anexado, mesmo que você não tenha executado o comando com a opção <code>--database</code> . Se você executar <code>eb create</code> com a opção <code>--database</code> , mas sem as opções <code>--database.username</code> e <code>--database.password</code> , a EB CLI solicitará o nome de usuário e a senha principais do banco de dados.</p>
<p><code>-db.version</code> <i>version</i></p> <p>ou</p> <p><code>--database.version</code> <i>versão</i></p>	<p>Usado para especificar a versão do mecanismo de banco de dados. Se esse sinalizador estiver presente, o ambiente será iniciado com um banco de dados com o número de versão especificado, mesmo que o sinalizador <code>--database</code> não esteja presente.</p>



Nome	Descrição
<code>--elb-type</code> <i>type</i>	Novo <a href="#">tipo de load balancer</a> .  Tipo: sequência  Valores válidos: classic, application , network  Padrão: application
<code>-es</code>  ou  <code>--enable-spot</code>	Habilite solicitações de instância spot para seu ambiente. Para obter mais informações, consulte <a href="#">Grupo de Auto Scaling</a> .  Opções relacionadas: <ul style="list-style-type: none"><li>• <code>--instance-types</code></li><li>• <code>--on-demand-base-capacity</code></li><li>• <code>--on-demand-above-base-capacity</code></li><li>• <code>--spot-max-price</code></li></ul>
<code>--env-group-suffix</code> <i>groupname</i>	Nome do grupo para anexar ao nome do ambiente. Somente para uso com <a href="#">ambientes do Compose</a> .
<code>--envvars</code>	<a href="#">Propriedades do ambiente</a> em uma lista separada por vírgula com o formato <i>name=value</i> . Para obter os limites, consulte <a href="#">Configurar propriedades do ambiente (variáveis de ambiente)</a> .
<code>-ip</code> <i>profile_name</i>  ou  <code>--instance_profile</code> <i>profile_name</i>	O perfil da instância com a função do IAM com as credenciais de segurança temporárias das quais sua aplicação precisa para acessar os recursos da AWS.


Nome	Descrição
<p>-it</p> <p>ou</p> <p>--instance-types <i>type1</i>[,<i>type2</i> ...]</p>	<p>Uma lista separada por vírgulas dos tipos de instância do Amazon EC2 que você deseja que o ambiente use. Se você não especificar essa opção, o Elastic Beanstalk fornecerá tipos de instância padrão.</p> <p>Para obter mais informações, consulte <a href="#">Instâncias do Amazon EC2</a> e <a href="#">Grupo de Auto Scaling</a>.</p> <div data-bbox="688 575 1507 989" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #fff9f9;"> <p> <b>Important</b></p> <p>A EB CLI aplica somente essa opção a instâncias spot. A menos que esta opção seja usada com a opção <code>--enable-spot</code>, a EB CLI a ignorará. Para especificar um tipo de instância para uma instância sob demanda, use a opção <code>--instance-type</code> (sem "s").</p> </div>
<p>-i</p> <p>ou</p> <p>--instance_type</p>	<p>O tipo de instância do Amazon EC2 que deseja que o ambiente use. Se você não especificar essa opção, o Elastic Beanstalk fornecerá um tipo de instância padrão.</p> <p>Para obter mais informações, consulte <a href="#">Instâncias do Amazon EC2</a>.</p> <div data-bbox="688 1325 1507 1738" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #fff9f9;"> <p> <b>Important</b></p> <p>A EB CLI aplica essa opção somente a instâncias sob demanda. Não use essa opção com a opção <code>--enable-spot</code>, pois a EB CLI a ignora quando você faz isso. Para especificar tipos de instância para uma instância spot, use a opção <code>--instance-types</code> (sem "s").</p> </div>

Nome	Descrição
<code>-k <i>key_name</i></code> ou <code>--keyname <i>key_name</i></code>	<p>O nome do par de chaves do Amazon EC2 a ser usado com o cliente Secure Shell (SSH) para fazer login com segurança nas instâncias do Amazon EC2 que executam a aplicação do Elastic Beanstalk. Se você incluir essa opção com o comando <code>eb create</code>, o valor fornecido substituirá qualquer nome de chave especificado com <code>eb init</code>.</p> <p>Valores válidos: um nome de chave existente que está registrado no Amazon EC2</p>
<code>-im <i>number-of-instances</i></code> ou <code>--min-instances <i>number-of-instances</i></code>	<p>O número mínimo de instâncias do Amazon EC2 exigidas por você para seu ambiente.</p> <p>Tipo: número (inteiro)</p> <p>Padrão: 1</p> <p>Valores válidos: 1 a 10000</p>
<code>-ix <i>number-of-instances</i></code> ou <code>--max-instances <i>number-of-instances</i></code>	<p>O número máximo de instâncias do Amazon EC2 permitidas por você para seu ambiente.</p> <p>Tipo: número (inteiro)</p> <p>Padrão: 4</p> <p>Valores válidos: 1 a 10000</p>
<code>--modules <i>component-a component-b</i></code>	<p>Uma lista de ambientes de componente para criar. Somente para uso com <a href="#">ambientes do Compose</a>.</p>

Nome	Descrição
<code>-sb</code> ou <code>--on-demand-base-capacity</code>	<p>O número mínimo de instâncias sob demanda que o grupo de Auto Scaling provisiona antes de considerar as instâncias spot à medida que o ambiente é dimensionado.</p> <p>Esta opção só pode ser especificada com a opção <code>--enable-spot</code>. Para obter mais informações, consulte <a href="#">Grupo de Auto Scaling</a>.</p> <p>Tipo: número (inteiro)</p> <p>Padrão: 0</p> <p>Valores válidos: 0 para <code>--max-instances</code> (quando ausente: opção <code>MaxSize</code> no namespace <a href="#">aws:auto-scaling:asg</a>)</p>
<code>-sp</code> ou <code>--on-demand-above-base-capacity</code>	<p>A porcentagem de instâncias sob demanda como parte da capacidade adicional que o grupo de Auto Scaling provisiona além do número de instâncias especificado pela opção <code>--on-demand-base-capacity</code>.</p> <p>Esta opção só pode ser especificada com a opção <code>--enable-spot</code>. Para obter mais detalhes, consulte <a href="#">Grupo de Auto Scaling</a>.</p> <p>Tipo: número (inteiro)</p> <p>Padrão: 0 para um ambiente de uma única instância; 70 para um ambiente com balanceamento de carga</p> <p>Valores válidos: 0 a 100</p>


Nome	Descrição
<p>-p <i>platform-version</i></p> <p>ou</p> <p>--platform <i>platform-version</i></p>	<p>A <a href="#">versão da plataforma</a> a ser usada. É possível especificar uma plataforma, uma plataforma e a versão, a ramificação de uma plataforma, o nome de uma pilha de solução ou o ARN de uma pilha de solução. Por exemplo:</p> <ul style="list-style-type: none"> <li>• php, PHP, node.js: a versão mais recente da plataforma especificada</li> <li>• php-7.2, "PHP 7.2": a versão recomendada da plataforma PHP 7.2 (geralmente a mais recente)</li> <li>• "PHP 7.2 running on 64bit Amazon Linux": a versão da plataforma PHP especificada por esse nome de pilha da solução</li> <li>• "64bit Amazon Linux 2017.09 v2.6.3 running PHP 7.1": a versão da plataforma PHP especificada por esse nome de pilha da solução</li> <li>• "arn:aws:elasticbeanstalk:us-east-2:platform/PHP 7.1 running on 64bit Amazon Linux/2.6.3" : a versão da plataforma PHP especificada por esse ARN de pilha da solução</li> </ul> <p>Use <a href="#">eb platform list</a> para obter uma lista de configurações disponíveis.</p> <p>Se você especificar a opção --platform , ela substituirá o valor que foi fornecido eb init.</p>
<p>-pr</p> <p>ou</p> <p>--process</p>	<p>Pré-processa e valida o manifesto de ambiente e os arquivos de configuração no pacote de origem. A validação dos arquivos de configuração pode identificar problemas antes de implantar a versão do aplicativo em um ambiente.</p>

Nome	Descrição
<code>-r <i>region</i></code> ou <code>--region <i>region</i></code>	A região da AWS na qual você deseja implantar a aplicação.  Para obter a lista de valores que você pode especificar para essa opção, consulte <a href="#">Endpoints e cotas do AWS Elastic Beanstalk</a> no Referência geral da AWS.
<code>--sample</code>	Implante o aplicativo de exemplo no novo ambiente, em vez do código no seu repositório.
<code>--scale <i>number-of-instances</i></code>	Iniciar com o número especificado de instâncias
<code>--service-role <i>servicerole</i></code>	Atribua ao ambiente uma função de serviço diferente do padrão.

 Note

Não insira um ARN. Insira apenas o nome da função. O Elastic Beanstalk insere um prefixo no nome da função com os valores corretos para criar internamente o ARN resultante.

Nome	Descrição
<code>-l <i>load-balancer</i></code> ou <code>--shared-lb <i>load-balancer</i></code>	<p>Configure o ambiente para usar um load balancer compartilhado. Forneça o nome ou o ARN de um balanceador de carga compartilhável em sua conta: um Application Load Balancer criado explicitamente, não um criado por outro ambiente do Elastic Beanstalk. Para obter mais informações, consulte <a href="#">Application Load Balancer compartilhado</a>.</p> <p>Exemplos de parâmetros:</p> <ul style="list-style-type: none"><li>• <code>FrontEndLB</code> : o nome de um balanceador de carga.</li><li>• <code>arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/FrontEndLB/0dbf78d8ad96abbc</code> : ARN de um Application Load Balancer.</li></ul> <p>Você pode especificar essa opção somente com <code>--elb-type application</code> . Se você especificar essa opção e não especificar <code>--shared-lb</code> , o Elastic Beanstalk criará um balanceador de carga dedicado para o ambiente.</p>
<code>-lp <i>port</i></code> ou <code>--shared-lb-port <i>port</i></code>	<p>A porta de listener padrão do balanceador de carga compartilhado para este ambiente. O Elastic Beanstalk adiciona uma regra de listener que encaminha todo o tráfego desse listener para o processo de ambiente padrão. Para obter mais informações, consulte <a href="#">Application Load Balancer compartilhado</a>.</p> <p>Tipo: número (inteiro)</p> <p>Padrão: 80</p> <p>Valores válidos: qualquer número inteiro que represente uma porta de listener do load balancer compartilhado.</p>

Nome	Descrição
<code>--single</code>	<p>Crie o ambiente com uma única instância do Amazon EC2 e sem um balanceador de carga.</p> <div data-bbox="688 352 1507 953" style="border: 1px solid #f08080; padding: 10px;"><p> <b>Warning</b></p><p>Um ambiente de única instância não está pronto para produção. Se a instância tornar-se instável durante a implantação, ou o Elastic Beanstalk encerrar e reiniciar a instância durante uma atualização de configuração, sua aplicação poderá ficar indisponível durante um determinado período. Use ambientes de única instância para desenvolvimento, teste ou preparação. Use ambientes com balanceamento de carga para produção.</p></div>
<code>-sm</code> ou <code>--spot-max-price</code>	<p>O preço máximo por hora, em USD, que você está disposto a pagar por uma instância spot.</p> <p>Esta opção só pode ser especificada com a opção <code>--enable-spot</code>. Para obter mais detalhes, consulte <a href="#">Grupo de Auto Scaling</a>.</p> <p>Tipo: número (flutuante)</p> <p>Padrão: Preço sob demanda, para cada tipo de instância. O valor da opção neste caso é <code>null</code>.</p> <p>Valores válidos: <code>0.001</code> a <code>20.0</code></p> <p>Para obter recomendações sobre opções de preço máximo para Instâncias Spot, consulte <a href="#">Histórico de definição de preço da instância spot</a> no Manual do usuário do Amazon EC2 para instâncias do Linux.</p>



Nome	Descrição
<code>--tags</code> <i>key1=value1[,key2=value2]</i>	<p>Marque os recursos no seu ambiente. As tags são especificadas como uma lista de <code>key=value</code> separados por vírgulas.</p> <p>Para obter mais informações, consulte <a href="#">Marcação em ambientes</a>.</p>
<code>-t worker</code> ou <code>--tier worker</code>	Crie um ambiente de operador. Omita essa opção para criar um ambiente de servidor web.
<code>--timeout</code> <i>minutos</i>	O número definido de minutos antes do limite de tempo do comando.
<code>--version</code> <i>version_label</i>	<p>Especifica a versão do aplicativo que você deseja implantar no ambiente, em vez do código-fonte do aplicativo no diretório do projeto local.</p> <p>Tipo: sequência</p> <p>Valores válidos: um rótulo de versão do aplicativo existente</p>
<code>--vpc</code>	Configure uma VPC em seu ambiente. Quando você incluir essa opção, a EB CLI solicitará que você insira todas as configurações obrigatórias antes de iniciar o ambiente.
<code>--vpc.dbsubnets</code> <i>subnet1,subnet2</i>	Especifica sub-redes para instâncias de banco de dados em uma VPC. Obrigatório quando <code>--vpc.id</code> for especificado.
<code>--vpc.ec2subnets</code> <i>subnet1,subnet2</i>	Especifica sub-redes para instâncias do Amazon EC2 em uma VPC. Obrigatório quando <code>--vpc.id</code> for especificado.

Nome	Descrição
<code>--vpc.elbpublic</code>	Inicia o balanceador de carga do Elastic Load Balancing em uma sub-rede pública em sua VPC.  Não é possível especificar essa opção com <code>--tier worker</code> e <code>--single</code> .
<code>--vpc.elbsubnets</code> <i>subnet1, subnet2</i>	Especifica sub-redes para o balanceador de carga do Elastic Load Balancing em uma VPC.  Não é possível especificar essa opção com <code>--tier worker</code> e <code>--single</code> .
<code>--vpc.id</code> <i>ID</i>	Inicia seu ambiente na VPC especificada.
<code>--vpc.publicip</code>	Inicia suas instâncias do Amazon EC2 em uma sub-rede pública em sua VPC.  Não é possível especificar essa opção com <code>--tier worker</code> .
<code>--vpc.securitygroups</code> <i>securitygroup1, securitygroup2</i>	Especifica IDs de grupo de segurança. Obrigatório quando <code>--vpc.id</code> for especificado.
<a href="#">Opções comuns</a>	

## Resultado

Se for bem-sucedido, o comando apresentará prompts com perguntas e retornará o status da operação de criação. Se houver problemas durante a inicialização, você poderá usar a operação [eb events](#) para obter mais detalhes.

Se você habilitou o suporte do CodeBuild à aplicação, `eb create` exibirá as informações do CodeBuild enquanto o código é compilado. Para obter informações sobre o suporte ao CodeBuild no Elastic Beanstalk, consulte [Usar o EB CLI com o AWS CodeBuild](#).

## Exemplos

O exemplo a seguir cria um ambiente no modo interativo.

```
$ eb create
Enter Environment Name
(default is tmp-dev): ENTER
Enter DNS CNAME prefix
(default is tmp-dev): ENTER
Select a load balancer type
1) classic
2) application
3) network
(default is 2): ENTER
Environment details for: tmp-dev
 Application name: tmp
 Region: us-east-2
 Deployed Version: app-141029_145448
 Environment ID: e-um3yfrzq22
 Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
 Tier: WebServer-Standard-1.0
 CNAME: tmp-dev.elasticbeanstalk.com
 Updated: 2014-10-29 21:54:51.063000+00:00
Printing Status:
...
```

O exemplo a seguir também cria um ambiente no modo interativo. Neste exemplo, o diretório do projeto não tem código de aplicativo. O comando implanta um aplicativo de amostra e faz o download deste no diretório local do projeto.

```
$ eb create
Enter Environment Name
(default is tmp-dev): ENTER
Enter DNS CNAME prefix
(default is tmp-dev): ENTER
Select a load balancer type
1) classic
2) application
3) network
(default is 2): ENTER
NOTE: The current directory does not contain any source code. Elastic Beanstalk is
 launching the sample application instead.
```

```
Do you want to download the sample application into the current directory?
(Y/n): ENTER
INFO: Downloading sample application to the current directory.
INFO: Download complete.
Environment details for: tmp-dev
 Application name: tmp
 Region: us-east-2
 Deployed Version: Sample Application
 Environment ID: e-um3yfrzq22
 Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
 Tier: WebServer-Standard-1.0
 CNAME: tmp-dev.elasticbeanstalk.com
 Updated: 2017-11-08 21:54:51.063000+00:00
Printing Status:
...
```

O seguinte comando cria um ambiente sem exibir prompts.

```
$ eb create dev-env
Creating application version archive "app-160312_014028".
Uploading test/app-160312_014028.zip to S3. This may take a while.
Upload Complete.
Application test has been created.
Environment details for: dev-env
 Application name: test
 Region: us-east-2
 Deployed Version: app-160312_014028
 Environment ID: e-6fgpkjxyyi
 Platform: 64bit Amazon Linux 2015.09 v2.0.8 running PHP 5.6
 Tier: WebServer-Standard
 CNAME: UNKNOWN
 Updated: 2016-03-12 01:40:33.614000+00:00
Printing Status:
...
```

O seguinte comando cria um ambiente em uma VPC personalizada.

```
$ eb create dev-vpc --vpc.id vpc-0ce8dd99 --vpc.elbsubnets subnet-
b356d7c6,subnet-02f74b0c --vpc.ec2subnets subnet-0bb7f0cd,subnet-3b6697c1 --
vpc.securitygroup sg-70cff265
Creating application version archive "app-160312_014309".
Uploading test/app-160312_014309.zip to S3. This may take a while.
Upload Complete.
```

```
Environment details for: dev-vpc
Application name: test
Region: us-east-2
Deployed Version: app-160312_014309
Environment ID: e-pqkqip3mns
Platform: 64bit Amazon Linux 2015.09 v2.0.8 running Java 8
Tier: WebServer-Standard
CNAME: UNKNOWN
Updated: 2016-03-12 01:43:14.057000+00:00
Printing Status:
...
```

## eb deploy

### Descrição

Implanta o pacote de origem do aplicativo, que se encontra no diretório do projeto inicializado, no aplicativo em execução.

Se o git for instalado, a EB CLI usará o comando `git archive` para criar um arquivo `.zip` do conteúdo do comando `git commit` mais recente.

No entanto, quando `.ebignore` está presente no diretório do projeto, a CLI do EB não usa semântica e comandos `git` para criar o pacote de origem. Isso significa que a EB CLI ignora os arquivos especificados em `.ebignore` e inclui todos os outros. Especificamente, ela inclui os arquivos de origem não confirmados.

#### Note

Você pode configurar a EB CLI para implantar um artefato do seu processo de compilação, em vez de criar um arquivo ZIP da pasta do projeto. Para mais detalhes, consulte [Implantar um artefato, em vez da pasta do projeto](#).

### Sintaxe

```
eb deploy
```

```
eb deploy environment-name
```

## Opções

Name (Nome)	Descrição
-l <i>version_label</i> ou --label <i>version_label</i>	Especifique um rótulo a ser usado para a versão criada pela EB CLI. Se o rótulo já tiver sido usado, a EB CLI reimplanta a versão anterior com esse rótulo.  Tipo: sequência
--env-group-suffix <i>groupname</i>	Nome do grupo para anexar ao nome do ambiente. Somente para uso com <a href="#">ambientes do Compose</a> .
-m " <i>version_description</i> " ou --message " <i>version_description</i> "	A descrição da versão do aplicativo, entre aspas duplas.  Tipo: sequência
--modules <i>component-a</i> <i>component-b</i>	Lista de componentes para atualizar. Somente para uso com <a href="#">ambientes do Compose</a> .
-p ou --process	Pré-processa e valida o manifesto de ambiente e os arquivos de configuração no pacote de origem. A validação dos arquivos de configuração pode identificar problemas antes de implantar a versão do aplicativo em um ambiente.
--source codecommit/ <i>repository-name/branch-name</i>	Repositório do CodeCommit e ramificação. Consulte <a href="#">Usar o EB CLI com o AWS CodeCommit</a> .
--staged	Implanta arquivos preparados no índice GIT, em vez da confirmação HEAD.
--timeout <i>minutos</i>	O número de minutos antes de esgotar o tempo limite do comando.

Name (Nome)	Descrição
<code>--version <i>version_1</i></code> <code><i>abel</i></code>	Uma versão do aplicativo existente para implantar. Tipo: sequência
<a href="#">Opções comuns</a>	

## Resultado

Se for bem-sucedido, o comando retornará o status da operação `deploy`.

Se você habilitou o suporte do CodeBuild à aplicação, `eb deploy` exibirá as informações do CodeBuild enquanto o código é compilado. Para obter informações sobre o suporte ao CodeBuild no Elastic Beanstalk, consulte [Usar o EB CLI com o AWS CodeBuild](#).

## Exemplo

O exemplo a seguir implanta o aplicativo atual.

```
$ eb deploy
2018-07-11 21:05:22 INFO: Environment update is starting.
2018-07-11 21:05:27 INFO: Deploying new version to instance(s).
2018-07-11 21:05:53 INFO: New application version was deployed to running EC2
instances.
2018-07-11 21:05:53 INFO: Environment update completed successfully.
```

## eb events

### Descrição

Retorna os eventos mais recentes do ambiente.

Se o diretório raiz contiver um arquivo `platform.yaml` especificando uma plataforma personalizada, esse comando também retornará os eventos mais recentes para o ambiente do criador.

### Sintaxe

```
eb events
```

eb events *environment-name*

## Opções

Name (Nome)	Descrição
-f ou --follow	Transmite eventos por stream. Para cancelar, pressione CTRL +C.

## Resultado

Se for bem-sucedido, o comando retornará os eventos recentes.

## Exemplo

O exemplo a seguir retorna os eventos mais recentes.

```
$ eb events
2014-10-29 21:55:39 INFO createEnvironment is starting.
2014-10-29 21:55:40 INFO Using elasticbeanstalk-us-west-2-111122223333 as Amazon
S3 storage bucket for environment data.
2014-10-29 21:55:57 INFO Created load balancer named: awseb-e-r-AWSEBLoa-
NSKU0K5X6Z9J
2014-10-29 21:56:16 INFO Created security group named: awseb-e-rxgrhjr9bx-stack-
AWSEBSecurityGroup-1UUHU5LZ20ZY7
2014-10-29 21:57:18 INFO Waiting for EC2 instances to launch. This may take a
few minutes.
2014-10-29 21:57:18 INFO Created Auto Scaling group named: awseb-e-rxgrhjr9bx-
stack-AWSEBAutoScalingGroup-1TE320ZCJ9RPD
2014-10-29 21:57:22 INFO Created Auto Scaling group policy named:
arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:2cced9e6-859b-421a-
be63-8ab34771155a:autoScalingGroupName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingGroup-1TE320ZCJ9RPD:policyName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingScaleUpPolicy-1I2ZSNVU4APRY
2014-10-29 21:57:22 INFO Created Auto Scaling group policy named:
arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:1f08b863-
bf65-415a-b584-b7fa3a69a0d5:autoScalingGroupName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingGroup-1TE320ZCJ9RPD:policyName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingScaleDownPolicy-1E3G7PZKZPS0G
```



```

2014-10-29 21:57:25 INFO Created CloudWatch alarm named: awseb-e-rxgrhjr9bx-
stack-AWSEBCloudwatchAlarmLow-VF5EJ549FZBL
2014-10-29 21:57:25 INFO Created CloudWatch alarm named: awseb-e-rxgrhjr9bx-
stack-AWSEBCloudwatchAlarmHigh-LA9YEW306WJ0
2014-10-29 21:58:50 INFO Added EC2 instance 'i-c7ee492d' to Auto ScalingGroup
'awseb-e-rxgrhjr9bx-stack-AWSEBAutoScalingGroup-1TE320ZCJ9RPD'.
2014-10-29 21:58:53 INFO Successfully launched environment: tmp-dev
2014-10-29 21:59:14 INFO Environment health has been set to GREEN
2014-10-29 21:59:43 INFO Adding instance 'i-c7ee492d' to your environment.

```

## eb health

### Descrição

Retorna a integridade mais recente do ambiente.

Se o diretório raiz contiver um arquivo `platform.yaml` especificando uma plataforma personalizada, esse comando também retornará a integridade mais recente do ambiente do compilador.

### Sintaxe

```
eb health
```

```
eb health environment-name
```

### Opções

Name (Nome)	Descrição
<code>-r</code> ou <code>--refresh</code>	Mostra informações de integridade interativamente e atualiza a cada dez segundos à medida que novas informações são relatadas.
<code>--mono</code>	Não exibe cor na saída.

### Resultado

Se for bem-sucedido, o comando retornará a integridade recente.

## Exemplo

O exemplo a seguir retorna as informações de integridade mais recentes para um ambiente Linux.

```
~/project $ eb health
elasticBeanstalkExa-env Ok
2015-07-08 23:13:20
WebServer
Ruby 2.1 (Puma)
total ok warning degraded severe info pending unknown
 5 5 0 0 0 0 0 0

instance-id status cause
health
Overall Ok
i-d581497d Ok
i-d481497c Ok
i-136e00c0 Ok
i-126e00c1 Ok
i-8b2cf575 Ok

instance-id r/sec %2xx %3xx %4xx %5xx p99 p90 p75 p50
p10
requests
Overall 671.8 100.0 0.0 0.0 0.0 0.003 0.002 0.001 0.001
0.000
i-d581497d 143.0 1430 0 0 0 0.003 0.002 0.001 0.001
0.000
i-d481497c 128.8 1288 0 0 0 0.003 0.002 0.001 0.001
0.000
i-136e00c0 125.4 1254 0 0 0 0.004 0.002 0.001 0.001
0.000
i-126e00c1 133.4 1334 0 0 0 0.003 0.002 0.001 0.001
0.000
i-8b2cf575 141.2 1412 0 0 0 0.003 0.002 0.001 0.001
0.000

instance-id type az running load 1 load 5 user% nice% system%
idle% iowait%
cpu
i-d581497d t2.micro 1a 12 mins 0.0 0.04 6.2 0.0 1.0
92.5 0.1
i-d481497c t2.micro 1a 12 mins 0.01 0.09 5.9 0.0 1.6
92.4 0.1
i-136e00c0 t2.micro 1b 12 mins 0.15 0.07 5.5 0.0 0.9
93.2 0.0
```

```

i-126e00c1 t2.micro 1b 12 mins 0.17 0.14 5.7 0.0 1.4
92.7 0.1
i-8b2cf575 t2.micro 1c 1 hour 0.19 0.08 6.5 0.0 1.2
92.1 0.1

```

```

instance-id status id version ago
 deployments
i-d581497d Deployed 1 Sample Application 12 mins
i-d481497c Deployed 1 Sample Application 12 mins
i-136e00c0 Deployed 1 Sample Application 12 mins
i-126e00c1 Deployed 1 Sample Application 12 mins
i-8b2cf575 Deployed 1 Sample Application 1 hour

```

## eb init

### Descrição

Define valores padrão para aplicações do Elastic Beanstalk criados com a CLI do EB pedindo que você responda a uma série de perguntas.

#### Note

Os valores definidos com `eb init` se aplicam somente ao diretório atual e ao repositório no computador atual.

O comando não cria nada na sua conta do Elastic Beanstalk. Para criar um ambiente do Elastic Beanstalk, execute [eb create](#) após a execução de `eb init`.

### Sintaxe

```
eb init
```

```
eb init application-name
```


### Opções

Se você executar `eb init` sem especificar a opção `--platform`, a CLI do EB solicita que você insira um valor para cada configuração.

**Note**

Para usar `eb init` para criar um novo par de chaves, instale `ssh-keygen` em sua máquina local e disponibilize-o na linha de comando.

Nome	Descrição
<p><code>-i</code></p> <p><code>--interactive</code></p>	<p>Força a EB CLI a pedir que você forneça um valor para cada opção de comando <code>eb init</code>.</p> <div data-bbox="548 703 672 743" data-label="Section-Header"><b>Note</b></div> <p>O comando <code>init</code> pede que você forneça valores para as opções do comando <code>eb init</code> que não têm um valor (padrão). Após a primeira execução do comando <code>eb init</code> em um diretório, a EB CLI pode não perguntar sobre nenhuma opção de comando. Portanto, use a opção <code>--interactive</code> quando desejar alterar uma configuração previamente definida.</p>
<p><code>-k <i>keyname</i></code></p> <p><code>--keyname <i>keyname</i></code></p>	<p>O nome do par de chaves do Amazon EC2 a ser usado com o cliente Secure Shell (SSH) para fazer login com segurança nas instâncias do Amazon EC2 que executam a aplicação do Elastic Beanstalk.</p>
<p><code>--modules <i>folder-1</i></code> <code><i>folder-2</i></code></p>	<p>Lista de diretórios filho para inicializar. Somente para uso com <a href="#">ambientes do Compose</a>.</p>
<p><code>-p <i>platform-version</i></code></p> <p><code>--platform <i>platform-version</i></code></p>	<p>A <a href="#">versão da plataforma</a> a ser usada. É possível especificar uma plataforma, uma plataforma e a versão, a ramificação de uma plataforma, o nome de uma pilha de solução ou o ARN de uma pilha de solução. Por exemplo:</p>

Nome	Descrição	
	<ul style="list-style-type: none"> <li>• php, PHP, node.js – a versão mais recente da plataforma especificada</li> <li>• php-7.2, "PHP 7.2": a versão recomendada da plataforma PHP 7.2 (geralmente a mais recente)</li> <li>• "PHP 7.2 running on 64bit Amazon Linux": a versão da plataforma PHP especificada por esse nome de pilha da solução</li> <li>• "64bit Amazon Linux 2017.09 v2.6.3 running PHP 7.1": a versão da plataforma PHP especificada por esse nome de pilha da solução</li> <li>• "arn:aws:elasticbeanstalk:us-east-2::platform/PHP 7.1 running on 64bit Amazon Linux/2.6.3" : a versão da plataforma PHP especificada por esse ARN de pilha da solução</li> </ul> <p>Use <a href="#">eb platform list</a> para obter uma lista de configurações disponíveis.</p> <p>Especifique a opção <code>--platform</code> para ignorar a configuração interativa.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Quando você especificar essa opção, a EB CLI não solicitará valores para as demais opções. Em vez disso, ele usa valores padrão para cada opção. É possível definir todas as opções para as quais não deseja usar valores padrão.</p> </div>	
<code>--source codecommit/ <i>repository-name/branch-name</i></code>	Repositório do CodeCommit e ramificação. Consulte <a href="#">Usar o EB CLI com o AWS CodeCommit</a> .	

Nome	Descrição
<code>--tags</code> <i>key1=value1</i>	Marque seu aplicativo. As tags são especificadas como uma lista de <code>key=value</code> separados por vírgulas.  Para obter mais detalhes, consulte <a href="#">Marcar aplicativos</a> .
<a href="#">Opções comuns</a>	

## Suporte a CodeBuild

Se você executar `eb init` em uma pasta com um arquivo [buildspec.yml](#) o Elastic Beanstalk analisará o arquivo para uma entrada `eb_codebuild_settings` com opções específicas do Elastic Beanstalk. Para obter informações sobre o suporte ao CodeBuild no Elastic Beanstalk, consulte [Usar o EB CLI com o AWS CodeBuild](#).

## Resultado

Se for bem-sucedido, o comando orientará você na configuração de uma nova aplicação do Elastic Beanstalk por meio de uma série de prompts.

## Exemplo

A solicitação de exemplo a seguir inicializa a EB CLI e pede que você insira informações sobre o aplicativo. Substitua o texto do *espaço reservado* por seus próprios valores.

```
$ eb init -i
Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : Europe (Ireland)
5) eu-central-1 : Europe (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
...
(default is 3): 3

Select an application to use
1) HelloWorldApp
```

```
2) NewApp
3) [Create new Application]
(default is 3): 3

Enter Application Name
(default is "tmp"):
Application tmp has been created.

It appears you are using PHP. Is this correct?
(y/n): y

Select a platform branch.
1) PHP 7.2 running on 64bit Amazon Linux
2) PHP 7.1 running on 64bit Amazon Linux (Deprecated)
3) PHP 7.0 running on 64bit Amazon Linux (Deprecated)
4) PHP 5.6 running on 64bit Amazon Linux (Deprecated)
5) PHP 5.5 running on 64bit Amazon Linux (Deprecated)
6) PHP 5.4 running on 64bit Amazon Linux (Deprecated)
(default is 1): 1

Do you want to set up SSH for your instances?
(y/n): y

Select a keypair.
1) aws-eb
2) [Create new KeyPair]
(default is 2): 1
```

## eb labs

### Descrição

Os subcomandos de `eb labs` oferecem suporte ao trabalho em andamento ou à funcionalidade experimental. Esses comandos podem ser removidos ou retrabalhados em futuras versões da EB CLI, sem garantia de compatibilidade.

Para obter uma lista dos subcomandos e descrições disponíveis, execute `eb labs --help`.

### eb list

#### Descrição

Lista todos os ambientes no aplicativo atual ou todos os ambientes em todos os aplicativos, conforme especificado pela opção `--all`.

Se o diretório raiz contém um arquivo `platform.yaml` especificando uma plataforma personalizada, este comando também lista o ambiente do compilador.

## Sintaxe

```
eb list
```

## Opções

Name (Nome)	Descrição
<code>-a</code> ou <code>--all</code>	Lista todos os ambientes de todos os aplicativos.
<code>-v</code> ou <code>--verbose</code>	Fornecer informações mais detalhadas sobre todos os ambientes, incluindo instâncias.
<a href="#">Opções comuns</a>	

## Resultado

Se for bem-sucedido, o comando retornará uma lista de nomes de ambiente na qual seu ambiente atual é marcado com um asterisco (\*).

## Exemplo 1

O exemplo a seguir relaciona os seus ambientes e indica que `tmp-dev` é o seu ambiente padrão.

```
$ eb list
* tmp-dev
```

## Exemplo 2

O exemplo a seguir relaciona seus ambientes com detalhes adicionais.



```
$ eb list --verbose
Region: us-west-2
Application: tmp
 Environments: 1
 * tmp-dev : ['i-c7ee492d']
```

## eb local

### Descrição

Use `eb local run` para executar os contêineres de seu aplicativo localmente no Docker. Verifique o status do contêiner do aplicativo com `eb local status`. Abra o aplicativo em um navegador web com `eb local open`. Recupere a localização de logs do aplicativo com `eb local logs`.

`eb local setenv` e `eb local printenv` permitem que você defina e visualize variáveis de ambiente que são fornecidas aos contêineres do Docker que você executa localmente com `eb local run`.

É necessário executar todos os comandos `eb local` no diretório de projeto de um aplicativo do Docker que foi inicializado como um repositório da EB CLI usando `eb init`.

#### Note

Use `eb local` em um computador local executando Linux ou macOS. O comando não oferece suporte ao Windows.

Antes de usar o comando no macOS, instale o Docker para Mac e verifique se o `boot2docker` não está instalado (ou se não está no caminho de execução). O comando `eb local` tenta usar o `boot2docker` se estiver presente, mas não funciona bem com ele no macOS.

### Sintaxe

`eb local run`

`eb local status`

`eb local open`

`eb local logs`

`eb local setenv`

eb local printenv

## Opções

eb local run

Name (Nome)	Descrição
<code>--envvars <i>key1=value1, key2=value2</i></code>	Define variáveis do ambiente que a EB CLI passará para os contêineres do Docker local. Em ambientes de vários contêineres, todas as variáveis são passadas para todos os contêineres.
<code>--port <i>hostport</i></code>	Mapeia uma porta no host para a porta exposta no contêiner. Se você não especificar essa opção, a EB CLI usa a mesma porta no host e no contêiner.  Essa opção funciona somente com aplicativos da plataforma do Docker. Ela não se aplica à plataforma Docker de vários contêineres.
<a href="#">Opções comuns</a>	

eb local status

eb local open

eb local logs

eb local setenv

eb local printenv

Nome	Descrição
<a href="#">Opções comuns</a>	

## Resultado

eb local run

Mensagens de status do Docker. Permanece ativa enquanto o aplicativo estiver em execução. Pressione Ctrl+C para interromper o aplicativo.

`eb local status`

O status de cada contêiner usado pelo aplicativo, em execução ou não.

`eb local open`

Abre o aplicativo em um navegador web e sai.

`eb local logs`

A localização dos logs gerados no diretório do projeto por aplicativos em execução localmente em `eb local run`.

`eb local setenv`

Nenhum

`eb local printenv`

O nome e os valores de variáveis de ambiente definidas com `eb local setenv`.

## Exemplos

`eb local run`

```
~/project$ eb local run
Creating elasticbeanstalk_phpapp_1...
Creating elasticbeanstalk_nginxproxy_1...
Attaching to elasticbeanstalk_phpapp_1, elasticbeanstalk_nginxproxy_1
phpapp_1 | [23-Apr-2015 23:24:25] NOTICE: fpm is running, pid 1
phpapp_1 | [23-Apr-2015 23:24:25] NOTICE: ready to handle connections
```

`eb local status`

Visualize o status de seus contêineres locais:

```
~/project$ eb local status
Platform: 64bit Amazon Linux 2014.09 v1.2.1 running Multi-container Docker 1.3.3
(Generic)
Container name: elasticbeanstalk_nginxproxy_1
```

```
Container ip: 127.0.0.1
Container running: True
Exposed host port(s): 80
Full local URL(s): 127.0.0.1:80

Container name: elasticbeanstalk_phpapp_1
Container ip: 127.0.0.1
Container running: True
Exposed host port(s): None
Full local URL(s): None
```

## eb local logs

Veja o caminho do log do projeto atual:

```
~/project$ eb local logs
Elastic Beanstalk will write logs locally to /home/user/project/.elasticbeanstalk/logs/local.
Logs were most recently created 3 minutes ago and written to /home/user/project/.elasticbeanstalk/logs/local/150420_234011665784.
```

## eb local setenv

Defina as variáveis do ambiente que serão utilizadas com eb local run.

```
~/project$ eb local setenv PARAM1=value
```

Imprima as variáveis do ambiente definidas com eb local setenv.

```
~/project$ eb local printenv
Environment Variables:
PARAM1=value
```

## eb logs

### Descrição

O comando `eb logs` tem duas finalidades distintas: habilitar ou desabilitar o streaming de log para o CloudWatch Logs e recuperar logs de instância ou logs do CloudWatch Logs. Com a opção `--cloudwatch-logs (-cw)`, o comando habilita ou desabilita o streaming de logs. Sem essa opção, ele recupera logs.

Ao recuperar logs, especifique a opção `--all`, `--zip` ou `--stream` para recuperar logs completos. Se você não especificar nenhuma dessas opções, o Elastic Beanstalk recuperará logs finais.

O comando processa logs do ambiente especificado ou do ambiente padrão. Os logs relevantes variam por tipo de contêiner. Se o diretório raiz contiver um arquivo `platform.yaml` especificando uma plataforma personalizada, esse comando também processará logs do ambiente do compilador.

Para obter mais informações, consulte [the section called “CloudWatch Logs”](#).

## Sintaxe

Como habilitar ou desabilitar o streaming de logs para o CloudWatch Logs:

```
eb logs --cloudwatch-logs [enable | disable] [--cloudwatch-log-source instance |
environment-health | all] [environment-name]
```

Para recuperar logs de instâncias:

```
eb logs [-all | --zip | --stream] [--cloudwatch-log-source instance] [--
instance instance-id] [--log-group log-group] [environment-name]
```

Para recuperar logs de integridade do ambiente:


```
eb logs [-all | --zip | --stream] --cloudwatch-log-source environment-health
[environment-name]
```

## Opções

Name (Nome)	Descrição
<code>-cw [enable   disable]</code> ou <code>--cloudwatch-logs [enable   disable]</code>	Habilita ou desabilita o streaming de logs para o CloudWatch Logs. Se nenhum argumento for fornecido, o streaming de logs será habilitado. Se a opção <code>--cloudwatch-log-source</code> ( <code>-cls</code> ) não for especificada, o streaming de logs de instância será habilitado ou desabilitado.
<code>-cls instance   environment-health   all</code>	Especifica a origem dos logs ao trabalhar com o CloudWatch Logs. Com o formulário de habilitação ou desabilitação do comando, esses são os logs para os quais habilitar ou desabilit

Name (Nome)	Descrição
<p>ou</p> <pre>--cloudwatch-log-source instance   environment-health   all</pre>	<p>Carregar o streaming do CloudWatch Logs. Com o formulário de recuperação do comando, esses são os logs a serem recuperados do CloudWatch Logs.</p> <p>Valores válidos:</p> <ul style="list-style-type: none"> <li>• Com o <code>--cloudwatch-logs (enable ou disable) - instance   environment-health   all</code></li> <li>• Sem <code>--cloudwatch-logs (retrieve) - instance   environment-health</code></li> </ul> <p>Significados dos valores:</p> <ul style="list-style-type: none"> <li>• <code>instance</code> (padrão) – logs de instâncias</li> <li>• <code>environment-health</code> – logs de integridade do ambiente (compatível somente quando a integridade avançada está habilitada no ambiente)</li> <li>• <code>all</code> - as duas origens do log</li> </ul>
<pre>-a</pre> <p>ou</p> <pre>--all</pre>	<p>Recupera logs completos e salva-os no diretório <code>.elasticbeanstalk/logs</code>.</p>
<pre>-z</pre> <p>ou</p> <pre>--zip</pre>	<p>Recupera logs completos, compacta-os em um arquivo <code>.zip</code> e, em seguida, salva o arquivo no diretório <code>.elasticbeanstalk/logs</code>.</p>
<pre>--stream</pre>	<p>Transmite (saídas contínuas) logs completos. Com essa opção, o comando continua a ser executado até que você o interrompa (pressione <b>Ctrl+C</b>).</p>

Name (Nome)	Descrição
<code>-i <i>instance-id</i></code>  ou  <code>--instance <i>instance-id</i></code>	Recupera apenas os logs de uma instância especificada.

Name (Nome)	Descrição
<p>-g <i>log-group</i></p> <p>ou</p> <p>--log-group <i>log-group</i></p>	<p>Especifica o grupo de registros do CloudWatch Logs a partir do qual recuperar logs. A opção é válida somente quando o streaming de logs de instância para o CloudWatch Logs está habilitado.</p> <p>Se o streaming de log de instância estiver habilitado, e você não especificar a opção <code>--log-group</code>, o grupo de logs padrão será um dos seguintes:</p> <ul style="list-style-type: none"><li>• Amazon Linux 2: <code>/aws/elasticbeanstalk/<i>environment-name</i>/var/log/eb-engine.log</code></li><li>• Plataformas Windows – <code>/aws/elasticbeanstalk/<i>environment-name</i>/EBDeploy-Log</code></li><li>• Amazon Linux AMI (AL1): <code>/aws/elasticbeanstalk/<i>environment-name</i>/var/log/eb-activity.log</code></li></ul> <div data-bbox="625 1010 1507 1514" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Em <a href="#">18 de julho de 2022</a>, o Elastic Beanstalk definiu o status de todas as ramificações de plataforma a baseadas no Amazon Linux AMI (AL1) como descontinuadas. Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte <a href="#">Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2</a>.</p></div> <p>Para obter informações sobre o grupo de logs correspondente a cada arquivo de log, visite <a href="#">Como o Elastic Beanstalk configura o CloudWatch Logs</a>.</p>
<a href="#">Opções comuns</a>	



## Resultado

Por padrão, exibe os logs diretamente no terminal. Usa um programa de paginação para exibir a saída. Pressione **Q** ou **q** para sair.

Com `--stream`, mostra os logs existentes no terminal e continua em execução. Pressione **Ctrl+C** para sair.

Com `--all` e `--zip`, salva os logs em arquivos locais e exibe o local do arquivo.

## Exemplos

O exemplo a seguir permite o streaming de logs de instância para o CloudWatch Logs.

```
$ eb logs -cw enable
Enabling instance log streaming to CloudWatch for your environment
After the environment is updated you can view your logs by following the link:
https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#logs:prefix=/aws/
elasticbeanstalk/environment-name/
Printing Status:
2018-07-11 21:05:20 INFO: Environment update is starting.
2018-07-11 21:05:27 INFO: Updating environment environment-name's configuration
settings.
2018-07-11 21:06:45 INFO: Successfully deployed new configuration to environment.
```

O exemplo a seguir recupera os logs de instância em um arquivo `.zip`.

```
$ eb logs --zip
Retrieving logs...
Logs were saved to /home/workspace/environment/.elasticbeanstalk/logs/150622_173444.zip
```

## eb open

### Descrição

Abre o URL público do site no navegador padrão.

### Sintaxe

```
eb open
```

eb open *environment-name*

## Opções

Name (Nome)	Descrição
<a href="#">Opções comuns</a>	

## Resultado

O comando eb open não gera saída. Em vez disso, ele abre o aplicativo em uma janela do navegador.

## eb platform

### Descrição

Este comando dá suporte a dois espaços de trabalho diferentes:

#### [Plataforma](#)

Use este espaço de trabalho para gerenciar plataformas personalizadas.

#### [Ambiente](#)

Use este espaço de trabalho para selecionar uma plataforma padrão ou mostrar informações sobre a plataforma atual.

O Elastic Beanstalk fornece o atalho ebp para eb platform.

#### Note

O Windows PowerShell usa ebp como alias de comando. Se estiver executando a CLI do EB no Windows PowerShell, utilize a forma longa deste comand — eb platform.

## Usar eb platform para plataformas personalizadas

Lista as versões da plataforma atual e permite que você gerencie plataformas personalizadas.

## Sintaxe

```
eb platform create [version] [options]
```

```
eb platform delete [version] [options]
```

```
eb platform events [version] [options]
```

```
eb platform init [platform] [options]
```

```
eb platform list [options]
```

```
eb platform logs [version] [options]
```

```
eb platform status [version] [options]
```

```
eb platform use [platform] [options]
```

## Opções

Name (Nome)	Descrição
create [ <i>version</i> ] [ <i>options</i> ]	Criar uma nova versão da plataforma. <a href="#">Saiba mais</a> .
delete <i>version</i> [ <i>options</i> ]	Excluir uma versão da plataforma. <a href="#">Saiba mais</a> .
events [ <i>version</i> ] [ <i>options</i> ]	Exibir os eventos de uma versão da plataforma. <a href="#">Saiba mais</a> .
init [ <i>platform</i> ] [ <i>options</i> ]	Inicializar um repositório de plataforma. <a href="#">Saiba mais</a> .
list [ <i>options</i> ]	Listar as versões da plataforma atual. <a href="#">Saiba mais</a> .
logs [ <i>version</i> ] [ <i>options</i> ]	Exibir logs do ambiente do compilador para uma versão da plataforma. <a href="#">Saiba mais</a> .
status [ <i>version</i> ] [ <i>options</i> ]	Exibir o status de uma versão da plataforma. <a href="#">Saiba mais</a> .

Name (Nome)	Descrição
use [ <i>platform</i> ] [ <i>options</i> ]	Selecionar uma plataforma diferente a partir da qual novas versões são criadas. <a href="#">Saiba mais</a> .
<a href="#">Opções comuns</a>	

## Opções comuns

Todos os comandos eb platform incluem as seguintes opções comuns.

Nome	Descrição
-h OU --help	Mostra uma mensagem de ajuda e sai.
--debug	Mostra saída de depuração adicional.
--quiet	Suprime toda a saída.
-v OU --verbose	Mostra saída adicional.
--profile <i>PROFILE</i>	Usa o <i>PERFIL</i> especificado em suas credenciais.
-r <i>REGION</i> OU --region <i>REGION</i>	Usa a região <i>REGIÃO</i> .
--no-verify-ssl	Não verifica os certificados SSL da AWS

## Eb platform create

Cria uma nova versão da plataforma e retorna o nome de região da Amazon (ARN) da nova versão. Se não houver um ambiente de compilador em execução na região atual, o comando iniciará uma. A *versão* e as opções de incremento (-M, -m e -p) são mutualmente exclusivas.

### Opções

Name (Nome)	Descrição
<i>versão</i>	Se a <i>versão</i> não estiver especificada, cria uma nova versão de acordo com a Plataforma mais recente com a versão de patch (N em n.n.N) incrementada.
-M OU --major-increment	Incrementa o número de versão principal (o N em N.n.n).
-m OU --minor-increment	Incrementa o número de versão secundária (o N em n.N.n).
-p OU --patch-increment	Incrementa o número de versão de patch (o N em n.n.N).
-i <i>INSTANCE_TYPE</i> OU --instance-type <i>INSTANCE_TYPE</i>	Use <i>INSTANCE_TYPE</i> como o tipo de instância, por exemplo, <b>t1.micro</b> .
-ip <i>INSTANCE_PROFILE</i> OU	Use <i>INSTANCE_PROFILE</i> como o perfil de instância ao criar AMIs para uma Plataforma personalizada.

Name (Nome)	Descrição
<code>--instance-profile</code> <i>INSTANCE_PROFILE</i>	Se a opção <code>-ip</code> não estiver especificada, criará o perfil de instância <code>aws-elasticbeanstalk-custom-platform-ec2-role</code> e o usará para a plataforma personalizada.
<code>--tags</code> <i>key1=value1[,key2=value2]</i>	<p>Marque sua versão da plataforma personalizada. As tags são especificadas como uma lista de <code>key=value</code> separados por vírgulas.</p> <p>Para obter mais detalhes, consulte <a href="#">Atribuir tags em versões de plataforma personalizada</a>.</p>
<code>--timeout</code> <i>minutos</i>	O número definido de minutos antes do limite de tempo do comando.
<code>--vpc.id</code> <i>VPC_ID</i>	O ID da VPC na qual o Packer compila.
<code>--vpc.subnets</code> <i>VPC_SUBNETS</i>	As sub-redes VPC nas quais o Packer compila.
<code>--vpc.publicip</code>	Associa IPs públicos a instâncias do EC2 iniciadas.

## Eb platform delete

Excluir uma versão da plataforma. A versão não será excluída se um ambiente a estiver usando.

### Opções

Name (Nome)	Descrição
<i>version</i>	A versão para excluir. Este valor é obrigatório.
<code>--cleanup</code>	Remover todas as versões de plataforma no estado Failed.
<code>--all-platforms</code>	Se <code>--cleanup</code> for especificado, remove todas as versões de plataforma no estado Failed para todas as plataformas.
<code>--force</code>	Não requer confirmação ao excluir uma versão.

## Eb platform events

Exibir os eventos de uma versão da plataforma. Se a *versão* for especificada, exibirá os eventos dessa versão; caso contrário, exibirá os eventos da versão atual.

### Opções

Name (Nome)	Descrição
<i>versão</i>	A versão para a qual os eventos são exibidos. Este valor é obrigatório.
-f OU --follow	Continuar a exibir os eventos à medida que ocorrem.

## Eb platform init

Inicializar um repositório de plataforma.

### Opções

Name (Nome)	Descrição
<i>platform</i>	O nome da plataforma para inicializar. Este valor é obrigatório, a menos que <i>-i</i> (modo interativo) esteja habilitado.
-i OU --interactive	Usar o modo interativo.
-k <i>KEYNAME</i> OU --keyname <i>KEYNAME</i>	O nome da chave EC2 padrão.

Você pode executar este comando em um diretório previamente inicializado, embora não possa alterar o tipo de espaço de trabalho se executado em um diretório previamente inicializado.

Para reinicializar com opções diferentes, use a opção `-i`.

## Eb platform list

Listar as versões da Plataforma associada a um espaço de trabalho (diretório) ou uma região.

O comando retorna resultados diferentes dependendo do tipo de espaço de trabalho no qual você o executa da seguinte forma:

- Em um workspace da Plataforma (um diretório inicializado por `eb platform init`), o comando retornará uma lista de todas as versões de Plataforma da Plataforma personalizada definida no workspace. Adicione a opção `--all-platforms` ou `--verbose` para obter uma lista de todas as versões de Plataforma de todas as Plataformas personalizadas que sua conta tiver na região associada ao workspace.
- Em um workspace da aplicação (um diretório inicializado por `eb init`), o comando retornará uma lista de todas as versões de plataforma, tanto para plataformas gerenciadas pelo Elastic Beanstalk quanto para as plataformas personalizadas da sua conta. A lista usa nomes curtos de versão da Plataforma, e algumas variáveis da versão da Plataforma podem ser combinadas. Adicione a opção `--verbose` para obter uma lista detalhada com nomes completos e todas as variáveis listadas separadamente.
- Em um diretório não inicializado, o comando só funciona com a opção `--region`. Ele retorna uma lista de todas as versões de plataforma gerenciada pelo Elastic Beanstalk compatíveis na região. A lista usa nomes curtos de versão da Plataforma, e algumas variáveis da versão da Plataforma podem ser combinadas. Adicione a opção `--verbose` para obter uma lista detalhada com nomes completos e todas as variáveis listadas separadamente.

## Opções

Name (Nome)	Descrição
<code>-a</code> OU <code>--all-platforms</code>	Válido somente em um workspace inicializado (um diretório inicializado por <code>eb platform init</code> ou <code>eb init</code> ). Lista as versões das Plataformas de todas as Plataformas associadas à sua conta.



Name (Nome)	Descrição
-s <i>STATUS</i> OU --status <i>STATUS</i>	Lista apenas as Plataformas que correspondem ao <i>STATUS</i> : <ul style="list-style-type: none"> <li>• Ready</li> <li>• Reprovada</li> <li>• Deleting (Excluindo)</li> <li>• Creating (Criando)</li> </ul>

## Eb platform logs

Exibir logs do ambiente do compilador para uma versão da plataforma.

### Opções

Name (Nome)	Descrição
<i>version</i>	A versão da plataforma cujos logs são exibidos. Se omitido, exibe logs da versão atual.
--stream	Fazer streaming de logs de implantação configurados com CloudWatch.

## Eb platform status

Exibir o status de uma versão da plataforma.

### Opções

Name (Nome)	Descrição
<i>version</i>	A versão da plataforma cujo status é recuperado. Se omitida, exibe o status da versão atual.

## Eb platform use

Selecionar uma plataforma diferente a partir da qual novas versões são criadas.

## Opções

Name (Nome)	Descrição
<i>platform</i>	Especifica <i>platform</i> como a versão ativa deste workspace. Este valor é obrigatório.

## Usar eb platform para ambientes

Lista as plataformas compatíveis e permite que você defina a plataforma padrão e a versão da plataforma a serem usadas ao iniciar um ambiente. Use `eb platform list` para ver uma lista de todas as plataformas compatíveis. Use `eb platform select` para alterar a plataforma de seu projeto. Use `eb platform show` para ver a plataforma selecionada de seu projeto.

### Sintaxe

```
eb platform list
```

```
eb platform select
```

```
eb platform show
```

### Opções

Name (Nome)	Descrição
<code>list</code>	Listar a versão da plataforma atual.
<code>select</code>	Selecionar a plataforma padrão.
<code>show</code>	Mostrar informações sobre a plataforma atual.

### Exemplo 1

O exemplo a seguir lista os nomes de todas as configurações para todas as plataformas compatíveis com o Elastic Beanstalk.

```
$ eb platform list
```

```
docker-1.5.0
glassfish-4.0-java-7-(preconfigured-docker)
glassfish-4.1-java-8-(preconfigured-docker)
go-1.3-(preconfigured-docker)
go-1.4-(preconfigured-docker)
iis-7.5
iis-8
iis-8.5
multi-container-docker-1.3.3-(generic)
node.js
php-5.3
php-5.4
php-5.5
python
python-2.7
python-3.4
python-3.4-(preconfigured-docker)
ruby-1.9.3
ruby-2.0-(passenger-standalone)
ruby-2.0-(puma)
ruby-2.1-(passenger-standalone)
ruby-2.1-(puma)
ruby-2.2-(passenger-standalone)
ruby-2.2-(puma)
tomcat-6
tomcat-7
tomcat-7-java-6
tomcat-7-java-7
tomcat-8-java-8
```

## Exemplo 2

O exemplo a seguir solicita que você escolha a plataforma em uma lista de plataformas e a versão que você deseja implantar para a plataforma especificada.

```
$ eb platform select
Select a platform.
1) PHP
2) Node.js
3) IIS
4) Tomcat
5) Python
6) Ruby
```

```

7) Docker
8) Multi-container Docker
9) GlassFish
10) Go
(default is 1): 5

```

Select a platform version.

```

1) Python 2.7
2) Python
3) Python 3.4 (Preconfigured - Docker)

```

### Exemplo 3

O exemplo a seguir mostra informações sobre a plataforma padrão atual.

```

$ eb platform show
Current default platform: Python 2.7
New environments will be running: 64bit Amazon Linux 2014.09 v1.2.0 running Python 2.7

Platform info for environment "tmp-dev":
Current: 64bit Amazon Linux 2014.09 v1.2.0 running Python
Latest: 64bit Amazon Linux 2014.09 v1.2.0 running Python

```

## eb printenv

### Descrição

Imprime todas as propriedades de ambiente na janela de comando.

### Sintaxe

```
eb printenv
```

```
eb printenv environment-name
```

### Opções

Name (Nome)	Descrição
<a href="#">Opções comuns</a>	

## Resultado

Se for bem-sucedido, o comando retornará o status da operação `printenv`.

## Exemplo

O exemplo a seguir imprime as propriedades de ambiente do ambiente especificado.

```
$ eb printenv
Environment Variables:
 PARAM1 = Value1
```

## eb restore

### Descrição

Reconstrói um ambiente encerrado, criando um novo ambiente com os mesmos nome, ID e configuração. O nome do ambiente, o nome de domínio e a versão do aplicativo devem estar disponíveis para uso para que a reconstrução seja bem-sucedida.

### Sintaxe

```
eb restore
```

```
eb restore environment_id
```

### Opções

Name (Nome)	Descrição
<a href="#">Opções comuns</a>	

## Resultado

A EB CLI exibe uma lista de ambientes encerrados que estão disponíveis para restauração.

## Exemplo

```
$ eb restore
```

```
Select a terminated environment to restore
```

#	Name	ID	Application Version	Date Terminated	Ago
3 mins	gamma	e-s7mimej8e9	app-77e3-161213_211138	2016/12/14 20:32 PST	13
2 mins	beta	e-sj28uu2wia	app-77e3-161213_211125	2016/12/14 20:32 PST	13
1 hours	alpha	e-gia8mpfu6q	app-77e3-161213_211109	2016/12/14 16:21 PST	4

```
(Commands: Quit, Restore, # #)
```

```
Selected environment alpha
```

```
Application: scorekeep
Description: Environment created from the EB CLI using "eb create"
CNAME: alpha.h23tbtbm92.us-east-2.elasticbeanstalk.com
Version: app-77e3-161213_211109
Platform: 64bit Amazon Linux 2016.03 v2.1.6 running Java 8
Terminated: 2016/12/14 16:21 PST
Restore this environment? [y/n]: y
```

```
2018-07-11 21:04:20 INFO: restoreEnvironment is starting.
2018-07-11 21:04:39 INFO: Created security group named: sg-e2443f72
...
```

## eb scale

### Descrição

Escala o ambiente para sempre executar um número específico de instâncias, definindo o número mínimo e máximo de instâncias como o número especificado.

### Sintaxe

```
eb scale number-of-instances
```

```
eb scale number-of-instances environment-name
```

## Opções

Name (Nome)	Descrição
--timeout	O número de minutos antes de esgotar o tempo limite do comando.
<a href="#">Opções comuns</a>	

## Resultado

Se for bem-sucedido, o comando atualizará o número mínimo e máximo de instâncias a executar para o número especificado.

## Exemplo

O exemplo a seguir define o número de instâncias como 2.

```
$ eb scale 2
2018-07-11 21:05:22 INFO: Environment update is starting.
2018-07-11 21:05:27 INFO: Updating environment tmp-dev's configuration settings.
2018-07-11 21:08:53 INFO: Added EC2 instance 'i-5fce3d53' to Auto Scaling Group
'awseb-e-2cpfjbra9a-stack-AWSEBAutoScalingGroup-7AXY7U13ZQ6E'.
2018-07-11 21:08:58 INFO: Successfully deployed new configuration to environment.
2018-07-11 21:08:59 INFO: Environment update completed successfully.
```

## eb setenv

### Descrição

Define as [propriedades do ambiente](#) para o ambiente padrão.

### Sintaxe

```
eb setenv key=value
```

Você pode incluir quantas propriedades quiser, mas o tamanho total de todas as propriedades não pode exceder 4096 bytes. Você pode excluir uma variável deixando o valor em branco. Para obter os limites, consulte [Configurar propriedades do ambiente \(variáveis de ambiente\)](#).

**Note**

Se o `value` contiver um [caractere especial](#), você deverá escapar esse caractere colocando o caractere `\` antes dele.

## Opções

Name (Nome)	Descrição
<code>--timeout</code>	O número de minutos antes de esgotar o tempo limite do comando.
<a href="#">Opções comuns</a>	

## Resultado

Se for bem-sucedido, o comando exibe que a atualização do ambiente foi bem-sucedida.

## Exemplo

O exemplo a seguir define a variável de ambiente `ExampleVar`.

```
$ eb setenv ExampleVar=ExampleValue
2018-07-11 21:05:25 INFO: Environment update is starting.
2018-07-11 21:05:29 INFO: Updating environment tmp-dev's configuration settings.
2018-07-11 21:06:50 INFO: Successfully deployed new configuration to environment.
2018-07-11 21:06:51 INFO: Environment update completed successfully.
```

O comando a seguir define várias propriedades de ambiente. Ele adiciona a propriedade de ambiente chamada `foo` e define seu valor para `bar`, altera o valor da propriedade `JDBC_CONNECTION_STRING` e exclui as propriedades `PARAM4` e `PARAM5`.

```
$ eb setenv foo=bar JDBC_CONNECTION_STRING=hello PARAM4= PARAM5=
```



## eb ssh

### Descrição

#### Note

Esse comando não funciona com ambientes que executam instâncias do Windows Server.

Conecte-se a uma instância Linux do Amazon EC2 do ambiente usando o Secure Shell (SSH). Se um ambiente tiver várias instâncias em execução, a EB CLI solicitará que você especifique uma delas para se conectar. Para usar esse comando, o SSH deve ser instalado na máquina local e estar disponível na linha de comando. Os arquivos de chave privada devem estar localizados em uma pasta chamada `.ssh` no diretório do usuário, e as instâncias do EC2 do ambiente devem ter endereços IP públicos.

Se o diretório raiz contiver um arquivo `platform.yaml` especificando uma plataforma personalizada, esse comando também se conectará a instâncias do ambiente personalizado.

#### Chaves SSH

Se você ainda não configurou o SSH, pode usar a EB CLI para criar uma chave ao executar `eb init`. Se você já executou o `eb init`, execute-o novamente com a opção `--interactive` e selecione Yes (Sim) e Create New Keypair (Criar novo par de chaves) quando for solicitada a configuração do SSH. As chaves criadas durante esse processo serão armazenadas na pasta correta pela EB CLI.

Esse comando abre temporariamente a porta 22 no security group do seu ambiente para o tráfego de entrada de 0.0.0.0/0 (todos os endereços IP), caso não haja regras em vigor para a porta 22. Se você tiver configurado o security group do ambiente para abrir a porta 22 para um intervalo restrito de CIDR, objetivando maior segurança, a EB CLI respeitará essa configuração, ignorando quaisquer alterações feitas no security group. Para substituir esse comportamento e forçar a EB CLI a abrir a porta 22 para todo o tráfego de entrada, use a opção `--force`.

Consulte [Grupos de segurança](#) para informações sobre como configurar o security group do ambiente.

## Sintaxe

```
eb ssh
```

```
eb ssh environment-name
```

## Opções

Name (Nome)	Descrição
<code>-i</code> ou <code>--instance</code>	Especifica o ID da instância à qual você se conecta. Recomendamos o uso dessa opção.
<code>-n</code> ou <code>--number</code>	Especifique a instância para conexão por número.
<code>-o</code> ou <code>--keep_open</code>	Deixe a porta 22 aberta no security group depois que a sessão SSH for encerrada.
<code>--command</code>	Execute um comando shell na instância especificada, em vez de iniciar uma sessão SSH.
<code>--custom</code>	Especifique um comando SSH para usar, em vez de 'ssh -i keyfile'. Não inclua o usuário remoto e o nome do host.
<code>--setup</code>	Altere o par de chaves atribuído às instâncias do ambiente (requer a substituição das instâncias).
<code>--force</code>	Abra a porta 22 para o tráfego de entrada de 0.0.0.0/0 no security group do ambiente, mesmo que o security group já esteja configurado para SSH.

Name (Nome)	Descrição
	Use essa opção se o security group do ambiente estiver configurado para abrir a porta 22 para um intervalo restrito de CIDR que não inclui o endereço IP do qual você está tentando se conectar.
<code>--timeout <i>minutos</i></code>	O número definido de minutos antes do limite de tempo do comando.  Pode ser usado com o argumento <code>--setup</code> .
<a href="#">Opções comuns</a>	

## Resultado

Se for bem-sucedido, o comando abrirá uma conexão SSH com a instância.

## Exemplo

O exemplo a seguir conecta você ao ambiente especificado.

```
$ eb ssh
Select an instance to ssh into
1) i-96133799
2) i-5931e053
(default is 1): 1
INFO: Attempting to open port 22.
INFO: SSH port 22 open.
The authenticity of host '54.191.45.125 (54.191.45.125)' can't be established.
RSA key fingerprint is ee:69:62:df:90:f7:63:af:52:7c:80:60:1b:3b:51:a9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.191.45.125' (RSA) to the list of known hosts.

 _| _|_)
 _| (/ Amazon Linux AMI
__|_|_|_|

https://aws.amazon.com/amazon-linux-ami/2014.09-release-notes/
No packages needed for security; 1 packages available
Run "sudo yum update" to apply all updates.
```

```
[ec2-user@ip-172-31-8-185 ~]$ ls
[ec2-user@ip-172-31-8-185 ~]$ exit
logout
Connection to 54.191.45.125 closed.
INFO: Closed port 22 on ec2 instance security group
```

## eb status

### Descrição

Fornecer informações sobre o status do ambiente.

Se o diretório raiz contiver um arquivo `platform.yaml` especificando uma plataforma personalizada, esse comando também fornecerá informações sobre o ambiente do compilador.

### Sintaxe

`eb status`

`eb status environment-name`

### Opções

Name (Nome)	Descrição
<code>-v</code>	Fornecer mais informações sobre instâncias individuais, como o status delas com o load balancer do Elastic Load Balancing.
ou	
<code>--verbose</code>	
<a href="#">Opções comuns</a>	

### Resultado

Se tiver êxito, o comando retornará as informações a seguir sobre o ambiente:

- Nome do ambiente
- Application name

- Versão do aplicativo implantada
- ID do ambiente
- Plataforma
- Nível do ambiente
- CNAME
- Hora em que o ambiente foi atualizado pela última vez
- Status
- Integridade

Se você usar o modo detalhado, a CLI do EB também fornecerá o número de instâncias do Amazon EC2 em execução.

## Exemplo

O exemplo a seguir mostra o status do ambiente tmp-dev.

```
$ eb status
Environment details for: tmp-dev
 Application name: tmp
 Region: us-west-2
 Deployed Version: None
 Environment ID: e-2cpfjbra9a
 Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
 Tier: WebServer-Standard-1.0
 CNAME: tmp-dev.elasticbeanstalk.com
 Updated: 2014-10-29 21:37:19.050000+00:00
 Status: Launching
 Health: Grey
```

## eb swap

### Descrição

Troca o CNAME do ambiente pelo CNAME de outro ambiente (por exemplo, para evitar tempo de inatividade quando você atualizar a versão do aplicativo).

**Note**

Se você tiver mais de dois ambientes, será solicitado a selecionar o nome do ambiente que está atualmente usando o CNAME desejado em uma lista de ambientes. Para suprimir isso, especifique o nome do ambiente a usar incluindo a opção `-n` quando executar o comando.

## Sintaxe

```
eb swap
```

```
eb swap environment-name
```

**Note**

O *environment\_name* é o ambiente para o qual é recomendado ter um CNAME diferente. Se você não especificar o *environment\_name* como um parâmetro da linha de comando quando executar `eb swap`, a EB CLI atualizará o CNAME do ambiente padrão.

## Opções

Name (Nome)	Descrição
<code>-n</code> ou <code>--destination_name</code>	Especifica o nome do ambiente com o qual você deseja trocar CNAMEs. Se você executar <code>eb swap</code> sem essa opção, a EB CLI solicitará que seja feita a seleção em uma lista de ambientes.
<a href="#">Opções comuns</a>	

## Resultado

Se for bem-sucedido, o comando retornará o status da operação swap.

## Exemplos

O exemplo a seguir troca o ambiente `tmp-dev` pelo `live-env`.

```
$ eb swap
Select an environment to swap with.
1) staging-dev
2) live-env
(default is 1): 2
2018-07-11 21:05:25 INFO: swapEnvironmentCNAMEs is starting.
2018-07-11 21:05:26 INFO: Swapping CNAMEs for environments 'tmp-dev' and 'live-env'.
2018-07-11 21:05:30 INFO: 'tmp-dev.elasticbeanstalk.com' now points to 'awseb-e-j-
AWSEBLoa-M7U21VXNLWHN-487871449.us-west-2.elb.amazonaws.com'.
2018-07-11 21:05:30 INFO: Completed swapping CNAMEs for environments 'tmp-dev' and
'live-env'.
```

O exemplo a seguir troca o ambiente tmp-dev pelo ambiente live-env, mas não solicita que você informe ou selecione um valor para as configurações.

```
$ eb swap tmp-dev --destination_name live-env
2018-07-11 21:18:12 INFO: swapEnvironmentCNAMEs is starting.
2018-07-11 21:18:13 INFO: Swapping CNAMEs for environments 'tmp-dev' and 'live-env'.
2018-07-11 21:18:17 INFO: 'tmp-dev.elasticbeanstalk.com' now points to 'awseb-e-j-
AWSEBLoa-M7U21VXNLWHN-487871449.us-west-2.elb.amazonaws.com'.
2018-07-11 21:18:17 INFO: Completed swapping CNAMEs for environments 'tmp-dev' and
'live-env'.
```

## eb tags

### Descrição

Adicione, exclua, atualize e liste tags de um recurso do Elastic Beanstalk.

Para obter detalhes sobre a marcação de recursos no Elastic Beanstalk, consulte [Marcar recursos da aplicação do Elastic Beanstalk](#).

### Sintaxe

```
eb tags [environment-name] [--resource ARN] -l | --list
```

```
eb tags [environment-name] [--resource ARN] -a | --add key1=value1[,key2=value2 ...]
```

```
eb tags [environment-name] [--resource ARN] -u | --update key1=value1[,key2=value2 ...]
```

```
eb tags [environment-name] [--resource ARN] -d | --delete key1[,key2 ...]
```

Você pode combinar as opções de subcomando `--add`, `--update` e `--delete` em um único comando. Pelo menos uma delas é necessária. Você não pode combinar nenhuma dessas três opções de subcomando com `--list`.

Sem nenhum argumento adicional, todos esses comandos listam ou modificam tags do ambiente padrão no aplicativo do diretório atual. Com um argumento *environment-name*, os comandos listam ou modificam as tags desse ambiente. Com a opção `--resource`, os comandos listam ou modificam as tags de qualquer recurso do Elastic Beanstalk: uma aplicação, um ambiente, uma versão da aplicação, uma configuração salva ou uma versão da plataforma personalizada. Especifique o recurso pelo seu nome de recurso da Amazon (ARN).

## Opções

Nenhuma dessas opções é obrigatória. Se você executar `eb create` sem opções, será solicitado a inserir ou selecionar um valor para cada configuração.

Nome	Descrição
<code>-l</code> ou <code>--list</code>	Listar todas as tags que estão sendo aplicadas no recurso atualmente.
<code>-a key1=value1[,key2=value2]</code> ou <code>--add key1=value1[,key2=va</code>	Aplique novas tags no recurso. Especifique as tags como uma lista de <code>key=value</code> separados por vírgulas. Não é possível especificar chaves de tags existentes. Valores válidos. Veja <a href="#">Marcar recursos</a> .
<code>-u key1=value1[,key2=value2]</code> ou <code>--updat</code> e <code>key1=value1[,key2=value2]</code>	Atualize os valores de tags de recursos existentes. Especifique as tags como uma lista de <code>key=value</code> separados por vírgulas. Você precisa especificar chaves de tags existentes. Valores válidos. Veja <a href="#">Marcar recursos</a> .
<code>-d key1[,key2 ...]</code> ou	Exclua as tags de recursos existentes. Especifique as tags como uma lista de chaves separados por vírgulas. Você precisa especificar chaves de tags existentes.



Nome	Descrição
<code>--delete</code> <i>key1[,key2 ...]</i>	Valores válidos. Veja <a href="#">Marcar recursos</a> .
<code>-r</code> <i>region</i>	A Região da AWS no qual seu recurso existe.
ou	Padronizado: a região padrão configurada.
<code>--region</code> <i>region</i>	Para obter a lista de valores que você pode especificar para essa opção, consulte <a href="#">Endpoints e cotas do AWS Elastic Beanstalk</a> no Referência geral da AWS.
<code>--resource</code> <i>ARN</i>	<p>O ARN do recurso para o qual o comando modifica ou lista tags. Se não for especificado, o comando refere-se ao ambiente (padrão ou especificado) no aplicativo do diretório atual.</p> <p>Valores válidos: consulte um dos subtópicos de <a href="#">Marcar recursos</a> específico do recurso de interesse. Esses tópicos mostram como o ARN do recurso é construído e explicam como obter uma lista dos ARNs desse recurso que existem para o aplicativo ou a conta.</p>

## Resultado

A opção de subcomando de `--list` exibe uma lista das tags de recursos. A saída mostra as tags que o Elastic Beanstalk aplica por padrão e suas tags personalizadas.

```
$ eb tags --list
Showing tags for environment 'MyApp-env':

Key Value
Name MyApp-env
elasticbeanstalk:environment-id e-63cmxwjaut
elasticbeanstalk:environment-name MyApp-env
mytag tagvalue
tag2 2nd value
```

As opções de subcomando `--add`, `--update` e `--delete`, quando bem-sucedidas, não têm nenhuma saída. Você pode adicionar a opção `--verbose` para ver saída detalhada da atividade do comando.

```
$ eb tags --verbose --update "mytag=tag value"
```

```
Updated Tags:
```

Key	Value
mytag	tag value

## Exemplos

O comando a seguir adiciona uma tag com êxito com a chave `tag1` e o valor `value1` ao ambiente padrão do aplicativo e, ao mesmo tempo, exclui a tag `tag2`.

```
$ eb tags --add tag1=value1 --delete tag2
```

O comando a seguir adiciona uma tag com êxito a uma configuração salva em um aplicativo.

```
$ eb tags --add tag1=value1 \
 --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-
id:configurationtemplate/my-app/my-template"
```

O comando a seguir falha porque ele tenta atualizar uma tag não existente.

```
$ eb tags --update tag3=newval
ERROR: Tags with the following keys can't be updated because they don't exist:

tag3
```

O comando a seguir falha porque ele tenta atualizar e excluir a mesma chave.

```
$ eb tags --update mytag=newval --delete mytag
ERROR: A tag with the key 'mytag' is specified for both '--delete' and '--update'. Each
tag can be either deleted or updated in a single operation.
```

## eb terminate

### Descrição

Termine o ambiente em execução para não ser cobrado por recursos da AWS não usados.

O uso da opção `--all` exclui o aplicativo para o qual o diretório atual foi inicializado usando [eb init](#). O comando termina todos os ambientes na aplicação. Termina também todos as [versões da aplicação](#) e as [configurações salvas](#) da aplicação e depois exclui a aplicação.

Se o diretório raiz contém um arquivo `platform.yaml` que especifica uma plataforma personalizada, esse comando encerra o ambiente personalizado em execução.

#### Note

Você sempre pode iniciar um novo ambiente usando a mesma versão mais tarde.

Se você tiver dados de um ambiente que deseja manter, defina a política de exclusão de banco de dados como `Retain` antes de encerrar o ambiente. Isso mantém o banco de dados operacional fora do Elastic Beanstalk. Depois disso, qualquer ambiente do Elastic Beanstalk deve se conectar a ele como um banco de dados externo. Se você quiser fazer backup dos dados sem manter o banco de dados operacional, defina a política de exclusão para fazer um snapshot do banco de dados antes de encerrar o ambiente. Para obter mais informações, consulte [Ciclo de vida do banco de dados](#) no capítulo Configuração de ambientes deste guia.

#### Important

Se você encerrar um ambiente, também deverá excluir quaisquer mapeamentos CNAME criados, pois outros clientes poderão reutilizar um nome de host disponível. Certifique-se de excluir os registros DNS que apontam para seu ambiente encerrado para evitar uma entrada de DNS pendente. Uma entrada de DNS pendente pode expor o tráfego da Internet destinado ao seu domínio a vulnerabilidades de segurança. Também pode apresentar outros riscos.

Para obter mais informações, consulte [Proteção contra um registro de delegação pendente no Route 53](#) no Guia do desenvolvedor do Amazon Route 53. Você também pode aprender mais sobre entradas de DNS pendentes em [Proteções de domínio aprimoradas para solicitações do Amazon CloudFront](#) no blog de segurança da AWS.

## Sintaxe

`eb terminate`

`eb terminate environment-name`

## Opções

Name (Nome)	Descrição
<code>--all</code>	Encerra todos os ambientes no aplicativo, bem como as <a href="#">versões</a> e as <a href="#">configurações salvas</a> do aplicativo, e em seguida exclui o aplicativo.
<code>--force</code>	Termina o ambiente sem pedir confirmação.
<code>--ignore-links</code>	Termina o ambiente, mesmo se houver ambientes dependentes vinculados a ele. Consulte <a href="#">Ambientes do Compose</a> .
<code>--timeout</code>	O número de minutos antes de esgotar o tempo limite do comando.

## Resultado

Se for bem-sucedido, o comando retornará o status da operação `terminate`.

## Exemplo

A solicitação de exemplo a seguir encerra o ambiente `tmp-dev`.

```
$ eb terminate
```

```
The environment "tmp-dev" and all associated instances will be terminated.
```

```
To confirm, type the environment name: tmp-dev
```

```
2018-07-11 21:05:25 INFO: terminateEnvironment is starting.
```

```
2018-07-11 21:05:40 INFO: Deleted CloudWatch alarm named: awseb-e-2cpfjbra9a-stack-AWSEBCloudwatchAlarmHigh-16V08Y0F2KQ7U
```

```
2018-07-11 21:05:41 INFO: Deleted CloudWatch alarm named: awseb-e-2cpfjbra9a-stack-AWSEBCloudwatchAlarmLow-6ZAWH9F20P7C
```

```
2018-07-11 21:06:42 INFO: Deleted Auto Scaling group policy named:
arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:5d7d3e6b-
```

```

d59b-47c5-b102-3e11fe3047be:autoScalingGroupName/awseb-e-2cpfjbra9a-stack-
AWSEBAutoScalingGroup-7AXY7U13ZQ6E:policyName/awseb-e-2cpfjbra9a-stack-AWSEBAutoSca
lingScaleUpPolicy-1876U27JEC34J
2018-07-11 21:06:43 INFO: Deleted Auto Scaling group policy named:
 arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:29c6e7c7-7ac8-46fc-91f5-
cfabb65b985b:autoScalingGroupName/awseb-e-2cpfjbra9a-stack-
AWSEBAutoScalingGroup-7AXY7U13ZQ6E:policyName/awseb-e-2cpfjbra9a-stack-AWSEBAutoSca
lingScaleDownPolicy-SL4LH0DM0MU
2018-07-11 21:06:48 INFO: Waiting for EC2 instances to terminate. This may take a
 few minutes.
2018-07-11 21:08:55 INFO: Deleted Auto Scaling group named: awseb-e-2cpfjbra9a-
stack-AWSEBAutoScalingGroup-7AXY7U13ZQ6E
2018-07-11 21:09:10 INFO: Deleted security group named: awseb-e-2cpfjbra9a-stack-
AWSEBSecurityGroup-XT4YYGFL7I99
2018-07-11 21:09:40 INFO: Deleted load balancer named: awseb-e-2-AWSEBLoa-
AK6RRYFQVV3S
2018-07-11 21:09:42 INFO: Deleting SNS topic for environment tmp-dev.
2018-07-11 21:09:52 INFO: terminateEnvironment completed successfully.

```

## eb upgrade

### Descrição

Atualiza a plataforma de seu ambiente para a versão mais recente da plataforma na qual ele está sendo executado no momento.

Se o diretório raiz contém um arquivo `platform.yaml` especificando uma plataforma personalizada, este comando atualiza o ambiente para a versão mais recente da plataforma personalizada em que está em execução no momento.

### Sintaxe

```
eb upgrade
```

```
eb upgrade environment-name
```

### Opções

Name (Nome)	Descrição
<code>--force</code>	Atualiza sem a necessidade de que você confirme o nome do ambiente antes de iniciar o processo de atualização.

Name (Nome)	Descrição
<code>--noroll</code>	Atualiza todas as instâncias sem usar as atualizações contínuas para manter algumas instâncias em serviço durante a atualização.
<a href="#">Opções comuns</a>	

## Resultado

O comando mostra uma visão geral da alteração e solicita que você confirme a atualização digitando o nome do ambiente. Em caso de êxito, seu ambiente é atualizado e iniciado com a versão mais recente da plataforma.

## Exemplo

O exemplo a seguir atualiza a versão da plataforma atual do ambiente especificado para a versão mais recente disponível da plataforma.

```
$ eb upgrade
Current platform: 64bit Amazon Linux 2014.09 v1.0.9 running Python 2.7
Latest platform: 64bit Amazon Linux 2014.09 v1.2.0 running Python 2.7

WARNING: This operation replaces your instances with minimal or zero downtime. You may
cancel the upgrade after it has started by typing "eb abort".
You can also change your platform version by typing "eb clone" and then "eb swap".

To continue, type the environment name:
```

## eb use

### Descrição

Define o ambiente especificado como o padrão.

Ao usar o Git, `eb use` define o ambiente padrão para a ramificação. Execute esse comando uma vez em cada ramificação que você deseje implantar no Elastic Beanstalk.

## Sintaxe

eb use ***environment-name***

## Opções

Name (Nome)	Descrição
<code>--source codecommit/ <i>repository-name/branch-name</i></code>	Repositório do CodeCommit e ramificação. Consulte <a href="#">Usar o EB CLI com o AWS CodeCommit</a> .
<code>-r <i>region</i></code> <code>--region <i>region</i></code>	Altere a região na qual você cria ambientes.
<a href="#">Opções comuns</a>	

## Opções comuns

Você pode usar as seguintes opções com todos os comandos da EB CLI.

Nome	Descrição
<code>--debug</code>	Imprimir informações para depuração.
<code>-h, --help</code>	Mostrar a mensagem de ajuda.  Tipo: sequência  Padrão: nenhum
<code>--no-verify-ssl</code>	Ignorar verificação de certificado SSL. Use essa opção se você tiver problemas ao utilizar a CLI com um proxy.
<code>--profile</code>	Usar um perfil específico de seu arquivo de credenciais da AWS.
<code>--quiet</code>	Suprimir todas as saídas do comando.

Nome	Descrição
<code>--region</code>	Usar a região especificada.
<code>-v, --verbose</code>	Exibir informações detalhadas.

## EB CLI 2.6 (desativada)

Essa versão da CLI do EB e a documentação foram substituídas pela versão 3 (nesta seção, a CLI do EB 3 representa a versão 3 e posteriores da CLI do EB). Para obter informações sobre a nova versão, consulte [Usar a Interface de Linhas de Comando do Elastic Beanstalk \(CLI do EB\)](#).

Você deve migrar para a versão mais recente da EB CLI 3. Ela pode gerenciar ambientes que você iniciou usando a EB CLI 2.6 ou versões anteriores da EB CLI.

## Diferenças da versão 3 da EB CLI

O EB é uma ferramenta de Interface de Linhas de Comando (CLI) do Elastic Beanstalk que pode ser usada para implantar aplicações de forma rápida e com mais facilidade. A versão mais recente do EB foi introduzida pelo Elastic Beanstalk na CLI 3 do EB. A EB CLI recupera automaticamente as configurações de um ambiente criado usando o EB se o ambiente estiver em execução. Observe que a EB CLI 3 não armazena opções de configuração localmente, como nas versões anteriores.

A EB CLI introduz os comandos `eb create`, `eb deploy`, `eb open`, `eb console`, `eb scale`, `eb setenv`, `eb config`, `eb terminate`, `eb clone`, `eb list`, `eb use`, `eb printenv` e `eb ssh`. Na EB CLI 3.1 ou posterior, você também pode usar o comando `eb swap`. Somente na EB CLI 3.2 você pode usar os comandos `eb abort`, `eb platform` e `eb upgrade`. Além dessas novidades, os comandos da EB CLI 3 diferem dos da EB CLI 2.6 em vários casos:

- `eb init`: use `eb init` para criar um diretório `.elasticbeanstalk` em um diretório de projeto existente e criar uma nova aplicação do Elastic Beanstalk para o projeto. Ao contrário das versões anteriores, a EB CLI 3 e as versões posteriores não solicitam que você crie um ambiente.
- `eb start`: a CLI 3 do EB não inclui o comando `eb start`. Use `eb create` para criar um ambiente.
- `eb stop`: a CLI 3 do EB não inclui o comando `eb stop`. Use `eb terminate` para encerrar completamente um ambiente e fazer a limpeza.
- `eb push` e `git aws.push`: a CLI 3 do EB não inclui os comandos `eb push` e `git aws.push`. Use `eb deploy` para atualizar o seu código de aplicativo.



- `eb update`: a CLI 3 do EB não inclui o comando `eb update`. Use `eb config` para atualizar um ambiente.
- `eb branch`: a CLI 3 do EB não inclui o comando `eb branch`.

Para obter mais informações sobre o uso de comandos da EB CLI 3 para criar e gerenciar um aplicativo, consulte [Referência de comandos da EB CLI](#). Para obter uma explicação detalhada sobre como implantar um aplicativo de exemplo usando a EB CLI 3, consulte [Gerenciar ambientes do Elastic Beanstalk com a CLI do EB](#).

## Migrar para o CLI 3 do EB e o CodeCommit

O Elastic Beanstalk não apenas retirou a CLI 2.6 do EB, mas também removeu algumas funcionalidades da versão 2.6. A alteração mais significativa da 2.6 é que a EB CLI não oferece mais suporte nativo a atualizações de código incrementais (`eb push`, `git aws .push`) nem ramificações (`eb branch`). Esta seção descreve como migrar da CLI 2.6 do EB para a versão mais recente da CLI do EB e usar o CodeCommit como o repositório de código.

Se você ainda não tiver feito isso, crie um repositório de código no CodeCommit como descrito em [Migrar para o CodeCommit](#).

Assim que você tiver [instalado](#) e [configurado](#) a CLI do EB, terá duas oportunidades para associar sua aplicação ao seu repositório do CodeCommit, incluindo uma ramificação específica.

- Ao executar `eb init`, como no exemplo a seguir, em que *myRepo* é o nome do seu repositório do CodeCommit e *myBranch* é a ramificação no CodeCommit.

```
eb init --source codecommit/myRepo/myBranch
```

- Ao executar `eb deploy`, como no exemplo a seguir, em que *myRepo* é o nome do seu repositório do CodeCommit e *myBranch* é a ramificação no CodeCommit.

```
eb deploy --source codecommit/myRepo/myBranch
```

Para obter mais informações, incluindo como implantar atualizações de código incrementais a um ambiente do Elastic Beanstalk sem precisar fazer novamente upload de todo o projeto, consulte [Usar o EB CLI com o AWS CodeCommit](#).

## Interface de Linhas de Comando da API do Elastic Beanstalk (retirada)

Essa ferramenta, a Elastic Beanstalk API Command Line Interface (API CLI), foi substituída pela AWS CLI, que fornece comandos equivalentes à API para todos os produtos da AWS. Consulte o Guia do Usuário do AWS Command Line Interface para começar a usar o AWS CLI. Teste também a [CLI do EB](#) para uma experiência de linha de comando simplificada, mais alta.

### Converter scripts da CLI da API do Elastic Beanstalk

Converta seus scripts antigos da CLI da API do EB para usar a AWS CLI ou o Tools for Windows PowerShell para obter acesso às APIs mais recentes do Elastic Beanstalk. A tabela a seguir lista os comandos da CLI baseados em API do Elastic Beanstalk e seus comandos equivalentes na AWS CLI e no Tools for Windows PowerShell.

CLI da API do Elastic Beanstalk	AWS CLI	AWS Tools for Windows PowerShell
elastic-beanstalk-check-dns-availability	<a href="#">check-dns-availability</a>	Get-EBDNSAvailability
elastic-beanstalk-create-application	<a href="#">create-application</a>	New-EBApplication
elastic-beanstalk-create-application-version	<a href="#">create-application-version</a>	New-EBApplicationVersion
elastic-beanstalk-create-configuration-template	<a href="#">create-configuration-template</a>	New-EBConfigurationTemplate

CLI da API do Elastic Beanstalk	AWS CLI	AWS Tools for Windows PowerShell
<code>elastic-beanstalk-create-environment</code>	<a href="#"><u>create-environment</u></a>	<code>New-EBEnvironment</code>
<code>elastic-beanstalk-create-storage-location</code>	<a href="#"><u>create-storage-location</u></a>	<code>New-EBStorageLocation</code>
<code>elastic-beanstalk-delete-application</code>	<a href="#"><u>delete-application</u></a>	<code>Remove-EBApplication</code>
<code>elastic-beanstalk-delete-application-version</code>	<a href="#"><u>delete-application-version</u></a>	<code>Remove-EBApplicationVersion</code>
<code>elastic-beanstalk-delete-configuration-template</code>	<a href="#"><u>delete-configuration-template</u></a>	<code>Remove-EBConfigurationTemplate</code>
<code>elastic-beanstalk-delete-environment-configuration</code>	<a href="#"><u>delete-environment-configuration</u></a>	<code>Remove-EBEnvironmentConfiguration</code>
<code>elastic-beanstalk-describe-application-versions</code>	<a href="#"><u>describe-application-versions</u></a>	<code>Get-EBApplicationVersion</code>
<code>elastic-beanstalk-describe-applications</code>	<a href="#"><u>describe-applications</u></a>	<code>Get-EBApplication</code>

CLI da API do Elastic Beanstalk	AWS CLI	AWS Tools for Windows PowerShell
<code>elastic-beanstalk-describe-configuration-options</code>	<a href="#"><u>describe-configuration-options</u></a>	<code>Get-EBConfigurationOption</code>
<code>elastic-beanstalk-describe-configuration-settings</code>	<a href="#"><u>describe-configuration-settings</u></a>	<code>Get-EBConfigurationSetting</code>
<code>elastic-beanstalk-describe-environment-resources</code>	<a href="#"><u>describe-environment-resources</u></a>	<code>Get-EBEnvironmentResource</code>
<code>elastic-beanstalk-describe-environments</code>	<a href="#"><u>describe-environments</u></a>	<code>Get-EBEnvironment</code>
<code>elastic-beanstalk-describe-events</code>	<a href="#"><u>describe-events</u></a>	<code>Get-EBEvent</code>
<code>elastic-beanstalk-list-available-solution-stacks</code>	<a href="#"><u>list-available-solution-stacks</u></a>	<code>Get-EBAvailableSolutionStack</code>
<code>elastic-beanstalk-rebuild-environment</code>	<a href="#"><u>rebuild-environment</u></a>	<code>Start-EBEnvironmentRebuild</code>
<code>elastic-beanstalk-request-environment-info</code>	<a href="#"><u>request-environment-info</u></a>	<code>Request-EBEnvironmentInfo</code>

CLI da API do Elastic Beanstalk	AWS CLI	AWS Tools for Windows PowerShell
<code>elastic-beanstalk-restart-app-server</code>	<a href="#"><u>restart-app-server</u></a>	<code>Restart-EBAppServer</code>
<code>elastic-beanstalk-retrieve-environment-info</code>	<a href="#"><u>retrieve-environment-info</u></a>	<code>Get-EBEnvironmentInfo</code>
<code>elastic-beanstalk-swap-environment-cnames</code>	<a href="#"><u>swap-environment-cnames</u></a>	<code>Set-EBEnvironmentCNAME</code>
<code>elastic-beanstalk-terminate-environment</code>	<a href="#"><u>terminate-environment</u></a>	<code>Stop-EBEnvironment</code>
<code>elastic-beanstalk-update-application</code>	<a href="#"><u>update-application</u></a>	<code>Update-EBApplication</code>
<code>elastic-beanstalk-update-application-version</code>	<a href="#"><u>update-application-version</u></a>	<code>Update-EBApplicationVersion</code>
<code>elastic-beanstalk-update-configuration-template</code>	<a href="#"><u>update-configuration-template</u></a>	<code>Update-EBConfigurationTemplate</code>
<code>elastic-beanstalk-update-environment</code>	<a href="#"><u>update-environment</u></a>	<code>Update-EBEnvironment</code>

CLI da API do Elastic Beanstalk	AWS CLI	AWS Tools for Windows PowerShell
<code>elastic-beanstalk-validate-configuration-settings</code>	<a href="#"><u>validate-configuration-settings</u></a>	<code>Test-EBConfigurationSetting</code>

# Segurança do AWS Elastic Beanstalk

A segurança da nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você se contará com um datacenter e uma arquitetura de rede criados para atender aos requisitos das organizações com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem.

Segurança da nuvem - A AWS é responsável pela proteção da infraestrutura que executa todos os serviços oferecidos na Nuvem AWS e por fornecer serviços que você pode usar com segurança. Nossa responsabilidade de segurança é a maior prioridade na AWS, e a eficácia da nossa segurança é regularmente testada e verificada por auditores terceirizados como parte dos [Programas de Compatibilidade da AWS](#). Consulte os [Produtos da AWS no escopo dos programas de garantia da AWS](#) para obter informações relacionadas ao Elastic Beanstalk.

Segurança na nuvem: sua responsabilidade é determinada pelo produto da AWS que você está usando e por outros fatores, incluindo a confidencialidade dos dados, os requisitos da organização e as leis e regulamentos aplicáveis. Esta documentação se destina a ajudar você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Elastic Beanstalk.

Use os tópicos de segurança a seguir para saber mais sobre as tarefas de segurança pelas quais o Elastic Beanstalk é responsável e as configurações de segurança que você deve considerar ao usar o Elastic Beanstalk para atender aos seus objetivos de segurança e conformidade.

## Tópicos

- [Proteção de dados no Elastic Beanstalk](#)
- [Gerenciamento de identidade e acesso para o Elastic Beanstalk](#)
- [Registrar em log e monitorar no Elastic Beanstalk](#)
- [Validação de conformidade para o Elastic Beanstalk](#)
- [Resiliência no Elastic Beanstalk](#)
- [Segurança da infraestrutura no Elastic Beanstalk](#)
- [Análise de configuração e vulnerabilidade no Elastic Beanstalk](#)
- [Melhores práticas de segurança para Elastic Beanstalk](#)

# Proteção de dados no Elastic Beanstalk

O [modelo de responsabilidade compartilhada](#) da AWS se aplica à proteção de dados no AWS Elastic Beanstalk. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para ter mais informações sobre a privacidade de dados, consulte as [Perguntas frequentes sobre privacidade de dados](#). Para ter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog [AWS Shared Responsibility Model and GDPR](#) no Blog de segurança da AWS.

Para fins de proteção de dados, recomendamos que você proteja as credenciais da Conta da AWS e configure as contas de usuário individuais com o AWS IAM Identity Center ou o AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA [multi-factor authentication]) com cada conta.
- Use SSL/TLS para se comunicar com os recursos da AWS. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure o registro em log das atividades da API e do usuário com o AWS CloudTrail.
- Use as soluções de criptografia da AWS, juntamente com todos os controles de segurança padrão dos Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para ter mais informações sobre endpoints do FIPS disponíveis, consulte [Federal Information Processing Standard \(FIPS\) 140-2](#).

É altamente recomendável que você nunca coloque informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em tags ou campos de texto de formato livre, como um campo Name (Nome). Isso inclui o trabalho com o Elastic Beanstalk ou outros Serviços da AWS usando o console, a API, a AWS CLI ou os AWS SDKs. Quaisquer dados inseridos em marcações ou campos de texto de formato livre usados para nomes podem ser usados para logs de cobrança



ou diagnóstico. Se você fornecer um URL para um servidor externo, recomendamos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

Para outros tópicos de segurança do Elastic Beanstalk, consulte [Segurança do AWS Elastic Beanstalk](#).

## Tópicos

- [Proteção de dados usando criptografia](#)
- [Privacidade do tráfego entre redes](#)

## Proteção de dados usando criptografia

O Elastic Beanstalk armazena vários objetos em um bucket do Amazon Simple Storage Service (Amazon S3) criado por ele para cada região da AWS em que você cria ambientes. Para obter mais detalhes, consulte [the section called “Amazon S3”](#).

Forneça alguns dos objetos armazenados e envie-os para o Elastic Beanstalk, por exemplo, versões de aplicações e pacotes de origem. O Elastic Beanstalk gera outros objetos, por exemplo, arquivos de log. Além dos dados que o Elastic Beanstalk armazena, a aplicação pode transferir e/ou armazenar dados como parte de sua operação.

Proteção de dados refere-se aos dados em trânsito (à medida que são transferidos para e do Elastic Beanstalk) e em repouso (enquanto estão armazenados em datacenters da AWS).

## Criptografia em trânsito

É possível obter proteção de dados em trânsito de duas maneiras: criptografar a conexão usando Secure Sockets Layer (SSL) ou usar a criptografia do lado do cliente (onde o objeto é criptografado antes de ser enviado). Ambos os métodos são válidos para proteger os dados do aplicativo. Para proteger a conexão, criptografe-a usando SSL sempre que o aplicativo e seus desenvolvedores, administradores e usuários finais enviarem ou receberem objetos. Para obter detalhes sobre como criptografar o tráfego da web de e para o aplicativo, consulte [the section called “HTTPS”](#).

A criptografia do lado do cliente não é um método válido para proteger seu código-fonte em versões de aplicações e pacotes de origem carregados. O Elastic Beanstalk precisa de acesso a esses objetos, para que eles não possam ser criptografados. Assim, certifique-se de proteger a conexão entre o ambiente de desenvolvimento ou de implantação e o Elastic Beanstalk.

## Criptografia em repouso

Para proteger os dados em repouso do aplicativo, saiba mais sobre a proteção de dados no serviço de armazenamento usado pelo aplicativo. Por exemplo, consulte [Data Protection in Amazon RDS](#) (Proteção de dados no Amazon RDS) no Amazon RDS User Guide (Guia do usuário do Amazon RDS), [Proteção de dados no Amazon S3](#) no Guia do usuário do Amazon Simple Storage Service, ou [Criptografia de dados e metadados no EFS](#) no Guia do usuário do Amazon Elastic File System.

O Elastic Beanstalk não ativa a criptografia padrão para o bucket do Amazon S3 criado por ele. Isso significa que, por padrão, os objetos são armazenados descriptografados no bucket (e podem ser acessados somente por usuários autorizados na ler o bucket). Se o aplicativo exigir criptografia em repouso, você poderá configurar os buckets da conta para a criptografia padrão. Para obter mais informações, consulte [Criptografia padrão do Amazon S3 para buckets do S3](#) no Guia do usuário do Amazon Simple Storage Service.

Para obter mais informações sobre proteção de dados, consulte a publicação [Modelo de responsabilidade compartilhada da AWS e do GDPR](#) no Blog de segurança da AWS.

Para outros tópicos de segurança do Elastic Beanstalk, consulte [Segurança do AWS Elastic Beanstalk](#).

## Privacidade do tráfego entre redes

É possível usar o Amazon Virtual Private Cloud (Amazon VPC) para criar limites entre recursos em sua aplicação do Elastic Beanstalk e controlar o tráfego entre eles, sua rede on-premises e a internet. Para obter mais detalhes, consulte [the section called “Amazon VPC”](#).

Para obter mais informações sobre segurança da Amazon VPC, consulte [Segurança](#) no Guia do usuário da Amazon VPC.

Para obter mais informações sobre proteção de dados, consulte a publicação [Modelo de responsabilidade compartilhada da AWS e do GDPR](#) no Blog de segurança da AWS.

Para outros tópicos de segurança do Elastic Beanstalk, consulte [Segurança do AWS Elastic Beanstalk](#).

## Gerenciamento de identidade e acesso para o Elastic Beanstalk

O AWS Identity and Access Management (IAM) é um serviço da AWS que ajuda a controlar o acesso aos recursos da AWS de forma segura. Os administradores do IAM controlam quem pode ser

autenticado (conectado) e autorizado (ter permissões) a usar os recursos do AWS Elastic Beanstalk. O IAM é um serviço da AWS que pode ser usado sem custo adicional.

Para obter detalhes sobre como trabalhar com o IAM, consulte [Uso do Elastic Beanstalk com o AWS Identity and Access Management](#).

Para outros tópicos de segurança do Elastic Beanstalk, consulte [Segurança do AWS Elastic Beanstalk](#).

## AWS políticas gerenciadas para AWS Elastic Beanstalk

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque elas estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo [políticas gerenciadas pelo cliente da](#) específicas para seus casos de uso.


Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) é lançada ou novas operações de API são disponibilizadas para serviços existentes.

Para mais informações, consulte [Políticas gerenciadas pela AWS](#) no Manual do usuário do IAM.

## Atualizações do Elastic AWS Beanstalk nas políticas gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do Elastic Beanstalk desde 1º de março de 2021.

Para ver a fonte JSON de uma política gerenciada específica, consulte o [Guia de referência de políticas AWS gerenciadas](#).

Alteração	Descrição	Data
<p>As seguintes políticas foram atualizadas:</p> <ul style="list-style-type: none"> <li>• AWSElasticBeanstalkInternalMaintenanceRolePolicy</li> <li>• AWSElasticBeanstalkMaintenance</li> <li>• AWSElasticBeanstalkManagedUpdatesInternalServiceRolePolicy</li> <li>• AWSElasticBeanstalkManagedUpdatesServiceRolePolicy</li> <li>• AWSElasticBeanstalkRoleCore</li> </ul>	<p>Essas políticas foram atualizadas para permitir que o Elastic Beanstalk adicione ou remova tags ao criar ou atualizar uma pilha ou um conjunto de AWS CloudFormation alterações.</p> <p>Para obter mais informações sobre o AWSElasticBeanstalkManagedUpdatesServiceRolePolicy, consulte <a href="#">Permissões de função vinculada ao serviço para o Elastic Beanstalk</a>.</p> <p>Para obter mais informações sobre o AWSElasticBeanstalkRoleCore, consulte <a href="#">Políticas de integração com outros serviços</a>.</p>	<p>30 de abril de 2024</p>
<p>AWSElasticBeanstalkService— Política existente atualizada</p>	<p>Essa política foi atualizada para permitir que o Elastic Beanstalk marque recursos para o Elastic Load Balancing, os grupos do Auto Scaling (ASG) e o Amazon ECS ao criá-los.</p> <div data-bbox="591 1604 1029 1879" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Essa política foi substituída anteriormente pela AWSElasticBeanstalk</p> </div>	<p>10 de maio de 2023</p>

Alteração	Descrição	Data
	<p data-bbox="667 212 997 768">kManagedU pdatesCus tomerRole Policy . Embora essa política não esteja mais disponíve l para anexação a novos usuários, grupos ou perfis do IAM, ela ainda pode estar anexada aos que existiam antes.</p> <p data-bbox="591 884 967 1062">Para obter mais informaçõ es, consulte as <a href="#">Políticas gerenciadas de função de serviços</a>.</p>	
<p data-bbox="110 1104 542 1234">AWS Elastic Beanstalk Multi-Container Docker— Política existente atualizada</p>	<p data-bbox="591 1104 1016 1331">Essa política foi atualizada para permitir que o Elastic Beanstalk marque os recursos para o Amazon ECS ao criá-los.</p> <p data-bbox="591 1377 1021 1507">Para ter mais informações, consulte <a href="#">Gerenciar perfis de instância do Elastic Beanstalk</a>.</p>	<p data-bbox="1068 1104 1373 1142">23 de março de 2023</p>

Alteração	Descrição	Data
AWSElasticBeanstalkRoleECS — Política existente atualizada	<p>Essa política foi atualizada para permitir que o Elastic Beanstalk marque os recursos para o Amazon ECS ao criá-los.</p> <p>Para ter mais informações, consulte <a href="#">Políticas de integração com outros serviços</a>.</p>	23 de março de 2023
AdministratorAccess-AWSElasticBeanstalk — Política existente atualizada	<p>Essa política foi atualizada para permitir que o Elastic Beanstalk marque os recursos para o Amazon ECS ao criá-los.</p> <p>Para ter mais informações, consulte <a href="#">Gerenciar políticas de usuário do Elastic Beanstalk</a>.</p>	23 de março de 2023
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy — Política existente atualizada	<p>Essa política foi atualizada para permitir que o Elastic Beanstalk adicione tags aos recursos do Amazon ECS ao criá-los.</p> <p>Para ter mais informações, consulte <a href="#">Permissões de função vinculada ao serviço para o Elastic Beanstalk</a>.</p>	23 de março de 2023

Alteração	Descrição	Data
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Política existente atualizada	<p>Essa política foi atualizada para permitir que o Elastic Beanstalk adicione tags aos recursos do Amazon ECS ao criá-los.</p> <p>Para obter mais informações, consulte as <a href="#">Políticas gerenciadas de função de serviços</a>.</p>	23 de março de 2023
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy— Política existente atualizada	<p>Esta política foi atualizada para permitir que o Elastic Beanstalk adicione tags a grupos do Auto Scaling ao criá-los.</p> <p>Para ter mais informações, consulte <a href="#">A função vinculada ao serviço de atualizações gerenciadas</a>.</p>	27 de janeiro de 2023
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Política existente atualizada	<p>Essa política foi atualizada para permitir que o Elastic Beanstalk adicione tags na criação de um grupo do Auto Scaling (ASG).</p> <p>Para obter mais informações, consulte as <a href="#">Políticas gerenciadas de função de serviços</a>.</p>	23 de janeiro de 2023

Alteração	Descrição	Data
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Política existente atualizada	<p>Essa política foi atualizada para permitir que o Elastic Beanstalk adicione tags na criação de um Elastic Load Balancing (ELB).</p> <p>Para obter mais informações, consulte as <a href="#">Políticas gerenciadas de função de serviços</a>.</p>	21 de dezembro de 2022



Alteração	Descrição	Data
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy— Política existente atualizada	<p>As permissões foram adicionadas a esta política para permitir que o Elastic Beanstalk faça o seguinte durante as atualizações gerenciadas:</p> <ul style="list-style-type: none"><li>• Crie e exclua modelos de lançamento e versões de modelos.</li><li>• Execute instâncias do Amazon EC2 com modelos de execução.</li><li>• Se um Amazon RDS estiver presente, recupere uma lista dos mecanismos de banco de dados disponíveis e informações sobre instâncias do RDS provisionadas.</li></ul> <p>Para ter mais informações, consulte <a href="#">A função vinculada ao serviço de atualizações gerenciadas</a>.</p>	23 de agosto de 2022

Alteração	Descrição	Data
<p>AWSElasticBeanstalkReadOnlyAccess— Obsoleto</p> <p>GovCloud (EUA) Região da AWS</p>	<p>Esta política foi substituída por AWSElasticBeanstalkReadOnly .</p> <p>Essa política será eliminada gradualmente nos GovCloud (EUA) Região da AWS.</p> <p>Quando essa política for eliminada, ela não estará mais disponível para anexar a novos usuários, grupos ou funções do IAM após 17 de junho de 2021.</p> <p>Para obter mais informações, consulte as <a href="#">Políticas do usuário</a>.</p>	<p>17 de junho de 2021</p>
<p>AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Política existente atualizada</p>	<p>Essa política foi atualizada para permitir que o Elastic Beanstalk leia atributos para Zonas de disponibilidade do EC2. Ele habilita que o Elastic Beanstalk forneça uma validação mais eficaz da seleção do tipo de instância em Zonas de disponibilidade.</p> <p>Para obter mais informações, consulte as <a href="#">Políticas gerenciadas de função de serviços</a>.</p>	<p>16 de junho de 2021</p>

Alteração	Descrição	Data
AWSElasticBeanstalkFullAccess— Obsoleto  GovCloud (EUA) Região da AWS	<p>Esta política foi substituída por <code>AdministratorAccess-AWSElasticBeanstalk</code> .</p> <p>Essa política será eliminada gradualmente nos GovCloud (EUA) Região da AWS.</p> <p>Quando essa política for eliminada, ela não estará mais disponível para anexação a novos usuários, grupos ou funções do IAM após 10 de junho de 2021.</p> <p>Para obter mais informações, consulte as <a href="#">Políticas do usuário</a>.</p>	10 de junho de 2021

Alteração	Descrição	Data
<p>As seguintes políticas gerenciadas foram preteridas em todos os países da China: Região da AWS</p> <ul style="list-style-type: none"><li>• AWSElasticBeanstalkFullAccess</li><li>• AWSElasticBeanstalkReadOnlyAccess</li></ul>	<p>A política AWSElasticBeanstalkFullAccess foi substituída por AdministratorAccess-AWSElasticBeanstalk .</p> <p>A política AWSElasticBeanstalkReadOnlyAccess foi substituída por AWSElasticBeanstalkReadOnly .</p> <p>Essas políticas foram eliminadas gradualmente em todos os países da China Região da AWS.</p> <p>Essas políticas não estarão mais disponíveis para anexar a novos usuários, grupos ou funções do IAM após 3 de junho de 2021.</p> <p>Para obter mais informações, consulte as <a href="#">Políticas do usuário</a>.</p>	<p>3 de junho de 2021</p>

Alteração	Descrição	Data
AWSElasticBeanstalkService— Obsoleto	<p>Essa política foi substituída pela <code>AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy</code>.</p> <p>Essa política está sendo descontinuada e não está mais disponível a novos usuários, grupos ou funções do IAM para anexação.</p> <p>Para obter mais informações, consulte as <a href="#">Políticas gerenciadas de função de serviços</a>.</p>	Junho de 2021 a janeiro de 2022

Alteração	Descrição	Data
As seguintes políticas gerenciadas foram preteridas em todos os Região da AWS países, exceto na China e GovCloud (EUA):	A política <code>AWSElasticBeanstalkFullAccess</code> foi substituída por <code>AdministratorAccess-AWSElasticBeanstalk</code> .	16 de abril de 2021
<ul style="list-style-type: none"><li>• <code>AWSElasticBeanstalkFullAccess</code></li><li>• <code>AWSElasticBeanstalkReadOnlyAccess</code></li></ul>	A política <code>AWSElasticBeanstalkReadOnlyAccess</code> foi substituída por <code>AWSElasticBeanstalkReadOnly</code> .  Essas políticas foram eliminadas gradualmente em todos os Região da AWS anos, exceto na China e GovCloud (EUA).  Essas políticas não estarão mais disponíveis para anexar a novos usuários, grupos ou funções do IAM após 16 de abril de 2021.  Para obter mais informações, consulte as <a href="#">Políticas do usuário</a> .	

Alteração	Descrição	Data
<p>As seguintes políticas gerenciadas foram atualizadas:</p> <ul style="list-style-type: none"> <li>• AdministratorAccess-AWSElasticBeanstalk</li> <li>• AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy</li> </ul>	<p>Ambas as políticas agora oferecem suporte a PassRole permissões na China Regiões da AWS.</p> <p>Para obter mais informações sobre o AdministratorAccess-AWSElasticBeanstalk, consulte as <a href="#">Políticas do usuário</a>.</p> <p>Para obter mais informações sobre AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy, consulte as <a href="#">Políticas gerenciadas de função de serviços</a>.</p>	<p>9 de março de 2021</p>
<p>AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy – Nova política</p>	<p>O Elastic Beanstalk adicionou uma nova política para substituir a política gerenciada AWSElasticBeanstalkService.</p> <p>Essa nova política gerenciada melhora a segurança de seus recursos aplicando um conjunto mais restritivo de permissões.</p> <p>Para obter mais informações, consulte as <a href="#">Políticas gerenciadas de função de serviços</a>.</p>	<p>3 de março de 2021</p>

Alteração	Descrição	Data
O Elastic Beanstalk começou a monitorar as alterações	O Elastic Beanstalk começou a monitorar as mudanças nas políticas gerenciadas. AWS	1º de março de 2021

## Registrar em log e monitorar no Elastic Beanstalk

O monitoramento é importante para a manutenção da confiabilidade, da disponibilidade e da performance do AWS Elastic Beanstalk e das soluções da AWS. Você deve coletar dados de monitoramento de todas as partes da solução da AWS para depurar uma falha de vários pontos com mais facilidade, caso ocorra. A AWS fornece várias ferramentas para monitorar seus recursos do Elastic Beanstalk e responder a incidentes em potencial.

Para obter mais informações sobre monitoramento, consulte [Monitoramento de ambiente](#).

Para outros tópicos de segurança do Elastic Beanstalk, consulte [Segurança do AWS Elastic Beanstalk](#).

## Relatórios de integridade aprimorada

Os relatórios de integridade aprimorada são um recurso que você pode habilitar no ambiente para permitir que o Elastic Beanstalk obtenha informações adicionais sobre os recursos no ambiente. O Elastic Beanstalk analisa as informações reunidas para fornecer uma visão melhor da integridade geral do ambiente e ajudar na identificação de problemas que podem fazer com que sua aplicação se torne indisponível. Para obter mais informações, consulte [Monitoramento e relatório de integridade aprimorada](#).

## Logs de instâncias do Amazon EC2

As instâncias do Amazon EC2 no ambiente do Elastic Beanstalk geram logs que podem ser visualizados para solucionar problemas com sua aplicação ou arquivos de configuração. Os logs criados pelo servidor Web, pelo servidor de aplicações, pelos scripts da plataforma do Elastic Beanstalk e pelo AWS CloudFormation são armazenados localmente em instâncias individuais. Você pode recuperá-los facilmente usando o [console de gerenciamento do ambiente](#) ou a CLI do EB. Também é possível configurar seu ambiente para fazer streaming de logs para o Amazon CloudWatch Logs em tempo real. Para obter mais informações, consulte [Visualizar logs de instâncias do Amazon EC2 no ambiente do Elastic Beanstalk](#).



## Notificações do ambiente

É possível configurar o ambiente do Elastic Beanstalk para usar o Amazon Simple Notification Service (Amazon SNS) para notificá-lo sobre eventos importantes que afetam a aplicação. Especifique um endereço de e-mail durante ou após a criação do ambiente para receber mensagens da AWS quando ocorrer um erro ou quando a integridade do ambiente mudar. Para obter mais informações, consulte [Notificações do ambiente do Elastic Beanstalk com o Amazon SNS](#).

## Alertas do Amazon CloudWatch

Ao usar alarmes do CloudWatch, você observa uma única métrica durante um período especificado. Se a métrica ultrapassar um limite especificado, uma notificação será enviada para um tópico do Amazon SNS ou para uma política do AWS Auto Scaling. Os alarmes do CloudWatch não invocam ações só porque estão em um determinado estado. Em vez disso, os alarmes do CloudWatch invocam ações quando o estado mudou e foi mantido por um número específico de períodos. Para obter mais informações, consulte [Usar o Elastic Beanstalk com o Amazon CloudWatch](#).

## Logs do AWS CloudTrail

O CloudTrail fornece um registro de ações executadas por um usuário, uma função ou um produto da AWS no Elastic Beanstalk. Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita para o Elastic Beanstalk, o endereço IP no qual foi feita a solicitação, quem a fez e quando ela foi feita, além de detalhes adicionais. Para obter mais informações, consulte [Registro em log de chamadas da API do Elastic Beanstalk com AWS CloudTrail](#).

## Depuração do AWS X-Ray

O X-Ray é um produto da AWS que coleta dados sobre as solicitações que o aplicativo atende, e os usa para construir um mapa de serviços que pode ser usado para identificar problemas no aplicativo e oportunidades de otimização. Use o console do AWS Elastic Beanstalk ou um arquivo de configuração para executar o daemon do X-Ray nas instâncias do ambiente. Para obter mais informações, consulte [Configurar depuração do AWS X-Ray](#).

## Validação de conformidade para o Elastic Beanstalk

Audidores terceirizados avaliam a segurança e a compatibilidade do AWS Elastic Beanstalk como parte de vários programas de compatibilidade da AWS. Esses programas incluem o SOC, o PCI, o FedRAMP, o HIPAA e outros. A AWS fornece uma lista atualizada dos produtos da AWS no escopo

de programas de compatibilidade específicos em [Produtos da AWS no escopo pelo programa de compatibilidade](#).

Os relatórios de auditoria de terceiros estão disponíveis para fazer download por meio do AWS Artifact. Para obter mais informações, consulte [Fazer download dos relatórios no AWS Artifact](#).

Para obter mais informações sobre programas de conformidade da AWS, consulte [Programas de conformidade da AWS](#).

Sua responsabilidade de conformidade ao usar o Elastic Beanstalk é determinada pela confidencialidade de seus dados, pelas metas de conformidade da sua empresa e pelas regulamentações e leis aplicáveis. Se o uso do Elastic Beanstalk estiver sujeito à compatibilidade com padrões, como HIPAA, PCI ou FedRAMP, a AWS fornecerá os recursos para ajudar:

- [Guias de início rápido de segurança e compatibilidade](#): guias de implantação que discutem as considerações de arquitetura e fornecem etapas para implantação de ambientes de linha de base focados em conformidade e segurança na AWS.
- [Arquitetura para segurança e compatibilidade com a HIPAA](#): um whitepaper que descreve como as empresas podem usar a AWS para criar aplicações em compatibilidade com a HIPAA.
- [Recursos de compatibilidade da AWS](#): uma coleção de guias e manuais de compatibilidade que pode ser aplicada ao seu setor e local.
- [AWS Config](#): um serviço que avalia até que ponto suas configurações de recursos estão compatíveis com práticas internas, diretrizes do setor e regulamentações.
- [AWS Security Hub](#): uma visão abrangente do estado da segurança na AWS que ajuda você a conferir a compatibilidade com os padrões e as práticas recomendadas do setor de segurança.

Para outros tópicos de segurança do Elastic Beanstalk, consulte [Segurança do AWS Elastic Beanstalk](#).

## Resiliência no Elastic Beanstalk

A infraestrutura global da AWS é criada com base em regiões e zonas de disponibilidade da AWS.

As regiões da AWS fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, que são conectadas com baixa latência, altas taxas de transferência e redes altamente redundantes.

Com as zonas de disponibilidade, você pode projetar e operar aplicativos e bancos de dados que executam o failover automaticamente entre as zonas de disponibilidade sem interrupção. As zonas

de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para mais informações sobre regiões e zonas de disponibilidade da AWS, consulte [Infraestrutura global da AWS](#).

O AWS Elastic Beanstalk gerencia e automatiza o uso da infraestrutura global da AWS em seu nome. Ao usar o Elastic Beanstalk, você se beneficia dos mecanismos de disponibilidade e tolerância a falhas oferecidos pela AWS.

Para outros tópicos de segurança do Elastic Beanstalk, consulte [Segurança do AWS Elastic Beanstalk](#).

## Segurança da infraestrutura no Elastic Beanstalk

Como um serviço gerenciado, o AWS Elastic Beanstalk é protegido pelos procedimentos de segurança de rede global da AWS que estão descritos no whitepaper [Amazon Web Services: Overview of Security Processes](#).

Use chamadas de API publicadas pela AWS para acessar o Elastic Beanstalk por meio da rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.0 ou posterior. Recomendamos o TLS 1.2 ou posterior. Os clientes também devem oferecer suporte a pacotes de criptografia com Perfect Forward Secrecy (PFS — Sigilo de encaminhamento perfeito), como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria das plataformas modernas como Java 7 e versões posteriores oferece suporte a esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Para outros tópicos de segurança do Elastic Beanstalk, consulte [Segurança do AWS Elastic Beanstalk](#).

## Análise de configuração e vulnerabilidade no Elastic Beanstalk

A AWS e nossos clientes compartilham responsabilidade por alcançar um alto nível de segurança e compatibilidade de componentes de software. O AWS Elastic Beanstalk ajuda você a executar

seu lado do modelo de responsabilidade compartilhada fornecendo um recurso de atualizações gerenciadas. Esse recurso aplica automaticamente atualizações secundárias e patches para uma versão da plataforma compatível do Elastic Beanstalk.

Para obter mais informações, consulte [Modelo de responsabilidade compartilhada para manutenção da plataforma do Elastic Beanstalk](#).

Para outros tópicos de segurança do Elastic Beanstalk, consulte [Segurança do AWS Elastic Beanstalk](#).

## Melhores práticas de segurança para Elastic Beanstalk

O AWS Elastic Beanstalk fornece vários recursos de segurança a serem considerados no desenvolvimento e na implementação das suas próprias políticas de segurança. As melhores práticas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas práticas recomendadas podem não ser adequadas ou suficientes no seu ambiente, trate-as como considerações úteis, não como requisitos.

Para outros tópicos de segurança do Elastic Beanstalk, consulte [Segurança do AWS Elastic Beanstalk](#).

### Práticas recomendadas de segurança preventiva

Os controles de segurança preventiva tentam evitar incidentes antes que ocorram.

#### Implemente o privilégio de acesso mínimo

O Elastic Beanstalk fornece políticas gerenciadas do AWS Identity and Access Management (IAM) para [perfis de instância](#), [funções de serviço](#) e [usuários do IAM](#). Essas políticas gerenciadas especificam todas as permissões que podem ser necessárias para a operação correta do ambiente e do aplicativo.

O aplicativo pode não exigir todas as permissões em nossas políticas gerenciadas. É possível personalizá-los e conceder apenas as permissões necessárias para que as instâncias do ambiente, o serviço do Elastic Beanstalk e os usuários executem suas tarefas. Isso é especialmente relevante para políticas de usuário, em que diferentes funções de usuário podem ter diferentes necessidades de permissão. A implementação do privilégio de acesso mínimo é fundamental para reduzir o risco de segurança e o impacto que pode resultar de erros ou usuários mal-intencionados.

## Atualizar as plataformas regularmente

O Elastic Beanstalk lança regularmente novas versões de plataforma para atualizar todas as suas plataformas. As novas versões de plataforma fornecem atualizações de sistema operacional, tempo de execução, servidor de aplicações, servidor Web e atualizações de componentes do Elastic Beanstalk. Muitas dessas atualizações de plataforma incluem correções de segurança importantes. Certifique-se de que os ambientes do Elastic Beanstalk estejam em execução em uma versão compatível da plataforma (normalmente a versão mais recente da plataforma). Para obter mais detalhes, consulte [Atualizar a versão de plataforma do ambiente Elastic Beanstalk](#).

A maneira mais fácil de manter a plataforma do ambiente atualizada é configurar o ambiente para usar [atualizações gerenciadas de plataforma](#).

## Aplicar o IMDSv2 em instâncias de ambiente

As instâncias do Amazon Elastic Compute Cloud (Amazon EC2) em seus ambientes do Elastic Beanstalk usam o serviço de metadados da instância (IMDS), um componente na instância, para acessar com segurança os metadados da instância. O IMDS oferece suporte a dois métodos para acessar dados: IMDSv1 e IMDSv2. O IMDSv2 usa solicitações orientadas para sessão e mitiga vários tipos de vulnerabilidades que podem ser usadas para tentar acessar o IMDS. Para obter detalhes sobre as vantagens do IMDSv2, consulte [melhorias na adição de defesa profunda ao serviço de metadados da instância do EC2](#).

O IMDSv2 é mais seguro, por isso é uma boa ideia aplicar o uso do IMDSv2 em suas instâncias. Para aplicar o IMDSv2, verifique se todos os componentes do aplicativo oferecem suporte ao IMDSv2 e desabilite o IMDSv1. Para obter mais informações, consulte [the section called "IMDS"](#).

## Práticas recomendadas de segurança de detecção

Os controles de segurança de detecção identificam violações de segurança depois de ocorrerem. Eles podem ajudar a detectar uma possível ameaça ou incidente de segurança.

## Implementar o monitoramento

O monitoramento é uma parte importante para manter a confiabilidade, a segurança, a disponibilidade e a performance das soluções do Elastic Beanstalk. A AWS fornece várias ferramentas e serviços para ajudar a monitorar os serviços da AWS.

Veja a seguir alguns exemplos de itens a serem monitorados:

- Métricas do Amazon CloudWatch para Elastic Beanstalk: defina alarmes para as principais métricas do Elastic Beanstalk e para as métricas personalizadas da sua aplicação. Para obter mais detalhes, consulte [Usar o Elastic Beanstalk com o Amazon CloudWatch](#).
- Entradas do AWS CloudTrail - Acompanhe as ações que podem afetar a disponibilidade, como UpdateEnvironment ou TerminateEnvironment. Para obter mais detalhes, consulte [Registro em log de chamadas da API do Elastic Beanstalk com AWS CloudTrail](#).

## Habilitar o AWS Config

O AWS Config fornece uma visão detalhada da configuração dos recursos da AWS na conta. Você pode ver como os recursos estão relacionados, obter um histórico de alterações de configuração e ver como os relacionamentos e as configurações foram alterados ao longo do tempo.

É possível usar o AWS Config para definir regras que avaliam configurações de recursos para compatibilidade de dados. As regras do AWS Config representam as definições de configuração ideais para os recursos do Elastic Beanstalk. Se um recurso violar uma regra e for sinalizado como não compatível, o AWS Config poderá alertá-lo usando um tópico do Amazon Simple Notification Service (Amazon SNS). Para obter mais detalhes, consulte [Localizar e acompanhar recursos do Elastic Beanstalk com o AWS Config](#).

# Solução de problemas

Este capítulo fornece orientações para a solução de problemas com o ambiente do Elastic Beanstalk. Ele fornece as informações a seguir.

- Uma introdução à ferramenta AWS Systems Manager, além de um procedimento para executar um runbook predefinido do Elastic Beanstalk que gera etapas e recomendações de solução de problemas.
- Orientação geral sobre ações que você poderá tomar e recursos que poderá visualizar se o status do seu ambiente se deteriorar.
- Dicas de solução de problemas mais específicas por categoria de assunto.

Se a integridade do seu ambiente mudar para vermelho, recomendamos usar primeiro a ferramenta AWS Systems Manager, que inclui runbooks predefinidos para solucionar problemas do Elastic Beanstalk. Para obter mais informações, consulte [Usar a ferramenta Systems Manager](#) na próxima seção deste capítulo.

## Tópicos

- [Usar os runbooks do AWS Systems Manager Elastic Beanstalk](#)
- [Orientações gerais](#)
- [Categorias](#)

## Usar os runbooks do AWS Systems Manager Elastic Beanstalk

Você pode usar o Systems Manager para solucionar problemas em seus ambientes do Elastic Beanstalk. Para ajudar você a começar rapidamente, o Systems Manager fornece runbooks de automação predefinidos para o Elastic Beanstalk. Um runbook de automação é um tipo de documento do Systems Manager que define ações a serem executadas nas instâncias e em outros recursos da AWS do seu ambiente.

O documento `AWSSupport-TroubleshootElasticBeanstalk` é um runbook de automação desenvolvido para ajudar a identificar vários problemas comuns que podem degradar seu ambiente do Elastic Beanstalk. Para fazer isso, ele verifica os componentes do seu ambiente, incluindo os seguintes: instâncias do EC2, VPC, pilha do AWS CloudFormation, balanceadores de carga, grupos

do Auto Scaling e configuração de rede associada a regras de grupos de segurança, tabelas de rotas e ACLs.

Ele também oferece a opção de fazer upload de arquivos de log agrupados do seu ambiente para o AWS Support.

Para obter mais informações, consulte [AWSSupport-TroubleshootElasticBeanstalk](#) na Referência do runbook do AWS Systems Manager Automation.

## Usar o Systems Manager para executar o runbook do **AWSSupport-TroubleshootElasticBeanstalk**

### Note

Execute esse procedimento na mesma Região da AWS em que seu ambiente do Elastic Beanstalk está localizado.

1. Abra o console do [AWS Systems Manager](#).
2. No painel de navegação, em Gerenciamento de alterações, escolha Automação.
3. Escolha Execute automation.
4. Na guia Pertencente à Amazon, na caixa de pesquisa Documentos de automação, insira `AWSSupport-TroubleshootElasticBeanstalk`.
5. Selecione o cartão `AWSSupport-TroubleshootElasticBeanstalk` e, em seguida Avançar.
6. Selecione Executar.
7. Na seção Parâmetros de entrada:
  - a. No menu suspenso `AutomationAssumeRole`, selecione o ARN do perfil que permite que o Systems Manager execute ações em seu nome.
  - b. Em `ApplicationName`, insira o nome da aplicação do Elastic Beanstalk.
  - c. Em Nome do ambiente, insira o ambiente do Elastic Beanstalk.
  - d. (Opcional) Para `S3UploaderLink`, insira um link se um engenheiro do AWS Support tiver fornecido um link do S3 para coleta de log.
8. Escolha Executar.

Se alguma das etapas falhar, selecione o link na coluna ID da etapa da etapa que falhou. Fazer isso exibe uma página Detalhes da execução para a etapa. A seção `VerificationErrorMessage`



exibirá um resumo das etapas que exigem atenção. Por exemplo, IAMPermissionCheck pode exibir uma mensagem de aviso. Nesse caso, é possível verificar se a função selecionada no menu suspenso AutomationAssumeRole tem as permissões necessárias.

Depois que todas as etapas forem concluídas com êxito, o resultado fornecerá etapas de solução de problemas e recomendações para restaurar seu ambiente a um estado íntegro.

## Orientações gerais

As mensagens de erro podem ser exibidas na página Eventos no console, em logs ou na página Integridade. Você também pode tomar medidas para se recuperar de um ambiente degradado causado por uma alteração recente. Se a integridade do seu ambiente mudar para vermelho, tente o seguinte:

- Analise os [eventos](#) recentes do ambiente. As mensagens do Elastic Beanstalk sobre problemas de implantação, carregamento e configuração geralmente aparecem aqui.
- Analise o recente [histórico de alterações](#) do ambiente. O histórico de alterações lista todas as alterações de configuração feitas nos seus ambientes e inclui outras informações, como qual usuário do IAM fez alterações e quais parâmetros de configuração foram definidos.
- [Extraia os logs](#) para visualizar as entradas recentes do arquivo de log. Os logs do servidor Web contêm informações sobre as solicitações de entrada e os erros.
- [Conecte-se a uma instância](#) e verifique os recursos do sistema.
- [Reverta](#) para uma versão anterior válida da aplicação.
- Desfaça as alterações de configuração recentes ou restaure uma [configuração salva](#).
- Implante um novo ambiente. Se ele parecer íntegro, execute uma [troca de CNAME](#) para rotear o tráfego para o novo ambiente e continuar depurando o anterior.

## Categorias

Este tópico fornece dicas de solução de problemas mais específicas por categoria.

### Tópicos

- [Conectividade](#)
- [Criação de ambiente e execução de instâncias](#)
- [Implantações](#)

- [Integridade](#)
- [Configuração](#)
- [Solução de problemas de contêineres do Docker](#)
- [Perguntas frequentes](#)

## Conectividade

Problema: os servidores criados no console do Elastic Beanstalk não aparecem no Toolkit for Eclipse

Você pode importar os servidores manualmente, de acordo com as instruções em [Importar ambientes existentes no Eclipse](#).

Problema: não é possível conectar ao Amazon RDS a partir do Elastic Beanstalk.

Para conectar um Amazon RDS desacoplado à sua aplicação Elastic Beanstalk, faça o seguinte:

- Certifique-se de que o RDS esteja na mesma região que a aplicação do Elastic Beanstalk.
- Verifique se o grupo de segurança do RDS da sua instância tem uma autorização para o grupo de segurança do Amazon EC2 que você está usando no ambiente do Elastic Beanstalk. Para obter instruções sobre como encontrar o nome do grupo de segurança do EC2 usando o Console de Gerenciamento da AWS, consulte [Grupos de segurança](#). Para obter mais informações sobre como configurar o grupo de segurança do EC2, acesse a seção “Autorização de acesso à rede para um grupo de segurança do Amazon EC2” de [Trabalhar com grupos de segurança de banco de dados](#) no Guia do usuário do Amazon Relational Database Service.
- Para Java, certifique-se de que o arquivo JAR MySQL está em seu WEB-INF/lib. Consulte [Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente de aplicações Java](#) para obter mais detalhes.

## Criação de ambiente e execução de instâncias

Evento: falha ao executar o ambiente

Este evento ocorre quando o Elastic Beanstalk tenta executar um ambiente e encontra falhas nesse processo. Os eventos anteriores na página Eventos irão alertá-lo para a causa raiz do problema.

Evento: operação de criação do ambiente concluída, mas com tempos limites de comando excedidos. Tente aumentar o tempo limite.

A implantação do seu aplicativo poderá levar muito tempo se você usar arquivos de configuração que executam comandos na instância, fizer download de arquivos grandes ou instalar pacotes. Aumente o [tempo limite do comando](#) para dar ao aplicativo mais tempo para iniciar a execução durante as implantações.

Evento: falha na criação do(s) recurso(s) a seguir: [AWSEBInstanceLaunchWaitCondition]

Essa mensagem indica que as instâncias do Amazon EC2 de seu ambiente não comunicaram ao Elastic Beanstalk que foram executadas com êxito. Isso pode ocorrer se as instâncias não têm conectividade com a Internet. Se você configurou o ambiente para executar instâncias em uma sub-rede da VPC privada, [a sub-rede precisará ter um NAT](#) para permitir que as instâncias se conectem ao Elastic Beanstalk.

Evento: um perfil de serviço é necessário nessa região. Adicione uma opção de perfil de serviço ao ambiente.

O Elastic Beanstalk usa uma função de serviço para monitorar os recursos no ambiente e oferecer suporte às [atualizações gerenciadas da plataforma](#). Consulte [Gerenciar funções de serviço do Elastic Beanstalk](#) para obter mais informações.

## Implantações

Problema: o aplicativo fica indisponível durante as implantações

Como o Elastic Beanstalk usa um processo de atualização "drop-in", pode haver alguns segundos de tempo de inatividade. Use as [implantações contínuas](#) para minimizar o efeito das implantações em seus ambientes de produção.

Evento: Falha ao criar a versão da aplicação do AWS Elastic Beanstalk

Seu pacote de origem do aplicativo pode ser muito grande ou você pode ter atingido a [cota de versões do aplicativo](#).

Evento: operação de atualização do ambiente concluída, mas com tempos limites de comando excedidos. Tente aumentar o tempo limite.

A implantação do seu aplicativo poderá levar muito tempo se você usar arquivos de configuração que executam comandos na instância, fizer download de arquivos grandes ou instalar pacotes. Aumente o [tempo limite do comando](#) para dar ao aplicativo mais tempo para iniciar a execução durante as implantações.

## Integridade

Evento: utilização da CPU excede 95%

Tente [executar mais instâncias](#) ou [escolher outro tipo de instância](#).

Evento: Elastic Load Balancer *awseb-myapp* sem instância de integridade

Se o aplicativo parece estar funcionando, verifique se o URL de verificação de integridade do aplicativo está configurado corretamente. Caso contrário, verifique os logs de ambiente e tela de integridade para obter mais informações.

Evento: Elastic Load Balancer *awseb-myapp* não foi encontrado

O load balancer do ambiente pode ter sido removido fora de banda. Somente faça alterações nos recursos do ambiente com as opções de configuração e [extensibilidade](#) fornecidas pelo Elastic Beanstalk. Reconstrua o ambiente ou inicie um novo.

Evento: falha ao iniciar instância de EC2. Aguardando uma nova instância de EC2 para iniciar...

A disponibilidade para o tipo de instância do seu ambiente pode estar baixa ou você pode ter atingido a cota de instâncias da sua conta. Verifique o [Painel de status dos serviços](#) para garantir que o serviço Elastic Compute Cloud (Amazon EC2) esteja verde ou [solicite um aumento de cota](#).

## Configuração

Evento: você não pode configurar um ambiente do Elastic Beanstalk com valores para as ambas as opções, Elastic Load Balancing Target (Destino do balanceamento de carga elástico) e Application Healthcheck URL (URL para verificação de integridade da aplicação)

A opção Target no namespace `aws:elb:healthcheck` está obsoleta. Remova o namespace da opção Target de seu ambiente e tente atualizar novamente.

Evento: O ELB não pode ser anexado a várias sub-redes na mesma zona de disponibilidade.

Esta mensagem poderá ser visualizada se você tentar mover um load balancer entre sub-redes na mesma zona de disponibilidade. A alteração de sub-redes no load balancer requer movê-lo para fora da(s) zona(s) de disponibilidade original(is) e, em seguida, de volta à original com as sub-redes desejadas. Durante o processo, todas as instâncias são migradas entre as zonas de disponibilidade, causando um tempo de inatividade considerável. Em vez disso, considere criar um novo ambiente e [executar uma troca CNAME](#).

## Solução de problemas de contêineres do Docker

Evento: falha ao extrair imagem do Docker: mais recentes: nome do repositório inválido (), apenas [a-z0-9-\_.] são permitidos. Siga os logs para obter mais detalhes.

Verifique a sintaxe do arquivo `dockerrun.aws.json` usando um validador JSON. Verifique também o conteúdo do `dockerfile` em relação aos requisitos descritos em [Configuração do Docker](#)

Evento: Nenhuma diretiva EXPOSE encontrada no Dockerfile, abortar a implantação

O arquivo `Dockerfile` ou `dockerrun.aws.json` não declara a porta do contêiner. Use a instrução EXPOSE (`Dockerfile`) ou Ports bloco (arquivo `dockerrun.aws.json`) para expor uma porta do tráfego de entrada.

Evento: Falha de download *do repositório* de credenciais de autenticação do *nome do bucket*

O `dockerrun.aws.json` fornece um par de chaves do EC2 inválido e/ou bucket do S3 para o arquivo `.dockercfg`. Ou, o perfil da instância não tem a autorização `GetObject` para o bucket do S3. Verifique se o arquivo `.dockercfg` contém um bucket do S3 e um par de chaves do EC2 válidos. Conceda permissões da ação `s3:GetObject` à função do IAM no perfil da instância. Para detalhes, consulte [Gerenciar perfis de instância do Elastic Beanstalk](#)

Evento: Falha na execução de atividade, porque: AVISO: arquivo de configuração de autorização inválido

Seu arquivo de autenticação (`config.json`) não está formatado corretamente. Consulte [Usar imagens de um repositório privado](#)

## Perguntas frequentes

Pergunta: Como posso alterar o URL do meu aplicativo de `myapp.us-west-2.elasticbeanstalk.com` para `www.myapp.com`?

Em um servidor DNS, registre um registro CNAME, como **`www.mydomain.com CNAME mydomain.elasticbeanstalk.com`**.

Pergunta: Como especifico uma zona de disponibilidade exclusiva para minha aplicação do Elastic Beanstalk?

Você pode escolher zonas de disponibilidade específicas usando as APIs, a CLI e o plug-in do Eclipse ou o plug-in do Visual Studio. Para obter instruções sobre como usar o console do Elastic

Beanstalk para especificar uma zona de disponibilidade, consulte [Grupo do Auto Scaling para o seu ambiente do Elastic Beanstalk](#).

Pergunta: Como posso alterar o tipo de instância do meu ambiente?

Para alterar o tipo de instância do ambiente, acesse a página de configuração do ambiente e escolha Edit (Editar) na categoria de configuração Instances (Instâncias). Selecione um novo tipo de instância e escolha Apply (Aplicar) para atualizar o ambiente. Depois disso, o Elastic Beanstalk encerra todas as instâncias em execução e as substitui por novas.

Pergunta: Como determino se alguém fez alterações de configuração em um ambiente?

Para ver essas informações, no painel de navegação do console do Elastic Beanstalk, escolha Change history (Alterar histórico) para exibir uma lista de alterações de configuração para todos os ambientes. Essa lista inclui a data e a hora da alteração, o parâmetro de configuração e o valor para o qual ele foi alterado e o usuário do IAM que fez a alteração. Para obter mais informações, consulte [Histórico de alterações](#).

Pergunta: posso impedir que os volumes do Amazon EBS sejam excluídos quando as instâncias são encerradas?

As instâncias do ambiente usam o Amazon EBS para armazenamento; no entanto, o volume raiz é excluído quando uma instância é encerrada pelo Auto Scaling. Nós não recomendamos que você armazene o estado ou outros dados em suas instâncias. Se necessário, você poderá impedir que volumes sejam excluídos com a AWS CLI: `$ aws ec2 modify-instance-attribute -b '/dev/sdc=<vol-id>:false` como descrito na Referência da [AWS CLI](#).

Pergunta: como faço para excluir informações pessoais da aplicação do Elastic Beanstalk?

Os recursos da AWS que a aplicação do Elastic Beanstalk usa podem armazenar informações pessoais. Quando um ambiente é encerrado, o Elastic Beanstalk encerra os recursos que ele criou. Os recursos que você adicionou usando os [arquivos de configuração](#) também são encerrados. No entanto, se você tiver criado recursos da AWS fora do ambiente do Elastic Beanstalk e os tiver associado à sua aplicação, talvez você precise conferir manualmente que as informações pessoais que a aplicação possa ter armazenado não sejam retidas. Ao longo deste guia do desenvolvedor, sempre que discutimos a criação de recursos adicionais, também mencionamos quando você deve pensar em excluí-los.

# Recursos do Elastic Beanstalk

Os recursos relacionados a seguir podem ajudar você à medida que trabalha com este serviço.

- [Referência da API do Elastic Beanstalk](#) uma descrição abrangente de todas as APIs SOAP e Query. Além disso, ele contém uma lista de todos os tipos de dados SOAP.
- [elastic-beanstalk-samples on GitHub](#) — Um GitHub repositório com exemplos de arquivos de configuração do Elastic Beanstalk (.ebextensions). O README .md arquivo do repositório tem links para GitHub repositórios adicionais com aplicativos de amostra.
- [Perguntas técnicas frequentes sobre o Elastic Beanstalk](#): as principais perguntas dos desenvolvedores sobre esse produto.
- [AWS Elastic Beanstalk Notas de lançamento](#) — Detalhes sobre novos recursos, atualizações e correções nas versões de serviço, plataforma, console e EB CLI do Elastic Beanstalk.
- [Aulas e workshops](#) — Links para cursos especializados e baseados em funções, além de laboratórios individualizados para ajudar a aprimorar suas AWS habilidades e ganhar experiência prática.
- [AWS Centro do desenvolvedor](#) — explore tutoriais, baixe ferramentas e saiba mais sobre eventos para AWS desenvolvedores.
- [AWS Ferramentas para desenvolvedores](#) — Links para ferramentas de desenvolvedor, SDKs, kits de ferramentas de IDE e ferramentas de linha de comando para desenvolver e gerenciar AWS aplicativos.
- [Centro de recursos de introdução](#) — Saiba como configurar seu aplicativo Conta da AWS, participar da AWS comunidade e lançar seu primeiro aplicativo.
- [Tutoriais práticos — Siga os tutoriais](#) para iniciar seu step-by-step primeiro aplicativo no. AWS
- [AWS Whitepapers](#) — Links para uma lista abrangente de AWS white papers técnicos, abrangendo tópicos como arquitetura, segurança e economia e criados por arquitetos de AWS soluções ou outros especialistas técnicos.
- [AWS Support Center](#) — O hub para criar e gerenciar seus AWS Support casos. Também inclui links para outros recursos úteis, como fóruns, perguntas frequentes técnicas, status de integridade do serviço e. AWS Trusted Advisor
- [AWS Support](#)— A principal página da web com informações sobre AWS Supportum one-on-one canal de suporte de resposta rápida para ajudá-lo a criar e executar aplicativos na nuvem.

- [Entrar em contato](#): um ponto central de contato para consultas relativas a faturas da AWS , contas, eventos, uso abusivo e outros problemas.
- [AWS Termos do site](#) — Informações detalhadas sobre nossos direitos autorais e nossa marca registrada; sua conta, licença e acesso ao site; e outros tópicos.

## Aplicações de exemplo

Veja a seguir os links de download para os aplicativos de exemplo que são implantados como parte do [Conceitos básicos do Elastic Beanstalk](#).

### Note

Alguns exemplos usam recursos que podem ter sido lançados após a versão da plataforma que você está usando. Se houver falha na execução do exemplo, tente atualizar sua plataforma para uma versão mais recente, como descrito em [the section called “Plataformas compatíveis”](#).

- Docker: [docker.zip](#)
- Docker com vários contêineres — [2.zip docker-multicontainer-v](#)
- Docker pré-configurado (Glassfish) — [1.zip docker-glassfish-v](#)
- Go: [go.zip](#)
- Corretto: [corretto.zip](#)
- Tomcat: [tomcat.zip](#)
- .NET Core no Linux — [dotnet-core-linux.zip](#)
- .NET Core — [dotnet-asp-windows.zip](#)
- Node.js: [nodejs.zip](#)
- PHP: [php.zip](#)
- Python: [python.zip](#)
- Ruby: [ruby.zip](#)



# Histórico da plataforma

O histórico da Plataforma AWS Elastic Beanstalk foi movido. Consulte [Histórico da plataforma](#) no documento Plataformas do AWS Elastic Beanstalk.

## Tópicos

- [Plataformas personalizadas do Elastic Beanstalk](#)

## Plataformas personalizadas do Elastic Beanstalk

### Note

Em [18 de julho de 2022](#), o Elastic Beanstalk definiu o status de todas as ramificações de plataforma baseadas no Amazon Linux AMI (AL1) como descontinuadas. Isso inclui plataformas personalizadas. O Elastic Beanstalk não é compatível com plataformas personalizadas. Para obter mais informações sobre a descontinuação do Amazon Linux AMI no Elastic Beanstalk, consulte [Perguntas frequentes sobre descontinuação de plataformas](#).

Este tópico foi mantido neste documento como referência para todos os clientes que usavam o recurso de plataforma personalizada do Elastic Beanstalk antes de ser descontinuado. No passado, as plataformas personalizadas do Elastic Beanstalk eram compatíveis com a criação de uma AMI a partir das AMIs de base do Amazon Linux AMI, do RHEL 7, do RHEL 6 ou do Ubuntu 16.04. O Elastic Beanstalk não é mais compatível com esses sistemas operacionais. Para ler mais sobre o recurso de plataformas personalizadas, que não é mais compatível, expanda o tópico a seguir.

## Plataformas personalizadas

Uma plataforma personalizada é uma personalização mais avançada do que uma [imagem personalizada](#) de várias maneiras. Uma plataforma personalizada permite que você desenvolva toda uma nova plataforma do zero, personalizando o sistema operacional, software adicional e scripts executados pelo Elastic Beanstalk em instâncias de plataforma. Essa flexibilidade permite que você crie uma plataforma para uma aplicação que usa uma linguagem ou outro software de infraestrutura para o qual o Elastic Beanstalk não oferece uma plataforma gerenciada. Compare isso a imagens personalizadas, em que você modifica uma imagem de máquina da Amazon (AMI) para uso com

uma plataforma existente do Elastic Beanstalk, e o Elastic Beanstalk ainda oferece os scripts de plataforma e controla a pilha de software. Além disso, com plataformas personalizadas você usa uma maneira automatizada e com scripts de criar e manter a personalização, enquanto com imagens personalizadas você faz as alterações manualmente em uma instância em execução.

Para criar uma plataforma personalizada, crie uma AMI de um dos sistemas operacionais compatíveis: Ubuntu, RHEL ou Amazon Linux (consulte a entrada `flavor` em [Formato do arquivo `platform.yaml`](#) para ver os números de versão exatos), e inclua outras personalizações. Crie sua própria plataforma do Elastic Beanstalk usando o [Packer](#), que é uma ferramenta de código aberto para criar imagens de máquina para muitas plataformas, incluindo AMIs para uso com o Amazon Elastic Compute Cloud (Amazon EC2). Uma plataforma do Elastic Beanstalk consiste em uma AMI configurada para executar um conjunto de software compatível com uma aplicação e metadados que podem incluir opções de configuração personalizadas e opções de configuração padrão.

O Elastic Beanstalk gerencia o Packer como uma plataforma integrada separada, e não é necessário se preocupar com a configuração e as versões do Packer.

Crie uma plataforma fornecendo ao Elastic Beanstalk um modelo do Packer, e os scripts e arquivos que o modelo invoca para criar uma AMI. Esses componentes são empacotados com um [arquivo de definição de plataforma](#), que especifica o modelo e os metadados, em um arquivo ZIP, conhecido como [arquivo de definição de plataforma](#).

Quando você cria uma plataforma personalizada, inicia um ambiente de uma única instância sem um IP elástico que executa o Packer. O Packer executa outra instância para criar uma imagem. Você pode reutilizar esse ambiente para várias plataformas e várias versões de cada plataforma.

#### Note

As plataformas personalizadas são específicas da região da AWS. Se você usar o Elastic Beanstalk em várias regiões, deverá criar suas plataformas separadamente em cada região. Em determinadas circunstâncias, as instâncias executadas pelo Packer não são limpas e precisam ser manualmente encerradas. Para saber como limpar manualmente essas instâncias, consulte [Limpeza da instância do Packer](#).

Os usuários em sua conta podem utilizar suas Plataformas personalizadas especificando um [Nome de região da Amazon \(ARN\) da Plataforma](#) durante a criação do ambiente. Esses ARNs são retornados pelo comando `eb platform create` que você usou para criar a plataforma personalizada.

Cada vez que você cria sua plataforma personalizada, o Elastic Beanstalk cria uma nova versão dela. Os usuários podem especificar uma plataforma por nome para obter apenas a versão mais recente dela ou incluir um número de versão para obter uma versão específica.

Por exemplo, para implantar a versão mais recente da plataforma personalizada com o ARN **MyCustomPlatformARN**, que pode ser a versão 3.0, a linha de comando da CLI do EB seria assim:

```
eb create -p MyCustomPlatformARN
```

Para implantar a versão 2.1, a linha de comando da CLI do EB seria assim:

```
eb create -p MyCustomPlatformARN --version 2.1
```

Você pode aplicar tags a uma versão da plataforma personalizada ao criá-la e editar tags de versões existentes da plataforma personalizada. Para obter mais detalhes, consulte [Atribuir tags em versões de plataforma personalizada](#).

## Criar uma plataforma personalizada

Para criar uma plataforma personalizada, a raiz da sua aplicação deve incluir um arquivo de definição de plataforma `platform.yaml`, que define o tipo de criador usado para criar a plataforma personalizada. O formato desse arquivo está descrito no tópico [Formato do arquivo platform.yaml](#). Você pode criar sua Plataforma personalizada do zero ou usar uma das [Plataformas personalizadas de exemplo](#) como ponto de partida.

## Usar uma plataforma personalizada de exemplo

Uma alternativa para criar sua própria plataforma personalizada é usar um dos arquivos demonstrativos de definição de plataforma para inicializá-la. Os únicos itens que você precisa configurar nos exemplos para que você possa usá-los são uma AMI de origem e uma região.

### Note

Não use uma plataforma personalizada de exemplo não modificada em produção. O objetivo dos exemplos é mostrar algumas das funcionalidades disponíveis em uma plataforma personalizada, mas eles não foram reforçados para uso em produção.

### [NodePlatform\\_Ubuntu.zip](#)

Essa Plataforma personalizada é baseada no Ubuntu 16.04 e é compatível com o Node.js 4.4.4. Vamos usar essa plataforma personalizada para os exemplos nesta seção.

### [NodePlatform\\_RHEL.zip](#)

Essa Plataforma personalizada é baseada no RHEL 7.2 e é compatível com o Node.js 4.4.4.

### [NodePlatform\\_AmazonLinux.zip](#)

Essa Plataforma personalizada é baseada no Amazon Linux 2016.09.1 e é compatível com o Node.js 4.4.4.

### [TomcatPlatform\\_Ubuntu.zip](#)

Essa Plataforma personalizada é baseada no Ubuntu 16.04 e é compatível com o Tomcat 7/Java 8.

### [CustomPlatform\\_NodeSampleApp.zip](#)

Um exemplo de Node.js que usa express e ejs para exibir uma página da web estática.

### [CustomPlatform\\_TomcatSampleApp.zip](#)

Um exemplo de Tomcat que exibe uma página da web estática quando implantado.

Faça download do arquivo demonstrativo de definição da plataforma: `NodePlatform_Ubuntu.zip`. Esse arquivo contém um arquivo de definição de plataforma, um modelo do Packer, scripts executados pelo Packer durante a criação de imagens e scripts e arquivos de configuração copiados pelo Packer na instância do criador durante a criação da plataforma.

#### Example NodePlatform\_Ubuntu.zip

```
|-- builder Contains files used by Packer to create the custom platform
|-- custom_platform.json Packer template
|-- platform.yaml Platform definition file
|-- ReadMe.txt Briefly describes the sample
```

O arquivo de definição da plataforma `platform.yaml`, informa ao Elastic Beanstalk o nome do modelo do Packer `custom_platform.json`.

```
version: "1.0"

provisioner:
```

```
type: packer
template: custom_platform.json
flavor: ubuntu1604
```

O modelo do Packer instrui o próprio Packer a criar as AMIs para a plataforma usando uma [AMI do Ubuntu](#) como base para a imagem da plataforma para tipos de instâncias HVM. A seção `provisioners` instrui o Packer a copiar todos os arquivos na pasta `builder` dentro do arquivo para a instância e a executar o script `builder.sh` na instância. Quando os scripts são concluídos, o Packer cria uma imagem da instância modificada.

O Elastic Beanstalk cria três variáveis de ambiente que podem ser usadas para marcar as AMIs no Packer:

`AWS_EB_PLATFORM_ARN`

O ARN da plataforma personalizada.

`AWS_EB_PLATFORM_NAME`

O nome da plataforma personalizada.

`AWS_EB_PLATFORM_VERSION`

A versão da plataforma personalizada.

O arquivo `custom_platform.json` de exemplo usa essas variáveis para definir os seguintes valores que ele utiliza nos scripts:

- `platform_name`, que é definido por `platform.yaml`
- `platform_version`, que é definido por `platform.yaml`
- `platform_arn`, que é definido pelo script de construção principal, `builder.sh`, exibido no fim do arquivo `custom_platform.json` de exemplo.

O arquivo `custom_platform.json` contém duas propriedades para as quais você precisa fornecer valores: `source_ami` e `region`. Para obter detalhes sobre a escolha dos valores certos de região e AMI, consulte [Atualizar o modelo do Packer](#) no repositório `eb-custom-platforms-samples` do GitHub.

Example `custom_platform.json`

```
{
 "variables": {
```

```
"platform_name": "{{env `AWS_EB_PLATFORM_NAME`}}",
"platform_version": "{{env `AWS_EB_PLATFORM_VERSION`}}",
"platform_arn": "{{env `AWS_EB_PLATFORM_ARN`}}"
},
"builders": [
 {
 ...
 "region": "",
 "source_ami": "",
 ...
 }
],
"provisioners": [
 {...},
 {
 "type": "shell",
 "execute_command": "chmod +x {{ .Path }}; {{ .Vars }} sudo {{ .Path }}",
 "scripts": [
 "builder/builder.sh"
]
 }
]
```

Os scripts e outros arquivos incluídos no arquivo de definição de plataforma variam muito de acordo com as modificações que você deseja fazer na instância. A plataforma de exemplo inclui os seguintes scripts:

- `00-sync-apt.sh`: comentados: `apt -y update`. Nós comentamos fora do comando porque ele solicita a entrada ao usuário, que divide a atualização do pacote automatizado. Isso pode ser um problema no Ubuntu. No entanto, executar `apt -y update` ainda é a melhor prática recomendada. Por esse motivo, nós deixamos o comando no script de exemplo para referência.
- `01-install-nginx.sh`: instala o nginx.
- `02-setup-platform.sh`: instala `wget`, `tree` e `git`. Copia hooks e [configurações de registro](#) na instância e cria os seguintes diretórios:
  - `/etc/SampleNodePlatform`: onde é feito upload do arquivo de configuração do contêiner durante a implantação.
  - `/opt/elasticbeanstalk/deploy/appsource/`: onde o script `00-unzip.sh` carrega o código-fonte da aplicação durante a implantação (consulte a seção [Ferramentas de script de plataforma](#) para obter informações sobre esse script).

- `/var/app/staging/`: onde o código-fonte da aplicação é processado durante a implantação.
- `/var/app/current/`: onde o código-fonte da aplicação é executado após o processamento.
- `/var/log/nginx/healthd/`: onde o [agente de integridade aprimorada](#) grava os logs.
- `/var/nodejs`: onde é feito upload dos arquivos Node.js durante a implantação.

Use a CLI do EB para criar sua primeira plataforma personalizada com o arquivo demonstrativo de definição de plataforma.

Para criar uma plataforma personalizada

1. [Instale a CLI do EB](#).
2. Crie um diretório no qual você extrairá a plataforma personalizada de exemplo.

```
~$ mkdir ~/custom-platform
```

3. Extraia `NodePlatform_Ubuntu.zip` no diretório e, em seguida, altere para o diretório extraído.

```
~$ cd ~/custom-platform
~/custom-platform$ unzip ~/NodePlatform_Ubuntu.zip
~/custom-platform$ cd NodePlatform_Ubuntu
```

4. Edite o arquivo `custom_platform.json` e forneça valores para as propriedades `source_ami` e `region`. Para obter detalhes, consulte [Atualizar o modelo do Packer](#).
5. Execute [eb platform init](#) e siga os prompts para inicializar um repositório da Plataforma.

Você pode abreviar `eb platform` como `ebp`.

#### Note

O Windows PowerShell usa `ebp` como alias de comando. Se estiver executando a CLI do EB no Windows PowerShell, utilize a forma longa deste comando `eb platform`.

```
~/custom-platform$ eb platform init
```

Esse comando também cria o diretório `.elasticbeanstalk` no diretório atual e adiciona o arquivo de configuração `config.yml` ao diretório. Não altere nem exclua esse arquivo, porque o Elastic Beanstalk o utiliza para criar a plataforma personalizada.

Por padrão, o `eb platform init` usa o nome da pasta atual como o nome da plataforma personalizada, que é `custom-platform` neste exemplo.

6. Execute [eb platform create](#) para iniciar um ambiente Packer e obter o ARN da Plataforma personalizada. Você precisará desse valor no futuro quando criar um ambiente com base na plataforma personalizada.

```
~/custom-platform$ eb platform create
...
```

Por padrão, o Elastic Beanstalk cria o perfil da instância `aws-elasticbeanstalk-custom-platform-ec2-role` para plataformas personalizadas. Em vez disso, se você deseja usar um perfil da instância existente, adicione a opção `-ip` `INSTANCE_PROFILE` ao comando [eb platform create](#).

#### Note

O Packer não poderá criar uma plataforma personalizada se você usar o perfil da instância padrão do Elastic Beanstalk `aws-elasticbeanstalk-ec2-role`.

A EB CLI mostra a saída do evento do ambiente Packer até a conclusão da criação. Você pode sair da visualização do evento pressionando `Ctrl+C`.

7. Você pode consultar os logs para ver se há erros usando o comando [eb platform logs](#).

```
~/custom-platform$ eb platform logs
...
```

8. Você pode verificar o processo mais tarde com [eb platform events](#).

```
~/custom-platform$ eb platform events
...
```

9. Verifique o status da sua Plataforma com [eb platform status](#).



```
~/custom-platform$ eb platform status
...
```

Quando a operação for concluída, você terá uma plataforma que poderá ser usada para executar um ambiente do Elastic Beanstalk.

Você pode usar a plataforma personalizada ao criar um ambiente do console. Consulte [O assistente de criação de novo ambiente](#).

Para iniciar um ambiente em sua plataforma personalizada

1. Crie um diretório para seu aplicativo.

```
~$ mkdir custom-platform-app
~$ cd ~/custom-platform-app
```

2. Inicialize um repositório do aplicativo.

```
~/custom-platform-app$ eb init
...
```

3. Faça download do aplicativo de exemplo [NodeSampleApp.zip](#).
4. Extraia o aplicativo de exemplo.

```
~/custom-platform-app$ unzip ~/NodeSampleApp.zip
```

5. Execute `eb create -p CUSTOM-PLATFORM-ARN`, em que **CUSTOM-PLATFORM-ARN** é o ARN retornado por um comando `eb platform create` para iniciar um ambiente que executa sua Plataforma personalizada.

```
~/custom-platform-app$ eb create -p CUSTOM-PLATFORM-ARN
...
```

Conteúdo do arquivamento de definição da plataforma

Um arquivo de definição de plataforma é a plataforma equivalente a um [pacote de origem de aplicação](#). O arquivo de definição de plataforma é um arquivo ZIP que contém um arquivo de

definição de plataforma, um modelo do Packer e os scripts e arquivos usados pelo modelo do Packer para criar a plataforma.

#### Note

Quando a CLI do EB é utilizada para criar uma plataforma personalizada, ela cria um arquivo de definição de plataforma com base nos arquivos e pastas no repositório de sua plataforma para que você não precise criar o arquivo manualmente.

O arquivo de definição de plataforma é um arquivo formatado em YAML que deve ser denominado `platform.yaml` e estar na raiz do arquivo de definição de plataforma. Consulte [Criar uma plataforma personalizada](#) para obter uma lista das chaves obrigatórias e opcionais compatíveis com um arquivo de definição de plataforma.

Você não precisa fornecer um nome específico ao modelo do Packer, mas o nome do arquivo deve corresponder ao modelo do provisionador especificado no arquivo de definição de plataforma. Consulte a [documentação oficial do Packer](#) para obter instruções sobre a criação de modelos.

Os outros arquivos no arquivo de definição de plataforma são scripts e arquivos usados pelo modelo para personalizar uma instância antes de criar uma AMI.

#### Hooks de plataforma personalizada

O Elastic Beanstalk usa uma estrutura de diretórios padronizada para hooks em plataformas personalizadas. Esses scripts são executados durante eventos de ciclo de vida e em resposta a operações de gerenciamento: quando as instâncias são executadas em seu ambiente ou quando um usuário inicia uma implantação ou usa o recurso de reinicialização do servidor de aplicativos.

Coloque os scripts que você deseja que os hooks acionem em uma das subpastas da pasta `/opt/elasticbeanstalk/hooks/`.

#### Warning

O uso de hooks de plataforma personalizadas não é compatível com plataformas gerenciadas. Os hooks de plataforma personalizada são projetados para plataformas personalizadas. Em plataformas gerenciadas do Elastic Beanstalk, eles podem funcionar de forma diferente ou ter alguns problemas, e o comportamento pode ser diferente entre plataformas. Nas plataformas da AMI do Amazon Linux (anteriores ao Amazon Linux 2), eles ainda podem funcionar de maneiras úteis em alguns casos; use-os com cautela.

Os ganchos de plataforma personalizada são um recurso legado existente em plataformas da AMI do Amazon Linux. Nas plataformas do Amazon Linux 2, os ganchos de plataforma personalizados na pasta `/opt/elasticbeanstalk/hooks/` foram totalmente descontinuados. O Elastic Beanstalk não os lê nem os executa. As plataformas do Amazon Linux 2 são compatíveis com um novo tipo de ganchos de plataforma, projetados especificamente para estender as plataformas gerenciadas do Elastic Beanstalk. É possível adicionar scripts e programas personalizados diretamente a um diretório de ganchos no pacote de origem da aplicação. O Elastic Beanstalk os executa durante vários estágios de provisionamento de instâncias. Para obter mais informações, expanda a seção Platform Hooks (Hooks de plataforma) em [the section called “Estender plataformas Linux”](#).

Os hooks são organizados nas seguintes pastas:

- `appdeploy` – Scripts executados durante uma implantação de aplicativo. O Elastic Beanstalk executa uma implantação de aplicativo quando novas instâncias são executadas e quando um cliente inicia a implantação de uma nova versão.
- `configdeploy`: scripts executados quando um cliente executa uma atualização de configuração que afeta a configuração do software na instância, por exemplo, definição das propriedades do ambiente ou habilitação da alternância de logs no Amazon S3.
- `restartappserver` – Scripts executados quando um cliente executa uma operação de reinicialização do servidor de aplicativos.
- `preinit` – Scripts executados durante o bootstrapping da instância.
- `postinit` – Scripts executados após o bootstrapping da instância.

As pastas `appdeploy`, `configdeploy` e `restartappserver` contêm as subpastas `pre`, `enact` e `post`. Em cada fase de uma operação, todos os scripts na pasta `pre` são executados em ordem alfabética, em seguida os da pasta `enact` e depois os da pasta `post`.

Quando uma instância é iniciada, o Elastic Beanstalk executa `preinit`, `appdeploy` e `postinit`, nessa ordem. Nas implantações subsequentes às instâncias em execução, o Elastic Beanstalk executa hooks `appdeploy`. Os hooks `configdeploy` são executados quando um usuário atualiza as definições de configuração de software da instância. Os hooks `restartappserver` são executados somente quando o usuário inicia uma reinicialização do servidor de aplicativos.

Quando os scripts encontram erros, eles podem sair com um status diferente de zero e gravar no arquivo `stderr` para falhar a operação. A mensagem que você gravar no `stderr` é exibida

no evento que é gerado quando a operação falha. O Elastic Beanstalk também captura essas informações no arquivo de log `/var/log/eb-activity.log`. Se você não deseja que a operação falhe, retorne 0. As mensagens que você gravar no `stderr` ou no `stdout` aparecerão nos [logs de implantação](#), mas não aparecerão no stream de eventos, a menos que a operação falhe.

## Limpeza da instância do Packer

Em determinadas circunstâncias, como interrupção do processo do construtor Packer antes de ser concluído, as instâncias executadas pelo Packer não serão limpas. Essas instâncias não fazem parte do ambiente do Elastic Beanstalk e podem ser visualizadas e encerradas somente usando o serviço Amazon EC2.

Para limpar manualmente essas instâncias

1. Abra o [console do Amazon EC2](#).
2. Verifique se você está na mesma região da AWS na qual criou a instância com o Packer.
3. Em Resources (Recursos), escolha **N** Running Instances (N instâncias em execução), em que **N** indica o número de instâncias em execução.
4. Clique na caixa de texto de consulta.
5. Selecione a tag Name (Nome).
6. Digite packer.

A consulta deve ter a seguinte aparência: `tag:Name: packer (tag:Nome: packer)`

7. Selecione quaisquer instâncias correspondentes à consulta.
8. Se Instance State (Estado da instância) for running (em execução), escolha Actions (Ações), Instance State (Estado da instância), Stop (Parar) e, em seguida, Actions (Ações), Instance State (Estado da instância) e Terminate (Encerrar).

## Formato do arquivo platform.yaml

O arquivo `platform.yaml` tem o seguinte formato.

```
version: "version-number"

provisioner:
 type: provisioner-type
 template: provisioner-template
 flavor: provisioner-flavor
```

```
metadata:
 maintainer: metadata-maintainer
 description: metadata-description
 operating_system_name: metadata-operating_system_name
 operating_system_version: metadata-operating_system_version
 programming_language_name: metadata-programming_language_name
 programming_language_version: metadata-programming_language_version
 framework_name: metadata-framework_name
 framework_version: metadata-framework_version

option_definitions:
 - namespace: option-def-namespace
 option_name: option-def-option_name
 description: option-def-description
 default_value: option-def-default_value

option_settings:
 - namespace: "option-setting-namespace"
 option_name: "option-setting-option_name"
 value: "option-setting-value"
```

Substitua os espaços reservados por esses valores:

### *version-number*

Obrigatório. A versão da definição de YAML. Deve ser **1.0**.

### *provisioner-type*

Obrigatório. O tipo de construtor usado para criar a plataforma personalizada. Deve ser **packer**.

### *provisioner-template*

Obrigatório. O arquivo JSON com as configurações de *provisioner-type*.

### *provisioner-flavor*

Opcional. O sistema operacional básico usado para a AMI. Um dos seguintes:

amazon (padrão)

Amazon Linux Se não for especificado, a versão mais recente do Amazon Linux quando a plataforma é criada.

O Amazon Linux 2 não é um tipo de sistema operacional compatível.

ubuntu1604

Ubuntu 16.04 LTS

rhel7

RHEL 7

rhel6

RHEL 6

### *metadata-maintainer*

Opcional. As informações de contato da pessoa que possui a plataforma (100 caracteres).

### *metadata-description*

Opcional. Descrição da plataforma (2.000 caracteres).

### *metadata-operating\_system\_name*

Opcional. Nome do sistema operacional da plataforma (50 caracteres). Esse valor está disponível na filtragem da saída para a API [ListPlatformVersions](#).

### *metadata-operating\_system\_version*

Opcional. Versão do sistema operacional da plataforma (20 caracteres).

### *metadata-programming\_language\_name*

Opcional. Linguagem de programação compatível com a plataforma (50 caracteres)

### *metadata-programming\_language\_version*

Opcional. Versão da linguagem da plataforma (20 caracteres).

### *metadata-framework\_name*

Opcional. Nome do framework da Web usado pela plataforma (50 caracteres).

### *metadata-framework\_version*

Opcional. Versão do framework da Web da plataforma (20 caracteres).

### *option-def-namespace*

Opcional. Um namespace sob `aws:elasticbeanstalk:container:custom` (100 caracteres).

### *option-def-option\_name*

Opcional. O nome da opção (100 caracteres). Você pode definir até 50 opções de configuração personalizadas que a plataforma fornece aos usuários.

### *option-def-description*

Opcional. Descrição da opção (1.024 caracteres).

### *option-def-default\_value*

Opcional. O valor padrão usado quando o usuário não especificar um.

O exemplo a seguir cria a opção **NPM\_START**.

```
options_definitions:
- namespace: "aws:elasticbeanstalk:container:custom:application"
 option_name: "NPM_START"
 description: "Default application startup command"
 default_value: "node application.js"
```

### *option-setting-namespace*

Opcional. Namespace da opção.

### *option-setting-option\_name*

Opcional. Nome da opção. Você pode especificar até 50 [opções fornecidas pelo Elastic Beanstalk](#).

### *option-setting-value*

Opcional. Valor padrão usado quando o usuário não especifica um.

O exemplo a seguir cria a opção **TEST**.

```
option_settings:
- namespace: "aws:elasticbeanstalk:application:environment"
 option_name: "TEST"
 value: "This is a test"
```

## Atribuir tags em versões de plataforma personalizada

Você pode aplicar tags às versões da plataforma personalizada do AWS Elastic Beanstalk. As tags são pares de chave-valor associados a recursos da AWS. Para obter informações sobre a atribuição

de tags do recurso do Elastic Beanstalk, casos de uso, restrições de chave e valor de tag, além de tipos de recursos compatíveis, consulte [Marcar recursos da aplicação do Elastic Beanstalk](#).

Você pode especificar tags quando criar uma versão da plataforma personalizada. Em uma versão da plataforma personalizada existente, você pode adicionar ou remover tags e atualizar os valores de tags existentes. Você pode adicionar até 50 tags para cada versão da plataforma personalizada.

Adicionar tags durante a criação da versão da plataforma personalizada

Se você usar a CLI do EB para criar a versão da plataforma personalizada, use a opção `--tags` com [eb platform create](#) para adicionar tags.

```
~/workspace/my-app$ eb platform create --tags mytag1=value1,mytag2=value2
```

Com a AWS CLI ou outros clientes baseados em API, adicione tags usando o parâmetro `--tags` no comando [create-platform-version](#).

```
$ aws elasticbeanstalk create-platform-version \
 --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
 --platform-name my-platform --platform-version 1.0.0 --platform-definition-bundle
 S3Bucket=DOC-EXAMPLE-BUCKET,S3Key=sample.zip
```

Gerenciar tags de uma versão da plataforma personalizada existente

É possível adicionar, atualizar e excluir tags em uma versão da plataforma personalizada existente do Elastic Beanstalk.

Se você usar a CLI do EB para atualizar a versão da plataforma personalizada, use [eb tags](#) para adicionar, atualizar, excluir ou listar tags.

Por exemplo, o comando a seguir lista as tags em uma versão da plataforma personalizada.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-
account-id:platform/my-platform/1.0.0"
```

O comando a seguir atualiza a tag `mytag1` e exclui a tag `mytag2`.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
 --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-
platform/1.0.0"
```



Para obter uma lista de opções e mais exemplos, consulte [eb tags](#).

Com a AWS CLI ou outros clientes baseados em API, use o comando [list-tags-for-resource](#) para listar as tags de uma versão de plataforma personalizada.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
"arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-platform/1.0.0"
```

Use o comando [update-tags-for-resource](#) para adicionar, atualizar ou excluir tags em uma versão da plataforma personalizada.

```
$ aws elasticbeanstalk update-tags-for-resource \
--tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
--resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-
platform/1.0.0"
```

Especifique as tags a serem adicionadas e atualizadas no parâmetro `--tags-to-add` do `update-tags-for-resource`. Uma tag não existente é adicionada, e o valor de uma tag existente é atualizado.

#### Note

Para usar alguns dos comandos da EB CLI e da AWS CLI com uma versão de plataforma personalizada do Elastic Beanstalk, você precisa do ARN da versão da plataforma personalizada. Você pode recuperar o ARN usando o comando a seguir.

```
$ aws elasticbeanstalk list-platform-versions
```

Use a opção `--filters` para filtrar a saída até o nome da plataforma personalizada.

## Histórico do documento

A tabela a seguir descreve as mudanças importantes no Guia do AWS Elastic Beanstalk desenvolvedor desde abril de 2024.

Alteração	Descrição	Data
<a href="#"><u>AWSElasticBeanstalkRoleCore AWS política gerenciada</u></a>	Permissões atualizadas na política AWS gerenciada.	30 de abril de 2024
<a href="#"><u>AWSElasticBeanstalkManagedUpdatesServiceRolePolicy AWS política gerenciada</u></a>	Permissões atualizadas na política AWS gerenciada.	30 de abril de 2024
<a href="#"><u>AWSElasticBeanstalkManagedUpdatesInternalServiceRolePolicy AWS política gerenciada</u></a>	Permissões atualizadas na política AWS gerenciada.	30 de abril de 2024
<a href="#"><u>AWSElasticBeanstalkMaintenance AWS política gerenciada</u></a>	Permissões atualizadas na política AWS gerenciada.	30 de abril de 2024
<a href="#"><u>AWSElasticBeanstalkInternalMaintenanceRolePolicy AWS política gerenciada</u></a>	Permissões atualizadas na política AWS gerenciada.	30 de abril de 2024

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.