
AWS Elastic Beanstalk

Entwicklerhandbuch



AWS Elastic Beanstalk: Entwicklerhandbuch

Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. Alle anderen Marken, die nicht Amazon gehören, sind Eigentum ihrer jeweiligen Inhaber, mit denen Amazon eine Partnerschaft, Verbindung oder Sponsorenbeziehung haben könnte oder auch nicht.

Table of Contents

Was ist AWS Elastic Beanstalk?	1
Preise	1
Nächste Schritte	2
Erste Schritte	3
Einrichten: Erstellen eines AWS-Kontos	3
Schritt 1: Erstellen	3
Erstellen einer Anwendung und einer Umgebung	3
Für die Beispielanwendung erstellte AWS-Ressourcen	5
Schritt 2: Erkunden	5
Schritt 3: Bereitstellen einer neuen Version	7
Schritt 4: Konfigurieren	8
Durchführen einer Konfigurationsänderung	9
Überprüfen der Konfigurationsänderung	9
Schritt 5: Bereinigen	10
Nächste Schritte	11
Konzepte	14
Anwendung	14
Anwendungsversion	14
Umgebung	14
Umgebungsschicht	14
Umgebungskonfiguration	14
Gespeicherte Konfiguration	15
Plattform	15
Webserver-Umgebungen	15
Worker-Umgebungen	16
Überlegungen zum Entwurf	17
Skalierbarkeit	17
Sicherheit	18
Persistenter Speicher	18
Fehlertoleranz	19
Bereitstellung von Inhalten	19
Software-Updates und -Patches	19
Konnektivität	20
Berechtigungen	21
Servicerolle	21
Instance-Profil	22
Richtlinie für Benutzer:	24
Plattformen	27
Glossar zu Plattformen	27
Modell der übergreifenden Verantwortlichkeit	29
Plattform-Unterstützungsrichtlinie	30
Zeitplan für die Ausmusterung von Plattformvarianten	31
Ausgemusterte Plattformvarianten	31
Unterstützte Plattformen	32
Unterstützte Plattformversionen	32
Linux-Plattformen	33
Linux-Plattformversionen	33
Liste der Elastic Beanstalk-Linux-Plattformen	34
Erweitern von Linux-Plattformen	34
Arbeiten mit Docker	46
Docker-Plattformfamilie	47
Die Docker-Plattform	48
Multicontainer-Docker (Amazon Linux AMI)	64
Vorkonfigurierte Container	82

Umgebungskonfiguration	85
Lokales Ausführen von Containern	93
Verwenden von Go	97
Erste Schritte	98
Entwicklungsumgebung	100
Die Go-Plattform	101
Tutorial für Go	106
Arbeiten mit Java	109
Erste Schritte	110
Entwicklungsumgebung	115
Die Tomcat-Plattform	117
Die Java SE-Plattform	128
Hinzufügen einer Datenbank	134
Eclipse-Toolkit	140
Ressourcen	155
Arbeiten mit .NET Core unter Linux	155
Erste Schritte	156
Entwicklungsumgebung	158
Die .NET Core-Plattform auf Linux	159
Tutorial - .NET core unter Linux	163
Das AWS Toolkit for Visual Studio	169
Migration von Windows zu Linux	188
Arbeiten mit .NET	189
Erste Schritte	189
Entwicklungsumgebung	192
Die .NET-Plattform	193
Tutorial – ASP.NET MVC5	201
Tutorial – .NET Core	208
Hinzufügen einer Datenbank	217
Das AWS Toolkit for Visual Studio	220
Migrieren von On-Premise-Anwendungen	247
Ressourcen	247
Arbeiten mit Node.js	247
Erste Schritte	248
Entwicklungsumgebung	249
Die Node.js-Plattform	250
Tutorial – Express	260
Tutorial – Express mit Clustering	264
Tutorial – Node.js mit DynamoDB	273
Hinzufügen einer Datenbank	282
Ressourcen	285
Arbeiten mit PHP	285
Entwicklungsumgebung	286
Die PHP-Plattform	288
Tutorial – Laravel	293
Tutorial – CakePHP	301
Tutorial – Symfony	308
Tutorial – HA-Produktion	313
Tutorial – WordPress mit hoher Verfügbarkeit	322
Tutorial – Drupal mit hoher Verfügbarkeit	336
Hinzufügen einer Datenbank	350
Verwenden von Python	352
Entwicklungsumgebung	353
Die Python-Plattform	355
Tutorial – Flask	361
Tutorial – Django	367
Hinzufügen einer Datenbank	378

Ressourcen	380
Arbeiten mit Ruby	381
Entwicklungsumgebung	381
Die Ruby-Plattform	383
Tutorial – Rails	388
Tutorial – Sinatra	394
Hinzufügen einer Datenbank	398
Tutorials und Beispiele	401
Verwalten von Anwendungen	403
Konsole zur Anwendungsverwaltung	405
Verwalten von Anwendungsversionen	406
Lebenszyklus der Version	408
Markieren von Anwendungsversionen	410
Erstellen eines Quell-Bundles	412
Erstellen eines Quell-Bundles mit der Befehlszeile	413
Erstellen eines Quell-Bundles mit Git	413
Komprimieren von Dateien im Mac OS X Finder oder Windows-Explorer	414
Erstellen eines Quell-Bundles für eine .NET-Anwendung	418
Testen des Quell-Bundles	419
Markieren von Ressourcen	420
Ressourcen, die markiert werden können	421
Markieren von Anwendungen	421
Verwalten von Umgebungen	425
Environment Management Console	426
Umgebungsübersicht	427
Umgebungsaktionen	428
Konfiguration	430
Protokolle	431
Integrität	431
Überwachung	432
Alarne	432
Verwaltete Aktualisierungen	433
Events	433
Tags	434
Erstellen von Umgebungen	435
Der Assistent zum Erstellen einer neuen Umgebung	437
Klonen einer Umgebung	456
Beenden einer Umgebung	459
Mit der AWS CLI	460
Mit der API	461
Launch-Now-URL	464
Compose-Umgebungen	469
Bereitstellungen	471
Auswählen einer Bereitstellungsrichtlinie	472
Bereitstellen einer neuen Anwendungsversion	473
Erneute Bereitstellung einer früheren Version	474
Weitere Möglichkeiten zur Bereitstellung Ihrer Anwendung	474
Optionen für die Bereitstellung	474
Blau/Grün-Bereitstellungen	481
Konfigurationsänderungen	483
Fortlaufende Updates	484
Unveränderliche Updates	488
Plattformaktualisierungen	491
Methode 1 – Aktualisieren der Plattformversion Ihrer Umgebung	493
Methode 2 – Ausführen einer Blau/Grün-Bereitstellung	495
Verwaltete Updates	496
Aktualisieren einer Legacy-Umgebung	502

Aktualisieren auf Amazon Linux 2	503
Abbrechen eines Updates	510
Neuerstellung einer Umgebung	511
Neuerstellung einer laufenden Umgebung	511
Neuerstellung einer beendeten Umgebung	512
Umgebungstypen	514
Skalierbare Umgebung mit Lastausgleich	514
Single-Instance-Umgebung	514
Ändern des Umgebungstyps	514
Worker-Umgebungen	516
SQS-Daemon in der Worker-Umgebung	518
Warteschlangen für unzustellbare Nachrichten	519
Regelmäßige Aufgaben	519
Amazon CloudWatch für die automatische Skalierung auf Worker-Umgebungs-Ebene verwenden ..	520
Konfigurieren von Worker-Umgebungen	521
Umgebungslinks	524
Konfigurieren von Umgebungen	526
Konfiguration mithilfe der Konsole	527
Seite "Konfigurationsübersicht"	527
Seite "Änderungen überprüfen"	530
Amazon EC2-Instances	531
Konfigurieren der Amazon EC2-Instances Ihrer Umgebung	532
Der Namespace aws: autoscaling: launchconfiguration	537
IMDS	538
Auto Scaling-Gruppe	540
Spot-Instance-Unterstützung	541
Auto Scaling-Gruppenkonfiguration mit der Elastic Beanstalk-Konsole	543
Auto Scaling-Gruppenkonfiguration mit der EB CLI	546
Konfigurationsoptionen	546
Auslöser	547
Geplante Aktionen	550
Zustandsprüfungseinstellung	554
Load Balancer	554
Classic Load Balancer	556
Application Load Balancer	566
Geteile Application Load Balancer	583
Netzwerk-Load-Balancer	599
Konfigurieren von Zugriffsprotokollen	609
Datenbank	609
Hinzufügen einer RDS-DB-Instance zur Umgebung	610
Verbinden mit der Datenbank	613
Konfigurieren einer integrierten RDS-DB-Instance mithilfe der Konsole	613
Konfigurieren einer integrierten RDS-DB-Instance mithilfe von Konfigurationsdateien	614
Sicherheit	615
Konfigurieren der Umgebungssicherheit	615
Namespaces für die Konfiguration der Umgebungssicherheit	617
Markieren von Umgebungen mit Tags	617
Hinzufügen von Tags beim Erstellen einer Umgebung	618
Verwalten von Tags von bestehenden Umgebungen	619
Softwareeinstellungen	621
Plattformspezifische Einstellungen konfigurieren	622
Konfigurieren von Umgebungseigenschaften	623
Softwareeinstellung Namespaces	625
Zugreifen auf Umgebungseigenschaften	626
Debugging	627
Anzeigen von Protokollen	630
Benachrichtigungen	633

Benachrichtigungen mit der Elastic Beanstalk-Konsole konfigurieren	634
Konfigurieren von Benachrichtigungen mithilfe von Konfigurationsoptionen	635
Konfigurieren von Berechtigungen zum Senden von Benachrichtigungen	636
Amazon VPC	637
VPC-Einstellungen in der Elastic Beanstalk-Konsole konfigurieren	638
Der aws:ec2:vpc-Namespace	640
Migration aus EC2-Classic zu einer VPC	641
Domänenname	644
Konfigurieren von Umgebungen (Erweitert)	646
Konfigurationsoptionen	646
Precedence	647
Empfohlene Werte	648
Vor der Umgebungserstellung	649
Während der Erstellung	653
Nach der Erstellung	658
Allgemeine Optionen	666
Plattformspezifische Optionen	712
Benutzerdefinierte Optionen	721
.Ebextensions	722
Optionseinstellungen	724
Linux-Server	725
Windows-Server	738
Angepasste Ressourcen	745
Gespeicherte Konfigurationen	764
Markieren von gespeicherten Konfigurationen	769
env.yaml	771
Benutzerdefiniertes Image	773
Erstellen eines benutzerdefinierten AMI	773
Bereinigen eines benutzerdefinierten AMI	776
Statische Dateien	776
Konfigurieren statischer Dateien über die Konsole	776
Konfigurieren statischer Dateien mithilfe von Konfigurationsoptionen	777
HTTPS	778
Erstellen eines Zertifikats	780
Hochladen eines Zertifikats	782
Beenden mit dem Load Balancer	783
Beenden auf der Instance	785
End-to-End-Verschlüsselung	810
TCP-Pass-Through	813
Sicheres Speichern von Schlüsseln	814
Umleitung von HTTP zu HTTPS	815
Überwachen einer Umgebung	817
Überwachungskonsole	817
Übersicht	818
Überwachungsdiagramme	818
Anpassen der Überwachungskonsole	819
Grundlegende Zustandsberichte	821
Zustandsfarben	822
Elastic Load Balancing-Zustandsprüfungen	822
Zustandsprüfungen für Umgebungen mit einer einzelnen Instance oder Worker-Ebene	823
Zusätzliche Prüfungen	823
Amazon CloudWatch-Metriken	823
Erweiterte Zustandsberichte und Überwachung	824
Der Elastic Beanstalk-Integritäts-Agent	827
Faktoren bei der Bestimmung des Instance- und Umgebungszustands	827
Anpassung der Regel für die Zustandsprüfung	829
Rollen in erweiterten Zustandsberichten	830

Erweiterte Zustandsautorisierung	830
Ereignisse in erweiterten Zustandsberichten	831
Verhalten der erweiterten Zustandsberichte bei Aktualisierungen, Bereitstellungen und Skalierung	831
Aktivieren der erweiterten Zustandsberichte	832
Konsole für die Integritätsüberwachung	835
Farben und Status in Zustandsangaben	841
Instance-Metriken	843
Regeln für den erweiterten Zustand	845
CloudWatch	849
API-Benutzer	855
Format der Protokolle der erweiterten Zustandsberichte	856
Benachrichtigungen und Fehlerbehebung	859
Verwalten von Alarmen	861
Anzeigen von Ereignissen	864
Überwachen von Instances	866
Anzeigen von Instance-Protokollen	869
Speicherort der Protokolle auf Amazon EC2-Instances	870
Speicherort der Protokolle in Amazon S3	871
Protokollrotations-Einstellungen auf Linux	872
Erweitern der Standardkonfiguration für Protokollaufgaben	872
Protokolldateien an Amazon CloudWatch Logs streamen	874
Integrieren von AWS-Services	876
Übersicht über die Architektur	876
CloudFront	877
CloudTrail	878
Elastic Beanstalk-Informationen in CloudTrail	878
Grundlegendes zu den Elastic Beanstalk-Einträgen in der Protokolldatei	879
CloudWatch	879
CloudWatch-Protokolle	880
Voraussetzungen für das Instance-Protokoll-Streaming an CloudWatch Logs	882
Wie Elastic Beanstalk CloudWatch Logs einrichtet	883
Streamen von Instance-Protokollen an CloudWatch Logs	886
Fehlerbehebung bei der CloudWatch Logs-Integration	888
Streamen des Umgebungszustands	888
EventBridge	891
Überwachen des Ressourcenstatus mit EventBridge	891
Beispiel für Elastic Beanstalk-Ereignisse	893
Elastic Beanstalk-Ereignisfeldzuordnung	894
AWS Config	895
Einrichten von AWS Config	896
Konfigurieren von AWS Config zur Aufzeichnung von Elastic Beanstalk-Ressourcen	896
Anzeigen von Elastic Beanstalk-Konfigurationsdetails in der AWS Config-Konsole	897
Evaluierung von Elastic Beanstalk-Ressourcen mit AWS Config-Regeln	900
DynamoDB	900
ElastiCache	901
Amazon EFS	901
Konfigurationsdateien	902
Verschlüsselte Dateisysteme	902
Beispielanwendungen	902
Bereinigen von Dateisystemen	903
IAM	903
Instance-Profile	904
Servicerollen	909
Verwenden von serviceverknüpften Rollen	915
Benutzerrichtlinien	926
ARN-Format	933
Ressourcen und Bedingungen	934

Tag-basierte Zugriffskontrolle	960
Beispiel für verwaltete Richtlinien	963
Beispiel für ressourcenspezifische Richtlinien	966
Amazon RDS	973
Amazon RDS in Standard-VPC	974
Amazon RDS in EC2 Classic	979
Verbindungszeichenfolge in Amazon S3	983
Bereinigen einer externen Amazon RDS-Instance	985
Amazon S3	985
Inhalt des Elastic Beanstalk Amazon S3-Buckets	985
Löschen von Objekten im Elastic Beanstalk Amazon S3-Bucket	986
Löschen des Elastic Beanstalk Amazon S3-Buckets	986
Amazon VPC	988
Öffentliche VPC	989
Öffentliche/private VPC	990
Private VPC	990
Bastion-Hosts	992
Amazon RDS	997
VPC-Endpunkte	1002
Konfigurieren des Entwicklungscomputers	1005
Erstellen eines Projektordners	1005
Einrichten der Quellsteuerung	1005
Konfigurieren eines Remote-Repositorys	1006
Installieren der EB CLI	1006
Installieren der AWS-CLI	1007
EB CLI	1008
Installieren der EB CLI	1009
Installieren der EB-CLI mit Setup-Skripts	1009
Manuelle Installation	1009
Konfigurieren der EB CLI	1017
Ignorieren von Dateien, die <code>.ebignore</code> verwenden	1018
Verwenden von benannten Profilen	1019
Bereitstellen eines Artefakts anstelle des Projektordners	1019
Konfigurationseinstellungen und Vorrang	1019
Instance-Metadaten	1020
EB-CLI – Grundlagen	1020
Eb create	1021
Eb status	1021
Eb health	1022
Eb events	1022
Eb logs	1023
Eb open	1023
Eb deploy	1023
Eb config	1024
Eb terminate	1024
CodeBuild	1025
Erstellen einer Anwendung	1025
Erstellen und Bereitstellen Ihres Anwendungscodes	1025
Verwenden der EB CLI mit Git	1027
Zuordnen von Git-Verzweigungen zu Elastic Beanstalk-Umgebungen	1027
Bereitstellen von Änderungen	1027
Verwenden von Git-Submodulen	1028
Zuweisen von Git-Tags zu Ihrer Anwendungsversion	1028
CodeCommit	1028
Voraussetzungen	1029
Erstellen eines CodeCommit-Repositorys mit der EB CLI	1029
Bereitstellen von Ihrem CodeCommit-Repository aus	1030

Konfigurieren von zusätzlichen Verzweigungen und Umgebungen	1031
Verwenden eines vorhandenen CodeCommit-Repositorys	1032
Überwachen des Zustands	1033
Lesen der Ausgabe	1035
Interaktive Ansicht des Zustands	1037
Optionen für die Interaktive Ansicht des Zustands	1038
Zusammenstellen von Umgebungen	1038
Fehlersuche	1040
Beheben von Bereitstellungsfehlern	1041
EB-CLI-Befehle	1042
eb abort	1043
eb appversion	1044
eb clone	1047
eb codesource	1049
eb config	1050
eb console	1052
eb create	1053
eb deploy	1063
eb events	1064
eb health	1066
eb init	1067
eb labs	1070
eb list	1070
eb local	1071
eb logs	1074
eb open	1076
eb platform	1077
eb printenv	1084
eb restore	1085
eb scale	1086
eb setenv	1086
eb ssh	1087
eb status	1089
eb swap	1091
eb tags	1092
eb terminate	1094
eb upgrade	1096
eb use	1097
Allgemeine Optionen	1097
EB-CLI 2.6 (veraltet)	1098
Unterschiede zu Version 3 von EB-CLI	1098
Zu EB CLI 3 und CodeCommit migrieren	1098
EB API-CLI (veraltet)	1099
Konvertieren von Elastic Beanstalk API CLI-Skripten	1099
Sicherheit	1102
Datenschutz	1102
Datenverschlüsselung	1103
Datenschutz zwischen Netzwerken	1104
Identitäts- und Zugriffsverwaltung	1104
Protokollierung und Überwachung	1105
Erweiterte Zustandsberichte	1105
Amazon EC2-Instance-Protokolle	1105
Umgebungsbenachrichtigungen	1105
Amazon CloudWatch-Alarme	1105
AWS CloudTrail-Protokolle	1106
AWS X-Ray-Debugging	1106
Compliance-Validierung	1106

Ausfallsicherheit	1107
Sicherheit der Infrastruktur	1107
Modell der übergreifenden Verantwortlichkeit	1107
Bewährte Methoden für die Gewährleistung der Sicherheit	1108
Bewährte Methoden für vorbeugende Sicherheitsmaßnahmen	1108
Bewährte Methoden für aufdeckende Sicherheitsmaßnahmen	1109
Fehlersuche	1110
Konnektivität	1110
Umgebungserstellung	1111
Bereitstellungen	1111
Integrität	1112
Konfiguration	1112
Docker	1112
Häufig gestellte Fragen	1113
Ressourcen	1115
Beispielanwendungen	1115
Plattformverlauf	1117
Benutzerdefinierte Plattformen	1117
Erstellen einer benutzerdefinierten Plattform	1118
Verwenden eines benutzerdefinierten Plattformbeispiels	1118
Inhalte des Plattformdefinitionsarchivs	1123
Benutzerdefinierte Plattform-Hooks	1123
Bereinigen von Instances durch Packer	1124
Platform.yaml-Format	1125
Markieren von benutzerdefinierten Plattformversionen	1127

Was ist AWS Elastic Beanstalk?

Amazon Web Services (AWS) umfasst mehr als hundert Services mit jeweils einem eigenen Funktionsbereich. Dieses breite Servicespektrum bietet Ihnen die Flexibilität für eine individuelle Verwaltung Ihrer AWS-Infrastruktur, die Auswahl der entsprechenden Services und deren Bereitstellung kann jedoch gelegentlich etwas unübersichtlich sein.

AWS Elastic Beanstalk ermöglicht eine schnelle Bereitstellung und Verwaltung von Anwendungen in der AWS Cloud, ohne aufwendiges Erstellen einer Infrastruktur, in der diese Anwendungen ausgeführt werden. AWS Elastic Beanstalk vereinfacht die komplexe Verwaltung, ohne Einschränkungen in Bezug auf Auswahl oder Kontrolle nach sich zu ziehen. Sie laden Ihre Anwendung einfach hoch, und AWS Elastic Beanstalk übernimmt automatisch Kapazitätsbereitstellung, Lastverteilung, Skalierung und Überwachung des Anwendungsstatus.

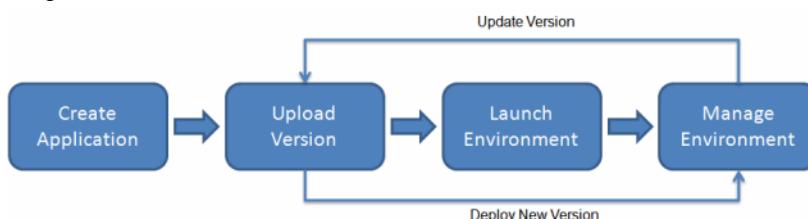
Elastic Beanstalk unterstützt Anwendungen, die in Go, Java, .NET, Node.js, PHP, Python und Ruby entwickelt wurden. Wenn Sie Ihre Anwendung bereitstellen, erstellt Elastic Beanstalk die ausgewählte unterstützte Plattformversion und stellt eine oder mehrere AWS-Ressourcen, z. B. Amazon EC2-Instances, zur Ausführung Ihrer Anwendung bereit.

Sie können mit Elastic Beanstalk interagieren, indem Sie die Elastic Beanstalk-Konsole, die AWS-Befehlszeilschnittstelle (AWS CLI) oder eb, eine High-Level-CLI, die speziell für Elastic Beanstalk entwickelt wurde, verwenden.

Weitere Informationen über das Bereitstellen einer Beispiel-Webanwendung mit Elastic Beanstalk finden Sie unter [Erste Schritte mit AWS: Bereitstellen einer Web-Anwendung](#).

Direkt über die Webschnittstelle (Konsole) von Elastic Beanstalk lassen sich auch die meisten Bereitstellungsaufgaben ausführen, wie z. B. die Anzahl der Amazon EC2-Instances ändern oder die Anwendung überwachen.

Zum Verwenden von Elastic Beanstalk erstellen Sie eine Anwendung, laden eine Anwendungsversion in Form eines Anwendungs-Quell-Bundles (z. B. eine Java-WAR-Datei) in Elastic Beanstalk hoch und stellen einige Informationen zur Anwendung bereit. Elastic Beanstalk startet automatisch eine Umgebung und erstellt und konfiguriert die zur Codeausführung benötigten AWS-Ressourcen. Nach dem Start der Umgebung können Sie diese verwalten und neue Anwendungsversionen bereitstellen. Das folgende Diagramm veranschaulicht den Arbeitsablauf von Elastic Beanstalk.



Nachdem Sie die Anwendung erstellt und bereitgestellt haben, können Sie über die Elastic Beanstalk-Konsole, APIs oder CLIs (darunter auch die einheitliche AWS CLI) Informationen zur Anwendung wie Metriken, Ereignisse und Umgebungsstatus aufrufen.

Preise

Für die Nutzung von Elastic Beanstalk fallen keine zusätzlichen Gebühren an. Sie zahlen nur für die zugrunde liegenden AWS-Ressourcen, die von der Anwendung benötigt werden. Einzelheiten zur Preisgestaltung finden Sie unter [Elastic Beanstalk Service-Detailseite](#).

Nächste Schritte

Dieses Handbuch bietet konzeptionelle Informationen zum Elastic Beanstalk-Webservice und enthält eine Anleitung, wie Sie mit dem Service Webanwendungen bereitstellen. In separaten Abschnitten wird die Verwendung der Elastic Beanstalk-Konsole, der Befehlszeilenschnittstellen-Tools (CLI) und der API zur Bereitstellung und Verwaltung Ihrer Elastic Beanstalk-Umgebungen beschrieben. Zudem dokumentiert dieses Handbuch die Integration von Elastic Beanstalk mit anderen Services von Amazon Web Services.

Es wird empfohlen, zum Kennenlernen von Elastic Beanstalk das Thema [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#) zu lesen. Erste Schritte führt Sie durch die Erstellung, Anzeige und Aktualisierung Ihrer Elastic Beanstalk-Anwendung sowie durch die Bearbeitung und Terminierung Ihrer Elastic Beanstalk-Umgebung. In Erste Schritte werden außerdem die verschiedenen Möglichkeiten des Zugriffs auf Elastic Beanstalk vorgestellt.

Weitere Informationen zur Elastic Beanstalk-Anwendung und deren Komponenten finden Sie auf den folgenden Seiten.

- [Elastic Beanstalk-Konzepte \(p. 14\)](#)
- [Elastic Beanstalk-Plattformen – Glossar \(p. 27\)](#)
- [Modell der übergreifenden Verantwortlichkeit für die Wartung der Elastic Beanstalk-Plattform \(p. 29\)](#)
- [Support-Richtlinie für die Elastic Beanstalk-Plattform \(p. 30\)](#)

Erste Schritte mit Elastic Beanstalk

Um Ihnen zu helfen, die Funktionsweise von AWS Elastic Beanstalk zu verstehen, führt dieses Tutorial Sie durch das Erstellen, Erkunden, Aktualisieren und Löschen einer Elastic Beanstalk-Anwendung. Die Fertigstellung dauert weniger als eine Stunde.

Für die Verwendung von Elastic Beanstalk fallen keine Kosten an, aber die AWS-Ressourcen, die es für dieses Tutorial erstellt, sind live (und werden nicht in einer Sandbox ausgeführt). Es fallen die standardmäßigen Nutzungsgebühren für diese Ressourcen an, bis sie von Ihnen am Ende dieses Tutorials beendet werden. Die Gesamtgebühren liegen in der Regel unter einem Dollar. Weitere Informationen zum Minimieren von Gebühren finden Sie unter [Kostenloses Kontingent für AWS](#).

Themen

- [Einrichten: Erstellen eines AWS-Kontos \(p. 3\)](#)
- [Schritt 1: Erstellen einer Beispielanwendung \(p. 3\)](#)
- [Schritt 2: Erkunden der Umgebung \(p. 5\)](#)
- [Schritt 3: Bereitstellen einer neuen Version der Anwendung \(p. 7\)](#)
- [Schritt 4: Konfigurieren der Umgebung \(p. 8\)](#)
- [Schritt 5: Bereinigen \(p. 10\)](#)
- [Nächste Schritte \(p. 11\)](#)

Einrichten: Erstellen eines AWS-Kontos

Wenn Sie noch kein AWS-Kunde sind, müssen Sie ein AWS-Konto erstellen. Durch die Anmeldung erhalten Sie Zugriff auf Elastic Beanstalk und andere benötigte AWS-Services.

Registrieren Sie sich für ein AWS-Konto wie folgt:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Folgen Sie den angezeigten Anweisungen.

Schritt 1: Erstellen einer Beispielanwendung

In diesem Schritt erstellen Sie eine neue Anwendung, die auf einer bereits bestehenden Beispielanwendung basiert. Elastic Beanstalk unterstützt Plattformen für verschiedene Programmiersprachen, Anwendungsserver und Docker-Container. Sie wählen eine Plattform aus, wenn Sie die Anwendung erstellen.

Erstellen einer Anwendung und einer Umgebung

Um Ihre Beispielanwendung zu erstellen, verwenden Sie den Konsolenassistenten Create a web app (Eine Webanwendung erstellen). Er erstellt eine Elastic Beanstalk-Anwendung und startet eine Umgebung darin. Eine Umgebung ist die Sammlung von AWS-Ressourcen, die für die Ausführung Ihres Anwendungscodes erforderlich sind.

So erstellen Sie eine Beispielanwendung

1. Öffnen Sie die Elastic Beanstalk-Konsole über diesen Link: <https://console.aws.amazon.com/elasticbeanstalk/home#/gettingStarted?applicationName=getting-started-app>
2. Fügen Sie optional [Anwendungs-Tags \(p. 421\)](#) hinzu.

3. Wählen Sie für Platform (Plattform) eine Plattform aus und klicken Sie dann auf Create application (Anwendung erstellen).

Zum Ausführen einer Beispielanwendung mit AWS-Ressourcen führt Elastic Beanstalk die folgenden Aktionen aus. Die Fertigstellung dauert etwa fünf Minuten.

1. Erstellt eine Elastic Beanstalk-Anwendung namens getting-started-app.
2. Eine Umgebung mit dem Namen GettingStartedApp-env mit diesen AWS-Ressourcen wird gestartet:
 - Eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance (virtuelle Maschine)
 - Eine Amazon EC2-Sicherheitsgruppe
 - Ein Amazon Simple Storage Service (Amazon S3)-Bucket
 - Amazon CloudWatch-Alarne
 - Ein AWS CloudFormation-Stack
 - Ein Domänenname

Weitere Informationen zu diesen AWS-Ressourcen finden Sie unter [the section called “Für die Beispielanwendung erstellte AWS-Ressourcen” \(p. 5\)](#).

3. Erstellt eine neue Anwendungsversion mit dem Namen Beispielanwendung. Dies ist die standardmäßige Elastic Beanstalk-Beispielanwendungsdatei.
4. Stellt den Code für die Beispielanwendung in der GettingStartedApp-env--Umgebung bereit.

Während der Umgebungserstellung werden von der Konsole Fortschritte erfasst und Ereignisse angezeigt.



Creating GettingStarted-env

This will take a few minutes....

```
8:40pm Successfully launched environment: GettingStarted-env
8:39pm Environment health has transitioned from Pending to Ok. Initialization completed 16 seconds ago and took 5 m
8:36pm Added instance [i-045eb69a24818d1d4] to your environment.
8:36pm Waiting for EC2 instances to launch. This may take a few minutes.
8:35pm Created EIP: 34.230.236.246
8:34pm Created security group named:
eb-dv-e-sbj4gzb2dm-stack-AWSEBSecurityGroup-KATGTR06V1J9
8:34pm Environment health has transitioned to Pending. Initialization in progress (running for 8 seconds). There are no
8:34pm Using elasticbeanstalk-us-east-1-270205402845 as Amazon S3 storage bucket for environment data.
8:34pm createEnvironment is starting.
```

Wenn alle Ressourcen gestartet werden und die EC2-Instances, auf denen die Anwendung ausgeführt wird, die Zustandsprüfungen bestehen, ändert sich der Zustand der Umgebung in **Ok**. Sie können jetzt die Website Ihrer Webanwendung verwenden.

Für die Beispielanwendung erstellte AWS-Ressourcen

Wenn Sie die Beispielanwendung erstellen, erstellt Elastic Beanstalk die folgenden AWS-Ressourcen:

- EC2 instance (EC2-Instance) – Eine virtuelle Amazon EC2-Maschine zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

Jede Plattform führt eine andere Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Framework, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder nginx als Reverse-Proxy, der Web-Datenverkehr vor Ihrer Web-App verarbeitet, Anforderungen an sie weiterleitet, statische Komponenten verarbeitet und Zugriffs- und Fehlerprotokolle generiert.
- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 zugelassen wird. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – Zwei CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Beim Auslösen eines Alarms wird Ihre Auto Scaling-Gruppe nach oben oder unten skaliert.
- AWS CloudFormation Stack – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domänenname – Ein Domänenname, der auf Ihre Webanwendung in der Form [subdomain.region.elasticbeanstalk.com](#) weiterleitet.

Schritt 2: Erkunden der Umgebung

Um einen Überblick über die Umgebung Ihrer Elastic Beanstalk-Anwendung zu erhalten, verwenden Sie die Umgebungsseite in der Elastic Beanstalk-Konsole.

So zeigen Sie die Umgebungsübersicht an

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

Im Umgebungsübersichtsbereich werden Übersichtsinformationen zu Ihrer Umgebung angezeigt. Dazu zählen der Name, die URL, der aktuelle Integritätsstatus, der Name der zurzeit bereitgestellten Anwendungsversion und die Plattformversion, auf der die Anwendung ausgeführt wird. Unterhalb des Übersichtsbereichs werden Ihnen die letzten fünf Ereignisse in der Umgebung angezeigt.

Weitere Informationen zu Umgebungsebenen, Plattformen, Anwendungsversionen und anderen Elastic Beanstalk-Konzepten finden Sie unter [Konzepte \(p. 14\)](#).

The screenshot shows the AWS Elastic Beanstalk console interface. On the left, a sidebar lists environments and applications. Under 'environments', 'getting-started-app' and 'GettingStartedApp-env' are listed. 'GettingStartedApp-env' is expanded, showing options like 'Go to environment', 'Configuration', 'Logs', 'Health', 'Monitoring', 'Alarms', 'Managed updates', 'Events', and 'Tags'. Below this is a section for 'Recent environments' with 'GettingStartedApp-env' highlighted. The main content area shows the 'GettingStartedApp-env' environment details. It includes a 'Health' section with a green checkmark icon and the status 'Ok'. A 'Running version' section shows 'Sample Application' and a 'Upload and deploy' button. At the bottom, there's a 'Recent events' table with the following data:

Time	Type	Details
2020-01-28 12:11:17 UTC-0800	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 9 seconds
2020-01-28 12:11:15 UTC-0800	INFO	Successfully launched environment: GettingStartedApp-env
2020-01-28 12:11:13 UTC-0800	INFO	Application available at GettingStartedApp-env.bx7dx222kw.us-east-2.elasticbeanstalk.com
2020-01-28 12:10:32 UTC-0800	INFO	Created CloudWatch alarm named: aiseb-e-cubmdjm6ga-stack-AWSEBCloudwatchAlarmHigh
2020-01-28 12:10:32 UTC-0800	INFO	Created CloudWatch alarm named: aiseb-e-cubmdjm6ga-stack-AWSEBCloudwatchAlarmLow

Während Elastic Beanstalk die AWS-Ressourcen erstellt und die Anwendung startet, befindet sich die Umgebung im Pending-Zustand. Statusmeldungen zu Startereignissen werden der Übersicht kontinuierlich hinzugefügt.

Die URL der Umgebung befindet sich oben in der Übersicht, unter dem Umgebungsnamen. Dies ist die URL der Webanwendung, die die Umgebung ausführt. Wählen Sie diese URL, um zur Seite Congratulations (Herzlichen Glückwunsch) der Beispielanwendung zu gelangen.

Die Navigationsseite auf der linken Seite der Konsole verweist auf andere Seiten, die detailliertere Informationen zu Ihrer Umgebung sowie Zugriff auf weitere Funktionen bieten:

- Configuration (Konfiguration) – Zeigt die Ressourcen an, die für diese Umgebung bereitgestellt wurden (z. B. die Amazon Elastic Compute Cloud [Amazon EC2]-Instances, die Ihre Anwendung hosten). Sie können einige der bereitgestellten Ressourcen auf dieser Seite konfigurieren.
- Health (Zustand) – Zeigt den Status und detaillierte Zustandsinformationen zu den Amazon EC2-Instances an, auf denen die Anwendung ausgeführt wird.
- Monitoring (Überwachung) – Zeigt Statistikwerte zur Umgebung an (z. B. CPU-Auslastung und durchschnittliche Latenz). Sie können auf dieser Seite Alarne für die Metriken erstellen, die Sie überwachen.
- Events (Ereignisse) – Zeigt Informationen oder Fehlermeldungen vom Elastic Beanstalk-Service und von anderen Services an, deren Ressourcen diese Umgebung verwendet.
- Tags – Zeigt Umgebungs-Tags an und ermöglicht Ihnen, sie zu verwalten. Tags sind Schlüssel-Wert-Paare, die auf Ihre Umgebung angewendet werden.

Schritt 3: Bereitstellen einer neuen Version der Anwendung

In regelmäßigen Abständen müssen Sie möglicherweise eine neue Version Ihrer Anwendung bereitstellen. Sie können jederzeit eine neue Version bereitstellen, sofern nicht andere Aktualisierungsvorgänge in der Umgebung ausgeführt werden.

Die Anwendungsversion, mit der Sie dieses Tutorial gestartet haben, heißt Beispielanwendung.

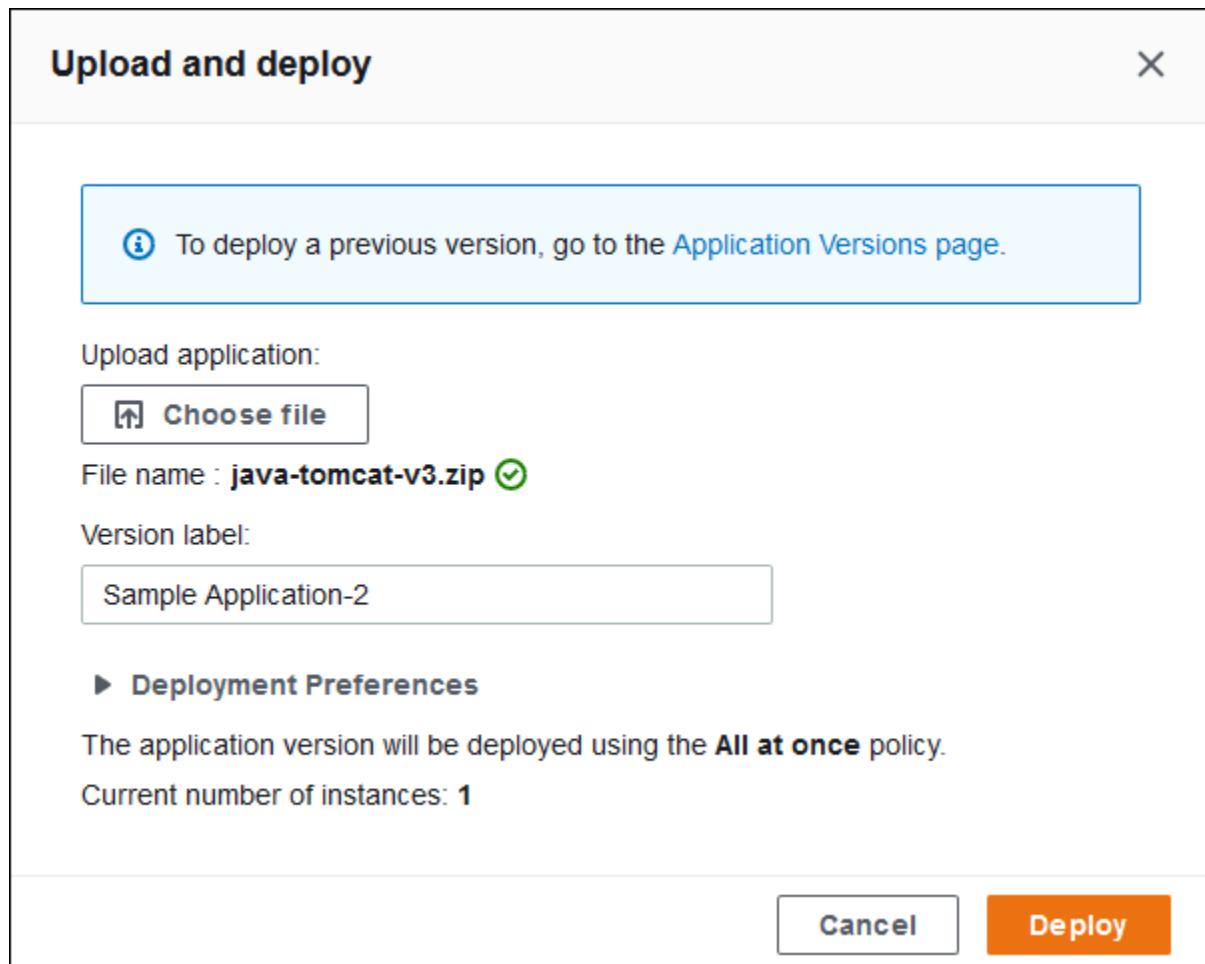
So aktualisieren Sie die Anwendungsversion

1. Laden Sie die Beispielanwendung herunter, die mit der Plattform Ihrer Umgebung übereinstimmt. Verwenden Sie eine der folgenden Anwendungen.
 - Docker – [docker.zip](#)
 - Multicontainer Docker – [docker-multicontainer-v2.zip](#)
 - Vorkonfigurierter Docker (Glassfish) – [docker-glassfish-v1.zip](#)
 - Go – [go.zip](#)
 - Corretto – [corretto.zip](#)
 - Tomcat – [tomcat.zip](#)
 - .NET Core unter Linux – [dotnet-core-linux.zip](#)
 - .NET – [dotnet-asp-v1.zip](#)
 - Node.js – [nodejs.zip](#)
 - PHP – [php.zip](#)
 - Python – [python.zip](#)
 - Ruby – [ruby.zip](#)
2. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
3. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

4. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
5. Wählen Sie Choose File (Datei auswählen) und laden Sie dann das Quell-Bundle der Beispielanwendung hoch, das Sie heruntergeladen haben.



Die Konsole füllt die Versionsbezeichnung automatisch mit einer neuen eindeutigen Bezeichnung aus. Wenn Sie Ihre eigene Versionsbezeichnung eingeben, stellen Sie sicher, dass sie eindeutig ist.

6. Wählen Sie Deploy (Bereitstellen) aus.

Während Elastic Beanstalk Ihre Datei auf Ihren Amazon EC2-Instances bereitstellt, können Sie den Bereitstellungsstatus in der Umgebungsübersicht anzeigen. Während die Anwendungsversion aktualisiert wird, wird der Environment Health (Umgebungszustand) grau angezeigt. Nach abgeschlossener Bereitstellung führt Elastic Beanstalk eine Zustandsprüfung für die Anwendung durch. Wenn die Anwendung auf die Zustandsprüfung reagiert, wird sie als fehlerfrei betrachtet und der Status ändert sich in grün. Die Umgebungs-Übersicht zeigt die neue Running Version (Ausgeführte Version) an – den Namen, den Sie als Version label (Versionsbezeichnung) angegeben haben.

Elastic Beanstalk lädt auch Ihre neue Anwendungsversion hoch und fügt sie der Tabelle der Anwendungsversionen hinzu. Um die Tabelle anzuzeigen, wählen Sie Application versions (Anwendungsversionen) unter getting-started-app im Navigationsbereich.

Schritt 4: Konfigurieren der Umgebung

Sie können Ihre Umgebung konfigurieren, damit sie besser auf die Anwendung ausgerichtet ist. Beispielsweise können Sie bei einer rechenintensiven Anwendung den Amazon Elastic Compute Cloud

(Amazon EC2)-Instance-Typ für die Anwendungsausführung ändern. Um Konfigurationsänderungen anzuwenden, führt Elastic Beanstalk eine Umgebungsaktualisierung durch.

Einige Konfigurationsänderungen lassen sich einfach und schnell durchführen. Einige Änderungen erfordern das Löschen und Neuerstellen von AWS-Ressourcen. Dies kann einige Minuten dauern. Wenn Sie die Konfigurationseinstellungen ändern, warnt Elastic Beanstalk Sie über potenzielle Anwendungsausfallzeiten.

Durchführen einer Konfigurationsänderung

In diesem Beispiel einer Konfigurationsänderung bearbeiten Sie die Kapazitätseinstellungen Ihrer Umgebung. Sie konfigurieren eine skalierbare Umgebung mit Lastenausgleich, die zwischen zwei und vier Amazon EC2-Instances in ihrer Auto Scaling-Gruppe hat. Dann überprüfen Sie, ob die Änderung durchgeführt wurde. Elastic Beanstalk erstellt eine zusätzliche Amazon EC2-Instance, die der einzelnen Instance hinzugefügt wird, die ursprünglich erstellt wurde. Anschließend ordnet Elastic Beanstalk beide Instances dem Load Balancer der Umgebung zu. Dadurch wird die Reaktionsfähigkeit Ihrer Anwendung verbessert und die Verfügbarkeit erhöht.

So ändern Sie die Kapazität Ihrer Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Ändern Sie im Abschnitt Auto Scaling Group (Auto Scaling-Gruppe) die Option Environment type (Umgebungstyp) in Load balanced (Lastverteilt).
6. Ändern Sie in der Zeile Instances die Optionen Max in **4** und Min in **2**.
7. Wählen Sie Apply (Anwenden) aus.
8. Eine Warnung weist darauf hin, dass dieses Update alle Ihre aktuellen Instances ersetzt. Klicken Sie auf Confirm.
9. Wählen Sie im Navigationsbereich die Option Events.

Das Aktualisieren der Umgebung kann einige Minuten dauern. Um festzustellen, ob es abgeschlossen ist, suchen Sie in der Ereignisliste nach dem Ereignis Successfully deployed new configuration to environment (Neue Konfiguration erfolgreich in der Umgebung bereitgestellt). Damit wird bestätigt, dass die Mindestanzahl der Instances für die Auto Scaling-Funktion auf den Wert 2 festgelegt wurde. Elastic Beanstalk startet automatisch die zweite Instance.

Überprüfen der Konfigurationsänderung

Wenn die Umgebungsaktualisierung abgeschlossen und die Umgebung bereit ist, überprüfen Sie Ihre Änderung.

So überprüfen Sie die erhöhte Kapazität

1. Wählen Sie im Navigationsbereich Health (Zustand) aus.
2. Sehen Sie sich die Seite Enhanced Health Overview (Erweiterte Zustandsberichte – Übersicht) an.

Es werden zwei Amazon EC2-Instances hinter der Zeile Overall (Gesamt) aufgelistet. Ihre Umgebungskapazität wurde auf zwei Instances erhöht.

The screenshot shows the 'Enhanced Health Overview' page in the AWS Elastic Beanstalk console. The navigation path is: Elastic Beanstalk > Environments > GettingStartedApp-env > Health. The page displays the following information:

- Instances:** 2 Total, 2 Ok
- Learn more** about enhanced health.
- Table Headers:** Instance ID, Status ▲, Running ▼, Deployment ID
- Table Data:**
 - Overall: Ok, N/A, N/A
 - i-0867c82b5baab7ef1: Ok, 10 minutes, 1
 - i-0106bcf1fb76efdc4: Ok, 10 minutes, 1

The two individual instances (i-0867c82b5baab7ef1 and i-0106bcf1fb76efdc4) are highlighted with a yellow border.

Instance ID	Status	Running	Deployment ID
Overall	Ok	N/A	N/A
i-0867c82b5baab7ef1	Ok	10 minutes	1
i-0106bcf1fb76efdc4	Ok	10 minutes	1

Schritt 5: Bereinigen

Herzlichen Glückwunsch! Sie haben erfolgreich eine Beispielanwendung in der AWS Cloud bereitgestellt, eine neue Version hochgeladen und die Konfiguration um eine zweite Auto Scaling-Instance erweitert. Um sicherzustellen, dass Ihnen keine Kosten für Services entstehen, die Sie nicht verwenden, löschen Sie alle Anwendungsversionen und beenden Sie die Umgebung. Dadurch werden auch die AWS-Ressourcen gelöscht, die die Umgebung für Sie erstellt hat.

So löschen Sie die Anwendung und alle zugehörigen Ressourcen

1. Alle Anwendungsversionen löschen.
 - a. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
 - b. Wählen Sie im Navigationsbereich Applications (Anwendungen) und anschließend getting-started-app.
 - c. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Application versions (Anwendungsversionen) aus.
 - d. Wählen Sie auf der Seite Application versions (Anwendungsversionen) alle zu löschen Anwendungsversionen aus.
 - e. Wählen Sie Actions (Instance-Aktionen) und anschließend Delete (Löschen) aus.

- f. Aktivieren Sie Delete versions from Amazon S3 (Versionen von Amazon S3 löschen).
 - g. Wählen Sie Delete (Löschen) und dann Done (Fertig).
2. Die Umgebung beenden.
 - a. Wählen Sie im Navigationsbereich getting-started-app und dann in der Umgebungsliste GettingStartedApp-env.
 - b. Wählen Sie Environment actions (Umgebungsaktionen) und dann Terminate Environment (Umgebung beenden) aus.
 - c. Bestätigen Sie, dass Sie GettingStartedApp-env beenden möchten, indem Sie den Umgebungsnamen eingeben und dann Terminate (Beenden) auswählen.
3. Löschen Sie die Anwendung getting-started-app.
 - a. Wählen Sie im Navigationsbereich die getting-started-app.
 - b. Wählen Sie Actions (Aktionen) und anschließend Delete Application (Anwendung löschen).
 - c. Bestätigen Sie, dass Sie getting-started-app löschen möchten, indem Sie den Anwendungsnamen eingeben und dann Delete (Löschen) auswählen.

Nächste Schritte

Da Sie nun wissen, wie Sie eine Elastic Beanstalk-Anwendung und eine Umgebung erstellen, empfehlen wir Ihnen, [Konzepte \(p. 14\)](#) zu lesen. Dieses Thema enthält Informationen zu den Komponenten und der Architektur von Elastic Beanstalk und beschreibt wichtige Überlegungen zum Entwurf Ihrer Elastic Beanstalk-Anwendung.

Zusätzlich zur Elastic Beanstalk-Konsole können Sie die folgenden Tools verwenden, um Elastic Beanstalk-Umgebungen zu erstellen und zu verwalten.

EB CLI

Die EB CLI ist ein Befehlszeilen-Tool für das Erstellen und Verwalten von Umgebungen. Details dazu finden Sie unter [Verwenden des Elastic Beanstalk Command Line Interface \(EB CLI\) \(p. 1008\)](#).

AWS SDK für Java

Das AWS SDK for Java bietet eine Java-API, mit der Sie Anwendungen entwickeln können, die AWS Infrastructure Services verwenden. Mit dem AWS SDK for Java können Sie in nur wenigen Minuten loslegen. Das einzelne Download-Paket enthält die AWS Java-Bibliothek, Codebeispiele und die Dokumentation.

Für das AWS SDK for Java ist das J2SE Development Kit 5.0 oder eine neuere Version erforderlich. Sie können die neueste Java-Software unter <http://developers.sun.com/downloads/> herunterladen. Das SDK erfordert zudem Apache Commons (Codec, HttpClient und Logging) sowie Saxon-HE-Pakete von Drittanbietern, die im Drittanbieter-Verzeichnis des SDK enthalten sind.

Weitere Informationen finden Sie unter [AWS SDK for Java](#).

AWS Toolkit for Eclipse

Das AWS Toolkit for Eclipse ist ein Open Source Plug-in für die Eclipse Java IDE. Sie können es verwenden, um AWS Java-Webprojekte zu erstellen, die mit dem AWS SDK for Java vorkonfiguriert sind, und dann die Webanwendungen in Elastic Beanstalk bereitzustellen. Das Elastic Beanstalk-Plug-in baut auf der Eclipse Web Tools Platform (WTP) auf. Das Toolkit bietet eine Vorlage für eine Travel-Log-Beispiel-Webanwendung, welche die Verwendung von Amazon S3 und Amazon SNS veranschaulicht.

Um sicherzustellen, dass Sie über alle WTP-Abhängigkeiten verfügen, empfehlen wir, mit der Java EE-Verteilung von Eclipse zu beginnen. Sie können sie unter <http://eclipse.org/downloads/> herunterladen.

Weitere Informationen zum Verwenden des Elastic Beanstalk-Plug-ins für Eclipse finden Sie unter [AWS Toolkit for Eclipse](#). Informationen zum Erstellen Ihrer Elastic Beanstalk-Anwendung mit Eclipse finden Sie unter [Java-Anwendungen in Elastic Beanstalk erstellen und bereitstellen \(p. 109\)](#).

AWS SDK für .NET

Mit dem AWS SDK for .NET können Sie Anwendungen entwickeln, die AWS Infrastructure Services verwenden. Mit dem AWS SDK for .NET können Sie in nur wenigen Minuten loslegen. Das einzelne Download-Paket enthält die AWS .NET-Bibliothek, Codebeispiele und die Dokumentation.

Weitere Informationen finden Sie unter [AWS SDK for .NET](#). Weitere unterstützte Versionen von .NET Framework und Visual Studio finden Sie im [AWS SDK for .NET-Entwicklerhandbuch](#).

AWS Toolkit for Visual Studio

Mit dem AWS Toolkit for Visual Studio-Plug-in können Sie eine vorhandene .NET-Anwendung in Elastic Beanstalk bereitstellen. Außerdem können Sie mit den AWS-Vorlagen, die vorkonfiguriert im AWS SDK for .NET enthalten sind, neue Projekte erstellen.

Weitere Informationen zu Voraussetzungen und zur Installation finden Sie unter [AWS Toolkit for Visual Studio](#). Informationen zum Erstellen Ihrer Elastic Beanstalk-Anwendung mit Visual Studio finden Sie unter [Erstellen und Bereitstellen von .NET-Anwendungen in Elastic Beanstalk \(p. 189\)](#).

AWS SDK für JavaScript in Node.js

Mit dem AWS SDK for JavaScript in Node.js können Sie Anwendungen auf den AWS Infrastructure Services erstellen. Mit dem AWS SDK for JavaScript in Node.js können Sie in nur wenigen Minuten loslegen. Das einzelne Download-Paket enthält die AWS Node.js-Bibliothek, Codebeispiele und die Dokumentation.

Weitere Informationen finden Sie unter [AWS SDK for JavaScript in Node.js](#).

AWS SDK for PHP

Mit dem AWS SDK for PHP können Sie Anwendungen auf den AWS Infrastructure Services erstellen. Mit dem AWS SDK for PHP können Sie in nur wenigen Minuten loslegen. Das einzelne Download-Paket enthält die AWS PHP-Bibliothek, Codebeispiele und die Dokumentation.

Das AWS SDK for PHP erfordert PHP 5.2 oder eine neuere Version. Details zum Herunterladen finden Sie unter <http://php.net/>.

Weitere Informationen finden Sie unter [AWS SDK for PHP](#).

AWS SDK für Python (Boto)

Mit dem AWS SDK for Python (Boto) können Sie in nur wenigen Minuten loslegen. Das einzelne Download-Paket enthält die AWS Python-Bibliothek, Codebeispiele und die Dokumentation. Sie können Python-Anwendungen auf APIs aufbauen, mit denen die direkte Programmierung für die Web-Service-Schnittstellen vereinfacht wird.

Die umfangreiche Bibliothek enthält benutzerfreundliche APIs für Python-Entwickler, die viele maschinenbasierte Aufgaben bei der Programmierung für die AWS Cloud übernehmen, darunter Authentifizierung, Anforderungswiederholungen und Fehlerbehandlung. Das SDK enthält auch praktische Python-Beispiele, um mit den Bibliotheken Anwendungen zu erstellen.

Weitere Informationen zu Boto sowie Beispielcode, Dokumentation, Tools und zusätzliche Ressourcen finden Sie im [Python-Entwicklerzentrum](#).

AWS SDK für Ruby

Sie können in nur wenigen Minuten loslegen. Das einzelne Download-Paket enthält die AWS Ruby-Bibliothek, Codebeispiele und die Dokumentation. Sie können Ruby-Anwendungen auf APIs aufbauen, mit denen die direkte Programmierung in den Webservice-Schnittstellen vereinfacht wird.

Die umfangreiche Bibliothek enthält benutzerfreundliche APIs für Ruby-Entwickler, die viele maschinenbasierte Aufgaben bei der Programmierung für die AWS Cloud übernehmen, darunter Authentifizierung, Anforderungswiederholungen und Fehlerbehandlung. Das SDK enthält auch praktische Ruby-Beispiele, um mit den Bibliotheken Anwendungen zu erstellen.

Weitere Informationen über das SDK sowie Beispielcode, Dokumentation, Tools und zusätzliche Ressourcen finden Sie im [Ruby-Entwicklerzentrum](#).

Elastic Beanstalk.Konzepte

Mit AWS Elastic Beanstalk können Sie alle Ressourcen, die Ihre Anwendung ausführt, als Umgebungen verwalten. Nachfolgend einige wichtige Elastic Beanstalk-Konzepte

Anwendung

Eine Elastic Beanstalk-Anwendung ist eine logische Sammlung von Elastic Beanstalk-Komponenten, einschließlich Umgebungen, Versionen und Umgebungskonfigurationen. Vom Konzept her gleicht eine Anwendung in Elastic Beanstalk einem Ordner.

Anwendungsversion

In Elastic Beanstalk bezieht sich eine Anwendungsversion auf eine spezifische, benannte Wiederholung des Bereitstellungscodes für eine Webanwendung. Eine Anwendungsversion verweist auf ein Amazon Simple Storage Service (Amazon S3)-Objekt mit dem bereitstellbaren Code wie eine Java WAR-Datei. Eine Anwendungsversion ist Teil einer Anwendung. Anwendungen können verschiedene Versionen haben und jede Anwendungsversion ist einzigartig. In einer laufenden Umgebung können Sie jede Anwendungsversion bereitstellen, die Sie bereits in die Anwendung hochgeladen haben, oder eine neue Anwendungsversion hochladen und sofort bereitstellen. Sie können mehrere Anwendungsversionen hochladen, um Unterschiede zwischen einer Version Ihrer Webanwendung und einer anderen zu testen.

Umgebung

Eine Umgebung ist eine Sammlung von AWS-Ressourcen, die eine Anwendungsversion ausführen. In jeder Umgebung wird jeweils nur eine einzige Anwendungsversion ausgeführt. Sie können jedoch dieselbe Anwendungsversion oder verschiedene Anwendungsversionen in vielen Umgebungen gleichzeitig ausführen. Wenn Sie eine Umgebung erstellen, stellt Elastic Beanstalk die Ressourcen für das Ausführen der Anwendungsversion, die Sie angegeben haben, bereit.

Umgebungsschicht

Wenn Sie eine Elastic Beanstalk-Umgebung starten, wählen Sie zuerst eine Umgebungsschicht aus. Die Umgebungsschicht bestimmt den Typ der Anwendung, die in der Umgebung ausgeführt wird, und legt fest, welche Ressourcen Elastic Beanstalk bereitstellt, um sie zu unterstützen. Anwendungen, die HTTP-Anforderungen senden, werden in einer [Webserver-Umgebungsschicht \(p. 15\)](#) ausgeführt. Eine Backend-Umgebung, die Aufgaben aus einer Amazon Simple Queue Service (Amazon SQS)-Warteschlange abruft, wird in einer [Worker-Umgebungsebene \(p. 16\)](#) ausgeführt.

Umgebungskonfiguration

Eine Umgebungskonfiguration definiert die Parameter und Einstellungen, die das Verhalten einer Umgebung und der zugeordneten Ressourcen definieren. Wenn Sie die Konfigurationseinstellungen einer Umgebung aktualisieren, wendet Elastic Beanstalk automatisch die Änderungen auf bestehende Ressourcen an oder löscht sie und stellt neue Ressourcen bereit (je nach Art der Änderung).

Gespeicherte Konfiguration

Eine gespeicherte Konfiguration ist eine Vorlage, die Sie als Ausgangspunkt zur Erstellung von eindeutigen Umgebungskonfigurationen verwenden können. Sie können gespeicherte Konfigurationen erstellen und ändern und sie über die Elastic Beanstalk-Konsole, die EB-CLI, die AWS CLI oder die API Umgebungen zuweisen. Die API und die AWS CLI beziehen sich auf gespeicherte Konfigurationen wie Konfigurationsvorlagen.

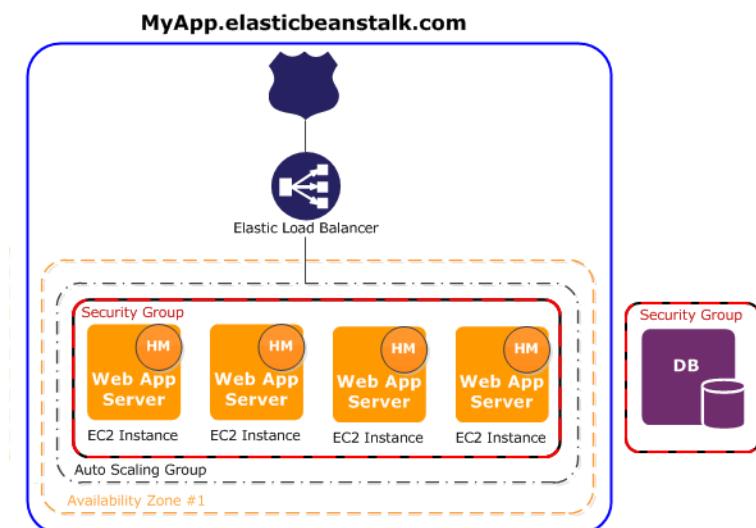
Plattform

Eine Plattform ist eine Kombination von Betriebssystem-, Programmiersprachen-Laufzeit-, Webserver-, Anwendungsserver- und Elastic Beanstalk-Komponenten. Sie entwerfen eine Webanwendung und richten diese auf eine Plattform aus. Elastic Beanstalk bietet eine Vielzahl von Plattformen, auf denen Sie Ihre Anwendungen aufbauen können.

Details dazu finden Sie unter [Elastic Beanstalk-Plattformen \(p. 27\)](#).

Webserver-Umgebungen

Das folgende Diagramm zeigt ein Beispiel für eine Elastic Beanstalk-Architektur für eine Webserver-Umgebungsschicht und veranschaulicht, wie die Komponenten in dieser Art von Umgebung zusammen funktionieren.



Die Umgebung ist der Kern der Anwendung. Im Diagramm wird die Umgebung innerhalb der obersten durchgezogenen Linie dargestellt. Wenn Sie eine Umgebung erstellen, stellt Elastic Beanstalk die Ressourcen für das Ausführen Ihrer Anwendung bereit. AWS-Ressourcen für eine Umgebung haben einen Elastic Load Balancer (ELB im Diagramm), eine Auto Scaling-Gruppe und eine oder mehrere Amazon Elastic Compute Cloud (Amazon EC2)-Instances.

Jede Umgebung hat einen CNAME (URL), der auf einen Load Balancer verweist. Die Umgebung verfügt über eine URL, z. B. `myapp.us-west-2.elasticbeanstalk.com`. Diese URL ist in [Amazon Route 53](#) als Alias für eine Elastic Load Balancing-URL festgelegt – etwa `abcdef-123456.us-`

`west-2.elb.amazonaws.com` – indem ein CNAME-Datensatz verwendet wird. [Amazon Route 53](#) ist ein hochverfügbarer und skalierbarer Domain Name System (DNS)-Web-Service. Er bietet sicheres und zuverlässiges Routing an Ihre Infrastruktur. Ihr Domänenname, den Sie beim DNS-Anbieter registriert haben, leitet Anfragen an den CNAME weiter.

Der Load Balancer befindet sich vor den Amazon EC2-Instances, die Teil einer Auto Scaling-Gruppe sind. Amazon EC2 Auto Scaling startet automatisch weitere Amazon EC2-Instances, um die Datenverarbeitung der Anwendung auch bei einer erhöhten Workload zu gewährleisten. Sinkt die Workload wieder, werden einige Instances von Amazon EC2 Auto Scaling gestoppt. Mindestens eine Instance wird jedoch immer ausgeführt.

Der Software-Stack, auf dem Amazon EC2-Instances ausgeführt werden, hängt vom Containerotyp ab. Ein Containerotyp definiert die Infrastrukturtopologie und den Softwarestack, die für ein bestimmte Umgebung verwendet werden sollen. Eine Elastic Beanstalk-Umgebung mit einem Apache Tomcat-Container verwendet beispielsweise das Amazon Linux-Betriebssystem, Apache-Webserver und Apache Tomcat-Software. Eine Liste mit unterstützten Containerotypen finden Sie unter [Von Elastic Beanstalk unterstützte Plattformen \(p. 32\)](#). Jeder Amazon EC2-Instance mit Ihrer Anwendung nutzt einen dieser Containerotypen. Darüber hinaus wird eine Software-Komponente mit dem Namen Host-Manager (HM) auf jeder Amazon EC2-Instance ausgeführt. Der Host-Manager ist für Folgendes verantwortlich:

- Bereitstellen der Anwendung
- Aggregieren von Ereignissen und Metriken für den Abruf über die Konsole, die API oder die Befehlszeile
- Generieren von Ereignissen auf Instance-Ebene
- Überwachen der Anwendungs-Protokolldateien auf kritische Fehler
- Überwachen der Anwendungsserver
- Patchen von Instance-Komponenten
- Rotieren der Protokolldateien der Anwendung und Veröffentlichen in Amazon S3

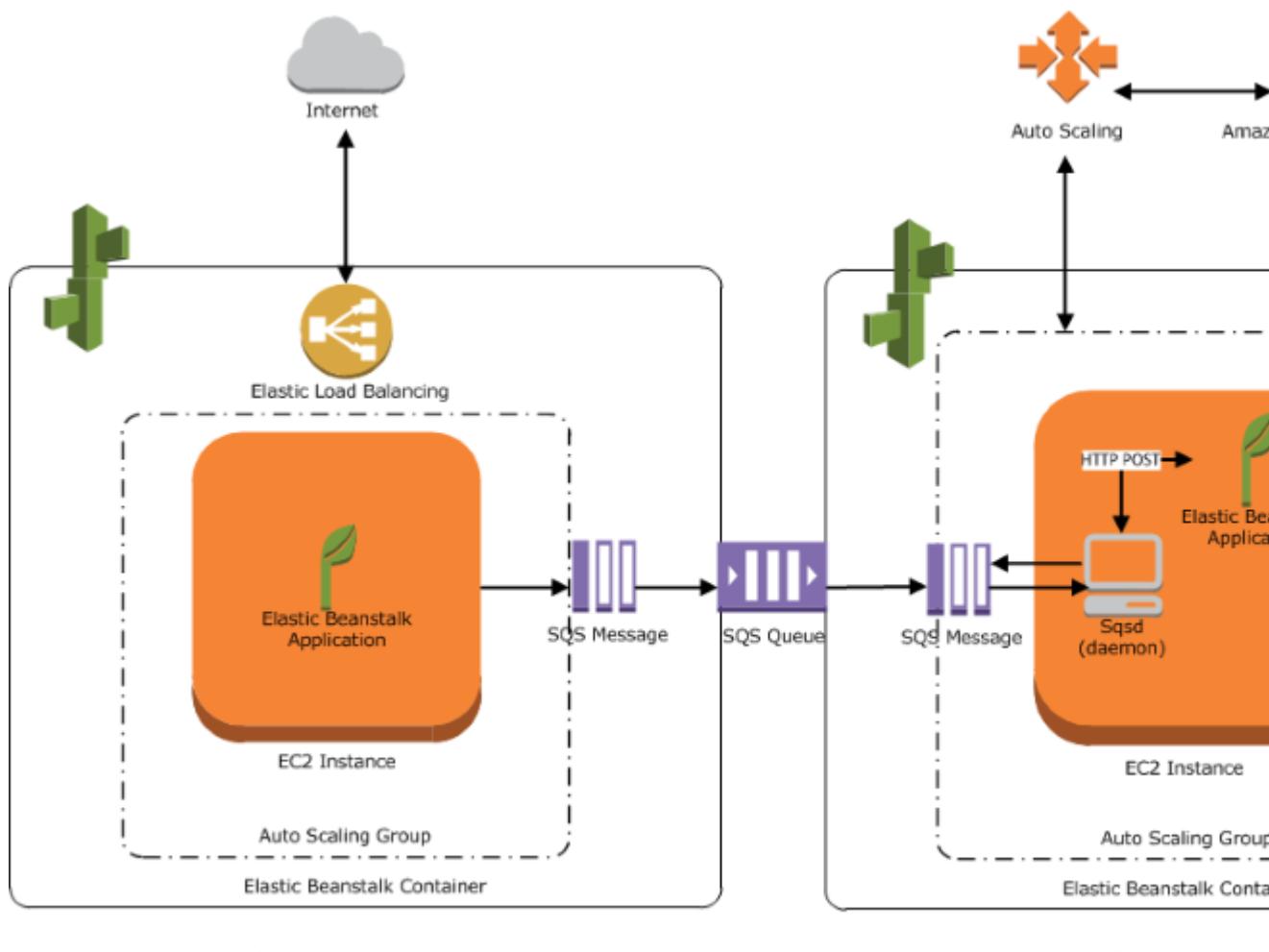
Der Host-Manager meldet Metriken, Fehler und Ereignisse und den Server-Instance-Status, verfügbar über die Elastic Beanstalk-Konsole, APIs und CLIs.

Die Amazon EC2-Instances im Diagramm sind Teil einer Sicherheitsgruppe. Eine Sicherheitsgruppe definiert Firewall-Regeln für Ihre Instances. Standardmäßig definiert Elastic Beanstalk eine Sicherheitsgruppe, die allen Benutzern die Verbindung über Port 80 (HTTP) erlaubt. Sie können mehr als eine Sicherheitsgruppe definieren. Sie können beispielsweise eine Sicherheitsgruppe für Ihren Datenbankserver definieren. Weitere Informationen zu Amazon EC2-Sicherheitsgruppen und zu deren Konfiguration für Ihre Elastic Beanstalk-Anwendung finden Sie unter [Sicherheitsgruppen \(p. 535\)](#).

Worker-Umgebungen

AWS-Ressourcen, die für eine Worker-Umgebungsschicht erstellt wurden, haben eine Auto Scaling-Gruppe, eine oder mehrere Amazon EC2-Instances und eine IAM-Rolle. Für die Worker-Umgebungs-Schicht erstellt Elastic Beanstalk außerdem eine Amazon SQS-Warteschlange und stellt diese bereit, falls Sie noch keine haben. Wenn Sie eine Worker-Umgebungsschicht starten, installiert Elastic Beanstalk die notwendigen Support-Dateien für Ihre gewählte Programmiersprache und einen Daemon auf jeder EC2-Instance in der Auto Scaling-Gruppe. Der Daemon liest Nachrichten aus einer Amazon SQS-Warteschlange. Der Daemon sendet Daten aus jeder gelesenen Nachricht zur Verarbeitung an die Webanwendung, die in der Worker-Umgebung ausgeführt wird. Wenn Sie über mehrere Instances in Ihrer Worker-Umgebung verfügen, hat jede Instance einen eigenen Daemon, aber sie alle werden aus derselben Amazon SQS-Warteschlange gelesen.

Das folgende Diagramm zeigt die verschiedenen Komponenten und deren Interaktion über Umgebungen und AWS-Services hinweg.



Amazon CloudWatch wird für Alarne und Zustandsüberwachung verwendet. Weitere Informationen finden Sie unter [Grundlegende Zustandsberichte \(p. 821\)](#).

Weitere Informationen darüber, wie die Worker-Umgebungsschicht funktioniert, finden Sie unter [AWS Elastic Beanstalk-Worker-Umgebungen \(p. 516\)](#).

Überlegungen zum Entwurf

Da Anwendungen, die mithilfe von Elastic Beanstalk bereitgestellt werden, auf Amazon-Cloud-Ressourcen ausgeführt werden, sollten Sie einige Dinge bei der Konzeption Ihrer Anwendung berücksichtigen: Skalierbarkeit, Sicherheit, dauerhafte Datenspeicherung, Fehlertoleranz, Bereitstellung von Inhalten, Software-Updates und Patches und Konnektivität. Eine umfangreiche Liste technischer AWS-Whitepaper zu Themen wie Architektur, Sicherheit und Wirtschaftlichkeit finden Sie unter [AWS Cloud Computing-Whitepapers](#).

Skalierbarkeit

Wenn Sie den Betrieb in einer physischen Hardware-Umgebung durchführen (im Gegensatz zu einer Cloud-Umgebung), können Sie die Skalierbarkeit auf zwei Art und Weisen angehen. Sie können nach

oben skalieren (vertikale Skalierung) oder nach unten skalieren (horizontale Skalierung). Das Scale-up-Verfahren erfordert die Investition in leistungsstarke Hardware, da die geschäftlichen Anforderungen zunehmen, während das Scale-out-Verfahren die Befolgung eines verteilten Modells an Investition erfordert, sodass Hardware- und Anwendungsübernahmen zielgerichteter, Datensätze verbundener und Design serviceorientierter sind. Das Scale-up-Verfahren kann sehr teuer sein und es bleibt noch das Risiko, dass der Bedarf die Kapazität übersteigt. Obwohl das Scale-out-Verfahren in der Regel effektiver ist, benötigen Sie eine genaue Prognose bezüglich des Bedarfs in regelmäßigen Abständen sowie eine schrittweise Bereitstellung von Infrastruktur, um den Bedarf zu decken. Dieser Ansatz führt oftmals zu ungenutzter Kapazität und erfordert eine sorgfältige Überwachung.

Durch den Wechsel in die Cloud können Sie die Nutzung Ihrer Infrastruktur in Übereinstimmung mit der Anforderung durch Ausnutzen der Elastizität der Cloud bringen. Elastizität ist die Optimierung der Ressourcen, sodass Ihre Infrastruktur, Akquisition und Freigabe rasch mit schwankendem Bedarf skaliert werden kann. Für die Elastizität konfigurieren Sie Ihre Auto Scaling-Einstellungen, um nach oben oder unten zu skalieren, basierend auf Metriken aus den Ressourcen in Ihrer Umgebung (z. B. Auslastung der Server oder Netzwerk-E/A). Mit Auto Scaling können Sie automatisch Rechenkapazität hinzufügen und entfernen, wenn die Nutzung steigt oder fällt. Veröffentlichen Sie Systemmetriken (CPU, Speicher, Festplatten-E/A und Netzwerk-E/A) in Amazon CloudWatch und konfigurieren Sie Alarne, um Auto Scaling-Aktionen auszulösen oder Benachrichtigungen zu senden. Weitere Informationen zur Konfiguration von Auto Scaling finden Sie unter [Auto Scaling-Gruppe für Ihre Elastic Beanstalk-Umgebung \(p. 540\)](#).

Elastic Beanstalk-Anwendungen sollten auch so zustandslos wie möglich sein, durch lose gekoppelten, fehlertolerante Komponenten, die bei Bedarf skaliert werden können. Weitere Informationen zum Konzipieren skalierbarer Anwendungsarchitekturen für AWS finden Sie im Whitepaper [Entwerfen für die Cloud: Bewährte Methoden](#).

Sicherheit

Sicherheit in AWS ist eine [übergreifende Verantwortlichkeit](#). AWS schützt die physischen Ressourcen in Ihrer Umgebung und stellt sicher, dass die Cloud ein sicherer Ort zum Ausführen von Anwendungen ist. Sie sind für die Sicherheit von Daten in und aus Ihrer Elastic Beanstalk-Umgebung und die Sicherheit Ihrer Anwendung verantwortlich.

Zum Schutz des Informationsflusses zwischen Ihrer Anwendung und Clients konfigurieren Sie SSL. Dazu benötigen Sie ein kostenloses Zertifikat von AWS Certificate Manager (ACM). Wenn Sie bereits über ein Zertifikat von einer externen Zertifikatsstelle (CA) verfügen, können Sie dieses Zertifikat mit AWS Certificate Manager programmgesteuert oder unter Verwendung der AWS-CLI importieren.

Wenn AWS Certificate Manager in Ihrer Region nicht [verfügbar ist](#), können Sie ein Zertifikat von einer externen Zertifizierungsstelle kaufen, wie beispielsweise VeriSign oder Entrust. Verwenden Sie dann die AWS CLI, um ein Drittanbieter- oder selbstsigniertes Zertifikat und einen privaten Schlüssel in AWS Identity and Access Management (IAM) hochzuladen. Der öffentliche Schlüssel des Zertifikats authentifiziert Ihren Server für den Browser. Außerdem dient er als Grundlage für die Erstellung des gemeinsamen Sitzungsschlüssels, der die Daten in beide Richtungen verschlüsselt. Weitere Anweisungen zum Erstellen, Hochladen und Zuweisen eines SSL-Zertifikats zu Ihrer Umgebung finden Sie unter [Konfigurieren von HTTPS für Elastic Beanstalk-Umgebung \(p. 778\)](#).

Wenn Sie ein SSL-Zertifikat für Ihre Umgebung konfigurieren, werden Ihre Daten zwischen dem Client und dem Elastic Load Balancing-Load Balancer verschlüsselt. Standardmäßig wird die Verschlüsselung beim Load Balancer beendet und Datenverkehr zwischen dem Load Balancer und Amazon EC2-Instances ist nicht verschlüsselt.

Persistenter Speicher

Elastic Beanstalk-Anwendungen werden auf Amazon EC2-Instances ausgeführt, die keine persistenten lokalen Speicher haben. Wenn die Amazon EC2-Instances beendet werden, wird das lokale Dateisystem

nicht gespeichert und neue Amazon EC2-Instances starten mit einem Standard-Dateisystem. Sie sollten Ihre Anwendung zum Speichern von Daten in einer persistenten Datenquelle konfigurieren. Amazon Web Services bietet eine Reihe von persistenten Speicherservices, von denen Sie Gebrauch machen können. In der folgende Tabelle sind sie aufgelistet.

Storage service	Servicedokumentation	Elastic Beanstalk integrieren
Amazon S3	Amazon Simple Storage Service – Dokumentation	Verwenden von Elastic Beanstalk mit Amazon S3 (p. 985)
Amazon Elastic File System	Amazon Elastic File System – Dokumentation	Elastic Beanstalk mit Amazon Elastic File System verwenden (p. 901)
Amazon Elastic Block Store	Amazon Elastic Block Store Funktionsleitfaden: Elastic Block Store	
Amazon DynamoDB	Amazon DynamoDB – Dokumentation	Elastic Beanstalk mit Amazon DynamoDB verwenden (p. 900)
Amazon Relational Database Service (RDS)	Amazon Relational Database Service – Dokumentation	Verwenden von Elastic Beanstalk mit Amazon RDS (p. 973)

Fehlertoleranz

In der Regel sollten Sie pessimistisch bei der Entwicklung einer Architektur für die Cloud sein. Berücksichtigen Sie beim Design, der Implementierung und Bereitstellung immer eine automatische Wiederherstellung nach einem Ausfall. Verwenden Sie mehrere Availability Zones für Ihre Amazon EC2-Instances und Amazon RDS. Availability Zones basieren auf dem gleichen Konzept wie logische Rechenzentren. Mit Amazon CloudWatch erhalten Sie bessere Einblicke in den Zustand Ihrer Elastic Beanstalk-Anwendung und können bei einem Hardware-Ausfall oder Leistungsabfall die angemessenen Maßnahmen ergreifen. Konfigurieren Sie Ihre Auto Scaling-Einstellungen so, dass Ihre Flotte von Amazon EC2-Instances eine feste Größe behält, sodass fehlerhafte Amazon EC2-Instances durch neue ersetzt werden. Wenn Sie Amazon RDS verwenden, richten Sie den Aufbewahrungszeitraum für Sicherungskopien ein, sodass Amazon RDS automatisch Sicherungskopien erstellen kann.

Bereitstellung von Inhalten

Wenn Benutzer eine Verbindung mit Ihrer Website herstellen, werden ihre Anforderungen über eine Reihe einzelner Netzwerke geleitet. Daher kann es sein, dass Anwender eine schlechtere Leistung durch hohe Latenz bemerken. Amazon CloudFront verteilt Ihre Webinhalte (beispielsweise Bilder, Videos usw.) über ein Netzwerk von Edge-Standorten weltweit und kann damit helfen, Latenzprobleme zu beheben. Endbenutzer werden an den nächsten Edge-Standort weitergeleitet, sodass die Bereitstellung der Inhalte mit der bestmöglichen Leistung erfolgt. Amazon CloudFront funktioniert außerdem reibungslos mit Amazon S3, auf denen die definitiven Originalversionen der Dateien gespeichert sind. Weitere Informationen über Amazon CloudFront finden Sie unter <http://aws.amazon.com/cloudfront>.

Software-Updates und -Patches

Elastic Beanstalk aktualisiert seine Plattformen regelmäßig mit neuer Software und Patches. Elastic Beanstalk aktualisiert laufende Umgebungen nicht automatisch auf neue Plattformversionen. Sie können jedoch eine [Plattformaktualisierung \(p. 491\)](#) initiieren, um Ihre laufende Umgebung zu aktualisieren. Plattformaktualisierungen verwenden [fortlaufende Updates \(p. 484\)](#), damit Ihre Anwendung verfügbar bleibt, indem die Änderungen in Stapeln angewendet werden.

Konnektivität

Elastic Beanstalk muss eine Verbindung zu den Instances in Ihrer Umgebung herstellen können, um Bereitstellungen abzuschließen. Wenn Sie eine Elastic Beanstalk-Anwendung in einer Amazon VPC bereitstellen, hängt die Konfiguration für die Konnektivität, die Sie erstellen, vom Typ der Amazon VPC-Umgebung ab, die Sie erstellen:

- Für Single-Instance-Umgebungen ist keine zusätzliche Konfiguration erforderlich, da Elastic Beanstalk weist jede Amazon EC2-Instance einer öffentlichen Elastic IP-Adresse zuweist, über die die Instance direkt mit dem Internet kommunizieren kann.
- Für skalierbare Umgebungen mit Lastausgleich in einer Amazon VPC mit öffentlichen und privaten Subnetzen müssen Sie die folgenden Schritte ausführen:
 - Erstellen Sie einen Load Balancer im öffentlichen Subnetz, um den eingehenden Datenverkehr des Internets an die Amazon EC2-Instances weiterzuleiten.
 - Erstellen Sie ein NAT-Gerät (Network Address Translation), um ausgehenden Datenverkehr von den Amazon EC2-Instances in privaten Subnetzen in das Internet weiterzuleiten.
 - Erstellen Sie Routing-Regeln für ein- und ausgehenden Datenverkehr für die Amazon EC2-Instances im privaten Subnetz.
 - Wenn Sie eine NAT-Instance verwenden, konfigurieren Sie die Sicherheitsgruppen für die NAT-Instance und Amazon EC2-Instances, um Internet-Kommunikation zu ermöglichen.
- Für eine skalierbare Umgebung mit Lastausgleich in einer Amazon VPC mit einem einzelnen öffentlichen Subnetz ist keine zusätzliche Konfiguration erforderlich, da die Amazon EC2-Instances mit einer öffentlichen IP-Adresse konfiguriert sind, über die die Instances mit dem Internet kommunizieren können.

Weitere Informationen zum Verwenden von AWS Elastic Beanstalk mit Amazon VPC finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon VPC \(p. 988\)](#).

Servicerollen, Instance-Profile und Benutzerrichtlinien

Wenn Sie eine Umgebung erstellen, werden Sie von AWS Elastic Beanstalk aufgefordert, zwei AWS Identity and Access Management (IAM)-Rollen bereitzustellen: eine Servicerolle und ein Instance-Profil. Die [Servicerolle \(p. 21\)](#) wird von Elastic Beanstalk angenommen, um andere AWS-Services in Ihrem Namen zu verwenden. Das [Instance-Profil \(p. 22\)](#) wird auf die Instances in Ihrer Umgebung angewendet. [Anwendungsversionen \(p. 14\)](#) können von Amazon Simple Storage Service (Amazon S3) abgerufen, Protokolle in Amazon S3 hochgeladen und andere Aufgaben ausgeführt werden, die vom Umgebungstyp und der Plattform abhängen.

Die beste Methode, eine korrekt konfigurierte Servicerolle und ein Instance-Profil zu erhalten, besteht im [Erstellen einer Umgebung zum Ausführen einer Beispielanwendung \(p. 435\)](#) in der Elastic Beanstalk-Konsole oder mit der Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI). Wenn Sie eine Umgebung erstellen, erstellen die Clients die erforderlichen Rollen und weisen diese den [verwalteten Richtlinien \(p. 926\)](#) zu, die alle erforderlichen Berechtigungen enthalten.

Zusätzlich zu den beiden Rollen, die Sie Ihrer Umgebung zuweisen, können Sie auch [Benutzerrichtlinien \(p. 24\)](#) erstellen und diese auf IAM-Benutzer und -Gruppen in Ihrem Konto anwenden, damit die Benutzer Elastic Beanstalk-Anwendungen und -Umgebungen erstellen und verwalten können. Elastic Beanstalk bietet verwaltete Richtlinien für vollständigen Zugriff und schreibgeschützten Zugriff.

Sie können Ihre eigenen Instance-Profile und Benutzerrichtlinien für erweiterte Szenarien erstellen. Wenn Ihre Instances Zugriff auf Services benötigen, die nicht in den Standardrichtlinien enthalten sind, können Sie weitere Richtlinien zum Standard hinzufügen oder eine neue erstellen. Sie können auch strengere Benutzerrichtlinien erstellen, wenn die verwaltete Richtlinie zu offen ist. Weitere Informationen zur umfassenden Abdeckung der AWS-Berechtigungen finden Sie im [AWS Identity and Access Management Benutzerhandbuch](#).

Themen

- [Elastic Beanstalk-Servicerolle \(p. 21\)](#)
- [Elastic Beanstalk-Instance-Profil \(p. 22\)](#)
- [Elastic Beanstalk-Benutzerrichtlinie \(p. 24\)](#)

Elastic Beanstalk-Servicerolle

Eine Servicerolle ist die IAM-Rolle, die Elastic Beanstalk annimmt, wenn andere Services in Ihrem Namen aufgerufen werden. Beispiel: Elastic Beanstalk nutzt die Servicerolle, die Sie bei der Erstellung einer Elastic Beanstalk-Umgebung angeben, beim Aufrufen von Amazon Elastic Compute Cloud (Amazon EC2), Elastic Load Balancing und Amazon EC2 Auto Scaling-APIs zum Erfassen von Informationen über die Integrität ihrer AWS-Ressourcen für die [erweiterte Integritätsüberwachung \(p. 824\)](#).

Die verwaltete `AWSElasticBeanstalkEnhancedHealth`-Richtlinie enthält alle Berechtigungen, die Elastic Beanstalk zur Überwachung der Umgebungsintegrität benötigt:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "logs:CreateLogStream",  
            "Resource": "arn:aws:logs:  
                <region>:  
                <account>/aws/elasticbeanstalk/enhancedhealth/  
                <environment>/  
                <loggroup>"  
        }  
    ]  
}
```

```
        "Action": [
            "elasticloadbalancing:DescribeInstanceHealth",
            "elasticloadbalancing:DescribeLoadBalancers",
            "elasticloadbalancing:DescribeTargetHealth",
            "ec2:DescribeInstances",
            "ec2:DescribeInstanceStatus",
            "ec2:GetConsoleOutput",
            "ec2:AssociateAddress",
            "ec2:DescribeAddresses",
            "ec2:DescribeSecurityGroups",
            "sns:Publish"
        ],
        "Resource": [
            "*"
        ]
    }
}
```

Diese Richtlinie umfasst auch Amazon SQS-Aktionen, damit Elastic Beanstalk die Warteschlangenaktivität für Worker-Umgebungen überwachen kann.

Wenn Sie eine Umgebung mit der Elastic Beanstalk-Konsole erstellen, werden Sie von Elastic Beanstalk aufgefordert, eine Servicerolle namens `aws-elasticbeanstalk-service-role` mit dem Standardsatz von Berechtigungen und einer Vertrauensrichtlinie zu erstellen, mit denen Elastic Beanstalk die Servicerolle annehmen kann. Wenn Sie [verwaltete Plattformaktualisierungen \(p. 496\)](#) aktivieren, fügt Elastic Beanstalk eine andere Richtlinie mit Berechtigungen an, die diese Funktion aktivieren.

Wenn Sie eine Umgebung mit dem Befehl [the section called “eb create” \(p. 1053\)](#) der Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI) erstellen und über die Option `--service-role` keine Servicerolle angeben, erstellt Elastic Beanstalk die standardmäßige Servicerolle `aws-elasticbeanstalk-service-role`. Wenn die standardmäßige Servicerolle bereits vorhanden ist, verwendet Elastic Beanstalk diese für die neue Umgebung.

Wenn Sie eine Umgebung mit der Aktion `CreateEnvironment` der Elastic Beanstalk-API erstellen und keine Servicerolle angeben, erstellt Elastic Beanstalk eine serviceverknüpfte Rolle. Dieser eindeutige, von Elastic Beanstalk vordefinierte Servicerollentyp schließt alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert. Die serviceverknüpfte Rolle ist Ihrem Konto zugeordnet. Sie wird von Elastic Beanstalk einmalig erstellt und dann zum Erstellen weiterer Umgebungen wiederverwendet. Sie können die serviceverknüpfte Rolle Ihres Kontos auch im Vorhinein mit IAM erstellen. Wenn das Konto über eine serviceverknüpfte Überwachungsrolle verfügt, können Sie diese zum Erstellen einer Umgebung mit der Elastic Beanstalk-API, der Elastic Beanstalk-Konsole oder der EB CLI verwenden. Weitere Informationen zur Verwendung von serviceverknüpften Rollen in Elastic Beanstalk-Umgebungen finden Sie unter [Servicegebundene Rollen für Elastic Beanstalk verwenden \(p. 915\)](#).

Weitere Informationen zu Servicerollen finden Sie unter [Rollen von Elastic Beanstalk Service verwalten \(p. 909\)](#).

Elastic Beanstalk-Instance-Profil

Ein Instance-Profil ist eine IAM-Rolle, die für in der Elastic Beanstalk-Umgebung gestartete Instances gilt. Wenn Sie eine Elastic Beanstalk-Umgebung erstellen, geben Sie das Instance-Profil an, das verwendet wird, wenn Ihre Instances Folgendes tun:

- Anwendungsversionen (p. 14) von Amazon Simple Storage Service (Amazon S3) abrufen
- Protokolle in Amazon S3 schreiben
- In integrierten AWS X-Ray-Umgebungen (p. 627) Debugging-Daten in X-Ray hochladen
- In Multicontainer-Docker-Umgebungen Container-Bereitstellungen mit Amazon Elastic Container Service koordinieren
- In Worker-Umgebungen aus einer Amazon Simple Queue Service (Amazon SQS)-Warteschlange lesen
- In Worker-Umgebungen die Leader-Wahl mit Amazon DynamoDB durchführen
- In Worker-Umgebungen Instance-Integritätsmetriken in Amazon CloudWatch veröffentlichen

Die verwaltete AWSElasticBeanstalkWebTier-Richtlinie enthält Anweisungen, mit denen Instances in Ihrer Umgebung Protokolle in Amazon S3 hochladen und Debugging-Informationen an X-Ray senden können:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "BucketAccess",  
            "Action": [  
                "s3:Get*",  
                "s3>List*",  
                "s3:PutObject"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:s3:::elasticbeanstalk-*",  
                "arn:aws:s3:::elasticbeanstalk-*/*"  
            ]  
        },  
        {  
            "Sid": "XRayAccess",  
            "Action": [  
                "xray:PutTraceSegments",  
                "xray:PutTelemetryRecords",  
                "xray:GetSamplingRules",  
                "xray:GetSamplingTargets",  
                "xray:GetSamplingStatisticSummaries"  
            ],  
            "Effect": "Allow",  
            "Resource": "*"  
        },  
        {  
            "Sid": "CloudWatchLogsAccess",  
            "Action": [  
                "logs:PutLogEvents",  
                "logs>CreateLogStream",  
                "logs:DescribeLogStreams",  
                "logs:DescribeLogGroups"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:logs:*log-group:/aws/elasticbeanstalk*"  
            ]  
        },  
        {  
            "Sid": "ElasticBeanstalkHealthAccess",  
            "Action": [  
                "elasticbeanstalk:PutInstanceStatistics"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:elasticbeanstalk:instance-statistics:  
            ]  
        }  
    ]  
}
```

```
        "arn:aws:elasticbeanstalk::*:application/*",
        "arn:aws:elasticbeanstalk::*:environment/*"
    ]
}
}
```

Elastic Beanstalk bietet auch verwaltete Richtlinien mit den Namen `AWSElasticBeanstalkWorkerTier` und `AWSElasticBeanstalkMulticontainerDocker` für die anderen Anwendungsfälle. Elastic Beanstalk fügt alle diese Richtlinien an das Standard-Instance-Profil `aws-elasticbeanstalk-ec2-role` an, wenn Sie eine Umgebung mit der Konsole oder EB CLI erstellen.

Wenn Ihre Webanwendung den Zugriff auf beliebige andere AWS-Services erfordert, fügen Sie Anweisungen oder verwaltete Richtlinien zum Instance-Profil zu, die den Zugriff auf diese Services erlauben.

Weitere Informationen zu Instance-Profilen finden Sie unter [Elastic Beanstalk Instance-Profile verwalten \(p. 904\)](#).

Elastic Beanstalk-Benutzerrichtlinie

Erstellen Sie IAM-Benutzer für jede Person, die Elastic Beanstalk verwendet, damit Sie Ihr Stammkonto nicht verwenden oder Anmeldeinformationen weitergeben müssen. Für ein höheres Maß an Sicherheit gewähren Sie nur diesen Benutzern die Berechtigung für den Zugriff auf Services und Funktionen, die sie benötigen.

Elastic Beanstalk erfordert Berechtigungen nicht nur für die eigenen API-Aktionen, sondern auch für einige andere AWS-Services. Elastic Beanstalk nutzt Benutzerberechtigungen zum Starten aller Ressourcen in einer Umgebung, einschließlich EC2-Instances, eines Elastic Load Balancing-Load Balancer und einer Auto Scaling-Gruppe. Elastic Beanstalk nutzt auch Benutzerberechtigungen zum Speichern von Protokollen und Vorlagen in Amazon Simple Storage Service (Amazon S3), Senden von Benachrichtigungen an Amazon SNS, Zuweisen von Instance-Profilen und Veröffentlichen von Metriken in CloudWatch. Elastic Beanstalk erfordert AWS CloudFormation-Berechtigungen, um Ressourcenbereitstellungen und Updates zu orchestrieren. Es sind auch Amazon RDS-Berechtigungen zum Erstellen von Datenbanken (bei Bedarf) und Amazon SQS-Berechtigungen zum Erstellen von Warteschlangen für Worker-Umgebungen erforderlich.

Mit der folgenden Richtlinie wird der Zugriff auf die Aktionen zum Erstellen und Verwalten von Elastic Beanstalk-Umgebungen gestattet. Diese Richtlinie ist in der IAM-Konsole als verwaltete Richtlinie mit dem Namen `AWSElasticBeanstalkFullAccess` verfügbar. Sie können die verwaltete Richtlinie auf IAM-Benutzer oder -Gruppen anwenden, um Berechtigung zur Nutzung von Elastic Beanstalk zu gewähren, oder Ihre eigene Richtlinie erstellen, die Berechtigungen ausschließt, die nicht von Benutzern benötigt werden.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticbeanstalk:*",
        "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": "*"
},
{
"Effect": "Allow",
"Action": [
"iam:AddRoleToInstanceProfile",
"iam>CreateInstanceProfile",
"iam>CreateRole"
],
"Resource": [
"arn:aws:iam::*:role/aws-elasticbeanstalk*",
"arn:aws:iam::*:instance-profile/aws-elasticbeanstalk*"
]
},
{
"Effect": "Allow",
"Action": [
"iam>CreateServiceLinkedRole"
],
"Resource": [
"arn:aws:iam::*:role/aws-service-role/autoscaling.amazonaws.com/
AWSServiceRoleForAutoScaling"
],
"Condition": {
"StringLike": {
"iam:AWSServiceName": "autoscaling.amazonaws.com"
}
}
},
{
"Effect": "Allow",
"Action": [
"iam>CreateServiceLinkedRole"
],
"Resource": [
"arn:aws:iam::*:role/aws-service-role/elasticbeanstalk.amazonaws.com/
AWSServiceRoleForElasticBeanstalk*"
],
"Condition": {
"StringLike": {
"iam:AWSServiceName": "elasticbeanstalk.amazonaws.com"
}
}
},
{
```

```
"Effect": "Allow",
"Action": [
    "iam:AttachRolePolicy"
],
"Resource": "*",
"Condition": {
    "StringLike": {
        "iam:PolicyArn": [
            "arn:aws:iam::aws:policy/AWSElasticBeanstalk*",
            "arn:aws:iam::aws:policy/service-role/AWSElasticBeanstalk*"
        ]
    }
}
]
```

Elastic Beanstalk bietet außerdem eine schreibgeschützte verwaltete Richtlinie mit dem Namen `AWSElasticBeanstalkReadOnlyAccess`. Diese Richtlinie ermöglicht Benutzern das Anzeigen, jedoch nicht das Ändern oder Erstellen, von Elastic Beanstalk-Umgebungen.

Weitere Informationen zu Benutzerrichtlinien finden Sie unter [Elastic Beanstalk-Benutzerrichtlinien verwalten \(p. 926\)](#).

Elastic Beanstalk-Plattformen

AWS Elastic Beanstalk bietet eine Vielzahl von Plattformen, auf denen Sie Ihre Anwendungen aufbauen können. Sie gestalten Ihre Webanwendung auf einer dieser Plattformen und Elastic Beanstalk stellt Ihren Code auf der von Ihnen gewählte Plattformversion bereit, um eine aktive Anwendungsumgebung zu erstellen.

Elastic Beanstalk bietet Plattformen für verschiedene Programmiersprachen, Anwendungsserver sowie Docker-Container. Einige Plattformen verfügen über mehrere gleichzeitig unterstützte Versionen.

Themen

- [Elastic Beanstalk-Plattformen – Glossar \(p. 27\)](#)
- [Modell der übergreifenden Verantwortlichkeit für die Wartung der Elastic Beanstalk-Plattform \(p. 29\)](#)
- [Support-Richtlinie für die Elastic Beanstalk-Plattform \(p. 30\)](#)
- [Von Elastic Beanstalk unterstützte Plattformen \(p. 32\)](#)
- [Elastic Beanstalk-Linux-Plattformen \(p. 33\)](#)
- [Bereitstellen von Elastic Beanstalk-Anwendungen aus Docker-Containern \(p. 46\)](#)
- [Erstellen und Bereitstellen von Go-Anwendungen in Elastic Beanstalk \(p. 97\)](#)
- [Java-Anwendungen in Elastic Beanstalk erstellen und bereitstellen \(p. 109\)](#)
- [Arbeiten mit .NET Core unter Linux \(p. 155\)](#)
- [Erstellen und Bereitstellen von .NET-Anwendungen in Elastic Beanstalk \(p. 189\)](#)
- [Bereitstellen von Node.js-Anwendungen in Elastic Beanstalk \(p. 247\)](#)
- [Erstellen und Bereitstellen von PHP-Anwendungen in Elastic Beanstalk \(p. 285\)](#)
- [Verwenden von Python \(p. 352\)](#)
- [Erstellen und Bereitstellen von Ruby-Anwendungen in Elastic Beanstalk \(p. 381\)](#)

Elastic Beanstalk-Plattformen – Glossar

Im Folgenden finden Sie wichtige Begriffe zu AWS Elastic Beanstalk-Plattformen und deren Lebenszyklus.

Laufzeit

Das Programmieren von sprachspezifischer Laufzeitsoftware (Framework, Bibliotheken, Interpreter, vm usw.), die zur Ausführung Ihres Anwendungscodes erforderlich sind.

Elastic Beanstalk-Komponenten

Softwarekomponenten, die Elastic Beanstalk zu einer Plattform hinzufügt, um die Elastic Beanstalk-Funktionalität zu ermöglichen. Beispiel: Der Agent erweiterter Zustandsberichte wird für die Erfassung und Berichterstellung von Zustandsinformationen benötigt.

Plattform

Eine Plattform ist eine Kombination aus Betriebssystem- (OS), Laufzeit-, Webserver-, Anwendungsserver- und Elastic Beanstalk-Komponenten. Plattformen stellen Komponenten bereit, die zur Ausführung Ihrer Anwendung verfügbar sind.

Plattformversion

Eine Kombination spezifischer Versionen von Betriebssystem- (OS), Laufzeit-, Webserver-, Anwendungsserver- und Elastic Beanstalk-Komponenten. Sie erstellen eine Elastic Beanstalk-Umgebung basierend auf einer Plattformversion und stellen Ihre Anwendung für diese bereit.

Eine Plattformversion verfügt über eine semantische Versionsnummer im Format X.Y.Z. Dabei ist X die Hauptversion, Y die Nebenversion und Z die Patch-Version.

Eine Plattformversion kann sich in einem der folgenden Zustände befinden:

- Unterstützt – Eine Plattformversion, die vollständig aus unterstützten Komponenten besteht. Keine Komponente hat das vom jeweiligen Lieferanten (Besitzer – AWS oder Drittanbieter bzw. Communitys) festgelegte Ende ihres Lebenszyklus (End of Life, EOL) erreicht. Sie erhalten regelmäßige Patches oder kleinere Updates von ihren Anbietern. Elastic Beanstalk stellt Ihnen unterstützte Plattformversionen für die Umgebungserstellung zur Verfügung.
- Außer Betrieb genommen – Eine Plattformversion mit mindestens einer außer Betrieb genommenen Komponente, die das Ende des vom Anbieter festgelegten Lebenszyklus erreicht hat. Nicht mehr aktive Plattformversionen sind weder für neue noch für vorhandene Kunden für die Verwendung in Elastic Beanstalk-Umgebungen verfügbar.

Details zu außer Betrieb genommenen Komponenten finden Sie unter [the section called “Plattform-Unterstützungsrichtlinie” \(p. 30\)](#).

Plattformzweig

Eine Reihe von Plattformversionen, denen bestimmte Versionen (in der Regel Hauptversionen) einiger Komponenten gemeinsam sind, z. B. das Betriebssystem, die Laufzeit oder Elastic Beanstalk-Komponenten. Zum Beispiel: Python 3.6 auf Amazon Linux 64 Bit; IIS 10.0 auf Windows Server 2016 64 Bit. Jede folgende Plattformversion im Zweig ist ein Update der vorherigen Plattformversion.

In jedem Plattformzweig steht Ihnen jeweils die neueste Plattformversion zur Erstellung von Umgebungen zur Verfügung, ohne dass es hierfür Bedingungen gibt. Frühere Plattformversionen im Zweig werden weiterhin unterstützt. Sie können eine Umgebung auf der Basis einer früheren Plattformversion erstellen, wenn Sie diese in den letzten 30 Tagen in einer Umgebung verwendet haben. Diesen früheren Plattformversionen fehlen jedoch die jeweils aktuellen Komponenten. Daher wird ihre Verwendung nicht empfohlen.

Ein Plattformzweig kann sich in einem der folgenden Zustände befinden:

- Unterstützt – Ein Plattformzweig auf dem aktuellen Stand. Er besteht vollständig aus unterstützten Komponenten. Er erhält kontinuierlich Plattformaktualisierungen und wird für die Verwendung in Produktionsumgebungen empfohlen. Eine Liste der unterstützten Plattformzweige finden Sie unter [Von Elastic Beanstalk unterstützte Plattformen](#) im Handbuch über AWS Elastic Beanstalk-Plattformen.///
- Beta – Ein Plattformzweig in der Vorversion vor der Veröffentlichung. Ein solcher Plattformzweig ist experimenteller Natur. Er erhält möglicherweise für eine Weile kontinuierlich Plattformaktualisierungen, wird jedoch nicht langfristig unterstützt. Ein Plattformzweig in der Betaphase wird nicht für die Verwendung in Produktionsumgebungen empfohlen. Sie sollten einen solchen Plattformzweig ausschließlich zu Evaluierungszwecken verwenden. Eine Liste der Plattformzweige in der Betaphase finden Sie unter [Elastic Beanstalk-Plattformversionen in öffentlicher Betaphase](#) im Leitfaden zu AWS Elastic Beanstalk-Plattformen.
- Veraltet – Ein Plattformzweig mit mindestens einer veralteten Komponente. Er erhält kontinuierliche Plattformaktualisierungen, wird jedoch nicht für die Verwendung in Produktionsumgebungen empfohlen. Eine Liste der veralteten Plattformzweige finden Sie unter [Für die Außerbetriebnahme geplante Elastic Beanstalk-Plattformversionen](#) im Handbuch zu AWS Elastic Beanstalk-Plattformen.///
- Außer Betrieb genommen – Ein Plattformzweig mit mindestens einer außer Betrieb genommenen Komponente. Er erhält keine weiteren Plattformaktualisierungen und wird nicht für die Verwendung in Produktionsumgebungen empfohlen. Außer Betrieb genommene Zweige sind im Handbuch zu AWS Elastic Beanstalk-Plattformen nicht aufgeführt. Elastic Beanstalk stellt Ihnen keine Plattformversionen außer Betrieb genommener Plattformzweige für die Erstellung von Umgebungen bereit.

Für eine unterstützte Komponente gibt es keinen vom Lieferanten (Besitzer oder Community) festgelegten Termin für die Außerbetriebnahme. Beim Lieferanten kann es sich um AWS oder einen

Drittanbieter handeln. Für eine veraltete Komponente gibt es einen vom Lieferanten festgelegten Termin für die Außerbetriebnahme. Eine außer Betrieb genommene Komponente hat das Ende des Lebenszyklus (End of Life, EOL) erreicht und wird vom Lieferanten nicht länger unterstützt. Details zu außer Betrieb genommenen Komponenten finden Sie unter [the section called “Plattform-Unterstützungsrichtlinie” \(p. 30\)](#).

Wenn Ihre Umgebung einen veralteten oder außer Betrieb genommenen Plattformzweig verwendet, sollten Sie diesen auf eine Plattformversion in einem unterstützten Plattformzweig aktualisieren. Details finden Sie unter [the section called “Plattformaktualisierungen” \(p. 491\)](#).

Plattformaktualisierung

Eine Veröffentlichung einer neuen Plattformversion, die Aktualisierungen für einige Komponenten der Plattform enthält. Dabei kann es sich um Betriebssystem-, Laufzeit-, Webserver-, Anwendungsserver- und Elastic Beanstalk-Komponenten handeln. Plattformaktualisierungen folgen der Taxonomie des semantischen Versioning und können mehrere Ebenen besitzen:

- Hauptaktualisierung – Eine Aktualisierung, die Änderungen enthält, die mit vorhandenen Plattformversionen nicht kompatibel sind. Möglicherweise sind Änderungen an Ihrer Anwendung erforderlich, damit sie auf einer neuen Hauptversion ordnungsgemäß ausgeführt wird. Eine Hauptaktualisierung besitzt eine neue Hauptversionsnummer für die Plattform.
- Nebenaktualisierung – Eine Aktualisierung, die Funktionen hinzufügt, die mit einer vorhandenen Plattformversion abwärtskompatibel sind. Es sind keine Änderungen an Ihrer Anwendung erforderlich, um sie auf einer neuen Nebenversion ordnungsgemäß auszuführen. Eine Nebenaktualisierung besitzt eine neue Nebenversionsnummer für die Plattform.
- Patch-Aktualisierung – Eine Aktualisierung, die aus Wartungsveröffentlichungen (Fehlerbehebungen, Sicherheitsupdates und Leistungsverbesserungen) besteht, die mit einer vorhandenen Plattformversion abwärtskompatibel sind. Eine Patch-Aktualisierung besitzt eine neue Patch-Plattform-Versionsnummer.

Verwaltete Aktualisierungen

Eine Elastic Beanstalk-Funktion, die automatisch Patch- und Nebenaktualisierungen auf Betriebssystem- (OS), Laufzeit-, Webserver-, Anwendungsserver- und Elastic Beanstalk-Komponenten für eine von Elastic Beanstalk unterstützte Plattformversion anwendet. Eine verwaltete Aktualisierung wendet eine neuere Plattformversion im selben Plattformzweig auf Ihre Umgebung an. Sie können verwaltete Aktualisierungen so konfigurieren, dass nur Patch-Aktualisierungen oder Neben- und Patch-Aktualisierungen angewendet werden. Sie können verwaltete Aktualisierungen auch vollständig deaktivieren.

Weitere Informationen finden Sie unter [Verwaltete Plattformaktualisierungen \(p. 496\)](#).

Modell der übergreifenden Verantwortlichkeit für die Wartung der Elastic Beanstalk-Plattform

AWS und unseren Kunden haben eine gemeinsame Verantwortung, um ein hohes Maß an Sicherheit und Compliance für Softwarekomponenten zu erzielen. Dieses Modell der übergreifenden Verantwortung bedeutet für Sie eine deutliche Verringerung des Betriebsaufwands.

Weitere Informationen finden Sie unter dem [AWS-Modell übergreifender Verantwortlichkeit](#).

AWS Elastic Beanstalk unterstützt Sie bei Umsetzung Ihres Anteils am Modell übergreifender Verantwortlichkeit, indem eine Funktion für verwaltete Aktualisierungen bereitgestellt wird. Diese Funktion wendet automatisch Patches und kleinere Aktualisierungen für eine von Elastic Beanstalk unterstützte Plattformversion an. Wenn eine verwaltete Aktualisierung fehlschlägt, benachrichtigt Elastic Beanstalk Sie über den Ausfall, um sicherzustellen, dass Sie informiert sind und unverzüglich reagieren können.

Weitere Informationen finden Sie unter [Verwaltete Plattformaktualisierungen \(p. 496\)](#).

Darüber hinaus leistet Elastic Beanstalk Folgendes:

- Veröffentlicht seine [Richtlinie zur Plattformunterstützung \(p. 30\)](#) und seinen Aussonderungsplan für die nächsten 12 Monate.
- Veröffentlichung von Patch-, Minor- und Major-Updates für Betriebssystem (OS), Laufzeitumgebung, Anwendungsserver und Webserverkomponenten in der Regel innerhalb von 30 Tagen nach ihrer Verfügbarkeit. Elastic Beanstalk ist für die Erstellung von Updates für Elastic Beanstalk-Komponenten verantwortlich, die auf den unterstützten Plattformversionen vorhanden sind. Alle anderen Aktualisierungen stammen direkt von ihren Lieferanten (Eigentümern oder Community).

Sie sind für Folgendes verantwortlich:

- Aktualisieren aller Komponenten, die Sie kontrollieren (im [AWS-Modell übergreifender Verantwortlichkeit](#) als Customer (Kunde) identifiziert). Dies umfasst die Gewährleistung der Sicherheit Ihrer Anwendung, Ihrer Daten und aller Komponenten, die Ihre Anwendung benötigt und die Sie heruntergeladen haben.
- Sicherstellen, dass Ihre Elastic Beanstalk-Umgebungen auf einer unterstützte Plattformversion ausgeführt werden, und Migrieren einer jeden Umgebung, die auf einer nicht mehr unterstützten Plattformversion ausgeführt wird, auf eine unterstützte Version.
- Lösen aller Probleme bei fehlgeschlagenen verwalteten Aktualisierungsversuchen und Neuversuchen der Aktualisierung.
- Selbstständiges Patchen von Betriebssystem-, Laufzeit-, Anwendungsserver- und Webserver-Komponenten, wenn Sie sich gegen Elastic Beanstalk-verwaltete Aktualisierungen entschieden haben. Sie können hierzu [Plattformaktualisierungen manuell anwenden \(p. 491\)](#) oder die Komponenten auf allen relevanten Umgebungsressourcen direkt patchen.
- Verwalten der Sicherheit und Compliance aller AWS-Services, die Sie außerhalb von Elastic Beanstalk verwenden, gemäß dem [AWS Modell der übergreifenden Verantwortlichkeit](#).

Support-Richtlinie für die Elastic Beanstalk-Plattform

AWS Elastic Beanstalk bietet eine Vielzahl von Plattformen zur Ausführung von Anwendungen bei AWS. Elastic Beanstalk unterstützt Plattform-Banches, die laufende Minor- und Patch-Updates von ihren Lieferanten (Eigentümer oder Community) erhalten. Eine vollständige Definition zugehöriger Begriffe finden Sie unter [Elastic Beanstalk-Plattformen – Glossar \(p. 27\)](#).

Wenn eine Komponente (Betriebssystem [OS], Laufzeit, Anwendungsserver oder Webserver) einer unterstützten Plattformvariante von ihrem Lieferanten als am Ende der Lebensdauer (End-of-Life, EOL) gekennzeichnet wird, kennzeichnet Elastic Beanstalk die Plattformvariante als nicht mehr aktiv. Wenn eine Plattformvariante als nicht mehr aktiv gekennzeichnet ist, stellt Elastic Beanstalk sie sowohl bestehenden als auch neuen Elastic Beanstalk-Kunden nicht mehr für Bereitstellungen auf neuen Umgebungen zur Verfügung. Nicht mehr aktive Plattformvarianten sind für vorhandene Kundenumgebungen noch für einen Zeitraum von 90 Tagen ab dem veröffentlichten Ausmusterungsdatum verfügbar.

Elastic Beanstalk kann für nicht mehr aktive Plattformvarianten keine Sicherheitsaktualisierungen, keinen technischen Support und keine Hotfixes mehr bereitstellen, da die Lieferanten ihre Komponente als EOL gekennzeichnet haben. Für bestehende Kunden, die eine Elastic Beanstalk-Umgebung auf einer nicht mehr aktiven Plattformversion über den Zeitraum von 90 Tagen hinaus ausführen, muss Elastic Beanstalk die Elastic Beanstalk-Komponenten möglicherweise automatisch entfernen und fortlaufende Verwaltungs- und Support-Verantwortlichkeiten der laufenden Anwendung und der zugehörigen AWS-Ressourcen auf den Kunden übertragen. Damit Sie weiterhin von wichtigen Sicherheits-, Leistungs-

und Funktionserweiterungen profitieren können, die von Komponentenlieferanten in neueren Versionen angeboten werden, empfehlen wir Ihnen dringend, alle Ihre Elastic Beanstalk-Umgebungen auf eine unterstützte Plattformversion zu aktualisieren.

Zeitplan für die Ausmusterung von Plattformvarianten

Die folgenden Tabellen listen vorhandene Plattformkomponenten auf, die entweder als nicht mehr aktiv gekennzeichnet sind oder für die in den nächsten 12 Monaten Ausmusterungstermine geplant sind. Die Tabellen geben das Enddatum der Verfügbarkeit für Elastic Beanstalk-Plattformvarianten an, die diese Komponenten enthalten.

Webserver-Versionen

Webserver-Version	Enddatum der Verfügbarkeit		
Apache HTTP Server 2.2	31. Oktober 2020		
Nginx 1.12.2	31. Oktober 2020		

Laufzeitversionen

Laufzeitversion	Enddatum der Verfügbarkeit		
Go 1.3–1.10	31. Oktober 2020		
Java 6	31. Oktober 2020		
Node.js 4.x-8.x	31. Oktober 2020		
PHP 5.4–5.6	31. Oktober 2020		
PHP 7.0–7.1	31. Oktober 2020		
Python 2.6, 2.7, 3.4	31. Oktober 2020		
Ruby 1.9.3	31. Oktober 2020		
Ruby 2.0–2.3	31. Oktober 2020		

Anwendungsserver-Versionen

Anwendungsserver-Version	Enddatum der Verfügbarkeit		
Tomcat 6	31. Oktober 2020		
Tomcat 8	31. Oktober 2020		

Ausgemusterte Plattformvarianten

In den folgenden Tabellen sind Plattformkomponenten aufgeführt, die in der Vergangenheit als ausgemustert markiert wurden. Die Tabellen geben das Datum an, an dem Elastic Beanstalk-Plattformvarianten ausgemustert hat, die diese Komponenten enthielten.

Betriebssystemversionen (OS)

Betriebssystemversion	Datum der Plattformausmusterung		
Windows Server 2008 R2	28. Oktober 2019		

Von Elastic Beanstalk unterstützte Plattformen

AWS Elastic Beanstalk bietet eine Vielzahl von Plattformen, auf denen Sie Ihre Anwendungen aufbauen können. Sie gestalten Ihre Webanwendung auf einer dieser Plattformen und Elastic Beanstalk stellt Ihren Code auf der von Ihnen gewählte Plattformversion bereit, um eine aktive Anwendungsumgebung zu erstellen.

Elastic Beanstalk bietet Plattformen für Programmiersprachen (Go, Java, Node.js, PHP, Python, Ruby), Anwendungsserver (Tomcat, Passenger, Puma) und Docker-Container. Einige Plattformen verfügen über mehrere gleichzeitig unterstützte Versionen.

Elastic Beanstalk stellt die Ressourcen bereit, die Sie für die Ausführung Ihrer Anwendung benötigen, einschließlich einer oder mehrerer Amazon EC2-Instances. Welcher Software-Stack auf den Amazon EC2-Instances ausgeführt wird, hängt von der spezifischen Plattformversion ab, die Sie für Ihre Umgebung ausgewählt haben.

Sie können über den Namen eines Lösungs-Stacks, der unter dem Namen der Plattformversion aufgelistet wird, eine Umgebung mit der [EB CLI \(p. 1008\)](#), [Elastic Beanstalk-API](#) oder [AWS CLI](#) starten. Sie können auch Namen von Lösungs-Stacks mit der [ListAvailableSolutionStacks-API \(aws elasticbeanstalk list-available-solution-stacks in der AWS CLI\)](#) aus dem Service abrufen. Diese Operation gibt alle Lösungs-Stacks zurück, die Sie verwenden können, um eine Umgebung erstellen.

Note

Jede Plattform verfügt über unterstützte und nicht mehr aktive Plattformversionen. Sie können jederzeit eine Umgebung basierend auf einer unterstützten Plattformversion erstellen. Nicht mehr aktive Plattformversionen sind für vorhandene Kundenumgebungen nur noch für einen Zeitraum von 90 Tagen ab dem veröffentlichten Ausmusterungsdatum verfügbar. Eine Liste der veröffentlichten Plattformversion-Ausmusterungstermine finden Sie unter [Zeitplan für die Ausmusterung von Plattformvarianten \(p. 31\)](#).

Wenn Elastic Beanstalk eine Plattform aktualisiert, werden vorherige Plattformversionen weiterhin unterstützt. Sie verfügen jedoch nicht über die neuesten Komponenten und es wird nicht empfohlen, sie zu verwenden. Wir raten Ihnen zur Umstellung auf die neueste Plattformversion. Sie können weiterhin eine Umgebung basierend auf einer früheren Plattformversion erstellen, sofern Sie sie in den letzten 30 Tagen (mit demselben Konto in derselben Region) in einer Umgebung verwendet haben.

Sie können die Software, von der Ihre Anwendung auf Ihrer Plattform abhängig ist, anpassen und konfigurieren. Weitere Informationen finden Sie unter [Anpassen von Software auf Linux-Servern \(p. 725\)](#) und [Anpassen von Software auf Windows-Servern \(p. 738\)](#). Ausführliche Versionshinweise für aktuelle Versionen finden Sie in den [Versionshinweisen zu AWS Elastic Beanstalk](#).

Unterstützte Plattformversionen

Alle aktuellen Plattformversionen finden Sie unter [Von Elastic Beanstalk unterstützte Plattformen](#) im Leitfaden für AWS Elastic Beanstalk-Plattformen. Jeder plattformspezifische Abschnitt verweist auch auf den Plattformverlauf, eine Liste früherer Plattformversionen. Verwenden Sie einen der folgenden Links für den direkten Zugriff auf die Versionsliste einer bestimmten Plattform.

- Docker
- Multicontainer-Docker
- Vorkonfigurierte Docker-Plattform
- Go
- Java SE
- Tomcat
- .NET Core auf Linux
- .NET mit Windows Server
- Node.js
- PHP
- Python
- Ruby

Elastic Beanstalk-Linux-Plattformen

AWS Elastic Beanstalk bietet eine Vielzahl von Plattformen, auf denen Sie Ihre Anwendungen aufbauen können. Sie gestalten Ihre Webanwendung auf einer dieser Plattformen und Elastic Beanstalk stellt Ihren Code auf der von Ihnen gewählten Plattformversion bereit, um eine aktive Anwendungsumgebung zu erstellen.

Elastic Beanstalk bietet Plattformen für verschiedene Programmiersprachen, Anwendungsserver sowie Docker-Container. Einige Plattformen verfügen über mehrere gleichzeitig unterstützte Versionen.

Eine vollständige Abdeckung der Elastic Beanstalk-Plattformen finden Sie unter [Elastic Beanstalk-Plattformen \(p. 27\)](#).

Viele der von Elastic Beanstalk unterstützten Plattformen basieren auf dem Linux-Betriebssystem (OS). Insbesondere basieren diese Plattformen auf Amazon Linux, einer von AWS bereitgestellten Linux-Distribution. Elastic Beanstalk-Linux-Plattformen verwenden Amazon Elastic Compute Cloud (Amazon EC2)-Instances und diese Instances führen Amazon Linux aus. Weitere Informationen finden Sie unter [Amazon Linux](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Die Elastic Beanstalk-Linux-Plattformen bieten eine Menge Funktionalität. Sie können die Plattformen auf verschiedene Arten erweitern, um Ihre Anwendung zu unterstützen. Details dazu finden Sie unter [the section called "Erweitern von Linux-Plattformen" \(p. 34\)](#).

Themen

- [Linux-Plattformversionen \(p. 33\)](#)
- [Liste der Elastic Beanstalk-Linux-Plattformen \(p. 34\)](#)
- [Erweitern von Elastic Beanstalk-Linux-Plattformen \(p. 34\)](#)

Linux-Plattformversionen

AWS bietet zwei Versionen von Amazon Linux: [Amazon Linux 2](#) und [Amazon Linux AMI](#). Einige wichtige Verbesserungen in Amazon Linux 2 im Vergleich zu Amazon Linux AMI sind:

- Amazon Linux 2 bietet langfristigen Support.
- Amazon Linux 2 ist in Form von VM-Abbildern für die lokale Entwicklung und das Testen verfügbar.
- Amazon Linux 2 enthält aktualisierte Komponenten: Linux-Kernel, C-Bibliothek, Compiler und Tools. Es verwendet auch den systemd-Service und den Systemmanager im Gegensatz zum System V init-System in Amazon Linux AMI.

Elastic Beanstalk verwaltet Plattformversionen mit beiden Amazon Linux-Versionen. Weitere Informationen zu unterstützten Plattformversionen finden Sie unter [Von Elastic Beanstalk unterstützte Plattformen \(p. 32\)](#).

Note

Amazon Linux 2-Plattformversionen sind mit früheren Amazon Linux AMI-Plattformversionen nicht kompatibel. Wenn Sie Ihre Elastic Beanstalk-Anwendung zu Amazon Linux 2 migrieren, lesen Sie [the section called “Aktualisieren auf Amazon Linux 2” \(p. 503\)](#).

Liste der Elastic Beanstalk-Linux-Plattformen

In der folgenden Liste werden die Linux-Plattformen erwähnt, die Elastic Beanstalk für verschiedene Programmiersprachen sowie für Docker-Container unterstützt, sowie Links zu Kapiteln über diese in diesem Entwicklerhandbuch.

- [Docker \(p. 46\)](#)
- [Go \(p. 97\)](#)
- [Java \(p. 109\)](#)
- [.NET Core auf Linux \(p. 155\)](#)
- [Node.js \(p. 247\)](#)
- [PHP \(p. 285\)](#)
- [Python \(p. 352\)](#)
- [Ruby \(p. 381\)](#)

Erweitern von Elastic Beanstalk-Linux-Plattformen

Die [AWS Elastic Beanstalk-Linux-Plattformen \(p. 33\)](#) bieten eine Vielzahl von gebrauchsfertiger Funktionen, um die Entwicklung und Ausführung Ihrer Anwendung zu unterstützen. Bei Bedarf können Sie die Plattformen auf verschiedene Arten erweitern, um Optionen zu konfigurieren, Software zu installieren, Dateien und Startbefehle hinzuzufügen, Build- und Laufzeitanweisungen bereitzustellen und Initialisierungsskripts hinzuzufügen, die in verschiedenen Bereitstellungsphasen der Amazon Elastic Compute Cloud (Amazon EC2)-Instances Ihrer Umgebung ausgeführt werden.

Buildfile und Procfile

Auf einigen Plattformen können Sie anpassen, wie Sie Ihre Anwendung erstellen oder vorbereiten und die Prozesse angeben, die Ihre Anwendung ausführen. Jedes einzelne Plattformthema erwähnt speziell Buildfile und/oder Procfile, wenn die Plattform sie unterstützt. Suchen Sie nach Ihrer spezifischen Plattform unter [Plattformen \(p. 27\)](#).

Syntax und Semantik sind für alle unterstützenden Plattformen identisch und sind wie auf dieser Seite beschrieben. Einzelne Plattformthemen erwähnen die spezifische Verwendung dieser Dateien zum Erstellen und Ausführen von Anwendungen in ihren jeweiligen Sprachen.

Buildfile

Um einen benutzerdefinierten Build- und Konfigurationsbefehl für Ihre Anwendung anzugeben, legen Sie eine Datei mit dem Namen `buildfile` im Stammverzeichnis Ihrer Anwendungsquelle ab. Beim Dateinamen muss die Groß- und Kleinschreibung beachtet werden. Verwenden Sie die folgende Syntax für Ihre `Buildfile`.

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

Der Befehl in Ihrer `Buildfile` muss folgendem regulären Ausdruck entsprechen: `^[A-Za-z0-9_-]+:\n\s*[^\s].*$`

Elastic Beanstalk überwacht nicht die Anwendung, die mit einer `Buildfile` ausgeführt wird. Verwenden Sie eine `Buildfile` für Befehle, die für kurze Zeiträume ausgeführt und nach Abschluss ihrer Aufgaben beendet werden. Verwenden Sie für länger dauernde Anwendungsprozesse, die nicht beendet werden sollten, eine [Procfile \(p. 35\)](#).

Alle Pfade in der `Buildfile` sind relativ zum Stammverzeichnis des Quell-Bundle. Im folgenden Beispiel einer `Buildfile` ist `build.sh` ein Shell-Skript, das sich im Stammverzeichnis des Quell-Bundle befindet.

Example Buildfile

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

Wenn Sie benutzerdefinierte Build-Schritte bereitstellen möchten, empfehlen wir Ihnen, für alles, außer für die einfachsten Befehle, anstelle einer `Buildfile` `predeploy`-Plattform-Hooks zu verwenden. Plattform-Hooks ermöglichen umfangreichere Skripts und eine bessere Fehlerbehandlung. Plattformhaken werden im nächsten Abschnitt beschrieben.

Procfile

Um benutzerdefinierte Befehle zum Starten und Ausführen der Anwendung anzugeben, legen Sie eine Datei mit dem Namen `Procfile` im Stammverzeichnis Ihrer Anwendungsquelle ab. Beim Dateinamen muss die Groß- und Kleinschreibung beachtet werden. Verwenden Sie die folgende Syntax für Ihre `Procfile`. Sie können einen oder mehrere Befehle angeben.

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

Jede Zeile in Ihrer `Procfile` muss folgendem regulären Ausdruck entsprechen: `^[A-Za-z0-9_-]+:\n\s*[^\s].*$`

Verwenden Sie eine `Procfile` für länger dauernde Anwendungsprozesse, die nicht beendet werden sollten. Elastic Beanstalk erwartet von Prozessen, die aus der `Procfile` ausgeführt werden, dass sie kontinuierlich laufen. Elastic Beanstalk überwacht diese Prozesse und startet alle Prozesse, die beendet werden, neu. Für kurz laufende Prozesse verwenden Sie eine [Buildfile \(p. 34\)](#).

Alle Pfade in der `Procfile` sind relativ zum Stammverzeichnis des Quell-Bundle. Im folgenden Beispiel definiert `Procfile` drei Prozesse. Die erste, die im Beispiel `web` genannt wird, ist die Hauptwebanwendung.

Example Procfile

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

Elastic Beanstalk konfiguriert den Proxy-Server so, dass Anforderungen an Ihre Haupt-Web-Anwendung über Port 5000 weitergeleitet werden. Sie können diese Port-Nummer konfigurieren. Eine übliche Verwendung für `Procfile` besteht darin, diese Portnummer als Befehlsargument an Ihre Anwendung zu übergeben. Weitere Informationen zur Proxy-Konfiguration finden Sie im erweiterbaren Abschnitt unter Reverse-Proxy-Konfiguration auf dieser Seite.

Elastic Beanstalk erfasst Standardausgabe- und Fehler-Streams von `Procfile`-Prozessen in Protokolldateien. Elastic Beanstalk benennt die Protokolldateien nach dem Prozess und speichert sie in `/var/log`. Beispielsweise generiert der `web`-Prozess aus dem vorherigen Beispiel Protokolle mit den Namen `web-1.log` und `web-1.error.log` für `stdout` bzw. `stderr`.

Plattform-Hooks

Plattform-Hooks wurden speziell entwickelt, um die Plattform Ihrer Umgebung zu erweitern. Es handelt sich um benutzerdefinierte Skripts und andere benutzerdefinierte ausführbare Dateien, die Sie als Teil des Quellcodes Ihrer Anwendung bereitstellen und die von Elastic Beanstalk in verschiedenen Phasen der Instance-Bereitstellung ausgeführt werden.

Note

Plattform-Hooks werden auf Amazon Linux AMI-Plattformversionen (vor Amazon Linux 2) nicht unterstützt.

Plattform-Hooks für die Anwendungsbereitstellung

Eine Anwendungsbereitstellung erfolgt, wenn Sie ein neues Quellpaket zur Bereitstellung angeben oder wenn Sie eine Konfigurationsänderung ausführen, für die alle Instances der Umgebung beendet und neu erstellt werden müssen.

Um Plattform-Hooks bereitzustellen, die während einer Anwendungsbereitstellung ausgeführt werden, speichern Sie die Dateien in einem der folgenden Unterverzeichnisse des Verzeichnisses `.platform/hooks` im Quellpaket.

- `prebuild` – Die hier aufgeführten Dateien werden ausgeführt, nachdem die Elastic Beanstalk-Plattform-Engine das Anwendungs-Quell-Bundle heruntergeladen und extrahiert hat und bevor die Anwendung und der Webserver eingerichtet und konfiguriert werden.

Die `prebuild`-Dateien werden ausgeführt, nachdem alle Befehle ausgeführt wurden, die im Abschnitt [Commands \(p. 731\)](#) einer beliebigen Konfigurationsdatei gefunden wurden, und bevor `Buildfile`-Befehle ausgeführt werden.

- `predeploy` – Die hier aufgeführten Dateien werden ausgeführt, nachdem die Elastic Beanstalk-Plattform-Engine die Anwendung und den Webserver eingerichtet und konfiguriert hat und bevor sie an ihrem endgültigen Laufzeitspeicherort bereitgestellt werden.

Die `predeploy`-Dateien werden nach dem Ausführen von Befehlen im Abschnitt [container_commands \(p. 734\)](#) einer beliebigen Konfigurationsdatei und vor dem Ausführen von `Procfile`-Befehlen ausgeführt.

- `postdeploy` – Die hier aufgeführten Dateien werden ausgeführt, nachdem die Elastic Beanstalk-Plattform-Engine die Anwendung und den Proxyserver bereitgestellt hat.

Dies ist der letzte Schritt im Bereitstellungsworkflow.

Plattform-Hooks für die Konfigurationsbereitstellung

Eine Konfigurationsbereitstellung erfolgt, wenn Sie Konfigurationsänderungen ausführen, mit denen die Instances der Umgebung aktualisiert, jedoch nicht neu erstellt werden. Die folgenden Optionsaktualisierungen führen zu einer Konfigurationsaktualisierung.

- [Umgebungseigenschaften und plattformspezifische Einstellungen \(p. 621\)](#)
- [Statische Dateien \(p. 776\)](#)
- [AWS X-Ray-Daemon \(p. 627\)](#)
- [Protokollspeicher und Streamen \(p. 630\)](#)
- Anwendungs-Port (erweitern Sie den Abschnitt Reverse-Proxy-Konfiguration auf dieser Seite, um Details anzuzeigen)

Um Hooks bereitzustellen, die während einer Konfigurationsbereitstellung ausgeführt werden, speichern Sie sie im Verzeichnis `.platform/confighooks` im Quellpaket. Es werden die gleichen drei Unterverzeichnisse wie für Hooks für die Anwendungsbereitstellung verwendet.

Weitere Informationen zu Platform-Hooks

Bei Hook-Dateien kann es sich um Binär- oder Skriptdateien handeln, die mit einer `#!`-Zeile mit dem Interpreter-Pfad beginnen, z. B. `#!/bin/bash`. Alle Dateien müssen über die Ausführungsberechtigung verfügen. Legen Sie mit `chmod +x` die Ausführungs berechtigung für Ihre Hook-Dateien fest.

Elastic Beanstalk führt in jedem dieser Verzeichnisse Dateien in lexikographischer Reihenfolge der Dateinamen aus. Alle Dateien werden als `root`-Benutzer ausgeführt. Das aktuelle Arbeitsverzeichnis (`cwd`) für Platform-Hooks ist das Stammverzeichnis der Anwendung. Für `prebuild`- und `predeploy`-Dateien ist es das Anwendungs-Staging-Verzeichnis, und für `postdeploy`-Dateien ist es das aktuelle Anwendungsverzeichnis. Wenn eine der Dateien fehlschlägt (mit einem Exit-Code ungleich Null beendet wird), wird die Bereitstellung abgebrochen und schlägt fehl.

Hook-Dateien können auf alle Umgebungseigenschaften zugreifen, die Sie in den Anwendungsoptionen definiert haben, sowie auf die Systemumgebungsvariablen `HOME`, `PATH` und `PORT`.

Um die Werte von Umgebungsvariablen und anderen Konfigurationsoptionen für Ihre Skripts für Platform-Hooks abzurufen, können Sie das `get-config`-Dienstprogramm verwenden, das Elastic Beanstalk auf Umgebungs-Instances verwendet. Details dazu finden Sie unter [the section called “Platform-Skript-Tools” \(p. 43\)](#).

Konfigurationsdateien

Sie können [Konfigurationsdateien \(p. 722\)](#) zum Verzeichnis `.ebextensions` des Quellcodes Ihrer Anwendung hinzufügen, um verschiedene Aspekte Ihrer Elastic Beanstalk-Umgebung zu konfigurieren. Mit Konfigurationsdateien können Sie unter anderem Software und andere Dateien auf den Instances Ihrer Umgebung anpassen und Initialisierungsbefehle für die Instances ausführen. Weitere Informationen finden Sie unter [the section called “Linux-Server” \(p. 725\)](#).

Sie können [Konfigurationsoptionen \(p. 646\)](#) auch mithilfe von Konfigurationsdateien festlegen. Viele der Optionen steuern das Plattformverhalten, und einige dieser Optionen sind [plattformspezifisch \(p. 712\)](#).

Auf Amazon Linux 2-Plattformen empfehlen wir die Verwendung von Buildfile, Procfile und Platform-Hooks zum Konfigurieren und Ausführen von benutzerdefiniertem Code auf Ihren Umgebungs-Instances während der Instance-Bereitstellung. Diese Mechanismen werden in den vorherigen Abschnitten auf dieser Seite beschrieben. Sie können in `.ebextensions`-Konfigurationsdateien weiterhin Befehle und Containerbefehle verwenden, aber sie sind nicht mehr so einfach in der Verwendung. Beispielsweise kann das Schreiben von Befehlsskripts in einer YAML-Datei vom Syntaxstandpunkt her eine Herausforderung darstellen. Sie müssen weiterhin `.ebextensions`-Konfigurationsdateien für jedes Skript verwenden, das einen Verweis auf eine AWS CloudFormation-Ressource benötigt.

Reverse-Proxy-Konfiguration

Alle Amazon Linux 2-Plattformversionen verwenden nginx als standardmäßigen Reverse-Proxy-Server. Die Plattformen Tomcat, Node.js, PHP und Python unterstützen ebenfalls Apache HTTPD als Alternative. Um Apache auf diesen Plattformen auszuwählen, legen Sie die Option `ProxyServer` im Namespace `aws:elasticbeanstalk:environment:proxy` auf `apache` fest. Alle Plattformen ermöglichen eine einheitliche Konfiguration des Proxy-Servers wie in diesem Abschnitt beschrieben.

Note

Im Fall von Amazon Linux AMI-Plattformversionen (vor Amazon Linux 2) müssen Sie Proxy-Server möglicherweise auf eine andere Art konfigurieren. Diese Legacy-Details finden Sie in den [jeweiligen Plattform-Themen \(p. 27\)](#) in diesem Handbuch.

Elastic Beanstalk konfiguriert den Proxyserver auf den Instances Ihrer Umgebung, um Webdatenverkehr an die Hauptwebanwendung über die Stamm-URL der Umgebung weiterzuleiten, z. B. `http://my-env.elasticbeanstalk.com`.

Standardmäßig wird der Proxy von Elastic Beanstalk so konfiguriert, dass alle Anforderungen an Port 80 an Ihre primäre Webanwendung unter Port 5000 weitergeleitet werden. Sie können diese Portnummer konfigurieren, indem Sie die `PORT`-Umgebungseigenschaft mithilfe des Namespace `aws:elasticbeanstalk:application:environment` (p. 684) in einer Konfigurationsdatei festlegen, wie im folgenden Beispiel gezeigt.

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

Weitere Informationen zum Festlegen von Umgebungsvariablen für Ihre Anwendung finden Sie unter [the section called “Optionseinstellungen”](#) (p. 724).

Ihre Anwendung sollte den Port überwachen, der dafür im Proxy konfiguriert ist. Wenn Sie den Standardport mithilfe der Umgebungseigenschaft `PORT` ändern, kann der Code durch Lesen des Wertes der Umgebungsvariablen `PORT` darauf zugreifen. Rufen Sie beispielsweise `os.Getenv("PORT")` in Go oder `System.getenv("PORT")` in Java auf. Wenn Sie Ihren Proxy so konfigurieren, dass Datenverkehr an mehrere Anwendungsprozesse gesendet wird, können Sie mehrere Umgebungseigenschaften konfigurieren und deren Werte sowohl in der Proxykonfiguration als auch in Ihrem Anwendungscode verwenden. Eine weitere Möglichkeit besteht darin, den Port-Wert als Befehlsargument in der `Procfile` an den Prozess zu übergeben. Weitere Informationen dazu erhalten Sie, indem Sie den Abschnitt `Buildfile` und `Procfile` auf dieser Seite erweitern.

Konfigurieren von nginx

Elastic Beanstalk verwendet nginx als Standard-Reverse-Proxy, um die Anwendung dem Elastic Load Balancing-Load Balancer zuzuweisen. Elastic Beanstalk bietet eine nginx-Standardkonfiguration, die Sie erweitern oder gänzlich mit einer eigenen Konfiguration überschreiben können.

Note

Wenn Sie eine `.conf`-Konfigurationsdatei von nginx hinzufügen oder bearbeiten, müssen Sie sie als UTF-8 codieren.

Um die nginx-Standardkonfiguration von Elastic Beanstalk zu erweitern, fügen Sie `.conf`-Konfigurationsdateien zum Ordner `.platform/nginx/conf.d` im Quell-Bundle der Anwendung hinzu. Die nginx-Konfiguration von Elastic Beanstalk fügt `.conf`-Dateien automatisch zu diesem Ordner hinzu.

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

Um die nginx-Standardkonfiguration von Elastic Beanstalk vollständig zu überschreiben, fügen Sie bei `.platform/nginx/nginx.conf` eine Konfiguration in das Quellpaket ein.

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

Wenn Sie die nginx-Konfiguration von Elastic Beanstalk überschreiben, fügen Sie folgende Zeile zu `nginx.conf` hinzu, damit die Elastic Beanstalk-Konfigurationen für [Erweiterte Zustandsberichte und Überwachung](#) (p. 824), automatische Anwendungszuweisungen und statische Dateien einbezogen werden.

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

Konfigurieren von Apache HTTPD

Die Plattformen Tomcat, Node.js, PHP und Python ermöglichen Ihnen die Auswahl des Apache-HTTPD-Proxy-Servers als Alternative zu nginx. Dies ist nicht die Standardoption. Im folgenden Beispiel wird Elastic Beanstalk für die Verwendung von Apache HTTPD konfiguriert.

Example .ebextensions/httpd-proxy.config

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

Sie können die Elastic Beanstalk-Apache-Standardkonfiguration mit Ihren zusätzlichen Konfigurationsdateien erweitern. Alternativ können Sie die Apache-Standardkonfiguration von Elastic Beanstalk vollständig überschreiben.

Um die Apache-Standardkonfiguration von Elastic Beanstalk zu erweitern, fügen Sie .conf-Konfigurationsdateien zum Ordner `.platform/httpd/conf.d` im Quell-Bundle der Anwendung hinzu. Die Apache-Konfiguration von Elastic Beanstalk fügt .conf-Dateien automatisch zu diesem Ordner hinzu.

```
~/workspace/my-app/  
|-- .ebextensions  
|  -- httpd-proxy.config  
|-- .platform  
|  -- httpd  
|    -- conf.d  
|      -- port5000.conf  
|      -- ssl.conf  
-- index.jsp
```

Beispielsweise wird mit der folgenden Apache 2.4-Konfiguration ein Listener auf Port 5000 hinzugefügt:

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>
```

Um die Apache-Standardkonfiguration von Elastic Beanstalk vollständig zu überschreiben, fügen Sie bei `.platform/httpd/conf/httpd.conf` eine Konfiguration in das Quellpaket ein.

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

Wenn Sie die Apache-Konfiguration von Elastic Beanstalk überschreiben, fügen Sie folgende Zeilen zu `httpd.conf` hinzu, damit die Elastic Beanstalk-Konfigurationen für [Erweiterte Zustandsberichte und Überwachung \(p. 824\)](#), automatische Anwendungszuweisungen und statische Dateien einbezogen werden.

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

Wenn Sie Ihre Elastic Beanstalk-Anwendung auf eine Amazon Linux 2-Plattform migrieren, lesen Sie auch die Informationen unter [the section called “Aktualisieren auf Amazon Linux 2” \(p. 503\)](#).

Themen

- [Anwendungsbeispiel mit Erweiterungen \(p. 40\)](#)
- [Instance-Bereitstellungs-Workflow \(p. 41\)](#)
- [Plattform-Skript-Tools \(p. 43\)](#)

Anwendungsbeispiel mit Erweiterungen

Im folgenden Beispiel wird ein Anwendungs-Quell-Bundle mit mehreren Erweiterbarkeitsfunktionen veranschaulicht, die von Elastic Beanstalk Amazon Linux 2-Plattformen unterstützt werden: eine `Procfile`, `.ebextensions`-Konfigurationsdateien, benutzerdefinierte Hooks und Proxy-Konfigurationsdateien.

```
~/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

Einige dieser Erweiterungen werden auf Amazon Linux AMI-Plattformversionen (vor Amazon Linux 2) nicht unterstützt.

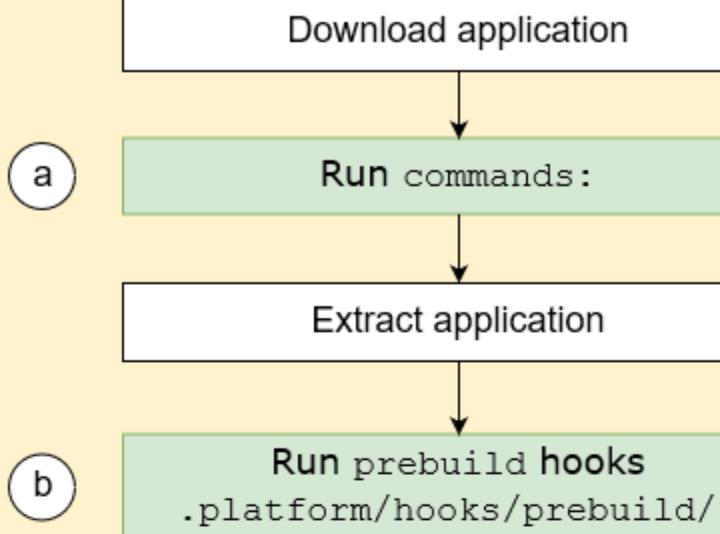
Instance-Bereitstellungs-Workflow

Da die Plattform Ihrer Umgebung auf viele Arten erweitert werden kann, ist es hilfreich, den Ablauf zu verstehen, wenn Elastic Beanstalk eine Instance bereitstellt oder eine Bereitstellung für eine Instance ausführt. Das folgende Diagramm zeigt diesen gesamten Bereitstellungsworkflow. Es zeigt die verschiedenen Phasen in einer Bereitstellung und die Schritte, die von Elastic Beanstalk in jeder Phase durchgeführt werden.

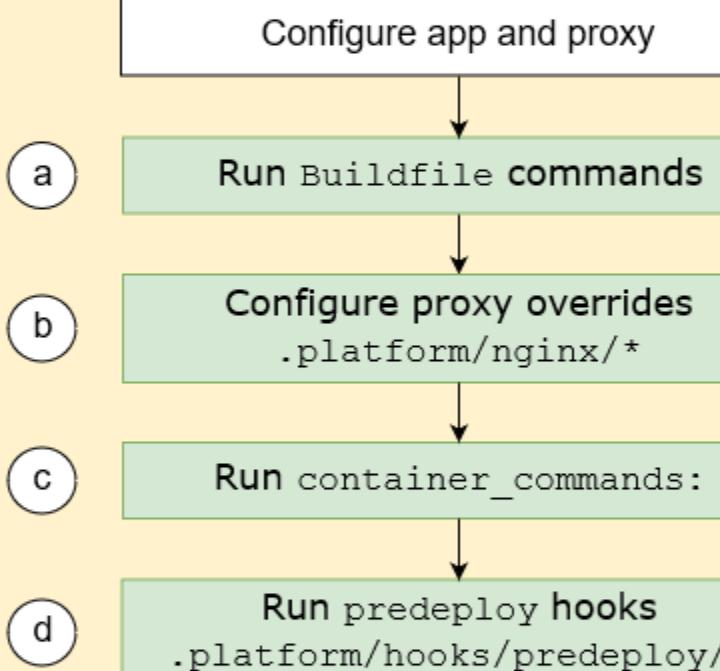
Hinweise

- Das Diagramm zeigt nicht den vollständigen Satz von Schritten, die Elastic Beanstalk während der Bereitstellung auf Umgebungs-Instances ausführt. Wir stellen dieses Diagramm zur Veranschaulichung bereit, um Ihnen die Reihenfolge und den Kontext für die Ausführung Ihrer Anpassungen zu liefern.
- Der Einfachheit halber zeigt das Diagramm nur die `.platform/hooks/*-Hook`-Unterverzeichnisse (für Anwendungsbereitstellungen) und nicht die `.platform/confighooks/*-Hook`-Unterverzeichnisse (für Konfigurationsbereitstellungen). Hooks in den zuletzt genannten Unterverzeichnissen werden während der gleichen Schritte ausgeführt wie die Hooks in den Unterverzeichnissen, die im Diagramm gezeigt werden.

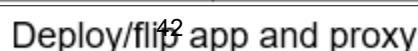
1. Initial steps



2. Configure



3. Deploy



In der folgenden Liste werden die Bereitstellungsphasen und -schritte beschrieben.

1. Anfängliche Schritte

Elastic Beanstalk lädt und extrahiert Ihre Anwendung. Nach jedem dieser Schritte führt Elastic Beanstalk einen Erweiterbarkeitsschritt aus.

- a. Führt Befehle aus, die im Abschnitt [Befehle: \(p. 731\)](#) einer beliebigen Konfigurationsdatei gefunden werden.
- b. Führt alle ausführbaren Dateien im Verzeichnis `.platform/hooks/prebuild` des Quellpaket aus (`.platform/confighooks/prebuild` für eine Konfigurationsbereitstellung).

2. Konfiguration

Elastic Beanstalk konfiguriert Ihre Anwendung und den Proxy-Server.

- a. Führt die Befehle aus, die in der `Buildfile` in Ihrem Quell-Bundle gefunden wurden.
- b. Kopiert Ihre benutzerdefinierten Proxy-Konfigurationsdateien, sofern sich welche im Verzeichnis `.platform/nginx` Ihres Quell-Bundles befinden, an ihrem Laufzeitspeicherort.
- c. Führt Befehle aus, die im Abschnitt [container_commands: \(p. 734\)](#) einer beliebigen Konfigurationsdatei gefunden wurden.
- d. Führt alle ausführbaren Dateien im Verzeichnis `.platform/hooks/predeploy` des Quellpaket aus (`.platform/confighooks/predeploy` für eine Konfigurationsbereitstellung).

3. Bereitstellen

Elastic Beanstalk stellt Ihre Anwendung und den Proxy-Server bereit und führt sie aus.

- a. Führt den Befehl aus, der in der `Procfile`-Datei in Ihrem Quell-Bundle gefunden wurde.
- b. Führt den Proxy-Server mit den benutzerdefinierten Proxy-Konfigurationsdateien aus, falls vorhanden, oder erneut aus.
- c. Führt alle ausführbaren Dateien im Verzeichnis `.platform/hooks/postdeploy` des Quellpaket aus (`.platform/confighooks/postdeploy` für eine Konfigurationsbereitstellung).

Plattform-Skript-Tools

In diesem Thema werden Tools beschrieben, die AWS Elastic Beanstalk für Instances von Umgebungen bereitstellt, die Amazon Linux-Plattformversionen verwenden. Sie können diese Tools verwenden, um Plattform-Hook-Skripte zu verbessern, die auf Instances in Ihrer Umgebung ausgeführt werden.

get-config

Verwenden Sie das Tool `get-config`, um Umgebungsvariablenwerte und andere Plattform- und Instance-Informationen abzurufen. Das Tool steht unter `/opt/elasticbeanstalk/bin/get-config` zur Verfügung.

get-config-Befehle

Jeder `get-config`-Tool-Befehl gibt einen bestimmten Typ von Informationen zurück. Verwenden Sie die folgende Syntax, um einen der Befehle des Tools auszuführen.

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

Im folgenden Beispiel wird der Befehl `environment` ausgeführt:

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

Abhängig von dem gewählten Befehl und den gewählten Optionen gibt das Tool ein Objekt (JSON oder YAML) mit Schlüssel-Wert-Paaren oder einem einzelnen Wert zurück.

Sie können `get-config` testen, indem Sie SSH verwenden, um eine Verbindung zu einer Instance in Ihrer Elastic Beanstalk-Umgebung herzustellen.

Note

Wenn Sie `get-config` zum Testen ausführen, benötigen einige Befehle möglicherweise Stammbenutzerrechte, um auf die zugrunde liegenden Informationen zuzugreifen. Wenn Sie einen Zugriffsberechtigungsfehler erhalten, führen Sie den Befehl unter `sudo` erneut aus. Sie müssen `sudo` nicht hinzufügen, wenn Sie das Tool in den Skripts verwenden, die Sie in Ihrer Umgebung bereitstellen. Elastic Beanstalk führt alle Ihre Skripte als Stammbenutzer aus.

In den folgenden Abschnitten werden die Befehle des Tools beschrieben.

optionsettings – Konfigurationsoptionen

Der Befehl `get-config optionsettings` gibt ein Objekt zurück, das die Konfigurationsoptionen auflistet, die für die Umgebung festgelegt und von der Plattform auf Umgebungs-Instances verwendet werden. Sie sind nach Namespace organisiert.

```
$ /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"}}
}
```

Um einen bestimmten Wert für die Konfigurationsoption zurückzugeben, verwenden Sie die Option `--namespace (-n)`, um einen Namespace anzugeben, und die Option `--option-name (-o)`, um einen Optionsnamen anzugeben.

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

environment – Umgebungseigenschaften

Der Befehl `get-config environment` gibt ein Objekt zurück, das eine Liste der Umgebungseigenschaften enthält. Dazu gehören sowohl benutzerkonfigurierte als auch die von Elastic Beanstalk bereitgestellten Eigenschaften.

```
$ /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": "testpassword"
}
```

Beispielsweise bietet Elastic Beanstalk Umgebungseigenschaften, mit denen die Verbindung zu einer integrierten Amazon RDS-DB-Instance (`RDS_HOSTNAME` usw.) hergestellt werden kann. Diese RDS-Verbindungseigenschaften werden in der Ausgabe von `get-config environment` angezeigt, aber nicht in der Ausgabe von `get-config optionsettings`, da sie in den Konfigurationsoptionen nicht festgelegt wurden.

Wenn Sie eine bestimmte Umgebungseigenschaft zurückzugeben möchten, verwenden Sie die Option `--key (-k)`, um einen Eigenschaftsschlüssel anzugeben.

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

container – Konfigurationswerte auf der Instance

Der Befehl `get-config container` gibt ein Objekt zurück, das Plattform- und Umgebungskonfigurationswerte auflistet, wie sie in Umgebungs-Instances reflektiert sind.

Das folgende Beispiel zeigt die Ausgabe des Befehls in einer Amazon Linux 2-Tomcat-Umgebung.

```
$ /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"}
```

Um den Wert eines bestimmten Schlüssels zurückzugeben, verwenden Sie die Option `--key (-k)`, um den Schlüssel anzugeben.

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

addons – Add-On-Konfigurationswerte

Der Befehl `get-config addons` gibt ein Objekt zurück, das Konfigurationsinformationen von Umgebungs-Add-Ons enthält. Verwenden Sie ihn, um die Konfiguration einer Amazon RDS-Datenbank abzurufen, die der Umgebung zugeordnet ist.

```
$ /opt/elasticbeanstalk/bin/get-config addons
{"rds": {"Description": "RDS Environment variables", "env": {
    "RDS_DB_NAME": "ebdb", "RDS_HOSTNAME": "ea13k2wimuldh8i.c18mnpu5rwvg.us-
    east-2.rds.amazonaws.com", "RDS_PASSWORD": "password", "RDS_PORT": "3306", "RDS_USERNAME": "user" }}}}
```

Sie können das Ergebnis auf zwei Arten einschränken. Um Werte für ein bestimmtes Add-On abzurufen, geben Sie mit der Option `--add-on (-a)` den Namen des Add-On an.

```
$ /opt/elasticbeanstalk/bin/get-config addons -a rds
{"Description": "RDS Environment variables", "env": {
    "RDS_DB_NAME": "ebdb", "RDS_HOSTNAME": "ea13k2wimuldh8i.c18mnpu5rwvg.us-
    east-2.rds.amazonaws.com", "RDS_PASSWORD": "password", "RDS_PORT": "3306", "RDS_USERNAME": "user" }}}}
```

Um den Wert eines bestimmten Schlüssels in einem Add-On zurückzugeben, fügen Sie die Option `--key (-k)` hinzu, um den Schlüssel anzugeben.

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

platformconfig – Konstante Konfigurationswerte

Der Befehl `get-config platformconfig` gibt ein Objekt zurück, das Plattformkonfigurationsinformationen enthält, die mit der Plattformversion konstant sind. Die Ausgabe ist in allen Umgebungen mit derselben Plattformversion gleich. Das Ausgabeobjekt des Befehls verfügt über zwei eingebettete Objekte:

- `GeneralConfig` – Enthält Informationen, die in den neuesten Versionen aller Amazon Linux 2-Plattformzweige konstant sind.
- `PlatformSpecificConfig` – Enthält Informationen, die für die Plattformversion konstant sind und für diese spezifisch sind.

Das folgende Beispiel zeigt die Ausgabe des Befehls in einer Umgebung, die den Corretto 11-Plattformzweig Tomcat 8.5 verwendet.

```
$ /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"}}
```

Um den Wert eines bestimmten Schlüssels zurückzugeben, verwenden Sie die Option `--key (-k)`, um den Schlüssel anzugeben. Diese Schlüssel sind in den beiden eingebetteten Objekten eindeutig. Sie müssen das Objekt, das den Schlüssel enthält, nicht angeben.

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

get-config-Ausgabeoptionen

Verwenden Sie die Option `--output`, um das Format des Ausgabeobjekts anzugeben. Gültige Werte sind `JSON` (Standard) und `YAML`. Dies ist eine globale Option und Sie müssen sie vor dem Befehlsnamen angeben.

Im folgenden Beispiel werden Werte für Konfigurationsoptionen im `YAML`-Format zurückgegeben.

```
$ /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"
```

download-source-bundle (nur Amazon Linux AMI)

Auf Amazon Linux AMI-Plattformzweigen (vor Amazon Linux 2) bietet Elastic Beanstalk ein zusätzliches Tool, `download-source-bundle`. Verwenden Sie dieses, um den Quellcode Ihrer Anwendung während der Bereitstellung Ihrer Plattform herunterzuladen. Das Tool steht unter `/opt/elasticbeanstalk/bin/download-source-bundle` zur Verfügung.

Das Beispielskript `00-unzip.sh` befindet sich auf Umgebungs-Instances im Ordner `appdeploy/pre`. Es demonstriert die Verwendung von `download-source-bundle`, um den Anwendungsquellcode während der Bereitstellung in den Ordner `/opt/elasticbeanstalk/deploy/appsource` herunterzuladen.

Bereitstellen von Elastic Beanstalk-Anwendungen aus Docker-Containern

Elastic Beanstalk unterstützt die Bereitstellung von Webanwendungen aus Docker-Containern. Mit Docker-Containern können Sie Ihre eigene Laufzeitumgebung definieren. Sie können auch Ihre eigene Plattform,

Programmiersprache und beliebige Anwendungsabhängigkeiten (z. B. Paketmanager oder Tools), auswählen, die in der Regel von anderen Plattformen nicht unterstützt werden. Docker-Container sind unabhängig und enthalten die Konfigurationsinformationen und Software, die Ihre Webanwendung für die Ausführung benötigt. Alle in der Elastic Beanstalk-Konsole definierten Umgebungsvariablen werden an die Container übergeben.

Durch die Verwendung von Docker mit Elastic Beanstalk erhalten Sie eine Infrastruktur, die sämtliche Details der Kapazitätsbereitstellung, Lastverteilung, Skalierung und Überwachung des Anwendungsstatus übernimmt. Sie können Ihre Webanwendung mühelos in einer Umgebung verwalten, die das Spektrum der in Elastic Beanstalk integrierten Services unterstützt. Zu diesen Umgebungen zählen unter anderem [VPC](#), [RDS](#) und [IAM](#). Weitere Informationen zu Docker, einschließlich einer Installationsanleitung sowie Informationen zur erforderlichen Software und zur Verwendung von Docker-Images zum Starten von Docker-Containern finden Sie unter [Docker: the Linux container engine](#).

In den Themen in diesem Kapitel wird davon ausgegangen, dass Sie über einige Kenntnisse der Elastic Beanstalk-Umgebungen verfügen. Wenn Sie Elastic Beanstalk noch nicht verwendet haben, machen Sie sich im [Tutorial "Erste Schritte" \(p. 3\)](#) mit den Grundlagen vertraut.

Docker-Plattformfamilie

Die Docker-Plattformfamilie für Elastic Beanstalk umfasst mehrere Plattformen. Die auf Amazon Linux 2 ausgeführte Docker-Plattform bietet die meisten Vorteile (beispielsweise langfristige Unterstützung). In den folgenden Abschnitten werden die von Elastic Beanstalk angebotenen Docker-Plattformen sowie die empfohlenen Migrationspfade zu Amazon Linux 2 beschrieben.

Weitere Informationen zu unterstützten Plattformversionen für die jeweilige Docker-Plattform finden Sie auf der Seite [Unterstützte Plattformen](#) im Dokument AWS Elastic Beanstalk-Plattformen.

Die Docker-Plattform

Elastic Beanstalk kann ein Docker-Image und einen Quellcode für EC2-Instances bereitstellen, von denen die Elastic Beanstalk-Docker-Plattform ausgeführt wird. Die Plattform unterstützt mehrere Container (und Einzelcontainer). Sie können auch das Docker Compose-Tool auf der Docker-Plattform nutzen, um das Konfigurieren, Testen und Bereitstellen von Anwendungen zu vereinfachen.

Diese Amazon Linux 2 Docker-Plattform bietet folgende Vorteile:

- Langfristige Unterstützung. Die auf Amazon Linux 2 basierende Docker-Plattform zeichnet sich durch langfristige Unterstützung mit Sicherheits- und Funktionsupdates aus.
- Docker Compose-Funktionen. Diese Plattform ermöglicht die Nutzung der Funktionen des Docker Compose-Tools, um mehrere Container zu definieren und auszuführen. Sie können die Datei docker-compose.yml zur Bereitstellung in Elastic Beanstalk einschließen.
- Verwenden von Anwendungs-Images aus öffentlichen oder privaten Repositorys Elastic Beanstalk ruft die Docker Compose-Befehlszeilenschnittstelle auf verarbeitet die Datei docker-compose.yml, um die Anwendungsimages abzurufen und als containerisierte Anwendungen auszuführen.
- Erstellen von Containerimages während der Bereitstellung. Sie müssen Ihre Anwendungsimages nicht vorab erstellen, bevor Sie sie zur Ausführung als Container bereitstellen. Während der Bereitstellung können Sie die Containerimages von Grund auf neu erstellen, indem Sie Abhängigkeiten in der Dockerfile angeben.

Weitere Informationen zu Beispielen sowie Unterstützung bei den ersten Schritten in einer Docker-Umgebung finden Sie unter [the section called “Die Docker-Plattform” \(p. 48\)](#). Weitere Informationen zu den Containerdefinitionsformaten und zu ihrer Verwendung finden Sie unter [the section called “Docker-Konfiguration” \(p. 53\)](#).

Die folgenden Abschnitte sind für Elastic Beanstalk-Docker-Umgebungen relevant, von denen die ältere Amazon Linux-AMI-Plattformversion (Vorgängerversion von Amazon Linux 2) verwendet wird.

Docker (Amazon Linux AMI)

Die Amazon Linux AMI-basierte Docker-Plattform kann verwendet werden, um ein (in einer Dockerfile oder in einer `Dockerrun.aws.json`-Definition beschriebenes) Docker-Image und Quellcode für EC2-Instances bereitzustellen, die in einer Elastic Beanstalk-Umgebung ausgeführt werden. Von dieser Docker-Plattform wird pro Instance nur jeweils ein Container ausgeführt.

Beispiele und Unterstützung bei den ersten Schritten mit einer Docker-Umgebung finden Sie unter [the section called “Die Docker-Plattform” \(p. 48\)](#). Weitere Informationen zu den Containerdefinitionsformaten und zu ihrer Verwendung finden Sie unter [the section called “Docker-Konfiguration” \(p. 53\)](#).

Multicontainer-Docker (Amazon Linux AMI)

Note

Diese Plattform unterstützt nur das Amazon Linux AMI-Betriebssystem (Vorgängerversion von Amazon Linux 2). Die [Docker \(p. 48\)](#)-Plattform bietet Multicontainer-Docker-Funktionalität mit Amazon Linux 2..

Die andere generische Plattform (Multicontainer-Docker) verwendet Amazon Elastic Container Service (Amazon ECS), um die Bereitstellung mehrerer Docker-Container in einem Amazon ECS-Cluster in einer Elastic Beanstalk-Umgebung zu koordinieren. Die Instances in der Umgebung werden jeweils im gleichen Satz von Containern ausgeführt, die in einer `Dockerrun.aws.json`-Datei definiert sind. Wenn von Ihrer Elastic Beanstalk-Umgebung eine Amazon Linux AMI-Plattformversion (Vorgängerversion von Amazon Linux 2) verwendet wird, verwenden Sie die Multicontainer-Plattform, um mehrere Docker-Container für die einzelnen Instances bereitzustellen.

Weitere Informationen zur Multicontainer-Docker-Plattform und zu ihrer Verwendung finden Sie unter [Verwenden der Multicontainer-Docker-Plattform \(Amazon Linux AMI\) \(p. 64\)](#). Im Thema [Multicontainer-Docker-Konfiguration \(p. 69\)](#) finden Sie Informationen zur Version 2 des Formats `Dockerrun.aws.json`. Diese ist zwar mit der Version vergleichbar, die bei der Docker-Plattform zum Einsatz kommt, aber nicht mit ihr kompatibel. Es steht auch ein [Tutorial \(p. 73\)](#) zur Verfügung, das Sie durch die gesamte Bereitstellung einer neuen Multicontainerumgebung führt. Die beschriebene Umgebung führt eine PHP-Website mit einem NGINX-Proxy aus, der in einem separaten Container davor ausgeführt wird.

Vorkonfigurierte Docker-Container

Zusätzlich zu den beiden generischen Docker-Plattformen gibt es mehrere vorkonfigurierte Docker-Plattformversionen, die Sie verwenden können, um Ihre Anwendung in einem von vielen gängigen Softwarestapeln wie Java mit GlassFish oder Python mit uWSGI auszuführen. Verwenden Sie einen vorkonfigurierten Container, wenn er zu der von Ihrer Anwendung verwendeten Software passt.

Note

Alle Zweige der vorkonfigurierten Docker-Plattform verwenden das Amazon Linux AMI-Betriebssystem (Vorgängerversion von Amazon Linux 2). Verwenden Sie zum Migrieren Ihrer GlassFish-Anwendung zu Amazon Linux 2 die generische Docker-Plattform und stellen Sie GlassFish und Ihren Anwendungscode in einem Amazon Linux 2-Docker-Image bereit. Weitere Informationen hierzu finden Sie unter [the section called “Tutorial: GlassFish auf Docker: Pfad zu AL2” \(p. 61\)](#).

Weitere Informationen finden Sie unter [Vorkonfigurierte Docker-Container \(p. 82\)](#).

Verwenden der Docker-Plattform

Important

Amazon Linux 2-Plattformversionen unterscheiden sich grundlegend von Amazon Linux AMI-Plattformversionen (Vorgängerversion von Amazon Linux 2). Diese unterschiedlichen

Plattformgenerationen sind auf verschiedene Arten inkompatibel. Wenn Sie zu einer Amazon Linux 2-Plattformversion migrieren, müssen Sie die Informationen in [the section called "Aktualisieren auf Amazon Linux 2" \(p. 503\)](#) lesen.

AWS Elastic Beanstalk kann Docker-Umgebungen starten, indem ein in einer `Dockerfile` beschriebenes Image erstellt oder ein Remote-Docker-Image abgerufen wird. Wenn Sie ein Remote-Docker-Image bereitstellen, müssen Sie keine `Dockerfile` einschließen. Falls Sie auch Docker Compose nutzen, verwenden Sie stattdessen eine `docker-compose.yml`-Datei, die das zu verwendende Image und zusätzliche Konfigurationsoptionen angibt. Wenn Sie Docker Compose in Ihren Docker-Umgebungen nicht nutzen, verwenden Sie stattdessen eine `Dockerrun.aws.json`-Datei.

Themen

- [Voraussetzungen \(p. 49\)](#)
- [Containerisieren einer Elastic Beanstalk-Anwendung \(p. 49\)](#)
- [Lokales Testen eines Containers \(p. 50\)](#)
- [Bereitstellen eines Containers mit einer Dockerfile-Datei \(p. 51\)](#)
- [Testen eines Remote-Docker-Images \(p. 51\)](#)
- [Bereitstellen eines Remote-Docker-Images in Elastic Beanstalk \(p. 52\)](#)
- [Bereinigen \(p. 53\)](#)
- [Docker-Konfiguration \(p. 53\)](#)
- [Bereitstellen einer GlassFish-Anwendung auf der Docker-Plattform: Migrationspfad zu Amazon Linux 2 \(p. 61\)](#)

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie über Kenntnisse zu den grundlegenden Elastic Beanstalk-Operationen, [Verwenden des Elastic Beanstalk Command Line Interface \(EB CLI\) \(p. 1008\)](#) und Docker verfügen. Damit Sie die Schritte in diesem Tutorial ausführen können, müssen Sie über eine funktionierende lokale Installation von Docker verfügen. Weitere Informationen über die Installation von Docker finden Sie im [Docker-Installationshandbuch](#).

Sofern noch nicht geschehen, befolgen Sie die Anweisungen in [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#), um Ihre erste Elastic Beanstalk-Umgebung zu starten. Dieses Tutorial verwendet die EB-CLI, aber Sie können auch mit der Elastic Beanstalk-Konsole Umgebungen erstellen und Anwendungen hochladen. Weitere Informationen zum Konfigurieren von Docker-Umgebungen finden Sie unter [Docker-Konfiguration \(p. 53\)](#).

Containerisieren einer Elastic Beanstalk-Anwendung

In diesem Beispiel erstellen Sie ein Docker-Image der Flask-Beispielanwendung aus [Bereitstellen einer Flask-Anwendung in Elastic Beanstalk \(p. 361\)](#). Die Anwendung besteht aus einer Hauptdatei `application.py`. Wir benötigen außerdem eine `Dockerfile`. Platzieren Sie beide Dateien im Stammverzeichnis.

```
~/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"]
```

Lokales Testen eines Containers

Verwenden Sie die Elastic Beanstalk CLI (EB CLI) zum Konfigurieren Ihres lokalen Repositorys für die Bereitstellung in Elastic Beanstalk. Legen Sie die `Dockerfile` Ihrer Anwendung im Stammverzeichnis fest.

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

(Optional) Verwenden Sie den Befehl `eb local run` für die lokale Erstellung und Ausführung Ihres Containers.

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

Note

Weitere Informationen zum `eb local`-Befehl finden Sie unter [the section called “eb local” \(p. 1071\)](#). Der Befehl wird unter Windows nicht unterstützt. Alternativ können Sie Ihre Container mit den Befehlen `docker build` und `docker run` erstellen und ausführen. Weitere Informationen finden Sie in der [Docker-Dokumentation](#).

(Optional) Wenn Ihr Container ausgeführt wird, verwenden Sie den eb local open-Befehl, um Ihre Anwendung in einem Webbrower anzeigen. Alternativ öffnen Sie <http://localhost:5000/> in einem Webbrower.

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

Bereitstellen eines Containers mit einer Dockerfile-Datei

Nachdem Sie Ihre Anwendung lokal getestet haben, stellen Sie sie in einer Elastic Beanstalk-Umgebung bereit. Elastic Beanstalk verwendet die Anweisungen in Ihrem Dockerfile, um das Image zu erstellen und auszuführen.

Verwenden Sie den Befehl eb create, um eine Umgebung zu erstellen und die Anwendung bereitzustellen.

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

Verwenden Sie nach dem Starten der Umgebung den Befehl eb open, um sie in einem Webbrower anzeigen.

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

Testen eines Remote-Docker-Images

Als Nächstes erstellen wir ein Docker-Image der Flask-Anwendung aus dem vorherigen Abschnitt und übertragen es in den Docker Hub.

Note

In den folgenden Schritten erstellen Sie ein öffentlich verfügbares Docker-Image.

Nachdem das Image erstellt und gepusht wurde, kann es in Elastic Beanstalk mit einer docker-compose.yml-Datei bereitgestellt werden, wenn Sie Docker Compose in Ihrer Docker-Umgebung verwenden. (Falls Sie Docker Compose in Ihrer Docker-Umgebung nicht nutzen, verwenden Sie stattdessen eine Dockerrun.aws.json-Datei.) Um ein Docker-Image der Flask-Anwendung zu erstellen und in den Docker Hub zu übertragen, führen Sie die folgenden Befehle aus: Wir verwenden dasselbe Verzeichnis aus dem vorherigen Beispiel, aber Sie können jedes Verzeichnis mit dem Code Ihrer Anwendung verwenden.

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

Note

Bevor Sie Ihr Image übertragen, müssen Sie möglicherweise docker login ausführen.

Falls Sie Ihre Docker-Umgebung mithilfe des Docker Compose-Tools verwalten, haben Sie nun die Möglichkeit, Ihre Anwendung nur mit einer docker-compose.yml-Datei bereitzustellen. Weitere Informationen zu docker-compose.yml-Dateien finden Sie unter [Docker-Konfiguration \(p. 53\)](#).

Falls Sie Docker Compose nicht nutzen, verwenden Sie stattdessen eine Dockerrun.aws.json-Datei. Weitere Informationen finden Sie unter [Bereitstellen mit Version 1 von „Dockerrun.aws.json“ \(ohne Docker Compose\) \(p. 52\)](#).

Erstellen Sie ein neues Verzeichnis und eine docker-compose.yml-Datei.

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

```
version: '3.8'
```

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

Bereitstellen mit Version 1 von „Dockerrun.aws.json“ (ohne Docker Compose)

Falls Sie Ihre Docker-Umgebung nicht mithilfe des Docker Compose-Tools verwalten, haben Sie nun die Möglichkeit, Ihre Anwendung nur mit einer `Dockerrun.aws.json`-Datei bereitzustellen. Weitere Informationen zu `Dockerrun.aws.json`-Dateien finden Sie unter [Konfiguration für Docker-Plattformen \(ohne Docker Compose\) \(p. 57\)](#).

Erstellen Sie ein neues Verzeichnis und eine `Dockerrun.aws.json`-Datei.

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

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

Verwenden Sie die EB CLI zum Konfigurieren Ihres lokalen Repositorys für die Bereitstellung in Elastic Beanstalk.

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

(Optional) Verwenden Sie `eb local run` für die lokale Erstellung und Ausführung Ihres Containers. Weitere Informationen zum `eb local`-Befehl finden Sie unter [eb local \(p. 1071\)](#).

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

(Optional) Wenn Ihr Container ausgeführt wird, verwenden Sie den `eb local open`-Befehl, um Ihre Anwendung in einem Webbrowser anzeigen. Alternativ öffnen Sie <http://localhost:5000/> in einem Webbrowser.

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

Bereitstellen eines Remote-Docker-Images in Elastic Beanstalk

Nachdem Sie Ihren Container lokal getestet haben, stellen Sie ihn in einer Elastic Beanstalk-Umgebung bereit. Elastic Beanstalk verwendet die `docker-compose.yml`-Datei, um Ihr Image abzurufen und auszuführen, wenn Sie Docker Compose verwenden. Ansonsten verwendet Elastic Beanstalk stattdessen `Dockerrun.aws.json`.

Verwenden Sie die EB CLI, um eine Umgebung zu erstellen und das Image bereitzustellen.

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

Sobald Ihre Umgebung gestartet wurde, verwenden Sie eb open, um sie in einem Webbrowser anzuzeigen.

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

Bereinigen

Falls Sie nicht länger mit Elastic Beanstalk arbeiten möchten, können Sie die Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 554\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie auf Environment actions (Umgebungsaktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Oder mit der EB CLI:

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

Docker-Konfiguration

In diesem Abschnitt wird beschrieben, wie Sie Ihr Docker-Image und Ihren Container für die Bereitstellung auf Elastic Beanstalk vorbereiten.

Docker-Umgebung mit Docker Compose

In diesem Abschnitt wird beschrieben, wie Sie Ihr Docker-Image und Ihren Container für die Bereitstellung auf Elastic Beanstalk vorbereiten. Jede Webanwendung, die Sie für Elastic Beanstalk in einer Docker-Umgebung bereitstellen, muss eine `docker-compose.yml`-Datei enthalten, wenn Sie auch das Docker Compose-Tool verwenden. Sie können Ihre Webanwendung als containerisierten Dienst für Elastic Beanstalk bereitstellen, indem Sie eine der folgenden Aktionen ausführen:

- Erstellen Sie eine `docker-compose.yml`-Datei, um ein Docker-Image aus einem gehosteten Repository für Elastic Beanstalk bereitzustellen. Wenn alle Ihre Bereitstellungen aus Images in öffentlichen Repositorys stammen, sind keine weiteren Dateien erforderlich. (Falls für Ihre Bereitstellung ein Image aus einem privaten Repository benötigt wird, müssen zusätzliche Konfigurationsdateien für die Authentifizierung eingeschlossen werden. Weitere Informationen finden Sie unter [Verwenden von Images aus einem privaten Repository \(p. 54\)](#).) Weitere Informationen zur `docker-compose.yml`-Datei finden Sie auf der Docker-Website in der [Referenz zu Compose-Dateien](#).
- Erstellen Sie eine `Dockerfile`-Datei, damit von Elastic Beanstalk ein benutzerdefiniertes Image erstellt und ausgeführt werden kann. Diese Datei ist optional und von Ihren Bereitstellungsanforderungen

abhängig. Weitere Informationen zur `Dockerfile`-Datei finden Sie auf der Docker-Website in der [Dockerfile-Referenz](#).

- Erstellen Sie eine `.zip`-Datei mit Ihren Anwendungsdateien, sämtlichen Abhängigkeiten der Anwendungsdatei, der `Dockerfile` und der `docker-compose.yml`-Datei. Wenn Sie die EB CLI verwenden, um Ihre Anwendung bereitzustellen, wird automatisch eine `.zip`-Datei erstellt. Die beiden Dateien müssen sich im Stammverzeichnis oder im oberste Level des `.zip`-Archivs befinden.

Wenn Sie nur eine `docker-compose.yml`-Datei verwenden, um Ihre Anwendung bereitzustellen, müssen Sie keine `.zip`-Datei erstellen.

Dieses Thema ist eine Syntaxreferenz. Ausführliche Verfahren zum Starten von Docker-Umgebungen mit Elastic Beanstalk finden Sie unter [Verwenden der Docker-Plattform \(p. 48\)](#).

Weitere Informationen zu Docker Compose und zur Installation von Docker Compose finden Sie auf den Docker-Websites [Overview of Docker Compose](#) und [Install Docker Compose](#).

Note

Wenn Sie Ihre Docker-Umgebungen nicht mit Docker Compose konfigurieren, sollten Sie die `docker-compose.yml`-Datei nicht verwenden. Verwenden Sie stattdessen die `Dockerrun.aws.json`-Datei, die `Dockerfile`-Datei oder beide.

Weitere Informationen finden Sie unter [the section called “Konfiguration für Docker-Plattformen \(ohne Docker Compose\) ” \(p. 57\)](#).

Verwenden von Images aus einem privaten Repository

Elastic Beanstalk muss sich bei der Onlineregistrierung authentifizieren, die als Host für das private Repository fungiert, um Ihre Images aus einem privaten Repository pullen und bereitzustellen zu können. Sie haben zwei Möglichkeiten, Anmeldeinformationen für Ihre Elastic Beanstalk-Umgebung zu speichern und abzurufen, um sich bei einem Repository zu authentifizieren.

- Der AWS Systems Manager (SSM) Parameter Store
- Die Datei `Dockerrun.aws.json v3`

Verwenden des AWS Systems Manager (SSM) Parameter Store

Sie können Elastic Beanstalk so konfigurieren, dass vor dem Start des Bereitstellungsprozesses eine Anmeldung bei Ihrem privaten Repository durchgeführt wird. Dadurch kann Elastic Beanstalk auf die Images aus dem Repository zugreifen und diese Images in Ihrer Elastic Beanstalk-Umgebung bereitzustellen.

Durch diese Konfiguration werden Ereignisse in der Prebuild-Phase des Bereitstellungsprozesses von Elastic Beanstalk initiiert. Verwenden Sie zur Einrichtung das Konfigurationsverzeichnis [.ebextensions \(p. 722\)](#). Von der Konfiguration werden Skripts vom Typ [Plattform-Hook \(p. 36\)](#) verwendet, die `docker login` aufrufen, um die Authentifizierung bei der Onlineregistrierung durchzuführen, die als Host für das private Repository fungiert. Diese Konfigurationsschritte werden im Anschluss detailliert beschrieben.

So konfigurieren Sie Elastic Beanstalk für die Authentifizierung bei Ihrem privaten Repository mit AWS SSM

Note

Sie müssen AWS Systems Manager einrichten, um diese Schritte ausführen zu können. Weitere Informationen finden Sie im [AWS Systems Manager-Benutzerhandbuch](#)

1. Erstellen Sie Ihre Verzeichnisstruktur für `.ebextensions` wie folgt:

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

2. Verwenden Sie den [AWS Systems Manager](#) Parameter Store, um die Anmeldeinformationen Ihres privaten Repositorys zu speichern, damit Elastic Beanstalk bei Bedarf Ihre Anmeldeinformationen abrufen kann. Führen Sie hierzu den Befehl [put-parameter](#) aus.

```
aws ssm put-parameter --name USER --type String --value "username"
aws ssm put-parameter --name PASSWD --type String --value "passwd"
```

3. Erstellen Sie die folgende Datei `env.config` und platzieren Sie sie im Verzeichnis `.ebextensions`, wie in der obigen Verzeichnisstruktur gezeigt.

Note

USER und PASSWD im Skript müssen den Zeichenfolgen entsprechen, die in den obigen Befehlen vom Typ `ssm put-parameter` verwendet wurden.

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    USER: '{{resolve:ssm:USER:1}}'
    PASSWD: '{{resolve:ssm:PASSWD:1}}'
```

4. Erstellen Sie die folgende Skriptdatei `01login.sh` und platzieren Sie sie in den folgenden Verzeichnissen, wie ebenfalls in der obigen Verzeichnisstruktur gezeigt:

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

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

Durch das Skript `01login.sh` wird zunächst das Tool `get-config` aufgerufen, um die Anmeldeinformationen für das Repository abzurufen. Anschließend wird `docker login` aufgerufen, um sich beim Repository zu authentifizieren.

Hinweise

- Alle Skriptdateien müssen über die Ausführungs berechtigung verfügen. Legen Sie mit `chmod +x` die Ausführungs berechtigung für Ihre Hook-Dateien fest.
- Bei Hook-Dateien kann es sich um Binär- oder Skriptdateien handeln, die mit einer `#!-` Zeile mit dem Interpreter-Pfad beginnen (z. B. `#!/bin/bash`).
- Weitere Informationen finden Sie in Erweitern von Elastic Beanstalk-Linux-Plattformen unter [the section called “Platform-Hooks” \(p. 36\)](#).

Wenn sich Elastic Beanstalk bei der Onlineregistrierung authentifizieren kann, die als Host für das private Repository fungiert, können Ihre Images bereitgestellt und abgerufen werden.

Verwenden der Datei `Dockerrun.aws.json v3`

In diesem Abschnitt wird eine andere Methode für die Authentifizierung von Elastic Beanstalk bei einem privaten Repository beschrieben. Dabei wird eine Authentifizierungsdatei mit dem Docker-Befehl erstellt und in einen Amazon S3-Bucket hochgeladen. In der Datei `Dockerrun.aws.json v3` müssen auch die Bucket-Informationen eingeschlossen werden.

So generieren Sie eine Authentifizierungsdatei und stellen sie in Elastic Beanstalk bereit

1. Erstellen Sie eine Authentifizierungsdatei mit dem `docker login`-Befehl. Für Repositorys auf Docker Hub führen Sie `docker login` aus:

```
$ docker login
```

Für andere Registries fügen Sie die URL des Registry-Servers ein:

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

Note

Wenn Ihre Elastic Beanstalk-Umgebung die Amazon Linux AMI-Docker-Plattformversion verwendet (vor Amazon Linux 2), lesen Sie die entsprechenden Informationen unter [the section called “Docker-Konfiguration auf Amazon Linux AMI \(Vorgängerversion von Amazon Linux 2\)” \(p. 92\)](#).

Weitere Informationen zur Authentifizierungsdatei finden Sie unter [Store images on Docker Hub](#) und [docker login](#) auf der Docker-Website.

2. Laden Sie eine Kopie der Authentifizierungsdatei mit dem Namen `.dockercfg` in einen sicheren Amazon S3-Bucket hoch.
 - Der Amazon S3-Bucket muss in derselben AWS-Region gehostet sein wie die Umgebung, die ihn verwendet. Elastic Beanstalk kann keine Dateien von einem Amazon S3-Bucket herunterladen, der in anderen Regionen gehostet wird.
 - Erteilen Sie Berechtigungen für den Vorgang `s3:GetObject` für die IAM-Rolle im Instance-Profil. Weitere Informationen finden Sie unter [Elastic Beanstalk Instance-Profile verwalten \(p. 904\)](#).
3. Schließen Sie die Amazon S3-Bucket-Informationen in den Authentication-Parameter der Datei `Dockerrun.aws.json v3` ein.

Im Anschluss sehen Sie ein Beispiel für eine `Dockerrun.aws.json v3`-Datei.

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

Note

Der `AWSEBDockerrunVersion`-Parameter gibt die Version der `Dockerrun.aws.json`-Datei an.

- Die Docker Amazon Linux 2-Plattform verwendet die Datei `Dockerrun.aws.json v3` für Umgebungen, in denen Docker Compose zum Einsatz kommt. Die Datei `Dockerrun.aws.json v1` wird für Umgebungen genutzt, in denen Docker Compose nicht verwendet wird.

- Die Amazon Linux AMI-Multicontainer-Docker-Plattform verwendet die Datei `Dockerrun.aws.json v2`.

Wenn sich Elastic Beanstalk bei der Onlineregistrierung authentifizieren kann, die als Host für das private Repository fungiert, können Ihre Images bereitgestellt und abgerufen werden.

Erstellen benutzerdefinierter Images mit einer Dockerfile-Datei

Falls Sie noch nicht über ein in einem Repository gehostetes Image verfügen, müssen Sie eine `Dockerfile` erstellen.

Der folgende Codeausschnitt ist ein Beispiel für die `Dockerfile`. Wenn Sie die Anweisungen in [Verwenden der Docker-Plattform \(p. 48\)](#) befolgen, können Sie die `Dockerfile` wie beschrieben hochladen. Elastic Beanstalk führt das Spiel 2048 aus, wenn Sie diese `Dockerfile` verwenden.

```
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" ]
```

Weitere Informationen zu den Anweisungen, die Sie in die `Dockerfile` einschließen können, finden Sie unter [Dockerfile-Referenz](#) auf der Docker-Website.

Konfiguration für Docker-Plattformen (ohne Docker Compose)

Wenn Docker Compose in Ihrer Elastic Beanstalk-Docker-Umgebung nicht verwendet wird, lesen Sie die zusätzlichen Informationen in den folgenden Abschnitten.

Konfiguration der Docker-Plattform ohne Docker Compose

Alle Webanwendungen, die Sie in Elastic Beanstalk in einer Docker-Umgebung bereitstellen, müssen eine `Dockerfile`- oder eine `Dockerrun.aws.json`-Datei enthalten. Sie können Ihre Webanwendung über einen Docker-Container in Elastic Beanstalk bereitstellen, indem Sie eine der folgenden Aktionen ausführen:

- Erstellen Sie eine `Dockerfile`-Datei, damit von Elastic Beanstalk ein benutzerdefiniertes Image erstellt und ausgeführt werden kann.
- Erstellen Sie eine `Dockerrun.aws.json`-Datei, um ein Docker-Image aus einem gehosteten Repository für Elastic Beanstalk bereitzustellen.
- Erstellen Sie eine `.zip`-Datei mit Ihren Anwendungsdateien, sämtlichen Abhängigkeiten der Anwendungsdatei, der `Dockerfile` und der `Dockerrun.aws.json`-Datei. Wenn Sie die EB CLI verwenden, um Ihre Anwendung bereitzustellen, wird automatisch eine `.zip`-Datei erstellt.

Wenn Sie nur eine `Dockerfile` oder nur eine `Dockerrun.aws.json`-Datei verwenden, um Ihre Anwendung bereitzustellen, müssen Sie keine `.zip`-Datei erstellen.

Dieses Thema ist eine Syntaxreferenz. Ausführliche Verfahren zum Starten von Docker-Umgebungen finden Sie unter [Verwenden der Docker-Plattform \(p. 48\)](#).

Dockerrun.aws.json-v1

Eine `Dockerrun.aws.json`-Datei beschreibt, wie ein Remote-Docker-Image als Elastic Beanstalk-Anwendung bereitgestellt wird. Diese JSON-Datei ist spezifisch für Elastic Beanstalk. Wenn Ihre Anwendung auf einem Bild ausgeführt, das in einem gehosteten Repository verfügbar ist, können Sie das Image in einer `Dockerrun.aws.json v1`-Datei angeben und die `Dockerfile` auslassen.

Zu den gültigen Schlüsseln und Werten für die `Dockerrun.aws.json v1`-Datei zählen unter anderem folgende Vorgänge:

AWSEBDockerrunVersion

(Erforderlich) Gibt die Versionsnummer als Wert 1 für Single-Container-Docker-Umgebungen an.

Authentifizierung

(Nur erforderlich für private Repositories) Gibt das Amazon S3-Objekt an, in dem die `.dockercfg`-Datei gespeichert ist.

Siehe [Verwenden von Images aus einem privaten Repository \(p. 60\)](#).

Abbild

Gibt das Docker-Basis-Image auf einem vorhandenen Docker-Repository an, aus dem Sie einen Docker-Container erstellt haben. Geben Sie den Wert des Name-Schlüssels im Format `<organization>/<image name>` für Images auf Docker-Hub oder `<site>/<organization name>/<image name>` für andere Websites an.

Wenn Sie ein Image in der `Dockerrun.aws.json`-Datei angeben, wird `docker pull` von jeder Instance Ihrer Elastic Beanstalk-Umgebung ausgeführt, um das Image auszuführen. Sie können optional den Update-Schlüssel einschließen. Der Standardwert ist `true`. Er weist Elastic Beanstalk an, das Repository zu prüfen, Updates auf das Image abzurufen und alle zwischengespeicherten Images zu überschreiben.

Geben Sie, wenn Sie eine `Dockerfile` verwenden, den Image-Schlüssel in der `Dockerrun.aws.json`-Datei nicht an. Elastic Beanstalk erstellt und verwendet immer das Image, das in `Dockerfile` Beschrieben ist, wenn eines vorhanden ist.

Ports

(Erforderlich, wenn Sie den Image-Schlüssel angeben) Listet die Ports auf, die auf dem Docker-Container bereitgestellt werden sollen. Elastic Beanstalk verwendet den ContainerPort-Wert für die Verbindung des Docker-Containers mit dem Reverse-Proxy, der auf dem Host ausgeführt wird.

Sie können mehrere Container-Ports angeben, von Elastic Beanstalk wird aber nur der erste Port verwendet. Dieser Port wird verwendet, um Ihren Container mit dem Reverse-Proxy des Hosts zu verbinden und Anforderungen aus dem öffentlichen Internet weiterzuleiten. Wenn Sie eine `Dockerfile` verwenden, sollte der erste ContainerPort-Wert mit dem ersten Eintrag in der EXPOSE-Liste der `Dockerfile` übereinstimmen.

Optional können Sie eine Liste der Ports in HostPort angeben. HostPort-Einträge geben die Host-Ports an, denen ContainerPort-Werte zugeordnet sind. Wenn Sie keinen HostPort-Wert angeben, wird standardmäßig der ContainerPort-Wert verwendet.

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

```
        "HostPort": 8000
    }
}
```

Datenträger

Ordnen Sie Volumes von einer EC2-Instance Ihrem Docker-Container zu. Geben Sie ein oder mehrere Volume-Arrays für die Zuordnung an.

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

Protokollierung

Geben Sie das Verzeichnis im Container an, in das Ihre Anwendung Protokolle schreibt. Elastic Beanstalk lädt alle Protokolle in dieses Verzeichnis in Amazon S3 hoch, wenn Sie Protokollfragmente oder Bundle-Protokolle anfordern. Wenn Sie Protokolle zu einem Ordner mit dem Namen `rotated` in diesem Verzeichnis rotieren, können Sie auch Elastic Beanstalk so konfigurieren, dass rotierte Protokolle für eine dauerhafte Speicherung in Amazon S3 hochgeladen werden. Weitere Informationen finden Sie unter [Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen \(p. 869\)](#).

Befehl

Geben Sie einen Befehl an, der im Container ausgeführt werden soll. Wenn Sie einen Entrypoint (Eintrittspunkt) angeben, dann wird Command (Befehl) als Argument für Entrypoint (Eintrittspunkt) hinzugefügt. Weitere Informationen finden Sie unter [CMD](#) in der Docker-Dokumentation.

Eintrittspunkt

Geben Sie einen Standard-Befehl an, der ausgeführt werden soll, wenn der Container gestartet wird. Weitere Informationen finden Sie unter [EINTRITTPUNKT](#) in der Docker-Dokumentation.

Der folgende Codeausschnitt ist ein Beispiel, das die Syntax der `Dockerrun.aws.json`-Datei für einen einzelnen Container.

```
{
  "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"
    }
```

Sie können Elastic Beanstalk nur mit der `Dockerrun.aws.json`-Datei oder mit einem `.zip`-Archiv bereitstellen, das sowohl die `Dockerrun.aws.json`-Datei als auch die `Dockerfile`-Datei enthält. Wenn Sie beide Dateien bereitstellen, beschreibt die `Dockerfile` das Docker-Image und die Datei `Dockerrun.aws.json` bietet zusätzliche Informationen für die Bereitstellung. Einzelheiten dazu finden Sie weiter unten in diesem Abschnitt.

Note

Die beiden Dateien müssen sich im Stammverzeichnis oder im oberste Level des `.zip`-Archivs befinden. Erstellen Sie kein Archiv aus einem Verzeichnis, das die Dateien enthält. Navigieren Sie stattdessen zu dem Verzeichnis und erstellen Sie das Archiv dort.

Wenn Sie die beide Dateien bereitstellen, geben Sie kein Image in der Datei `Dockerrun.aws.json` an. Elastic Beanstalk erstellt und verwendet das Image, das in `Dockerfile` beschrieben wird, und ignoriert das in der Datei `Dockerrun.aws.json` angegebene Image.

[Verwenden von Images aus einem privaten Repository](#)

Fügen Sie die Informationen über den Amazon S3-Bucket hinzu, der die Authentifizierungsdatei im Authentication-Parameter der `Dockerrun.aws.json v1`-Datei enthält. Stellen Sie sicher, dass der Authentication-Parameter einen gültigen Amazon S3-Bucket und -Schlüssel enthält. Der Amazon S3-Bucket muss in derselben AWS-Region gehostet sein wie die Umgebung, die ihn verwendet. Elastic Beanstalk lädt keine Dateien von Amazon S3-Buckets herunter, die in anderen Regionen gehostet werden.

Weitere Informationen zum Generieren und Hochladen der Authentifizierungsdatei finden Sie unter [Verwenden von Images aus einem privaten Repository \(p. 90\)](#).

Das folgende Beispiel zeigt die Verwendung einer Authentifizierungsdatei mit dem Namen `mydockercfg` in einem Bucket namens `DOC-EXAMPLE-BUCKET`, um ein privates Image in einem Drittanbieter-Registry zu verwenden.

```
{
    "AWSEBDockerrunVersion": "1",
    "Authentication": {
        "Bucket": "DOC-EXAMPLE-BUCKET",
        "Key": "mydockercfg"
    },
    "Image": {
        "Name": "quay.io/john Doe/private-image",
        "Update": "true"
    },
    "Ports": [
        {
            "ContainerPort": "1234"
        }
    ],
    "Volumes": [
        {
            "HostDirectory": "/var/app/mydb",
            "ContainerDirectory": "/etc/mysql"
        }
    ],
    "Logging": "/var/log/nginx"
}
```

Bereitstellen einer GlassFish-Anwendung auf der Docker-Plattform: Migrationspfad zu Amazon Linux 2

Das Ziel dieses Tutorials ist es, Kunden, die die vorkonfigurierte Docker GlassFish-Plattform (basierend auf dem Amazon Linux AMI) verwenden, einen Migrationspfad zu Amazon Linux 2 bereitzustellen. Sie können Ihre GlassFish-Anwendung zu Amazon Linux 2 migrieren, indem Sie GlassFish und Ihren Anwendungscode für ein Amazon Linux 2-Docker-Image bereitzustellen.

Das Tutorial führt Sie durch die Verwendung der AWS Elastic Beanstalk-Docker-Plattform, um eine Anwendung, die auf dem [Java EE GlassFish-Anwendungsserver](#) basiert, in einer Elastic Beanstalk-Umgebung bereitzustellen.

Wir demonstrieren zwei Ansätze zum Erstellen eines Docker-Images:

- Einfach – Stellen Sie Ihren GlassFish-Anwendungsquellcode bereit und lassen Sie Elastic Beanstalk ein Docker-Image als Teil der Bereitstellung Ihrer Umgebung erstellen und ausführen. Dies ist einfach einzurichten, zu Kosten einer erhöhten Bereitstellungszeit der Instance.
- Fortgeschritten – Erstellen Sie ein benutzerdefiniertes Docker-Image, das Ihren Anwendungscode und Abhängigkeiten enthält, und stellen Sie es Elastic Beanstalk für die Verwendung in Ihrer Umgebung bereit. Dieser Ansatz ist etwas komplexer und verkürzt die Bereitstellungszeit von Instances in Ihrer Umgebung.

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie über Kenntnisse zu den grundlegenden Elastic Beanstalk-Operationen, [Verwenden des Elastic Beanstalk Command Line Interface \(EB CLI\) \(p. 1008\)](#) und Docker verfügen. Damit Sie die Schritte in diesem Tutorial ausführen können, müssen Sie über eine funktionierende lokale Installation von Docker verfügen. Weitere Informationen über die Installation von Docker finden Sie im [Docker-Installationshandbuch](#).

Sofern noch nicht geschehen, befolgen Sie die Anweisungen in [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#), um Ihre erste Elastic Beanstalk-Umgebung zu starten. Dieses Tutorial verwendet die EB-CLI, aber Sie können auch mit der Elastic Beanstalk-Konsole Umgebungen erstellen und Anwendungen hochladen. Weitere Informationen zum Konfigurieren von Docker-Umgebungen finden Sie unter [Docker-Konfiguration \(p. 53\)](#).

Einfaches Beispiel: Geben Sie Ihren Anwendungscode ein

Dies ist eine einfache Möglichkeit, Ihre GlassFish-Anwendung bereitzustellen. Sie geben den Quellcode Ihrer Anwendung zusammen mit der in diesem Tutorial enthaltenen Dockerfile an. Elastic Beanstalk erstellt ein Docker-Image, das Ihre Anwendung und den GlassFish-Software-Stack enthält. Dann führt Elastic Beanstalk das Image auf den Instances Ihrer Umgebung aus.

Ein Problem mit diesem Ansatz besteht darin, dass Elastic Beanstalk das Docker-Image lokal erstellt, wenn es eine Instance für Ihre Umgebung erstellt. Das Erstellen des Image erhöht die Zeit für die Bereitstellung von Instances. Diese Auswirkung beschränkt sich nicht auf die anfängliche Erstellung der Umgebung – auch bei Scale-Out-Aktionen kann dies stattfinden.

So starten Sie eine Umgebung mit einer GlassFish-Beispielanwendung

1. Laden Sie das Beispiel docker-glassfish-al2-v1.zip herunter und entpacken Sie die .zip-Datei dann in ein Verzeichnis in Ihrer Entwicklungsumgebung.

```
~$ 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-al2-v1.zip
```

Ihre Verzeichnisstruktur sollte wie folgt aussehen.

```
~/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
```

Die folgenden Dateien sind der Schlüssel zum Erstellen und Ausführen eines Docker-Containers in Ihrer Umgebung:

- **Dockerfile** – Enthält Anweisungen, die Docker verwendet, um ein Image mit Ihrer Anwendung und erforderlichen Abhängigkeiten zu erstellen.
 - **glassfish-start.sh** – Ein Shell-Skript, das das Docker-Image ausführt, um Ihre Anwendung zu starten.
 - **Dockerrun.aws.json** – Stellt einen Protokollschlüssel bereit, um die GlassFish-Anwendungsserveranmeldung in [Protokolldateianforderungen \(p. 869\)](#) einzuschließen. Wenn Sie nicht an GlassFish-Protokollen interessiert sind, können Sie diese Datei weglassen.
2. Konfigurieren Sie Ihr lokales Verzeichnis für die Bereitstellung in Elastic Beanstalk.

```
~/glassfish-example$ eb init -p docker glassfish-example
```

3. (Optional) Verwenden Sie den Befehl eb local run für die lokale Erstellung und Ausführung Ihres Containers.

```
~/glassfish-example$ eb local run --port 8080
```

Note

Weitere Informationen zum eb local-Befehl finden Sie unter [the section called “eb local” \(p. 1071\)](#). Der Befehl wird unter Windows nicht unterstützt. Alternativ können Sie Ihre Container mit den Befehlen docker build und docker run erstellen und ausführen. Weitere Informationen finden Sie in der [Docker-Dokumentation](#).

4. (Optional) Wenn Ihr Container ausgeführt wird, verwenden Sie den eb local open-Befehl, um Ihre Anwendung in einem Webbrowser anzuzeigen. Alternativ öffnen Sie <http://localhost:8080/> in einem Webbrowser.

```
~/glassfish-example$ eb local open
```

5. Verwenden Sie den Befehl eb create, um eine Umgebung zu erstellen und die Anwendung bereitzustellen.

```
~/glassfish-example$ eb create glassfish-example-env
```

6. Verwenden Sie nach dem Starten der Umgebung den Befehl eb open, um sie in einem Webbrowser anzuzeigen.

```
~/glassfish-example$ eb open
```

Wenn Sie mit dem Beispiel fertig sind, beenden Sie die Umgebung und löschen Sie zugehörige Ressourcen.

```
~/glassfish-example$ eb terminate --all
```

Fortgeschrittenes Beispiel: Bereitstellen eines vorgefertigten Docker-Images

Dies ist eine fortgeschrittene Möglichkeit, Ihre GlassFish-Anwendung bereitzustellen. Aufbauend auf dem ersten Beispiel erstellen Sie ein Docker-Image, das Ihren Anwendungscode und den GlassFish-Softwarestack enthält, und übertragen es an Docker Hub. Nachdem Sie diesen einmaligen Schritt ausgeführt haben, können Sie Elastic Beanstalk-Umgebungen basierend auf Ihrem benutzerdefinierten Image starten.

Wenn Sie eine Umgebung starten und Ihr Docker-Image bereitzustellen, laden Instances in Ihrer Umgebung dieses Image direkt herunter, verwenden es direkt und müssen kein Docker-Image erstellen. So wird die Bereitstellungszeit der Instance verkürzt.

Note

In den folgenden Schritten erstellen Sie ein öffentlich verfügbares Docker-Image.

So starten Sie eine Umgebung mit einem vordefinierten GlassFish-Anwendung-Docker-Image

1. Laden Sie das Beispiel `docker-glassfish-al2-v1.zip` herunter und entpacken Sie es wie im vorherigen [einfachen Beispiel \(p. 61\)](#). Wenn Sie dieses Beispiel bereits abgeschlossen haben, können Sie das bereits vorhandene Verzeichnis verwenden.
2. Erstellen Sie ein Docker-Image und schieben Sie es an Docker Hub.

```
~/glassfish-example$ docker build -t docker-username/beanstalk-glassfish-example:latest .
~/glassfish-example$ docker push docker-username/beanstalk-glassfish-example:latest
```

Note

Bevor Sie Ihr Image übertragen, müssen Sie möglicherweise `docker login` ausführen.

3. Erstellen Sie ein zusätzliches Verzeichnis.

```
~$ mkdir glassfish-prebuilt
~$ cd glassfish-prebuilt
```

4. Kopieren Sie das folgende Beispiel in eine Datei mit dem Namen `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. Konfigurieren Sie Ihr lokales Verzeichnis für die Bereitstellung in Elastic Beanstalk.

```
~/glassfish-prebuilt$ eb init -p docker glassfish-prebuilt$
```

6. (Optional) Verwenden Sie den Befehl eb local run, um den Container lokal auszuführen.

```
~/glassfish-prebuilt$ eb local run --port 8080
```

7. (Optional) Wenn Ihr Container ausgeführt wird, verwenden Sie den eb local open-Befehl, um Ihre Anwendung in einem Webbrowser anzusehen. Alternativ öffnen Sie <http://localhost:8080> in einem Webbrowser.

```
~/glassfish-prebuilt$ eb local open
```

8. Verwenden Sie den Befehl eb create, um eine Umgebung zu erstellen und Ihr Docker-Image bereitzustellen.

```
~/glassfish-prebuilt$ eb create glassfish-prebuilt-env
```

9. Verwenden Sie nach dem Starten der Umgebung den Befehl eb open, um sie in einem Webbrowser anzusehen.

```
~/glassfish-prebuilt$ eb open
```

Wenn Sie mit dem Beispiel fertig sind, beenden Sie die Umgebung und löschen Sie zugehörige Ressourcen.

```
~/glassfish-prebuilt$ eb terminate --all
```

Verwenden der Multicontainer-Docker-Plattform (Amazon Linux AMI)

Note

Diese Plattform unterstützt nur das Amazon Linux AMI-Betriebssystem (Vorgängerversion von Amazon Linux 2). Multicontainer-Docker-Funktionalität unter Amazon Linux 2 wird von der [Docker \(p. 48\)](#)-Plattform bereitgestellt und langfristig unterstützt.

Wenn von Ihrer Elastic Beanstalk-Docker-Umgebung eine Amazon Linux-AMI-Plattformversion (Vorgängerversion von Amazon Linux 2) verwendet wird, können Sie mit der Multicontainer-Docker-Plattform für Elastic Beanstalk Docker-Umgebungen erstellen, die mehrere Container pro Amazon EC2-Instance unterstützen.

Elastic Beanstalk verwendet den Amazon Elastic Container Service (Amazon ECS), um Container-Bereitstellungen in Multicontainer-Docker-Umgebungen zu koordinieren. Amazon ECS bietet Tools zum Verwalten eines Instance-Clusters mit Docker-Containern. Elastic Beanstalk übernimmt Amazon ECS-Aufgaben wie Clustererstellung, Aufgabendefinition und -ausführung. Die Instances in der Umgebung werden jeweils im gleichen Satz von Containern ausgeführt, die in einer `Dockerrun.aws.json`-Datei definiert sind.

Note

Einige Regionen bieten Amazon ECS nicht an. Multicontainer Docker-Umgebungen werden in diesen Regionen nicht unterstützt.

Weitere Informationen zu den AWS-Services, die in den einzelnen Regionen angeboten werden, finden Sie in der [Tabelle der Regionen](#).

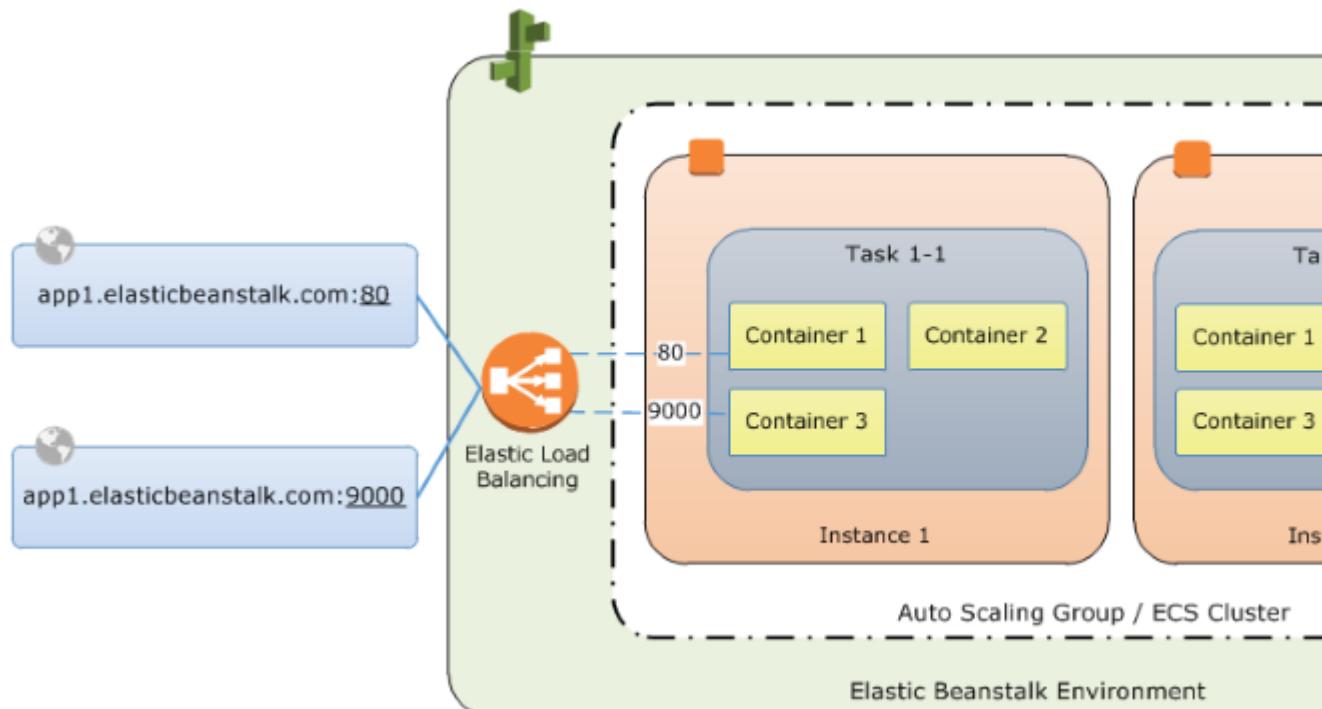
Themen

- [Multicontainer-Docker-Plattform \(p. 65\)](#)
- [Dockerrun.aws.json-Datei \(p. 66\)](#)
- [Docker-Images \(p. 66\)](#)
- [Rolle für Container-Instances \(p. 66\)](#)
- [Von Elastic Beanstalk erstellte Amazon ECS-Ressourcen \(p. 67\)](#)
- [Verwenden mehrerer Elastic Load Balancing-Listener \(p. 67\)](#)
- [Fehlgeschlagene Container-Bereitstellungen \(p. 68\)](#)
- [Multicontainer-Docker-Konfiguration \(p. 69\)](#)
- [Multicontainer Docker-Umgebungen mit der Elastic Beanstalk-Konsole \(p. 73\)](#)
- [Migration zur Docker Amazon Linux 2-Plattform \(p. 79\)](#)

Multicontainer-Docker-Plattform

Standardmäßige generische und vorkonfigurierte Docker-Plattformen auf Elastic Beanstalk unterstützen nur einen einzelnen Docker-Container pro Elastic Beanstalk-Umgebung. Um Docker bestmöglich zu nutzen, können Sie mit Elastic Beanstalk eine Umgebung erstellen, in der Ihre Amazon EC2-Instances mehrere Docker-Container parallel ausführen.

Das folgende Diagramm zeigt ein Beispiel für eine Elastic Beanstalk-Umgebung, die mit drei Docker-Containern konfiguriert ist, die auf jeder Amazon EC2-Instance in einer Auto Scaling-Gruppe ausgeführt werden:



Dockerrun.aws.json-Datei

Container-Instances – Amazon EC2-Instances, die Multicontainer Docker in einer Elastic Beanstalk-Umgebung ausführen, benötigen eine Konfigurationsdatei mit dem Namen `Dockerrun.aws.json`. Diese Datei gilt nur für Elastic Beanstalk und kann eigenständig oder in Kombination mit Quellcode und Inhalten in einem [Quell-Bundle \(p. 412\)](#) verwendet werden, um eine Umgebung auf einer Docker-Plattform zu erstellen.

Note

Version 1 des `Dockerrun.aws.json`-Format wird verwendet, um einen einzelnen Docker-Container auf einer Elastic-Beanstalk-Umgebung zu starten. Version 2 fügt Unterstützung für mehrere Container pro Amazon EC2-Instance hinzu und kann nur mit der Multicontainer-Docker-Plattform verwendet werden. Das Format unterscheidet sich deutlich von der früheren Version, die unter [Docker-Konfiguration \(p. 53\)](#) im Detail beschrieben wird.

Detaillierte Informationen zum aktualisierten Format und eine Beispieldatei finden Sie unter [Dockerrun.aws.json-v2 \(p. 69\)](#).

Docker-Images

Die Multicontainer-Docker-Plattform für Elastic Beanstalk erfordert, dass Images vorkonfiguriert und in einem öffentlichen oder privaten Online-Image-Repository gespeichert werden.

Note

Das Erstellen von benutzerdefinierten Images während der Bereitstellung mit einer `Dockerfile`-Datei wird von der Multicontainer-Docker-Plattform auf Elastic Beanstalk nicht unterstützt. Erstellen Sie Ihre Images und stellen Sie sie in einem Online-Repository bereit, bevor Sie eine Elastic Beanstalk-Umgebung erstellen.

Geben Sie Images mit Namen in `Dockerrun.aws.json` an. Beachten Sie diese Konventionen:

- Abbilder in offiziellen Repositorys in Docker Hub verwenden einen einzelnen Namen (z. B. `ubuntu` oder `mongo`).
- Images in anderen Repositorys in Docker Hub sind mit einem Organisationsnamen qualifiziert (z. B. `amazon/amazon-ecs-agent`).
- Images in anderen Online-Verzeichnissen sind durch einen Domänenamen zusätzlich qualifiziert (z. B. `quay.io/assemblyline/ubuntu`).

Zum Konfigurieren von Elastic Beanstalk für die Authentifizierung mit einem privaten Repository fügen Sie den `authentication`-Parameter in die `Dockerrun.aws.json`-Datei ein.

Rolle für Container-Instances

Elastic Beanstalk verwendet ein Amazon ECS-optimiertes AMI mit einem Amazon ECS-Container-Agenten, der in einem Docker-Container ausgeführt wird. Der Agent kommuniziert mit Amazon ECS zum Koordinieren von Container-Bereitstellungen. Für die Kommunikation mit Amazon ECS muss jede Amazon EC2-Instance über entsprechende Berechtigungen in IAM verfügen. Diese Berechtigungen werden an das standardmäßige [Instance-Profile \(p. 21\)](#) angefügt, wenn Sie eine Umgebung in der Elastic Beanstalk-Verwaltungskonsole erstellen:

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

Wenn Sie Ihr eigenes Instance-Profil erstellt haben, können Sie die verwaltete AWS-Elastic-Beanstalk-Multicontainer-Docker-Richtlinie anfügen, um sicherzustellen, dass die Berechtigungen immer auf dem neuesten Stand sind. Anweisungen zum Erstellen von Richtlinien und Rollen in IAM; finden Sie unter [Erstellen von IAM-Rollen](#) im IAM-Benutzerhandbuch.

Von Elastic Beanstalk erstellte Amazon ECS-Ressourcen

Wenn Sie eine Umgebung mit der Multicontainer-Docker-Plattform erstellen, erstellt und konfiguriert Elastic Beanstalk automatisch verschiedene Amazon Elastic Container Service-Ressourcen beim Entwickeln der Umgebung, um die erforderlichen Container auf den einzelnen Amazon EC2-Instances zu erstellen.

- Amazon ECS-Cluster – Container-Instances in Amazon ECS sind in Cluster organisiert. Bei Verwendung mit Elastic Beanstalk wird ein Cluster immer für jede Multicontainer-Docker-Umgebung erstellt.
- Amazon ECS-Aufgabendefinition – Elastic Beanstalk nutzt die `Dockerrun.aws.json`-Datei in Ihrem Projekt zum Generieren der Amazon ECS-Aufgabendefinition, die für die Konfiguration der Container-Instances in der Umgebung verwendet wird.
- Amazon ECS-Aufgabe – Elastic Beanstalk kommuniziert mit Amazon ECS über die Ausführung einer Aufgabe auf allen Instances in der Umgebung zum Koordinieren der Container-Bereitstellung. In einer skalierbaren Umgebung initiiert Elastic Beanstalk eine neue Aufgabe, wenn eine Instance zum Cluster hinzugefügt wird. In seltenen Fällen müssen Sie möglicherweise die Größe des für Container und Images vorgesehenen Platzes erhöhen. Erfahren Sie mehr im [Konfigurieren von Docker-Umgebungen \(p. 85\)](#)-Abschnitt.
- Amazon ECS Container-Agent – Der Agent wird in einem Docker-Container auf den Instances in Ihrer Umgebung ausgeführt. Der Agent ruft den Amazon ECS Service ab und wartet, bis eine Aufgabe ausgeführt wird.
- Amazon ECS-Daten-Volumes – Elastic Beanstalk fügt Volume-Definitionen (zusätzlich zu den Volumes, die Sie in `Dockerrun.aws.json` definieren) zu den Aufgabendefinitionen hinzu, um die Protokollerfassung zu erleichtern.

Elastic Beanstalk erstellt Protokoll-Volumes auf der Container-Instance, eines für jeden Container, unter `/var/log/containers/containername`. Diese Volumes haben den Namen `awseb-logs-containername` und werden für Container für das Mounting bereitgestellt. Details zum Mounting finden Sie unter [Format der Containerdefinition \(p. 71\)](#).

Verwenden mehrerer Elastic Load Balancing-Listener

Sie können mehrere Elastic Load Balancing-Listener auf einer Multicontainer-Docker-Umgebung konfigurieren, um eingehenden Datenverkehr für Proxys oder andere Services zu unterstützen, die nicht auf dem Standard-HTTP-Port ausgeführt werden.

Erstellen Sie einen `.ebextensions`-Ordner in Ihrem Quell-Bundle und fügen Sie eine Datei mit einer `.config`-Dateierweiterung hinzu. Das folgende Beispiel zeigt eine Konfigurationsdatei, die einen Elastic Load Balancing-Listener auf Port 8080 erstellt.

`.ebextensions/elb-listener.config`

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

Wenn Ihre Umgebung in einer von Ihnen erstellten benutzerdefinierten [Amazon Virtual Private Cloud](#) (Amazon VPC) läuft, kümmert sich Elastic Beanstalk um den Rest. In einer Standard-VPC müssen Sie die Sicherheitsgruppe Ihrer Instance so konfigurieren, dass Sie eingehenden Datenverkehr vom Load Balancer erlaubt. Fügen Sie eine zweite Konfigurationsdatei hinzu, die eine Ingress-Regel zur Sicherheitsgruppe hinzufügt:

`.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"] }
```

Weitere Informationen zum Konfigurationsdateiformat finden Sie unter [Hinzufügen und Anpassen von Elastic Beanstalk-Umgebungsressourcen \(p. 745\)](#) und [Optionseinstellungen \(p. 724\)](#).

Zusätzlich zum Hinzufügen eines Listener zur Elastic Load Balancing-Konfiguration und dem Öffnen eines Port in der Sicherheitsgruppe müssen Sie den Port auf der Host-Instance einem Port auf dem Docker-Container im `containerDefinitions`-Abschnitt der `Dockerrun.aws.json`-Datei hinzufügen. Der folgende Auszug zeigt ein Beispiel:

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

Unter [Dockerrun.aws.json-v2 \(p. 69\)](#) finden Sie weitere Details zum `Dockerrun.aws.json`-Dateiformat.

Fehlgeschlagene Container-Bereitstellungen

Wenn eine Amazon ECS-Aufgabe fehlschlägt, werden ein oder mehrere Container in Ihrer Elastic Beanstalk-Umgebung nicht gestartet. Elastic Beanstalk setzt Multicontainer-Umgebungen aufgrund einer fehlgeschlagenen Amazon ECS-Aufgabe nicht zurück. Wenn ein Container in der Umgebung nicht startet, stellen Sie die aktuelle Version oder eine frühere funktionierende Version mit der Elastic Beanstalk-Konsole erneut bereit.

So stellen Sie eine vorhandene Version bereit

1. Öffnen Sie die Elastic Beanstalk-Konsole in der Region Ihrer Umgebung.

2. Klicken Sie auf Actions (Aktionen) rechts neben Ihrem Anwendungsnamen und klicken Sie dann auf Anwendungsversionen anzeigen.
3. Wählen Sie eine Version Ihrer Anwendung und klicken Sie auf Deploy (Bereitstellen).

Multicontainer-Docker-Konfiguration

Eine `Dockerrun.aws.json`-Datei ist eine Elastic Beanstalk-spezifische JSON-Datei, die beschreibt, wie eine Reihe von Docker-Containern als Elastic Beanstalk-Anwendung bereitgestellt wird. Sie können eine `Dockerrun.aws.json`-Datei für eine Multicontainer-Docker-Umgebung verwenden.

`Dockerrun.aws.json` beschreibt die Container, die für jede Container-Instance (Amazon EC2-Instance, die Docker-Container hostet) in der Umgebung bereitgestellt werden sollen, sowie die Daten-Volumes, die auf der Host-Instance der bereitzustellenden Container erstellt werden sollen.

Eine `Dockerrun.aws.json`-Datei kann alleine verwendet oder mit einem zusätzlichen Quell-Code in einem einzigen Archiv zu einer ZIP-Datei gepackt werden. Quellcode, der mit einer `Dockerrun.aws.json`-Datei archiviert wird, wird in Amazon EC2-Container-Instances bereitgestellt und ist im `/var/app/current/-`-Verzeichnis zugänglich. Verwenden Sie den `volumes`-Abschnitt der `config`, um Datei-Volumes für die Docker-Container bereitzustellen, die auf der Host-Instance ausgeführt werden. Verwenden Sie den `mountPoints`-Abschnitt der eingebetteten Container-Definitionen, um diese Volumes auf Bereitstellungspunkte abzubilden, die Anwendungen in den Docker-Containern verwenden können.

Themen

- [Dockerrun.aws.json-v2 \(p. 69\)](#)
- [Verwenden von Images aus einem privaten Repository \(p. 71\)](#)
- [Format der Containerdefinition \(p. 71\)](#)

Dockerrun.aws.json-v2

Die `Dockerrun.aws.json`-Datei enthält drei Abschnitte:

AWSEBDockerrunVersion

Gibt die Versionsnummer als Wert 2 für Multicontainer-Docker-Umgebungen an.
`containerDefinitions`

Ein Array von Containerdefinitionen, wie unten aufgeführt.
`volumes`

Erstellt Volumes aus Ordnern in der Amazon EC2-Container-Instance oder aus Ihrem Quell-Bundle (bereitgestellt in `/var/app/current`). Mounten Sie diese Volumes auf Pfade innerhalb Ihrer Docker-Container mit `mountPoints` in der [Containerdefinition \(p. 71\)](#).

Note

Elastic Beanstalk konfiguriert zusätzliche Volumes für Protokolle, eines für jeden Container. Diese sollten von Ihren Docker-Containern bereitgestellt werden, um Protokolle an die Host-Instance zu schreiben. Details dazu finden Sie unter [Format der Containerdefinition \(p. 71\)](#).

Volumes werden in folgendem Format angegeben:

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

```
        }
    ],
}
```

Authentifizierung

(optional) Der Speicherort in Amazon S3 für eine `.dockercfg`-Datei, die Authentifizierungsdaten für ein privates Repository enthält. Verwendet das folgende Format:

```
"authentication": {
    "bucket": "DOC-EXAMPLE-BUCKET",
    "key": "mydockercfg"
},
```

Details dazu finden Sie unter [Verwenden von Images aus einem privaten Repository \(p. 71\)](#).

Der folgende Codeausschnitt ist ein Beispiel, das die Syntax der `Dockerrun.aws.json`-Datei für eine Instance mit zwei Containern zeigt.

```
{
    "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"
    }
]
}
]
```

Verwenden von Images aus einem privaten Repository

Fügen Sie die Informationen über den Amazon S3-Bucket hinzu, der die Authentifizierungsdatei im authentication-Parameter der `Dockerrun.aws.json`-Datei enthält. Stellen Sie sicher, dass der authentication-Parameter einen gültigen Amazon S3-Bucket und -Schlüssel enthält. Der Amazon S3-Bucket muss in derselben Region gehostet sein wie die Umgebung, die ihn verwendet. Elastic Beanstalk lädt keine Dateien von Amazon S3-Buckets herunter, die in anderen Regionen gehostet werden.

Weitere Informationen zum Generieren und Hochladen der Authentifizierungsdatei finden Sie unter [Verwenden von Images aus einem privaten Repository \(p. 90\)](#).

Format der Containerdefinition

Die Bereiche für die Containerdefinition und Volumes von `Dockerrun.aws.json` verwenden die gleiche Formatierung wie die entsprechenden Bereiche einer Amazon ECS-Datei zur Aufgabendefinition.

Die folgenden Beispiele zeigen eine Teilmenge von Parametern, die allgemein verwendet werden. Weitere optionale Parameter sind verfügbar. Weitere Informationen zum Aufgabendefinitionsformat und eine vollständige Liste der Aufgabendefinitionsparameter finden Sie in den [Amazon ECS-Aufgabendefinitionen](#) im Amazon Elastic Container Service-Entwicklerhandbuch.

Eine `Dockerrun.aws.json`-Datei enthält ein Array aus einem oder mehreren Containerdefinitionsobjekten mit den folgenden Feldern:

Name

Name des Containers. Weitere Informationen über die maximale Länge und die zulässigen Zeichen finden Sie unter [Standardparameter für Containerdefinition](#).

Abbild

Der Name eines Docker-Image in einem Online-Docker-Repository, aus dem Sie einen Docker-Container erstellt haben. Beachten Sie diese Konventionen:

- Abbilder in offiziellen Repositorys in Docker Hub verwenden einen einzelnen Namen (z. B. `ubuntu` oder `mongo`).

- Images in anderen Repositorys in Docker Hub sind mit einem Organisationsnamen qualifiziert (z. B. `amazon/amazon-ecs-agent`).
- Abbilder in anderen Online-Repositorys sind durch einen Domänennamen zusätzlich qualifiziert (z. B. `quay.io/assemblyline/ubuntu`).

Umgebung

Ein Array an Umgebungsvariablen, die an den Container übergeben werden.

Der folgende Eintrag definiert beispielsweise eine Umgebungsvariable mit dem Namen **Container** und dem Wert **PHP**:

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

essential

"true", wenn die Aufgabe stoppen soll, wenn der Container fehlschlägt. Unwesentliche Container können abschließen oder abstürzen, ohne dass dies Auswirkungen auf die restlichen Container in der Instance hat.

memory

Größe des Speichers auf der Container-Instance, die für den Container reserviert werden soll. Geben Sie eine Ganzzahl ungleich null für einen oder beide der Parameter **memory** oder **memoryReservation** in Containerdefinitionen an.

memoryReservation

Die weiche Arbeitsspeichergrenze (in MiB) für die Reservierung für den Container. Geben Sie eine Ganzzahl ungleich null für einen oder beide der Parameter **memory** oder **memoryReservation** in Containerdefinitionen an.

mountPoints

Volumes aus der Amazon EC2-Container-Instance, die gemountet werden soll und der Speicherort im Docker-Container-Dateisystem, an dem sie gemountet werden sollen. Wenn Sie Volumes mounten, die mit Anwendungs-Inhalten enthalten, kann Ihr Container die Daten lesen, die Sie in Ihrem Quell-Bundle hochladen. Wenn Sie Protokoll-Volumes für das Schreiben von Protokolldaten mounten, kann Elastic Beanstalk Protokolldaten von diese Volumes erfassen.

Elastic Beanstalk erstellt Protokoll-Volumes auf der Container-Instance, eines für jeden Docker-Container, unter `/var/log/containers/containername`. Diese Volumes haben die Bezeichnung `awseb-logs-containername` und sollten in das Verzeichnis innerhalb der Container-Dateistruktur bereitgestellt werden, in dem Protokolle geschrieben werden.

Beispiel: Der folgende Mounting-Punkt weist den nginx-Protokoll-Standort im Container dem von Elastic Beanstalk generierten Volume für den `nginx-proxy`-Container zu.

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

portMappings

Ordnet Netzwerk-Ports auf dem Container Ports auf dem Host zu.

links

Liste der Container, zu denen Link erstellt werden soll. Verknüpfte Container können sich gegenseitig erkennen und sicher kommunizieren.

volumesFrom

Mounten aller Volumes aus einem anderen Container. Zum Beispiel zum Mounten von Volumes aus einem Container mit dem Namen web:

```
"volumesFrom": [  
  {  
    "sourceContainer": "web"  
  }  
,
```

Multicontainer Docker-Umgebungen mit der Elastic Beanstalk-Konsole

Sie können einen Cluster aus Multicontainer-Instances in einer Single-Instance-Umgebung oder einer skalierbaren Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-Konsole starten. In diesem Tutorial werden Details zur Container-Konfiguration und zur Vorbereitung des Quellcodes für eine Umgebung mit zwei Containern beschrieben.

Die Container, eine PHP-Anwendung und ein nginx-Proxy werden parallel auf jeder der Amazon Elastic Compute Cloud-Instances (Amazon EC2) in einer Elastic Beanstalk-Umgebung ausgeführt. Nach dem Erstellen der Umgebung und der Überprüfung, ob die Anwendungen ausgeführt werden, müssen Sie eine Verbindung zu einer Container-Instance herstellen, um zu sehen, ob alles zusammenpasst.

Abschnitte

- [Definieren von Docker-Containern \(p. 73\)](#)
- [Inhalte hinzufügen \(p. 75\)](#)
- [Bereitstellen in Elastic Beanstalk \(p. 76\)](#)
- [Verbinden mit einer Container-Instance \(p. 77\)](#)
- [Überprüfen des Amazon ECS-Container-Agenten \(p. 77\)](#)

Definieren von Docker-Containern

Der erste Schritt beim Erstellen einer neuen Docker-Umgebung ist das Erstellen eines Verzeichnisses für Ihre Anwendungsdaten. Dieser Ordner kann sich in einem beliebigen Verzeichnis auf Ihrem lokalen Computer befinden und einen beliebigen Namen haben. Zusätzlich zu einer Container-Konfigurationsdatei enthält dieser Ordner den Inhalt, den Sie in Elastic Beanstalk hochladen und für Ihre Umgebung bereitstellen.

Note

Der Code für dieses Tutorial ist im "awslabs"-Repository auf GitHub unter <https://github.com/awslabs/eb-docker-nginx-proxy> verfügbar.

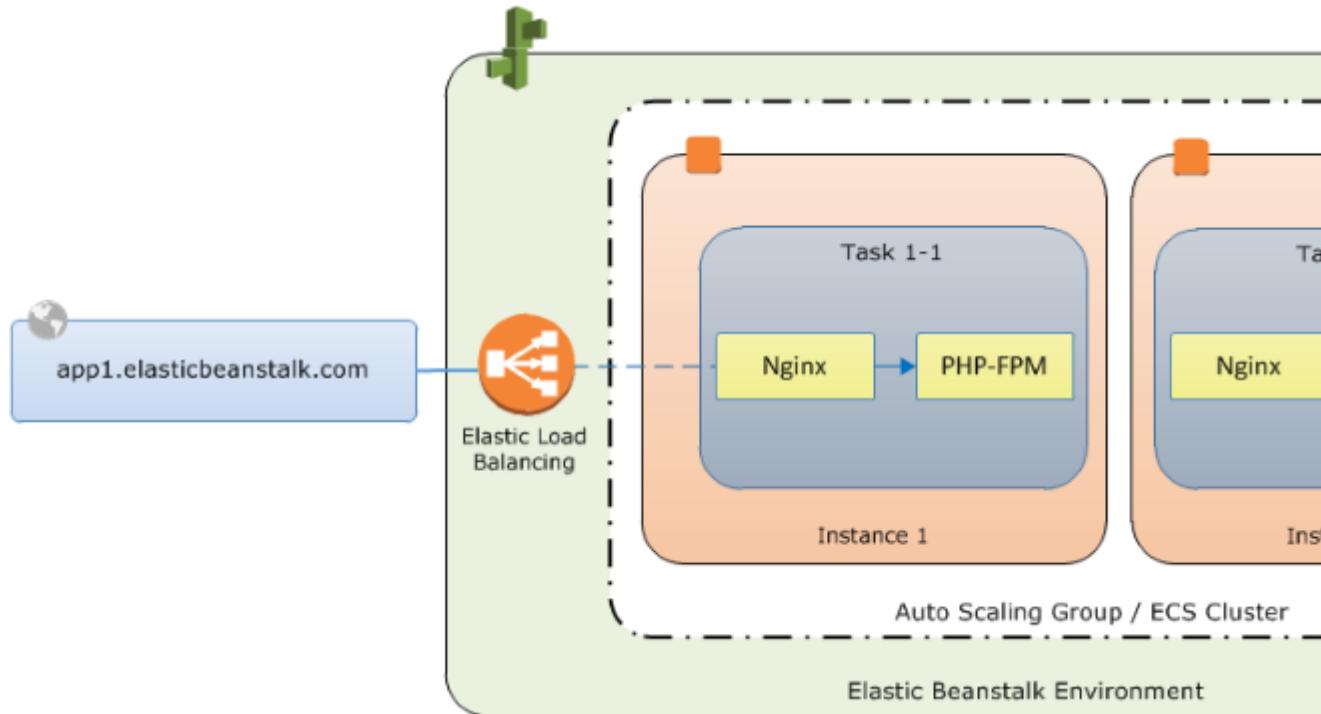
Die Datei, die Elastic Beanstalk zum Konfigurieren der Container auf einer Amazon EC2-Instance verwendet, ist eine JSON-formatierte Textdatei mit dem Namen `Dockerrun.aws.json`. Erstellen Sie eine Textdatei mit diesem Namen im Stammverzeichnis Ihrer Anwendung und fügen Sie den folgenden Text hinzu:

```
{  
  "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"
      }
    ]
  }
]
```

In dieser Beispielkonfiguration werden zwei Container definiert, eine PHP-Website mit einem nginx-Proxy davor. Diese beiden Containern werden parallel in Docker-Containern auf jeder Instance in Ihrer Elastic Beanstalk-Umgebung ausgeführt. Dabei greifen sie auf gemeinsame Inhalte (die Inhalte der Website) aus

Volumes auf der Host-Instance zu, die auch in dieser Datei definiert sind. Die Container selbst werden aus Images erstellt, die in offiziellen Repositorys auf Docker Hub gehostet werden. Die sich ergebende Umgebung sieht wie folgt aus:



Die Volumes in der Konfiguration entsprechen dem Inhalt, den Sie als Nächstes erstellen und als Teil Ihres Anwendungs-Quell-Bundle hochladen. Die Container greifen auf Inhalt auf dem Host zu, indem Volumes im `mountPoints`-Abschnitt der Container-Definitionen gemountet werden.

Weitere Informationen zum Format von `Dockerrun.aws.json` und dessen Parametern finden Sie unter [Format der Containerdefinition \(p. 71\)](#).

Inhalte hinzufügen

Als Nächstes fügen Sie einige Inhalte für Ihre PHP-Website hinzu, um sie für Besucher anzuzeigen, und eine Konfigurationsdatei für den nginx-Proxy.

`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(/|$) {
    fastcgi_split_path_info ^(.+?\.(php|html|htm))(.*$);
    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;
}
}
```

Bereitstellen in Elastic Beanstalk

Ihr Anwendungsordner enthält nun die folgenden Dateien:

```
### Dockerrun.aws.json
### php-app
#   ### index.php
#   ### static.html
### proxy
    ### conf.d
        ### default.conf
```

Dies ist alles, was Sie benötigen, um die Elastic Beanstalk-Umgebung zu erstellen. Erstellen Sie ein .zip-Archiv der obigen Dateien und Ordner (nicht im Projektordner der obersten Ebene enthalten). Zum Erstellen des Archivs in Windows Explorer wählen Sie den Inhalt des Projektordners, klicken Sie mit der rechten Maustaste, wählen Sie Send To (Senden an) und klicken Sie dann auf Compressed (zipped) Folder (Komprimierter (gezippter) Ordner).

Note

Weitere Informationen über die erforderliche Dateistruktur und Anweisungen für die Erstellung von Archiven in anderen Umgebungen finden Sie unter [Erstellen des Quell-Bundles einer Anwendung. \(p. 412\)](#)

Laden Sie als Nächstes das Quell-Bundle in Elastic Beanstalk hoch und erstellen Sie Ihre Umgebung. Wählen Sie für Platform (Plattform) die Option Docker aus. Wählen Sie für Platform branch (Plattformvariante) die Option Multi-container Docker running on 64bit Amazon Linux (Unter Amazon Linux mit 64 Bit ausgeführte Multicontainer-Docker).

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced.
2. Wählen Sie für Platform (Plattform) die Plattform und den Plattformzweig aus, die der von Ihrer Anwendung verwendeten Sprache entsprechen, oder die Docker-Plattform für containerbasierte Anwendungen.
3. Wählen Sie für Application code (Anwendungscode) die Option Upload your code (Eigenen Code hochladen).
4. Wählen Sie Local file (Lokale Datei), wählen Sie Choose file (Datei wählen) und öffnen Sie das Quell-Bundle.
5. Klicken Sie auf Review and launch (Überprüfen und starten).
6. Prüfen Sie die verfügbaren Einstellungen und wählen Sie Create app (Anwendung erstellen) aus.

Die Elastic Beanstalk-Konsole leitet Sie zum Verwaltungs-Dashboard für die neue Umgebung um. Dieser Bildschirm zeigt den Status der Umgebung und Ereignisse, die vom Elastic Beanstalk-Service ausgegeben werden. Wenn der Status grün ist, klicken Sie auf die URL neben dem Umgebungsnamen, um Ihre neue Website anzuzeigen.

Verbinden mit einer Container-Instance

Als Nächstes stellen Sie eine Verbindung zu einer Amazon EC2-Instance in Ihrer Elastic Beanstalk-Umgebung her, um einige der beweglichen Teile in Aktion zu sehen.

Am einfachsten ist es, in Ihrer Umgebung mithilfe der EB CLI eine Verbindung mit einer Instance herzustellen. Sofern noch nicht geschehen, müssen Sie [die EB CLI zuerst installieren \(p. 1009\)](#), um sie nutzen zu können. Sie müssen Ihre Umgebung außerdem mit einem Amazon EC2-SSH-Schlüsselpaar konfigurieren. Verwenden Sie hierzu entweder die [Sicherheitskonfigurationsseite \(p. 615\)](#) der Konsole oder den EB CLI-Befehl `eb init (p. 1067)`. Um eine Verbindung mit einer Umgebungs-Instance herzustellen, verwenden Sie den EB CLI-Befehl `eb ssh (p. 1087)`.

Nachdem Sie eine Verbindung zu einer Amazon EC2-Instance hergestellt haben, auf der Ihre Docker-Container gehostet werden, können Sie sehen, wie die Dinge eingerichtet sind. Führen Sie `ls` auf `/var/app/current` aus:

```
[ec2-user@ip-10-0-0-117 ~]$ ls /var/app/current
Dockerrun.aws.json  php-app  proxy
```

Dieses Verzeichnis enthält die Dateien aus dem Quell-Bundle, die Sie während der Umgebungserstellung in Elastic Beanstalk hochgeladen haben.

```
[ec2-user@ip-10-0-0-117 ~]$ ls /var/log/containers
nginx                           nginx-proxy-fffffd873ada5-stdouterr.log  rotated
nginx-66a4fd37eb63-stdouterr.log  php-app
nginx-proxy                      php-app-b894601a1364-stdouterr.log
```

Hier werden Protokolle auf der Container-Instance erstellt und von Elastic Beanstalk erfasst. Elastic Beanstalk erstellt ein Volume in diesem Verzeichnis für jeden Container, den Sie auf den Container-Standort mounten, in dem Protokolle geschrieben werden.

Sie können auch einen Blick auf Docker werfen, um die laufenden Containern mit `docker ps` anzuzeigen.

```
[ec2-user@ip-10-0-0-117 ~]$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
STATUS            PORTS
fffffd873ada5    nginx:1.7          "nginx -g 'daemon of
hour ago         Up About an hour   443/tcp, 0.0.0.0:80->80/tcp   ecs-eb-dv-example-env-
yckmk5geqrm-2-nginx-proxy-90fce996cc8cbe82800
b894601a1364      php:5-fpm          "php-fpm"
hour ago         Up About an hour   9000/tcp
yckmk5geqrm-2-php-app-cec0918ed1a3a49a8001
09fb19828e38      amazon/amazon-ecs-agent:latest  "/agent"
ago              Up About an hour   127.0.0.1:51678->51678/tcp  ecs-agent
```

Dies zeigt die zwei laufenden Container, die Sie bereitgestellt haben, sowie den Amazon ECS-Container-Agenten, der die Bereitstellung koordiniert hat.

Überprüfen des Amazon ECS-Container-Agenten

Amazon EC2-Instances in einer Multicontainer-Docker-Umgebung auf Elastic Beanstalk führen einen Agentvorgang in einem Docker-Container aus. Dieser Agent stellt eine Verbindung mit dem Amazon ECS Service her, um die Container-Bereitstellungen zu koordinieren. Diese Bereitstellungen werden als

Aufgaben in Amazon ECS ausgeführt, die in Aufgabendefinitionsdateien konfiguriert sind. Elastic Beanstalk erstellt diese Aufgabendefinitionsdateien basierend auf der `Dockerrun.aws.json`, die Sie in ein Quell-Bundle hochladen.

Überprüfen Sie den Status des Container-Agenten mit einer HTTP-Get-Anforderung an `http://localhost:51678/v1/metadata`:

```
[ec2-user@ip-10-0-0-117 ~]$ curl http://localhost:51678/v1/metadata
{
  "Cluster": "eb-dv-example-env-qpoxiguye24",
  "ContainerInstanceArn": "arn:aws:ecs:us-east-2:123456789012:container-
instance/6a72af64-2838-400d-be09-3ab2d836ebcd"
}
```

Diese Struktur zeigt den Namen des Amazon ECS-Clusters und den ARN ([Amazon-Ressourcennamen](#)) der Cluster-Instance (die Amazon EC2-Instance, mit der Sie verbunden sind) an.

Weitere Informationen zur Durchführung einer HTTP-Get-Anforderung finden Sie unter `http://localhost:51678/v1/tasks`:

```
[ec2-user@ip-10-0-0-117 ~]$ curl http://localhost:51678/v1/tasks
{
  "Tasks": [
    {
      "Arn": "arn:aws:ecs:us-east-2:123456789012:task/3ff2bf0f-790d-4f6d-affb-5b127b3b6e4a",
      "DesiredStatus": "RUNNING",
      "KnownStatus": "RUNNING",
      "Family": "eb-dv-example-env-qpoxiguye24",
      "Version": "2",
      "Containers": [
        {
          "DockerId": "b894601a1364a438156a239813c77cdef17040785bc4d5e49349470dc1556b15",
          "DockerName": "ecs-eb-dv-example-env-qpoxiguye24-2-php-app-cec0918ed1a3a49a8001",
          "Name": "php-app"
        },
        {
          "DockerId": "fffffd873ada5f537c88862cce4e1de7ec3edf962645982fb236961c833a5d0fe",
          "DockerName": "ecs-eb-dv-example-env-qpoxiguye24-2-nginx-
proxy-90fce996cc8cbe8b2800",
          "Name": "nginx-proxy"
        }
      ]
    }
  ]
}
```

Diese Struktur beschreibt die Aufgabe, die zur Bereitstellung der beiden Docker-Container aus dem Beispielprojekt dieses Tutorials ausgeführt wird. Die folgenden Informationen werden angezeigt:

- KnownStatus – Der RUNNING-Status gibt an, dass die Container noch aktiv sind.
- Family (Familie) – Der Name der Aufgabendefinition, die Elastic Beanstalk aus `Dockerrun.aws.json` erstellt hat.
- Version – Die Version der Aufgabendefinition. Diese wird mit jeder Aktualisierung der Aufgabendefinitionsdatei erhöht.
- Containers (Container) – Informationen über die Container, die in der Instance ausgeführt werden.

Noch mehr Informationen finden Sie im Amazon ECS Service selbst, den Sie mit der AWS-Befehlszeilenschnittstelle aufrufen. Anweisungen zur Verwendung der AWS CLI mit Amazon ECS und Informationen zu Amazon ECS im Allgemeinen finden Sie im [Amazon ECS-Benutzerhandbuch](#).

Migration zur Docker Amazon Linux 2-Plattform

Important

Amazon Linux 2-Plattformversionen unterscheiden sich grundlegend von Amazon Linux AMI-Plattformversionen (Vorgängerversion von Amazon Linux 2). Diese unterschiedlichen Plattformgenerationen sind auf verschiedene Arten inkompatibel. Wenn Sie zu einer Amazon Linux 2-Plattformversion migrieren, müssen Sie die Informationen in [the section called "Aktualisieren auf Amazon Linux 2" \(p. 503\)](#) lesen.

Sie können Ihre Anwendungen, die auf der [Multicontainer-Docker-Plattform unter Amazon Linux AMI ausgeführt werden \(p. 64\)](#), zur Amazon Linux 2 Docker-Plattform migrieren. Bei der Multicontainer-Docker-Plattform unter Amazon Linux AMI müssen vordefinierte Anwendungimages für die Ausführung als Container angegeben werden. Nach der Migration besteht diese Einschränkung nicht mehr, da die Amazon Linux 2 Docker-Plattform die Erstellung Ihrer Container-Images mit Elastic Beanstalk auch während der Bereitstellung ermöglicht.

Ihre Anwendungen werden weiterhin in Umgebungen mit mehreren Containern ausgeführt – mit den zusätzlichen Vorteilen des Docker Compose-Tools. Weitere Informationen zu Docker Compose und zur Installation von Docker Compose finden Sie auf den Docker-Websites [Overview of Docker Compose](#) und [Install Docker Compose](#).

Die Datei docker-compose.yml

Das Docker Compose-Tool verwendet die docker-compose.yml-Datei für die Konfiguration Ihrer Anwendungsservices. Diese Datei ersetzt die Dockerrun.aws.json v2-Datei in Ihrem Anwendungsprojektverzeichnis und Anwendungsquellpaket. Die docker-compose.yml-Datei muss zwar manuell erstellt werden, die meisten der Parameterwerte können Sie jedoch Ihrer Dockerrun.aws.json v2-Datei entnehmen.

Im Anschluss finden Sie ein Beispiel für eine docker-compose.yml-Datei und die entsprechende Dockerrun.aws.json v2-Datei für die gleiche Anwendung. Weitere Informationen zur docker-compose.yml-Datei finden Sie in der [Referenz zu Compose-Dateien](#). Weitere Informationen zur Dockerrun.aws.json v2-Datei finden Sie unter [Dockerrun.aws.json-v2 \(p. 69\)](#).

docker-compose.yml	Dockerrun.aws.json v2
<pre>version: '2.4' 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</pre>	<pre>{ "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",</pre>

docker-compose.yml	Dockerrun.aws.json v2
<pre>links: - php-app</pre>	<pre>"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" }] }</pre>

Weitere Überlegungen zur Migration

Area	Multi-Container-Docker-Plattform unter Amazon Linux AMI	Docker-Plattform auf Amazon Linux 2 mit Docker Compose
Umgebungseigenschaften	Elastic Beanstalk kann Umgebungseigenschaften direkt an den Container übergeben. Der im Container ausgeführte Code kann auf diese Eigenschaften ohne zusätzliche Konfiguration in Form von Umgebungsvariablen zugreifen.	Damit Ihre Container auf Umgebungseigenschaften zugreifen können, müssen Sie einen Verweis auf die .env-Datei in der docker-compose.yml-Datei hinzufügen. Elastic Beanstalk generiert die .env-Datei und listet jede der Eigenschaften als Umgebungsvariable auf. Weitere Informationen finden Sie unter Verweisen auf Umgebungsvariablen in Containern (p. 86) .
Protokollverzeichnis	Für jeden Container erstellt Elastic Beanstalk ein Protokollverzeichnis namens /var/log/containers/<containernname>. Weitere Informationen finden Sie im Feld mountPoints unter Format der Containerdefinition (p. 71) .	Für jeden Container erstellt Elastic Beanstalk ein Protokollverzeichnis namens /var/log/eb-docker/containers/<service name> (oder \${EB_LOG_BASE_DIR}/<service name>). Weitere Informationen finden Sie unter Angepasste Protokollierung für Docker-Container (Docker Compose) (p. 87) .

Migrationsschritte

So migrieren Sie zur Amazon Linux 2 Docker-Plattform

1. Erstellen Sie auf der Grundlage der vorhandenen `Dockerrun.aws.json` v2-Datei die `docker-compose.yml`-Datei für Ihre Anwendung. Weitere Informationen finden Sie weiter oben im Abschnitt [Die Datei docker-compose.yml \(p. 79\)](#).
2. Ersetzen Sie im Stammverzeichnis Ihres Anwendungsprojektordners die `Dockerrun.aws.json` v2-Datei durch die soeben erstellte `docker-compose.yml`-Datei.

Ihre Verzeichnisstruktur sollte wie folgt aussehen.

```
~/myApplication
|-- docker-compose.yml
|-- .ebextensions
|-- php-app
|-- proxy
```

3. Verwenden Sie den Befehl `eb init`, um Ihr lokales Verzeichnis für die Bereitstellung in Elastic Beanstalk zu konfigurieren.

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

4. Verwenden Sie den Befehl `eb create`, um eine Umgebung zu erstellen und Ihr Docker-Image bereitzustellen.

```
~/myApplication$ eb create environment-name
```

5. Wenn es sich bei Ihrer App um eine Webanwendung handelt, verwenden Sie nach dem Start der Umgebung den Befehl `eb open`, um sie in einem Webbrower anzuzeigen.

```
~/myApplication$ eb open environment-name
```

- Den Status Ihrer neu erstellten Umgebung können Sie mithilfe des Befehls eb status anzeigen.

```
~/myApplication$ eb status environment-name
```

Vorkonfigurierte Docker-Container

Elastic Beanstalk besitzt einen Plattformzweig, auf der ein Docker-Container ausgeführt wird, der mit dem Java EE GlassFish-Anwendungsserver-Softwarestack vorkonfiguriert ist. Sie können den vorkonfigurierten Docker-Container verwenden, um Ihre Anwendung lokal zu entwickeln und zu testen und anschließend in einer Elastic Beanstalk-Umgebung bereitzustellen, die Ihrer lokalen Umgebung entspricht.

Hinweise

- Elastic Beanstalk unterstützt auch Plattformzweige mit vorkonfigurierten Docker-Containern für Go und Python. Diese Plattformzweige sollen ausgemustert werden.
- Alle Zweige der vorkonfigurierten Docker-Plattform verwenden das Amazon Linux AMI-Betriebssystem (Vorgängerversion von Amazon Linux 2). Verwenden Sie zum Migrieren Ihrer GlassFish-Anwendung zu Amazon Linux 2 die generische Docker-Plattform und stellen Sie GlassFish und Ihren Anwendungscode in einem Amazon Linux 2-Docker-Image bereit. Details dazu finden Sie unter [the section called “Tutorial: GlassFish auf Docker: Pfad zu AL2” \(p. 61\)](#).

Der folgende Abschnitt bietet eine detaillierte Anleitung für die Bereitstellung einer Anwendung in Elastic Beanstalk mithilfe eines vorkonfigurierten Docker-Containers.

Details zu den derzeit unterstützten vorkonfigurierten Docker-Plattformversionen finden Sie unter [Vorkonfigurierte Docker-Plattform](#) im Dokument AWS Elastic Beanstalk-Plattformen.

Erste Schritte mit vorkonfigurierten Docker-Containern (?? Amazon Linux AMI ??)

Dieser Abschnitt zeigt die lokale Entwicklung einer Beispielanwendung und die anschließende Bereitstellung in Elastic Beanstalk mit einem vorkonfigurierten Docker-Container.

Einrichten der lokalen Entwicklungsumgebung

Für diese Anleitung verwenden wir eine GlassFish-Beispielanwendung.

Einrichten Ihrer Umgebung

- Erstellen Sie einen neuen Ordner für die Beispielanwendung.

```
~$ mkdir eb-preconf-example  
~$ cd eb-preconf-example
```

- Laden Sie den Beispiel-Anwendungscode in den neuen Ordner herunter.

```
~$ wget https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/samples/docker-glassfish-v1.zip  
~$ unzip docker-glassfish-v1.zip  
~$ rm docker-glassfish-v1.zip
```

Lokales Entwickeln und Testen

Entwickeln einer Beispiel-GlassFish-Anwendung

1. Fügen Sie eine Dockerfile-Datei zum Stammordner der Anwendung hinzu. Geben Sie in der Datei das AWS Elastic Beanstalk-Docker-Basis-Image an, das für die Ausführung Ihres lokalen vorkonfigurierten Docker-Containers verwendet wird. Sie stellen Ihre Anwendung später in einer vorkonfigurierten Plattformversion von Elastic Beanstalk-Docker-Glassfish bereit. Wählen Sie das Docker-Basis-Image, das diese Plattformversion verwendet. Informationen zum aktuellen Docker-Image der Plattformversion finden Sie im Abschnitt [Vorkonfigurierte Docker-Plattform](#) auf der Seite Von Elastic Beanstalk unterstützte Plattformen im Handbuch AWS Elastic Beanstalk-Plattformen.

Example ~/Eb-preconf-example/Dockerfile

```
# For Glassfish 5.0 Java 8
FROM amazon/aws-eb-glassfish:5.0-al-onbuild-2.11.1
```

Weitere Informationen zur Verwendung von Dockerfile finden Sie unter [Docker-Konfiguration \(p. 53\)](#).

2. Erstellen Sie das Docker-Image.

```
~/eb-preconf-example$ docker build -t my-app-image .
```

3. Führen Sie den Docker-Container aus dem Image aus.

Note

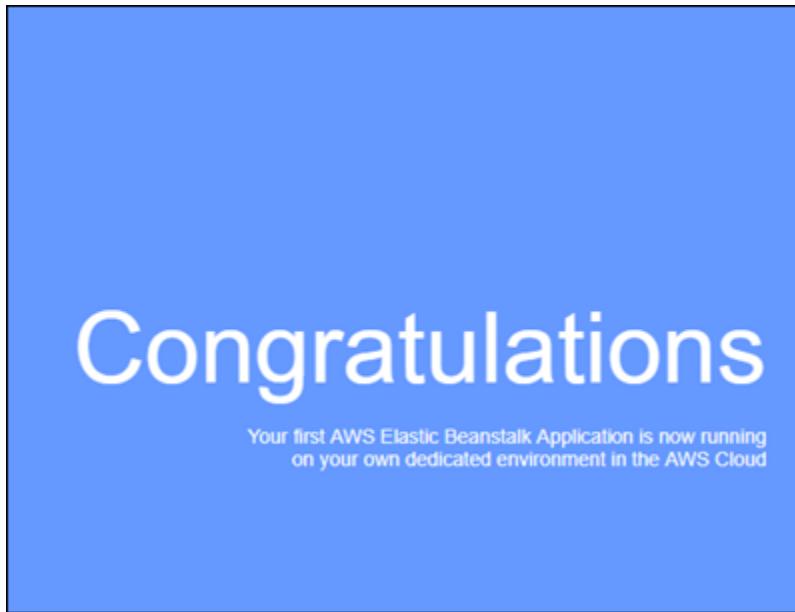
Sie müssen das Flag `-p` einbinden, um Port 8080 auf dem Container dem Localhost-Port 3000 zuzuordnen. In Elastic Beanstalk stellen die Docker-Container die Anwendung stets über Port 8080 auf dem Container bereit. Mit den Flags `-it` wird das Image als interaktiver Prozess ausgeführt. Mithilfe des Flags `--rm` wird das Container-Dateisystem nach Beendigung des Containers bereinigt. Optional können Sie das Flag `-d` einbeziehen und das Image als Daemon ausführen.

```
$ docker run -it --rm -p 3000:8080 my-app-image
```

4. Geben Sie die folgende URL in den Webbrowser ein, um die Beispieldatenanwendung anzuzeigen.

```
http://localhost:3000
```

Die folgende Webseite wird angezeigt.



What's Next?

- [Learn how to build, deploy, and manage your applications on AWS Elastic Beanstalk](#)
- [AWS Elastic Beanstalk concepts](#)
- [Learn how to create new application versions](#)
- [Learn how to manage your application environments](#)

GlassFish Samples

- [Explore community-provided sample applications](#)

Bereitstellen in Elastic Beanstalk

Nachdem Sie die Anwendung getestet haben, können Sie diese nun in Elastic Beanstalk bereitstellen.

So stellen Sie Ihre Anwendung in Elastic Beanstalk bereit

1. Ändern Sie im Stammordner der Anwendung den Dateinamen von `Dockerfile` in `Dockerfile.local`. Dieser Schritt ist erforderlich, damit Elastic Beanstalk die `Dockerfile`-Datei mit den richtigen Anweisungen für Elastic Beanstalk verwendet, um ein angepasstes Docker-Image auf den einzelnen Amazon EC2-Instances in Ihrer Elastic Beanstalk-Umgebung zu erstellen.

Note

Dieser Schritt ist nicht nötig, wenn die `Dockerfile`-Datei Anweisungen enthält, mit denen das Docker-Basis-Image der Plattformversion geändert wird. Sie brauchen überhaupt keine `Dockerfile`-Datei, sofern die `Dockerfile`-Datei nur eine `FROM`-Zeile enthält und darin das Basis-Image angibt, mit dem der Container erstellt werden soll. In diesem Fall ist die `Dockerfile`-Datei überflüssig.

2. Erstellen Sie das Quell-Bundle der Anwendung.

```
~/eb-preconf-example$ zip myapp.zip -r *
```

3. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced.
4. Wählen Sie für Platform (Plattform) unter Preconfigured – Docker (Vorkonfiguriert – Docker) die Option Glassfish aus.
5. Unter Application code wählen Sie Upload your code und dann Upload aus.
6. Wählen Sie Local file (Lokale Datei), wählen Sie Browse (Durchsuchen) und öffnen Sie dann das Quell-Bundle, das Sie erstellt haben.
7. Klicken Sie auf Upload.
8. Klicken Sie auf Review and launch (Überprüfen und starten).
9. Prüfen Sie die verfügbaren Einstellungen und wählen Sie Create app (Anwendung erstellen) aus.

10. Wenn die Umgebung erstellt wurde, können Sie die bereitgestellte Anwendung anzeigen. Wählen Sie die Umgebungs-URL, die oben im Konsolendashboard angezeigt wird.

Konfigurieren von Docker-Umgebungen

Es gibt mehrere Möglichkeiten, das Verhalten Ihrer Elastic Beanstalk-Docker-Umgebung zu konfigurieren.

Note

Wenn Ihre Elastic Beanstalk-Umgebung eine Amazon Linux AMI Docker-Plattformversion verwendet (Vorgängerversion von Amazon Linux 2), lesen Sie unbedingt die zusätzlichen Informationen unter [the section called “Docker-Konfiguration auf Amazon Linux AMI \(Vorgängerversion von Amazon Linux 2\)” \(p. 92\)](#).

Abschnitte

- [Konfigurieren von Software in Docker-Umgebungen \(p. 85\)](#)
- [Verweisen auf Umgebungsvariablen in Containern \(p. 86\)](#)
- [Generieren von Protokollen für erweiterte Zustandsberichte \(Docker Compose\) \(p. 87\)](#)
- [Angepasste Protokollierung für Docker-Container \(Docker Compose\) \(p. 87\)](#)
- [Docker-Images \(p. 88\)](#)
- [Zurückgewinnen von Docker-Speicherplatz \(p. 90\)](#)
- [Konfigurieren von verwalteten Aktualisierungen für Docker-Umgebungen \(p. 91\)](#)
- [Docker-Konfigurations-Namespaces \(p. 91\)](#)
- [Docker-Konfiguration auf Amazon Linux AMI \(Vorgängerversion von Amazon Linux 2\) \(p. 92\)](#)

Konfigurieren von Software in Docker-Umgebungen

Sie können die Elastic Beanstalk-Konsole zum Konfigurieren der Software verwenden, die in den Instances der Umgebung ausgeführt wird.

So konfigurieren Sie Ihre Docker-Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Nehmen Sie die notwendigen Konfigurationsänderungen vor.
6. Wählen Sie Apply (Anwenden) aus.

Hinweise zum Konfigurieren von Software-Einstellungen in einer beliebigen Umgebung finden Sie unter [the section called “Softwareeinstellungen” \(p. 621\)](#). Die folgenden Abschnitte behandeln Docker-spezifische Informationen.

Containeroptionen

Der Abschnitt Container options (Containeroptionen) enthält plattformspezifische Optionen. In Docker-Umgebungen können Sie auswählen, ob Ihre Umgebung den Nginx-Proxy-Server enthält.

Umgebungen mit Docker Compose

Wenn Sie Ihre Docker-Umgebung mit Docker Compose verwalten, geht Elastic Beanstalk davon aus, dass Sie einen Proxy-Server als Container ausführen. Daher wird die Einstellung Proxy-Server standardmäßig auf Keine festgelegt und von Elastic Beanstalk wird keine NGINX-Konfiguration bereitgestellt.

Note

Selbst wenn Sie NGINX als Proxy-Server auswählen, wird diese Einstellung in einer Umgebung mit Docker Compose ignoriert. Die Einstellung Proxy-Server wird trotzdem standardmäßig auf Keine festgelegt.

Da der NGINX-Webserver-Proxy für die Amazon Linux 2-basierte Docker-Plattform mit Docker Compose deaktiviert ist, müssen Sie die Schritte zum Generieren von Protokollen für erweiterte Zustandsberichte ausführen. Weitere Informationen finden Sie unter [Generieren von Protokollen für erweiterte Zustandsberichte \(Docker Compose\) \(p. 87\)](#).

Umgebungseigenschaften und Umgebungsvariablen

Im Bereich Environment Properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon Elastic Compute Cloud-Instances (Amazon EC2) angeben, auf denen die Anwendung ausgeführt wird. Umgebungseigenschaften werden als Schlüssel-Wert-Paare an die Anwendung weitergeleitet. In einer Docker-Umgebung übergibt Elastic Beanstalk Umgebungseigenschaften als Umgebungsvariablen an Container.

Ihr Anwendungscode, der in einem Container ausgeführt wird, kann nach Namen auf eine UmgebungsvARIABLE verweisen und dessen Wert lesen. Der Quellcode, von dem diese UmgebungsvARIABLEN gelesen werden, variiert je nach Programmiersprache. Anweisungen zum Lesen von Werten der UmgebungsvARIABLEN in den Programmiersprachen, die von Elastic Beanstalk verwaltete Plattformen unterstützen, finden Sie im jeweiligen Plattformthema. Eine Liste der Links zu diesen Themen finden Sie unter [the section called “Softwareeinstellungen” \(p. 621\)](#).

Umgebungen mit Docker Compose

Wenn Sie Ihre Docker-Umgebung mit Docker Compose verwalten, sind einige zusätzliche Konfigurationsschritte erforderlich, um die UmgebungsvARIABLEN in den Containern abzurufen. Damit die ausführbaren Dateien, die in Ihrem Container ausgeführt werden, auf diese UmgebungsvARIABLEN zugreifen können, muss in der `docker-compose.yml`-Datei auf sie verwiesen werden. Weitere Informationen finden Sie unter [Verweisen auf UmgebungsvARIABLEN in Containern \(p. 86\)](#).

Verweisen auf UmgebungsvARIABLEN in Containern

Wenn Sie das Docker Compose-Tool auf der Amazon Linux 2-Docker-Plattform verwenden, wird von Elastic Beanstalk im Stammverzeichnis Ihres Anwendungsprojekts eine Docker Compose-Umgebungsdatei mit der Bezeichnung `.env` generiert. In dieser Datei werden die für Elastic Beanstalk konfigurierten UmgebungsvARIABLEN gespeichert.

Note

Wenn Sie eine `.env`-Datei in Ihr Anwendungspaket aufnehmen, wird von Elastic Beanstalk keine `.env`-Datei generiert.

Damit ein Container auf die in Elastic Beanstalk definierten UmgebungsvARIABLEN verweisen kann, muss mindestens einer der folgenden Konfigurationsansätze verwendet werden.

- Fügen Sie die von Elastic Beanstalk generierte `.env`-Datei zur Konfigurationsoption `env_file` in der `docker-compose.yml`-Datei hinzu.

- Definieren Sie die Umgebungsvariablen direkt in der `docker-compose.yml`-Datei.

Im Anschluss finden Sie entsprechende Beispieldateien. Die `docker-compose.yml`-Beispieldatei veranschaulicht beide Ansätze.

- Wenn Sie die Umgebungseigenschaften `DEBUG_LEVEL=1` und `LOG_LEVEL=error` definieren, erstellt Elastic Beanstalk die folgende `.env`-Datei für Sie:

```
DEBUG_LEVEL=1
LOG_LEVEL=error
```

- In dieser `docker-compose.yml`-Datei verweist die Konfigurationsoption `env_file` auf die `.env`-Datei. Außerdem wird die Umgebungsvariable `DEBUG=1` direkt in der `docker-compose.yml`-Datei definiert.

```
services:
  web:
    build: .
    environment:
      - DEBUG=1
    env_file:
      - .env
```

Hinweise

- Wenn Sie die gleiche Umgebungsvariable in beiden Dateien festlegen, hat die in der `docker-compose.yml`-Datei definierte Variable Vorrang vor der Variablen in der `.env`-Datei.
- Achten Sie darauf, dass sich zwischen dem Gleichheitszeichen (=) und dem zugewiesenen Wert Ihrer Variablen keine Leerzeichen befinden. Andernfalls werden der Zeichenfolge Leerzeichen hinzugefügt.

Weitere Informationen zu Umgebungsvariablen in Docker Compose finden Sie unter [Environment variables in Compose](#).

Generieren von Protokollen für erweiterte Zustandsberichte (Docker Compose)

Der [Elastic Beanstalk-Zustandsagent \(p. 827\)](#) stellt Betriebssystem- und Anwendungsintegritätsmetriken für Elastic Beanstalk-Umgebungen bereit. Es nutzt Webserver-Protokollformate, die Informationen in einem bestimmten Format weiterleiten.

Elastic Beanstalk nimmt an, dass Sie einen Webserver-Proxy als Container ausführen. Daher ist der NGINX-Webserver-Proxy für Docker-Umgebungen deaktiviert, in denen Docker Compose ausgeführt wird. Sie müssen Ihren Server so konfigurieren, dass Protokolle an den vom Elastic Beanstalk-Zustandsagenten verwendeten Speicherort und im entsprechenden Format geschrieben werden. Dies ermöglicht die uneingeschränkte Nutzung der erweiterten Zustandsberichte, auch wenn der Webserver-Proxy deaktiviert ist.

Eine entsprechende Anleitung finden Sie unter [Konfiguration von Webserver-Protokollen \(p. 857\)](#).

Anangepasste Protokollierung für Docker-Container (Docker Compose)

Zur effizienten Behandlung von Problemen sowie zur effizienten Überwachung Ihrer containerisierten Services können Sie von Elastic Beanstalk über die Umgebungsverwaltungskonsole oder über die EB CLI [Instance-Protokolle anfordern \(p. 869\)](#). Instance-Protokolle umfassen kombinierte und verpackte Bundle-

Protokolle und Protokollfragmente, um eine effiziente und unkomplizierte Betrachtung von Protokollen und aktuellen Ereignissen zu ermöglichen.

Für jeden in der `docker-compose.yml`-Datei definierten Service wird von Elastic Beanstalk unter `/var/log/eb-docker/containers/<service name>` in der Container-Instance ein Protokollverzeichnis erstellt. Wenn Sie die Docker Compose-Funktion auf der Amazon Linux 2-Docker-Plattform verwenden, können Sie diese Verzeichnisse an dem Speicherort innerhalb der Containerdateistruktur mounten, an dem Protokolle geschrieben werden. Wenn Sie Protokollverzeichnisse zum Schreiben von Protokolldaten mounten, kann Elastic Beanstalk Protokolldaten aus diesen Verzeichnissen erfassen.

Wenn sich Ihre Anwendungen auf einer Docker-Plattform befinden, von der Docker Compose nicht genutzt wird, können Sie das unter [Angepasste Protokollierung für Docker-Container \(Docker Compose\) \(p. 87\)](#) beschriebene Standardverfahren verwenden.

So konfigurieren Sie die Protokolldateien Ihres Service als abrufbare Fragmentdateien und Bundle-Protokolle

1. Bearbeiten Sie die `docker-compose.yml`-Datei.
2. Fügen Sie unter dem Schlüssel `volumes` für Ihren Service ein Bind-Mount hinzu:

```
"${EB_LOG_BASE_DIR}/<service name>:<log directory inside container>
```

In der folgenden `docker-compose.yml`-Beispieldatei gilt Folgendes:

- `nginx-proxy` ist `<Name des Service>`.
- `/var/log/nginx` ist `<Protokollverzeichnis innerhalb des Containers>`.

```
services:  
  nginx-proxy:  
    image: "nginx"  
    volumes:  
      - "${EB_LOG_BASE_DIR}/nginx-proxy:/var/log/nginx"
```

- Das Verzeichnis `var/log/nginx` enthält die Protokolle für den Dienst `nginx-proxy` im Container und wird dem Verzeichnis `/var/log/eb-docker/containers/nginx-proxy` auf dem Host zugeordnet.
- Alle Protokolle in diesem Verzeichnis sind nun als Bundle und als Protokollfragmente über die Elastic Beanstalk-Funktion zum [Anfordern von Instance-Protokollen \(p. 869\)](#) abrufbar.

Hinweise

- `${EB_LOG_BASE_DIR}` ist eine Umgebungsvariable, die von Elastic Beanstalk mit dem Wert `/var/log/eb-docker/containers` versehen wird
- Elastic Beanstalk erstellt das `/var/log/eb-docker/containers/<service name>`-Verzeichnis für jeden Service in der `docker-compose.yml`-Datei.

Docker-Images

Die Docker- und Multicontainer-Docker-Plattformen für Elastic Beanstalk unterstützen die Verwendung von Docker-Images, die in einem öffentlichen oder privaten Online-Image-Repository gespeichert sind.

Geben Sie Images mit Namen in `Dockerrun.aws.json` an. Beachten Sie diese Konventionen:

- Abbilder in offiziellen Repositorys in Docker Hub verwenden einen einzelnen Namen (z. B. `ubuntu` oder `mongo`).

- Images in anderen Repositorys in Docker Hub sind mit einem Organisationsnamen qualifiziert (z. B. `amazon/amazon-ecs-agent`).
- Images in anderen Online-Repositorys sind durch einen Domänennamen zusätzlich qualifiziert (z. B. `quay.io/assemblyline/ubuntu` oder `account-id.dkr.ecr.us-east-2.amazonaws.com/ubuntu:trust`y).

Für Umgebungen, die nur die Docker-Plattform verwenden, können Sie auch während der Erstellung der Umgebung mit einer Docker-Datei ein eigenes Abbild erstellen. Details dazu finden Sie unter [Erstellen benutzerdefinierter Images mit einer Dockerfile-Datei \(p. 57\)](#). Die Multicontainer Docker-Plattform unterstützt diese Funktionalität nicht.

Verwenden von Images aus einem Amazon ECR-Repository

Sie können Ihre benutzerdefinierten Docker-Images in AWS mit [Amazon Elastic Container Registry \(Amazon ECR\)](#) speichern. Wenn Sie Ihre Docker-Images in Amazon ECR speichern, authentifiziert sich Elastic Beanstalk automatisch bei der Amazon ECR-Registry mit dem [Instance-Profil \(p. 22\)](#) Ihrer Umgebung, so dass Sie keine [Authentifizierungsdatei generieren \(p. 90\)](#) und in Amazon Simple Storage Service (Amazon S3) hochladen müssen.

Sie müssen allerdings Ihren Instances die Berechtigung für den Zugriff auf die Images in Ihrem Amazon ECR-Repository erteilen, indem Sie Berechtigungen zum Instance-Profil Ihrer Umgebung hinzufügen. Sie können die verwaltete Richtlinie [AmazonEC2ContainerRegistryReadOnly](#) an das Instance-Profil anfügen, um schreibgeschützten Zugriff auf alle Amazon ECR-Repositorys in Ihrem Konto zu ermöglichen. Sie können auch Zugriff auf einzelne Repositorys gewähren, indem Sie eine benutzerdefinierte Richtlinie gemäß folgender Vorlage erstellen:

```
{  
    "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"  
            ]  
        }  
    ]  
}
```

Ersetzen Sie den Amazon-Ressourcennamen (ARN) in der oben genannten Policy durch den ARN Ihres Repository.

Verweisen Sie in Ihrer `Dockerrun.aws.json`-Datei über die URL auf das Image. Für die [Docker-Plattform \(p. 53\)](#) lautet die URL in der Image-Definition:

```
"Image": {  
    "Name": "account-id.dkr.ecr.us-east-2.amazonaws.com/repository-name:latest",  
    "Update": "true"  
},
```

Verwenden Sie bei der [Multicontainer-Docker-Plattform \(p. 69\)](#) den `image`-Schlüssel in einem Containerdefinitionsobjekt:

```
"containerDefinitions": [  
    {  
        "name": "my-image",  
        "image": "account-id.dkr.ecr.us-east-2.amazonaws.com/repository-name:latest",  
    },
```

Verwenden von Images aus einem privaten Repository

Für die Verwendung eines Docker-Image in einem privaten Repository, das von einem Online-Registry gehostet wird, müssen Sie eine Authentifizierungsdatei angeben, die Informationen enthält, die zur Authentifizierung mit dem Registry erforderlich sind.

Erstellen Sie eine Authentifizierungsdatei mit dem `docker login`-Befehl. Für Repositorys auf Docker Hub führen Sie `docker login` aus:

```
$ docker login
```

Für andere Registrys fügen Sie die URL des Registry-Servers ein:

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

Note

Wenn Ihre Elastic Beanstalk-Umgebung eine Amazon Linux AMI Docker-Plattformversion (Vorgängerversion von Amazon Linux 2) verwendet, lesen Sie die zusätzlichen Informationen unter [the section called “Docker-Konfiguration auf Amazon Linux AMI \(Vorgängerversion von Amazon Linux 2\)” \(p. 92\)](#).

Laden Sie eine Kopie mit dem Namen `.dockercfg` der Authentifizierungsdatei in einen sicheren Amazon S3-Bucket hoch. Der Amazon S3-Bucket muss in derselben AWS-Region gehostet sein wie die Umgebung, die ihn verwendet. Elastic Beanstalk kann keine Dateien von einem Amazon S3-Bucket herunterladen, der in anderen Regionen gehostet wird. Erteilen Sie Berechtigungen für den Vorgang `s3:GetObject` für die IAM-Rolle im Instance-Profil. Details dazu finden Sie unter [Elastic Beanstalk Instance-Profile verwalten \(p. 904\)](#).

Fügen Sie die Amazon S3-Bucket-Informationen zum `Authentication- (v1)` oder `authentication- (v2)` Parameter in der `Dockerrun.aws.json`-Datei hinzu.

Weitere Informationen zum `Dockerrun.aws.json`-Format für Docker-Umgebungen finden Sie unter [Docker-Konfiguration \(p. 53\)](#). Informationen zu Multicontainer-Umgebungen finden Sie unter [Multicontainer-Docker-Konfiguration \(p. 69\)](#).

Weitere Informationen zur Authentifizierungsdatei finden Sie unter [Store images on Docker Hub](#) und `docker login` auf der Docker-Website.

Zurückgewinnen von Docker-Speicherplatz

Docker bereinigt (löscht) den verwendeten Speicherplatz nicht, wenn eine Datei erstellt und dann aus einem laufenden Container gelöscht wird. Der Speicherplatz wird erst zum Pool zurückgegeben, wenn der

Container gelöscht wird. Dies ist ein Problem, wenn ein Container-Prozess viele Dateien, z. B. regelmäßige Dumping-Datenbank-Backups, die den Speicherplatz der Anwendung beanspruchen, erstellt und löscht.

Eine Lösung ist, die Größe des Speicherplatzes der Anwendung zu erhöhen, wie im vorherigen Abschnitt beschrieben. Die andere Möglichkeit führt zu schlechterer Leistung: Führen Sie `fstrim` regelmäßig über Speicherplatz ohne Container auf dem Host-BS aus, z. B. `cron`, um die ungenutzten Containerdatenblöcke zurückzugewinnen.

```
docker ps -q | xargs docker inspect --format='{{ .State.Pid }}' | xargs -IZ sudo fstrim /proc/{Z}/root/
```

Konfigurieren von verwalteten Aktualisierungen für Docker-Umgebungen

Mit [verwalteten Plattformaktualisierungen \(p. 496\)](#) können Sie die Umgebung so konfigurieren, dass automatisch nach einem Zeitplan eine Aktualisierung auf die neueste Version einer Plattform durchgeführt wird.

Im Fall von Docker-Umgebungen können Sie entscheiden, ob eine automatische Plattformaktualisierung innerhalb von Docker-Versionen stattfinden soll – falls die neue Version der Plattform eine neue Docker-Version enthält. Elastic Beanstalk unterstützt verwaltete Plattform-Updates für alle Docker-Versionen, wenn von einer Umgebung aus aktualisiert wird, auf der eine Docker-Plattformversion neuer als 2.9.0 läuft. Wenn eine neue Plattformversion eine neue Version von Docker enthält, inkrementiert Elastic Beanstalk die Nebenversionsnummer der Aktualisierung. Um verwaltete Plattformaktualisierungen für Docker-Versionen zuzulassen, aktivieren Sie deshalb verwaltete Plattformaktualisierungen für Nebenversions- und Patch-Versionsaktualisierungen. Um verwaltete Plattformaktualisierungen innerhalb von Docker-Versionen zu verhindern, aktivieren Sie verwaltete Plattformaktualisierungen nur für Patch-Versionsaktualisierungen.

Die folgende [Konfigurationsdatei \(p. 722\)](#) beispielsweise ermöglicht verwaltete Plattformaktualisierungen jeden Dienstag um 9:00 Uhr UTC für Neben- und Patch-Versionsaktualisierungen und lässt damit verwaltete Aktualisierungen innerhalb von Docker-Versionen zu:

Example `.ebextensions/managed-platform-update.config`

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: minor
```

Für Umgebungen mit Docker-Plattformversionen 2.9.0 oder früher führt Elastic Beanstalk nie verwaltete Plattformaktualisierungen durch, wenn die neue Plattformversion eine neue Docker-Version enthält.

Docker-Konfigurations-Namespaces

Mithilfe einer [Konfigurationsdatei \(p. 722\)](#) können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Konfigurationsoptionen können durch den Elastic Beanstalk-Service oder die verwendete Plattform definiert und in Namespaces organisiert werden.

Note

Diese Informationen gelten nur für Docker-Umgebungen, in denen Docker Compose nicht ausgeführt wird. In Docker-Umgebungen, in denen Docker Compose ausgeführt wird, weist diese Option ein anderes Verhalten auf. Weitere Informationen zu Proxy-Services mit Docker Compose finden Sie unter [Containeroptionen \(p. 86\)](#).

Die Docker-Plattform unterstützt neben den [unterstützten Optionen für alle Elastic Beanstalk-Umgebungen \(p. 666\)](#) auch Optionen in den folgenden Namespaces:

- `aws:elasticbeanstalk:environment:proxy` – Wählen Sie den Proxy-Server für Ihre Umgebung aus. Docker unterstützt entweder die Ausführung von Nginx oder keinen Proxy-Server.

In der folgenden Beispielkonfigurationsdatei wird eine Docker-Umgebung so konfiguriert, dass kein Proxy-Server ausgeführt wird.

Example `.ebextensions/docker-settings.config`

```
option_settings:  
  aws:elasticbeanstalk:environment:proxy:  
    ProxyServer: none
```

Docker-Konfiguration auf Amazon Linux AMI (Vorgängerversion von Amazon Linux 2)

Wenn Ihre Elastic Beanstalk Docker-Umgebung eine Amazon Linux AMI-Plattformversion (Vorgängerversion von Amazon Linux 2) verwendet, lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

Verwenden einer Authentifizierungsdatei für ein privates Repository

Diese Informationen sind für Sie relevant, wenn Sie [Bilder aus einem privaten Repository verwenden \(p. 90\)](#). Beginnend mit der Docker-Version 1.7 hat der Befehl docker login den Namen der Authentifizierungsdatei und das Format der Datei geändert. Amazon Linux AMI Docker-Plattformversionen (Vorgängerversion von Amazon Linux 2) erfordern die Konfigurationsdatei für ein älteres `~/.dockercfg` Format.

Ab Docker-Version 1.7 und höher erstellt der docker login-Befehl die Authentifizierungsdatei in `~/.docker/config.json` im folgenden Format:

```
{  
  "auths": {  
    "server": {  
      "auth": "key"  
    }  
  }  
}
```

Ab Docker-Version 1.6.2 und früher erstellt der docker login-Befehl die Authentifizierungsdatei in `~/.dockercfg` im folgenden Format:

```
{  
  "server":  
  {  
    "auth" : "auth_token",  
    "email" : "email"  
  }  
}
```

Wenn Sie eine `config.json`-Datei konvertieren möchten, entfernen Sie den äußeren `auths`-Schlüssel, fügen Sie einen `email`-Schlüssel hinzu und reduzieren Sie das JSON-Dokument, um dem alten Format zu entsprechen.

Bei Amazon Linux 2 Docker-Plattformversionen verwendet Elastic Beanstalk den Namen und das Format der neueren Authentifizierungsdatei. Wenn Sie eine Amazon Linux 2-Docker-Plattformversion verwenden, können Sie die vom docker login-Befehl erstellte Authentifizierungsdatei ohne Konvertierung nutzen.

Konfigurieren von zusätzlichen Speicher-Volumes

Zur Verbesserung der Amazon Linux AMI-Leistung konfiguriert Elastic Beanstalk zwei Amazon EBS-Speicher-Volumes für die Amazon EC2-Instances Ihrer Docker-Umgebung. Zusätzlich zum Stamm-Volume, das für alle Elastic Beanstalk-Umgebungen bereitgestellt wird, wird ein zweites 12-GB-Volume mit dem Namen `xvdcz` für die Image-Speicherung auf Docker-Umgebungen verfügbar gemacht.

Wenn Sie zusätzlichen Speicherplatz oder mehr IOPS für Docker-Images benötigen, können Sie das Image-Speicher-Volume mit der `BlockDeviceMapping`-Konfigurationsoption im `aws:autoscaling:launchconfiguration` (p. 668)-Namespace anpassen.

Die folgende [Konfigurationsdatei](#) (p. 722) erhöht die Größe des Speicher-Volumes beispielsweise auf 100 GB mit 500 bereitgestellten IOPS:

Example .ebextensions/blockdevice-xvdcz.config

```
option_settings:  
  aws:autoscaling:launchconfiguration:  
    BlockDeviceMappings: /dev/xvdcz=:100::io1:500
```

Wenn Sie die `BlockDeviceMappings`-Option zur Konfiguration zusätzlicher Volumes für Ihre Anwendung verwenden, sollten Sie eine Zuordnung für `xvdcz` einschließen, um sicherzustellen, dass es erstellt wurde. Im folgenden Beispiel werden zwei Volumes konfiguriert, das Image-Speicher-Volume `xvdcz` mit Standardeinstellungen und ein zusätzliches 24-GB-Anwendungs-Volume mit dem Namen `sdh`:

Example .ebextensions/blockdevice-sdh.config

```
option_settings:  
  aws:autoscaling:launchconfiguration:  
    BlockDeviceMappings: /dev/xvdcz=:12:true:gp2,/dev/sdh=:24
```

Note

Wenn Sie die Einstellungen in diesem Namespace ändern, ersetzt Elastic Beanstalk alle Instances in Ihrer Umgebung durch Instances, die mit der neuen Konfiguration ausgeführt werden. Details dazu finden Sie unter [Konfigurationsänderungen](#) (p. 483).

Lokales Ausführen einer Docker-Umgebung mit der EB-CLI

Sie können die Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI) verwenden, um die Docker-Container, die in Ihrer AWS Elastic Beanstalk-Anwendung konfiguriert sind, lokal auszuführen. Die EB CLI verwendet die Docker-Konfigurationsdatei (Dockerfile oder Dockerrun.aws.json) und Quellcode in Ihrem Projektverzeichnis, um Ihre Anwendung lokal in Docker auszuführen.

Die EB-CLI unterstützt lokal ausgeführte Anwendungen, die mit den Plattformen Docker, Multicontainer Docker und Preconfigured Docker definiert wurden.

Themen

- [Voraussetzungen für das lokale Ausführen von Docker-Anwendungen](#) (p. 94)
- [Vorbereiten einer Docker-Anwendung zur Verwendung mit der EB-CLI](#) (p. 94)
- [Lokale Ausführung einer Docker-Anwendung](#) (p. 95)
- [Bereinigen nach dem Ausführen einer laufenden Docker-Anwendung](#) (p. 96)

Voraussetzungen für das lokale Ausführen von Docker-Anwendungen

- Linux OS oder Mac OS X
- [EB CLI-Version 3.3 oder höher \(p. 1009\)](#)

Führen Sie `eb init` im Projektverzeichnis aus, um ein EB CLI-Repository zu initialisieren. Wenn Sie die EB CLI nicht zuvor verwendet haben, finden Sie Informationen unter [Elastic Beanstalk-Umgebungen mit der EB CLI verwalten \(p. 1020\)](#).

- [Docker-Version 1.6 oder höher](#)

Fügen Sie sich selbst zur `docker`-Gruppe hinzu, melden Sie sich ab und wieder an, um sicherzustellen, dass Sie Docker-Befehle ohne `sudo` ausführen können:

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

Führen Sie `docker ps` aus, um zu überprüfen, dass der Docker-Daemon ausgeführt wird:

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

- Eine Docker-Anwendung

Wenn Sie nicht über eine Docker-Anwendung in einem Projektordner auf Ihrem lokalen Computer verfügen, finden Sie unter [Bereitstellen von Elastic Beanstalk-Anwendungen aus Docker-Containern \(p. 46\)](#) eine Einführung in die Verwendung von Docker mit AWS Elastic Beanstalk.

- Docker-Profil (optional)

Wenn Ihre Anwendung Docker-Images verwendet, die sich in einem privaten Repository befinden, führen Sie `docker login` aus und befolgen Sie die Anweisungen zum Erstellen eines Authentifizierungsprofils.

- `w3m` (optional)

`W3m` ist ein Webbrowser, den Sie verwenden können, um Ihre laufende Webanwendung in einem Befehlszeilen-Terminal mit `eb local run` anzuzeigen. Wenn Sie die Befehlszeile in einer Desktop-Umgebung verwenden, brauchen Sie `w3m` nicht.

Docker-Container werden lokal ausgeführt, ohne dass AWS-Ressourcen emuliert werden, die zur Verfügung gestellt werden, wenn Sie eine Anwendung in Elastic Beanstalk bereitstellen, darunter Sicherheitsgruppen und Daten oder Worker-Tiers.

Sie können Ihre lokalen Container so konfigurieren, dass sie eine Verbindung mit einer Datenbank herstellen, indem die erforderliche Verbindungszeichenfolge oder andere Variablen mit der Option `envvars` übergeben werden. Sie müssen jedoch sicherstellen, dass alle Ressourcen in AWS von Ihrem lokalen Computer aus zugänglich sind, indem Sie die [entsprechenden Ports](#) in ihren zugewiesenen Sicherheitsgruppen öffnen oder ein [Standard-Gateway](#) oder eine [Elastic IP-Adresse](#) anfügen.

Vorbereiten einer Docker-Anwendung zur Verwendung mit der EB-CLI

Bereiten Sie Ihre Docker-Konfigurationsdatei und Quell-Daten so vor, als ob Sie sie in Elastic Beanstalk bereitstellen würden. In diesem Thema werden die PHP und das nginx-Proxy-Beispiel aus dem vorherigen

Multicontainer Docker-Tutorial (p. 73) als Beispiel verwendet, Sie können aber die gleichen Befehle mit einem jeder Docker-, Multicontainer Docker- oder Preconfigured Docker-Anwendung nutzen.

Lokale Ausführung einer Docker-Anwendung

Führen Sie Ihre Docker-Anwendung lokal mit dem eb local run-Befehl im Projekt-Verzeichnis aus:

```
~/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
```

Die EB CLI liest die Docker-Konfiguration und führt die Docker-Befehle aus, die erforderlich sind, um Ihre Anwendung auszuführen. Bei der ersten lokalen Ausführung eines Projekts lädt Docker Images aus einem Remote-Repository herunter und speichert sie auf Ihrem lokalen Computer. Dieser Vorgang kann einige Minuten dauern.

Note

Der eb local run-Befehl nimmt zwei optionale Parameter: port und envvars.

Um den Standard-Port für eine Docker-Anwendung zu übergehen, verwenden Sie die Option port:

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

Dieser Befehl teilt der EB CLI mit, den Port 8080 auf dem Host zu verwenden und dem offenen Port auf dem Container zuzuordnen. Wenn Sie keinen Port angeben, verwendet der EB CLI den Container-Port für den Host. Diese Option funktioniert nur mit Anwendungen, die die Docker-Plattform verwenden.

Um die Umgebungsvariablen an die Anwendungs-Container zu übergeben, verwenden Sie die Option envvars:

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

Verwenden Sie Umgebungsvariablen zum Konfigurieren einer Datenbankverbindung, Einrichten von Debug-Optionen oder sicheren Übergeben von Geheimnissen an Ihre Anwendung. Weitere Informationen zu den Optionen, die von den eb local-Unterbefehlen unterstützt werden, finden Sie unter eb local (p. 1071).

Nachdem die Container in Docker laufen, können Sie sie Anforderungen von Clients entgegennehmen. Der eb local-Prozess bleibt offen, solange die Container ausgeführt werden. Wenn Sie den Prozess und die Container stoppen müssen, drücken Sie Strg+C.

Öffnen Sie ein zweites Terminal für die Ausführung zusätzlicher Befehle, während der eb local-Prozess ausgeführt wird. Verwenden Sie eb local status, um den Status Ihrer Anwendung anzuzeigen:

```
~/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
```

Sie können `docker ps` verwenden, um den Status der Container aus Docker-Sicht anzuzeigen:

```
~/project$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
6a8e71274fed      nginx:latest       "nginx -g 'daemon of 9 minutes ago
minutes           0.0.0.0:80->80/tcp, 443/tcp   elasticbeanstalk_nginxproxy_1
82cbf620bd1       php:fpm           "php-fpm"          9 minutes ago     Up 9
minutes           9000/tcp          elasticbeanstalk_phpapp_1
```

Anschließend zeigen Sie Ihre Anwendung mit `eb local open` in Aktion an:

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

Dieser Befehl öffnet Ihre Anwendung im Standard-Webbrowser. Wenn Sie ein Terminal in einer Desktop-Umgebung ausführen, kann Firefox, Safari oder Google Chrome gewählt werden. Wenn Sie ein Terminal in einer Headless-Umgebung oder über eine SSH-Verbindung ausführen, wird ein Befehlszeilen-Browser, wie z. B. w3m, verwendet, sofern vorhanden.

Wechseln Sie für einen Moment zurück zum Terminal, auf dem der Anwendungsprozess ausgeführt wird, und notieren Sie sich die zusätzliche Ausgaben:

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

Dies zeigt, dass die Webanwendung im Docker-Container eine HTTP GET-Anforderung für `index.php` erhalten hat, die erfolgreich mit einem 200-Status (ohne Fehler) zurückgegeben wurde.

Führen Sie `eb local logs` aus, um zu sehen, wo EB CLI die Protokolle schreibt.

```
~/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.
```

Bereinigen nach dem Ausführen einer laufenden Docker-Anwendung

Wenn Sie die lokalen Tests Ihrer Anwendung abgeschlossen haben, können Sie die Anwendungen stoppen und die Images entfernen, die von Docker bei der Verwendung von `eb local run` heruntergeladen wurden. Das Entfernen der Images ist optional. Heben Sie sie für die spätere Verwendung auf.

Kehren Sie zum Terminal zurück, auf dem der Prozess `eb local` ausgeführt wird, und drücken Sie Strg+C, um die Anwendung zu stoppen:

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

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

Die EB CLI versucht, jeden laufenden Container elegant mit Docker-Befehlen zu stoppen. Wenn Sie einen Prozess nicht sofort stoppen müssen, drücken Sie Strg+C erneut.

Nachdem Sie die Anwendungen gestoppt haben, sollten die Docker-Container auch angehalten werden. Überprüfen Sie dies mit `docker ps`:

\$ docker ps --all	CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
		PORTS	NAMES		
	73d515d99d2a	nginx:latest	"nginx -g 'daemon off;' 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		

Die Option `all` zeigt angehaltene Container (falls Sie diese Option ausgelassen haben, ist die Ausgabe leer). Im vorangegangenen Beispiel zeigt Docker, dass beide Containern mit einem 0-Status (keine Fehler) beendet wurden.

Wenn Sie keine Docker- und lokalen EB CLI-Befehle mehr verwenden, können Sie die Docker-Images aus dem lokalen Computer entfernen, um Platz zu sparen.

So entfernen Sie Docker-Images von Ihrem lokalen Computer

1. Zeigen Sie die Images an, die Sie mit `docker images` heruntergeladen haben:

\$ 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. Entfernen Sie die beiden Docker-Container mit `docker rm`:

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

3. Entfernen Sie die Images mit `docker rmi`:

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

Erstellen und Bereitstellen von Go-Anwendungen in Elastic Beanstalk

Themen

- [Erste Schritte mit Go in Elastic Beanstalk \(p. 98\)](#)
- [Einrichten der Go-Entwicklungsumgebung \(p. 100\)](#)
- [Verwenden der Elastic Beanstalk Go-Plattform \(p. 101\)](#)
- [Bereitstellen einer Go-Anwendung in Elastic Beanstalk \(p. 106\)](#)

AWS Elastic Beanstalk for Go erleichtert die Bereitstellung, Verwaltung und Skalierung Ihrer Go-Webanwendungen mit Amazon Web Services. Elastic Beanstalk for Go steht jedem zur Verfügung, der eine Webanwendung mit Go entwickelt oder hostet. Dieses Kapitel enthält Schritt-für-Schritt-Anleitungen für das Bereitstellen Ihrer Webanwendung in Elastic Beanstalk.

Nach der Bereitstellung der Elastic Beanstalk-Anwendung können Sie die EB CLI weiterhin zum Verwalten der Anwendung und der Umgebung verwenden. Alternativ können Sie aber auch die Elastic Beanstalk-Konsole, die AWS CLI oder APIs nutzen.

In den Themen in diesem Kapitel wird davon ausgegangen, dass Sie über einige Kenntnisse der Elastic Beanstalk-Umgebungen verfügen. Wenn Sie Elastic Beanstalk noch nicht verwendet haben, machen Sie sich im [Tutorial "Erste Schritte" \(p. 3\)](#) mit den Grundlagen vertraut.

Erste Schritte mit Go in Elastic Beanstalk

Für die ersten Schritte mit Go-Anwendungen in AWS Elastic Beanstalk benötigen Sie nur das [Quell-Bundle \(p. 412\)](#) einer Anwendung, damit Sie die erste Anwendungsversion hochladen und in einer Umgebung bereitstellen können. Im Rahmen der Umgebungserstellung werden von Elastic Beanstalk alle erforderlichen AWS-Ressourcen für eine hoch skalierbare Webanwendung zugewiesen.

Starten einer Umgebung mit einer Go-Beispielanwendung

Elastic Beanstalk stellt für jede Plattform Beispielanwendungen bereit, die eine Seite enthalten. Elastic Beanstalk bietet auch komplexere Beispiele, die die Verwendung weiterer AWS-Ressourcen wie Amazon RDS sowie sprach- oder plattformspezifische Funktionen und APIs demonstrieren.

Beispiele

Unterstützte Konfigurationen	Umgebungs	Quell-Bundle	Beschreibung
Go	Webserver	go.zip	Anwendung mit einer Seite.

Laden Sie die Beispielanwendung herunter und stellen Sie diese in Elastic Beanstalk bereit. Führen Sie dazu die folgenden Schritte aus.

So starten Sie eine Umgebung mit einer Beispielanwendung (Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Applications (Anwendungen) und anschließend in der Liste den Namen einer vorhandenen Anwendung aus. Sie können auch [eine Anwendung erstellen \(p. 403\)](#).
3. Wählen Sie auf der Anwendungsübersichtsseite die Option Create a new environment (Neue Umgebung erstellen).

Environment name	Health	Date created	Last modified	URL	Platform
GettingStartedApp-env	OK	2020-01-28 12:06:50 UTC-0800	2020-01-30 15:02:35 UTC-0800	GettingStartedApp-env.bn7dx222kw.us-east-2.elasticbeanstalk.com	Tomcat running on Linux
GettingStartedApp-Worker	OK	2020-01-28 16:34:29 UTC-0800	2020-01-28 16:38:20 UTC-0800	GettingStartedApp-Worker.bn7dx222kw.us-east-2.elasticbeanstalk.com	IIS 10.0 running on Windows

- Wählen Sie die Web server environment (Webserverumgebung) oder die Worker environment (Worker-Umgebung) als [Umgebungsebene \(p. 14\)](#) aus. Nach der Erstellung lässt sich die Umgebungsebene nicht mehr ändern.

Note

Die [.NET-Plattform auf Windows Server \(p. 189\)](#) unterstützt die Worker-Umgebungsschicht nicht.

Select environment tier

AWS Elastic Beanstalk has two types of environment tiers to support different types of web applications. Web server environments run a website, web application, or web API that serves HTTP requests. Worker environments run a specialized application that performs a background processing task that listens for messages on an Amazon SQS queue. Worker applications post the results of their processing back to your application by using HTTP.

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](#)

- Wählen Sie in Platform (Plattform) die Plattform und den Plattformzweig aus, die/der mit der Sprache Ihrer Anwendung übereinstimmt.

Note

Elastic Beanstalk unterstützt mehrere [Versionen \(p. 32\)](#) für die meisten Plattformen, die aufgelistet sind. Standardmäßig wählt die Konsole die empfohlene Version für die gewählte Plattform und Plattformvariante. Wenn Ihre Anwendung eine andere Version erfordert, können

Sie diese hier auswählen oder Configure more options (Weitere Optionen konfigurieren) auswählen wie in Schritt 7 beschrieben. Informationen zu unterstützten Plattformversionen finden Sie unter [the section called “Unterstützte Plattformen” \(p. 32\)](#).

6. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
7. Wählen Sie Configure more options (Weitere Optionen konfigurieren) aus, um Ihre Umgebung weiter anzupassen. Sie können die folgenden Optionen nur während des Erstellens der Umgebung festlegen:
 - Environment name
 - Domänenname
 - Plattformversion
 - VPC
 - Stufe

Sie können die folgenden Einstellungen nach der Umgebungserstellung zwar ändern, jedoch müssen dafür neue Instances oder andere Ressourcen bereitgestellt werden, deren Anwendung viel Zeit in Anspruch nehmen kann:

- Instance-Typ, Stamm-Volume, Schlüsselpaar und AWS Identity and Access Management (IAM)-Rolle
- Interne Amazon RDS-Datenbank
- Load Balancer

Weitere Informationen zu allen verfügbaren Einstellungen finden Sie unter [Der Assistent zum Erstellen einer neuen Umgebung \(p. 437\)](#).

8. Wählen Sie Create environment (Umgebung erstellen) aus.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine andere Anwendung bereitstellen. Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine EC2-Instances bereitgestellt oder neu gestartet werden müssen.

Wenn Sie ein oder zwei Beispielanwendungen bereitgestellt haben, können Sie mit der lokalen Entwicklung und Ausführung von Go-Anwendungen beginnen. Informationen dazu finden Sie unter [Einrichten der Go-Entwicklungsumgebung \(p. 100\)](#).

Einrichten der Go-Entwicklungsumgebung

Richten Sie eine Go-Entwicklungsumgebung ein, um Ihre Anwendung lokal zu testen, bevor Sie sie in AWS Elastic Beanstalk bereitstellen. In diesem Thema werden die Einrichtungsschritte für Ihre Entwicklungsumgebung beschrieben. Zudem enthält es Links zu den Installationsseiten nützlicher Tools.

Allgemeine Einrichtungsschritte und Tools für alle Sprachen finden Sie unter [Konfigurieren des Entwicklungscomputers für die Verwendung mit Elastic Beanstalk \(p. 1005\)](#).

Installieren von Go

Zum lokalen Ausführen von Go-Anwendungen installieren Sie Go. Wenn Sie keine bestimmte Version benötigen, laden Sie die neueste Version herunter, die Elastic Beanstalk unterstützt. Eine Liste der unterstützten Versionen finden Sie unter [Go im Dokument zu AWS Elastic Beanstalk-Plattformen](#).

Laden Sie Go unter <https://golang.org/doc/install> herunter.

Installieren des AWS SDK for Go

Wenn Sie die AWS-Ressourcen aus der Anwendung heraus verwalten möchten, installieren Sie das AWS SDK for Go anhand des folgenden Befehls.

```
$ go get github.com/aws/aws-sdk-go
```

Weitere Informationen finden Sie unter [AWS SDK for Go](#).

Verwenden der Elastic Beanstalk Go-Plattform

Important

Amazon Linux 2-Plattformversionen unterscheiden sich grundlegend von Amazon Linux AMI-Plattformversionen (Vorgängerversion von Amazon Linux 2). Diese unterschiedlichen Plattformgenerationen sind auf verschiedene Arten inkompatibel. Wenn Sie zu einer Amazon Linux 2-Plattformversion migrieren, müssen Sie die Informationen in [the section called "Aktualisieren auf Amazon Linux 2" \(p. 503\)](#) lesen.

Mit AWS Elastic Beanstalk können Sie Go-basierte Webanwendungen ausführen, erstellen und konfigurieren. Bei einfachen Go-Anwendungen gibt es zwei Möglichkeiten, um Ihre Anwendung bereitzustellen:

- Stellen Sie ein Quell-Bundle mit einer Quelldatei im Stammverzeichnis namens `application.go` bereit, das das Hauptpaket für Ihre Anwendung enthält. Elastic Beanstalk entwickelt den binären Wert mithilfe des folgenden Befehls:

```
go build -o bin/application application.go
```

Nachdem die Anwendung erstellt wurde, startet Elastic Beanstalk sie auf Port 5000.

- Stellen Sie ein Quell-Bundle mit einer Binärdatei namens `application` bereit. Die Binärdatei befindet sich entweder im Stammverzeichnis des Quell-Bundle oder im `bin/-`-Verzeichnis des Quell-Bundle. Wenn Sie die `application`-Binärdatei in beiden Speicherorten platzieren, verwendet Elastic Beanstalk die Datei im `bin/-`-Verzeichnis.

Elastic Beanstalk startet diese Anwendung auf Port 5000.

In beiden Fällen können Sie mit Go 1.11 oder höher auch Modulanforderungen in einer Datei namens `go.mod` angeben. Weitere Informationen finden Sie unter [Migrating to Go Modules](#) im Go-Blog.

Bei komplexeren Go-Anwendungen gibt es zwei Möglichkeiten, um Ihre Anwendung bereitzustellen:

- Stellen Sie ein Quell-Bundle bereit, das Ihre Anwendungsquelldateien zusammen mit einer [Buildfile \(p. 104\)](#) und einer [Procfile \(p. 103\)](#) enthält. Die Buildfile enthält einen Befehl zum Erstellen der Anwendung und die Procfile enthält Anweisungen zum Ausführen der Anwendung.
- Stellen Sie ein Quell-Bundle bereit, das Ihre Binärdateien der Anwendung mit einer Procfile enthält. Die Procfile enthält Anweisungen zur Ausführung der Anwendung.

Die Go-Plattform umfasst einen Proxy-Server, der statische Komponenten bereitstellt und den Datenverkehr an Ihre Anwendung weiterleitet. Sie können die für erweiterte Szenarien [Standard-Proxy-Konfiguration erweitern oder überschreiben \(p. 105\)](#).

Details zu den verschiedenen Möglichkeiten zum Erweitern einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter [the section called "Erweitern von Linux-Plattformen" \(p. 34\)](#).

Konfigurieren Ihrer Go-Umgebung

Mit den Einstellungen der Go-Plattform können Sie das Verhalten Ihrer Amazon EC2-Instances optimieren. Sie können die Konfiguration der Amazon EC2-Instance der Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-Konsole bearbeiten.

Mithilfe der Elastic Beanstalk-Konsole aktivieren Sie die Protokollrotation an Amazon S3 und konfigurieren Variablen, die die Anwendung aus der Umgebung auslesen kann.

So konfigurieren Sie Ihre Go-Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) –Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren) – Gibt an, ob die Protokolldateien für die Amazon EC2-Instances der Anwendung in den zugeordneten Amazon S3-Bucket kopiert werden sollen.

Statische Dateien

Um die Leistung zu verbessern, können Sie im Abschnitt Static files (Statische Dateien) den Proxy-Server so konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zur Verzeichniszuordnung fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, statt die Anfrage an Ihre Anwendung weiterzuleiten.

Weitere Informationen zum Konfigurieren statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter [the section called “Statische Dateien” \(p. 776\)](#).

Umgebungseigenschaften

Im Bereich Environment Properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Umgebungseigenschaften werden als Schlüssel-Wert-Paare an die Anwendung weitergeleitet.

Innerhalb der Go-Umgebung, die in Elastic Beanstalk ausgeführt wird, können Sie über die Funktion `os.Getenv` auf Umgebungsvariablen zugreifen. Beispielsweise können Sie mit folgendem Code eine Eigenschaft mit dem Namen `API_ENDPOINT` als Variable lesen:

```
endpoint := os.Getenv("API_ENDPOINT")
```

Weitere Informationen finden Sie unter [Umgebungseigenschaften und andere Softwareeinstellungen \(p. 621\)](#).

Go-Konfigurations-Namespace

Mithilfe einer [Konfigurationsdatei \(p. 722\)](#) können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Konfigurationsoptionen können durch den Elastic Beanstalk-Service oder die verwendete Plattform definiert und in Namespaces organisiert werden.

Die Go-Plattform definiert keine plattformspezifischen Namespaces. Sie können den Proxy so konfigurieren, dass statische Dateien mittels des Namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` bereitgestellt werden. Einzelheiten und ein Beispiel finden Sie unter [the section called "Statische Dateien" \(p. 776\)](#).

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Die Amazon Linux AMI (Vorgängerversion von Amazon Linux 2) Go-Plattform

Wenn Ihre Elastic Beanstalk Go-Umgebung eine Amazon Linux AMI-Plattformversion verwendet (Vorgängerversion von Amazon Linux 2), lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

Go-Konfigurations-Namespaces

Mithilfe einer [Konfigurationsdatei \(p. 722\)](#) können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Konfigurationsoptionen können durch den Elastic Beanstalk-Service oder die verwendete Plattform definiert und in Namespaces organisiert werden.

Die Amazon Linux AMI-Go-Plattform unterstützt zusätzlich zu den [Namespaces, die von allen Plattformen unterstützt werden \(p. 666\)](#), eine weitere Konfigurationsoption für einen plattformspezifischen Namespace. Mit dem Namespace `aws:elasticbeanstalk:container:golang:staticfiles` können Sie Optionen definieren, um für die Webanwendung Pfade zu Ordnern im Quell-Bundle der Anwendung zuzuweisen, die statische Inhalte enthalten.

Diese [Konfigurationsdatei \(p. 722\)](#) weist den Proxy-Server beispielsweise an, Dateien im `staticimages`-Ordner im Pfad `/images` bereitzustellen:

Example `.ebextensions/go-settings.config`

```
option_settings:  
  aws:elasticbeanstalk:container:golang:staticfiles:  
    /html: statichtml  
    /images: staticimages
```

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Konfigurieren der Anwendung mit einer Procfile-Datei

Um benutzerdefinierte Befehle zum Starten einer Go-Anwendung anzugeben, schließen Sie eine Datei mit dem Namen `Procfile` in das Stammverzeichnis Ihres Quell-Bundle ein.

Einzelheiten zum Schreiben und Verwenden eines `Procfile` finden Sie im Abschnitt `Buildfile` und `Procfile` unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#).

Example Procfile

```
web: bin/server
queue_processor: bin/queue_processor
foo: bin/fooapp
```

Sie müssen die Hauptanwendung `web` aufrufen und als ersten Befehl in Ihrem `Procfile` auflisten. Elastic Beanstalk stellt die `web`-Hauptanwendung auf der Stamm-URL der Umgebung bereit, z. B. `http://my-go-env.elasticbeanstalk.com`.

Elastic Beanstalk führt außerdem alle Anwendungen aus, deren Namen kein `web_`-Präfix haben. Diese Anwendungen sind jedoch nicht von außerhalb Ihrer Instance verfügbar.

Elastic Beanstalk erwartet von Prozessen, die aus der `Procfile`-Datei ausgeführt werden, dass sie kontinuierlich laufen. Elastic Beanstalk überwacht diese Anwendungen und startet alle Prozesse, die beendet werden, neu. Für kurz laufende Prozesse verwenden Sie einen [Buildfile \(p. 104\)](#)-Befehl.

Verwenden eines Proffiles auf Amazon Linux AMI (Vorgängerversion von Amazon Linux 2)

Wenn Ihre Elastic Beanstalk Go-Umgebung eine Amazon Linux AMI-Plattformversion verwendet (Vorgängerversion von Amazon Linux 2), lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

Übergeben eines Portwertes

Elastic Beanstalk konfiguriert den nginx-Proxy für das Weiterleiten von Anfragen an Ihre Anwendung, und zwar auf der Portnummer, die in der `PORT` [Umgebungseigenschaft \(p. 102\)](#) für Ihre Anwendung angegeben ist. Ihre Anwendung sollte stets auf diesem Port überwachen. Sie können auf diese Variable im Anwendungscode zugreifen, indem Sie die `os.Getenv("PORT")`-Methode aufrufen.

Elastic Beanstalk nutzt die Portnummer, die in der `PORT`-Umgebungseigenschaft für den Port der ersten Anwendung in `Procfile` angegeben wurde und erhöht die Portnummer dann für jede nachfolgende Anwendung in der `Procfile`-Datei um 100. Wenn die `PORT`-Umgebungseigenschaft nicht festgelegt ist, verwendet Elastic Beanstalk 5000 für den anfänglichen Port.

Im obigen Beispiel ist die `PORT`-Umgebungseigenschaft für die `web`-Anwendung 5000, die `queue_processor`-Anwendung ist 5100 und die `foo`-Anwendung ist 5200.

Sie können den anfängliche Port angeben, indem Sie die `PORT`-Option mit dem [aws:elasticbeanstalk:application:environment \(p. 684\)](#)-Namespace festlegen, wie im folgenden Beispiel veranschaulicht.

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

Weitere Informationen zum Festlegen von Umgebungseigenschaften für Ihre Anwendung finden Sie unter [Optionseinstellungen \(p. 724\)](#).

Erstellen von ausführbaren Dateien auf dem Server mit einer Buildfile-Datei

Um einen benutzerdefinierten Build- und Konfigurationsbefehl für Ihre Go-Anwendung anzugeben, schließen Sie eine Datei mit dem Namen `Buildfile` in das Stammverzeichnis Ihres Quell-Bundle ein.

Beim Dateinamen muss die Groß- und Kleinschreibung beachtet werden. Verwenden Sie das folgende Format für die `Buildfile`:

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

Der Befehl in Ihrer `Buildfile` muss folgendem regulären Ausdruck entsprechen: `^ [A-Za-z0-9_]+ : \s* .+$`.

Elastic Beanstalk überwacht nicht die Anwendung, die mit einem `Buildfile` ausgeführt wird. Verwenden Sie eine `Buildfile` für Befehle, die für kurze Zeiträume ausgeführt und nach Abschluss ihrer Aufgaben beendet werden. Für lange laufende Anwendungsprozesse, die nicht beendet werden sollten, verwenden Sie stattdessen die [Profile \(p. 103\)](#).

Im folgenden Beispiel einer `Buildfile` ist `build.sh` ein Shell-Skript, das sich im Stammverzeichnis des Quell-Bundle befindet:

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

Alle Pfade in der `Buildfile` sind relativ zum Stammverzeichnis des Quell-Bundle. Wenn Sie im Voraus wissen, wo sich die Dateien auf der Instance befinden, können Sie absolute Pfade in die `Buildfile` aufnehmen.

Konfigurieren des Reverse-Proxy-Servers

Elastic Beanstalk verwendet nginx als Reverse-Proxy-Server, um die Anwendung dem Elastic Load Balancing Load Balancer auf Port 80 zuzuweisen. Elastic Beanstalk bietet eine nginx-Standardkonfiguration, die Sie entweder erweitern oder gänzlich mit einer eigenen Konfiguration überschreiben können.

Standardmäßig wird der nginx-Proxy von Elastic Beanstalk so konfiguriert, dass alle Anforderungen an die Anwendung über Port 5000 weitergeleitet werden. Diese Port-Standardeinstellung können Sie überschreiben. Legen Sie dazu die [PORT-Umgebungseigenschaft \(p. 102\)](#) auf den Überwachungsport der Hauptanwendung fest.

Note

Der Überwachungsport der Anwendung kollidiert nicht mit dem Überwachungsport des nginx-Servers, von dem dieser die Anforderungen des Load Balancers empfängt.

Alle Amazon Linux 2-Plattformen unterstützen eine einheitliche Proxy-Konfigurationsfunktion. Details zum Konfigurieren des Proxy-Servers auf den neuen Amazon Corretto-Plattformversionen, die Amazon Linux 2 ausführen, erhalten Sie, wenn Sie den Abschnitt Reverse Proxy-Konfiguration unter [the section called "Erweitern von Linux-Plattformen" \(p. 34\)](#) erweitern.

Konfigurieren des Proxys auf Amazon Linux AMI (Vorgängerversion von Amazon Linux 2)

Wenn Ihre Elastic Beanstalk Go-Umgebung eine Amazon Linux AMI-Plattformversion verwendet (Vorgängerversion von Amazon Linux 2), lesen Sie die Informationen in diesem Abschnitt.

Erweitern und Überschreiben der Standard-Proxykonfiguration

Elastic Beanstalk nutzt einen nginx-Server als Reverse-Proxy-Server, um die Anwendung dem Load Balancer auf Port 80 zuzuweisen. Wenn Sie Ihre eigene nginx-Konfiguration bereitstellen möchten, können Sie die Standardkonfiguration von Elastic Beanstalk überschreiben, indem Sie die `.ebextensions/nginx/nginx.conf`-Datei in Ihr Quell-Bundle aufnehmen. Wenn diese Datei vorhanden ist, nutzt Elastic Beanstalk sie anstelle der Standard-nginx-Konfigurationsdatei.

Wenn Sie Richtlinien zusätzlich zu denen im `nginx.conf` http-Block hinzufügen möchten, können Sie auch weitere Konfigurationsdateien im `.ebextensions/nginx/conf.d/-Verzeichnis` Ihres Quell-Bundle bereitstellen. Alle Dateien in diesem Verzeichnis müssen über die `.conf`-Erweiterung verfügen.

Wenn Sie die Funktionalität von Elastic Beanstalk nutzen möchten, z. B. [Erweiterte Zustandsberichte und Überwachung \(p. 824\)](#), automatische Anwendungszuordnungen und statische Dateien, müssen Sie die folgende Zeile in den `server`-Block der `nginx`-Konfigurationsdatei einfügen:

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

Bereitstellen einer Go-Anwendung in Elastic Beanstalk

Dieses Tutorial erläutert den Prozess zum Erstellen einer Go-Anwendung und zum Bereitstellen dieser Anwendung in einer AWS Elastic Beanstalk-Umgebung.

Abschnitte

- [Voraussetzungen \(p. 106\)](#)
- [Erstellen Sie eine Go-Anwendung. \(p. 106\)](#)
- [Bereitstellen Ihrer Go-Anwendung mit der EB-CLI \(p. 107\)](#)
- [Bereinigen \(p. 109\)](#)

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie über Kenntnisse über die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole verfügen. Sofern noch nicht geschehen, befolgen Sie die Anweisungen in [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#), um Ihre erste Elastic Beanstalk-Umgebung zu starten.

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine -Shell für die Ausführung der Befehle. Befehle werden in Auflistungen mit einem vorangestellten Eingabeaufforderungs-Symbol (\$) und dem Namen des aktuellen Verzeichnisses angezeigt, falls angebracht.

```
~/eb-project$ this is a command  
this is output
```

Verwenden Sie auf Linux und macOS Ihren bevorzugten Shell- und Paket-Manager. Sie können unter Windows 10 das [Windows-Subsystem für Linux](#) installieren, um eine Windows-Version von Ubuntu und Bash zu erhalten.

Dieses Tutorial verwendet die Elastic Beanstalk Command Line Interface (EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter [Installieren der EB CLI \(p. 1009\)](#) und [Konfigurieren der EB CLI \(p. 1017\)](#).

Erstellen Sie eine Go-Anwendung.

Erstellen Sie ein Projektverzeichnis.

```
~$ mkdir eb-go  
~$ cd eb-go
```

Als Nächstes erstellen Sie eine Anwendung, die Sie mit Elastic Beanstalk bereitstellen. Wir erstellen einen "Hello World"-RESTful-Webservice.

In diesem Beispiel wird eine benutzerdefinierte Begrüßung gedruckt, die je nach Pfad, der für den Zugriff auf den Service verwendet wurde, variiert.

Erstellen Sie eine Textdatei in diesem Verzeichnis mit dem Namen `application.go` mit folgendem Inhalt.

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)
}
```

Bereitstellen Ihrer Go-Anwendung mit der EB-CLI

Als Nächstes erstellen Sie die Anwendungsumgebung und stellen die konfigurierte Anwendung in Elastic Beanstalk bereit.

Erstellen einer Umgebung und Bereitstellen der Go-Anwendung

1. Initialisieren Sie das EB CLI-Repository mit dem Befehl `eb init`.

```
~/eb-go$ eb init -p go go-tutorial --region us-east-2
Application go-tutorial has been created.
```

Mit diesem Befehl wird die Anwendung `go-tutorial` erstellt und das lokale Repository wird zur Umgebungserstellung mit der neuesten Go-Plattformversion eingerichtet.

2. (Optional) Führen Sie `eb init` erneut aus, um ein Standardschlüsselpaar für die SSH-Verbindung zur EC2-Instance (auf der die Anwendung ausgeführt wird) zu konfigurieren.

```
~/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 ]
```

Wählen Sie ein Schlüsselpaar aus (sofern vorhanden) oder befolgen Sie die Anweisungen, um eines zu erstellen. Falls keine Anweisungen angezeigt werden oder Sie die Einstellungen später ändern möchten, führen Sie `eb init -i` aus.

3. Verwenden Sie `eb create`, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen. Elastic Beanstalk erstellt automatisch eine Binärdatei für Ihre Anwendung und startet sie auf Port 5000.

```
~/eb-go$ eb create go-env
```

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die so konfiguriert ist, dass Web-Apps auf der von Ihnen ausgewählten Plattform ausgeführt werden.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance security group (Instance-Sicherheitsgruppe): Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer) – Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe): Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form **subdomain.region.elasticbeanstalk.com** weiterleitet.

Elastic Beanstalk verwaltet all diese Ressourcen. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Note

Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#).

Wenn die Umgebungserstellung abgeschlossen ist, öffnen Sie die Website mit eb open.

```
~/eb-go$ eb open
```

Ein Browserfenster wird mit dem für die Anwendung erstellten Domänennamen geöffnet.

Falls die Anwendung nicht ausgeführt wird oder Sie eine Fehlermeldung erhalten, finden Sie unter [Troubleshooting deployments \(p. 1111\)](#) weitere Informationen zur Ermittlung der Fehlerursache.

Falls die Anwendung ausgeführt wird, haben Sie Ihre erste Go-Anwendung in Elastic Beanstalk bereitgestellt. Herzlichen Glückwunsch!

Bereinigen

Falls Sie nicht länger mit Elastic Beanstalk arbeiten möchten, können Sie die Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 531\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie auf Environment actions (Umgebungsaktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Mit der EB CLI gehen Sie folgendermaßen vor.

```
~/eb-go$ eb terminate
```

Java-Anwendungen in Elastic Beanstalk erstellen und bereitstellen

AWS Elastic Beanstalk unterstützt zwei Plattformen für Java-Anwendungen.

- Tomcat – Eine Plattform auf der Grundlage von Apache Tomcat, einem Open-Source-Webcontainer für Anwendungen mit Java-Servlets und JavaServer Pages (JSPs) zur Verarbeitung von HTTP-Anforderungen. Tomcat vereinfacht die Entwicklung von Webanwendungen durch Multithreading, eine deklarative Sicherheitskonfiguration und umfassende Anpassungsmöglichkeiten. Elastic Beanstalk hat Plattform-Branches für jede der aktuellen Hauptversionen von Tomcat. Weitere Informationen finden Sie unter [Die Tomcat-Plattform \(p. 117\)](#).
- Java SE – Eine Plattform für Anwendungen, die keinen Webcontainer oder einen anderen Webcontainer als Tomcat verwenden, z. B. Jetty oder GlassFish. Sie können beliebige Java Archives (JARs) der Bibliothek einschließen, die von Ihrer Anwendung im Quell-Bundle verwendet werden, das Sie in Elastic Beanstalk bereitstellen. Weitere Informationen finden Sie unter [Die Java SE-Plattform \(p. 128\)](#).

Die jüngsten Branches sowohl der Tomcat- als auch der Java SE-Plattform basieren auf Amazon Linux 2 und verwenden Corretto – die AWS Java SE-Distribution. Namen dieser Zweige in den Plattformlisten enthalten das Wort Corretto anstelle von Java, zum Beispiel Corretto 11 with Tomcat 8.5.

Eine Liste der aktuellen Plattformversionen finden Sie unter [Tomcat](#) und [Java SE](#) im Leitfaden AWS Elastic Beanstalk-Plattformen.

AWS stellt mehrere Tools für das Arbeiten mit Java und Elastic Beanstalk bereit. Unabhängig von der Plattformversion, für die Sie sich entscheiden, können Sie das [AWS SDK for Java \(p. 116\)](#) verwenden, um andere AWS-Services in Ihrer Java-Anwendung zu nutzen. Das AWS SDK for Java ist ein Satz von Bibliotheken, die es Ihnen ermöglichen, AWS-APIs von Ihrem Anwendungscode zu nutzen, ohne die unformatierten HTTP-Aufrufe von Grund auf zu schreiben.

Wenn Sie die integrierte Entwicklungsumgebung (Integrated Development Environment, IDE) Eclipse verwenden, um Ihre Java-Anwendung zu entwickeln, können Sie auch das [AWS Toolkit for Eclipse \(p. 140\)](#) erhalten. Das AWS Toolkit for Eclipse ist ein Open-Source-Plugin, mit dem Sie AWS-Ressourcen, einschließlich Elastic Beanstalk-Anwendungen und -Umgebungen, in der Eclipse IDE verwalten können.

Wenn die Befehlszeile eher nach Ihrem Geschmack ist, installieren Sie die [Elastic Beanstalk Command Line Interface \(p. 1008\)](#) (EB CLI), um Ihre Elastic Beanstalk-Umgebungen über die Befehlszeile zu erstellen, zu überwachen und zu verwalten. Wenn Sie mehrere Umgebungen für Ihre Anwendung ausführen, kann die EB CLI in Git integriert werden, sodass Sie jede Umgebung einer anderen Git-Verzweigung zuordnen können.

In den Themen in diesem Kapitel wird davon ausgegangen, dass Sie über einige Kenntnisse zu Elastic Beanstalk-Umgebungen verfügen. Wenn Sie Elastic Beanstalk noch nicht verwendet haben, machen Sie sich im [Tutorial "Erste Schritte" \(p. 3\)](#) mit den Grundlagen vertraut.

Themen

- [Erste Schritte mit Java auf Elastic Beanstalk \(p. 110\)](#)
- [Einrichten der Java-Entwicklungsumgebung \(p. 115\)](#)
- [Elastic Beanstalk Tomcat-Plattform verwenden \(p. 117\)](#)
- [Elastic Beanstalk Java SE-Plattform verwenden \(p. 128\)](#)
- [Hinzufügen einer Amazon RDS-DB-Instance zur Java-Anwendungsumgebung \(p. 134\)](#)
- [Verwenden des AWS Toolkit for Eclipse \(p. 140\)](#)
- [Ressourcen \(p. 155\)](#)

Erste Schritte mit Java auf Elastic Beanstalk

Für die ersten Schritte mit Java-Anwendungen in AWS Elastic Beanstalk benötigen Sie nur das [Quell-Bundle \(p. 412\)](#) einer Anwendung, damit Sie die erste Anwendungsversion hochladen und in einer Umgebung bereitstellen können. Im Rahmen der Umgebungserstellung werden von Elastic Beanstalk alle erforderlichen AWS-Ressourcen für eine skalierbare Webanwendung zugewiesen.

Starten einer Umgebung mit einer Java-Beispielanwendung

Elastic Beanstalk bietet für jede Plattform sowohl einfache Beispielanwendungen mit einer Seite als auch komplexe Beispiele, in denen die Nutzung weiterer AWS-Ressourcen (wie z. B. Amazon RDS sowie sprach- oder plattformspezifische Funktionen und APIs) veranschaulicht wird.

Die einzelnen Seitenbeispiele haben den gleichen Code, den Sie auch erhalten, wenn Sie eine Umgebung erstellen, ohne Ihren eigenen Quellcode anzugeben. Die komplexeren Beispiele werden auf GitHub gehostet und müssen möglicherweise kompiliert oder erstellt werden, bevor Sie auf einer Elastic Beanstalk-Umgebung bereitgestellt werden.

Beispiele

Name	Unterstützte Versionen	Umg. Worker	Source	Beschreibung
Tomcat	All Tomcat mit Corretto-Banches der einzelnen Platform Seite	Webserver	tomcat.zip	<p>Tomcat-Webanwendung mit einer Seite (<code>index.jsp</code>), die im Stamm der Website angezeigt wird.</p> <p>Für Worker-Umgebungen (p. 516) enthält dieses Beispiel eine <code>cron.yaml</code>-Datei, mit der eine geplante Aufgabe konfiguriert wird, die <code>scheduled.jsp</code> einmal pro Minute aufruft. Wenn <code>scheduled.jsp</code> aufgerufen wird, schreibt sie in eine Protokolldatei unter <code>/tmp/sample-app.log</code>. Schließlich ist eine Konfigurationsdatei in <code>.ebextensions</code> enthalten, die die Protokolle aus <code>/tmp/</code> in die Standorte kopiert, die von Elastic Beanstalk gelesen werden, wenn Sie Umgebungsprotokolle anfordern.</p> <p>Wenn Sie in einer Umgebung, auf der dieses Beispiel ausgeführt wird, X-Ray-Integration aktivieren (p. 627), zeigt diese Anwendung zusätzlichen Inhalt hinsichtlich X-Ray und bietet eine Option zum Generieren von Debug-Informationen, die Sie in der X-Ray-Konsole anzeigen können.</p>
Corretto 11	(einzelne Seite)Corretto 8	Webserver	corretto.zip	<p>Corretto-Anwendung mit Buildfile- und Profile-Konfigurationsdateien.</p> <p>Wenn Sie in einer Umgebung, auf der dieses Beispiel ausgeführt wird, X-Ray-Integration aktivieren (p. 627), zeigt diese Anwendung zusätzlichen Inhalt hinsichtlich X-Ray und bietet eine Option zum Generieren von Debug-Informationen, die Sie in der X-Ray-Konsole anzeigen können.</p>
Scorekeep	Java 8	Webserver	the repo at GitHub.com (Das)	Scorekeep ist eine RESTful-Web-API, die das Spring-Framework verwendet, um eine Schnittstelle zum Erstellen und Verwalten von Benutzern, Sessions und

Nam	Unterstützte Versionen	Umg	Source	Beschreibung
			Repository klonen unter GitHub.com	<p>Spielden bereitzustellen. Die API ist mit einer Angular 1.5-Web-App gebündelt, von der die API über HTTP eingesetzt wird.</p> <p>Die Anwendung nutzt Funktionen der Java SE-Plattform, um Abhängigkeiten herunterzuladen und auf Instances zu erstellen, und minimiert so die Größe des Quell-Bundles. Die Anwendung umfasst auch nginx-Konfigurationsdateien, mit denen die Standardkonfiguration überschrieben wird, sodass die Bereitstellung der Frontend-Webanwendung statisch auf Port 80 über den Proxy erfolgt und Anforderungen über Pfade unter /api an die auf localhost:5000 ausgeführte API weitergeleitet werden.</p> <p>Scorekeep enthält auch eine xray-Verzweigung, die zeigt, wie eine Java-Anwendung für die Verwendung mit AWS X-Ray instrumentiert wird. Sie zeigt die Instrumentierung der eingehenden HTTP-Anforderungen mit einem Servlet-Filter, automatischer und manueller AWS SDK Client-Instrumentierung, Recorder-Konfiguration und Instrumentierung von ausgehenden HTTP-Anforderungen und SQL-Clients.</p> <p>Anweisungen dazu, wie Sie die Anwendung mit X-Ray testen, finden Sie in der Readme-Datei oder im AWS X-Ray getting started tutorial.</p>

Name	Unterstützte Versionen	Umgang mit der Quellcodebasis	Beschreibung
DoesTomcat 8 mit Java 8 it Have Snakes?		Webservicer the repo at GitHub.com (Das Repository klonen unter GitHub.com)	Does it Have Snakes? ist eine Tomcat-Webanwendung, die die Nutzung von Elastic Beanstalk-Konfigurationsdateien, Amazon RDS, JDBC, PostgreSQL, Servlets, JSPs, Simple Tag Support, Tag Files, Log4J, Bootstrap und Jackson zeigt. Der Quellcode für dieses Projekt enthält ein minimales Build-Skript, das die Servlets und Modelle in Klassendateien kompiliert und die erforderlichen Dateien in ein Webarchiv verpackt, das Sie in einer Elastic Beanstalk-Umgebung bereitstellen können. Vollständige Anweisungen dazu finden Sie in der Readme-Datei im Projekt-Repository.
LocustJava 8 Load Generator		Webservicer the repo at GitHub.com (Das Repository klonen unter GitHub.com)	Webanwendung, die Sie verwenden können, um eine andere laufende Webanwendung in einer Elastic Beanstalk-Umgebung zu testen. Zeigt die Verwendung von Buildfile- und Procfile-Dateien, DynamoDB und Locust, ein Open-Source-Tool für Belastungstests.

Laden Sie eine Beispielanwendung herunter und stellen Sie diese in Elastic Beanstalk bereit. Führen Sie dazu die folgenden Schritte aus:

So starten Sie eine Umgebung mit einer Beispielanwendung (Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Applications (Anwendungen) und anschließend in der Liste den Namen einer vorhandenen Anwendung aus. Sie können auch [eine Anwendung erstellen \(p. 403\)](#).
3. Wählen Sie auf der Anwendungsübersichtsseite die Option Create a new environment (Neue Umgebung erstellen).

Environment name	Health	Date created	Last modified	URL	Platform
GettingStartedApp-env	OK	2020-01-28 12:06:50 UTC-0800	2020-01-30 15:02:35 UTC-0800	GettingStartedApp-env.bn7dx222kw.us-east-2.elasticbeanstalk.com	Tomcat running on Linux
GettingStartedApp-Worker	OK	2020-01-28 16:34:29 UTC-0800	2020-01-28 16:38:20 UTC-0800	GettingStartedApp-Worker.bn7dx222kw.us-east-2.elasticbeanstalk.com	IIS 10.0 running on Windows

- Wählen Sie die Web server environment (Webserverumgebung) oder die Worker environment (Worker-Umgebung) als [Umgebungsebene \(p. 14\)](#) aus. Nach der Erstellung lässt sich die Umgebungsebene nicht mehr ändern.

Note

Die [.NET-Plattform auf Windows Server \(p. 189\)](#) unterstützt die Worker-Umgebungsschicht nicht.

Select environment tier

AWS Elastic Beanstalk has two types of environment tiers to support different types of web applications. Web server environments run web applications that listen for and then process HTTP requests, typically over port 80. Workers are specialized applications that perform a background processing task that listens for messages on an Amazon SQS queue. Worker applications post the results of their processing back to the application by using HTTP.

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](#)

- Wählen Sie in Platform (Plattform) die Plattform und den Plattformzweig aus, die/der mit der Sprache Ihrer Anwendung übereinstimmt.

Note

Elastic Beanstalk unterstützt mehrere [Versionen \(p. 32\)](#) für die meisten Plattformen, die aufgelistet sind. Standardmäßig wählt die Konsole die empfohlene Version für die gewählte Plattform und Plattformvariante. Wenn Ihre Anwendung eine andere Version erfordert, können

Sie diese hier auswählen oder Configure more options (Weitere Optionen konfigurieren) auswählen wie in Schritt 7 beschrieben. Informationen zu unterstützten Plattformversionen finden Sie unter [the section called “Unterstützte Plattformen” \(p. 32\)](#).

6. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
7. Wählen Sie Configure more options (Weitere Optionen konfigurieren) aus, um Ihre Umgebung weiter anzupassen. Sie können die folgenden Optionen nur während des Erstellens der Umgebung festlegen:
 - Environment name
 - Domänenname
 - Plattformversion
 - VPC
 - Stufe

Sie können die folgenden Einstellungen nach der Umgebungserstellung zwar ändern, jedoch müssen dafür neue Instances oder andere Ressourcen bereitgestellt werden, deren Anwendung viel Zeit in Anspruch nehmen kann:

- Instance-Typ, Root-Volumen, Schlüsselpaar und AWS Identity and Access Management (IAM)-Rolle
- Interne Amazon RDS-Datenbank
- Load Balancer

Weitere Informationen zu allen verfügbaren Einstellungen finden Sie unter [Der Assistent zum Erstellen einer neuen Umgebung \(p. 437\)](#).

8. Wählen Sie Create environment (Umgebung erstellen) aus.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit [eine neue Version \(p. 471\)](#) der Anwendung oder eine gänzlich andere Anwendung bereitstellen. Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine EC2-Instances bereitgestellt oder neu gestartet werden müssen.

Wenn Sie ein oder zwei Beispielanwendungen bereitgestellt haben, können Sie mit der lokalen Entwicklung und Ausführung von Java-Anwendungen beginnen. Die entsprechenden Informationen zum Einrichten einer Java-Entwicklungsumgebung und zu allen benötigten Tools und Bibliotheken finden Sie im [nächsten Abschnitt \(p. 115\)](#).

Einrichten der Java-Entwicklungsumgebung

Richten Sie eine Java-Entwicklungsumgebung ein, um Ihre Anwendung vor der Bereitstellung in AWS Elastic Beanstalk lokal zu testen. In diesem Thema finden Sie die Schritte zum Einrichten der Entwicklungsumgebung sowie Links zu den Installationsseiten nützlicher Tools.

Allgemeine Einrichtungsschritte und Tools für alle Sprachen finden Sie unter [Konfigurieren des Entwicklungskomputers \(p. 1005\)](#).

Abschnitte

- [Installieren des Java Development Kits \(p. 116\)](#)
- [Installieren eines Webcontainers \(p. 116\)](#)
- [Herunterladen von Bibliotheken \(p. 116\)](#)
- [Installieren des AWS SDK for Java \(p. 116\)](#)

- [Installieren einer IDE oder eines Texteditors \(p. 116\)](#)
- [Installieren des AWS Toolkit for Eclipse \(p. 117\)](#)

Installieren des Java Development Kits

Installieren des Java Development Kits (JDK). Falls Sie keine Version bevorzugen, wählen Sie die neueste aus. Laden Sie das JDK unter oracle.com herunter.

Das JDK enthält den Java-Compiler, mit dem Sie Quelldateien in Klassendateien erstellen können, die sich auf einem Elastic Beanstalk-Webserver ausführen lassen.

Installieren eines Webcontainers

Falls Sie noch keinen anderen Webcontainer oder ein Framework nutzen, können Sie die entsprechende Tomcat-Version installieren:

- [Tomcat 8 herunterladen \(erfordert Java 7 oder eine neuere Version\)](#)
- [Tomcat 7 herunterladen \(erfordert Java 6 oder eine neuere Version\)](#)

Herunterladen von Bibliotheken

Elastic Beanstalk-Plattformen enthalten standardmäßig nur wenige Bibliotheken. Laden Sie die Bibliotheken, die von der Anwendung genutzt werden, herunter und speichern Sie diese im Projektordner, um sie im Quell-Bundle der Anwendung bereitzustellen.

Sofern Sie Tomcat lokal installiert haben, können Sie die Bibliotheken für die Servlet-API und die JavaServer Pages (JSP)-API aus dem Installationsordner kopieren. Wenn die Bereitstellung für eine Tomcat-Plattformversion erfolgt, ist es nicht nötig, diese Dateien in das Quell-Bundle einzubinden. Allerdings müssen Sie zur Kompilierung aller Klassen, die sie verwenden, im `classpath` vorhanden sein.

JUnit, Google Guava und Apache Commons bieten mehrere nützliche Bibliotheken. Weitere Informationen finden Sie auf den entsprechenden Homepages:

- [JUnit herunterladen](#)
- [Google Guava herunterladen](#)
- [Apache Commons herunterladen](#)

Installieren des AWS SDK for Java

Wenn Sie die AWS-Ressourcen aus der Anwendung heraus verwalten möchten, installieren Sie das AWS SDK for Java. Mit dem AWS SDK for Java können Sie z. B. Amazon DynamoDB (DynamoDB) verwenden, um Sitzungszustände von Apache Tomcat-Anwendungen über mehrere Webserver hinweg gemeinsam zu nutzen. Weitere Informationen finden Sie unter [Manage Tomcat Session State with Amazon DynamoDB](#) in der AWS SDK for Java-Dokumentation.

Weitere Informationen und Installationsanleitungen finden Sie auf der [AWS SDK for Java-Homepage](#).

Installieren einer IDE oder eines Texteditors

Integrierte Entwicklungsumgebungen (IDEs) bieten zahlreiche Funktionen, mit denen die Anwendungsentwicklung vereinfacht wird. Falls Sie noch keine IDE für die Java-Entwicklung eingesetzt haben, probieren Sie Eclipse und IntelliJ aus. Dann sehen Sie, welche IDE für Sie am besten geeignet ist.

- [Eclipse IDE for Java EE Developers installieren](#)
- [IntelliJ installieren](#)

Note

Mit einer IDE werden dem Projektordner möglicherweise Dateien hinzugefügt, die Sie unter Umständen nicht an die Quellüberwachung übergeben möchten. Damit diese Dateien nicht an die Quellüberwachung übergeben werden, setzen Sie `.gitignore` oder die Entsprechung Ihres Quellüberwachungstools ein.

Wenn Sie gerade mit dem Programmieren beginnen und nicht alle Funktionen einer IDE benötigen, können Sie [Sublime Text installieren](#).

Installieren des AWS Toolkit for Eclipse

Das [AWS Toolkit for Eclipse \(p. 140\)](#) ist ein Open Source-Plug-in für die Eclipse Java-IDE und vereinfacht es für Entwickler, Java-Anwendungen mit AWS zu entwickeln, zu debuggen und bereitzustellen. Installationsanleitungen finden Sie auf der [AWS Toolkit for Eclipse-Homepage](#).

Elastic Beanstalk Tomcat-Plattform verwenden

Important

Die Amazon Linux 2-Plattformversionen unterscheiden sich grundlegend von den Amazon Linux AMI-Plattformversionen (vor Amazon Linux 2). Diese unterschiedlichen Plattformgenerationen sind auf verschiedene Arten inkompatibel. Wenn Sie zu einer Amazon Linux 2-Plattformversion migrieren, müssen Sie die Informationen in [the section called “Aktualisieren auf Amazon Linux 2” \(p. 503\)](#) lesen.

Die AWS Elastic Beanstalk Tomcat-Plattform umfasst eine Reihe von [Plattformversionen](#) für Java-Webanwendungen, die in einem Tomcat-Webcontainer ausgeführt werden können. Tomcat läuft hinter einem nginx-Proxy-Server. Jede Plattformvariante entspricht einer Hauptversion von Tomcat, z. B. Java 8 mit Tomcat 8.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für die [Änderung der Konfiguration einer ausgeführten Umgebung \(p. 658\)](#) verfügbar. Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie [gespeicherte Konfigurationen \(p. 764\)](#) verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie [Konfigurationsdateien \(p. 722\)](#) einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Die Elastic Beanstalk Tomcat-Plattform beinhaltet einen Reverse-Proxy-Server, der Anforderungen an die Anwendung weiterleitet. Sie können [Konfigurationsoptionen \(p. 120\)](#) verwenden, um den Proxy-Server zur Verarbeitung statischer Komponenten aus einem Ordner im Quellcode zu konfigurieren und so die Auslastung für die Anwendung zu reduzieren. In erweiterten Szenarien können Sie [eigene .conf-Dateien in das Quell-Bundle einbinden \(p. 124\)](#), um die Proxy-Konfiguration von Elastic Beanstalk zu erweitern oder zu überschreiben.

Note

Elastic Beanstalk unterstützt [nginx](#) (Standard) und [Apache HTTP Server](#) als Proxy-Server auf der Tomcat-Plattform. Wenn Ihre Elastic Beanstalk Tomcat-Umgebung einen Branch der Amazon-Linux-AMI-Plattform (vor Amazon Linux 2) verwendet, haben Sie auch die Möglichkeit, [Apache](#)

[HTTP Server Version 2.2](#) zu verwenden. Apache (neueste) ist der Standardwert für diese älteren Plattformzweige.

Java-Anwendungen müssen in einem Webanwendungsarchiv (WAR-Datei) mit einer bestimmten Struktur platziert werden. Weitere Informationen zur erforderlichen Struktur und deren Bezug zur Projektverzeichnisstruktur finden Sie unter [Strukturieren Ihres Projektordners \(p. 122\)](#).

Für die Ausführung von mehreren Anwendungen auf demselben Webserver können Sie in einem einzigen Quell-Bundle [mehrere WAR-Dateien bündeln \(p. 121\)](#). Jede Anwendung in einem Quell-Bundle mit mehreren WAR-Dateien wird je nach dem WAR-Dateinamen entweder mit dem Stammpfad (`ROOT.war` mit `myapp.elasticbeanstalk.com/`) oder mit einem direkt darunter liegenden Pfad (`app2.war` mit `myapp.elasticbeanstalk.com/app2/`) ausgeführt. In einem Quell-Bundle mit einer einzigen WAR-Datei wird die Anwendung immer mit dem Stammpfad ausgeführt.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, wenn vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe [Konfigurationsoptionen \(p. 646\)](#).

Details zu den verschiedenen Möglichkeiten zum Erweitern einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#).

Themen

- [Konfigurieren Ihrer Tomcat-Umgebung \(p. 118\)](#)
- [Tomcat-Konfigurations-Namespaces \(p. 120\)](#)
- [Die Amazon Linux-AMI \(vor Amazon Linux 2\) Tomcat-Plattform \(p. 121\)](#)
- [Bündeln mehrerer WAR-Dateien für Tomcat-Umgebungen \(p. 121\)](#)
- [Strukturieren Ihres Projektordners \(p. 122\)](#)
- [Konfigurieren des Proxy-Servers Ihrer Tomcat-Umgebung \(p. 124\)](#)

Konfigurieren Ihrer Tomcat-Umgebung

Die Elastic Beanstalk Tomcat-Plattform bietet einige plattformspezifische Optionen zusätzlich zu den Standard-Optionen, die allen Plattformen gemeinsam sind. Mit diesen Optionen können Sie die JVM (Java Virtual Machine) konfigurieren, die auf den Webservern der Umgebung ausgeführt wird, und die Systemeigenschaften definieren, die Zeichenfolgen mit Konfigurationsinformationen für die Anwendung bereitstellen.

Mit der Elastic Beanstalk-Konsole können Sie die Protokollrotation für Amazon S3 aktivieren, Variablen konfigurieren, die die Anwendung in der Umgebung lesen kann, und die PHP-Einstellungen ändern.

So konfigurieren Sie Ihre Tomcat-Umgebung in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.

Containeroptionen

Sie können die folgenden plattformspezifischen Optionen angeben:

- Proxy server (Proxy-Server) – Der Proxy-Server, der in Ihren Umgebungs-Instances verwendet werden soll. Standardmäßig wird nginx verwendet.

JVM-Containeroptionen

Über die Heap-Größe in der JVM wird bestimmt, wie viele Objekte die Anwendung im Speicher erstellen kann, bevor eine [Speicherbereinigung](#) erfolgt. Sie können die Angaben für Initial JVM Heap Size (-Xms argument) und Maximum JVM Heap Size (-Xmx argument) ändern. Wenn Sie einen höheren Wert für die anfängliche Heap-Größe angeben, können mehr Objekte erstellt werden, bevor eine Speicherbereinigung ausgeführt wird. Jedoch dauert es länger, bis der Heap im Rahmen der Speicherbereinigung komprimiert ist. Die maximale Heap-Größe gibt die maximale Größe des Arbeitsspeichers an, die bei einer großen Auslastung von der JVM für die Heap-Erweiterung zugewiesen werden kann.

Note

Der verfügbare Speicher hängt vom Amazon EC2-Instance-Typ ab. Weitere Informationen zu den für die Elastic Beanstalk-Umgebung verfügbaren EC2-Instance-Typen finden Sie unter [Instance Types \(Instance-Typen\)](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Der JVM-Heap verfügt über einen Bereich für permanente Generierung, in dem Klassendefinitionen und zugehörige Metadaten gespeichert werden. Um die Größe für die permanente Generierung zu ändern, geben Sie im Feld Maximum JVM PermGen Size (-XX:MaxPermSize argument) die neue Größe ein. Diese Einstellung gilt nur für Java 7 und frühere Versionen.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren): gibt an, ob die Protokolldateien für die Amazon EC2-Instances der Anwendung in den der Anwendung zugeordneten Amazon S3-Bucket kopiert werden sollen.

Statische Dateien

Um die Leistung zu verbessern, können Sie im Abschnitt Static files (Statische Dateien) den Proxy-Server so konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zur Verzeichniszuordnung fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, statt die Anfrage an Ihre Anwendung weiterzuleiten.

Einzelheiten zur Konfiguration statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter [the section called "Statische Dateien" \(p. 776\)](#).

Umgebungseigenschaften

Im Bereich Environment Properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Umgebungseigenschaften werden als Schlüssel-Wert-Paare an die Anwendung weitergeleitet.

Die Tomcat-Plattform definiert für Tomcat-Umgebungen die Platzhaltereigenschaft `JDBC_CONNECTION_STRING`, um eine Verbindungszeichenfolge an eine externe Datenbank zu übergeben.

Note

Wenn Sie eine RDS DB-Instance an Ihre Umgebung anhängen, konstruieren Sie die JDBC-Verbindungszeichenfolge dynamisch aus den Eigenschaften der Amazon Relational Database Service (Amazon RDS)-Umgebung, die von Elastic Beanstalk bereitgestellt werden. Verwenden Sie `JDBC_CONNECTION_STRING` nur für Datenbank-Instances, die nicht mithilfe von Elastic Beanstalk bereitgestellt werden.

Weitere Informationen zur Verwendung von Amazon RDS mit Ihrer Java-Anwendung finden Sie unter [Hinzufügen einer Amazon RDS-DB-Instance zur Java-Anwendungsumgebung \(p. 134\)](#).

In der Tomcat-Umgebung mit Elastic Beanstalk-Ausführung kann über `System.getProperty()` auf die Umgebungsvariablen zugegriffen werden. Beispielsweise können Sie mit folgendem Code eine Eigenschaft mit dem Namen `API_ENDPOINT` als Variable lesen.

```
String endpoint = System.getProperty("API_ENDPOINT");
```

Weitere Informationen finden Sie unter [Umgebungseigenschaften und andere Softwareeinstellungen \(p. 621\)](#).

Tomcat-Konfigurations-Namespaces

Mithilfe einer [Konfigurationsdatei \(p. 722\)](#) können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Konfigurationsoptionen können durch den Elastic Beanstalk-Service oder die verwendete Plattform definiert und in Namespaces organisiert werden.

Die Tomcat-Plattform unterstützt neben den [unterstützten Optionen für alle Elastic Beanstalk-Umgebungen \(p. 666\)](#) auch Optionen in den folgenden Namespaces:

- `aws:elasticbeanstalk:container:tomcat:jvmoptions` – Ändern von JVM-Einstellungen. Die Optionen in diesem Namespace entsprechen wie folgt den Optionen in der Managementkonsole:
 - `Xms` – JVM-Befehlszeilenoptionen
 - `JVM Options` – JVM-Befehlszeilenoptionen
- `aws:elasticbeanstalk:environment:proxy` – Wählen Sie den Proxy-Server der Umgebung aus.

In der folgenden Beispielkonfigurationsdatei wird die Verwendung von Tomcat-spezifischen Konfigurationsoptionen veranschaulicht.

Example .ebextensions/tomcat-settings.config

```
option_settings:
  aws:elasticbeanstalk:container:tomcat:jvmoptions:
    Xms: 512m
    JVM Options: '-Xmn128m'
  aws:elasticbeanstalk:application:environment:
    API_ENDPOINT: mywebapi.zkpexsjtmd.us-west-2.elasticbeanstalk.com
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
```

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Die Amazon Linux-AMI (vor Amazon Linux 2) Tomcat-Plattform

Falls Ihre Elastic Beanstalk Tomcat-Umgebung eine Amazon Linux AMI-Plattformversion (vor Amazon Linux 2) verwendet, lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

Tomcat-Konfigurations-Namespaces

Die Tomcat Amazon Linux AMI-Plattform unterstützt zusätzliche Optionen in den folgenden Namespaces:

- `aws:elasticbeanstalk:container:tomcat:jvmoptions` – Zusätzlich zu den oben auf dieser Seite für diesen Namespace erwähnten Optionen unterstützen ältere Amazon Linux-AMI-Plattformversionen auch Folgendes:
 - `XX:MaxPermSize` – Maximale Größe für die permanente JVM-Generierung
- `aws:elasticbeanstalk:environment:proxy` – Konfigurieren Sie nicht nur den Proxy-Server, sondern auch die Antwortkomprimierung.

In der folgenden Beispielkonfigurationsdatei wird die Verwendung der Proxy-Namespace-Konfigurationsoptionen veranschaulicht.

Example `.ebextensions/tomcat-settings.config`

```
option_settings:  
  aws:elasticbeanstalk:environment:proxy:  
    GzipCompression: 'true'  
    ProxyServer: nginx
```

Elastic Beanstalk-Konfigurationsdateien einbinden

Um `.ebextensions`-Konfigurationsdateien bereitzustellen, schließen Sie sie in Ihre Anwendungsquelle ein. Fügen Sie für eine einzelne Anwendung `.ebextensions` zu einer komprimierten WAR-Datei hinzu, indem Sie den folgenden Befehl ausführen:

Example

```
zip -ur your_application.war .ebextensions
```

Für eine Anwendung, die mehrere WAR-Dateien benötigt, finden Sie weitere Anweisungen unter [Bündeln mehrerer WAR-Dateien für Tomcat-Umgebungen \(p. 121\)](#).

Bündeln mehrerer WAR-Dateien für Tomcat-Umgebungen

Wenn Ihre Web-App mehrere Web-Anwendungskomponenten umfasst, können Sie die Bereitstellung vereinfachen und die Betriebskosten senken, indem Sie Komponenten in einer einzigen Umgebung ausführen, anstatt eine separate Umgebung für jede Komponente auszuführen. Diese Strategie ist für leichte Anwendungen effektiv, die nicht viele Ressourcen erfordern, und für Entwicklungs- und Testumgebungen.

Zur Bereitstellung von mehreren Webanwendungen in Ihrer Umgebung kombinieren Sie die WAR-Dateien jeder Komponente zu einem einzelnen [Quell-Bundle \(p. 412\)](#).

Zum Erstellen eines Quell-Bundle der Anwendung mit mehreren WAR-Dateien organisieren Sie die WAR-Dateien mithilfe der folgenden Struktur.

```
MyApplication.zip  
### .ebextensions  
### .platform  
### foo.war
```

```
### bar.war
### ROOT.war
```

Wenn Sie ein Quell-Bundle mit mehreren WAR-Dateien in einer AWS Elastic Beanstalk-Umgebung bereitstellen, ist jede Anwendung auf einen anderen Pfad außerhalb des Stamm-Domänenamens verfügbar. Das vorausgehende Beispiel enthält drei Anwendungen: `foo`, `bar` und `ROOT`. `ROOT.war` ist ein spezieller Dateiname, der Elastic Beanstalk auffordert, diese Anwendung in der Stamm-Domäne auszuführen, damit die drei Anwendungen unter `http://MyApplication.elasticbeanstalk.com/foo`, `http://MyApplication.elasticbeanstalk.com/bar` und `http://MyApplication.elasticbeanstalk.com` verfügbar sind.

Das Quellpaket kann WAR-Dateien, einen optionalen `.ebextensions`-Ordner und einen optionalen `.platform`-Ordner enthalten. Weitere Informationen zu diesen optionalen Konfigurationsordnern finden Sie unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#).

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced
2. Wählen Sie für Platform (Plattform) die Plattform und den Plattformzweig aus, die der von Ihrer Anwendung verwendeten Sprache entsprechen, oder die Docker-Plattform für containerbasierte Anwendungen.
3. Wählen Sie für Application code (Anwendungscode) die Option Upload your code (Eigenen Code hochladen).
4. Wählen Sie Local file (Lokale Datei), wählen Sie Choose file (Datei wählen) und öffnen Sie das Quell-Bundle.
5. Klicken Sie auf Review and launch (Überprüfen und starten).
6. Prüfen Sie die verfügbaren Einstellungen und wählen Sie Create app (Anwendung erstellen) aus.

Weitere Informationen zum Erstellen von Quell-Bundles finden Sie unter [Erstellen des Quell-Bundles einer Anwendung. \(p. 412\)](#).

Strukturieren Ihres Projektordners

Damit auf einem Tomcat-Server bereitgestellte kompilierte Java EE (Java Platform Enterprise Edition)-Webanwendungsarchive (WAR-Dateien) genutzt werden können, müssen sie nach bestimmten [Richtlinien](#) strukturiert sein. Ihr Projektverzeichnis muss nicht denselben Standards entsprechen, dies wäre aber sinnvoll, da Kompilierung und Paketerstellung vereinfacht werden. Wenn Sie den Projektordner genauso strukturieren wie die WAR-Dateiinhalte, können Sie die Beziehung der Dateien untereinander und deren Verhalten auf dem Webserver leichter nachvollziehen.

In der folgenden empfohlenen Hierarchie wird der Quellcode der Webanwendung im Verzeichnis `src` platziert und so vom Build-Skript und der generierten WAR-Datei separiert.

```
~/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
    |-- WEB-INF         - Source code folder
    |   |-- classes      - Folder for private supporting files
    |   |-- lib           - Compiled classes
    |   |-- tags          - JAR libraries
    |   |-- tlds          - Tag files
    |   `-- web.xml       - Tag Library Descriptor files
    |-- com              - Deployment Descriptor
    |-- css              - Uncompiled classes
    |-- images           - Style sheets
    `-- images           - Image files
```

```
|-- js          - JavaScript files
`-- default.jsp - JSP (JavaServer Pages) webpage
```

Die Ordnerinhalte von `src` sind identisch mit den Daten, die Sie packen und auf dem Server bereitstellen, mit Ausnahme des Ordners `com`. Der `com`-Ordner enthält die nicht kompilierten Klassen (`.java`-Dateien). Diese müssen kompiliert und im Verzeichnis `WEB-INF/classes` gespeichert werden, damit der Anwendungscode darauf zugreifen kann.

Das Verzeichnis `WEB-INF` enthält Code und Konfigurationen, die nicht öffentlich auf dem Webserver bereitgestellt werden. Die anderen Ordner im Stamm des Quellverzeichnisses (`css`, `images` und `js`) sind unter dem entsprechenden Pfad öffentlich auf dem Webserver verfügbar.

Das folgende Beispiel ist identisch mit dem vorherigen Projektverzeichnis, weist aber mehr Dateien und Unterverzeichnisse auf. Dieses Beispielprojekt umfasst einfache Tags, Modell- und Support-Klassen sowie eine JSP (Java Server Pages)-Datei für eine `record`-Ressource. Es enthält zudem eine Formatvorlage und JavaScript für `Bootstrap`, eine JSP-Standarddatei und eine Fehlerseite für 404-Fehler.

`WEB-INF/lib` enthält eine JAR-Datei (Java Archive) mit dem JDBC-Treiber (Java Database Connectivity) für PostgreSQL. `WEB-INF/classes` ist leer, da die Klassendateien noch nicht kompiliert wurden.

```
~/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
```

Erstellen einer WAR-Datei mit einem Shell-Skript

`build.sh` ist ein sehr einfaches Shell-Skript, das Java-Klassen kompiliert, eine WAR-Datei erstellt und diese für lokale Tests in das Tomcat-Verzeichnis `webapps` kopiert:

```
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 ../
```

Die Struktur in der WAR-Datei entspricht der Struktur im Verzeichnis `src` aus dem vorigen Beispiel, mit Ausnahme des Ordners `src/com`. Mit dem Befehl `jar` wird die Datei `META-INF/MANIFEST.MF` automatisch erstellt.

```
~/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
```

Die Verwendung von `.gitignore`

Damit keine kompilierten Klassendateien und WAR-Dateien an das Git-Repository übergeben oder diesbezügliche Meldungen beim Ausführen von Git-Befehlen angezeigt werden, fügen Sie die entsprechenden Dateitypen zur Datei `.gitignore` im Projektordner hinzu:

```
~/workspace/myapp/.gitignore
```

```
*.zip  
.class
```

Konfigurieren des Proxy-Servers Ihrer Tomcat-Umgebung

Die Tomcat-Plattform verwendet [nginx](#) (die Standardeinstellung) oder [Apache HTTP Server](#) als Reverse-Proxy, um Anfragen von Port 80 auf der Instance an Ihren Tomcat Web-Container weiterzuleiten, der auf Port 8080 lauscht. Elastic Beanstalk bietet eine Proxy-Standardkonfiguration, die Sie entweder erweitern oder gänzlich mit einer eigenen Konfiguration überschreiben können.

Alle Amazon Linux 2-Plattformen unterstützen eine einheitliche Proxy-Konfigurationsfunktion. Details zum Konfigurieren des Proxy-Servers auf den neuen Tomcat-Plattformversionen, die Amazon Linux 2 ausführen, finden Sie durch Erweitern des Abschnitts Reverse Proxy-Konfiguration unter [the section called "Erweitern von Linux-Plattformen" \(p. 34\)](#).

Proxy auf der Amazon Linux AMI (vor Amazon Linux 2) Tomcat-Plattform konfigurieren

Falls Ihre Elastic Beanstalk Tomcat-Umgebung eine Amazon Linux AMI-Plattformversion (vor Amazon Linux 2) verwendet, lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

Auswählen eines Proxy-Servers für Ihre Tomcat-Umgebung

Tomcat-Plattformversionen, die auf Amazon Linux-AMIs (vor Amazon Linux 2) basieren, verwenden standardmäßig [Apache 2.4](#) für den Proxy. Sie können [Apache 2.2](#) oder [nginx](#) verwenden, indem Sie eine [Konfigurationsdatei \(p. 722\)](#) in Ihren Quellcode einbinden. Im folgenden Beispiel wird Elastic Beanstalk für die Verwendung von nginx konfiguriert.

Example .ebextensions/nginx-proxy.config

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

Migrieren von Apache 2.2 auf Apache 2.4

Wenn Ihre Anwendung für [Apache 2.2 entwickelt wurde](#), lesen Sie diesen Abschnitt, um mehr über die Migration zu [Apache 2.4](#) zu erfahren.

Beginnend mit Tomcat-Plattform Version 3.0.0- Konfigurationen, die mit der [Java mit Tomcat-Plattformaktualisierung am 24. Mai 2018 veröffentlicht wurden](#), ist Apache 2.4 ist der standardmäßige Proxy der Tomcat-Plattform. Die Apache 2.4 .conf-Dateien sind häufig, aber nicht vollständig abwärtskompatibel zu denjenigen von Apache 2.2. Elastic Beanstalk enthält standardmäßige .conf-Dateien, die einwandfrei mit jeder Apache-Version funktionieren. Wenn Ihre Anwendung die Apache-Konfiguration nicht anpasst, wie unter [Erweitern und Überschreiben der Apache-Standardkonfiguration \(p. 126\)](#) beschrieben , sollte es bei der Migration zu Apache 2.4 keine Probleme geben.

Wenn Ihre Anwendung die Konfiguration von Apache erweitert oder überschreibt, müssen Sie möglicherweise einige Änderungen vornehmen, um zu Apache 2.4 zu migrieren. Weitere Informationen finden Sie unter [Upgrading to 2.4 from 2.2 \(Upgrade von 2.2 auf 2.4\)](#) auf der Website von Apache Software Foundation. Bis zur erfolgreichen Migration auf Apache 2.4 können Sie den Apache 2.2 mit Ihrer Anwendung verwenden, indem Sie die folgende [Konfigurationsdatei \(p. 722\)](#) in Ihren Quellcode aufnehmen.

Example .ebextensions/apache-legacy-proxy.config

```
option_settings:  
  aws:elasticbeanstalk:environment:proxy:  
    ProxyServer: apache/2.2
```

Für eine schnelle Lösung können Sie auch den Proxy-Server in der Elastic Beanstalk-Konsole auswählen.

So wählen Sie den Proxy-Server in Ihrer Tomcat-Umgebung in der Elastic Beanstalk-Konsole aus:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

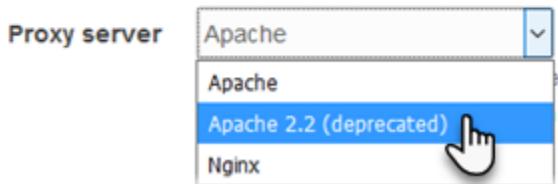
3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.

4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Für Proxy-Server wählen Sie Apache 2.2 (deprecated).
6. Wählen Sie Apply (Anwenden) aus.

Modify software

Container Options

The following settings control container behavior and let you pass key-value pairs in as OS environment variables. [Learn more](#)



Erweitern und Überschreiben der Apache-Standardkonfiguration

Sie können die Elastic Beanstalk-Apache-Standardkonfiguration mit Ihren zusätzlichen Konfigurationsdateien erweitern. Alternativ können Sie die Apache-Standardkonfiguration von Elastic Beanstalk vollständig überschreiben.

Um die Apache-Standardkonfiguration von Elastic Beanstalk zu erweitern, fügen Sie im Quellpaket der Anwendung .conf-Konfigurationsdateien zu einem Ordner namens `.ebextensions/httpd/conf.d` hinzu. Die Apache-Konfiguration von Elastic Beanstalk fügt .conf-Dateien automatisch zu diesem Ordner hinzu.

```
~/workspace/my-app/
|-- .ebextensions
|   -- httpd
|       -- conf.d
|           -- myconf.conf
|           -- ssl.conf
-- index.jsp
```

Beispielsweise wird mit der folgenden Apache 2.4-Konfiguration ein Listener auf Port 5000 hinzugefügt:

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>
```

Um die Apache-Standardkonfiguration von Elastic Beanstalk vollständig zu überschreiben, fügen Sie bei `.ebextensions/httpd/conf/httpd.conf` eine Konfiguration in das Quellpaket ein.

```
~/workspace/my-app/
|-- .ebextensions
|   '-- httpd
|       '-- conf
|           '-- httpd.conf
`-- index.jsp
```

Wenn Sie die Apache-Konfiguration von Elastic Beanstalk überschreiben, fügen Sie folgende Zeilen zu `httpd.conf` hinzu, damit die Konfigurationen von Elastic Beanstalk für [Erweiterte Zustandsberichte und Überwachung \(p. 824\)](#), Antwortkomprimierungen und statische Dateien einbezogen werden.

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

Wenn Ihre Umgebung Apache 2.2 als Proxy verwendet, ersetzen Sie die `IncludeOptional`-Anweisungen durch `Include`. Weitere Informationen über das Verhalten dieser beiden Anweisungen in den beiden Apache-Versionen finden Sie unter [Include in Apache 2.4](#), [IncludeOptional in Apache 2.4](#) und [Include in Apache 2.2](#).

Note

Um den Standard-Listener auf Port 80 zu überschreiben, binden Sie eine Datei mit dem Namen `00_application.conf` bei `.ebextensions/httpd/conf.d/elasticbeanstalk/` ein, um die Konfiguration von Elastic Beanstalk zu überschreiben.

Ein funktionierendes Beispiel für die Elastic Beanstalk-Standardkonfigurationsdatei finden Sie unter `/etc/httpd/conf/httpd.conf` auf einer Umgebungs-Instance. Alle Dateien im Ordner `.ebextensions/httpd` des Quell-Bundles werden bei Bereitstellungen in `/etc/httpd` kopiert.

Erweitern der nginx-Standardkonfiguration

Um die nginx-Standardkonfiguration von Elastic Beanstalk zu überschreiben, fügen Sie `.conf`-Konfigurationsdateien zum Ordner `.ebextensions/nginx/conf.d/` im Quell-Bundle der Anwendung hinzu. Die Elastic Beanstalk nginx-Konfiguration bindet `.conf`-Dateien in diesen Ordner automatisch ein.

```
~/workspace/my-app/
|-- .ebextensions
|   '-- nginx
|       '-- conf.d
|           '-- elasticbeanstalk
|               |-- my-server-conf.conf
|               '-- my-http-conf.conf
`-- index.jsp
```

Dateien mit der Erweiterung `".conf"` im Ordner `conf.d` sind im `http`-Block der Standardkonfiguration enthalten. Dateien im Ordner `conf.d/elasticbeanstalk` befinden sich im `server`-Block innerhalb des `http`-Blocks.

Um die nginx-Standardkonfiguration von Elastic Beanstalk vollständig zu überschreiben, fügen Sie bei `.ebextensions/nginx/nginx.conf` eine Konfiguration in das Quellpaket ein.

```
~/workspace/my-app/
|-- .ebextensions
|   '-- nginx
|       '-- nginx.conf
`-- index.jsp
```

Wenn Sie die nginx-Konfiguration von Elastic Beanstalk überschreiben, fügen Sie folgende Zeile zum `server`-Block der Konfiguration hinzu, damit die Konfigurationen von Elastic Beanstalk für den Listener auf Port 80, Antwortkomprimierungen und statische Dateien einbezogen werden.

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

Note

Um den Standard-Listener auf Port 80 zu überschreiben, binden Sie eine Datei mit dem Namen `00_application.conf` bei `.ebextensions/nginx/conf.d/elasticbeanstalk/` ein, um die Konfiguration von Elastic Beanstalk zu überschreiben.

Fügen Sie zudem die folgende Zeile in den `http`-Block der Konfiguration ein, damit die Konfigurationen von Elastic Beanstalk für [Erweiterte Zustandsberichte und Überwachung \(p. 824\)](#) und Protokollierung einbezogen werden.

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

Ein funktionierendes Beispiel für die Elastic Beanstalk-Standardkonfigurationsdatei finden Sie unter `/etc/nginx/nginx.conf` auf einer Umgebungs-Instance. Alle Dateien im Ordner `.ebextensions/nginx` des Quell-Bundles werden bei Bereitstellungen in `/etc/nginx` kopiert.

Elastic Beanstalk Java SE-Plattform verwenden

Important

Die Amazon Linux 2-Plattformversionen unterscheiden sich grundlegend von den Amazon Linux AMI-Plattformversionen (vor Amazon Linux 2). Diese unterschiedlichen Plattformgenerationen sind auf verschiedene Arten inkompatibel. Wenn Sie zu einer Amazon Linux 2-Plattformversion migrieren, müssen Sie die Informationen in [the section called “Aktualisieren auf Amazon Linux 2” \(p. 503\)](#) lesen.

Die AWS Elastic Beanstalk-Java SE-Plattform umfasst eine Reihe von [Plattformversionen](#) für Java-Webanwendungen, die aus einer kompilierten JAR-Datei eigenständig ausgeführt werden. Sie können Ihre Anwendung lokale kompilieren oder den Quellcode mit einem Build-Skript hochladen, um die Anwendung auf der Instance zu kompilieren. Java SE-Plattformversionen sind in Plattformvarianten gruppiert, von denen jede einer Hauptversion von Java entspricht, zum Beispiel Java 8 und Java 7.

Note

Elastic Beanstalk analysiert die JAR-Datei Ihrer Anwendung nicht. Speichern Sie Dateien, die Elastic Beanstalk benötigt, außerhalb der JAR-Datei auf. Beziehen Sie beispielsweise die `cron.yaml`-Datei einer [Worker-Umgebung \(p. 516\)](#) im Stammverzeichnis des Quell-Bundles Ihrer Anwendung neben der JAR-Datei ein.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für die [Änderung der Konfiguration einer ausgeführten Umgebung \(p. 658\)](#) verfügbar. Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie [gespeicherte Konfigurationen \(p. 764\)](#) verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie [Konfigurationsdateien \(p. 722\)](#) einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Die Java SE-Plattform in Elastic Beanstalk umfasst einen [nginx](#)-Server, der als Reverse-Proxy-Server fungiert und zwischengespeicherte statische Inhalte liefert sowie Anforderungen an die Anwendung weiterleitet. Die Plattform stellt auch Konfigurationsoptionen bereit, um den Proxy-Server zur Verarbeitung statischer Komponenten aus einem Ordner im Quellcode zu konfigurieren und so die Auslastung für die Anwendung zu reduzieren. In erweiterten Szenarien können Sie [eigene CONF-Dateien in das Quell-Bundle einbinden \(p. 132\)](#), um die Proxy-Konfiguration von Elastic Beanstalk zu erweitern oder zu überschreiben.

Wenn Sie nur eine einzelne JAR-Datei für Ihre Anwendungsquelle bereitstellen (allein, nicht innerhalb eines Quellpakets), benennt Elastic Beanstalk die JAR-Datei in `application.jar` um und führt sie dann mit `java -jar application.jar` aus. Schließen Sie eine optionale [Procfile-Datei \(p. 131\)](#) in Ihr Quell-Bundle ein, um die Prozesse zu konfigurieren, die auf den Server-Instances in Ihrer Umgebung ausgeführt werden. Wenn Ihr Quell-Bundle mehrere JAR-Dateien enthält oder wenn Sie den Java-Befehl zum Festlegen von JVM-Optionen anpassen möchten, benötigen Sie eine [Procfile-Datei](#).

Wir empfehlen, immer eine [Procfile](#) im Quellpaket mit Ihrer Anwendung bereitzustellen. Auf diese Weise können Sie genau steuern, welche Prozesse Elastic Beanstalk für Ihre Anwendung ausführt und welche Argumente diese Prozesse erhalten.

Damit Java-Klassen kompiliert und weitere Build-Befehle auf den EC2-Instances der Umgebung während der Bereitstellung ausgeführt werden können, binden Sie eine [Buildfile \(p. 131\)](#)-Datei in das Quell-Bundle der Anwendung ein. Mit einer [Buildfile](#)-Datei können Sie Quellcode direkt bereitstellen und den Server-Build ausführen, anstatt JAR-Dateien lokal zu kompilieren. Die Java SE-Plattform bietet gängige Build-Tools für Server-Builds.

Details zu den verschiedenen Möglichkeiten zum Erweitern einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#).

Konfigurieren der Java SE-Umgebung

Mit den Einstellungen der Java SE-Plattform können Sie das Verhalten Ihrer Amazon EC2-Instances optimieren. Sie können die Amazon EC2-Instance-Konfiguration der Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-Konsole bearbeiten.

Mithilfe der Elastic Beanstalk-Konsole aktivieren Sie die Protokollrotation an Amazon S3 und konfigurieren Variablen, die die Anwendung aus der Umgebung auslesen kann.

So konfigurieren Sie Ihre Java SE-Umgebung in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren): gibt an, ob die Protokolldateien für die Amazon EC2-Instances der Anwendung in den zugeordneten Amazon S3-Bucket kopiert werden sollen.

Statische Dateien

Um die Leistung zu verbessern, können Sie im Abschnitt Static files (Statische Dateien) den Proxy-Server so konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer

Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zur Verzeichniszuordnung fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, statt die Anfrage an Ihre Anwendung weiterzuleiten.

Einzelheiten zur Konfiguration statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter [the section called "Statische Dateien" \(p. 776\)](#).

Umgebungseigenschaften

Im Bereich Environment Properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Umgebungseigenschaften werden als Schlüssel-Wert-Paare an die Anwendung weitergeleitet.

In der Java SE-Umgebung mit Elastic Beanstalk-Ausführung kann über `System.getenv()` auf die Umgebungsvariablen zugegriffen werden. Beispielsweise können Sie mit folgendem Code eine Eigenschaft mit dem Namen `API_ENDPOINT` als Variable lesen:

```
String endpoint = System.getenv("API_ENDPOINT");
```

Weitere Informationen finden Sie unter [Umgebungseigenschaften und andere Softwareeinstellungen \(p. 621\)](#).

Java SE-Konfigurations-Namespace

Mithilfe einer [Konfigurationsdatei \(p. 722\)](#) können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Konfigurationsoptionen können durch den Elastic Beanstalk-Service oder die verwendete Plattform definiert und in Namespaces organisiert werden.

Die Java SE-Plattform definiert keine plattformspezifischen Namespaces. Sie können den Proxy so konfigurieren, dass statische Dateien mittels des Namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` bereitgestellt werden. Einzelheiten und ein Beispiel finden Sie unter [the section called "Statische Dateien" \(p. 776\)](#).

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Die Java SE-Plattform im Amazon Linux-AMI (vor Amazon Linux 2)

Falls Ihre Elastic Beanstalk Java SE-Umgebung eine Amazon Linux AMI-Plattformversion (vor Amazon Linux 2) verwendet, lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

Java SE-Konfigurations-Namespace

Mithilfe einer [Konfigurationsdatei \(p. 722\)](#) können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Konfigurationsoptionen können durch den Elastic Beanstalk-Service oder die verwendete Plattform definiert und in Namespaces organisiert werden.

Die Java SE-Plattform unterstützt zusätzlich zu den [von allen Plattformen unterstützten Namespaces \(p. 666\)](#) eine weitere Konfigurationsoption für einen plattformspezifischen Namespace. Mit dem Namespace `aws:elasticbeanstalk:container:java:staticfiles` können Sie Optionen definieren, um für die Webanwendung Pfade zu Ordnern im Quell-Bundle der Anwendung zuzuweisen, die statische Inhalte enthalten.

Zum Beispiel werden mit diesem [option_settings \(p. 724\)](#)-Snippet zwei Optionen im Namespace der statischen Dateien definiert. Die erste ordnet den Pfad /public dem Ordner public zu, die zweite ordnet den Pfad /images dem Ordner img zu:

```
option_settings:  
  aws:elasticbeanstalk:container:java:staticfiles:  
    /html: statichtml  
    /images: staticimages
```

Die über diesen Namespace zugewiesenen Ordner müssen sich tatsächlich im Stamm des Quell-Bundles befinden. Sie können keinen Pfad zu einem Ordner in einer JAR-Datei zuweisen.

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Erstellen von JAR-Dateien auf dem Server mit einer Buildfile-Datei

Sie können die Klassendateien und die JAR-Datei(en) der Anwendung auf den EC2-Instances der Umgebung erstellen, indem Sie den Build-Befehl aus einer [Buildfile](#)-Datei im Quell-Bundle aufrufen.

Befehle in einer [Buildfile](#)-Datei werden nur einmal ausgeführt und müssen anschließend beendet werden, wohingegen die Befehle in einer [Procfile \(p. 131\)](#) normalerweise für die Lebensdauer der Anwendung ausgeführt und im Falle einer Beendigung neu gestartet werden. Verwenden Sie für die Ausführung der JAR-Dateien in der Anwendung eine [Procfile](#).

Details zu Platzierung und Syntax einer [Buildfile](#) erhalten Sie, indem Sie den Abschnitt [Buildfile](#) und [Procfile](#) unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#) erweitern.

In obigem [Buildfile](#)-Beispiel wird Apache Maven ausgeführt, um eine Webanwendung aus Quellcode zu erstellen. Eine Beispielanwendung mit dieser Funktion mit dieser Funktion finden Sie unter [Java-Webanwendungsbeispiele \(p. 110\)](#).

Example Buildfile

```
build: mvn assembly:assembly -DdescriptorId=jar-with-dependencies
```

Die Java SE-Plattform enthält die folgenden Build-Tools, die Sie über das Build-Skript aufrufen können:

- `javac` – Java-Compiler
- `ant` – Apache Ant
- `mvn` – Apache Maven
- `gradle` – Gradle

Konfigurieren der Anwendung mit einer Procfile-Datei

Sofern im Stamm des Quell-Bundles der Anwendung mehr als eine JAR-Datei vorhanden ist, müssen Sie eine [Procfile](#)-Datei einbinden, anhand derer Elastic Beanstalk erkennt, welche JAR-Dateien(en) ausgeführt werden sollen. Bei einer einzelnen JAR-Anwendung können Sie auch eine [Procfile](#)-Datei einbinden. Damit wird die JVM (Java Virtual Machine) konfiguriert, auf der die Anwendung ausgeführt wird.

Wir empfehlen, immer eine [Procfile](#) im Quellpaket mit Ihrer Anwendung bereitzustellen. Auf diese Weise können Sie genau steuern, welche Prozesse Elastic Beanstalk für Ihre Anwendung ausführt und welche Argumente diese Prozesse erhalten.

Einzelheiten zum Schreiben und Verwenden eines `Procfile` finden Sie im Abschnitt `Buildfile` und `Procfile` unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#).

Example `Procfile`

```
web: java -jar server.jar -Xms256m
cache: java -jar mycache.jar
web_foo: java -jar other.jar
```

Der Befehl für die Ausführung der wichtigsten JAR-Datei in der Anwendung muss den Namen `web` tragen und der erste Befehl in der `Procfile`-Datei sein. Der nginx-Server leitet alle HTTP-Anforderungen, die er vom Load Balancer der Umgebung erhält, an diese Anwendung weiter.

Elastic Beanstalk geht davon aus, dass alle Einträge in der `Procfile`-Datei stets ausgeführt werden sollen. Daher wird automatisch jede Anwendung, die in der `Procfile`-Datei definiert ist und beendet wird, neu gestartet. Wenn die Anwendung mit einem Befehl beendet, aber nicht wieder neu gestartet werden soll, verwenden Sie eine [Buildfile \(p. 131\)](#)-Datei.

Procfile auf Amazon Linux-AMI (vor Amazon Linux 2) verwenden

Falls Ihre Elastic Beanstalk Java SE-Umgebung eine Amazon Linux AMI-Plattformversion (vor Amazon Linux 2) verwendet, lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

Übergeben eines Portwertes

Standardmäßig wird der nginx-Proxy von Elastic Beanstalk so konfiguriert, dass alle Anforderungen an die Anwendung über Port 5000 weitergeleitet werden. Diese Port-Standardeinstellung können Sie überschreiben. Legen Sie dazu die [PORT-Umgebungseigenschaft \(p. 129\)](#) auf den Überwachungspunkt der Hauptanwendung fest.

Wenn Sie einen `Procfile` verwenden, um mehrere Anwendungen auszuführen, erwartet Elastic Beanstalk auf den AMI-Plattformversionen von Amazon Linux, dass jede zusätzliche Anwendung auf einem Port lauscht, der 100 höher ist als der vorherige. Elastic Beanstalk legt die `PORT`-Variable, auf die von jeder Anwendung zugegriffen werden kann, auf den Port fest, auf den die Anwendung erwartungsgemäß zugreift. Sie können auf diese Variable im Anwendungscode zugreifen, indem Sie `System.getenv("PORT")` aufrufen.

Im vorherigen Beispiel für `Procfile` wurden für die `web`-Anwendung der Überwachungspunkt 5000, für `cache` der Überwachungspunkt 5100 und für `web_foo` der Überwachungspunkt 5200 konfiguriert. Für `web` wird der Überwachungspunkt durch Auslesen der `PORT`-Variable und Addieren der Zahl 100 konfiguriert. Damit wird der Überwachungspunkt von `cache` für die Anforderungsübermittlung bestimmt.

Konfigurieren des Reverse-Proxy-Servers

Elastic Beanstalk verwendet `nginx` als Reverse-Proxy, um Ihre Anwendung Ihrem Elastic Load Balancing Load Balancer auf Port 80 zuzuordnen. Elastic Beanstalk bietet eine nginx-Standardkonfiguration, die Sie entweder erweitern oder gänzlich mit einer eigenen Konfiguration überschreiben können.

Standardmäßig wird der nginx-Proxy von Elastic Beanstalk so konfiguriert, dass alle Anforderungen an die Anwendung über Port 5000 weitergeleitet werden. Diese Port-Standardeinstellung können Sie überschreiben. Legen Sie dazu die [PORT-Umgebungseigenschaft \(p. 129\)](#) auf den Überwachungspunkt der Hauptanwendung fest.

Note

Der Überwachungspunkt der Anwendung kollidiert nicht mit dem Überwachungspunkt des nginx-Servers, von dem dieser die Anforderungen des Load Balancers empfängt.

Alle Amazon Linux 2-Plattformen unterstützen eine einheitliche Proxy-Konfigurationsfunktion. Details zum Konfigurieren des Proxy-Servers auf den neuen Amazon Corretto-Plattformversionen, die Amazon Linux

2 ausführen, erhalten Sie, finden Sie den Abschnitt Reverse Proxy-Konfiguration unter [the section called "Erweitern von Linux-Plattformen" \(p. 34\)](#) erweitern.

Proxy auf Amazon Linux-AMI (vor Amazon Linux 2) konfigurieren

Falls Ihre Elastic Beanstalk Java SE-Umgebung eine Amazon Linux AMI-Plattformversion (vor Amazon Linux 2) verwendet, lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

Erweitern und Überschreiben der Standard-Proxykonfiguration

Um die nginx-Standardkonfiguration von Elastic Beanstalk zu überschreiben, fügen Sie `.conf`-Konfigurationsdateien zum Ordner `.ebextensions/nginx/conf.d/` im Quell-Bundle der Anwendung hinzu. Die nginx-Konfiguration von Elastic Beanstalk enthält standardmäßig `.conf`-Dateien in diesem Ordner.

```
~/workspace/my-app/
|-- .ebextensions
|   '-- nginx
|       '-- conf.d
|           '-- myconf.conf
`-- web.jar
```

Um die nginx-Standardkonfiguration von Elastic Beanstalk vollständig zu überschreiben, binden Sie bei `.ebextensions/nginx/nginx.conf` eine Konfiguration in Ihr Quell-Bundle ein:

```
~/workspace/my-app/
|-- .ebextensions
|   '-- nginx
|       '-- nginx.conf
`-- web.jar
```

Wenn Sie die nginx-Konfiguration von Elastic Beanstalk überschreiben, fügen Sie folgende Zeile zu `nginx.conf` hinzu, damit die Konfigurationen von Elastic Beanstalk für [Erweiterte Zustandsberichte und Überwachung \(p. 824\)](#), automatische Anwendungszuordnungen und statische Dateien einbezogen werden.

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

Die folgende Beispielkonfiguration aus der [Beispielanwendung Scorekeep](#) überschreibt die Standardkonfiguration von Elastic Beanstalk, um eine statische Webanwendung aus dem public-Unterverzeichnis von `/var/app/current` zu bedienen, wohin die Java SE-Plattform den Anwendungsquellcode kopiert. Der `/api`-Standort leitet den Datenverkehr zu Routen unter `/api/` an die Spring-Anwendung weiter, die den Port 5000 liest. Jeder andere Datenverkehr wird von der Web-Anwendung auf dem Root-Pfad bedient.

Example `.ebextensions/nginx/nginx.conf`

```
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 / {

        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;
    }
}
```

Hinzufügen einer Amazon RDS-DB-Instance zur Java-Anwendungsumgebung

Sie können eine Amazon Relational Database Service (Amazon RDS)-DB-Instance verwenden, um Daten zu speichern, die Ihre Anwendung sammelt und modifiziert. Die Datenbank kann Ihrer Umgebung angefügt und von Elastic Beanstalk verwaltet werden oder sie kann extern erstellt und verwaltet werden.

Bei der ersten Verwendung von Amazon RDS fügen Sie in der Testumgebung mithilfe der Elastic Beanstalk-Konsole eine DB-Instance hinzu und überprüfen, ob die Anwendung eine Verbindung zu dieser herstellen kann.

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie Apply (Anwenden) aus.

Das Hinzufügen einer DB-Instance dauert ca. 10 Minuten. Sobald die Umgebungsaktualisierung abgeschlossen ist, stehen der Hostname der DB-Instance und andere Verbindungsinformationen über die folgenden Umgebungseigenschaften zur Verfügung:

Eigenschaftenname	Beschreibung	Eigenschaftswert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbankname, ebdb	Auf der Registerkarte Configuration (Konfiguration) der Amazon RDS-Konsole: DB-Name.
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	In der Amazon RDS-Konsole nicht als Referenz verfügbar.

Weitere Informationen zum Konfigurieren einer internen DB-Instance finden Sie unter [Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung \(p. 609\)](#). Anweisungen zum Konfigurieren einer externen Datenbank für die Verwendung mit Elastic Beanstalk finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon RDS \(p. 973\)](#).

Für die Verbindungserstellung zu einer Datenbank fügen Sie die geeignete JAR-Treiberdatei zur Anwendung hinzu, laden die Klasse in den Code und erstellen ein Verbindungsobjekt mit den von Elastic Beanstalk bereitgestellten Umgebungseigenschaften.

Abschnitte

- [Herunterladen eines JDBC-Treibers \(p. 136\)](#)
- [Verbinden mit einer Datenbank \(Java SE-Plattformen\) \(p. 136\)](#)
- [Verbinden mit einer Datenbank \(Tomcat-Plattformen\) \(p. 137\)](#)

- Fehlerbehebung bei Datenbankverbindungen (p. 139)

Herunterladen eines JDBC-Treibers

Sie benötigen die JAR-Datei des JDBC-Treibers für die DB-Engine, die Sie gewählt haben. Speichern Sie die JAR-Datei in Ihrem Quellcode und fügen Sie sie in Ihren Klassenpfad ein, wenn Sie die Klasse kompilieren, die Verbindungen mit der Datenbank herstellt.

Die neuesten Treiber für Ihre DB-Engine finden Sie an den folgenden Speicherorten:

- MySQL – [MySQL Connector/J](#)
- Oracle SE-1 – [Oracle JDBC Driver](#)
- Postgres – [PostgreSQL JDBC Driver](#)
- SQL Server – [Microsoft JDBC Driver](#)

Um den JDBC-Treiber zu verwenden, rufen Sie `Class.forName()` auf, um ihn zu laden, bevor Sie die Verbindung mit `DriverManager.getConnection()` in Ihrem Code herstellen.

JDBC verwendet eine Verbindungszeichenfolge mit folgendem Format:

```
jdbc:driver://hostname:port/dbName?user=userName&password=password
```

Sie können den Hostnamen, Port, Datenbanknamen, Benutzernamen und das Passwort von den Umgebungsvariablen abrufen, die Elastic Beanstalk für Ihre Anwendung bereitstellt. Der Treibername ist spezifisch für Ihren Datenbanktyp und Ihre Treiberversion. Es folgen Beispiele für Treibernamen:

- mysql for MySQL
- postgresql for PostgreSQL
- oracle:thin for Oracle Thin
- oracle:oci for Oracle OCI
- oracle:oci8 for Oracle OCI 8
- oracle:kprb for Oracle KPRB
- sqlserver for SQL Server

Verbinden mit einer Datenbank (Java SE-Plattformen)

In einer Java SE-Umgebung verwenden Sie `System.getenv()`, um die Verbindungsvariablen aus der Umgebung zu lesen. Der folgende Beispielcode zeigt eine Klasse, die eine Verbindung zu einer PostgreSQL-Datenbank erstellt.

```
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);  
        } catch (Exception e) {  
            logger.error("Error connecting to database: " + e.getMessage());  
        }  
    }  
}
```

```
        logger.info("Remote connection successful.");
        return con;
    }
    catch (ClassNotFoundException e) { logger.warn(e.toString());}
    catch (SQLException e) { logger.warn(e.toString());}
}
return null;
}
```

Verbinden mit einer Datenbank (Tomcat-Plattformen)

In einer Tomcat-Umgebung werden Umgebungseigenschaften als Systemeigenschaften bereitgestellt, die mit `System.getProperty()` zugreifbar sind.

Der folgende Beispielcode zeigt eine Klasse, die eine Verbindung zu einer PostgreSQL-Datenbank erstellt.

```
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;
}
```

Wenn Sie Probleme haben, eine Verbindung herzustellen oder SQL-Anweisungen auszuführen, versuchen Sie, den folgenden Code in einer JSP-Datei zu platzieren. Dieser Code stellt eine Verbindung zu einer DB-Instance her, erstellt eine Tabelle und schreibt in sie.

```
<%@ 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) {}
}
%>
```

Zum Anzeigen der Ergebnisse platzieren Sie den folgenden Code im Text des HTML-Teils der JSP-Datei.

```
<p>Established connection to RDS. Read first two rows: <%= results %></p>
```

Fehlerbehebung bei Datenbankverbindungen

Wenn Sie auf Probleme mit der Verbindung zu einer Datenbank über Ihre Anwendung stoßen, überprüfen Sie das Web-Container-Protokoll und die Datenbank.

Überprüfen von Protokollen

Sie können alle Protokolle aus Ihrer Elastic Beanstalk-Umgebung in Eclipse anzeigen. Wenn Sie die AWS Explorer-Ansicht nicht geöffnet haben, wählen Sie den Pfeil neben dem orangefarbenen AWS-Symbol in der Menüleiste und wählen Sie Show AWS Explorer View. Erweitern Sie AWS Elastic Beanstalk und Ihren Umgebungsnamen und öffnen Sie dann das Kontextmenü (rechte Maustaste) für den Server. Klicken Sie auf Open in WTP Server Editor.

Wählen Sie die Registerkarte Log der Server-Ansicht, um die aggregierten Protokolle aus Ihrer Umgebung zu sehen. Um die neuesten Protokolle zu öffnen, wählen Sie die Schaltfläche Refresh oben rechts auf der Seite.

Scrollen Sie nach unten, um die Tomcat-Protokolle in `/var/log/tomcat7/catalina.out` zu suchen. Wenn Sie die Webseite aus unserem oben stehenden Beispiel mehrmals geladen haben, sehen Sie möglicherweise Folgendes:

```
-----
/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.
```

Alle Informationen, die die Webanwendung an die Standardausgabe sendet, wird im Web-Container-Protokoll angezeigt. Im vorherigen Beispiel versucht die Anwendung, die Tabelle jedes Mal zu erstellen, wenn die Seite geladen wird. Dies führt zur Erfassung einer SQL-Ausnahme auf jeder Seite, die nach der ersten geladen wird.

Beispielsweise ist die vorhergehende akzeptabel. Aber in tatsächlichen Anwendungen behalten Sie Ihre Datenbankdefinitionen in Schemaobjekten, führen Transaktionen in Modellklassen durch und koordinieren Anforderungen mit Controller-Servlets.

Herstellen einer Verbindung mit einer RDS DB-Instance

Sie können eine direkte Verbindung zur RDS DB-Instance in Ihrer Elastic Beanstalk-Umgebung herstellen, indem Sie die MySQL-Client-Anwendung verwenden.

Öffnen Sie zunächst die Sicherheitsgruppe für Ihre RDS DB-Instance, um Datenverkehr von Ihrem Computer zuzulassen.

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.

4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie neben Endpoint (Endpunkt) den Link zur Amazon RDS-Konsole aus.
6. Wählen Sie auf der Detailseite der RDS Dashboard-Instance unter Security and Network die Sicherheitsgruppe beginnend mit rds- neben Security Groups.

Note

Die Datenbank kann über mehrere Einträge mit der Bezeichnung Security Groups verfügen. Wenn Sie ein älteres Konto haben, das nicht standardmäßig über [Amazon Virtual Private Cloud](#) (Amazon VPC) verfügt, verwenden Sie die erste, die mit awseb beginnt.

7. Rufen Sie in Security group details die Registerkarte Inbound auf und klicken Sie auf Edit.
8. Fügen Sie eine Regel für MySQL (Port 3306) hinzu, die Datenverkehr von Ihrer IP-Adresse erlaubt, angegeben im CIDR-Format.
9. Wählen Sie Save aus. Die Änderungen werden sofort wirksam.

Kehren Sie zu den Elastic Beanstalk-Konfigurationsdetails für Ihre Umgebung zurück und beachten Sie den Endpunkt. Sie verwenden den Domänennamen für die Verbindung mit der RDS-DB-Instance.

Installieren Sie den MySQL-Client und initiieren Sie eine Verbindung mit der Datenbank auf Port 3306. Installieren Sie unter Windows MySQL Workbench von der MySQL-Startseite und befolgen Sie die Anweisungen.

Installieren Sie unter Linux den MySQL-Client mithilfe des Paket-Managers für Ihre Verteilung. Das folgende Beispiel funktioniert auf Ubuntu und anderen Debian-Derivaten.

```
// 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
...
```

Nachdem Sie eine Verbindung hergestellt haben, können Sie SQL-Befehle ausfehlen, um den Status der Datenbank, unabhängig davon, ob Ihre Tabellen und Zeilen erstellt wurden, und andere Informationen anzuzeigen.

```
mysql> SELECT Resource from Beanstalk;
+-----+
| Resource      |
+-----+
| EC2 Instance  |
| RDS Instance  |
+-----+
2 rows in set (0.01 sec)
```

Verwenden des AWS Toolkit for Eclipse

Das AWS Toolkit for Eclipse integriert AWS Elastic Beanstalk-Verwaltungsfunktionen in Ihre Tomcat-Entwicklungsumgebung, um die Umgebungserstellung, Konfiguration und Codebereitstellung zu

erleichtern. Das Toolkit bietet Unterstützung für mehrere AWS-Konten, die Verwaltung von vorhandenen Umgebungen und eine direkte Verbindung zu Instances in Ihrer Umgebung für die Fehlerbehebung.

Note

Das AWS Toolkit for Eclipse unterstützt nur Projekte, die die Java mit Tomcat-Plattform und nicht die Java SE-Plattform verwenden.

Weitere Informationen zu Voraussetzungen und die Installation des AWS Toolkit for Eclipse finden Sie unter <http://aws.amazon.com/eclipse>. Sie können sich auch das Video [Using AWS Elastic Beanstalk with the AWS Toolkit for Eclipse](#) ansehen. Dieses Thema bietet auch nützliche Informationen zu Tools, Themen zur Vorgehensweise und zusätzliche Ressourcen für Java-Entwickler.

Importieren von vorhandenen Umgebungen in Eclipse

Sie können vorhandene Umgebungen, die Sie in der AWS Management Console erstellt haben, in Eclipse importieren.

Zum Importieren vorhandener Umgebungen erweitern Sie den AWS Elastic Beanstalk-Knoten und doppelklicken Sie auf eine Umgebung im AWS Explorer in Eclipse. Sie können nun Ihre Elastic Beanstalk-Anwendungen in dieser Umgebung bereitstellen.

Elastic Beanstalk-Anwendungsumgebungen verwalten

Themen

- [Ändern der Umgebungskonfigurationseinstellungen \(p. 141\)](#)
- [Ändern des Umgebungstyps \(p. 142\)](#)
- [EC2 Server-Instances mit dem AWS Toolkit for Eclipse konfigurieren \(p. 143\)](#)
- [Elastic Load Balancing mit dem AWS Toolkit for Eclipse konfigurieren \(p. 145\)](#)
- [Auto Scaling mit dem AWS Toolkit for Eclipse konfigurieren \(p. 148\)](#)
- [Konfigurieren von Benachrichtigungen mit dem AWS Toolkit for Eclipse \(p. 150\)](#)
- [Java-Container mit AWS Toolkit for Eclipse konfigurieren \(p. 151\)](#)
- [Festlegen von Systemeigenschaften mit AWS Toolkit for Eclipse \(p. 152\)](#)

Mit dem AWS Toolkit for Eclipse können Sie die Bereitstellung und Konfiguration der AWS-Ressourcen ändern, die von Ihren Anwendungsumgebungen verwendet werden. Weitere Informationen zur Verwaltung Ihres Anwendungsumgebungen mithilfe der AWS Management Console finden Sie unter [Verwalten von Umgebungen \(p. 425\)](#). In diesem Abschnitt werden die spezifischen Service-Einstellungen beschrieben, die Sie im AWS Toolkit for Eclipse als Teil Ihrer Anwendungsumgebungskonfiguration bearbeiten können. Weitere Informationen über das AWS Toolkit for Eclipse finden Sie unter [AWS Toolkit for Eclipse – Erste Schritte](#).

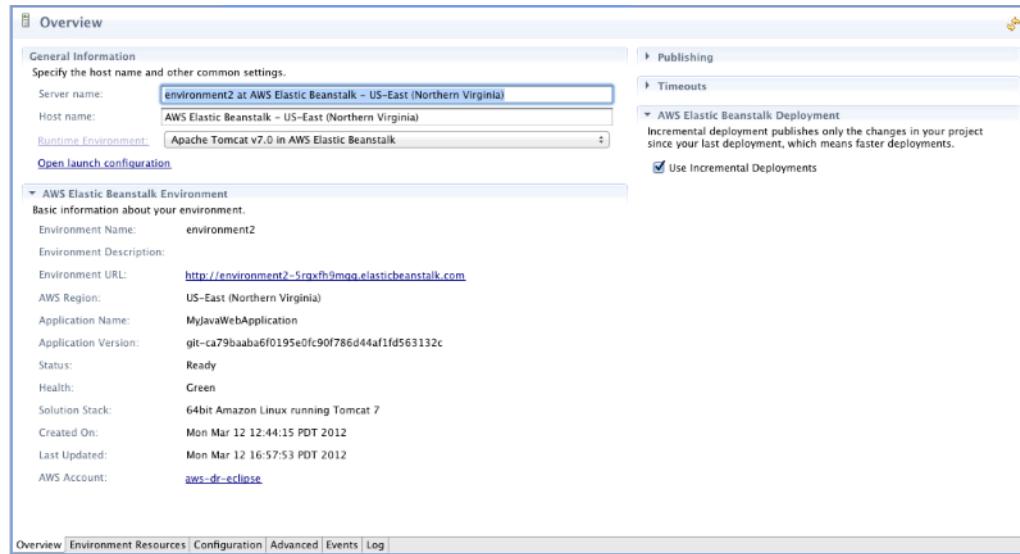
Ändern der Umgebungskonfigurationseinstellungen

Wenn Sie Ihre Anwendung bereitstellen, konfiguriert Elastic Beanstalk eine Reihe von AWS Cloud Computing-Services. Sie können mit dem AWS Toolkit for Eclipse steuern, wie diese einzelnen Services konfiguriert sind.

So bearbeiten Sie die Umgebungseinstellungen einer Anwendung

1. Wenn Eclipse die Ansicht AWS Explorer nicht anzeigt, wählen Sie im Menü Window (Fenster), Show View (Ansicht anzeigen), AWS Explorer aus. Erweitern Sie den Elastic Beanstalk-Knoten und Ihren Anwendungsknoten.

2. Klicken Sie im AWS Explorer doppelt auf Ihre Elastic Beanstalk-Umgebung.
3. Klicken Sie unten im Fenster auf die Registerkarte Configuration (Konfiguration).



Sie können nun Einstellungen für Folgendes konfigurieren:

- EC2-Server-Instances
- Load Balancer
- Auto Scaling
- Benachrichtigungen
- Umgebungstypen
- Umgebungseigenschaften

Ändern des Umgebungstyps

In AWS Toolkit for Eclipse können Sie im Abschnitt Environment Type (Umgebungstyp) der Registerkarte Configuration (Konfiguration) Ihrer Umgebung entweder eine Load balanced, auto scaled (lastenverteilt, automatisch skaliert) Umgebung oder eine Umgebung mit Single instance (einer Instanz) auswählen, je nach den Anforderungen der Anwendung, die Sie bereitstellen. Für eine Anwendung, die Skalierbarkeit erfordert, wählen Sie Load balanced, auto scaled aus. Für eine einfache Anwendung mit geringem Datenverkehr wählen Sie Single instance. Weitere Informationen finden Sie unter [Umgebungstypen \(p. 514\)](#).

Environment Type

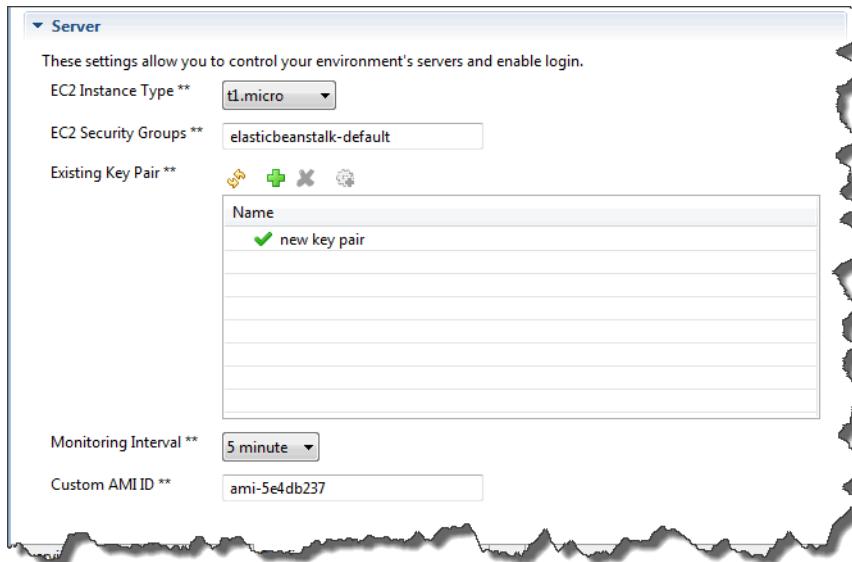
Select an environment type, either load balanced and auto scaled or single instance. A load-balanced, auto-scaled environment automatically distributes traffic across multiple Amazon EC2 instances and can stop and start instances based on demand. A single-instance environment includes just a single Amazon EC2 instance, which costs less.

Environment Type SingleInstance ▾

EC2 Server-Instances mit dem AWS Toolkit for Eclipse konfigurieren

Amazon Elastic Compute Cloud (EC2) ist ein Webservice für das Starten und Verwalten von Server-Instances in den Rechenzentren von Amazon. Sie können Amazon EC2-Server-Instances zu jedem beliebigen Zeitpunkt, so lange wie nötig und für alle rechtlichen Zwecke verwenden. Instances sind in verschiedenen Größen und Konfigurationen verfügbar. Weitere Informationen finden Sie auf der [Produktseite für Amazon EC2](#).

Unter Server auf der Registerkarte Konfiguration Ihrer Umgebung innerhalb des Toolkits for Eclipse können Sie die Amazon EC2 Instance-Konfiguration der Elastic Beanstalk-Umgebung bearbeiten.



Amazon EC2-Instance-Typen

Instance type (Instance-Typ) zeigt die für Ihre Elastic Beanstalk-Anwendung verfügbaren Instance-Typen an. Ändern Sie den Instance-Typ, um einen Server mit den Merkmalen (einschließlich Speichergröße und CPU-Leistung) auszuwählen, die für Ihre Anwendung am besten geeignet sind. Anwendungen mit aufwändigen und lange dauernden Operationen erfordern mehr CPU oder Speicher.

Weitere Informationen über die für die Elastic Beanstalk-Anwendung verfügbaren Amazon EC2-Instance-Typen finden Sie unter [Instance-Typen](#) im Amazon EC2-Benutzerhandbuch.

Amazon EC2-Sicherheitsgruppen

Sie können den Zugriff auf Ihre Elastic Beanstalk-Anwendung mit einer Amazon EC2-Sicherheitsgruppe steuern. Eine Sicherheitsgruppe definiert Firewall-Regeln für Ihre Instances. Diese Regeln legen fest, welcher eingehende Netzwerkverkehr an Ihre Instance übertragen werden soll. Anderer eingehender Datenverkehr wird verworfen. Sie können die Regeln für eine Gruppe jederzeit ändern. Die neuen Regeln werden automatisch für alle laufenden und künftigen Instances durchgesetzt.

Sie können Ihre Amazon EC2-Sicherheitsgruppen über die AWS-Managementkonsole oder mit dem AWS Toolkit for Eclipse einrichten. Sie können angeben, welche Amazon EC2-Sicherheitsgruppen den Zugriff auf Ihre Elastic Beanstalk-Anwendung steuern, indem Sie die Namen von einer oder mehreren Amazon EC2-Sicherheitsgruppen (durch Kommas getrennt) in das Textfeld EC2 Security Groups (EC2-Sicherheitsgruppen) eingeben.

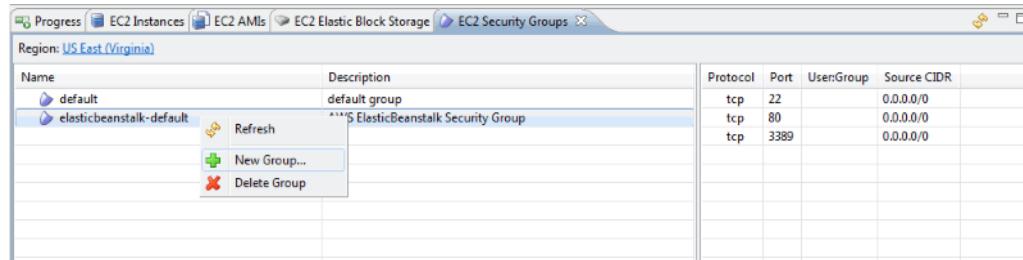
Note

Wenn Sie Ihre Anwendung mit einem älteren Container-Typen ausführen, stellen Sie sicher, dass Port 80 (HTTP) von 0.0.0.0/0 als Quell-CIDR-Bereich zugänglich ist, wenn Sie Zustandsprüfungen

für Ihre Anwendung aktivieren möchten. Weitere Informationen zu Zustandsprüfungen finden Sie unter [Zustandsprüfungen \(p. 147\)](#). Wenn Sie überprüfen möchten, ob Sie einen Legacy-Container verwenden, erhalten Sie Informationen unter [the section called “Warum sind einige Plattformversionen als Legacy gekennzeichnet?” \(p. 502\)](#).

So erstellen Sie eine Sicherheitsgruppe mit dem AWS Toolkit for Eclipse

1. Klicken Sie im AWS Toolkit for Eclipse auf AWS Explorer. Erweitern Sie den Amazon EC2-Knoten und klicken Sie dann doppelt auf Security Groups (Sicherheitsgruppen).
2. Klicken Sie mit der rechten Maustaste auf eine beliebige Stelle in der linken Tabelle und klicken Sie dann auf New Group (Neue Gruppe).



3. Geben Sie im Dialogfeld Security Group (Sicherheitsgruppe) den Namen und eine Beschreibung für die Sicherheitsgruppe ein und wählen Sie dann OK aus.

Weitere Informationen zu Amazon EC2-Sicherheitsgruppen finden Sie unter [Using Security Groups](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Amazon EC2-Schlüsselpaare

Mit einem Amazon EC2-Schlüsselpaar können Sie sich sicher an den für Ihre Elastic Beanstalk-Anwendung bereitgestellten Amazon EC2-Instances anmelden.

Important

Bevor Sie auf die von Amazon EC2 bereitgestellten Amazon EC2-Instances zugreifen können, müssen Sie zunächst ein Amazon EC2-Schlüsselpaar erstellen und die von Amazon EC2 bereitgestellten Amazon EC2-Instances zur Verwendung des Amazon EC2-Schlüsselpaares konfigurieren. Sie können ein Schlüsselpaar mit dem Assistenten Publish to Beanstalk Wizard im AWS Toolkit for Eclipse erstellen, wenn Sie Ihre Anwendung für Elastic Beanstalk bereitstellen. Alternativ können Sie Ihre Amazon EC2-Schlüsselpaare mit der [AWS-Managementkonsole](#) einrichten. Anleitungen zum Erstellen eines Schlüsselpaares für Amazon EC2 finden Sie unter [Amazon Elastic Compute Cloud Getting Started Guide](#).

Weitere Informationen zu Amazon EC2-Schlüsselpaaren finden Sie unter [Using Amazon EC2 Credentials](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch. Weitere Informationen zum Herstellen einer Verbindung zu Amazon EC2-Instances finden Sie unter [Connect to Your Instance](#) und [Connecting to a Linux/UNIX Instance from Windows using PuTTY](#) im Amazon EC2 User Guide.

CloudWatch-Kennzahlen

Standardmäßig werden nur grundlegende Amazon CloudWatch-Metriken aktiviert. Sie geben Daten in 5-Minuten-Intervallen zurück. Sie können noch genauere CloudWatch-Metriken in 1-Minuten-Intervallen aktivieren, indem Sie auf der Registerkarte Configuration im Abschnitt Server für den Wert Monitoring Interval den Wert 1 minute für Ihre Umgebung im AWS Toolkit for Eclipse auswählen.

Note

Für Metriken in 1-Minuten-Intervallen können Amazon CloudWatch-Servicegebühren anfallen. Weitere Informationen finden Sie unter [Amazon CloudWatch](#).

Benutzerdefinierte AMI-ID

Sie können das für Ihre Amazon EC2 Instances verwendete Standard-AMI mit einem eigenen, benutzerdefinierten AMI überschreiben. Geben Sie dazu den Bezeichner Ihres benutzerdefinierten AMI auf der Registerkarte Configuration (Konfiguration) im Abschnitt Server in das Feld Custom AMI ID für Ihre Umgebung im AWS Toolkit for Eclipse ein.

Important

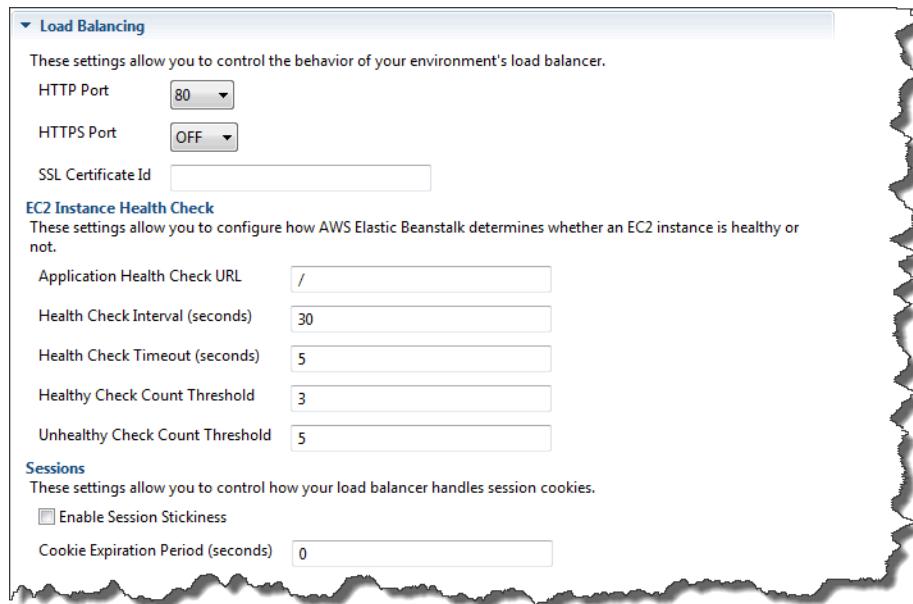
Die Verwendung eines eigenen AMI ist eine Aufgabe für Fortgeschrittene und sollte mit Bedacht ausgeführt werden. Wenn Sie ein benutzerdefiniertes AMI benötigen, empfehlen wir, das standardmäßige Elastic Beanstalk-AMI als Ausgangspunkt zu nehmen und es zu ändern. Für einen ordnungsgemäßen Zustand erwartet Elastic Beanstalk, dass die Amazon EC2 Instances eine Reihe von Anforderungen erfüllen, einschließlich Vorhandensein eines laufenden Hostmanagers. Werden diese Anforderungen nicht erfüllt, funktioniert Ihre Umgebung möglicherweise nicht ordnungsgemäß.

Elastic Load Balancing mit dem AWS Toolkit for Eclipse konfigurieren

Elastic Load Balancing ist ein Amazon-Wbservice, der die Verfügbarkeit und Skalierbarkeit Ihrer Anwendung verbessert. Mit Elastic Load Balancing können Sie Anwendungslasten zwischen zwei oder mehr Amazon EC2-Instances verteilen. Elastic Load Balancing verbessert die Verfügbarkeit durch Redundanz und unterstützt wachsenden Datenverkehr für Ihre Anwendung.

Mit Elastic Load Balancer können Sie den eingehenden Datenverkehr der Anwendung automatisch auf sämtliche EC2-Server-Instances verteilen, die Sie ausführen. Mit dem Service ist es auch ganz einfach, neue Instances hinzuzufügen, wenn Sie die Kapazität Ihrer Anwendung erhöhen.

Elastic Beanstalk stellt Elastic Load Balancing automatisch bereit, wenn Sie eine Anwendung bereitstellen. Unter Load-Balancing auf der Registerkarte Configuration (Konfiguration) für Ihre Umgebung innerhalb des Toolkits for Eclipse können Sie die Load-Balancing-Konfiguration der Elastic Beanstalk-Umgebung bearbeiten.



In den folgenden Abschnitten werden die Elastic Load Balancing-Parameter beschrieben, die Sie für Ihre Anwendung konfigurieren können.

Ports

Der Load Balancer, der für die Verarbeitung von Anforderungen für Ihre Elastic Beanstalk-Anwendung bereitgestellt wurde, sendet Anforderungen an Amazon EC2-Instances, auf denen Ihre Anwendung ausgeführt wird. Der bereitgestellte Load Balancer kann auf Anforderungen über HTTP- und HTTPS-Ports warten und diese an Amazon EC2-Instances in Ihrer AWS Elastic Beanstalk-Anwendung weiterleiten. Standardmäßig verarbeitet der Load Balancer Anfragen auf dem HTTP-Port. Mindestens einer der Ports (HTTP oder HTTPS) muss aktiviert werden.



Important

Stellen Sie sicher, dass der von Ihnen angegebene Port nicht gesperrt ist. Andernfalls können die Benutzer keine Verbindung zu Ihrer Elastic Beanstalk-Anwendung herstellen.

Steuern des HTTP-Ports

Um den HTTP-Port auszuschalten, wählen Sie OFF für HTTP Listener Port. Zum Aktivieren des HTTP-Ports wählen Sie einen HTTP-Port (z. B. 80) aus.

Note

Für den Zugriff auf Ihre Umgebung mit einem anderen als dem Standardport 80, z. B. Port 8080, fügen Sie dem vorhandenen Load Balancer einen Listener hinzu und konfigurieren Sie den neuen Listener für die Überwachung des entsprechenden Ports.

Geben Sie beispielsweise unter Verwendung der [AWS-CLI für Classic Load Balancers](#) den folgenden Befehl ein, wobei Sie `LOAD_BALANCER_NAME` durch den Namen des Load Balancers für Elastic Beanstalk ersetzen.

```
aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
--listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
InstancePort=80"
```

Geben Sie beispielsweise unter Verwendung der [AWS-CLI für Application Load Balancers](#) den folgenden Befehl ein, wobei Sie `LOAD_BALANCER_ARN` durch den ARN des Load Balancers für Elastic Beanstalk ersetzen.

```
aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP --
port 8080
```

Wenn Elastic Beanstalk Ihre Umgebung überwachen soll, entfernen Sie den Listener für Port 80 nicht.

Steuern des HTTPS-Ports

Elastic Load Balancing unterstützt das HTTPS-/TLS-Protokoll, um die Verschlüsselung des Datenverkehrs für Client-Verbindungen zum Load Balancer zu aktivieren. Verbindungen vom Load Balancer mit EC2-Instances erfolgen in Klartext. Standardmäßig ist der HTTPS-Port deaktiviert.

So aktivieren Sie den HTTPS-Port

1. Erstellen Sie ein neues Zertifikat mit AWS Certificate Manager (ACM) oder laden Sie ein Zertifikat und einen Schlüssel in AWS Identity and Access Management (IAM) hoch. Weitere Informationen zum Anfordern eines ACM-Zertifikats finden Sie unter [Zertifikat anfordern](#) im AWS Certificate Manager-Benutzerhandbuch. Weitere Informationen zum Importieren von Zertifikaten von Drittanbietern in

AWS Certificate Manager finden Sie unter [Importieren von Zertifikaten](#) im Leitfaden AWS Certificate Manager-Benutzerhandbuch. Wenn ACM nicht in Ihrer AWS Region verfügbar ist, verwenden Sie AWS Identity and Access Management (IAM), um ein Zertifikat eines Drittanbieters hochzuladen. Die AWS Certificate Manager- und IAM-Services speichern das Zertifikat und stellen einen Amazon-Ressourcennamen (ARN) für das SSL-Zertifikat bereit. Weitere Informationen zum Erstellen und Hochladen von Zertifikaten in IAM finden Sie unter [Arbeiten mit Serverzertifikaten](#) im IAM-Benutzerhandbuch.

2. Geben Sie den HTTPS-Port an, indem Sie einen Port aus der Dropdown-Liste HTTPS Listener Port auswählen.
3. Geben Sie im Textfeld SSL Certificate ID den Amazon-Ressourcennamen (ARN) Ihres SSL-Zertifikats ein. Zum Beispiel `arn:aws:iam::123456789012:server-certificate/abc/certs/build` oder `arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`. Verwenden Sie das SSL-Zertifikat, das Sie in Schritt 1 erstellt und hochgeladen haben.

Um den HTTPS-Port auszuschalten, wählen Sie OFF (AUS) für HTTPS Listener Port.

Zustandsprüfungen

Sie können die Einstellungen für die Zustandsprüfung im Abschnitt EC2 Instance Health Check (Zustandsprüfung für EC2-Instances) des Bereichs Load Balancing steuern.

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

In der folgenden Liste werden die Zustandsprüfungs-Parameter beschrieben, die Sie für Ihre Anwendung festlegen können.

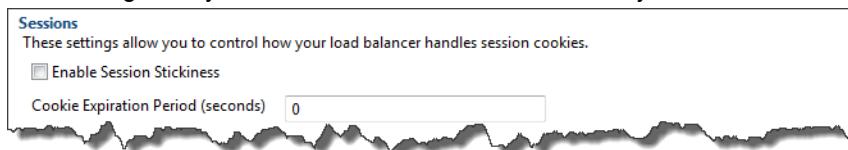
- Um den Instance-Zustand zu ermitteln, sucht Elastic Beanstalk nach einem Antwortcode 200 auf einer URL, die angefordert wird. Standardmäßig überprüft Elastic Beanstalk TCP:80 für Nonlegacy-Container und HTTP:80 für Legacy-Container. Sie können sie überschreiben, sodass sie mit einer vorhandenen Ressource der Anwendung übereinstimmt (z. B. `/myapp/index.jsp`). Dazu geben Sie diese im Feld Application Health Check URL (URL für Anwendungszustandsprüfung) ein. Wenn Sie die Standard-URL überschreiben, verwendet Elastic Beanstalk HTTP zur Abfrage der Ressource. Wenn Sie überprüfen möchten, ob Sie einen Legacy-Container verwenden, erhalten Sie Informationen unter [the section called "Warum sind einige Plattformversionen als Legacy gekennzeichnet?" \(p. 502\)](#).
- Geben Sie für Health Check Interval (seconds) die Anzahl der Sekunden zwischen den Amazon EC2-Instance-Zustandsprüfungen Ihrer Anwendung ein.
- Für Health Check Timeout geben Sie die Anzahl der Sekunden ein, die Elastic Load Balancing auf eine Antwort warten soll, bevor eine Instance als nicht reagierend eingestuft wird.
- In den Feldern Healthy Check Count Threshold und Unhealthy Check Count Threshold geben Sie die Anzahl der aufeinanderfolgenden erfolgreichen oder nicht erfolgreichen URL-Proben ein, bevor Elastic Load Balancing den Instance-Status ändert. Beispiel: Die Angabe von 5 im Textfeld Unhealthy Check Count Threshold bedeutet, dass die URL fünfmal hintereinander eine Fehlermeldung oder eine Zeitüberschreitung zurückgeben müssten, bevor Elastic Load Balancing die Zustandsprüfung als fehlgeschlagen einstufen würde.

Sitzungen

Standardmäßig leitet ein Load Balancer jede Anforderung getrennt an die Server-Instance mit der kleinsten Last weiter. Im Vergleich dazu bindet eine Sticky Session eine Benutzersitzung an eine bestimmte Server-Instance, sodass alle vom Benutzer während der Sitzung gesendeten Anforderungen an die gleiche Server-Instance übermittelt werden.

Elastic Beanstalk nutzt von Load Balancer generierte HTTP-Cookies, wenn Sticky Sessions für eine Anwendung aktiviert sind. Der Load Balancer verwendet ein vom Load-Balancer generiertes spezielles Cookie, um die Anwendungs-Instance für jede Anforderung nachzuverfolgen. Wenn der Load Balancer eine Anforderung empfängt, prüft er zunächst, ob dieses Cookie in der Anforderung vorhanden ist. Wenn ja, wird die Anforderung an die im Cookie angegebene Anwendungs-Instance gesendet. Wenn kein Cookie vorhanden ist, wählt der Load Balancer eine Anwendungs-Instance basierend auf dem vorhandenen Load Balancing-Algorithmus aus. Ein Cookie wird in die Antwort eingefügt, um nachfolgende Anforderungen von demselben Benutzer an diese Anwendungs-Instance zu binden. Mit der Konfiguration der Richtlinie wird ein Cookie-Ablauf definiert, der die Dauer der Gültigkeit für jedes Cookie festlegt.

Sie können über den Abschnitt Sessions unter Load Balancer angeben, ob der Load Balancer für Ihre Anwendung "Sticky Sessions" erlaubt. Auch die Dauer für jedes Cookie wird angegeben.



Weitere Informationen zu Elastic Load Balancing finden Sie im [Elastic Load Balancing-Entwicklerhandbuch](#).

Auto Scaling mit dem AWS Toolkit for Eclipse konfigurieren

Amazon EC2 Auto Scaling ist ein Amazon-Webservice, mit dem Amazon EC2-Instances basierend auf benutzerdefinierten Auslösern automatisch gestartet oder beendet werden. Benutzer können Auto Scaling-Gruppen einrichten und diesen Gruppen Auslöser zuweisen, um Datenverarbeitungsressourcen automatisch basierend auf Metriken, wie Bandbreitennutzung oder CPU-Auslastung, zu skalieren. Amazon EC2 Auto Scaling ruft mithilfe von Amazon CloudWatch Metriken für die Server-Instances ab, auf denen die Anwendung ausgeführt wird.

Mit Amazon EC2 Auto Scaling können Sie eine Gruppe von Amazon EC2-Instances nehmen und verschiedene Parameter festlegen, damit diese Gruppe automatisch größer oder kleiner wird. Mit Amazon EC2 Auto Scaling können Amazon EC2-Instances zur Gruppe hinzugefügt oder daraus entfernt werden, damit sie nahtlos auf Datenverkehränderungen an Ihrer Anwendung eingehen können.

Des Weiteren wird mit Amazon EC2 Auto Scaling auch der Zustand der einzelnen gestarteten Amazon EC2-Instances überwacht. Falls eine Instance unerwartet beendet wird, erkennt Amazon EC2 Auto Scaling dies und startet eine andere Instance als Ersatz. Mit dieser Funktion wird automatisch stets die von Ihnen festgelegte Anzahl an Amazon EC2-Instances ausgeführt.

Elastic Beanstalk bietet Amazon EC2 Auto Scaling für Ihre Anwendung. Unter Auto Scaling auf der Registerkarte Configuration (Konfiguration) Ihrer Umgebung innerhalb des Toolkits for Eclipse können Sie die Konfiguration für die automatische Skalierung der Elastic Beanstalk-Umgebung bearbeiten.

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	1
Maximum Instance Count	4
Availability Zones	Any 1
Scaling Cooldown Time (seconds)	360

Scaling Trigger

Trigger Measurement	NetworkOut
Trigger Statistic	Average
Unit of Measurement	Bytes
Measurement Period (seconds)	5
Breach Duration (seconds)	5
Upper Threshold	600000
Scale-up Increment	1
Lower Threshold	200000
Scale-down Increment	-1

In den folgenden Abschnitten wird beschrieben, wie Sie Auto Scaling-Parameter für die Anwendung konfigurieren.

Startkonfiguration

Sie können die Startkonfiguration bearbeiten, um zu steuern, wie Ihre Elastic Beanstalk-Anwendung Amazon EC2 Auto Scaling-Ressourcen bereitstellt.

In den Einstellungen Minimum Instance Count und Maximum Instance Count geben Sie die minimale sowie die maximale Größe der Auto Scaling-Gruppe an, die von der Elastic Beanstalk-Anwendung genutzt wird.

Minimum Instance Count	1
Maximum Instance Count	4
Availability Zones	Any 1
Scaling Cooldown Time (seconds)	360

Note

Um die Anzahl der Amazon EC2-Instances als Festwert festzulegen, geben Sie in den Textfeldern Minimum Instance Count und Maximum Instance Count denselben Wert ein.

Für Availability Zones geben Sie die Anzahl der Availability Zones an, in denen sich Ihre Amazon EC2-Instances befinden sollen. Es ist wichtig, diese Zahl festzulegen, wenn Sie fehlertolerante Anwendungen erstellen möchten: Wenn eine Availability Zone ausfällt, werden Ihre Instances weiterhin in Ihren anderen Availability Zones ausgeführt.

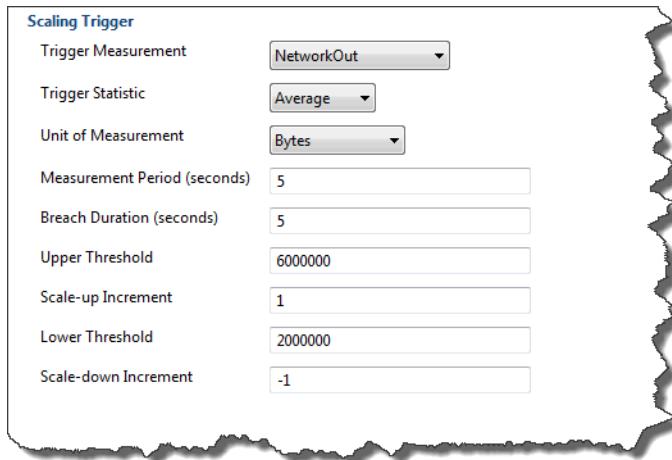
Note

Derzeit ist es nicht möglich anzugeben, in welcher Availability Zone sich Ihre Instance befinden wird.

Auslöser

Bei einem Auslöser handelt es sich um einen Amazon EC2 Auto Scaling-Mechanismus, mit dem Sie definieren, wann die Instance-Anzahl erhöht (hochskaliert) bzw. verringert (herunterskaliert) werden soll. Sie können die Auslöser so konfigurieren, dass Sie bei allen Metriken, die in Amazon CloudWatch veröffentlicht werden, ausgelöst werden, wie z. B. die CPU-Auslastung. Zudem wird ermittelt, ob die vorgegebenen Bedingungen erfüllt sind. Wird der obere oder untere Schwellenwert der Metrik für einen bestimmten Zeitraum überschritten, startet der Auslöser den anhaltenden Prozess der Skalierung.

Sie können mit dem AWS Toolkit for Eclipse einen Skalierungsauslöser für Ihre Elastic Beanstalk-Anwendung definieren.



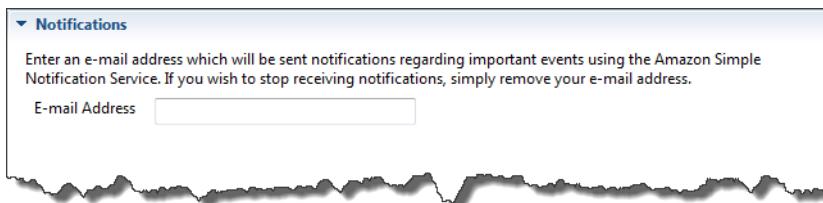
Sie können die folgende Liste an Auslöseparametern im Bereich Scaling Trigger der Registerkarte Configuration für Ihre Umgebung innerhalb des Toolkit for Eclipse konfigurieren.

- Geben Sie für Trigger Measurement die Metrik für den Auslöser an.
- Für Trigger Statistic (Auslöserstatistik) geben Sie an, welche Statistik der Auslöser verwenden wird – **Minimum, Maximum, Sum oder Average**.
- Für Unit of Measurement geben Sie die Einheiten für den Auslöser an.
- Mit Measurement Period geben Sie an, wie häufig die Metriken für den Auslöser von Amazon CloudWatch gemessen werden. Für Breach Duration wird der Zeitraum angegeben, für den der Grenzwert einer Metrik überschritten werden kann (wie unter Upper Threshold und Lower Threshold definiert), bevor der Auslöser ausgelöst wird.
- Für Scale-up Increment und Scale-down Increment geben Sie an, wie viele Amazon EC2-Instances im Rahmen der Skalierung hinzugefügt oder entfernt werden sollen.

Weitere Informationen zur Amazon EC2 Auto Scaling finden Sie im Abschnitt Amazon EC2 Auto Scaling der [Amazon Elastic Compute Cloud-Dokumentation](#).

Konfigurieren von Benachrichtigungen mit dem AWS Toolkit for Eclipse

Elastic Beanstalk verwendet den Amazon Simple Notification Service (Amazon SNS) für Benachrichtigungen zu wichtigen Ereignissen in Ihrer Anwendung. Zur Aktivierung von Amazon SNS-Benachrichtigungen geben Sie einfach Ihre E-Mail-Adresse in das Textfeld Email Address unter Notifications auf der Registerkarte Configuration für Ihre Umgebung innerhalb des Toolkit for Eclipse ein. Um Amazon SNS-Benachrichtigungen zu deaktivieren, entfernen Sie Ihre E-Mail-Adresse aus dem Textfeld.



Java-Container mit AWS Toolkit for Eclipse konfigurieren

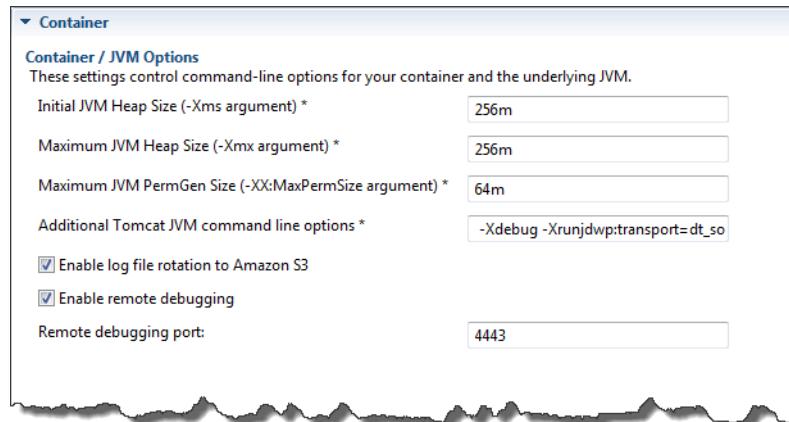
Über das Feld Container/JVM Options können Sie das Verhalten der Java Virtual Machine auf Ihren Amazon EC2-Instances optimieren und die Amazon S3-Protokollrotation aktivieren oder deaktivieren. Sie können das AWS Toolkit for Eclipse verwenden, um Ihre Container-Informationen zu konfigurieren. Weitere Informationen zu den für Tomcat-Umgebungen verfügbaren Optionen finden Sie unter [the section called "Konfigurieren Ihrer Tomcat-Umgebung" \(p. 118\)](#).

Note

Sie können Ihre Konfigurationseinstellungen ändern, ohne eine Ausfallzeit in Kauf nehmen zu müssen, indem Sie den CNAME für Ihre Umgebungen austauschen. Weitere Informationen finden Sie unter [Blau/Grün-Bereitstellungen mit AWS Elastic Beanstalk \(p. 481\)](#).

So greifen Sie auf den Container/JVM-Optionsbereich für Ihre Elastic Beanstalk-Anwendung zu:

1. Wenn Eclipse die Ansicht AWS Explorer nicht anzeigt, wählen Sie im Menü Window (Fenster), Show View (Ansicht anzeigen), AWS Explorer aus. Erweitern Sie den Elastic Beanstalk-Knoten und Ihren Anwendungsknoten.
2. Klicken Sie im AWS Explorer doppelt auf Ihre Elastic Beanstalk Umgebung.
3. Klicken Sie unten im Fenster auf die Registerkarte Configuration (Konfiguration).
4. Konfigurieren Sie im Bereich Container die Containeroptionen.



Remote-Debugging

Wenn Sie Ihre Anwendung remote testen möchten, können Sie sie im Debug-Modus ausführen.

So aktivieren Sie Remote-Debugging

1. Wählen Sie Enable remote debugging aus.
2. Geben Sie für Remote debugging port die Portnummer für die Nutzung von Remote-Debugging an.

Die Einstellung Additional Tomcat JVM command line options wird automatisch ausgefüllt.

So starten Sie das Remote-Debugging

1. Wählen Sie im AWS Toolkit for Eclipse-Menü Windows (Fenster), Show View (Ansicht anzeigen), Other (Andere) aus.
2. Erweitern Sie den Ordner Server und wählen Sie dann Servers aus. Klicken Sie auf OK.
3. Klicken Sie im Bereich Servers mit der rechten Maustaste auf den Server, auf dem Ihre Anwendung ausgeführt wird, und klicken Sie dann auf Restart in Debug.

Festlegen von Systemeigenschaften mit AWS Toolkit for Eclipse

Im folgenden Beispiel wird die `JDBC_CONNECTION_STRING`-Systemeigenschaft im AWS Toolkit for Eclipse festgelegt. Nachdem Sie diese Eigenschaften festgelegt haben, werden sie als Systemeigenschaften mit dem Namen `JDBC_CONNECTION_STRING` für die Elastic Beanstalk-Anwendung verfügbar.

Note

Das AWS Toolkit for Eclipse unterstützt noch nicht das Ändern von Umgebungskonfigurationen, einschließlich Systemeigenschaften, für Umgebungen in einer VPC. Sie müssen die AWS-Managementkonsole (im nächsten Abschnitt beschrieben) oder die [EB CLI \(p. 1008\)](#) verwenden, es sei denn, sie haben ein älteres Konto mit EC2 Classic.

Note

Umgebungskonfigurationseinstellungen können alle druckbaren ASCII-Zeichen außer dem Gravis (`, ASCII 96) enthalten und max. 200 Zeichen umfassen.

So legen Sie die Systemeigenschaften für Ihre Elastic Beanstalk-Anwendung fest:

1. Wenn Eclipse die Ansicht AWS Explorer nicht anzeigt, wählen Sie Window, Show View, Other. Erweitern Sie AWS Toolkit und wählen Sie dann AWS Explorer aus.
2. Erweitern Sie im Bereich AWS-Explorer Elastic Beanstalk, erweitern Sie den Knoten für Ihre Anwendung und klicken Sie dann doppelt auf Ihre Elastic Beanstalk-Umgebung.
3. Klicken Sie unten im Fenster für Ihre Umgebung auf die Registerkarte Advanced (Erweitert).
4. Klicken Sie unter `aws:elasticbeanstalk:application:environment` auf `JDBC_CONNECTION_STRING` und geben Sie dann eine Verbindungszeichenfolge ein. Die folgende JDBC-Verbindungszeichenfolge würde beispielsweise eine Verbindung zu einer MySQL-Datenbank-Instance auf Port 3306 des "localhost" herstellen, mit dem Benutzernamen `me` und dem Passwort `mypassword`:

`jdbc:mysql://localhost:3306/mydatabase?user=me&password=mypassword`

Diese ist für Ihre Elastic Beanstalk-Anwendung als eine Systemeigenschaft namens `JDBC_CONNECTION_STRING` verfügbar.

5. Drücken Sie Strg+C auf der Tastatur oder wählen Sie File (Datei), Save (Speichern) zum Speichern der Änderungen an der Umgebungskonfiguration. Änderungen werden in etwa einer Minute wirksam.

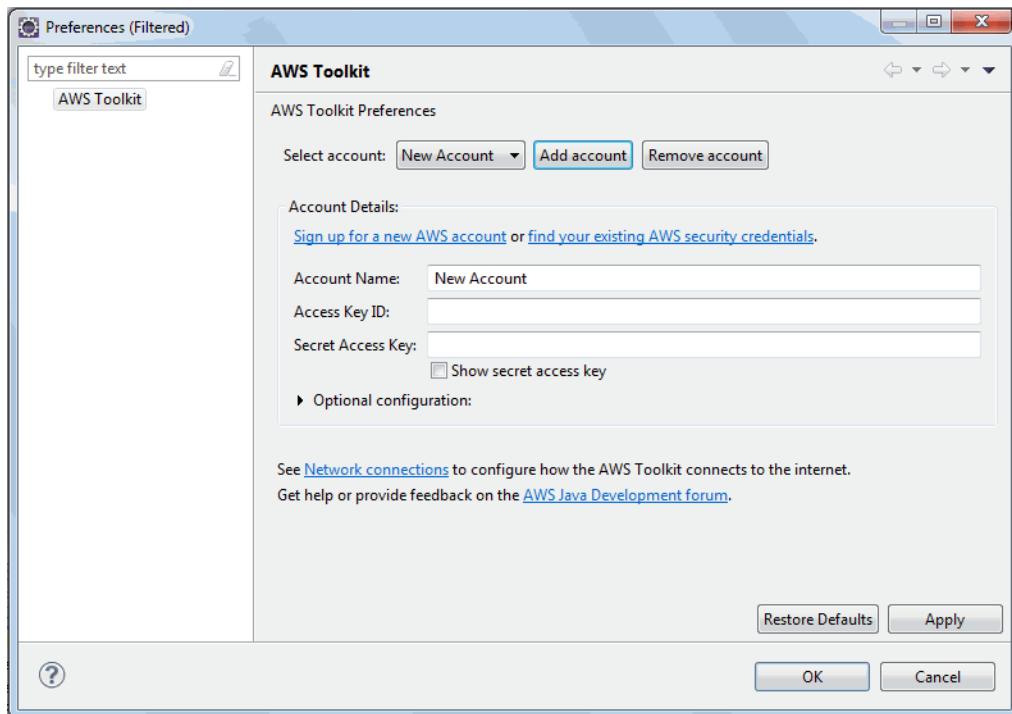
Verwalten mehrerer AWS-Konten

Sie können verschiedene AWS-Konten einrichten, um unterschiedliche Aufgaben auszuführen, wie z. B. Tests, Staging und Produktion. Sie können das AWS Toolkit for Eclipse zum einfachen Hinzufügen, Bearbeiten und Löschen von Konten verwenden.

So fügen Sie ein AWS-Konto mit dem AWS Toolkit for Eclipse hinzu

1. Stellen Sie in Eclipse sicher, dass die Symbolleiste angezeigt wird. Klicken Sie auf der Symbolleiste auf den Pfeil neben dem AWS-Symbol und wählen Sie Preferences.

2. Klicken Sie auf Add account.



3. Geben Sie im Feld Account Name den Anzeigenamen des Kontos ein.
4. Geben Sie im Textfeld Access Key ID die AWS-Zugriffsschlüssel-ID ein.
5. Geben Sie im Textfeld Secret Access Key den AWS-Geheimschlüssel ein.

Für den API-Zugriff benötigen Sie eine Zugriffsschlüssel-ID und einen geheimen Zugriffsschlüssel. Verwenden Sie IAM-Benutzer-Zugriffsschlüssel anstelle von AWS-Account-Root-Benutzer-Zugriffsschlüsseln. Weitere Information über IAM-Zugriffsschlüssel finden Sie unter [Verwalten von Zugriffsschlüsseln für IAM-Benutzer](#) im IAM-Benutzerhandbuch.

6. Klicken Sie auf OK.

So verwenden Sie ein anderes Konto zur Bereitstellung einer Anwendung in Elastic Beanstalk

1. Klicken Sie in der Eclipse-Symbolleiste auf den Pfeil neben dem AWS-Symbol und wählen Sie Preferences.
2. Wählen Sie für Default Account das Konto, das Sie verwenden möchten, um Anwendungen in Elastic Beanstalk bereitzustellen.
3. Klicken Sie auf OK.
4. Klicken Sie im Bereich Project Explorer mit der rechten Maustaste auf die Anwendung, die Sie bereitstellen möchten, und wählen Sie dann Amazon Web Services > Deploy to Elastic Beanstalk aus.

Anzeigen von Ereignissen

Verwenden Sie das AWS Toolkit for Eclipse für den Zugriff auf Ereignisse und Benachrichtigungen im Zusammenhang mit Ihrer Anwendung.

So zeigen Sie Anwendungsereignisse an

1. Wenn Eclipse die Ansicht AWS Explorer nicht anzeigt, klicken Sie im Menü auf Window > Show View > AWS Explorer. Erweitern Sie den Elastic Beanstalk-Knoten und Ihren Anwendungsknoten.

2. Klicken Sie im AWS-Explorer doppelt auf Ihre Elastic Beanstalk-Umgebung.
3. Klicken Sie unten im Fenster auf die Registerkarte Events.

Eine Liste der Ereignisse für alle Umgebungen für Ihre Anwendung wird angezeigt.

The screenshot shows the 'Event Log' view in the AWS Toolkit for Eclipse. The title bar says 'Event Log'. Below it, a section titled 'Environment Events' contains the message: 'Events recorded to your Elastic Beanstalk environment'. A table lists various deployment events:

Message	Version	Date
i Successfully launched environment: myjavawebappenv	First Release	Tue Jul 12 12:18:41 PDT 2011
i Application available at myjavawebappenv.elasticbeanstalk.com.	First Release	Tue Jul 12 12:18:39 PDT 2011
i Adding Instance 'i-0d5eab6c' to your environment.	First Release	Tue Jul 12 12:18:17 PDT 2011
i Added EC2 instance 'i-0d5eab6c' to Auto Scaling Group 'awseb-myjavawebappenv-z9PeEx9Form'.	First Release	Tue Jul 12 12:17:08 PDT 2011
i 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
i Waiting for an EC2 instance to be launched...	First Release	Tue Jul 12 12:16:38 PDT 2011
i Added Elastic Load Balancer 'awseb-myjavawebappenv' to your environment.	First Release	Tue Jul 12 12:16:37 PDT 2011
i Added URLCheck healthcheck for 'http://myjavawebappenv.elasticbeanstalk.com:80/'	First Release	Tue Jul 12 12:16:37 PDT 2011
i Adding Auto Scaling Group 'awseb-myjavawebappenv-z9PeEx9Form' to your environment.	First Release	Tue Jul 12 12:16:35 PDT 2011
i Created Auto Scaling trigger named: awseb-myjavawebappenv-z9PeEx9Form.	First Release	Tue Jul 12 12:16:35 PDT 2011
i Created Auto Scaling group named: awseb-myjavawebappenv-z9PeEx9Form.	First Release	Tue Jul 12 12:16:34 PDT 2011
i Created Auto Scaling launch configuration named: awseb-myjavawebappenv-DKMTv8YQsX.	First Release	Tue Jul 12 12:16:32 PDT 2011
i Created load balancer named: awseb-myjavawebappenv.	First Release	Tue Jul 12 12:16:31 PDT 2011
i Using elasticbeanstalk-us-east-1-049020475370 as Amazon S3 storage bucket for environment data.	First Release	Tue Jul 12 12:16:29 PDT 2011
i createEnvironment is starting.	First Release	Tue Jul 12 12:16:27 PDT 2011

Auflisten von Server-Instances/Verbinden mit Server-Instances

Sie können im AWS Toolkit for Eclipse oder in der AWS-Managementkonsole eine Liste der Amazon EC2-Instances anzeigen, auf denen die Elastic Beanstalk-Anwendungsumgebung ausgeführt wird. Eine Verbindung zu diesen Instances stellen Sie mit Secure Shell (SSH) her. Weitere Informationen zum Auflisten von Server-Instances und zum Herstellen einer Verbindung zu diesen mit der AWS Management Console finden Sie unter [Auflisten von Server-Instances/Verbinden mit Server-Instances \(p. 866\)](#). Der folgende Abschnitt führt Sie durch das Anzeigen und Herstellen einer Verbindung mit Ihren Server-Instances mit dem AWS Toolkit for Eclipse.

So zeigen Sie Amazon EC2-Instances für eine Umgebung an und stellen eine Verbindung zu diesen her

1. Klicken Sie im AWS Toolkit for Eclipse auf AWS Explorer. Erweitern Sie den Amazon EC2-Knoten und doppelklicken Sie dann auf Instances.
2. Klicken Sie im Fenster mit den Amazon EC2-Instances in der Spalte Instance ID mit der rechten Maustaste auf die Instance ID für die Amazon EC2-Instance, die im Load Balancer Ihrer Anwendung ausgeführt wird. Klicken Sie dann auf Open Shell.

The screenshot shows the 'Amazon EC2 Instances' view in the AWS Toolkit for Eclipse. The title bar says 'Amazon EC2 Instances'. Below it, a message says 'These instances make up the fleet in your environment'. A table lists instances:

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	ami-00000000	standard	running	t1.micro	us-east-1d	new key pair	Jul 12, 2011 2:09:31 PM	elasticbeanstalk-default	

A context menu is open over the first instance. The 'Open Shell' option is highlighted. Other options include Refresh, Reboot, Terminate, Start instances, Stop instances, Open Shell As..., Bundle AMI..., Copy Public DNS Name, and Elastic Block Storage.

Eclipse öffnet den SSH-Client automatisch und stellt die Verbindung mit der EC2-Instance her.

Weitere Informationen zum Herstellen einer Verbindung zu einer Amazon EC2-Instance finden Sie im [Amazon Elastic Compute Cloud Getting Started Guide](#).

Beenden einer Umgebung

Beenden Sie eine ausgeführte Umgebung mit dem AWS Toolkit for Eclipse, damit keine Kosten für ungenutzte AWS-Ressourcen anfallen. Weitere Informationen zum Beenden einer Umgebung finden Sie unter [Elastic Beanstalk-Umgebung terminieren \(p. 459\)](#).

So beenden Sie eine Umgebung

1. Klicken Sie im AWS Toolkit for Eclipse auf den Bereich AWS Explorer. Erweitern Sie den Knoten Elastic Beanstalk.
2. Erweitern Sie die Elastic Beanstalk-Anwendung und klicken Sie mit der rechten Maustaste auf die Elastic Beanstalk-Umgebung.
3. Wählen Sie Terminate Environment (Umgebung beenden). Es dauert einige Minuten, bis Elastic Beanstalk die AWS-Ressourcen beendet hat, die in der Umgebung ausgeführt werden.

Ressourcen

Es gibt zahlreiche Möglichkeiten, um zusätzliche Unterstützung bei der Entwicklung von Java-Anwendungen zu erhalten.

Ressource	Beschreibung
The AWS Java Development Forum	Hier können Sie Fragen stellen und Feedback erhalten.
Java Developer Center	Umfangreiches Kit mit Beispiel-Code, Dokumentation, Tools und zusätzlichen Ressourcen.

Arbeiten mit .NET Core unter Linux

Dieser Abschnitt enthält Informationen zur Bereitstellung von .NET Core-Anwendungen mit AWS Elastic Beanstalk unter Linux.

In den Themen in diesem Kapitel wird davon ausgegangen, dass Sie über einige Kenntnisse der Elastic Beanstalk-Umgebungen verfügen. Wenn Sie Elastic Beanstalk noch nicht verwendet haben, machen Sie sich im [Tutorial „Erste Schritte“ \(p. 3\)](#) mit den Grundlagen vertraut.

Themen

- [Erste Schritte mit .NET Core unter Linux \(p. 156\)](#)
- [Einrichten der .NET Core unter Linux-Entwicklungsumgebung \(p. 158\)](#)
- [Verwenden der .NET Core unter Linux-Plattform \(p. 159\)](#)
- [Tutorial: Bereitstellen einer ASP.NET Core-Anwendung mit Elastic Beanstalk unter Linux \(p. 163\)](#)
- [Das AWS Toolkit for Visual Studio – Arbeiten mit .Net Core \(p. 169\)](#)
- [Migration von .NET unter Windows Server-Plattformen zur .NET Core unter Linux-Plattform \(p. 188\)](#)

Erste Schritte mit .NET Core unter Linux

Für die ersten Schritte mit .NET Core unter Linux-Anwendungen in AWS Elastic Beanstalk benötigen Sie lediglich ein [Quellpaket \(p. 412\)](#) einer Anwendung, das Sie als Ihre erste Anwendungsversion hochladen und anschließend in einer Umgebung bereitstellen können. Im Rahmen der Umgebungserstellung werden von Elastic Beanstalk alle erforderlichen AWS-Ressourcen für eine hoch skalierbare Webanwendung zugewiesen.

Starten einer Umgebung mit einer .NET Core unter Linux-Beispielanwendung

Elastic Beanstalk stellt für jede Plattform Beispielanwendungen bereit, die eine Seite enthalten.

Beispiele

Unterstützte Konfigurationen	Umgebungs	Quell-Bundle	Beschreibung
.NET Core auf Linux	Webserver	dotnet-core-linux.zip	Anwendung, die eine Seite enthält.
.NET Core auf Linux	Webserver	dotnet-core-linux-multiple-apps.zip	Zwei Webanwendungen, die auf demselben Webserver ausgeführt werden.

Laden Sie die Beispielanwendung herunter und stellen Sie diese in Elastic Beanstalk bereit. Führen Sie dazu die folgenden Schritte aus.

So starten Sie eine Umgebung mit einer Beispielanwendung (Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Applications (Anwendungen) und anschließend in der Liste den Namen einer vorhandenen Anwendung aus. Sie können auch [eine Anwendung erstellen \(p. 403\)](#).
3. Wählen Sie auf der Anwendungsübersichtsseite die Option Create a new environment (Neue Umgebung erstellen).

The screenshot shows the AWS Elastic Beanstalk console interface. On the left, there's a sidebar with navigation links: Environments, Applications, and Recent environments. Under Applications, 'getting-started-app' is selected. Below the sidebar, the main content area displays the 'Application 'getting-started-app' environments' page. It includes a search bar labeled 'Filter results matching the display values'. A table lists two environments: 'GettingStartedApp-env' and 'GettingStartedApp-Windows'. The table columns are: Environment name, Health, Date created, Last modified, URL, and Platform. The 'GettingStartedApp-env' row shows 'OK' for Health, '2020-01-28 12:06:50 UTC-0800' for Date created, '2020-01-30 15:02:35 UTC-0800' for Last modified, 'GettingStartedApp-env.bn7dx222kw.us-east-2.elasticbeanstalk.com' for URL, and 'Tomcat running on Linux' for Platform. The 'GettingStartedApp-Windows' row shows 'OK' for Health, '2020-01-28 16:34:29 UTC-0800' for Date created, '2020-01-28 16:38:20 UTC-0800' for Last modified, 'GettingStartedApp-Windows.bn7dx222kw.us-east-2.elasticbeanstalk.com' for URL, and 'IIS 10.0 running on Windows' for Platform.

4. Wählen Sie die Web server environment (Webserverumgebung) oder die Worker environment (Worker-Umgebung) als [Umgebungsebene \(p. 14\)](#) aus. Nach der Erstellung lässt sich die Umgebungsebene nicht mehr ändern.

Note

Die .NET-Plattform auf Windows Server (p. 189) unterstützt die Worker-Umgebungsschicht nicht.

The screenshot shows the 'Select environment tier' step in the AWS Elastic Beanstalk console. At the top, the navigation path is 'Elastic Beanstalk > Applications > getting-started-app'. The main title is 'Select environment tier'. Below it, a descriptive text explains that AWS Elastic Beanstalk has two types of environment tiers: 'Web server environment' and 'Worker environment'. The 'Web server environment' is selected, indicated by a blue circle and bold text. A sub-description states: 'Run a website, web application, or web API that serves HTTP requests.' Below this, there is a 'Learn more' link with a blue icon. The 'Worker environment' option is also shown with its own description: 'Run a worker application that processes long-running workloads on demand or performs tasks on a schedule.' and a 'Learn more' link.

5. Wählen Sie in Platform (Plattform) die Plattform und den Plattformzweig aus, die/der mit der Sprache Ihrer Anwendung übereinstimmt.

Note

Elastic Beanstalk unterstützt mehrere [Versionen \(p. 32\)](#) für die meisten Plattformen, die aufgelistet sind. Standardmäßig wählt die Konsole die empfohlene Version für die gewählte Plattform und Plattformvariante. Wenn Ihre Anwendung eine andere Version erfordert, können Sie diese hier auswählen oder [Configure more options \(Weitere Optionen konfigurieren\)](#) auswählen wie in Schritt 7 beschrieben. Informationen zu unterstützten Plattformversionen finden Sie unter [the section called "Unterstützte Plattformen" \(p. 32\)](#).

6. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
7. Wählen Sie [Configure more options \(Weitere Optionen konfigurieren\)](#) aus, um Ihre Umgebung weiter anzupassen. Sie können die folgenden Optionen nur während des Erstellens der Umgebung festlegen:
 - Environment name
 - Domänenname
 - Plattformversion
 - VPC
 - Stufe

Sie können die folgenden Einstellungen nach der Umgebungserstellung zwar ändern, jedoch müssen dafür neue Instances oder andere Ressourcen bereitgestellt werden, deren Anwendung viel Zeit in Anspruch nehmen kann:

- Instance-Typ, Stamm-Volume, Schlüsselpaar und AWS Identity and Access Management (IAM)-Rolle
- Interne Amazon RDS-Datenbank
- Load Balancer

Weitere Informationen zu allen verfügbaren Einstellungen finden Sie unter [Der Assistent zum Erstellen einer neuen Umgebung \(p. 437\)](#).

8. Wählen Sie Create environment (Umgebung erstellen) aus.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine andere Anwendung bereitstellen. Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine Amazon EC2-Instances bereitgestellt oder neu gestartet werden müssen.

Wenn Sie ein oder zwei Beispielanwendungen bereitgestellt haben und bereit sind, .NET Core-Anwendungen zu entwickeln und auszuführen, finden Sie unter [Einrichten der .NET Core unter Linux-Entwicklungsumgebung \(p. 158\)](#) weiterführende Informationen.

Einrichten der .NET Core unter Linux-Entwicklungsumgebung

Richten Sie eine .NET Core-Entwicklungsumgebung ein, um Ihre Anwendung lokal zu testen, bevor Sie diese in AWS Elastic Beanstalk bereitstellen. In diesem Thema finden Sie die Schritte zum Einrichten der Entwicklungsumgebung sowie Links zu den Installationsseiten nützlicher Tools.

Allgemeine Einrichtungsschritte und Tools für alle Sprachen finden Sie unter [Konfigurieren des Entwicklungskomputers für die Verwendung mit Elastic Beanstalk \(p. 1005\)](#).

Abschnitte

- [Installation des .NET Core SDK \(p. 158\)](#)
- [Installieren einer IDE \(p. 159\)](#)
- [Installieren des AWS Toolkit for Visual Studio \(p. 159\)](#)

Installation des .NET Core SDK

Sie können das .NET Core SDK für die Entwicklung von Anwendungen verwenden, die auf Linux ausgeführt werden.

Sie können das .NET Core SDK von der Seite für [.NET-Downloads](#) zur anschließenden Installation herunterladen.

Installieren einer IDE

Integrierte Entwicklungsumgebungen (IDEs) stellen zahlreiche Funktionen zur Vereinfachung der Anwendungsentwicklung bereit. Wenn Sie noch nie eine IDE für die .NET-Entwicklung verwendet haben, können Sie mit Visual Studio Community erste Schritte unternehmen.

Sie können Visual Studio Community von der Seite [Visual Studio Community](#) zur anschließenden Installation herunterladen.

Installieren des AWS Toolkit for Visual Studio

Das [AWS Toolkit for Visual Studio \(p. 220\)](#) ist ein Open-Source-Plug-in für die Visual Studio-IDE, das die Entwicklung, das Debuggen und die Bereitstellung von .NET-Anwendungen mit AWS für Entwickler vereinfacht. Installationsanleitungen finden Sie auf der [Toolkit for Visual Studio-Homepage](#).

Verwenden der .NET Core unter Linux-Plattform

Die AWS Elastic Beanstalk-.NET Core unter Linux-Plattform besteht aus einer Reihe von [Plattformversionen](#) für .NET Core-Anwendungen, die auf dem Linux-Betriebssystem ausgeführt werden.

Details zu den verschiedenen Möglichkeiten für das Erweitern einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#). Im Folgenden werden einige plattformspezifische Aspekte beschreiben, die berücksichtigt werden sollten.

Einführung in die .NET Core unter Linux-Plattform

Proxy-Server

Die Elastic Beanstalk-.NET Core unter Linux-Plattform enthält einen Reverse-Proxy-Server, der Anforderungen an die Anwendung weiterleitet. Standardmäßig verwendet Elastic Beanstalk [nginx](#) als Proxy-Server. Sie können festlegen, dass kein Proxy-Server verwendet wird, und [Kestrel](#) als Webserver Ihrer Anwendung konfigurieren. Kestrel ist standardmäßig in ASP.NET Core-Projektvorlagen enthalten.

Anwendungsstruktur

Sie können laufzeitabhängige Anwendungen veröffentlichen, die die von Elastic Beanstalk bereitgestellte .NET Core-Laufzeitumgebung verwenden. Sie können auch eigenständige Anwendungen veröffentlichen, die die .NET Core-Laufzeit und die Abhängigkeiten Ihrer Anwendung im Quellpaket enthalten. Weitere Informationen hierzu finden Sie unter [the section called “Anwendungsbündelung” \(p. 161\)](#).

Plattformkonfiguration

Um die Prozesse auf den Server-Instances in Ihrer Umgebung ausgeführten Prozesse zu konfigurieren, fügen Sie eine optionale [Procfile \(p. 162\)](#)-Datei in das Quellpaket ein. Eine -Datei ist erforderlich, wenn das Quellpaket mehr als eine Anwendung enthält.

Sie sollten mit Ihrer Anwendung jedoch stets eine [Procfile](#)-Datei im Quellpaket bereitstellen. So können Sie genau steuern, welche Prozesse von Elastic Beanstalk für Ihre Anwendung ausgeführt werden.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das [Ändern der Konfiguration einer ausgeführten Umgebung \(p. 658\)](#) verfügbar. Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie [gespeicherte Konfigurationen \(p. 764\)](#) verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie [Konfigurationsdateien \(p. 722\)](#) einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung

erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, wenn vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe [Konfigurationsoptionen \(p. 646\)](#).

Konfigurieren Ihrer .NET Core unter Linux-Umgebung

Mithilfe der .NET Core unter Linux-Plattformeinstellungen können Sie das Verhalten Ihrer Amazon EC2-Instances optimieren. Sie können die Amazon EC2-Instance-Konfiguration der Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-Konsole bearbeiten.

Mithilfe der Elastic Beanstalk-Konsole aktivieren Sie die Protokollrotation an Amazon S3 und konfigurieren Variablen, die die Anwendung aus der Umgebung auslesen kann.

So konfigurieren Sie Ihre .NET Core unter Linux-Umgebung mit der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren) – Gibt an, ob die Protokolldateien für die Amazon EC2-Instances der Anwendung in den zugeordneten Amazon S3-Bucket kopiert werden sollen.

Umgebungseigenschaften

Im Abschnitt Environment properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Umgebungseigenschaften werden als Schlüssel-Wert-Paare an die Anwendung weitergeleitet.

Innerhalb der .NET Core unter Linux-Umgebung, die in Elastic Beanstalk ausgeführt wird, kann über `Environment.GetEnvironmentVariable("variable-name")` auf Umgebungsvariablen zugegriffen werden. Beispielsweise können Sie mit folgendem Code eine Eigenschaft mit dem Namen `API_ENDPOINT` als Variable lesen.

```
string endpoint = Environment.GetEnvironmentVariable("API_ENDPOINT");
```

Weitere Informationen finden Sie unter [Umgebungseigenschaften und andere Softwareeinstellungen \(p. 621\)](#).

.NET Core unter Linux-Konfigurations-Namespace

Mithilfe einer [Konfigurationsdatei \(p. 722\)](#) können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Konfigurationsoptionen können durch den Elastic Beanstalk-Service oder die verwendete Plattform definiert und in Namespaces organisiert werden.

Die .NET Core unter Linux-Plattform unterstützt zusätzlich zu den [für alle Elastic Beanstalk-Umgebungen unterstützten Optionen \(p. 666\)](#) auch Optionen im folgenden Namespace:

- `aws:elasticbeanstalk:environment:proxy` – Wählen Sie einen nginx-Proxy-Server oder keinen Proxy-Server aus. Gültige Werte sind `nginx` oder `none`.

In der folgenden Beispielkonfigurationsdatei wird die Verwendung der für .NET Core unter Linux spezifischen Konfigurationsoptionen gezeigt.

Example `.ebextensions/proxy-settings.config`

```
option_settings:  
  aws:elasticbeanstalk:environment:proxy:  
    ProxyServer: none
```

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Anwendungsbündelung für die .NET Core unter Linux-Plattform

Sie können sowohl laufzeitabhängige als auch eigenständige .NET Core-Anwendungen auf AWS Elastic Beanstalk ausführen.

Eine laufzeitabhängige Anwendung verwendet eine .NET Core-Laufzeit, die Elastic Beanstalk für die Ausführung Ihrer Anwendung bereitstellt. Elastic Beanstalk verwendet die Datei `runtimeconfig.json` in Ihrem Quellpaket, um die für Ihre Anwendung zu verwendende Laufzeit zu bestimmen. Elastic Beanstalk wählt die neueste kompatible Laufzeit, die für Ihre Anwendung verfügbar ist.

Eine eigenständige Anwendung enthält die .NET Core-Laufzeit, Ihre Anwendung und ihre Abhängigkeiten. Um eine Version der .NET Core-Laufzeit zu verwenden, die nicht von Elastic Beanstalk mit den Plattformen bereitgestellt wird, stellen Sie eine eigenständige Anwendung bereit.

Beispiele

Sie können mit dem Befehl `dotnet publish` sowohl eigenständige als auch laufzeitabhängige Anwendungen kompilieren. Weitere Informationen zum Veröffentlichen von .NET Core-Apps finden Sie unter [Übersicht über die Veröffentlichung von .NET Core-Anwendungen](#) in der .NET Core-Dokumentation.

Die folgende Beispieldateistruktur definiert eine einzelne Anwendung, die eine von Elastic Beanstalk bereitgestellte .NET Core-Laufzeit verwendet.

```
### appsettings.Development.json  
### appsettings.json  
### dotnetcoreapp.deps.json  
### dotnetcoreapp.dll  
### dotnetcoreapp.pdb
```

```
### dotnetcoreapp.runtimeconfig.json
### web.config
### Procfile
### .ebextensions
### .platform
```

Sie können in das Quellpaket mehrere Anwendungen einfügen. Im folgenden Beispiel werden zwei Anwendungen definiert, die auf demselben Webserver ausgeführt werden sollen. Um mehrere Anwendungen auszuführen, müssen Sie eine [Procfile \(p. 162\)](#)-Datei in das Quellpaket einfügen. Eine vollständige Beispianwendung finden Sie unter [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
```

Verwendung einer Procfile-Datei für die Konfiguration Ihrer .NET Core unter Linux-Umgebung

Um mehrere Anwendungen auf demselben Webserver auszuführen, müssen Sie eine `Procfile`-Datei in das Quellpaket einfügen, die Elastic Beanstalk mitteilt, welche Anwendungen ausgeführt werden sollen.

Sie sollten mit Ihrer Anwendung jedoch stets eine `Procfile`-Datei im Quellpaket bereitstellen. Auf diese Weise können Sie genau steuern, welche Prozesse Elastic Beanstalk für Ihre Anwendung ausführt und welche Argumente diese Prozesse erhalten.

Im folgenden Beispiel wird eine `Procfile`-Datei verwendet, um Elastic Beanstalk zwei Anwendungen anzugeben, die auf demselben Webserver ausgeführt werden sollen.

Example Procfile

```
web: dotnet ./dotnet-core-app1/dotnetcoreapp1.dll
web2: dotnet ./dotnet-core-app2/dotnetcoreapp2.dll
```

Einzelheiten zum Schreiben und Verwenden eines `Procfile` finden Sie im Abschnitt Buildfile und `Procfile` unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#).

Konfiguration des Proxy-Servers für Ihre .NET Core unter Linux-Umgebung

AWS Elastic Beanstalk nutzt `nginx` als Reverse-Proxy, um Anfragen an Ihre Anwendung weiterzuleiten. Elastic Beanstalk bietet eine `nginx`-Standardkonfiguration, die Sie entweder erweitern oder gänzlich mit einer eigenen Konfiguration überschreiben können.

Standardmäßig wird der `nginx`-Proxy von Elastic Beanstalk so konfiguriert, dass alle Anforderungen an die Anwendung über Port 5000 weitergeleitet werden. Sie können den Standardport überschreiben, indem Sie die [PORT-Umgebungseigenschaft \(p. 160\)](#) auf den Überwachungsport Ihrer Hauptanwendung festlegen.

Note

Der Überwachungsport Ihrer Anwendung wirkt sich nicht auf den Überwachungsport des `nginx`-Servers aus, über den dieser Anforderungen aus dem Load Balancer erhält.

Alle Amazon Linux 2-Plattformen unterstützen eine einheitliche Proxy-Konfigurationsfunktion. Um Details zur Konfiguration des Proxy-Servers anzuzeigen, erweitern Sie den Abschnitt Reverse-Proxy-Konfiguration in [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#).

Die folgende beispielhafte Konfigurationsdatei erweitert die `nginx`-Konfiguration Ihrer Umgebung. Die Konfiguration leitet Anforderungen für `/api` an eine zweite Webanwendung weiter, die Port 5200 auf dem Webserver überwacht. Standardmäßig leitet Elastic Beanstalk Anforderungen an eine einzelne Anwendung weiter, die den Port 5000 überwacht.

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: Bereitstellen einer ASP.NET Core-Anwendung mit Elastic Beanstalk unter Linux

In diesem Tutorial werden die Schritte zum Erstellen einer neuen ASP.NET Core-Anwendung und deren Bereitstellung in einer Amazon Linux 2-Umgebung mit Elastic Beanstalk beschrieben.

In diesem Tutorial verwenden Sie zunächst das `dotnet`-Befehlszeilen-Tool des .NET Core SDK, um Folgendes zu tun:

- Generieren Sie eine Anwendung, die HTTP-Anforderungen mit ASP.NET bedient.
- Installieren Sie Laufzeitabhängigkeiten.
- Kompilieren und führen Sie Ihre Webanwendung lokal aus.
- Veröffentlichen Sie Ihre Anwendungsartefakte in einem Ausgabeverzeichnis. Die Artefakte umfassen den kompilierten Quellcode, Laufzeitabhängigkeiten und Konfigurationsdateien.

Als Nächstes führen Sie die folgenden Schritte mit Ihrer neu erstellten Anwendung aus:

- Erstellen Sie das Quell-Bundle Ihrer Anwendung, das Ihre veröffentlichten Artefakte enthält.
- Erstellen Sie eine Amazon Linux 2-Umgebung und stellen Sie Ihre Anwendung mit Elastic Beanstalk darin bereit.
- Öffnen Sie die Website-URL, die von Elastic Beanstalk erstellt wurde, um Ihre Anwendung auszuführen.

Der Quellcode der Anwendung ist hier verfügbar: [dotnet-core-linux-tutorial-source.zip](#).

Das bereitstellbare Quell-Bundle finden Sie hier: [dotnet-core-linux-tutorial-bundle](#).

Abschnitte

- [Voraussetzungen \(p. 164\)](#)
- [Generieren eines .NET Core-Projekts als Webanwendung \(p. 165\)](#)
- [Starten einer Elastic Beanstalk-Umgebung und Bereitstellen der Anwendung \(p. 167\)](#)
- [Bereinigen \(p. 168\)](#)
- [Nächste Schritte \(p. 168\)](#)

Voraussetzungen

In diesem Tutorial wird das .NET Core SDK verwendet, um eine einfache .NET Core-Webanwendung zu generieren, lokal auszuführen und ein Bereitstellungspaket zu erstellen.

Voraussetzungen

- .NET-Core (x64) 2.1.19 oder höher

So installieren Sie das .NET Core SDK

1. Laden Sie das Installationsprogramm unter [microsoft.com/net/core](#) herunter. Wählen Sie Ihre Entwicklungsplattform. Wählen Sie Download .NET Core SDK (.NET Core SDK herunterladen) aus.
2. Führen Sie das Installationsprogramm aus und folgen Sie den Anweisungen.

Note

Obwohl die Beispiele in diesem Tutorial Auflistungen aus der Windows-Befehlszeile sind, unterstützt das .NET Core SDK Entwicklungsplattformen auf mehreren Betriebssystemen. Die in diesem Tutorial gezeigten `dotnet`-Befehle sind über verschiedene Entwicklungsplattformen hinweg konsistent.

In diesem Tutorial wird ein ZIP-Befehlszeilenprogramm zum Erstellen eines Quell-Bundles verwendet, das Sie in Elastic Beanstalk bereitstellen können. Um den Befehl `zip` in Windows zu verwenden, können Sie `UnxUtils` installieren. (`UnxUtils` ist eine leichte Sammlung nützlicher Befehlszeilen-Dienstprogramme wie `zip` und `ls`.) Alternativ können Sie den [Windows-Explorer \(p. 414\)](#) oder ein beliebiges anderes ZIP-Programm für die Erstellung von Quell-Bundle-Archiven nutzen.

So installieren Sie UnxUtils

1. Herunterladen [UnxUtils](#).
2. Extrahieren Sie das Archiv in ein lokales Verzeichnis. Beispiel, `C:\Program Files (x86)`
3. Fügen Sie den Pfad zu den binären Dateien zu Ihrer Windows PATH-Benutzervariable hinzu. Beispiel, `C:\Program Files (x86)\UnxUtils\usr\local\wbin`.
 - a. Betätigen Sie die Windows-Taste und geben Sie anschließend `environment variables` ein.

- b. Wählen Sie Edit environment variables for your account (Umgebungsvariablen für Ihr Konto bearbeiten).
 - c. Wählen Sie PATH (PFAD) und Edit (Bearbeiten) aus.
 - d. Fügen Sie im Feld Variable value (Variablenwert) Pfade hinzu, getrennt durch Semikola. Beispiel:
C:\item1\path;C:\item2\path
 - e. Klicken Sie zweimal auf OK, um die neuen Einstellungen anzuwenden.
 - f. Schließen Sie alle laufenden Eingabeaufforderungsfenster und öffnen Sie dann erneut ein Eingabeaufforderungsfenster.
4. Öffnen Sie ein neues Eingabeaufforderungsfenster und führen Sie den `zip`-Befehl aus, um sicherzustellen, dass es funktioniert.

```
> zip -h
Copyright (C) 1990-1999 Info-ZIP
Type 'zip "-L"' for software license.
...
```

Generieren eines .NET Core-Projekts als Webanwendung

Verwenden Sie das Befehlszeilen-Tool `dotnet`, um ein neues C# .NET Core-Webanwendungsprojekt zu erstellen und lokal auszuführen. Die standardmäßige .NET Core-Webanwendung zeigt `Hello World!` an.

So erstellen Sie ein neues .NET Core-Projekt

1. Öffnen Sie ein neues Befehlszeilenfenster und navigieren Sie zu Ihrem Benutzerordner.

```
> cd %USERPROFILE%
```

2. Erstellen Sie mit dem Befehl `dotnet new web` ein neues .NET Core-Projekt.

```
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.
```

So führen Sie die Website lokal aus

1. Verwenden Sie den Befehl `dotnet restore` zum Installieren von Abhängigkeiten.

```
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. Verwenden Sie den `dotnet run` Befehl, um die Anwendung zu erstellen und lokal zu starten.

```
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. Öffnen Sie localhost:5000, um die Website in Ihrem Standard-Webbrowser anzuzeigen.

Die Anwendung gibt `Hello World!` aus, was in Ihrem Webbrowser angezeigt wird.

Um die Anwendung auf einem Webserver auszuführen, muss der kompilierte Quellcode mit Laufzeitabhängigkeiten in einer `web.config`-Konfigurationsdatei gebündelt werden. Das Tool `dotnet` bietet einen `publish`-Befehl, mit dem diese Dateien basierend auf der Konfiguration in `dotnet-core-tutorial.csproj` in einem Verzeichnis gesammelt werden.

So erstellen Sie die Website

- Mit dem Befehl `dotnet publish` geben Sie kompilierten Code und Abhängigkeiten in einem Ordner mit dem Namen `site` aus.

```
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\
```

So erstellen Sie ein Quell-Bundle

- Verwenden Sie den Befehl `zip` zum Erstellen eines Quell-Bundles mit dem Namen `dotnet-core-tutorial.zip`.

Das Quell-Bundle enthält alle Dateien, die im Website-Ordner veröffentlicht wurden.

Note

Wenn Sie ein anderes ZIP-Dienstprogramm verwenden, müssen Sie alle Dateien dem Stammordner des resultierenden ZIP-Archivs hinzufügen. Dies ist für eine erfolgreiche Bereitstellung der Anwendung in Ihrer Elastic Beanstalk-Umgebung erforderlich.

```
C:\users\username\dotnet-core-tutorial> cd site
C:\users\username\dotnet-core-tutorial\site>zip ..\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

In diesem Tutorial führen Sie nur eine Anwendung auf dem Webserver aus, daher ist in Ihrem Quell-Bundle keine Procfile erforderlich. Um jedoch mehrere Anwendungen auf demselben Webserver bereitzustellen zu können, müssen Sie eine Procfile einschließen. Weitere Informationen finden Sie unter [Verwendung einer Procfile-Datei für die Konfiguration Ihrer .NET Core unter Linux-Umgebung \(p. 162\)](#).

Starten einer Elastic Beanstalk-Umgebung und Bereitstellen der Anwendung

Sie verwenden die Elastic Beanstalk-Konsole, um eine Elastic Beanstalk-Umgebung zu starten und das Quell-Bundle bereitzustellen.

Sie können das Quell-Paket hier herunterladen: [dotnet-core-linux-tutorial-bundle](#).

So starten Sie eine Umgebung und stellen Ihren Code bereit (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced.
2. Wählen Sie für Platform (Plattform) die Option .NET Core on Linux (.NET Core unter Linux) aus.
3. Wählen Sie Local file (Lokale Datei), wählen Sie Choose file (Datei wählen) und öffnen Sie das Quell-Bundle.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Prüfen Sie die verfügbaren Einstellungen und wählen Sie Create app (Anwendung erstellen) aus. Die Anwendung schreibt Hello World! als Antwort und gibt diese zurück.

Es dauert etwa 10 Minuten, die Umgebung zu erstellen und Ihren Code bereitzustellen.

Beim Starten einer Umgebung werden die folgenden Ressourcen erstellt:

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2) zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.
- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer) – Ein Elastic Load Balancing-Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn sie beendet wird oder nicht verfügbar ist.

- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – Zwei CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form **subdomain.region.elasticbeanstalk.com** weiterleitet.

Elastic Beanstalk verwaltet all diese Ressourcen. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Note

Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird zwischen den Umgebungen geteilt und nicht gelöscht, wenn Sie die Umgebung beenden. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#).

Bereinigen

Wenn Sie nicht mehr mit Elastic Beanstalk arbeiten, können Sie Ihre Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 531\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) und dann Terminate environment (Umgebung beenden) aus.
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Nächste Schritte

Im weiteren Verlauf der Anwendungsentwicklung möchten Sie vielleicht Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die [Elastic Beanstalk Command Line Interface \(p. 1008\)](#) (EB CLI) stellt benutzerfreundliche Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk-Umgebungen über die Befehlszeilenschnittstelle bereit.

Wenn Sie Visual Studio verwenden, um Ihre Anwendung zu entwickeln, können Sie auch das AWS Toolkit for Visual Studio verwenden, um Änderungen in Ihrem Code bereitzustellen oder Ihre Elastic Beanstalk-

Umgebungen und andere AWS-Ressourcen zu verwalten. Weitere Informationen finden Sie unter [Das AWS Toolkit for Visual Studio \(p. 220\)](#).

Zu Entwicklungs- und Testzwecken können Sie die Bereitstellungsfunktionalität von Elastic Beanstalk verwenden, um eine verwaltete DB-Instance direkt zu Ihrer Umgebung hinzuzufügen. Informationen zur Einrichtung einer Datenbank in der Umgebung finden Sie unter [Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung \(p. 609\)](#).

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, empfehlen wir, dass Sie [einen benutzerdefinierten Domänennamen für die Umgebung konfigurieren \(p. 644\)](#) und [HTTPS aktivieren \(p. 778\)](#) für sichere Verbindungen.

Das AWS Toolkit for Visual Studio – Arbeiten mit .Net Core

Das AWS Toolkit for Visual Studio ist ein Plugin für die Visual Studio-IDE. Mit dem Toolkit können Sie Anwendungen in Elastic Beanstalk bereitstellen und verwalten, während Sie in Ihrer Visual Studio-Umgebung arbeiten.

In diesem Thema wird gezeigt, wie Sie die folgenden Aufgaben mit dem AWS Toolkit for Visual Studio ausführen können:

- Erstellen Sie eine ASP.NET Core-Webanwendung mit einer Visual Studio-Vorlage.
- Erstellen Sie eine Elastic Beanstalk Amazon Linux-Umgebung.
- Stellen Sie die ASP.NET Core-Webanwendung in der neuen Amazon Linux-Umgebung bereit.

In diesem Thema wird auch erläutert, wie Sie mit dem AWS Toolkit for Visual Studio Ihre Elastic Beanstalk-Anwendungsumgebungen verwalten und die Integrität Ihrer Anwendung überwachen können.

Abschnitte

- [Voraussetzungen \(p. 169\)](#)
- [Erstellen eines neuen Anwendungsprojekts \(p. 169\)](#)
- [Erstellen einer Elastic Beanstalk-Umgebung und Bereitstellen Ihrer Anwendung \(p. 171\)](#)
- [Beenden einer Umgebung \(p. 174\)](#)
- [Verwalten Ihrer Elastic Beanstalk-Anwendungsumgebungen \(p. 175\)](#)
- [Überwachen des Anwendungsstatus \(p. 186\)](#)

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie das AWS Toolkit for Visual Studio installieren. Anweisungen finden Sie unter [Einrichten des AWS Toolkit for Visual Studio](#).

Wenn Sie das Toolkit noch nie genutzt haben, müssen Sie nach dessen Installation zunächst Ihre AWS-Anmeldeinformationen mit dem Toolkit registrieren. Weitere Informationen hierzu finden Sie unter [Bereitstellen von AWS-Anmeldeinformationen](#).

Erstellen eines neuen Anwendungsprojekts

Wenn Sie über kein .NET Core-Anwendungsprojekt in Visual Studio verfügen, können Sie ganz einfach eines mit einer der Visual Studio-Projektvorlagen erstellen.

So erstellen Sie ein neues ASP.NET Core-Webanwendungsprojekt:

1. Wählen Sie im Menü File (Datei) von Visual Studio New (Neu) aus und dann Project (Projekt).

2. Wählen Sie im Dialogfeld Create a new project (Neues Projekt erstellen) die Option C#, klicken Sie dann auf Linux und anschließend auf Cloud.
3. Wählen Sie in der Liste der angezeigten Projektvorlagen ASP.NET Core Web Application (ASP.NET Core-Webanwendung) aus, und klicken Sie dann auf Weiter.

Note

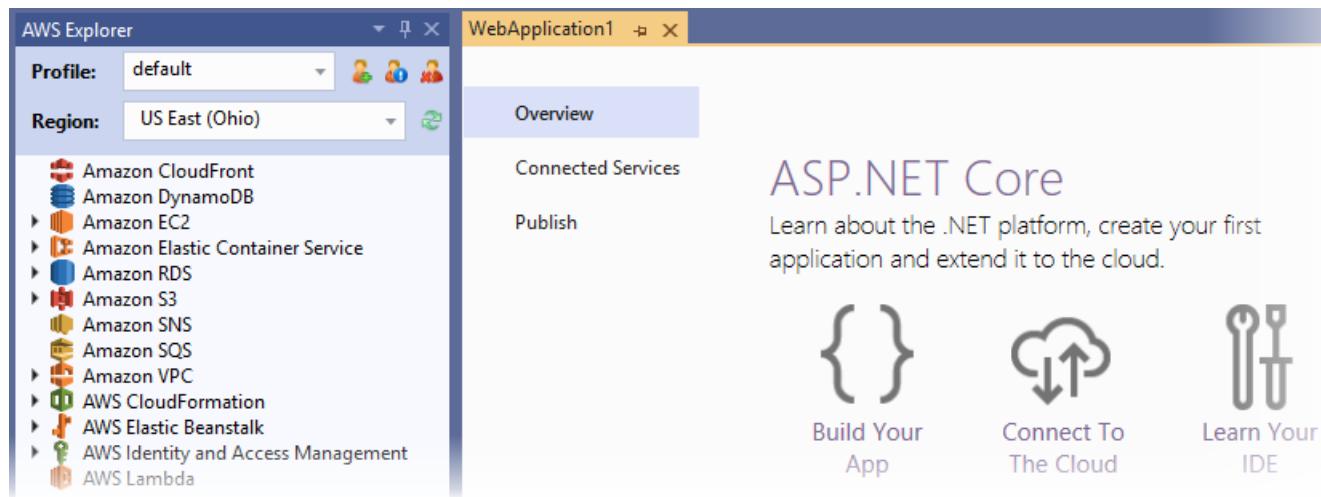
Wenn ASP.NET Core Web Application (ASP.NET Core-Webanwendung) nicht in den Projektvorlagen aufgeführt ist, können Sie sie in Visual Studio installieren.

1. Führen Sie einen Bildlauf zum Ende der Vorlagenliste durch, und wählen Sie den Link Weitere Tools und Funktionen installieren unter der Vorlagenliste aus.
2. Wenn Sie aufgefordert werden, zuzulassen, dass die Visual Studio-Anwendung Änderungen an Ihrem Gerät vornimmt, wählen Sie Yes (Ja) aus.
3. Wählen Sie die Registerkarte Workloads aus, und klicken Sie dann auf ASP.NET and web development (ASP.NET und Webentwicklung).
4. Klicken Sie auf die Schaltfläche Ändern. Über den Visual Studio Installer (Installationsprogramm von Visual-Studio) wird dann die Projektvorlage installiert.
5. Nachdem das Installationsprogramm abgeschlossen ist, verlassen Sie das Fenster, um wieder an die Stelle zurückzukehren, an der Sie in Visual Studio aufgehört haben.
4. Geben Sie im Dialogfeld Configure your new project (Neues Projekt erstellen) einen Projektnamen ein. Als Solution name (Lösungsname) wird standardmäßig der Projektname verwendet. Klicken Sie als Nächstes auf Erstellen.
5. Wählen Sie im Dialogfeld Create a new ASP.NET Core web application (Neue ASP.NET Core-Webanwendung erstellen) die Option .NET Core aus, und klicken Sie dann auf ASP.NET Core 3.1. Wählen Sie in der Liste der angezeigten Anwendungstypen Webanwendung aus, und klicken Sie dann auf die Schaltfläche Erstellen.

Create a new ASP.NET Core web application

The screenshot shows the 'Create a new ASP.NET Core web application' dialog. At the top, there are two dropdown menus: '.NET Core' and 'ASP.NET Core 3.1'. Below them, the 'Empty' template is selected, showing its description: 'An empty project template for creating an ASP.NET Core application. This template does not have any content in it.' The 'API' template is also listed with its description: '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.' The 'Web Application' template is highlighted with a blue background, showing its description: 'A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.' The 'Web Application (Model-View-Controller)' template is also listed with its description: '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.'

In Visual Studio wird während der Anwendungserstellung das Dialogfeld Creating Project (Projekt wird erstellt) angezeigt. Nachdem Visual Studio Ihre Anwendung erstellt hat, wird ein Fenster mit dem Namen Ihrer Anwendung angezeigt.

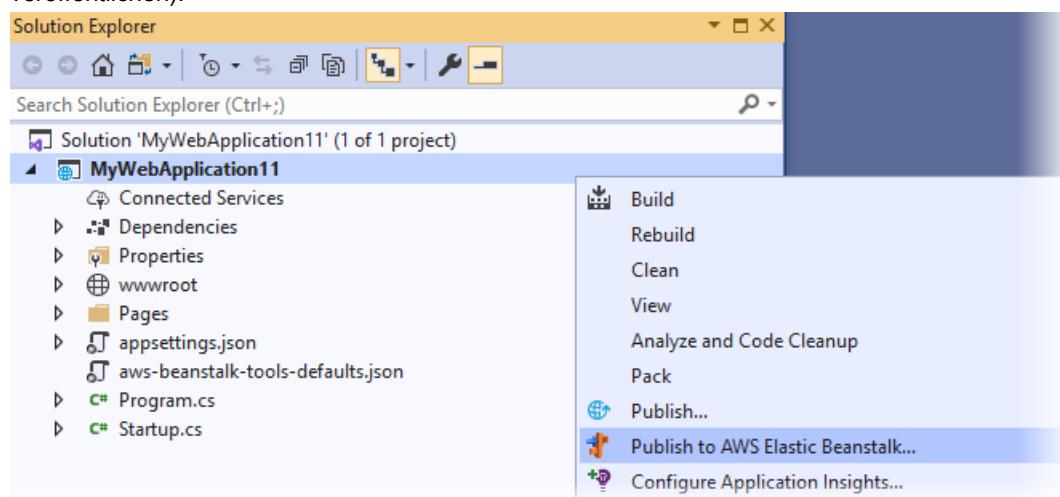


Erstellen einer Elastic Beanstalk-Umgebung und Bereitstellen Ihrer Anwendung

In diesem Abschnitt wird beschrieben, wie Sie eine Elastic Beanstalk-Umgebung für Ihre Anwendung erstellen und Ihre Anwendung in dieser Umgebung bereitstellen.

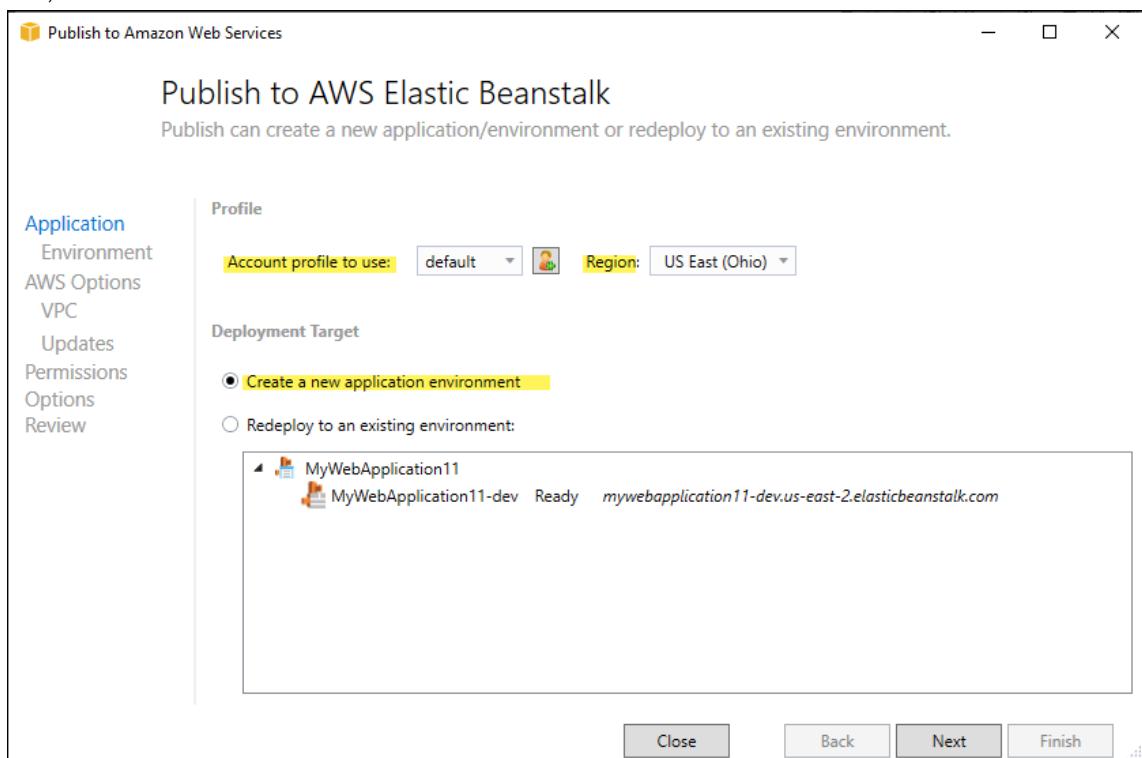
Erstellen einer neuen Umgebung und Bereitstellen Ihrer Anwendung

1. Wählen Sie in Visual Studio die Option View (Anzeigen) und dann Solution Explorer (Lösungs-Explorer) aus.
2. Öffnen Sie im Solution Explorer (Lösungs-Explorer) das Kontextmenü (mit Rechtsklick) für die Anwendung und klicken Sie dann auf Publish to AWS Elastic Beanstalk (In AWS Elastic Beanstalk veröffentlichen).



3. Geben Sie im Assistenten Publish to AWS Elastic Beanstalk (In AWS Elastic Beanstalk veröffentlichen) Ihre Kontoinformationen ein.
 - a. Wählen Sie für Account profile to use (Zu verwendetes Kontoprofil) Ihr Standard-Konto oder klicken Sie auf das Add another account (Ein anderes Konto hinzufügen)-Symbol, um neue Kontoinformationen einzugeben.

- b. Wählen Sie unter Region die Region aus, in der die Anwendung bereitgestellt werden soll. Informationen zu verfügbaren AWS-Regionen finden Sie unter [AWS Elastic Beanstalk Endpunkte und Kontingente](#) in der Allgemeinen AWS-Referenz. Falls Sie eine nicht von Elastic Beanstalk unterstützte Region auswählen, ist die Option für eine Elastic Beanstalk-Bereitstellung nicht verfügbar.
- c. Wählen Sie Create a new application environment (Eine neue Anwendungsumgebung erstellen) aus, und klicken Sie dann auf Weiter.



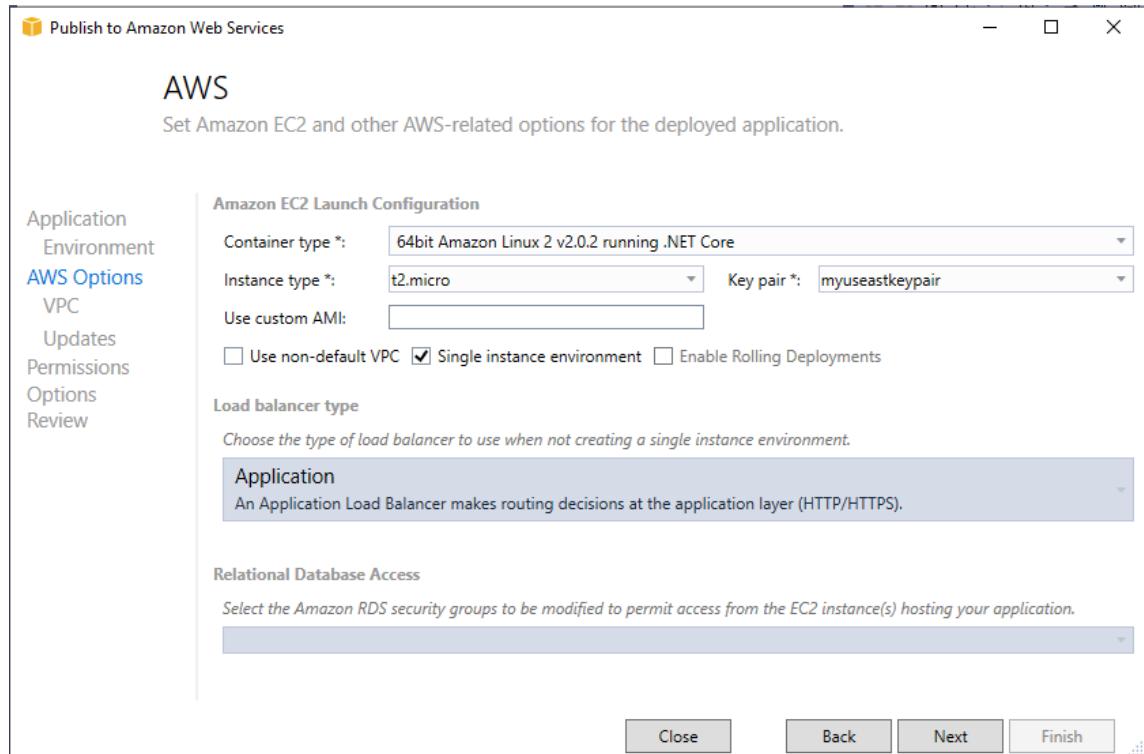
4. Geben Sie im Dialogfeld Application Environment (Anwendungsumgebung) die Details für die neue Anwendungsumgebung ein.
5. Legen Sie im nächsten AWS-Optionsdialogfeld die Amazon EC2-Optionen und andere AWS-bezogene Optionen für die bereitgestellte Anwendung fest.
 - a. Wählen Sie für Container-Typ die Option 64bit Amazon Linux 2 v<n.n.n> mit .NET Core aus.

Note

Wir empfehlen Ihnen, die aktuelle Plattformversion von Linux auszuwählen. Diese Version enthält die neuesten Sicherheits- und Fehlerbehebungen, die in unserem neuesten Amazon Machine Image (AMI) enthalten sind.

- b. Wählen Sie für Instance-Typ die Option t2.micro aus. (Durch die Wahl eines Micro-Instance-Typs werden die mit dem Betrieb der Instance verbundenen Kosten minimiert).
- c. Wählen Sie im Feld Key pair (Schlüsselpaar) die Option Create new key pair (Neues Schlüsselpaar erstellen) aus. Geben Sie einen Namen für das neue Schlüsselpaar ein, und wählen Sie dann OK aus. (In diesem Beispiel verwenden wir **myuseastkeypair**.) Ein Schlüsselpaar ermöglicht den Remote-Desktop-Zugriff auf Ihre Amazon EC2-Instances. Weitere Informationen zu Amazon EC2-Schlüsselpaaren finden Sie unter [Verwenden von Anmeldeinformationen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.
- d. Für eine einfache Anwendung mit geringem Datenverkehr wählen Sie Single instance environment (Single-Instance-Umgebung) aus. Weitere Informationen finden Sie unter [Umgebungstypen \(p. 514\)](#).

- e. Klicken Sie auf Next (Weiter).



Weitere Informationen zu den AWS-Optionen, die in diesem Beispiel nicht verwendet werden, finden Sie auf den folgenden Seiten:

- Informationen zu Use custom AMI (Verwenden eines benutzerdefinierten AMI) finden Sie unter [Verwenden eines benutzerdefinierten Amazon Machine Image \(AMI\) \(p. 773\)](#).
 - Wenn Sie keine Single instance environment (Single-Instance-Umgebung) verwenden, müssen Sie einen Load balance type (Load-Balance-Typ) auswählen. Weitere Informationen finden Sie unter [Load Balancer Ihrer AWS Elastic Beanstalk-Umgebung \(p. 554\)](#).
 - Elastic Beanstalk verwendet die Standardkonfiguration [Amazon VPC](#) (Amazon Virtual Private Cloud), sofern Sie nicht die Option Use non-default VPC (Nicht-Standard-VPC verwenden) ausgewählt haben. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon VPC \(p. 988\)](#).
 - Wenn Sie die Option Enable Rolling Deployments (Rollende Bereitstellungen aktivieren) auswählen, wird eine Bereitstellung in Stapel aufgeteilt, um mögliche Ausfallzeiten während der Bereitstellungen zu vermeiden. Weitere Informationen finden Sie unter [Anwendungen in Elastic Beanstalk-Umgebungen bereitstellen \(p. 471\)](#).
 - Durch die Auswahl der Option Relational Database Access (Zugriff auf relationale Datenbank) wird die Elastic Beanstalk-Umgebung in einer zuvor erstellten Amazon RDS-Datenbank mit Amazon RDS-DB-Sicherheitsgruppen verknüpft. Weitere Informationen finden Sie unter [Zugriffskontrolle mit Sicherheitsgruppen](#) im Amazon RDS-Benutzerhandbuch.
6. Klicken Sie im Dialogfeld Berechtigungen auf Weiter.
 7. Klicken Sie im Dialogfeld Applications Options (Anwendungsoptionen) auf Weiter.
 8. Überprüfen Sie die Bereitstellungsoptionen. Nachdem Sie überprüft haben, dass Ihre Einstellungen korrekt sind, wählen Sie Deploy (Bereitstellen) aus.

Ihre ASP.NET Core-Webanwendung wird als Webbereitstellungsdatei exportiert. Diese Datei wird anschließend in Amazon S3 hochgeladen und als neue Anwendungsversion in Elastic Beanstalk registriert. Mit der Bereitstellungsfunktion von Elastic Beanstalk wird die Umgebung überwacht, bis sie mit dem neu bereitgestellten Code verfügbar ist. Auf der Registerkarte „Env:<Umgebungsname>“ wird der Status für Ihre Umgebung angezeigt. Nachdem der Status auf Environment is healthy (Umgebung ist fehlerfrei) aktualisiert wurde, können Sie die URL-Adresse auswählen, um die Webanwendung zu starten.

The screenshot shows the AWS Toolkit for Visual Studio interface. The title bar says "MyWebApplication11" and "Env: MyWebApplication11-dev". The main window displays environment details:

URL:	http://mywebapplication11-dev.us-east-2.elasticbeanstalk.com/		
Application:	MyWebApplication11	Running Version:	v20200721020104
Container Type:	64bit Amazon Linux 2 v1.0.0 running .NET Core	Created:	7/21/2020 2:54:23 AM
Status:	Environment is healthy		

Below this is a table titled "Events" showing logs:

Event	Event Time	Event Type	Version Label	Event Details
Monitoring	7/21/2020 2:56:54 AM	INFO		Successfully launched environment: MyWebApplication11
Resources	7/21/2020 2:56:54 AM	INFO		Application available at mywebapplication11-dev.us-east-
AWS X-Ray	7/21/2020 2:56:45 AM	INFO	v20200721020104	Added EC2 instance 'i-00c5680f13fc6f089' to Auto Scaling
Server	7/21/2020 2:56:45 AM	INFO	v20200721020104	Environment health has been set to GREEN
Notifications	7/21/2020 2:56:45 AM	INFO	v20200721020104	Adding instance 'i-00c5680f13fc6f089' to your environment
Container	7/21/2020 2:56:18 AM	INFO		Waiting for EC2 instances to launch. This may take a few minutes.
Advanced	7/21/2020 2:54:58 AM	INFO		Created EIP: 3.14.235.39
Logs	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 Amazon VPC
	7/21/2020 2:54:23 AM	INFO		createEnvironment is starting.

Beenden einer Umgebung

Beenden Sie eine ausgeführte Umgebung mit dem AWS Toolkit for Visual Studio, damit keine Kosten für ungenutzte AWS-Ressourcen anfallen.

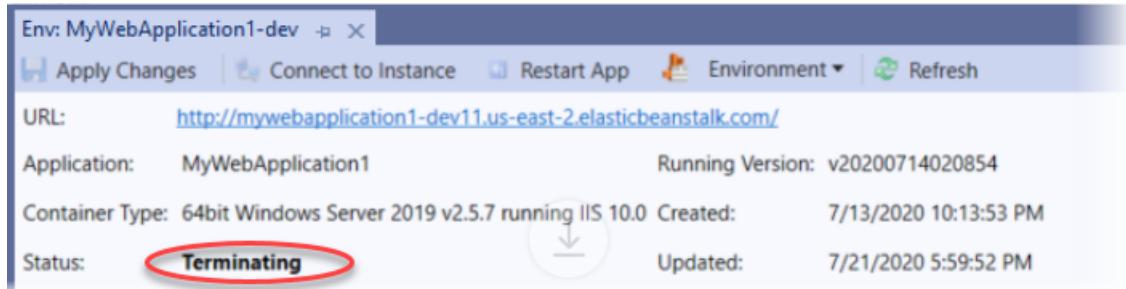
Note

Sie können zu einem späteren Zeitpunkt problemlos eine neue Umgebung mit derselben Version starten.

So beenden Sie eine Umgebung

1. Erweitern Sie den Elastic Beanstalk-Knoten und den Anwendungsknoten. Öffnen Sie im AWS Explorer das Kontextmenü (mit Rechtsklick) für Ihre Anwendungsumgebung und wählen Sie Terminate Environment (Umgebung beenden) aus.
2. Wenn Sie dazu aufgefordert werden, wählen Sie Ja, um zu bestätigen, dass Sie die Umgebung beenden möchten. Es dauert einige Minuten, bis Elastic Beanstalk die AWS-Ressourcen beendet hat, die in der Umgebung ausgeführt werden.

Der Status für Ihre Umgebung auf der Registerkarte „Env:<Umgebungsname>“ ändert sich in Wird beendet und schließlich in Beendet.



Note

Nachdem die Umgebung beendet ist, steht ihr zugehöriger CNAME wieder zur Verfügung und kann von allen verwendet werden.

Verwalten Ihrer Elastic Beanstalk-Anwendungsumgebungen

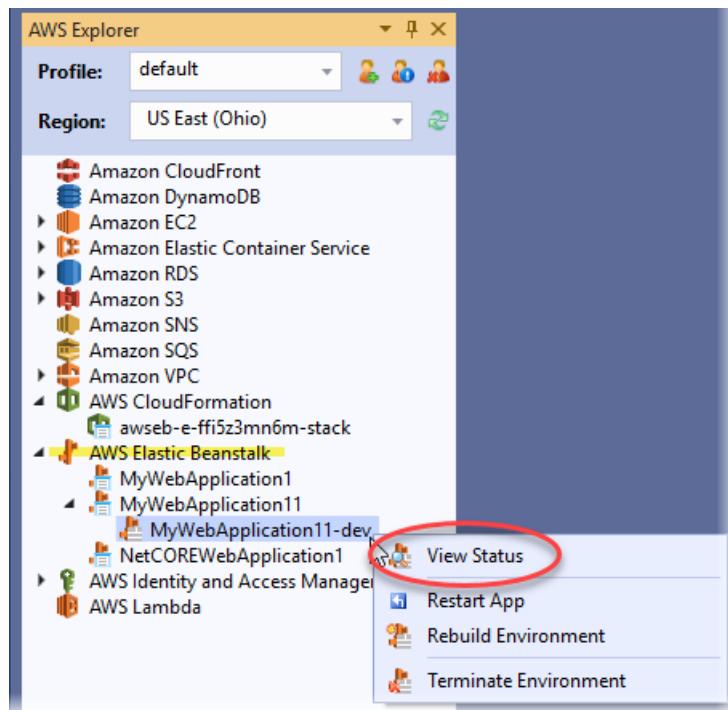
Mit dem AWS Toolkit for Visual Studio und der AWS-Managementkonsole können Sie die Bereitstellung und Konfiguration der AWS-Ressourcen, die von Ihren Anwendungsumgebungen verwendet werden, ändern. Weitere Informationen zur Verwaltung Ihres Anwendungsumgebungen mithilfe der AWS Management Console finden Sie unter [Verwalten von Umgebungen \(p. 425\)](#). In diesem Abschnitt werden die spezifischen Service-Einstellungen beschrieben, die Sie im AWS Toolkit for Visual Studio als Teil Ihrer Anwendungs-Umgebungskonfiguration bearbeiten können.

Ändern der Umgebungskonfigurationseinstellungen

Wenn Sie Ihre Anwendung bereitstellen, konfiguriert Elastic Beanstalk eine Reihe von AWS Cloud Computing-Services. Sie können mit dem AWS Toolkit for Visual Studio steuern, wie diese einzelnen Services konfiguriert sind.

So bearbeiten Sie die Umgebungseinstellungen einer Anwendung

1. Wählen Sie in Visual Studio im Menü Datei die Option AWS Explorer.
2. Erweitern Sie den Elastic Beanstalk-Knoten und Ihren Anwendungsknoten. Öffnen Sie das Kontextmenü (mit Rechtsklick) für Ihre Anwendungsumgebung, und wählen Sie View Status (Status anzeigen).



Sie können nun Einstellungen für Folgendes konfigurieren:

- AWS X-Ray
- herstellen
- Load Balancer (gilt nur für Umgebungen mit mehreren Instances)
- Auto Scaling (gilt nur für Umgebungen mit mehreren Instances)
- Benachrichtigungen
- Container
- Erweiterte Konfigurationsoptionen

Konfigurieren von AWS X-Ray mit dem AWS Toolkit for Visual Studio

AWS X-Ray bietet Funktionen für die Anfragenachverfolgung, Ausnahmensammlung und Profilerstellung. Im AWS X-Ray-Bereich können Sie AWS X-Ray für Ihre Anwendung aktivieren oder deaktivieren. Weitere Informationen zu X-Ray finden Sie im [AWS X-Ray-Entwicklerhandbuch](#).

AWS X-Ray Console. To learn how to instrument your .NET application visit the [AWS X-Ray SDK for .NET GitHub repository](#).'"/>

Konfigurieren von EC2-Instances mit dem AWS Toolkit for Visual Studio

Sie können Amazon Elastic Compute Cloud (Amazon EC2) verwenden, um Server-Instances in den Rechenzentren von Amazon zu starten und zu verwalten. Sie können Amazon EC2-Server-Instances zu jedem beliebigen Zeitpunkt, so lange wie nötig und für alle rechtlichen Zwecke verwenden. Instances sind in verschiedenen Größen und Konfigurationen verfügbar. Weitere Informationen finden Sie unter [Amazon EC2](#).

Sie können die Amazon EC2-Instance-Konfiguration über die Registerkarte Server innerhalb der Registerkarte für die Anwendungsumgebung im AWS Toolkit for Visual Studio bearbeiten.

Amazon EC2-Instance-Typen

Instance type (Instance-Typ) zeigt die Instance-Typen an, die für die Elastic Beanstalk-Anwendung verfügbar sind. Ändern Sie den Instance-Typ, um einen Server mit den Merkmalen (einschließlich Speichergröße und CPU-Leistung) auszuwählen, die für Ihre Anwendung am besten geeignet sind. Anwendungen mit aufwändigen und lange dauernden Operationen erfordern mehr CPU oder Speicher.

Weitere Informationen über die für die Elastic Beanstalk-Anwendung verfügbaren Amazon EC2-Instance-Typen finden Sie unter [Instance-Typen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Amazon EC2-Sicherheitsgruppen

Sie können den Zugriff auf Ihre Elastic Beanstalk-Anwendung mit einer Amazon EC2-Sicherheitsgruppe steuern. Eine Sicherheitsgruppe definiert Firewall-Regeln für Ihre Instances. Diese Regeln legen fest,

welcher eingehende Netzwerkverkehr an Ihre Instance übertragen werden soll. Der gesamte andere eingehende Datenverkehr wird verworfen. Sie können die Regeln für eine Gruppe jederzeit ändern. Die neuen Regeln werden automatisch für alle laufenden und künftigen Instances durchgesetzt.

Sie können angeben, welche Amazon EC2-Sicherheitsgruppen den Zugriff auf Ihre Elastic Beanstalk-Anwendung steuern. Geben Sie dazu die Namen bestimmter Amazon EC2-Sicherheitsgruppen (trennen Sie dabei mehrere Sicherheitsgruppen durch Kommas) in das Textfeld EC2 Security Groups (EC2-Sicherheitsgruppen) ein. Dazu können Sie entweder die AWS-Managementkonsole oder das AWS Toolkit for Visual Studio verwenden.

So erstellen Sie eine Sicherheitsgruppe mit dem AWS Toolkit for Visual Studio

1. Erweitern Sie in Visual Studio unter AWS Explorer den Amazon EC2-Knoten, und wählen Sie dann die Option Sicherheitsgruppen aus.
2. Wählen Sie Sicherheitsgruppe erstellen aus und geben Sie einen Namen und eine Beschreibung für Ihre Sicherheitsgruppe ein.
3. Wählen Sie OK.

Weitere Informationen zu Amazon EC2-Sicherheitsgruppen finden Sie unter [Verwenden von Sicherheitsgruppen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Amazon EC2-Schlüsselpaare

Mit einem Amazon EC2-Schlüsselpaar können Sie sich sicher an den für Ihre Elastic Beanstalk-Anwendung bereitgestellten Amazon EC2-Instances anmelden.

Important

Sie müssen ein Amazon EC2-Schlüsselpaar erstellen und Ihre von Elastic Beanstalk bereitgestellten Amazon EC2-Instances konfigurieren, um auf diese Instances zugreifen zu können. Sie können ein Schlüsselpaar mit dem Assistenten Publish to AWS (In AWS veröffentlichen) im AWS Toolkit for Visual Studio erstellen, wenn Sie Ihre Anwendung für Elastic Beanstalk bereitstellen. Wenn Sie zusätzliche Schlüsselpaare mit dem Toolkit erstellen möchten, führen Sie die hier beschriebenen Schritte aus. Alternativ können Sie Ihre Amazon EC2-Schlüsselpaare mit der [AWS-Managementkonsole](#) einrichten. Anleitungen zum Erstellen eines Schlüsselpaares für Amazon EC2 finden Sie im [Amazon Elastic Compute Cloud-Handbuch „Erste Schritte“](#).

Im Textfeld Existing Key Pair (Vorhandenes Schlüsselpaar) können Sie den Namen eines Amazon EC2-Schlüsselpaares angeben, das Sie verwenden können, um sich sicher bei den Amazon EC2-Instances anzumelden, die Ihre Elastic Beanstalk-Anwendung ausführen.

So geben Sie den Namen eines Amazon EC2-Schlüsselpaares ein

1. Erweitern Sie den Amazon EC2-Knoten und wählen Sie Schlüsselpaareaus.
2. Wählen Sie Schlüsselpaar erstellen aus und geben Sie den Namen des Schlüsselpaares ein.
3. Wählen Sie OK.

Weitere Informationen zu Amazon EC2-Schlüsselpaaren finden Sie unter [Verwenden von Amazon EC2-Anmeldeinformationen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch. Weitere Information zum Verbinden mit Amazon EC2-Instances erhalten Sie unter

Überwachungsintervall

Standardmäßig werden nur grundlegende Amazon CloudWatch-Metriken aktiviert. Sie geben Daten in 5-Minuten-Intervallen zurück. Sie können noch genauere CloudWatch-Metriken in 1-Minuten-Intervallen

aktivieren, indem Sie auf der Registerkarte Configuration (Konfiguration) im Abschnitt Server für den Wert Monitoring Interval (Überwachungsintervall) den Wert 1 minute (1 Minute) für Ihre Umgebung im AWS Toolkit for Eclipse auswählen.

Note

Für Metriken in 1-Minuten-Intervallen können Amazon CloudWatch-Servicegebühren anfallen. Weitere Informationen finden Sie unter [Amazon CloudWatch](#).

Benutzerdefinierte AMI-ID

Sie können das für Ihre Amazon EC2-Instances verwendete Standard-AMI mit einem eigenen, benutzerdefinierten AMI überschreiben. Geben Sie dazu den Bezeichner Ihres benutzerdefinierten AMI auf der Registerkarte Configuration (Konfiguration) im Abschnitt Server in das Feld Custom AMI ID (ID für benutzerdefiniertes AMI) für Ihre Umgebung im AWS Toolkit for Eclipse ein.

Important

Die Verwendung eines eigenen AMI ist eine Aufgabe für Fortgeschrittene und sollte mit Bedacht ausgeführt werden. Wenn Sie ein benutzerdefiniertes AMI benötigen, empfehlen wir, das standardmäßige Elastic Beanstalk-AMI als Ausgangspunkt zu nehmen und es zu ändern. Um als fehlerfrei zu gelten, erwartet Elastic Beanstalk, dass die Amazon EC2-Instances eine Reihe von Anforderungen erfüllen, einschließlich Vorhandensein eines laufenden Hostmanagers. Werden diese Anforderungen nicht erfüllt, funktioniert Ihre Umgebung möglicherweise nicht ordnungsgemäß.

Konfigurieren von Elastic Load Balancing mit dem AWS Toolkit for Visual Studio

Elastic Load Balancing ist ein Amazon-Webservice, der Ihnen hilft, die Verfügbarkeit und Skalierbarkeit Ihrer Anwendung zu verbessern. Dieser Service macht es für Sie einfacher, Anwendungslasten auf zwei oder mehr Amazon EC2-Instances zu verteilen. Elastic Load Balancing verbessert die Verfügbarkeit durch die Bereitstellung zusätzlicher Redundanz und unterstützt wachsenden Datenverkehr für Ihre Anwendung.

Mit Elastic Load Balancing können Sie den eingehenden Anwendungsdatenverkehr automatisch auf alle ausgeführten Instances verteilen und ausgleichen. Sie können auch ganz einfach neue Instances hinzufügen, wenn eine Erhöhung der Kapazität Ihrer Anwendung erforderlich ist.

Elastic Beanstalk stellt Elastic Load Balancing automatisch bereit, wenn Sie eine Anwendung bereitstellen. Sie können die Amazon EC2-Instance-Konfiguration der Elastic Beanstalk-Umgebung über die Registerkarte Load Balancer innerhalb der Registerkarte für die Anwendungsumgebung im AWS Toolkit for Visual Studio bearbeiten.

The screenshot shows the AWS Toolkit for Visual Studio interface. On the left is a navigation sidebar with the following tabs: Events, Monitoring, Resources, AWS X-Ray, Server, Load Balancer (which is selected and highlighted in blue), Auto Scaling, Notifications, Container, Advanced, and Logs. The main content area has several sections:

- Events**: These settings allow you to control the behavior of your environment's load balancer.
- Monitoring**: HTTP Listener Port: 80, HTTPS Listener Port: OFF, SSL Certificate ID: [empty input field].
- Resources**: Application Health Check: / [empty input field].
- AWS X-Ray**: [empty section]
- Server**: [empty section]
- Load Balancer**:
 - Auto Scaling**: [empty section]
 - Notifications**: [empty section]
 - Container**: [empty section]
 - Advanced**:
 - Health Check Interval (seconds): 60
 - Health Check Timeout (seconds): 120
 - Healthy Check Count Threshold: 2
 - Unhealthy Check Count Threshold: 3
 - Logs**: [empty section]

In den folgenden Abschnitten werden die Elastic Load Balancing-Parameter beschrieben, die Sie für Ihre Anwendung konfigurieren können.

Ports

Der Load Balancer, der für die Verarbeitung von Anforderungen für Ihre Elastic Beanstalk-Anwendung bereitgestellt wurde, sendet Anforderungen an Amazon EC2-Instances, auf denen Ihre Anwendung ausgeführt wird. Der bereitgestellte Load Balancer kann auf Anforderungen über HTTP- und HTTPS-Ports warten und diese an Amazon EC2-Instances in Ihrer AWS Elastic Beanstalk-Anwendung weiterleiten. Standardmäßig verarbeitet der Load Balancer Anfragen auf dem HTTP-Port. Dazu muss mindestens einer der Ports (HTTP oder HTTPS) aktiviert werden.



Important

Stellen Sie sicher, dass der von Ihnen angegebene Port nicht gesperrt ist. Andernfalls können Sie keine Verbindung zu Ihrer Elastic Beanstalk-Anwendung herstellen.

Steuern des HTTP-Ports

Um den HTTP-Port auszuschalten, wählen Sie OFF (AUS) für HTTP Listener Port aus. Zum Aktivieren des HTTP-Ports wählen Sie einen HTTP-Port (z. B. 80) aus der Liste aus.

Note

Für den Zugriff auf Ihre Umgebung mit einem anderen als dem Standardport 80, z. B. Port 8080, fügen Sie dem vorhandenen Load Balancer einen Listener hinzu und konfigurieren Sie den neuen Listener für die Überwachung des entsprechenden Ports.

Geben Sie beispielsweise unter Verwendung der [AWS CLI für Classic Load Balancer](#) den folgenden Befehl ein, wobei Sie `LOAD_BALANCER_NAME` durch den Namen des Load Balancers für Elastic Beanstalk ersetzen.

```
aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
--listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
InstancePort=80"
```

Geben Sie beispielsweise unter Verwendung der [AWS CLI für Application Load Balancer](#) den folgenden Befehl ein, wobei Sie `LOAD_BALANCER_ARN` durch den ARN des Load Balancers für Elastic Beanstalk ersetzen.

```
aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP --
port 8080
```

Wenn Elastic Beanstalk Ihre Umgebung überwachen soll, entfernen Sie den Listener für Port 80 nicht.

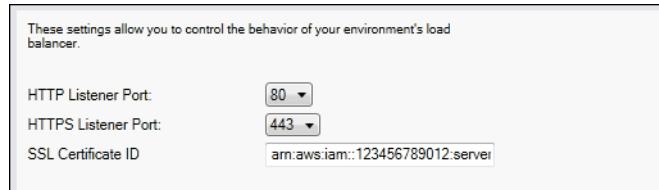
Steuern des HTTPS-Ports

Elastic Load Balancing unterstützt das HTTPS/TLS-Protokoll, um die Verschlüsselung des Datenverkehrs für Client-Verbindungen zum Load Balancer zu aktivieren. Verbindungen vom Load Balancer mit EC2-Instances verwenden eine Klartext-Verschlüsselung. Standardmäßig ist der HTTPS-Port deaktiviert.

So aktivieren Sie den HTTPS-Port

1. Erstellen Sie ein neues Zertifikat mit AWS Certificate Manager (ACM) oder laden Sie ein Zertifikat und einen Schlüssel in AWS Identity and Access Management (IAM) hoch. Weitere Informationen zum Anfordern von ACM-Zertifikaten finden Sie unter [Anfordern eines Zertifikats](#) im AWS Certificate Manager-Benutzerhandbuch. Weitere Informationen zum Importieren von Zertifikaten von Drittanbietern in ACM finden Sie unter [Importieren von Zertifikaten](#) im AWS Certificate Manager-Benutzerhandbuch. Wenn ACM in Ihrer Region nicht verfügbar ist, verwenden Sie AWS Identity and Access Management (IAM), um ein Zertifikat von einem Drittanbieter hochzuladen. Die ACM- und IAM-Services speichern das Zertifikat und stellen einen Amazon-Ressourcennamen (ARN) für das SSL-Zertifikat bereit. Weitere Informationen zum Erstellen und Hochladen von Zertifikaten in IAM finden Sie unter [Arbeiten mit Serverzertifikaten](#) im IAM-Benutzerhandbuch.

2. Geben Sie den HTTPS-Port an, indem Sie einen Port für HTTPS Listener Port auswählen.



3. Geben Sie für SSL Certificate ID (SSL-Zertifikat-ID) den Amazon-Ressourcennamen (ARN) Ihres SSL-Zertifikats ein. Zum Beispiel `arn:aws:iam::123456789012:server-certificate/abc/certs/build` oder `arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`. Verwenden Sie das SSL-Zertifikat, das Sie in Schritt 1 erstellt oder hochgeladen haben.

Um den HTTPS-Port auszuschalten, wählen Sie OFF (AUS) für HTTPS Listener Port.

Zustandsprüfungen

Die Definition der Zustandsprüfung enthält einen URL, die für den Instance-Zustand abgefragt wird. Standardmäßig nutzt Elastic Beanstalk TCP:80 für Nonlegacy-Container und HTTP:80 für Legacy-

Container. Sie können die Standard-URL überschreiben, sodass sie mit einer vorhandenen Ressource der Anwendung übereinstimmt (z. B. `/myapp/default.aspx`). Dazu geben Sie diese im Feld Application health check URL (Zustandsprüfungs-URL für die Anwendung) ein. Wenn Sie die Standard-URL überschreiben, verwendet Elastic Beanstalk HTTP zum Abfragen der Ressource. Wenn Sie überprüfen möchten, ob Sie einen Legacy-Container verwenden, erhalten Sie Informationen unter [the section called "Warum sind einige Plattformversionen als Legacy gekennzeichnet?" \(p. 502\)](#).

Sie können die Einstellungen für die Zustandsprüfung im Abschnitt EC2 Instance Health Check (Zustandsprüfung für EC2-Instances) des Bereichs Load Balancing steuern.

The screenshot shows a configuration dialog for EC2 Instance Health Check. It includes a note: "These settings allow you to configure how Elastic Beanstalk determines whether an EC2 instance is healthy or not." Below are four input fields:

Setting	Value	Description
Application Health Check:	/	(5 - 300)
Health Check Interval (seconds):	30	(2 - 60)
Health Check Timeout (seconds):	5	(2 - 10)
Healthy Check Count Threshold:	3	(2 - 10)
Unhealthy Check Count Threshold:	5	(2 - 10)

Die Definition der Zustandsprüfung enthält einen URL, die für den Instance-Zustand abgefragt wird. Überschreiben Sie die Standard-URL, sodass sie mit einer vorhandenen Ressource der Anwendung übereinstimmt (z. B. `/myapp/index.jsp`). Dazu geben Sie diese im Feld Application Health Check URL (Zustandsprüfungs-URL für die Anwendung) ein.

In der folgenden Liste werden die Zustandsprüfungs-Parameter beschrieben, die Sie für Ihre Anwendung festlegen können.

- Für Health Check Interval (seconds) (Zustandsprüfungsintervall (Sekunden)) geben Sie die Anzahl der Sekunden ein, die Elastic Load Balancing zwischen Zustandsprüfungen für die Amazon EC2-Instances Ihrer Anwendung wartet.
- Für Health Check Timeout (seconds) (Zustandsprüfungs-Timeout (Sekunden)) geben Sie die Anzahl der Sekunden ein, die Elastic Load Balancing auf eine Antwort wartet, bevor die Instance als nicht reagierend eingestuft wird.
- Für Healthy Check Count Threshold (Schwellenwert Anzahl erfolgreicher Zustandsprüfungen) und Unhealthy Check Count Threshold (Schwellenwert Anzahl fehlgeschlagener Zustandsprüfungen) geben Sie die Anzahl der aufeinanderfolgenden erfolgreichen oder nicht erfolgreichen URL-Proben ein, bevor Elastic Load Balancing den Status des Instance-Zustands ändert. Beispiel: Die Angabe von 5 für Unhealthy Check Count Threshold (Schwellenwert Anzahl fehlgeschlagener Zustandsprüfungen) bedeutet, dass die URL fünfmal hintereinander eine Fehlermeldung oder eine Zeitüberschreitung zurückgeben muss, bevor Elastic Load Balancing die Zustandsprüfung als fehlgeschlagen einstufen würde.

Sitzungen

Standardmäßig leitet ein Load Balancer jede Anforderung getrennt an die Server-Instance mit der kleinsten Last weiter. Im Vergleich dazu bindet eine Sticky Session eine Benutzersitzung an eine bestimmte Server-Instance, sodass alle vom Benutzer während der Sitzung gesendeten Anforderungen an die gleiche Server-Instance übermittelt werden.

Elastic Beanstalk nutzt von Load Balancer generierte HTTP-Cookies, wenn Sticky Sessions für eine Anwendung aktiviert sind. Der Load Balancer verwendet ein vom Load-Balancer generiertes spezielles Cookie, um die Anwendungs-Instance für jede Anforderung nachzuverfolgen. Wenn der Load Balancer eine Anforderung empfängt, prüft er zunächst, ob dieses Cookie in der Anforderung vorhanden ist. Wenn es vorhanden ist, wird die Anforderung an die Anwendungs-Instance gesendet, die im Cookie angegeben ist. Wenn kein Cookie vorhanden ist, wählt der Load Balancer eine Anwendungs-Instance basierend auf dem vorhandenen Load Balancing-Algorithmus aus. Ein Cookie wird in die Antwort eingefügt, um nachfolgende Anforderungen von demselben Benutzer an diese Anwendungs-Instance zu binden. Mit der Konfiguration der Richtlinie wird ein Cookie-Ablauf definiert, der die Dauer der Gültigkeit für jedes Cookie festlegt.

Sie können über den Abschnitt Sessions (Sitzungen) auf der Registerkarte Load Balancer angeben, ob der Load Balancer für Ihre Anwendung "Sticky Sessions" erlaubt.

These settings allow you to control how your load balancer handles session cookies.

Enable Session Stickiness

Cookie Expiration Period (seconds): (0 - 1000000)

Weitere Informationen zu Elastic Load Balancing finden Sie im [Elastic Load Balancing-Entwicklerhandbuch](#).

Konfigurieren von Auto Scaling mit dem AWS Toolkit for Visual Studio

Amazon EC2 Auto Scaling ist ein Amazon-Webservice, mit dem Amazon EC2-Instances basierend auf benutzerdefinierten Auslösern automatisch gestartet oder beendet werden. Sie können Auto Scaling-Gruppen einrichten und diesen Gruppen Auslöser zuweisen, um Datenverarbeitungsressourcen automatisch basierend auf Metriken, wie Bandbreitennutzung oder CPU-Auslastung, zu skalieren. Amazon EC2 Auto Scaling ruft mithilfe von Amazon CloudWatch Metriken für die Server-Instances ab, auf denen die Anwendung ausgeführt wird.

Mit Amazon EC2 Auto Scaling können Sie eine Gruppe von Amazon EC2-Instances nehmen und verschiedene Parameter festlegen, damit diese Gruppe automatisch größer oder kleiner wird. Mit Amazon EC2 Auto Scaling können Amazon EC2-Instances zur Gruppe hinzugefügt oder daraus entfernt werden, damit sie nahtlos auf Datenverkehränderungen an Ihrer Anwendung eingehen können.

Des Weiteren wird mit Amazon EC2 Auto Scaling auch die Integrität der einzelnen gestarteten Amazon EC2-Instances überwacht. Falls eine Instance unerwartet beendet wird, erkennt Amazon EC2 Auto Scaling dies und startet eine andere Instance als Ersatz. Mit dieser Funktion wird automatisch stets die von Ihnen festgelegte Anzahl an Amazon EC2-Instances ausgeführt.

Elastic Beanstalk stellt Amazon EC2 Auto Scaling für Ihre Anwendung bereit. Sie können die Amazon EC2-Instance-Konfiguration der Elastic Beanstalk-Umgebung über die Registerkarte Auto Scaling innerhalb der Registerkarte für die Anwendungsumgebung im AWS Toolkit for Visual Studio bearbeiten.

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: (0 - 10000)

Maximum Instance Count: (0 - 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:

Upper Breach Scalement Increment:

Lower Threshold:

Lower Breach Scalement Increment:

Im folgenden Abschnitt wird beschrieben, wie Sie Auto Scaling-Parameter für die Anwendung konfigurieren.

Starten der Konfiguration

Sie können die Startkonfiguration so bearbeiten, dass gesteuert wird, wie Ihre Elastic Beanstalk-Anwendung Amazon EC2 Auto Scaling-Ressourcen bereitstellt.

In den Feldern Minimum Instance Count (Minimale Instance-Anzahl) und Maximum Instance Count (Maximale Instance-Anzahl) geben Sie die minimale sowie die maximale Größe der Auto Scaling-Gruppe an, die von der Elastic Beanstalk-Anwendung genutzt wird.

The screenshot shows a configuration dialog for an Auto Scaling group. At the top, there is a note: "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." Below the note are four input fields:

- Minimum Instance Count: A text input field containing "1" with a tooltip "(1 - 10000)".
- Maximum Instance Count: A text input field containing "4" with a tooltip "(1 - 10000)".
- Availability Zones: A dropdown menu set to "Any".
- Scaling Cooldown Time (seconds): A text input field containing "360" with a tooltip "(0 - 10000)".

Note

Um die Anzahl der Amazon EC2-Instances als Festwert festzulegen, geben Sie in den Feldern Minimum Instance Count (Minimale Instance-Anzahl) und Maximum Instance Count (Maximale Instance-Anzahl) denselben Wert ein.

Im Feld Availability Zones geben Sie die Anzahl der Availability Zones an, in denen sich Ihre Amazon EC2-Instances befinden sollen. Es ist wichtig, diese Zahl festzulegen, wenn Sie fehlertolerante Anwendungen erstellen möchten. Falls eine Availability Zone ausfällt, werden die Instances in den anderen Availability Zones weiterhin ausgeführt.

Note

Derzeit ist es nicht möglich anzugeben, in welcher Availability Zone sich Ihre Instance befinden wird.

Auslöser

Bei einem Auslöser handelt es sich um einen Amazon EC2 Auto Scaling-Mechanismus, mit dem Sie definieren, wann die Instance-Anzahl erhöht (hochskaliert) bzw. verringert (herunterskaliert) werden soll. Sie können die Auslöser so konfigurieren, dass Sie bei allen Metriken, die in Amazon CloudWatch veröffentlicht werden, ausgelöst werden (z. B. die CPU-Auslastung). Zudem wird ermittelt, ob die vorgegebenen Bedingungen erfüllt sind. Wird der obere oder untere Schwellenwert der Metrikbedingungen für einen bestimmten Zeitraum überschritten, startet der Auslöser den anhaltenden Prozess der Skalierung.

Sie können einen Skalierungsauslöser für Ihre Elastic Beanstalk-Anwendung mit dem AWS Toolkit for Visual Studio definieren.

Trigger Measurement:	NetworkOut
Trigger Statistic:	Average
Unit of Measurement:	Bytes
Measurement Period (minutes):	5 (1 - 600)
Breach Duration (minutes):	5 (1 - 600)
Upper Threshold:	6000000 (0 - 20000000)
Upper Breach Scalement Increment:	1
Lower Threshold:	2000000 (0 - 20000000)
Lower Breach Scalement Increment:	-1

Die Amazon EC2 Auto Scaling-Auslöser überwachen eine angegebene Amazon CloudWatch-Metrik einer bestimmten Instance. Zu den Metriken zählen CPU-Auslastung, Netzwerkdatenverkehr und Datenträgeraktivität. Mit der Einstellung Trigger Measurement (Auslösermessung) wählen Sie eine Metrik für den Auslöser aus.

In der folgenden Liste werden die Auslöserparameter beschrieben, die Sie mit der AWS-Managementkonsole konfigurieren können.

- Sie können angeben, welche Statistik der Auslöser verwenden soll. Sie können Minimum, Maximum, Sum (Summe) oder Average (Durchschnitt) für Trigger Statistic (Auslöserstatistik) auswählen.
- Für Unit of Measurement (Maßeinheit) geben Sie die Maßeinheit für den Auslöser an.
- Der Wert im Feld Measurement Period (Messzeitraum) gibt an, wie häufig Amazon CloudWatch die Metriken für Ihren Auslöser misst. Mit Breach Duration (Verletzungsdauer) wird der Zeitraum angegeben, für den der Grenzwert einer Metrik überschritten werden kann (wie unter Upper Threshold (Oberer Schwellenwert) und Lower Threshold (Unterer Schwellenwert) definiert), bevor der Auslöser ausgelöst wird.
- Unter Upper Breach Scale Increment (Oberes Skalierungssinkrement für eine Verletzung) und Lower Breach Scale Increment (Unteres Skalierungssinkrement für eine Verletzung) geben Sie an, wie viele Amazon EC2-Instances im Rahmen der Skalierung hinzugefügt oder entfernt werden sollen.

Weitere Informationen zu Amazon EC2 Auto Scaling finden Sie im Abschnitt Amazon EC2 Auto Scaling in der [Dokumentation zu Amazon Elastic Compute Cloud](#).

Konfigurieren von Benachrichtigungen mit dem AWS Toolkit for Visual Studio

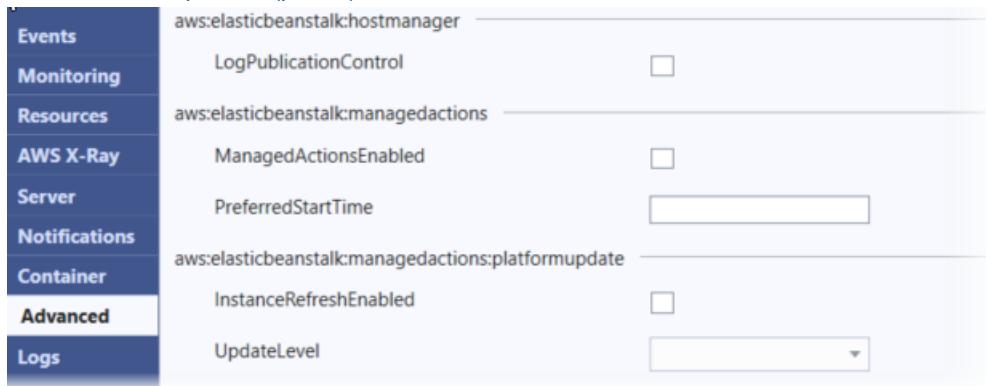
Elastic Beanstalk verwendet den Amazon Simple Notification Service (Amazon SNS) für Benachrichtigungen zu wichtigen Ereignissen in Ihrer Anwendung. Um Amazon SNS-Benachrichtigungen zu aktivieren, geben Sie Ihre E-Mail-Adresse in das Feld Email Address ein. Um diese Benachrichtigungen zu deaktivieren, entfernen Sie Ihre E-Mail-Adresse aus dem Feld.

The screenshot shows a sidebar menu with the following items: Events, Monitoring, Resources, AWS X-Ray, Server, **Notifications**, Container, Advanced, and Logs. The Notifications item is currently selected. To its right, there is a text input field labeled "Email Address" with the placeholder 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 the input field, there is a small text area with the placeholder "Email Address:" followed by a blank input field.

Konfigurieren zusätzlicher Umgebungsoptionen mithilfe von AWS Toolkit for Visual Studio

Elastic Beanstalk definiert eine große Anzahl von Konfigurationsoptionen, mithilfe derer Sie das Verhalten Ihrer Umgebung und die darin enthaltenen Ressourcen konfigurieren können. Konfigurationsoptionen sind in Namespaces wie `aws:autoscaling:asg` organisiert. In jedem Namespace werden Optionen für die Auto Scaling-Gruppe einer Umgebung definiert. Im Fenster Erweitert werden die Namespaces der Konfigurationsoptionen in alphabetischer Reihenfolge aufgelistet, die Sie nach der Erstellung der Umgebung aktualisieren können.

Eine vollständige Liste der Namespaces und Optionen, einschließlich der Standard- und unterstützten Werte, finden Sie unter [Allgemeine Optionen für alle Umgebungen \(p. 666\)](#) und unter [.NET Core auf Linux-Plattformoptionen \(p. 716\)](#).



Konfigurieren von .NET Core-Containern mit dem AWS Toolkit for Visual Studio

Im Fenster Container können Sie Umgebungsvariablen angeben, die Sie aus Ihrem Anwendungscode lesen können.

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"/>

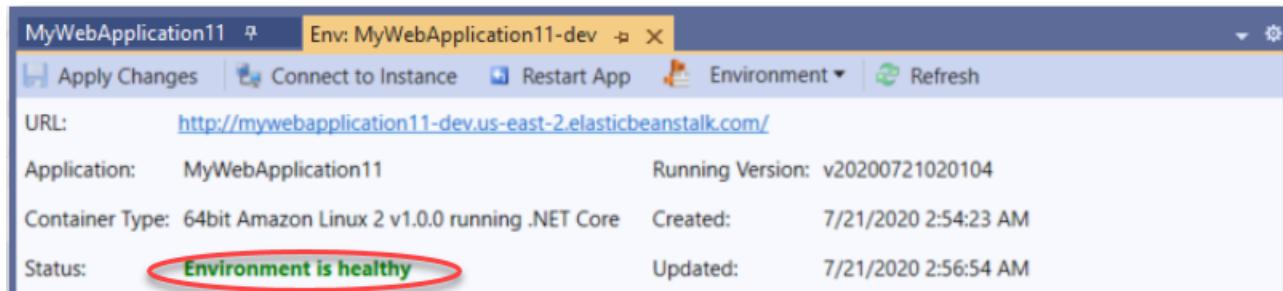
Überwachen des Anwendungsstatus

Es ist wichtig zu wissen, dass Ihre Produktionswebsite verfügbar ist und auf Anforderungen reagiert. Elastic Beanstalk bietet Funktionen, mit denen Sie die Reaktionsfähigkeit Ihrer Anwendung überwachen können. Es werden Statistiken über Ihre Anwendung überwacht und Sie werden gewarnt, wenn Schwellenwerte überschritten werden.

Weitere Informationen über die Integritätsüberwachung von Elastic Beanstalk finden Sie unter [Grundlegende Zustandsberichte \(p. 821\)](#).

Sie können auf operative Informationen zu Ihrer Anwendung zugreifen, indem Sie das AWS Toolkit for Visual Studio oder die AWS-Managementkonsole verwenden.

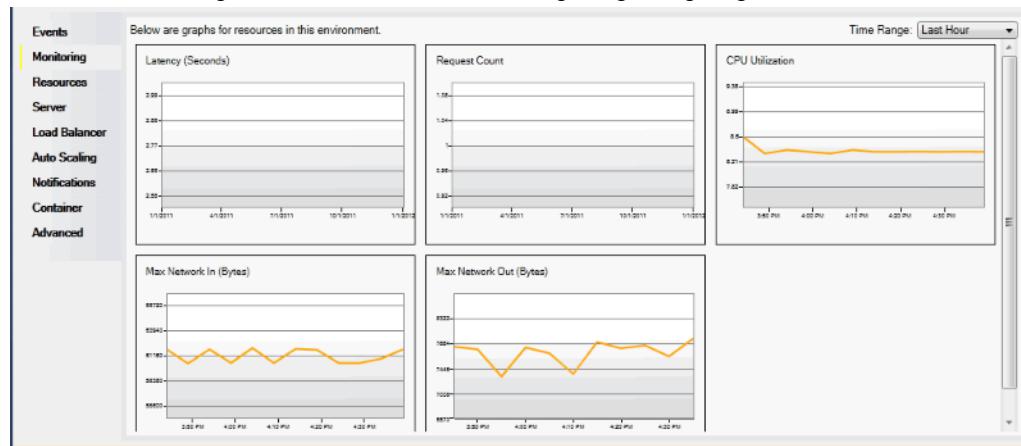
Das Toolkit zeigt den Zustand Ihrer Umgebung und den Anwendungsstatus im Feld Status an.



So überwachen Sie den Anwendungsstatus

1. Erweitern Sie im AWS Toolkit for Visual Studio in AWS Explorer den Elastic Beanstalk-Knoten und erweitern Sie dann Ihren Anwendungsknoten.
2. Öffnen Sie das Kontextmenü (mit Rechtsklick) für Ihre Anwendungsumgebung, und wählen Sie View Status (Status anzeigen).
3. Wählen Sie auf der Registerkarte der Anwendungsumgebung Monitoring (Überwachung) aus.

Der Bereich Monitoring (Überwachung) enthält eine Reihe von Diagrammen, die die Ressourcennutzung für Ihre bestimmte Anwendungsumgebung zeigen.



Note

Standardmäßig ist der Zeitraum auf der letzten Stunde gesetzt. Ändern Sie diese Einstellung, indem Sie in der Liste Zeitraum einen anderen Zeitraum auswählen.

Sie können das AWS Toolkit for Visual Studio oder die AWS-Managementkonsole verwenden, um Ereignisse im Zusammenhang mit Ihrer Anwendung anzuzeigen.

So zeigen Sie Anwendungsereignisse an

1. Erweitern Sie im AWS Toolkit for Visual Studio in AWS Explorer den Elastic Beanstalk-Knoten und Ihren Anwendungsknoten.
2. Öffnen Sie das Kontextmenü (mit Rechtsklick) für Ihre Anwendungsumgebung, und wählen Sie View Status (Status anzeigen).
3. Wählen Sie in der Registerkarte der Anwendungsumgebung Ereignisse aus.

Events	Filter: <input type="text"/>			
	Event Time	Event Type	Version Label	Event Details
Monitoring	7/21/2020 2:56:54 AM	INFO		Successfully launched environment: MyWebApplication11
Resources	7/21/2020 2:56:54 AM	INFO		Application available at mywebapplication11-dev.us-east-
AWS X-Ray	7/21/2020 2:56:45 AM	INFO	v20200721020104	Added EC2 instance 'i-00c5680f13fc6f089' to Auto Scaling
Server	7/21/2020 2:56:45 AM	INFO	v20200721020104	Environment health has been set to GREEN
Notifications	7/21/2020 2:56:45 AM	INFO	v20200721020104	Adding instance 'i-00c5680f13fc6f089' to your environme
Container	7/21/2020 2:56:18 AM	INFO		Waiting for EC2 instances to launch. This may take a few r
Advanced	7/21/2020 2:54:58 AM	INFO		Created EIP: 3.14.235.39
Logs	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 Amazo
	7/21/2020 2:54:23 AM	INFO		createEnvironment is starting.

Migration von .NET unter Windows Server-Plattformen zur .NET Core unter Linux-Plattform

Sie können Anwendungen, die auf Plattformen auf der Basis von [.NET unter Windows Server](#) ausgeführt werden, zu .NET Core unter Linux-Plattformen migrieren. Im Folgenden werden einige Aspekte beschrieben, die bei der Migration von Windows zu Linux-Plattformen berücksichtigt werden sollten.

Aspekte, die bei der Migration zur .NET Core unter Linux-Plattform berücksichtigt werden sollten

Area	Änderungen und Informationen
Anwendungskontext	Auf Windows-Plattformen verwenden Sie ein Bereitstellungsmanifest (p. 197) zur Angabe der in Ihrer Umgebung ausgeführten Anwendungen. Die .NET Core unter Linux-Plattformen verwenden eine Procfile (p. 162) -Datei zur Angabe der auf den Instances Ihrer Umgebung ausgeführten Anwendungen. Details zur Bündelung von Anwendungen finden Sie unter the section called “Anwendungsbündelung” (p. 161) .
Proxy-Server	Auf Windows-Plattformen verwenden Sie IIS als Proxy-Server Ihrer Anwendung. Die .NET Core unter Linux-Plattformen enthalten standardmäßig nginx als Reverse-Proxy. Sie können festlegen, dass kein Proxy-Server verwendet wird, und Kestrel als Webserver Ihrer Anwendung verwenden. Weitere Informationen hierzu finden Sie unter the section called “Proxy-Server” (p. 163) .
Weiterleitung	Auf Windows-Plattformen verwenden Sie IIS in Ihrem Anwendungscode und fügen ein Bereitstellungsmanifest (p. 197) ein, um den IIS-Pfad zu konfigurieren. Für die .NET Core unter Linux-Plattform verwenden Sie ASP .NET Core-Routing im Anwendungscode und aktualisieren die nginx-Konfiguration Ihrer Umgebung. Weitere Informationen hierzu finden Sie unter the section called “Proxy-Server” (p. 163) .
Protokolle	Linux- und Windows-Plattformen streamen verschiedene Protokolle. Details dazu finden Sie unter the section called “Wie Elastic Beanstalk CloudWatch Logs einrichtet” (p. 883) .

Erstellen und Bereitstellen von .NET-Anwendungen in Elastic Beanstalk

AWS Elastic Beanstalk for .NET erleichtert das Bereitstellen, Verwalten und Skalieren Ihrer ASP.NET-Webanwendungen, die Amazon Web Services verwenden. Elastic Beanstalk for .NET ist für jede Person verfügbar, die eine Webanwendung bereitstellt oder hostet, die IIS verwendet.

Get started now (Jetzt einsteigen): Unter [Tutorial: Erstellen und Bereitstellen einer .NET-Beispielanwendung mit Elastic Beanstalk. \(p. 201\)](#) können Sie in einem Tutorial direkt die ersten Schritte erlernen. In diesem Tutorial stellen Sie eine Beispiel-ASP.NET-Webanwendung für einen AWS Elastic Beanstalk-Anwendungscontainer bereit.

Der Rest dieses Abschnitts enthält Anweisungen zum Erstellen, Testen, Bereitstellen und erneutem Bereitstellen Ihrer ASP.NET-Webanwendung für Elastic Beanstalk. Einige Beispiele zeigen die Verwendung des AWS Toolkit for Visual Studio. Im Unterabschnitt [the section called "Das AWS Toolkit for Visual Studio" \(p. 220\)](#) wird erläutert, wie Sie Ihre Anwendungen und Umgebungen mit dem Toolkit verwalten und konfigurieren. Weitere Informationen über Voraussetzungen, Installationsanweisungen und das Ausführen von Codebeispielen finden Sie unter [AWS Toolkit für Microsoft Visual Studio](#). Diese Website bietet auch nützliche Informationen über Tools, Themen zur Vorgehensweise und zusätzliche Ressourcen für ASP.NET-Entwickler.

Note

Diese Plattform unterstützt die folgenden Funktionen von Elastic Beanstalk nicht:

- Worker-Umgebungen. Details dazu finden Sie unter [AWS Elastic Beanstalk-Worker-Umgebungen \(p. 516\)](#).
- Bundle-Protokolle. Details dazu finden Sie unter [Anzeigen von Instance-Protokollen \(p. 869\)](#).

Darüber hinaus unterstützen Plattformversionen vor v2.0.0.0 keine erweiterten Zustandsberichte, verwalteten Plattformaktualisierungen, unveränderlichen Aktualisierungen, unveränderlichen Bereitstellungen und fortlaufenden Bereitstellungen mit einem zusätzlichen Stapel.

In den Themen in diesem Kapitel wird davon ausgegangen, dass Sie über einige Kenntnisse der Elastic Beanstalk-Umgebungen verfügen. Wenn Sie Elastic Beanstalk noch nicht verwendet haben, machen Sie sich im [Tutorial "Erste Schritte" \(p. 3\)](#) mit den Grundlagen vertraut.

Themen

- [Erste Schritte mit .NET in Elastic Beanstalk \(p. 189\)](#)
- [Einrichten der .NET-Entwicklungsumgebung \(p. 192\)](#)
- [Verwenden der Elastic Beanstalk .NET-Plattform \(p. 193\)](#)
- [Tutorial: Erstellen und Bereitstellen einer .NET-Beispielanwendung mit Elastic Beanstalk. \(p. 201\)](#)
- [Tutorial: Bereitstellen einer ASP.NET Core-Anwendung mit Elastic Beanstalk \(p. 208\)](#)
- [Hinzufügen einer Amazon RDS-DB-Instance zur .NET-Anwendungsumgebung \(p. 217\)](#)
- [Das AWS Toolkit for Visual Studio \(p. 220\)](#)
- [Migrieren Ihrer lokalen .NET-Anwendung zu Elastic Beanstalk \(p. 247\)](#)
- [Ressourcen \(p. 247\)](#)

Erste Schritte mit .NET in Elastic Beanstalk

Für die ersten Schritte mit .NET-Anwendungen in AWS Elastic Beanstalk benötigen Sie nur das [Quell-Bundle \(p. 412\)](#) einer Anwendung, damit Sie die erste Anwendungsversion hochladen und in einer

Umgebung bereitstellen können. Im Rahmen der Umgebungserstellung werden von Elastic Beanstalk alle erforderlichen AWS-Ressourcen für eine hoch skalierbare Webanwendung zugewiesen.

Starten einer Umgebung mit einer .NET-Beispielanwendung

Elastic Beanstalk bietet für jede Plattform sowohl einfache Beispielanwendungen mit einer Seite als auch komplexe Beispiele, in denen die Nutzung weiterer AWS-Ressourcen (wie z. B. Amazon RDS; sowie sprach- oder plattformspezifische Funktionen und APIs) veranschaulicht wird.

Beispiele

Name	Unterstützte Konfigurationen	Umgang mit dem Source	Beschreibung
.NET WS 2012 R2	Default	Webseitenvorlage-asp-v1.zip	ASP.NET-Webanwendung mit einer Seite, die im Stamm der Website angezeigt wird.
WS 2012	WS 2008 R2		
ASP.NET 2012 R2	MVC5	Webseitenvorlage-aspmvc5-v1.zip	ASP.NET-Webanwendung mit einer klassischen Model-View-Steuerungsarchitektur.

Laden Sie eine beliebige der Beispielanwendungen herunter und stellen Sie sie mit dem folgenden Verfahren für Elastic Beanstalk bereit.

So starten Sie eine Umgebung mit einer Beispielanwendung (Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Applications (Anwendungen) und anschließend in der Liste den Namen einer vorhandenen Anwendung aus. Sie können auch [eine Anwendung erstellen \(p. 403\)](#).
3. Wählen Sie auf der Anwendungsübersichtsseite die Option Create a new environment (Neue Umgebung erstellen).

Environment name	Health	Date created	Last modified	URL	Platform
GettingStartedApp-env	OK	2020-01-28 12:06:50 UTC-0800	2020-01-30 15:02:35 UTC-0800	GettingStartedApp-env.17dx222kw.us-east-2.elasticbeanstalk.com	Tomcat running Linux
GettingStartedApp-Windows	OK	2020-01-28 16:34:29 UTC-0800	2020-01-28 16:38:20 UTC-0800	GettingStartedApp-Windows.17dx222kw.us-east-2.elasticbeanstalk.com	IIS 10.0 Windows

4. Wählen Sie die Web server environment (Webserverumgebung) oder die Worker environment (Worker-Umgebung) als [Umgebungsebene \(p. 14\)](#) aus. Nach der Erstellung lässt sich die Umgebungsebene nicht mehr ändern.

Note

Die .NET-Plattform auf Windows Server (p. 189) unterstützt die Worker-Umgebungsschicht nicht.

The screenshot shows the 'Select environment tier' step in the AWS Elastic Beanstalk console. At the top, the breadcrumb navigation shows 'Elastic Beanstalk > Applications > getting-started-app'. The main title 'Select environment tier' is displayed. Below it, a descriptive text explains that AWS Elastic Beanstalk has two types of environment tiers: 'Web server environment' and 'Worker environment'. The 'Web server environment' option is selected, indicated by a blue circular icon with a dot. A sub-description states: 'Run a website, web application, or web API that serves HTTP requests.' Below this, a 'Learn more' link with a blue icon is visible. The 'Worker environment' option is also listed with its own sub-description: 'Run a worker application that processes long-running workloads on demand or performs tasks on a schedule.' Another 'Learn more' link with a blue icon is provided. The background of the page is white, and the overall interface is clean and modern.

5. Wählen Sie in Platform (Plattform) die Plattform und den Plattformzweig aus, die/der mit der Sprache Ihrer Anwendung übereinstimmt.

Note

Elastic Beanstalk unterstützt mehrere [Versionen \(p. 32\)](#) für die meisten Plattformen, die aufgelistet sind. Standardmäßig wählt die Konsole die empfohlene Version für die gewählte Plattform und Plattformvariante. Wenn Ihre Anwendung eine andere Version erfordert, können Sie diese hier auswählen oder [Configure more options \(Weitere Optionen konfigurieren\)](#) auswählen wie in Schritt 7 beschrieben. Informationen zu unterstützten Plattformversionen finden Sie unter [the section called "Unterstützte Plattformen" \(p. 32\)](#).

6. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
7. Wählen Sie [Configure more options \(Weitere Optionen konfigurieren\)](#) aus, um Ihre Umgebung weiter anzupassen. Sie können die folgenden Optionen nur während des Erstellens der Umgebung festlegen:
 - Environment name
 - Domänenname
 - Plattformversion
 - VPC
 - Stufe

Sie können die folgenden Einstellungen nach der Umgebungserstellung zwar ändern, jedoch müssen dafür neue Instances oder andere Ressourcen bereitgestellt werden, deren Anwendung viel Zeit in Anspruch nehmen kann:

- Instance-Typ, Stamm-Volume, Schlüsselpaar und AWS Identity and Access Management (IAM)-Rolle
- Interne Amazon RDS-Datenbank
- Load Balancer

Weitere Informationen zu allen verfügbaren Einstellungen finden Sie unter [Der Assistent zum Erstellen einer neuen Umgebung \(p. 437\)](#).

8. Wählen Sie Create environment (Umgebung erstellen) aus.

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit [eine neue Version \(p. 471\)](#) der Anwendung oder eine gänzlich andere Anwendung bereitstellen. Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine EC2-Instances bereitgestellt oder neu gestartet werden müssen.

Wenn Sie eine oder zwei Beispielanwendungen bereitgestellt haben, können Sie mit der lokalen Entwicklung beginnen. Die entsprechenden Informationen zum Einrichten einer .NET-Entwicklungsumgebung finden Sie im [nächsten Abschnitt \(p. 192\)](#).

Einrichten der .NET-Entwicklungsumgebung

Richten Sie eine .NET-Entwicklungsumgebung ein, um Ihre Anwendung vor der Bereitstellung in AWS Elastic Beanstalk lokal zu testen. In diesem Thema finden Sie die Schritte zum Einrichten der Entwicklungsumgebung sowie Links zu den Installationsseiten nützlicher Tools.

Allgemeine Einrichtungsschritte und Tools für alle Sprachen finden Sie unter [Konfigurieren des Entwicklungskomputers für die Verwendung mit Elastic Beanstalk \(p. 1005\)](#).

Abschnitte

- [Installieren einer IDE \(p. 159\)](#)
- [Installieren des AWS Toolkit for Visual Studio \(p. 159\)](#)

Wenn Sie die AWS-Ressourcen aus der Anwendung heraus verwalten möchten, installieren Sie das AWS SDK for .NET. Beispielsweise können Sie mit Amazon S3 Daten speichern und abrufen.

Mit dem AWS SDK for .NET können Sie in nur wenigen Minuten loslegen. Das einzelne Download-Paket enthält Visual Studio-Projektvorlagen, die AWS .NET-Bibliothek, C#-Codebeispiele und die Referenzdokumentation. Praktische Beispiele dazu, wie mit den Bibliotheken Anwendungen erstellt werden können, sind in C# enthalten. Online-Videotutorials und Referenzdokumentationen werden bereitgestellt, um Sie dabei zu unterstützen, die Verwendung der Bibliotheken und Beispielcodes zu erlernen.

Weitere Informationen und Installationsanleitungen finden Sie auf der [AWS SDK for .NET-Homepage](#).

Installieren einer IDE

Integrierte Entwicklungsumgebungen (IDEs) bieten zahlreiche Funktionen, mit denen die Anwendungsentwicklung vereinfacht wird. Wenn Sie noch nie eine IDE für die .NET-Entwicklung verwendet haben, können Sie mit Visual Studio Community erste Schritte unternehmen.

Besuchen Sie die [Visual Studio Community](#)-Webseite, um Visual Studio Community herunterzuladen und zu installieren.

Installieren des AWS Toolkit for Visual Studio

Das [AWS Toolkit for Visual Studio \(p. 220\)](#) ist ein Open-Source-Plug-in für die Visual Studio-IDE, das die Entwicklung, das Debuggen und die Bereitstellung von .NET-Anwendungen mit AWS für Entwickler vereinfacht. Installationsanleitungen finden Sie auf der [Toolkit for Visual Studio-Homepage](#).

Verwenden der Elastic Beanstalk .NET-Plattform

AWS Elastic Beanstalk unterstützt zahlreiche Plattformen für verschiedene Versionen der .NET-Programmierumgebung und von Windows Server. Eine komplette Liste finden Sie unter [.NET on Windows Server mit IIS](#) im Dokument AWS Elastic Beanstalk-Plattformen.

Elastic Beanstalk bietet [Konfigurationsoptionen \(p. 646\)](#), mit denen sich die auf den EC2-Instances in der Elastic Beanstalk-Umgebung ausgeführte Software anpassen lässt. Sie können die für Ihre Anwendung erforderlichen Umgebungsvariablen konfigurieren, die Protokollrotation an Amazon S3 aktivieren und .NET-Umgebungseinstellungen festlegen.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das [Ändern der Konfiguration einer ausgeführten Umgebung verfügbar \(p. 658\)](#). Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie [gespeicherte Konfigurationen \(p. 764\)](#) verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie [Konfigurationsdateien \(p. 722\)](#) einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, wenn vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe [Konfigurationsoptionen \(p. 646\)](#).

Konfigurieren Ihrer .NET-Umgebung in der Elastic Beanstalk-Konsole

Mit der Elastic Beanstalk-Konsole können Sie die Protokollrotation für Amazon S3 aktivieren, Variablen konfigurieren, die die Anwendung in der Umgebung lesen kann, und die .NET-Framework-Einstellungen ändern.

So konfigurieren Sie Ihre .NET-Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.

4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.

Containeroptionen

- Target .NET runtime (.NET-Ziellaufzeit) – Legen Sie den Wert 2.0 fest, um CLR v2 auszuführen.
- Enable 32-bit applications (32-Bit-Anwendungen aktivieren) – Legen Sie den Wert True fest, um 32-Bit-Anwendungen auszuführen.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren) – Gibt an, ob die Protokolldateien für die Amazon EC2-Instances der Anwendung in den zugeordneten Amazon S3-Bucket kopiert werden sollen.

Umgebungseigenschaften

Im Bereich Environment Properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Diese Einstellungen werden als Schlüssel-Wert-Paare an die Anwendung weitergeleitet. Verwenden Sie `System.EnvironmentVariable`, um diese zu lesen. Identische Schlüssel können in `web.config` und als Umgebungseigenschaften vorhanden sein. Verwenden Sie den `System.Configuration`-Namespace zum Lesen von Werten aus `web.config`.

```
NameValueCollection appConfig = ConfigurationManager.AppSettings;  
string endpoint = appConfig["API_ENDPOINT"];
```

Weitere Informationen finden Sie unter [Umgebungseigenschaften und andere Softwareeinstellungen \(p. 621\)](#).

Der Namespace aws:elasticbeanstalk:container:dotnet:apppool

Mithilfe einer [Konfigurationsdatei \(p. 722\)](#) können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Konfigurationsoptionen können durch den Elastic Beanstalk-Service oder die verwendete Plattform definiert und in Namespaces organisiert werden.

Über die .NET-Plattform werden Optionen im Namespace `aws:elasticbeanstalk:container:dotnet:apppool` festgelegt, mithilfe derer Sie die .NET-Laufzeitumgebung konfigurieren können.

In der folgenden Beispielkonfigurationsdatei werden die Einstellungen der einzelnen Optionen angezeigt, die in diesem Namespace verfügbar sind:

Example `.ebextensions/dotnet-settings.config`

```
option_settings:  
  aws:elasticbeanstalk:container:dotnet:apppool:  
    Target Runtime: 2.0  
    Enable 32-bit Applications: True
```

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Migration über die Hauptversionen der Elastic Beanstalk- Windows Server-Plattform hinweg

AWS Elastic Beanstalk hatte mehrere Hauptversionen seiner Windows Server-Plattform. Diese Seite geht auf die wichtigsten Verbesserungen für jede Hauptversion und darauf ein, was Sie bedenken müssen, bevor Sie auf eine neuere Version migrieren.

Die Windows Server-Plattform liegt derzeit in Version 2 (v2) vor. Wenn Ihre Anwendung eine Version der Windows Server-Plattform vor v2 verwendet, empfehlen wir die Migration auf v2.

Neues in den Hauptversionen der Windows Server-Plattform

Windows Server-Plattform V2

Version 2 (v2) der Elastic Beanstalk Windows Server Plattform wurde [im Februar 2019 veröffentlicht](#). Mit V2 ist das Verhalten der Windows Server-Plattform in mehreren wichtigen Punkten demjenigen der Linux-basierten Plattformen von Elastic Beanstalk ähnlicher. V2 ist vollständig abwärtskompatibel zu v1, wodurch die Migration von v1 ganz einfach ist.

Die Windows Server-Plattform unterstützt jetzt Folgendes:

- Versioning – Jede Version erhält eine neue Versionsnummer, und Sie können auf frühere Versionen (die noch verfügbar sind) verweisen, wenn Sie Umgebungen erstellen und verwalten.
- Erweiterter Zustandsbericht – Weitere Informationen finden Sie unter [Erweiterte Zustandsberichte und Überwachung \(p. 824\)](#).
- Bereitstellungen, die unveränderlich und fortlaufend mit zusätzlichem Stapel sind – Weitere Informationen zu Bereitstellungsrichtlinien finden Sie unter [Anwendungen in Elastic Beanstalk-Umgebungen bereitstellen \(p. 471\)](#).
- Unveränderliche Updates – Weitere Informationen zu den Update-Typen finden Sie unter [Konfigurationsänderungen \(p. 483\)](#).
- Verwaltete Plattformaktualisierungen – Weitere Informationen finden Sie unter [Verwaltete Plattformaktualisierungen \(p. 496\)](#).

Note

Die neuen Bereitstellungs- und Update-Funktionen sind vom erweiterten Zustandsbericht abhängig. Um sie verwenden zu können, müssen Sie den erweiterten Zustandsbericht aktivieren. Details dazu finden Sie unter [Aktivieren der erweiterten Elastic Beanstalk-Integritätsberichte \(p. 832\)](#).

Windows Server-Plattform V1

Version 1.0.0 (v1) der Elastic Beanstalk-Windows Server-Plattform wurde im Oktober 2015 veröffentlicht. In dieser Version ändert sich die Reihenfolge, in der die Elastic Beanstalk-Prozesse Befehle in [Konfigurationsdateien \(p. 722\)](#) bei der Erstellung und Aktualisierungen der Umgebung verarbeitet werden.

Vorherige Plattformversionen haben keine Versionsnummer im Lösungs-Stack-Namen:

- Windows Server 2012 R2 mit 64 Bit und IIS 8.5
- Windows Server Core 2012 R2 mit 64 Bit und IIS 8.5

- Windows Server 2012 mit 64 Bit und IIS 8
- Windows Server 2008 R2 mit 64 Bit und IIS 7.5

In früheren Versionen war die Verarbeitungsreihenfolge für Konfigurationsdateien inkonsistent. Während der Umgebungserstellung wurde `Container Commands` ausgeführt, nachdem die Anwendungsquelle für IIS bereitgestellt wurde. Bei einer Bereitstellung in einer ausgeführten Umgebung werden Container-Befehle ausgeführt, bevor die neue Version bereitgestellt wird. Bei einer Skalierung nach oben werden Konfigurationsdateien nicht verarbeitet.

Darüber hinaus wird IIS gestartet, bevor Container-Befehle ausgeführt werden. Dieses Verhalten hat einige Kunden zur Implementierung von Workarounds in Container-Befehlen veranlasst, wobei der IIS-Server angehalten wird, bevor Befehle ausgeführt werden, und nach Abschluss wieder gestartet wird.

Version 1 behebt die Inkonsistenzen. Damit ist das Verhalten der Windows Server-Plattform demjenigen der Linux-basierten Plattformen von Elastic Beanstalk ähnelicher. Bei der v1-Plattform führt Elastic Beanstalk immer Container-Befehle vor dem Start des IIS-Servers aus.

Die Lösungs-Stacks der v1-Plattform haben ein v1 nach der Windows Server-Version:

- Windows Server 2012 R2 v1.1.0 mit 64 Bit und IIS 8.5
- Windows Server Core 2012 R2 v1.1.0 mit 64 Bit und IIS 8.5
- Windows Server 2012 v1.1.0 mit 64 Bit und IIS 8
- Windows Server 2008 R2 v1.1.0 mit 64 Bit und IIS 7.5

Außerdem extrahiert die v1-Plattform den Inhalt Ihres Anwendungs-Quell-Bundles in `C:\staging\`, bevor Container-Befehle ausgeführt werden. Nachdem die Container-Befehle abgeschlossen wurden, wird der Inhalt dieses Ordners als ZIP-Datei komprimiert und für IIS bereitgestellt. In diesem Workflow können Sie den Inhalt Ihres Anwendungs-Quell-Bundles mit Befehlen oder einem Skript vor der Bereitstellung ändern.

Migration von früheren Hauptversionen der Windows Server-Plattform

Lesen Sie diesen Abschnitt mit Hinweisen zur Migration, bevor Sie Ihre Umgebung aktualisieren. Weitere Informationen zur Aktualisierung Ihrer Umgebungsplattform auf eine neuere Version finden Sie unter [Aktualisieren der Plattformversion für die Elastic Beanstalk-Umgebung \(p. 491\)](#).

Kopieren von V1 in V2

.NET Core 1.x und 2.0 werden von der v2-Plattform von Windows Server nicht unterstützt. Wenn Sie Ihre Anwendung von Windows Server v1 auf v2 migrieren und Ihre Anwendung eine dieser .NET Core-Versionen nutzt, aktualisieren Sie Ihre Anwendung auf eine .NET-Core Version, die von v2 unterstützt wird. Eine Liste der unterstützten Versionen finden Sie unter [.NET on Windows Server with IIS](#) in den AWS Elastic Beanstalk-Plattformen.

Wenn Ihre Anwendung ein benutzerdefiniertes Amazon Machine Image (AMI) verwendet, erstellen Sie ein neues benutzerdefiniertes AMI basierend auf einer Windows Server-Plattform v2 AMI. Weitere Informationen hierzu finden Sie unter [Verwenden eines benutzerdefinierten Amazon Machine Image \(AMI\) \(p. 773\)](#).

Note

Welche Bereitstellungs- und Update-Funktionen in Windows Server v2 neu sind, ist vom erweiterten Zustandsbericht abhängig. Wenn Sie eine Umgebung auf v2 migrieren, ist der erweiterte Zustandsbericht deaktiviert. Sie müssen sie aktivieren, um diese Funktionen verwenden zu können. Details dazu finden Sie unter [Aktivieren der erweiterten Elastic Beanstalk-Integritätsberichte \(p. 832\)](#).

Von Pre-V1-Versionen

Wenn Sie Ihre Anwendung von einem Windows Server-Lösungs-Stack vor v1 migrieren und derzeit Containerbefehle verwenden, müssen Sie zusätzlich zu den Überlegungen, die auf eine Migration von v1 zutreffen, alle Befehle entfernen, die Sie hinzugefügt haben, um die Verarbeitungskonsistenzen beim Migrieren auf eine neuere Version zu umgehen. Ab v1 werden Container-Befehle garantiert vollständig ausgeführt, bevor die Anwendungsquelle bereitgestellt und bevor IIS gestartet wird. Auf diese Weise können Sie Änderungen an der Quelle in c:\staging durchführen und IIS-Konfigurationsdateien in diesem Schritt ohne Probleme ändern.

Sie können über die AWS CLI zum Beispiel eine DLL-Datei in Ihre Anwendungsquelle aus Amazon S3 herunterladen:

```
.ebextensions\copy-dll.config
```

```
container_commands:  
copy-dll:  
  command: aws s3 cp s3://DOC-EXAMPLE-BUCKET/dlls/large-dll.dll .\lib\
```

Weitere Informationen zur Verwendung von Konfigurationsdateien finden Sie unter [Erweiterte Umgebungsanpassung mit Konfigurationsdateien \(.ebextensions\) \(p. 722\)](#).

Ausführen mehrerer Anwendungen und ASP.NET Core-Anwendungen mit einem Bereitstellungsmanifest

Sie können mit einem Bereitstellungsmanifest Elastic Beanstalk anweisen, wie Ihre Anwendung bereitgestellt werden soll. Beispiel: Statt msDeploy zum Generieren eines Quell-Bundle für eine einzelne ASP.NET-Anwendung zu verwenden, die auf dem Stammpfad Ihrer Website ausgeführt wird, können Sie eine Manifestdatei verwenden, um mehrere Anwendungen auf unterschiedlichen Pfaden auszuführen oder Elastic Beanstalk anzulegen, die App mit ASP.NET Core auszuführen. Sie können ein Bereitstellungsmanifest auch zum Konfigurieren eines Anwendungspools verwenden, in dem Sie Ihre Anwendungen ausführen.

Bereitstellungsmanifeste fügen Unterstützung für [.NET Core-Anwendungen \(p. 198\)](#) zu Elastic Beanstalk hinzu. Sie können eine .NET Framework-Anwendung ohne ein Bereitstellungsmanifest bereitstellen, .NET Core-Anwendungen benötigen dieses jedoch, um auf Elastic Beanstalk ausgeführt zu werden. Wenn Sie ein Bereitstellungsmanifest verwenden, können Sie ein Website-Archiv für jede erstellen und die Website-Archive dann in einem zweiten ZIP-Archiv mit dem Bereitstellungsmanifest bündeln.

Bereitstellungsmanifeste bieten auch die Möglichkeit, [mehrere Anwendungen auf verschiedenen Pfaden auszuführen \(p. 199\)](#). Ein Bereitstellungsmanifest definiert eine Reihe von Bereitstellungszielen, jeweils mit einem Website-Archiv und einem Pfad, auf dem IIS es ausführen soll. Sie können beispielsweise einen Web-API auf dem /api-Pfad ausführen, um asynchrone Anfragen zu verarbeiten, und eine Webanwendung auf dem Stammpfad ausführen, die die API in Anspruch nimmt.

Sie können ein Bereitstellungsmanifest auch zum [Erstellen von Anwendungspools in IIS \(p. 200\)](#) verwenden, in denen Sie eine oder mehrere Anwendungen ausführen. Sie können einen Anwendungspool so konfigurieren, dass Ihre Anwendungen in regelmäßigen Abständen neu gestartet werden, 32-Bit-Anwendungen ausführen oder eine bestimmte Version der .NET Framework-Laufzeit verwenden.

Für die vollständige Anpassung können Sie in Windows PowerShell [eigene Bereitstellungs-Skripts schreiben \(p. 200\)](#) und Elastic Beanstalk anweisen, welche Skripts zum Installieren, Deinstallieren und Neustarten Ihrer Anwendung verwendet werden sollen.

Bereitstellungsmanifeste und verwandte Funktionen erfordern als Windows Server-Plattformkonfiguration [Version 1.2.0 oder höher \(p. 195\)](#).

Abschnitte

- [.NET Core-Apps \(p. 198\)](#)
- [Ausführen mehrerer Anwendungen \(p. 199\)](#)
- [Konfigurieren der Anwendungspools \(p. 200\)](#)
- [Definieren von benutzerdefinierten Bereitstellungen \(p. 200\)](#)

.NET Core-Apps

Sie können ein Bereitstellungsmanifest für die Ausführung von .NET Core-Anwendungen auf Elastic Beanstalk verwenden. .NET Core ist eine plattformübergreifende Version von .NET mit einem Befehlszeilen-Tool (dotnet), das Sie verwenden können, um eine Anwendung zu generieren, sie lokal auszuführen und für die Veröffentlichung vorzubereiten.

Note

Unter [Tutorial: Bereitstellen einer ASP.NET Core-Anwendung mit Elastic Beanstalk \(p. 208\)](#) finden Sie ein Tutorial sowie eine Beispielanwendung, in der ein Bereitstellungsmanifest zum Ausführen einer .NET Core-Anwendung auf Elastic Beanstalk verwendet wird.

Zum Ausführen einer .NET Core-Anwendung auf Elastic Beanstalk führen Sie `dotnet publish` aus und packen Sie die Ausgabe in ein ZIP-Archiv (ohne enthaltene Verzeichnisse). Platzieren Sie das Website-Archiv in einem Quell-Bundle mit einem Bereitstellungsmanifest mit einem Bereitstellungsziel des Typs `aspNetCoreWeb`.

Das folgende Bereitstellungsmanifest führt eine .NET Core-Anwendung von einem Website-Archiv mit dem Namen `dotnet-core-app.zip` auf dem Stammpfad aus.

Example aws-windows-deployment-manifest.json – .NET Core

```
{  
  "manifestVersion": 1,  
  "deployments": [  
    "aspNetCoreWeb": [  
      {  
        "name": "my-dotnet-core-app",  
        "parameters": {  
          "archive": "dotnet-core-app.zip",  
          "iisPath": "/"  
        }  
      }  
    ]  
  }  
}
```

Bündeln Sie das Manifest und Website-Archiv in einem ZIP-Archiv zum Erstellen eines Quell-Bundle.

Example dotnet-core-bundle.zip

```
.  
|-- aws-windows-deployment-manifest.json  
`-- dotnet-core-app.zip
```

Die Website-Archiv enthält den kompilierten Anwendungscode, Abhängigkeiten und `web.config`-Dateien.

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
```

Ein vollständiges Beispiel finden Sie im [Tutorial \(p. 208\)](#).

Ausführen mehrerer Anwendungen

Sie können mehrere Anwendungen mit einem Bereitstellungsmanifest ausführen, indem Sie mehrere Bereitstellungsziele definieren.

Das folgende Bereitstellungsmanifest konfiguriert zwei .NET Core-Anwendungen. Die `WebAPITest`-Anwendung implementiert ein paar Web-APIs und bedient asynchrone Anfragen im `/api`-Pfad. Die `ASPNetTest`-Anwendung ist eine Webanwendung, die Anfragen im Root-Pfad bedient.

Example `aws-windows-deployment-manifest.json` – mehrere Apps

```
{
  "manifestVersion": 1,
  "deployments": [
    {
      "aspNetCoreWeb": [
        {
          "name": "WebAPITest",
          "parameters": {
            "AppBundle": "webapi.zip",
            "iisPath": "/api"
          }
        },
        {
          "name": "ASPNetTest",
          "parameters": {
            "AppBundle": "aspnet.zip",
            "iisPath": "/"
          }
        }
      ]
    }
  ]
}
```

```
        }
    ]
}
```

Eine Beispielanwendung mit mehreren Anwendungen finden Sie hier:

- Bereitstellbares Quell-Bundle – [dotnet-multiapp-sample-bundle-v2.zip](#)
- Quellcode – [dotnet-multiapp-sample-source-v2.zip](#)

Konfigurieren der Anwendungspools

Sie können ein Bereitstellungsmanifest zum Konfigurieren eines Anwendungspools in IIS und zum Ausführen mehrerer Anwendungen verwenden.

Das folgende Bereitstellungsmanifest konfiguriert einen Anwendungspool, der seine Anwendungen alle zehn Minuten neu startet, und fügt ihn an eine .NET Framework-Webanwendung an, die auf dem Stammpfad ausgeführt wird.

Example aws-windows-deployment-manifest.json – App-Pool

```
{
  "manifestVersion": 1,
  "iisConfig": {
    "appPools": [
      {
        "name": "App pool",
        "recycling": {
          "regularTimeInterval": 10
        }
      }
    ]
  },
  "deployments": {
    "msDeploy": [
      {
        "name": "Web app",
        "parameters": {
          "archive": "site.zip",
          "iisPath": "/",
          "appPool": "MyPool"
        }
      }
    ]
  }
}
```

Der `appPools`-Block unter `iisConfig` definiert den Anwendungspool.

Jede Bereitstellung im `deployments`-Block gibt ein Archiv an, einen Pfad für dessen Ausführung und einen `appPool`, in dem es ausgeführt werden soll.

Definieren von benutzerdefinierten Bereitstellungen

Für noch mehr Kontrolle können Sie eine Anwendungsbereitstellung komplett anpassen, indem Sie eine benutzerdefinierte Bereitstellung definieren.

Das folgende Bereitstellungsmanifest weist Elastic Beanstalk an, ein `install`-Skript mit dem Namen `siteInstall.ps1` auszuführen, um die Website beim Starten der Instance und Bereitstellungen zu

installieren, ein `uninstall`-Skript vor der Installation einer neuen Version während einer Bereitstellung auszuführen und ein `restart`-Skript zum Neustart der Anwendung zu verwenden, wenn Sie [Neustart des App-Servers \(p. 428\)](#) in der Management Console auswählen.

Example aws-windows-deployment-manifest.json – benutzerdefinierte Bereitstellung

```
{  
  "manifestVersion": 1,  
  "deployments": {  
    "custom": [  
      {  
        "name": "Custom site",  
        "scripts": {  
          "install": {  
            "file": "siteInstall.ps1"  
          },  
          "restart": {  
            "file": "siteRestart.ps1"  
          },  
          "uninstall": {  
            "file": "siteUninstall.ps1"  
          }  
        }  
      }  
    ]  
  }  
}
```

Fügen Sie alle Artefakte hinzu, die erforderlich sind, um die Anwendung in Ihrem Quell-Bundle mit dem Manifest und Skripts auszuführen.

Example Custom-site-bundle.zip

```
.  
|--- aws-windows-deployment-manifest.json  
|--- siteInstall.ps1  
|--- siteRestart.ps1  
|--- siteUninstall.ps1  
`-- site-contents.zip
```

Tutorial: Erstellen und Bereitstellen einer .NET-Beispielanwendung mit Elastic Beanstalk.

In diesem Tutorial lernen Sie, wie eine .NET-Beispielanwendung in AWS Elastic Beanstalk mit dem AWS Toolkit for Visual Studio bereitgestellt wird.

Note

Die in diesem Tutorial verwendete ASP.NET-Webanwendung können Sie [hier](#) herunterladen. Darüber hinaus wird auch das [Toolkit for Visual Studio](#) eingesetzt und die Tests erfolgten mit Visual Studio Professional 2012.

Erstellen der Umgebung

Verwenden Sie zunächst den Assistenten zum Erstellen neuer Anwendungen in der Elastic Beanstalk-Konsole, um die Anwendungsumgebung zu erstellen. Wählen Sie für Platform (Plattform) die Option .NET aus.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced.
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus, und wenn Sie bereit sind, wählen Sie Create App (App erstellen) aus.

Wenn die Umgebung gestartet wurde und läuft, fügen Sie eine Amazon RDS-Datenbank-Instance hinzu, die die Anwendung zum Speichern von Daten verwendet. Wählen Sie für DB engine (DB-Engine) sqlserver-ex aus.

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie Apply (Anwenden) aus.

Veröffentlichen Ihrer Anwendung in Elastic Beanstalk

Veröffentlichen Sie Ihre Anwendung mit dem AWS Toolkit for Visual Studio in Elastic Beanstalk.

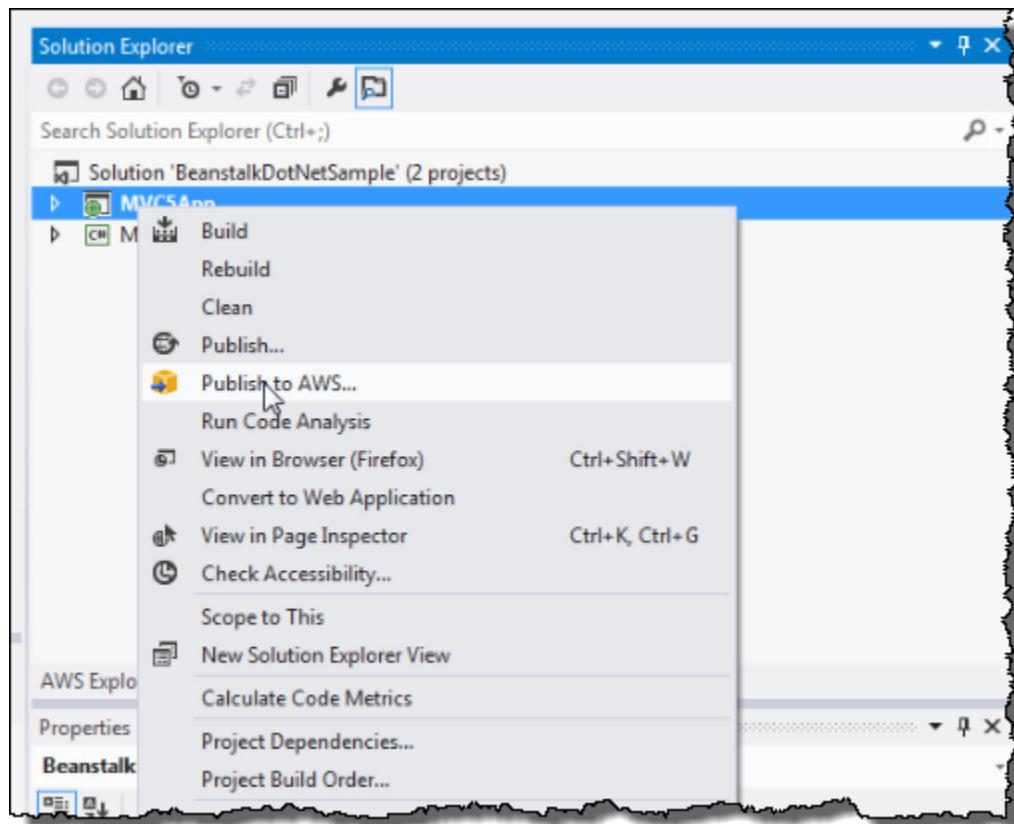
So veröffentlichen Sie Ihre Anwendung in Elastic Beanstalk

1. Vergewissern Sie sich, dass die Umgebung erfolgreich gestartet wurde. Prüfen Sie dazu den Health (Zustand)-Status in der Elastic Beanstalk-Konsole. Es sollte Ok (grün) sein.
2. Öffnen Sie in Visual Studio die Datei BeanstalkDotNetSample.sln.

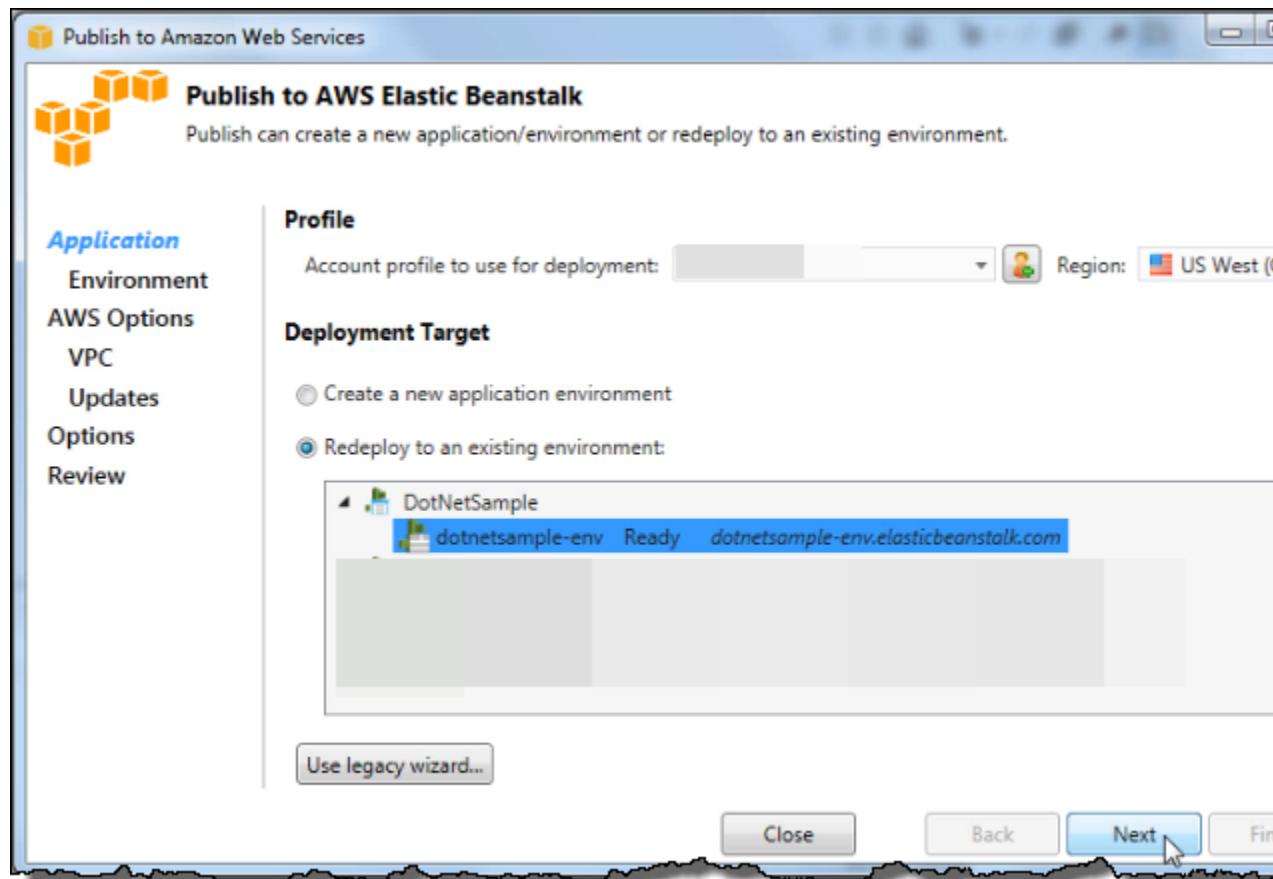
Note

Sie können die Beispielanwendung [hier](#) herunterladen (sofern nicht bereits erfolgt).

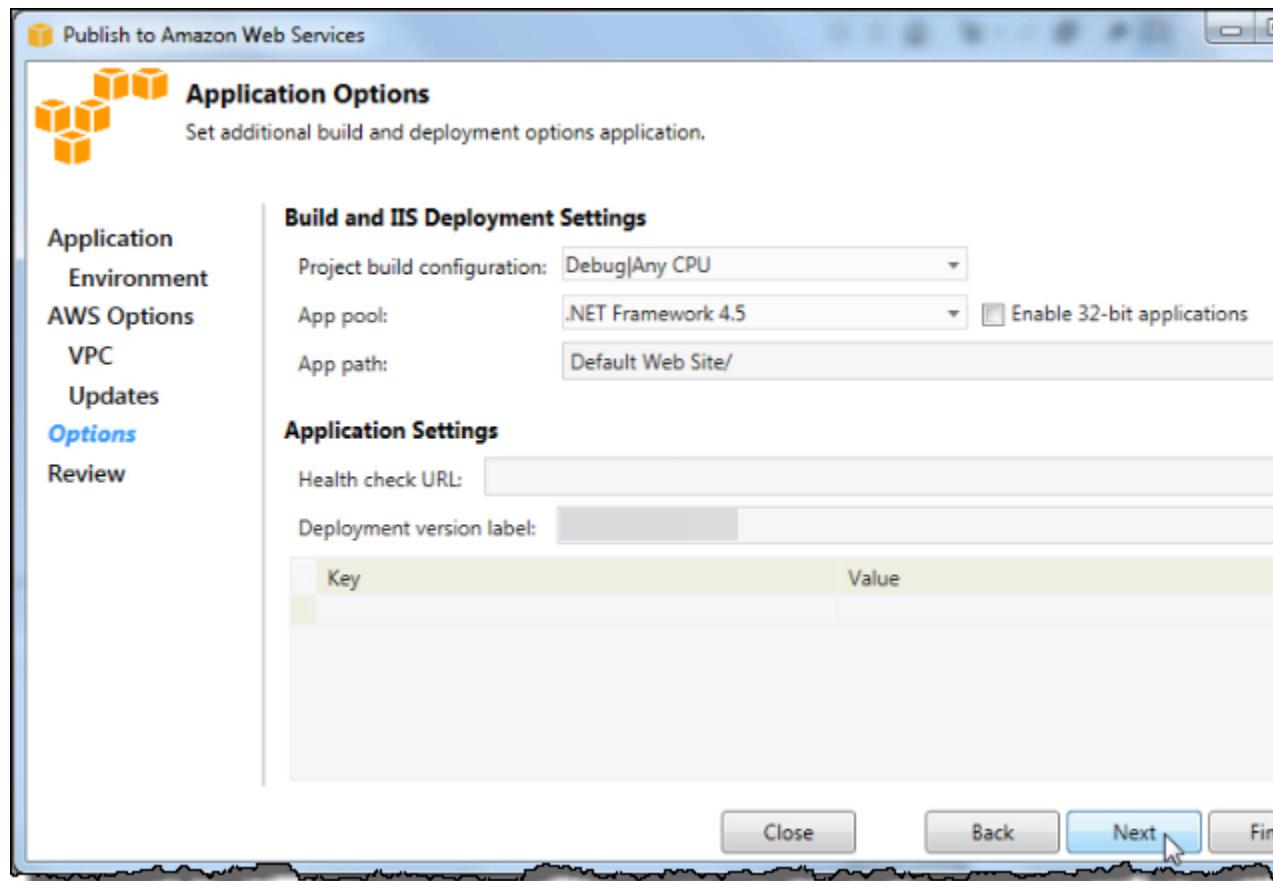
3. Wählen Sie im Menü View (Anzeigen) die Option Solution Explorer (Projektmappen-Explorer) aus.
4. Erweitern Sie Solution 'BeanstalkDotNetSample' (2 projects) (Projektmappe 'BeanstalkDotNetSample' (2 Projekte)).
5. Öffnen Sie das Kontextmenü (rechte Maustaste) für MVC5App und wählen Sie dann Publish to AWS (In AWS veröffentlichen) aus.



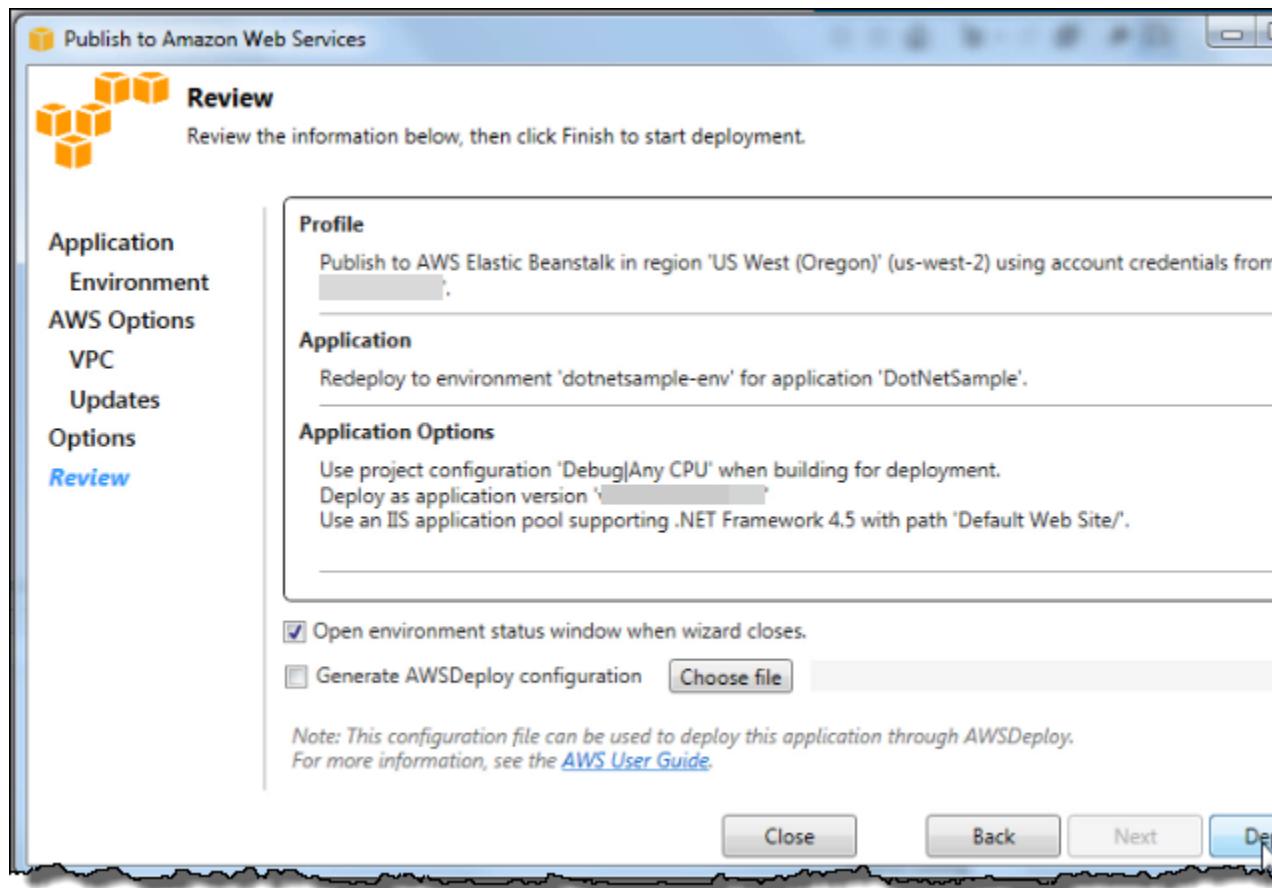
6. Wählen Sie auf der Seite Publish to AWS Elastic Beanstalk (In AWS Elastic Beanstalk veröffentlichen) unter Deployment Target (Bereitstellungsziel) die gerade erstellte Umgebung aus und wählen Sie Next (Weiter) aus.



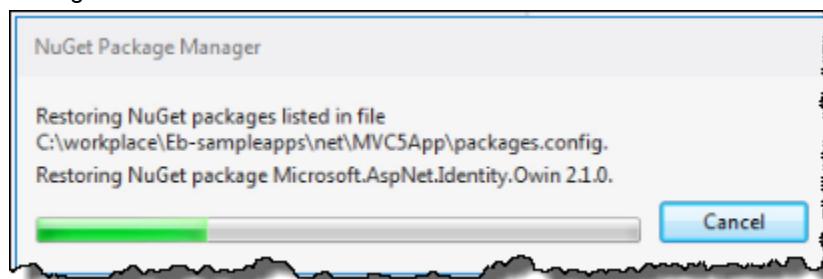
7. Übernehmen Sie auf der Seite Application Options (Anwendungsoptionen) alle Standardeinstellungen und wählen Sie Next (Weiter) aus.



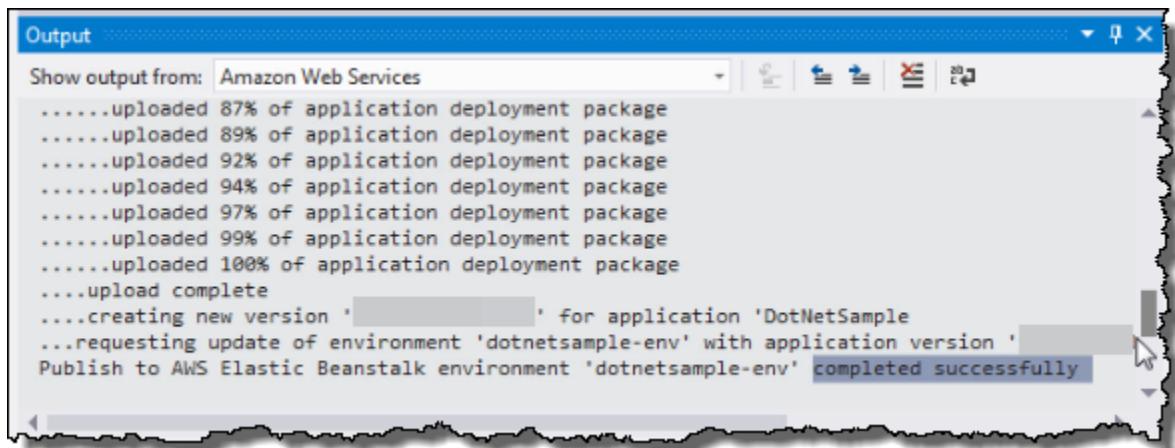
8. Wählen Sie auf der Seite Review (Prüfen) die Option Deploy (Bereitstellen) aus.



9. Falls Sie den Bereitstellungsstatus überwachen möchten, können Sie dazu den NuGet Package Manager in Visual Studio nutzen.



Wenn die Anwendung erfolgreich bereitgestellt wurde, wird im Feld Output (Ausgabe) der Text completed successfully (erfolgreich abgeschlossen) angezeigt.



10. Kehren Sie zur Elastic Beanstalk-Konsole zurück. Wählen Sie im Navigationsbereich Go to environment (Zur Umgebung wechseln).

Die ASP.NET-Anwendung wird in einer neuen Registerkarte geöffnet.

The screenshot shows a web browser displaying an ASP.NET application. The header bar includes links for 'AWS Elastic Beanstalk', 'Home', 'About', and 'Contact'. The main content area features a large heading 'My ASP.NET Application' and a message: 'Congratulations!, Your AWS Elastic Beanstalk ASP.NET application is now running on your own environment in the AWS Cloud.' Below this, there is a link 'In-Depth Introduction'. The background of the page features the 're:Invent' logo and the text 'APP201 Going Zero to Sixty with AWS Elastic'.

Bereinigen Ihrer AWS-Ressourcen

Nach der erfolgreichen Anwendungsbereitstellung können Sie in der Anwendung das [Video ansehen](#), um weitere Informationen zu erhalten.

Falls Sie jetzt nicht länger mit Elastic Beanstalk arbeiten möchten, können Sie die .NET-Umgebung beenden.

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie auf Environment actions (Umgebungsaktionen) und dann auf Terminate Environment (Umgebung beenden).

Elastic Beanstalk bereinigt alle AWS-Ressourcen der Umgebung, darunter z. B. EC2-Instances, DB-Instance, Load Balancer, Sicherheitsgruppen und CloudWatch-Alarme.

Weitere Informationen finden Sie unter [Erstellen und Bereitstellen von .NET-Anwendungen in Elastic Beanstalk \(p. 189\)](#), im [AWS .NET-Entwicklungsblog \(AWS .NET Development Blog\)](#) oder im [AWS-Anwendungsverwaltungsblog \(AWS Application Management Blog\)](#).

Tutorial: Bereitstellen einer ASP.NET Core-Anwendung mit Elastic Beanstalk

In diesem Tutorial werden die Schritte zum Erstellen einer neuen ASP.NET Core-Anwendung und deren Bereitstellung in AWS Elastic Beanstalk beschrieben.

Zuerst verwenden Sie das im .NET Core SDK enthaltene Befehlszeilen-Tool `dotnet`, um eine einfache .NET Core-Befehlszeilenanwendung zu erstellen, Abhängigkeiten zu installieren, Code zu kompilieren und Anwendungen lokal auszuführen. Anschließend erstellen Sie die Standardklasse `Program.cs` und fügen die ASP.NET-Klasse `Startup.cs` sowie Konfigurationsdateien hinzu, um eine Anwendung zur Verarbeitung von HTTP-Anforderungen mit ASP.NET und IIS zu erstellen.

Zum Schluss verwendet Elastic Beanstalk ein [Bereitstellungsmanifest \(p. 197\)](#), um Bereitstellungen für .NET Core-Anwendungen, benutzerdefinierte Anwendungen und mehrere .NET Core- oder MSBuild-Anwendungen auf einem einzigen Server zu konfigurieren. Zum Bereitstellen einer .NET Core-Anwendung in einer Windows Server-Umgebung fügen Sie ein Website-Archiv zum Bereitstellungsmanifest im Quell-Bundle der Anwendung hinzu. Mit dem Befehl `dotnet publish` werden kompilierte Klassen und Abhängigkeiten generiert, die Sie in einer `web.config`-Datei bündeln, um ein Website-Archiv zu erstellen. Das Bereitstellungsmanifest über gibt Elastic Beanstalk den Pfad, unter dem die Website ausgeführt werden soll, und es kann zum Konfigurieren von Anwendungspools und zum Ausführen mehrerer Anwendungen mit verschiedenen Pfaden eingesetzt werden.

Der Quellcode der Anwendung ist hier verfügbar: [dotnet-core-tutorial-source.zip](#)

Das bereitstellbare Source-Bundle ist hier verfügbar: [dotnet-core-tutorial-bundle.zip](#)

Abschnitte

- [Voraussetzungen \(p. 209\)](#)
- [Erstellen eines .NET Core-Projekts \(p. 209\)](#)
- [Starten einer Elastic Beanstalk-Umgebung \(p. 210\)](#)
- [Aktualisieren des Quellcodes \(p. 211\)](#)
- [Bereitstellen Ihrer Anwendung \(p. 215\)](#)
- [Bereinigen \(p. 216\)](#)

- [Nächste Schritte \(p. 217\)](#)

Voraussetzungen

In diesem Tutorial wird das .NET Core SDK verwendet, um eine einfache .NET Core-Anwendung zu generieren, lokal auszuführen und ein Bereitstellungspaket zu erstellen.

Voraussetzungen

- .NET-Core (x64) 1.0.1, 2.0.0 oder höher

So installieren Sie das .NET Core SDK

1. Laden Sie das Installationsprogramm unter microsoft.com/net/core herunter. Wählen Sie Windows aus. Wählen Sie Download .NET SDK (.NET SDK herunterladen) aus.
2. Führen Sie das Installationsprogramm aus und folgen Sie den Anweisungen.

In diesem Tutorial wird ein ZIP-Befehlszeilenprogramm zum Erstellen eines Quell-Bundles verwendet, das Sie in Elastic Beanstalk bereitstellen können. Um den Befehl `zip` in Windows zu verwenden, installieren Sie `UnxUtils`, eine Lightweight-Kollektion nützlicher Befehlszeilenprogramme wie z. B. `zip` und `ls`. Alternativ können Sie den [Windows-Explorer \(p. 414\)](#) oder ein beliebiges anderes ZIP-Programm für die Erstellung von Quell-Bundle-Archiven nutzen.

So installieren Sie UnxUtils

1. Herunterladen [UnxUtils](#).
2. Extrahieren Sie das Archiv in ein lokales Verzeichnis. Beispiel, `C:\Program Files (x86)`
3. Fügen Sie den Pfad zu den binären Dateien zu Ihrer Windows PATH-Benutzervariable hinzu. Beispiel, `C:\Program Files (x86)\UnxUtils\usr\local\wbin`.
 - a. Betätigen Sie die Windows-Taste und geben Sie anschließend **environment variables** ein.
 - b. Wählen Sie Edit environment variables for your account (Umgebungsvariablen für Ihr Konto bearbeiten).
 - c. Wählen Sie PATH (PFAD) und Edit (Bearbeiten) aus.
 - d. Fügen Sie im Feld Variable value (Variablenwert) Pfade hinzu, getrennt durch Semikola. Beispiel: `C:\item1\path;C:\item2\path`
 - e. Klicken Sie zweimal auf OK, um die neuen Einstellungen anzuwenden.
 - f. Schließen Sie alle laufenden Eingabeaufforderungsfenster und öffnen Sie dann erneut ein Eingabeaufforderungsfenster.
4. Öffnen Sie ein neues Eingabeaufforderungsfenster und führen Sie den `zip`-Befehl aus, um sicherzustellen, dass es funktioniert.

```
> zip -h
Copyright (C) 1990-1999 Info-ZIP
Type 'zip "-L"' for software license.
...
```

Erstellen eines .NET Core-Projekts

Verwenden Sie das Befehlszeilen-Tool `dotnet`, um ein neues C# .NET Core-Projekt zu erstellen und lokal auszuführen. Die .NET Core-Standardanwendung ist ein Befehlszeilenprogramm, das `Hello World!` ausgibt und dann beendet ist.

So erstellen Sie ein neues .NET Core-Projekt

1. Öffnen Sie ein neues Befehlszeilenfenster und navigieren Sie zu Ihrem Benutzerordner.

```
> cd %USERPROFILE%
```

2. Erstellen Sie mit dem Befehl `dotnet new` ein neues .NET Core-Projekt.

```
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. Verwenden Sie den Befehl `dotnet restore` zum Installieren von Abhängigkeiten.

```
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. Verwenden Sie den Befehl `dotnet run`, um die Anwendung zu erstellen und lokal auszuführen.

```
C:\Users\username\dotnet-core-tutorial> dotnet run
Hello World!
```

Starten einer Elastic Beanstalk-Umgebung

Verwenden Sie die Elastic Beanstalk-Konsole, um eine Elastic Beanstalk-Umgebung zu starten. Für dieses Beispiel starten Sie mit einer .NET-Plattform. Nachdem Sie die Umgebung gestartet und konfiguriert haben, können Sie jederzeit neuen Quellcode bereitstellen.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced.
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus, und wenn Sie bereit sind, wählen Sie Create App (App erstellen) aus.

Das Erstellen einer Umgebung dauert ca. 10 Minuten. Währenddessen können Sie den Quellcode aktualisieren.

Aktualisieren des Quellcodes

Ändern Sie die Standardanwendung in eine Webanwendung, die ASP.NET und IIS verwendet.

- ASP.NET ist das Website-Framework für .NET.
- IIS ist der Webserver, der die Anwendung auf den Amazon EC2-Instances in Ihrer Elastic Beanstalk-Umgebung ausführt.

Die Quellcodebeispiele, die Sie nachvollziehen können, finden Sie hier: [dotnet-core-tutorial-source.zip](#)

Note

Das folgende Verfahren zeigt die Konvertierung des Projektcodes in eine Webanwendung. Um den Vorgang zu vereinfachen, können Sie das Projekt direkt als Webanwendung generieren. Modifizieren Sie im vorangehenden Abschnitt [Erstellen eines .NET Core-Projekts \(p. 209\)](#) den Befehl `dotnet new` des Schritts mit dem folgenden Befehl.

```
C:\Users\username> dotnet new web -o dotnet-core-tutorial
```

So fügen Sie ASP.NET- und IIS-Unterstützung zum Code hinzu

1. Kopieren Sie `Program.cs` in Ihr Anwendungsverzeichnis, um es als Webhost-Builder auszuführen

Example c:\users\username\dotnet-core-tutorial\Program.cs

```
using System;
using Microsoft.AspNetCore.Hosting;
using System.IO;

namespace aspnetcoreapp
{
    public class Program
    {
        public static void Main(string[] args)
        {
            var host = new WebHostBuilder()
                .UseKestrel()
                .UseContentRoot(Directory.GetCurrentDirectory())
                .UseIISIntegration()
                .UseStartup<Startup>()
                .Build();

            host.Run();
        }
    }
}
```

2. Fügen Sie eine `Startup.cs`-Datei zur Ausführung einer ASP.NET-Website hinzu.

Example c:\users\username\dotnet-core-tutorial\Startup.cs

```
using System;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Http;
```

```
namespace aspnetcoreapp
{
    public class Startup
    {
        public void Configure(IApplicationBuilder app)
        {
            app.Run(context =>
            {
                return context.Response.WriteAsync("Hello from ASP.NET Core!");
            });
        }
    }
}
```

3. Fügen Sie die web.config-Datei zum Konfigurieren des IIS-Servers hinzu.

Example c:\users\username\dotnet-core-tutorial\web.config

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <system.webServer>
        <handlers>
            <add name="aspNetCore" path="*" verb="*" modules="AspNetCoreModule"
resourceType="Unspecified" />
        </handlers>
        <aspNetCore processPath="dotnet" arguments=".\\dotnet-core-
tutorial.dll" stdoutLogEnabled="false" stdoutLogFile=".\\logs\\stdout"
forwardWindowsAuthToken="false" />
    </system.webServer>
</configuration>
```

4. Fügen Sie dotnet-core-tutorial.csproj hinzu, die IIS-Middleware enthält, ebenso wie die web.config-Datei aus der Ausgabe von dotnet publish.

Note

Das folgende Beispiel wurde mittels .NET Core Runtime 2.2.1 entwickelt. Möglicherweise müssen Sie die Attributwerte für TargetFramework oder Version in den PackageReference-Elementen ändern, damit sie mit der .NET Core Runtime-Version übereinstimmen, die Sie in Ihren benutzerdefinierten Projekten verwenden.

Example c:\users\username\dotnet-core-tutorial\dotnet-core-tutorial.csproj

```
<Project Sdk="Microsoft.NET.Sdk">

    <PropertyGroup>
        <OutputType>Exe</OutputType>
        <TargetFramework>netcoreapp2.2</TargetFramework>
    </PropertyGroup>

    <ItemGroup>
        <PackageReference Include="Microsoft.AspNetCore.Server.Kestrel"
Version="2.2.0" />
    </ItemGroup>

    <ItemGroup>
        <PackageReference Include="Microsoft.AspNetCore.Server.IISIntegration"
Version="2.2.0" />
    </ItemGroup>

    <ItemGroup>
        <None Include="web.config" CopyToPublishDirectory="Always" />
    </ItemGroup>
```

```
</Project>
```

Im nächsten Schritt installieren Sie die neuen Abhängigkeiten und führen die ASP.NET-Website lokal aus.

So führen Sie die Website lokal aus

1. Verwenden Sie den Befehl `dotnet restore` zum Installieren von Abhängigkeiten.
2. Verwenden Sie den Befehl `dotnet run`, um die App zu erstellen und lokal auszuführen.
3. Öffnen Sie localhost:5000, um die Website anzuzeigen.

Um die Anwendung auf einem Webserver auszuführen, muss der kompilierte Quellcode mit Laufzeitabhängigkeiten in einer `web.config`-Konfigurationsdatei gebündelt werden. Das Tool `dotnet` bietet einen `publish`-Befehl, mit dem diese Dateien basierend auf der Konfiguration in `dotnet-core-tutorial.csproj` in einem Verzeichnis gesammelt werden.

So erstellen Sie die Website

- Mit dem Befehl `dotnet publish` geben Sie kompilierten Code und Abhängigkeiten in einem Ordner mit dem Namen `site` aus.

```
C:\users\username\dotnet-core-tutorial> dotnet publish -o site
```

Für die Bereitstellung der Anwendung in Elastic Beanstalk bündeln Sie das Website-Archiv mit einem [Bereitstellungsmanifest \(p. 197\)](#). Anhand dessen erkennt Elastic Beanstalk, wie die Ausführung erfolgen muss.

So erstellen Sie ein Quell-Bundle

1. Fügen Sie die Dateien im Website-Ordner zu einem ZIP-Archiv hinzu.

Note

Wenn Sie ein anderes ZIP-Dienstprogramm verwenden, müssen Sie alle Dateien dem Stammordner des resultierenden ZIP-Archivs hinzufügen. Dies ist für eine erfolgreiche Bereitstellung der Anwendung in Ihrer Elastic Beanstalk-Umgebung erforderlich.

```
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. Fügen Sie ein Bereitstellungsmanifest hinzu, das auf das Website-Archiv verweist.

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"
        }
      }
    ]
  }
}
```

```
        }
    ]
}
```

3. Verwenden Sie den Befehl `zip` zum Erstellen eines Quell-Bundles mit dem Namen `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%)
```

Bereitstellen Ihrer Anwendung

Stellen Sie das Quell-Bundle in der von Ihnen erstellten Elastic Beanstalk-Umgebung bereit.

Sie können das Quell-Bundle hier herunterladen: [dotnet-core-tutorial-bundle.zip](#)

So stellen Sie ein Quell-Bundle bereit

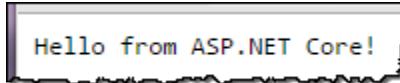
1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Die Anwendung schreibt einfach `Hello from ASP.NET Core!` als Antwort und gibt diese zurück.



Beim Starten einer Umgebung werden die folgenden Ressourcen erstellt:

- EC2-Instance – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2) zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattformen.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance-Sicherheitsgruppe – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom

Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.

- Load Balancer – Ein Elastic Load Balancing-Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load Balancer-Sicherheitsgruppe – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling-Gruppe – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch-Alarme – Zwei CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Beim Auslösen eines Alarms wird Ihre Auto Scaling-Gruppe nach oben oder unten skaliert.
- AWS CloudFormation Stack – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domänenname – Ein Domänenname, der auf Ihre Webanwendung in der Form `subdomain.region.elasticbeanstalk.com` weiterleitet.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Note

Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#).

Bereinigen

Wenn Sie nicht mehr mit Elastic Beanstalk arbeiten, können Sie Ihre Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 554\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie auf Environment actions (Umgebungsaktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Nächste Schritte

Im weiteren Verlauf der Anwendungsentwicklung möchten Sie vermutlich Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die [Elastic Beanstalk Command Line Interface \(p. 1008\)](#) (EB CLI) stellt benutzerfreundliche Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk-Umgebungen über die Befehlszeile bereit.

Wenn Sie Visual Studio zur Entwicklung Ihrer Anwendung verwenden, können Sie das AWS Toolkit for Visual Studio auch verwenden, um Änderungen bereitzustellen, Ihre Elastic Beanstalk-Umgebungen zu verwalten und andere AWS-Ressourcen zu verwalten. Weitere Informationen finden Sie unter [Das AWS Toolkit for Visual Studio \(p. 220\)](#).

Für die Entwicklung sowie für Tests können Sie die Elastic Beanstalk-Funktionalität nutzen und so eine verwaltete DB-Instance direkt in der Umgebung hinzufügen. Detaillierte Anweisungen zum Einrichten einer Datenbank in der Umgebung finden Sie unter [Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung \(p. 609\)](#).

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, [konfigurieren Sie einen benutzerdefinierten Domänennamen \(p. 644\)](#) für die Umgebung und [aktivieren Sie HTTPS \(p. 778\)](#) für sichere Verbindungen.

Hinzufügen einer Amazon RDS-DB-Instance zur .NET-Anwendungsumgebung

Sie können eine DB-Instance von Amazon Relational Database Service (Amazon RDS) verwenden, um Daten zu speichern, die von Ihrer Anwendung gesammelt und geändert wurden. Die Datenbank kann Ihrer Umgebung angefügt und von Elastic Beanstalk verwaltet werden oder sie kann extern erstellt und verwaltet werden.

Bei der ersten Verwendung von Amazon RDS fügen Sie in der Testumgebung mithilfe der Elastic Beanstalk Management Console eine [DB-Instance \(p. 218\)](#) hinzu und überprüfen, ob die Anwendung eine Verbindung zu dieser herstellen kann.

Für die Verbindungserstellung zu einer Datenbank [fügen Sie den Treiber \(p. 219\)](#) zur Anwendung hinzu, laden Sie die Treiberklasse in den Code und [erstellen Sie eine Verbindungszeichenfolge \(p. 219\)](#) mit den von Elastic Beanstalk bereitgestellten Umgebungseigenschaften. Die Konfiguration und der Verbindungscode hängen von der Datenbank-Engine und dem Framework ab, die Sie nutzen.

Note

Für Lernzwecke oder Testumgebungen können Sie Elastic Beanstalk verwenden, um eine DB-Instance hinzuzufügen.

Für Produktionsumgebungen können Sie eine DB-Instance außerhalb Ihrer Elastic Beanstalk-Umgebung erstellen, um Ihre Umgebungsressourcen von Ihren Datenbankressourcen abzukoppeln. Auf diese Weise wird beim Beenden Ihrer Umgebung die DB-Instance nicht gelöscht. Mit einer externen DB-Instance können Sie von mehreren Umgebungen aus eine Verbindung mit derselben Datenbank herstellen und [Blau/Grün-Bereitstellungen](#) durchführen. Detaillierte Anweisungen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon RDS \(p. 973\)](#).

Abschnitte

- [Hinzufügen einer DB-Instance zu Ihrer Umgebung \(p. 218\)](#)
- [Herunterladen eines Treibers \(p. 219\)](#)
- [Verbinden mit einer Datenbank \(p. 219\)](#)

Hinzufügen einer DB-Instance zu Ihrer Umgebung

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie Apply (Anwenden) aus.

Das Hinzufügen einer DB-Instance dauert ca. 10 Minuten. Sobald die Umgebungsaktualisierung abgeschlossen ist, stehen der Hostname der DB-Instance und andere Verbindungsinformationen über die folgenden Umgebungseigenschaften zur Verfügung:

Eigenschaftenname	Beschreibung	Eigenschaftswert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbankname, ebdb	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: DB Name (DB-Name).
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	In der Amazon RDS-Konsole nicht als Referenz verfügbar.

Weitere Informationen zum Konfigurieren einer internen DB-Instance finden Sie unter [Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung \(p. 609\)](#).

Herunterladen eines Treibers

Laden Sie das EntityFramework-Paket und einen Datenbanktreiber für Ihre Entwicklungsumgebung mit NuGet herunter und installieren sie es bzw. ihn.

Gängige Entitäts-Framework-Datenbankanbieter für .NET

- SQL Server – Microsoft.EntityFrameworkCore.SqlServer
- MySQL – Pomelo.EntityFrameworkCore.MySql
- PostgreSQL – Npgsql.EntityFrameworkCore.PostgreSQL

Verbinden mit einer Datenbank

Elastic Beanstalk stellt Verbindungsinformationen für zugehörige DB-Instances in Umgebungseigenschaften bereit. Mit ConfigurationManager.AppSettings können Sie die Eigenschaften lesen und eine Datenbankverbindung konfigurieren.

Example Helpers.cs – Verbindungszeichenfolgen-Methode

```
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 + ";";
        }
    }
}
```

Verwenden Sie die Verbindungszeichenfolge, um Ihren Datenbankkontext zu initialisieren.

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(GetRDSCConnectionString())  
        {}  
  
        public static RDSContext Create()  
        {  
            return new RDSContext();  
        }  
    }  
}
```

Das AWS Toolkit for Visual Studio

Visual Studio bietet Vorlagen für verschiedene Programmiersprachen und Anwendungstypen. Sie können mit einer beliebigen Vorlage beginnen. Das AWS Toolkit for Visual Studio enthält zudem drei Projektvorlagen für den Bootstrap der Anwendungsentwicklung: das AWS-Konsolenprojekt, das AWS-Webprojekt und ein leeres AWS-Projekt. In diesem Beispiel erstellen Sie eine neue ASP.NET-Webanwendung.

So erstellen Sie ein neues ASP.NET-Webanwendungsprojekt:

1. Klicken Sie in Visual Studio im Menü Datei auf Neu und dann auf Projekt.
2. Klicken Sie im Dialogfeld Neues Projekt auf Installierte Vorlagen, auf Visual C# und auf Web. Klicken Sie auf Leere ASP.NET-Webanwendung, geben Sie einen Projektnamen ein und klicken Sie auf OK.

So führen Sie ein Projekt aus

Führen Sie eine der folgenden Aufgaben aus:

1. Drücken Sie F5.
2. Wählen Sie im Menü Debuggen die Option Debuggen starten aus.

Ausführen lokaler Tests

Mit Visual Studio können Sie die Anwendung einfach lokal testen. Zum Testen oder Ausführen von ASP.NET-Webanwendungen ist ein Webserver erforderlich. Visual Studio bietet mehrere Optionen, z. B. Internet Information Services (IIS), IIS Express oder den integrierten Visual Studio Development Server. Für weitere Informationen zu den einzelnen Optionen und zur Auswahl der am besten für Sie geeigneten Option rufen Sie [Webserver in Visual Studio für ASP.NET-Webprojekte](#) auf.

Erstellen einer Elastic Beanstalk-Umgebung

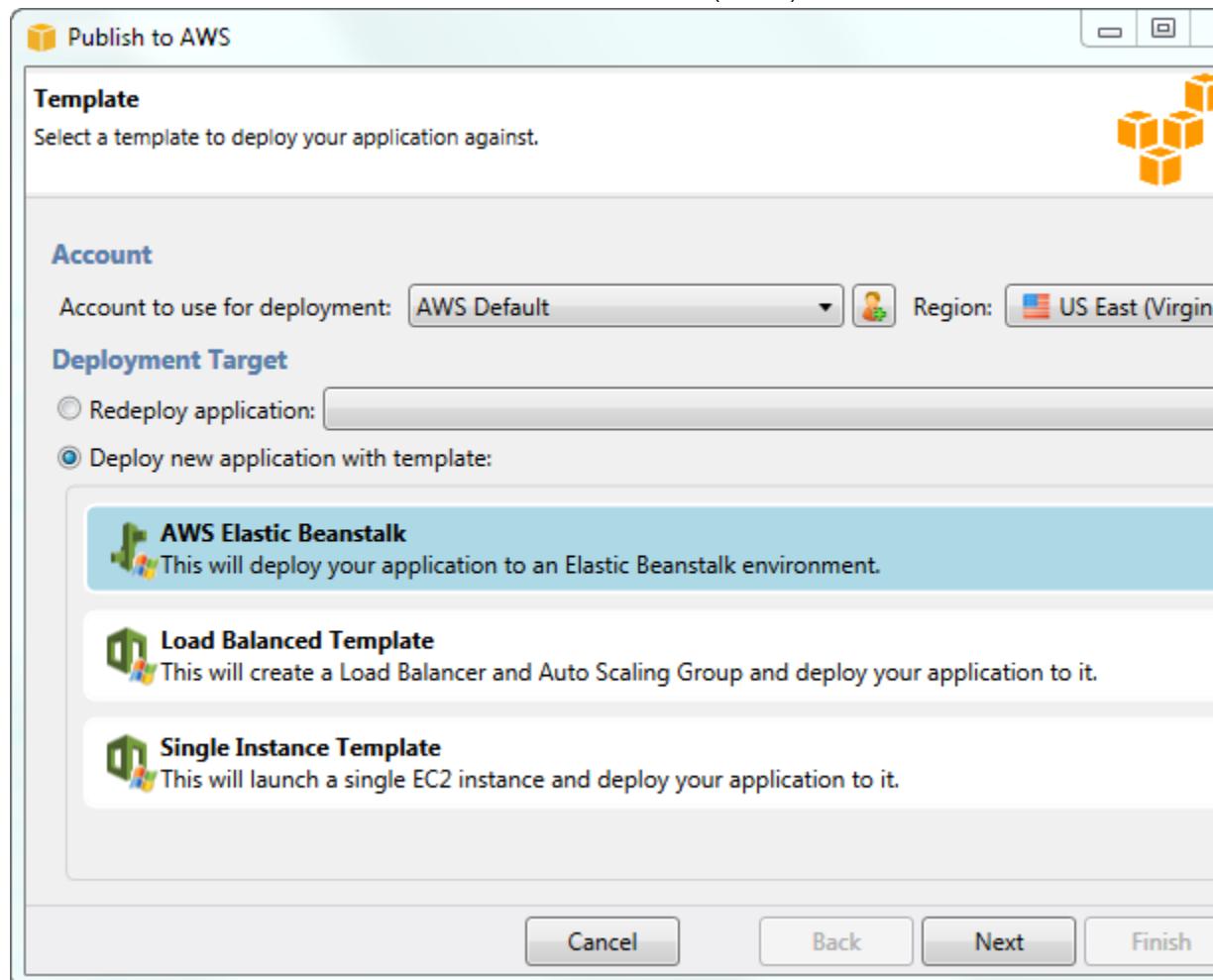
Nachdem Sie die Anwendung getestet haben, können Sie diese nun in Elastic Beanstalk bereitstellen.

Note

Die [Konfigurationsdatei \(p. 722\)](#) muss Teil des Projekts sein und in das Archiv eingebunden werden. Anstatt die Konfigurationsdateien in das Projekt einzubinden, können Sie mit Visual Studio alternativ alle Dateien im Projektordner bereitstellen. Klicken Sie im Projektmappen-Explorer mit der rechten Maustaste auf den Projektnamen und klicken Sie dann auf Eigenschaften. Klicken Sie auf die Registerkarte Web packen/veröffentlichen. Wählen Sie unter Bereitzustellende Elemente in der Dropdown-Liste die Option Alle Dateien in diesem Projektordner aus.

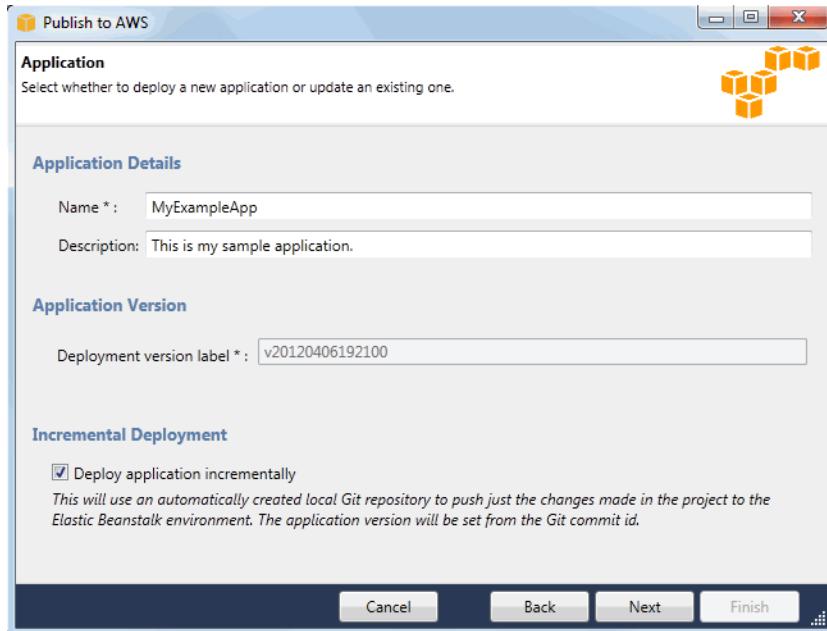
Stellen Sie Ihre Anwendung mit dem AWS Toolkit for Visual Studio in Elastic Beanstalk bereit.

1. Klicken Sie im Solution Explorer (Projektmappen-Explorer) mit der rechten Maustaste auf die Anwendung und wählen Sie Publish to AWS (In AWS veröffentlichen) aus.
2. Geben Sie im Assistenten Publish to AWS (In AWS veröffentlichen) Ihre Kontoinformationen ein.
 - a. Wählen Sie unter AWS account to use for deployment (AWS-Konto für Bereitstellung) Ihr Konto aus. Alternativ wählen Sie Other (Anderes) aus, um neue Kontoinformationen anzugeben.
 - b. Wählen Sie unter Region die Region aus, in der die Anwendung bereitgestellt werden soll. Informationen zu verfügbaren AWS-Regionen finden Sie unter [AWS Elastic Beanstalk Endpunkte und Kontingente](#) in der Allgemeinen AWS-Referenz. Wenn Sie eine Region auswählen, die nicht von Elastic Beanstalk unterstützt wird, ist die Option zur Bereitstellung in Elastic Beanstalk nicht verfügbar.
 - c. Klicken Sie auf Deploy new application with template (Neue Anwendung mit Vorlage bereitstellen) und wählen Sie Elastic Beanstalk aus. Klicken Sie dann auf Next (Weiter).



3. Geben Sie auf der Seite Application (Anwendung) die Anwendungsdetails an.
 - a. Geben Sie im Feld Name den Namen der Anwendung ein.
 - b. Geben Sie im Feld Description eine Beschreibung der Anwendung ein. Dieser Schritt ist optional.
 - c. Die Versionsbezeichnung der Anwendung wird automatisch im Feld Deployment version label (Versionsbezeichnung der Bereitstellung) angezeigt.

- d. Wählen Sie Deploy application incrementally (Anwendung inkrementell bereitstellen) aus, wenn nur die geänderten Dateien bereitgestellt werden sollen. Eine inkrementelle Bereitstellung ist schneller, da anstelle von allen Dateien nur solche aktualisiert werden, die geändert wurden. Bei Auswahl dieser Option wird die Anwendungsversion über die Git-Commit-ID festgelegt. Soll die Anwendung nicht inkrementell bereitgestellt werden, können Sie die Versionsbezeichnung im Feld Deployment version label (Versionsbezeichnung der Bereitstellung) aktualisieren.



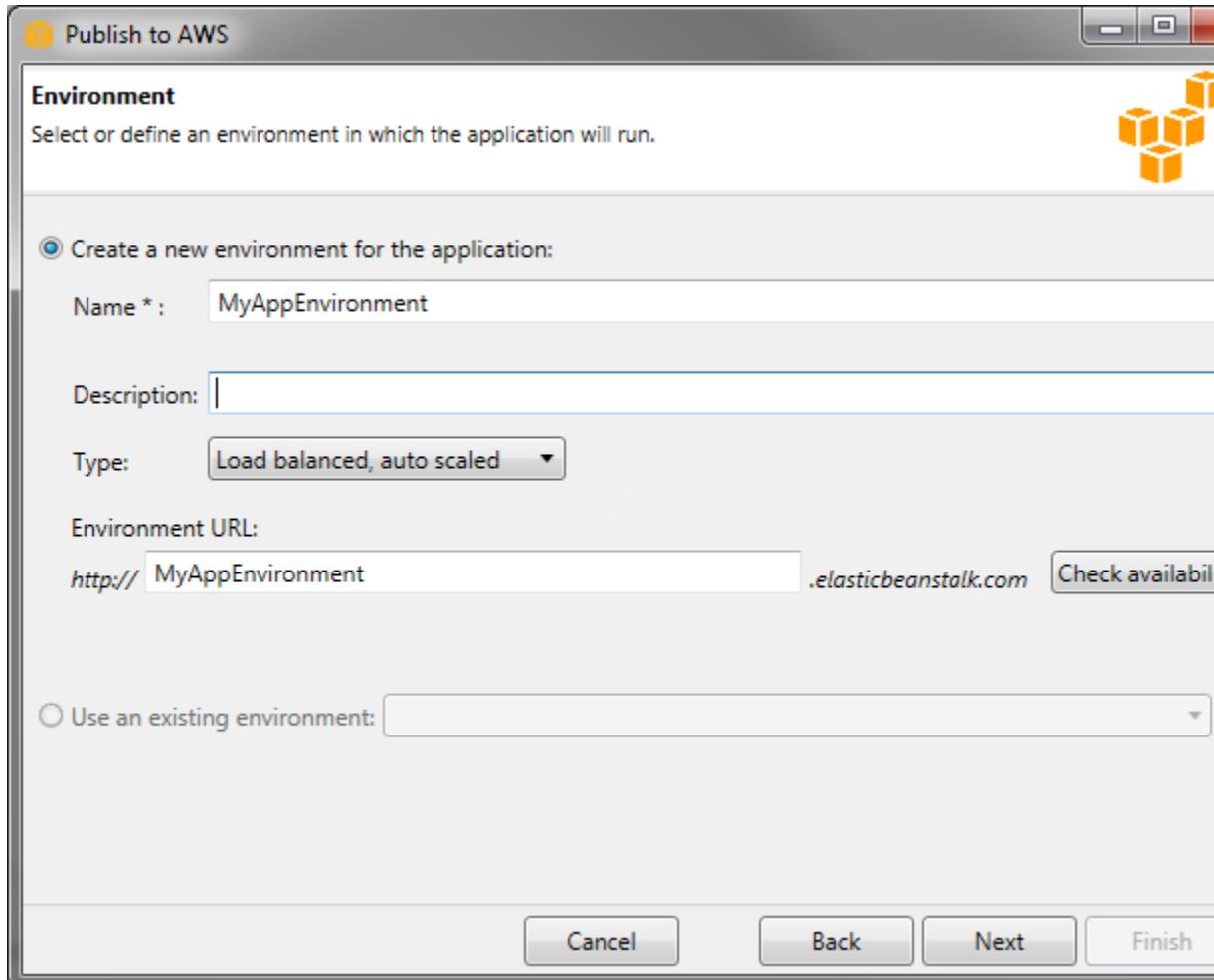
- e. Klicken Sie auf Weiter.
4. Beschreiben Sie auf der Seite Environment (Umgebung) die Details der Umgebung.
- Wählen Sie Create a new environment for this application (Neue Umgebung für diese Anwendung erstellen) aus.
 - Geben Sie unter Name einen Namen für die Umgebung ein.
 - Geben Sie im Feld Description (Beschreibung) eine Beschreibung der Umgebung ein. Dieser Schritt ist optional.
 - Wählen Sie unter Type (Typ) den gewünschten Umgebungstyp aus.

Sie können als Umgebungstyp entweder Load balanced, auto scaled (lastenverteilt, automatisch skaliert) oder Single instance (Einzelne Instance) auswählen. Weitere Informationen finden Sie unter [Umgebungstypen \(p. 514\)](#).

Note

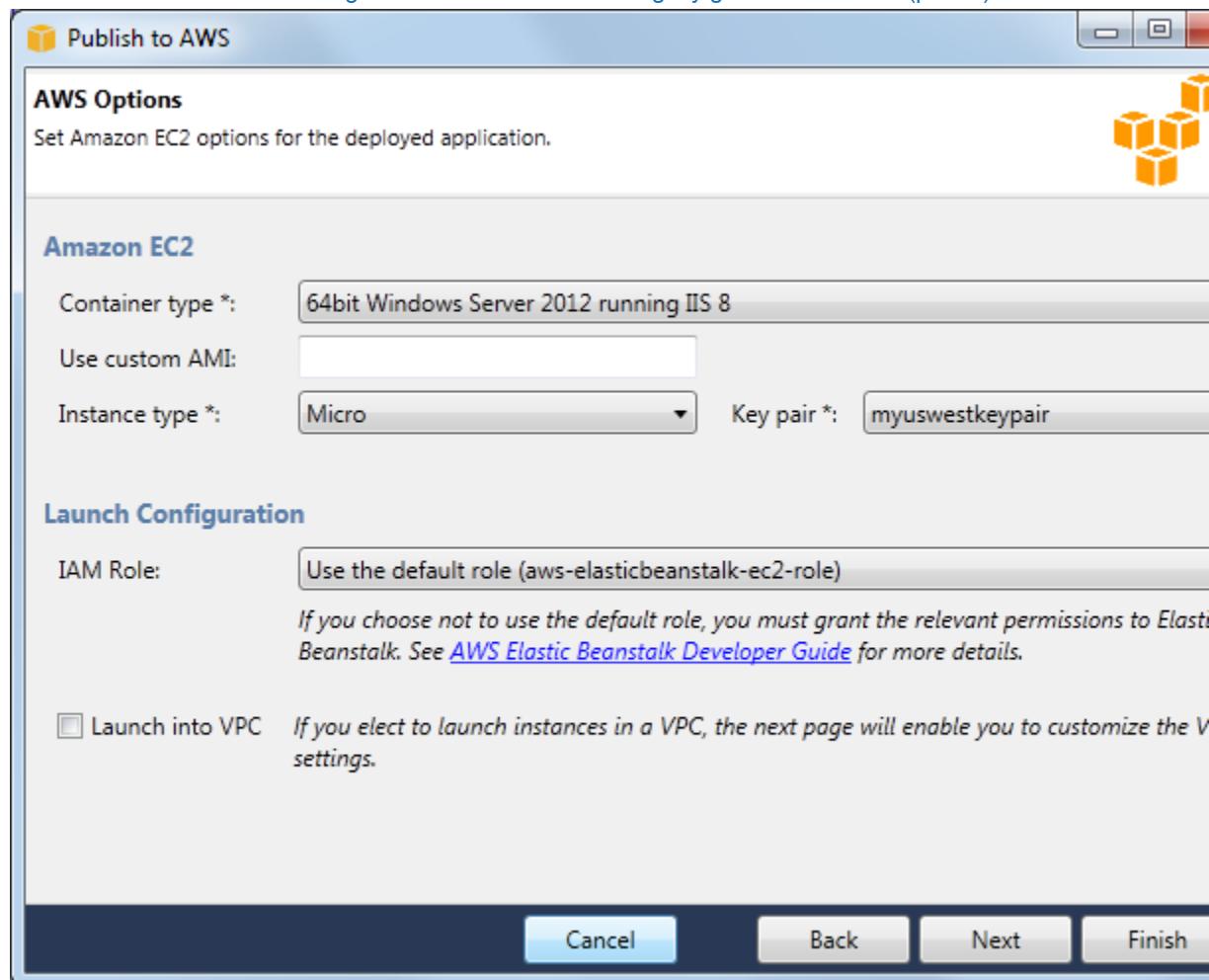
Die Einstellungen für Lastausgleich, Auto Scaling und Integritätsprüfungs-URL können nicht auf Single-Instance-Umgebungen angewendet werden.

- Im Feld Environment URL (Umgebungs-URL) wird die Umgebungs-URL automatisch angezeigt, sobald Sie den Mauszeiger zu diesem Feld bewegen.
- Klicken Sie auf Check availability (Verfügbarkeit prüfen), um sicherzustellen, dass die URL verfügbar ist.

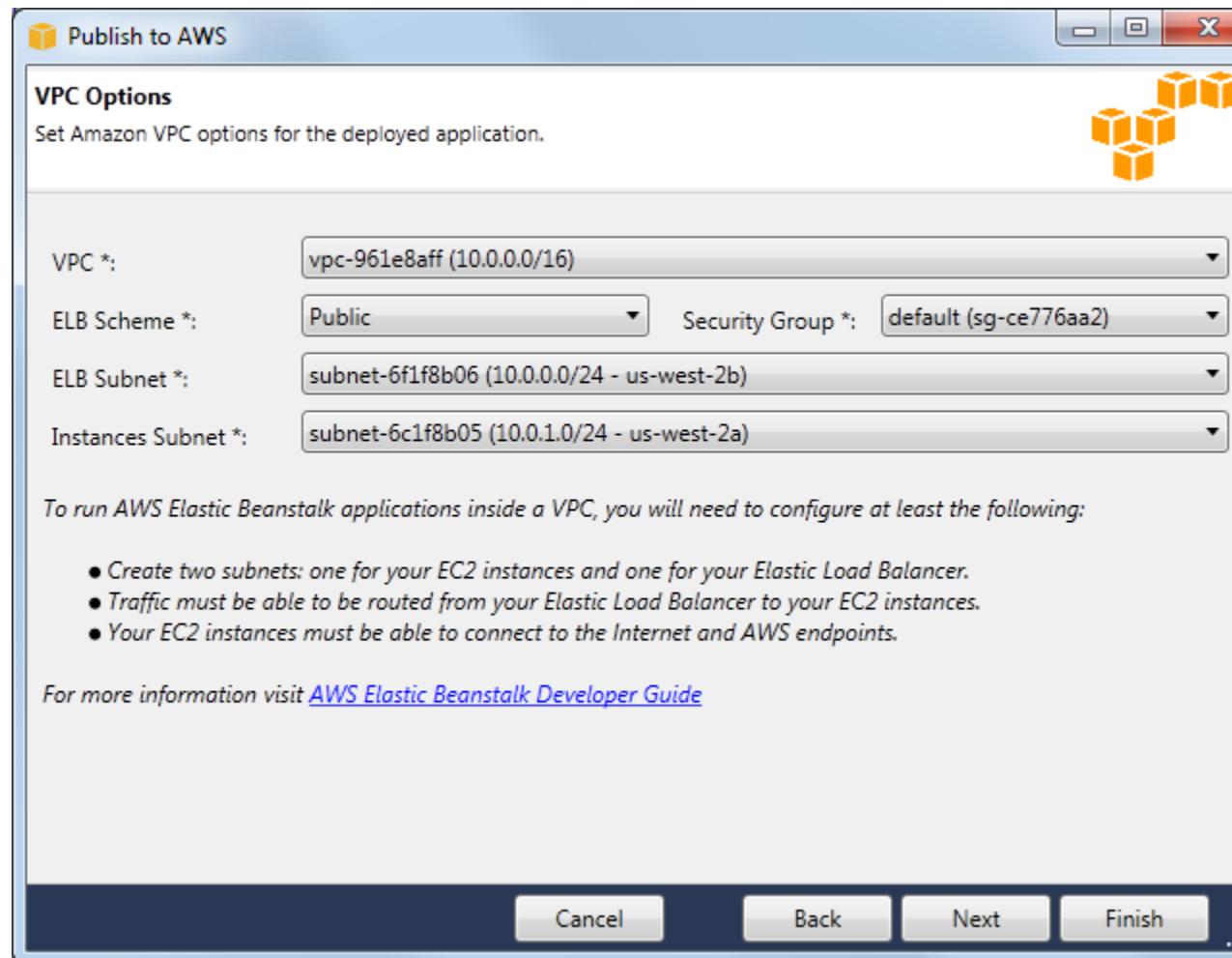


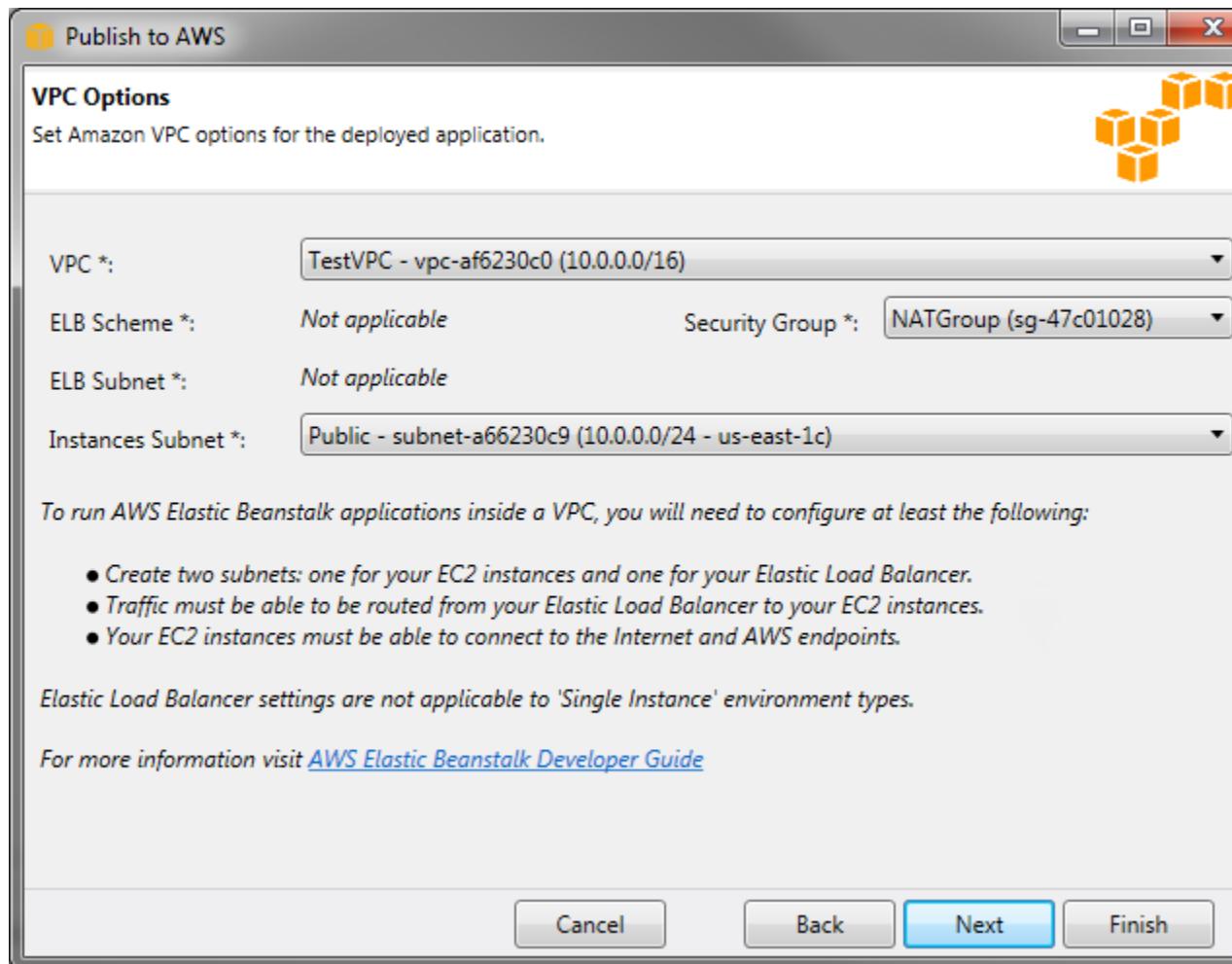
- g. Klicken Sie auf Weiter.
5. Konfigurieren Sie auf der Seite AWS Options (AWS-Optionen) zusätzliche Optionen und Sicherheitsinformationen für die Bereitstellung.
 - a. Wählen Sie im Feld Container Type (Container-Typ) den Eintrag 64bit Windows Server 2012 running IIS 8 (Windows Server 2012 mit 64 Bit und IIS 8) oder 64bit Windows Server 2008 running IIS 7.5 (Windows Server 2008 mit 64 Bit und IIS 7.5) aus.
 - b. Wählen Sie im Feld Instance Type (Instance-Typ) die Option Micro aus.
 - c. Wählen Sie im Feld Key pair (Schlüsselpaar) die Option Create new key pair (Neues Schlüsselpaar erstellen) aus. Geben Sie einen Namen für das neue Schlüsselpaar ein – in diesem Beispiel wird **myuswestkeypair** verwendet – und klicken Sie auf OK. Ein Schlüsselpaar ermöglicht den Remote-Desktop-Zugriff auf Ihre Amazon EC2-Instances. Weitere Informationen zu Amazon EC2-Schlüsselpaaren finden Sie unter [Using Credentials \(Verwenden von Anmeldeinformationen\)](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.
 - d. Wählen Sie ein Instance-Profil aus.
- Falls Sie nicht über ein Instance-Profil verfügen, wählen Sie Instance-Standardprofil erstellen aus. Weitere Informationen zum Verwenden von Instance-Profilen in Elastic Beanstalk finden Sie unter [Elastic Beanstalk Instance-Profile verwalten \(p. 904\)](#).
- e. Wenn Sie eine benutzerdefinierte VPC mit der Umgebung nutzen möchten, klicken Sie auf Launch into VPC (In VPC starten). Sie können die VPC-Informationen auf der nächsten Seite konfigurieren. Weitere Informationen zu Amazon VPC finden Sie unter [Amazon Virtual Private](#)

Cloud (Amazon VPC). Eine Liste mit unterstützten aktuellen Containertypen finden Sie unter [the section called "Warum sind einige Plattformversionen als Legacy gekennzeichnet?" \(p. 502\)](#).



- f. Klicken Sie auf Weiter.
6. Wenn Sie ausgewählt haben, dass die Umgebung in einer VPC gestartet werden soll, wird die Seite VPC Options (VPC-Optionen) eingeblendet. Andernfalls wird die Seite Additional Options (Weitere Optionen) angezeigt. Hier können Sie die VPC-Optionen konfigurieren.





- a. Wählen Sie die VPC-ID der VPC aus, in der die Umgebung gestartet werden soll.
- b. Für eine skalierbare Umgebung mit Lastausgleich wählen Sie im Feld ELB Scheme (ELB-Schema) die Option private (privat) aus, wenn der Elastic Load Balancer nicht für das Internet verfügbar sein soll.

Bei einer Umgebung mit einer einzelnen Instance ist diese Option nicht auswählbar, da kein Load Balancer vorhanden ist. Weitere Informationen finden Sie unter [Umgebungstypen \(p. 514\)](#).

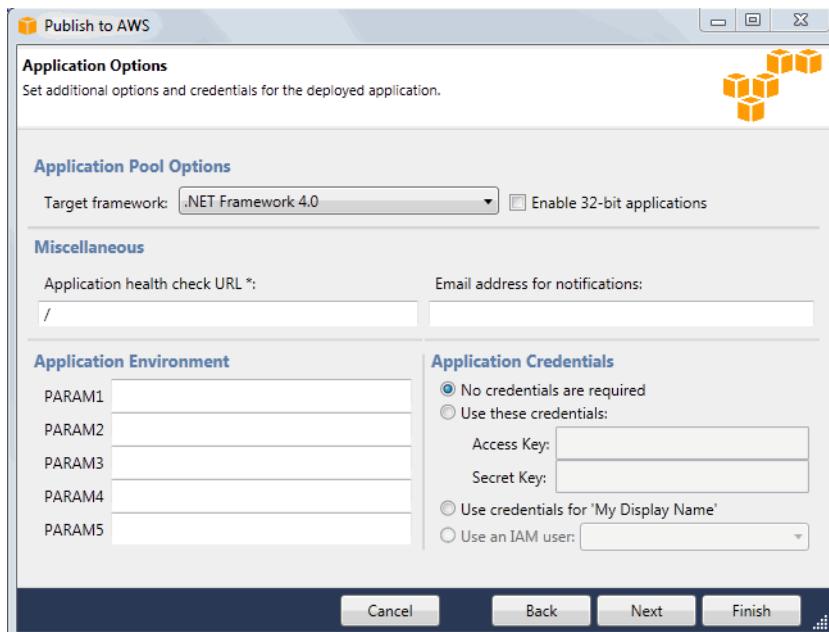
- c. Für eine skalierbare Umgebung mit Lastausgleich wählen Sie die Subnetze für den Elastic Load Balancer und die EC2-Instances aus. Sofern Sie öffentliche und private Subnetze erstellt haben, achten Sie darauf, den Elastic Load Balancer und die EC2-Instances mit dem richtigen Subnetz zu verknüpfen. Standardmäßig erstellt Amazon VPC ein öffentliches Standardsubnetz unter 10.0.0.0/24 und ein privates Subnetz unter 10.0.1.0/24. Ihre vorhandenen Subnetze können Sie in der Amazon VPC-Konsole unter <https://console.aws.amazon.com/vpc/> anzeigen.

Bei einer Umgebung mit einer einzelnen Instance benötigt die VPC nur ein öffentliches Subnetz für die Instance. Die Auswahl eines Subnetzes für den Load Balancer ist nicht möglich, da für diese Umgebung kein Load Balancer vorhanden ist. Weitere Informationen finden Sie unter [Umgebungstypen \(p. 514\)](#).

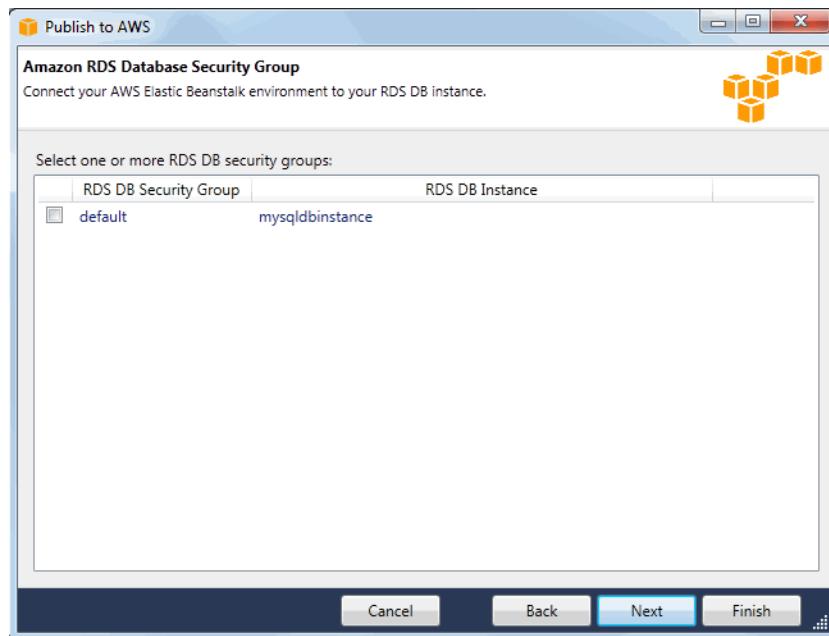
- d. Für eine skalierbare Umgebung mit Lastausgleich wählen Sie die Sicherheitsgruppe aus, die Sie für Ihre Instances erstellt haben (wenn zutreffend).

Für eine Umgebung mit einer einzelnen Instance ist kein NAT-Gerät erforderlich. Wählen Sie die Standardsicherheitsgruppe aus. Elastic Beanstalk weist der Instance eine Elastic IP-Adresse zu, sodass sie Zugang zum Internet hat.

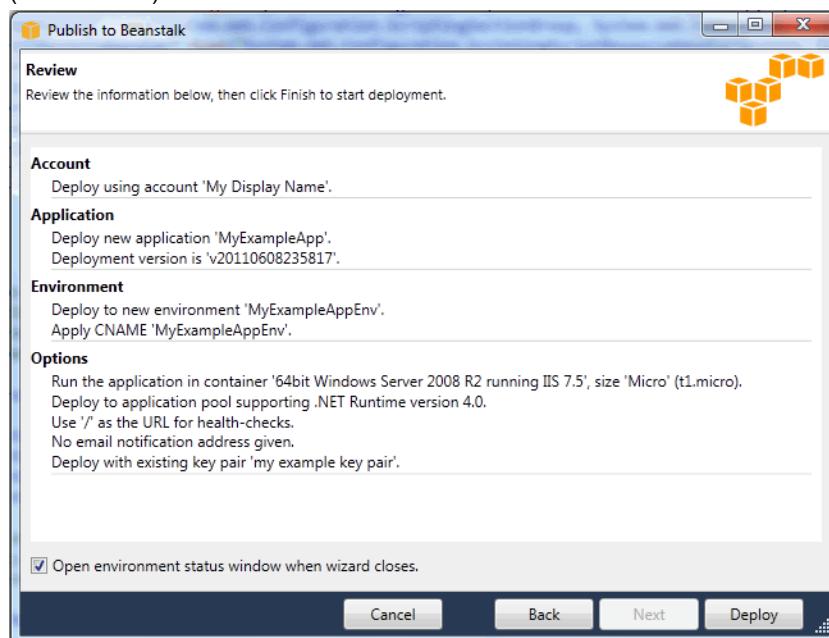
- e. Klicken Sie auf Weiter.
7. Konfigurieren Sie auf der Seite Application Options (Anwendungsoptionen) die Anwendungsoptionen.
 - a. Wählen Sie im Feld "Target framework" die Option .NET Framework 4.0 aus.
 - b. Elastic Load Balancing überprüft mittels einer Zustandsprüfung, ob die Amazon EC2-Instances, auf denen die Anwendung ausgeführt wird, stabil sind. Bei der Zustandsprüfung wird der Instance-Zustand durch Überprüfung einer vorgegebenen URL in festgelegten Intervallen bestimmt. Sie können die Standard-URL überschreiben, sodass sie mit einer vorhandenen Ressource der Anwendung übereinstimmt (z. B. /myapp/index.aspx). Dazu geben Sie diese im Feld Application health check URL (Zustandsprüfungs-URL für die Anwendung) ein. Weitere Informationen zu Anwendungszustandsprüfungen finden Sie unter [Zustandsprüfung \(p. 563\)](#).
 - c. Geben Sie eine E-Mail-Adresse an, wenn Sie Amazon Simple Notification Service (Amazon SNS)-Benachrichtigungen zu wichtigen Ereignissen für die Anwendung erhalten möchten.
 - d. Im Bereich Application Environment (Anwendungsumgebung) können Sie Umgebungsvariablen für die Amazon EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Diese Einstellung ermöglicht eine bessere Portabilität, da der Quellcode beim Wechseln zwischen den Umgebungen nicht neu kompiliert werden muss.
 - e. Wählen Sie die für die Anwendungsbereitstellung zu verwendende Option für die Anmeldeinformationen aus.



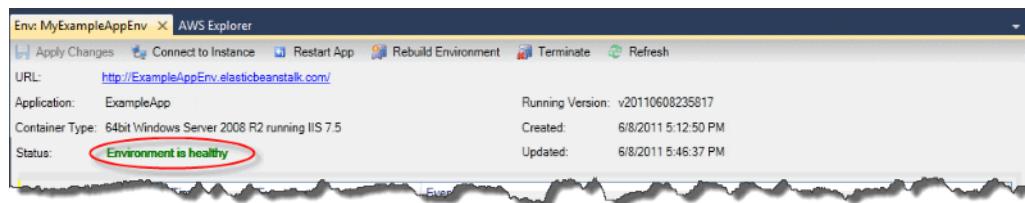
- f. Klicken Sie auf Weiter.
8. Sofern Sie bereits eine Amazon RDS-Datenbank eingerichtet haben, wird die Seite Amazon RDS DB Security Group (Amazon RDS DB-Sicherheitsgruppe) angezeigt. Wenn die Elastic Beanstalk-Umgebung mit der Amazon RDS-DB-Instance verbunden werden soll, wählen Sie eine oder mehrere Sicherheitsgruppen aus. Andernfalls fahren Sie mit dem nächsten Schritt fort. Sobald Sie fertig sind, klicken Sie auf Next (Weiter).



9. Überprüfen Sie die Bereitstellungsoptionen. Wenn alles korrekt ist, klicken Sie auf Deploy (Bereitstellen).



Das ASP.NET-Projekt wird als Webbereitstellungsdatei exportiert, in Amazon S3 hochgeladen und als neue Anwendungsversion in Elastic Beanstalk registriert. Mit der Bereitstellungsfunktion von Elastic Beanstalk wird die Umgebung überwacht, bis sie mit dem neu bereitgestellten Code verfügbar ist. Auf der Registerkarte "env (Umg.):<environment name (Umgebungsname)>" können Sie den Status der Umgebung anzeigen.



Beenden einer Umgebung

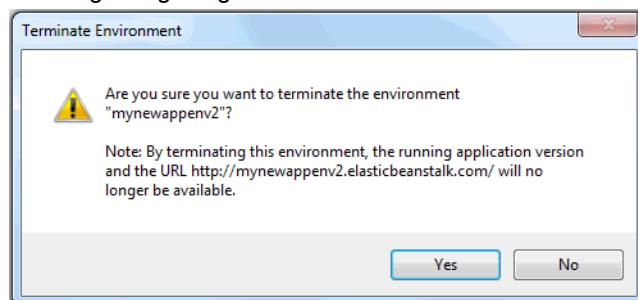
Sie können eine ausgeführte Umgebung mit dem AWS Toolkit for Visual Studio beenden, damit keine Kosten für ungenutzte AWS-Ressourcen anfallen.

Note

Sie können zu einem späteren Zeitpunkt problemlos eine neue Umgebung mit derselben Version starten.

So beenden Sie eine Umgebung

1. Erweitern Sie den Elastic Beanstalk-Knoten und den Anwendungsknoten in AWS Explorer. Klicken Sie mit der rechten Maustaste auf die Anwendungsumgebung und wählen Sie Terminate Environment (Umgebung beenden) aus.
2. Wenn Sie aufgefordert werden, klicken Sie auf Ja, um zu bestätigen, dass die Umgebung beendet werden soll. Es dauert einige Minuten, bis Elastic Beanstalk die AWS-Ressourcen beendet hat, die in der Umgebung ausgeführt werden.



Note

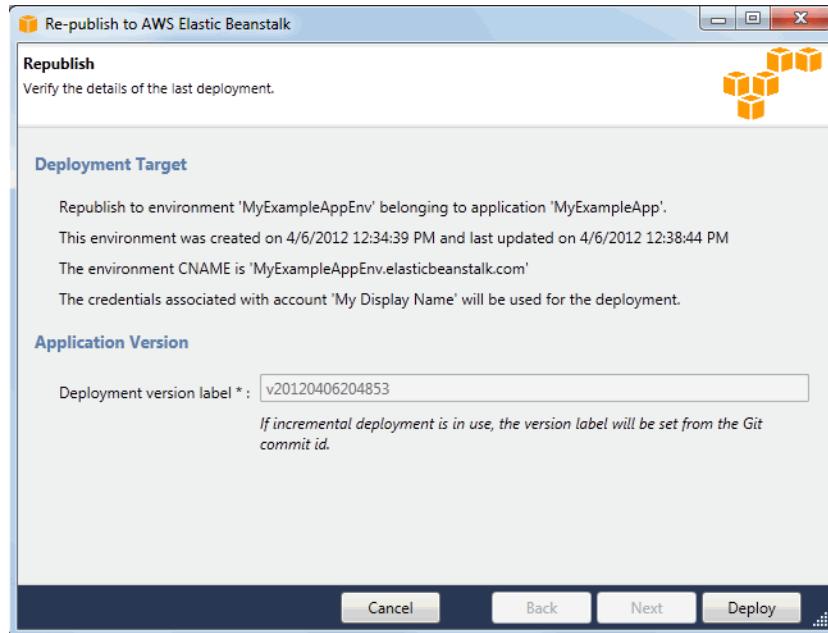
Nachdem die Umgebung beendet ist, steht ihr zugehöriger CNAME wieder zur Verfügung und kann von allen verwendet werden.

Bereitstellen für Ihre Umgebung

Nachdem Sie Ihre Anwendung getestet haben, können Sie Ihre Anwendung einfach bearbeiten und erneut bereitstellen und die Ergebnisse in wenigen Augenblicken anzeigen.

So bearbeiten Sie Ihre ASP.NET-Webanwendung und stellen Sie erneut bereit

1. Klicken Sie in Solution Explorer (Projektmappen-Explorer) mit der rechten Maustaste auf die Anwendung und dann auf Republish to Environment (Erneut in Umgebung veröffentlichen) <**Name Ihrer Umgebung**>. Der Re-publish to AWS Elastic Beanstalk (Erneut in AWS Elastic Beanstalk veröffentlichen)-Assistent wird geöffnet.



- Überprüfen Sie die Bereitstellungsdetails und klicken Sie dann auf Deploy (Bereitstellen).

Note

Wenn Sie Ihre Einstellungen ändern möchten, klicken Sie auf Cancel (Abbrechen) und verwenden Sie stattdessen den Publish to AWS (In AWS veröffentlichen)-Assistenten. Detaillierte Anweisungen finden Sie unter [Erstellen einer Elastic Beanstalk-Umgebung \(p. 220\)](#).

Ihr aktualisiertes ASP.NET-Webprojekt wird als Webbereitstellungsdatei mit der neuen Versionsbezeichnung exportiert, in Amazon S3 hochgeladen und als neue Anwendungsversion in Elastic Beanstalk registriert. Mit der Bereitstellungsfunktion von Elastic Beanstalk wird die vorhandene Umgebung überwacht, bis sie mit dem neu bereitgestellten Code verfügbar ist. Auf der Registerkarte env: (Umg.:)<environment name (*Umgebungsname*)> können Sie den Status der Umgebung anzeigen.

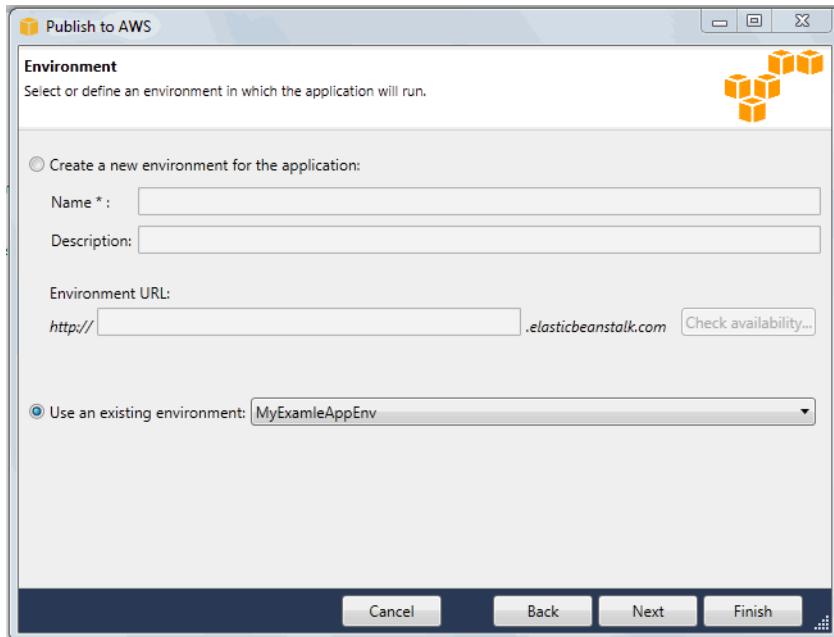
Sie können auch eine vorhandene Anwendung in einer vorhandenen Umgebung bereitstellen, wenn Sie zum Beispiel auf eine frühere Anwendungsversion zurücksetzen.

So stellen Sie eine Anwendungsversion in einer vorhandenen Umgebung bereit

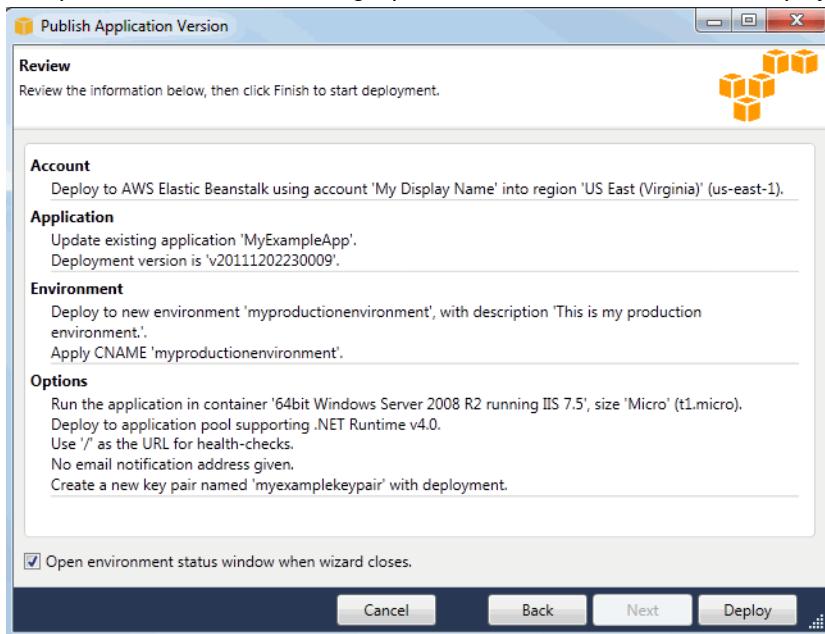
- Klicken Sie mit der rechten Maustaste auf Ihre Elastic Beanstalk-Anwendung, indem Sie den Elastic Beanstalk-Knoten in AWS Explorererweitern. Wählen Sie View Status (Status anzeigen) aus.
- Klicken Sie auf der Registerkarte App: <*Anwendungsname*> auf Versions (Versionen).

Events	Publish Version	Delete Version			
Versions	Version Label	Description	Created On	S3 Bucket	S3 Key
	v201120223102		12/2/2011 3:42:59 PM	elasticbeanstalk-us-east-1-akiajcpa2ajx5z2fgdq	MyExampleApp/AVS
	v201120223009	This is my sample application.	12/2/2011 3:18:19 PM	elasticbeanstalk-us-east-1-akiajcpa2ajp5z2fgdq	MyExampleApp/AVS

- Klicken Sie auf die Anwendungsversion, die Sie bereitstellen möchten, und dann auf Publish Version (Version veröffentlichen).
- Klicken Sie im Assistenten Publish Application Version (Anwendungsversion veröffentlichen) auf Next (Weiter).



- Überprüfen Sie die Bereitstellungsoptionen und klicken Sie dann auf Deploy (Bereitstellen).



Ihr ASP.NET-Projekt wird als eine Webbereitstellungsdatei exportiert und in Amazon S3 hochgeladen. Mit der Bereitstellungsfunktion von Elastic Beanstalk wird die Umgebung überwacht, bis sie mit dem neu bereitgestellten Code verfügbar ist. Auf der Registerkarte env: (Umg.:) **<environment name>** (**Umgebungsname**) können Sie den Status der Umgebung anzeigen.

Verwalten Ihrer Elastic Beanstalk-Anwendungsumgebungen

Mit dem AWS Toolkit for Visual Studio und der AWS-Managementkonsole können Sie die Bereitstellung und Konfiguration der AWS-Ressourcen, die von Ihren Anwendungsumgebungen verwendet werden, ändern. Weitere Informationen zur Verwaltung Ihres Anwendungsumgebungen mithilfe der AWS

Management Console finden Sie unter [Verwalten von Umgebungen \(p. 425\)](#). In diesem Abschnitt werden die spezifischen Service-Einstellungen beschrieben, die Sie im AWS Toolkit for Visual Studio als Teil Ihrer Anwendungsumgebungs-Konfiguration bearbeiten können.

Ändern der Umgebungskonfigurationseinstellungen

Wenn Sie Ihre Anwendung bereitstellen, konfiguriert Elastic Beanstalk eine Reihe von AWS Cloud Computing-Services. Sie können mit dem AWS Toolkit for Visual Studio steuern, wie diese einzelnen Services konfiguriert sind.

So bearbeiten Sie die Umgebungseinstellungen einer Anwendung

- Erweitern Sie den Elastic Beanstalk-Knoten und Ihren Anwendungsknoten. Klicken Sie anschließend im AWS Explorer mit der rechten Maustaste auf Ihre Elastic Beanstalk-Umgebung. Wählen Sie View Status (Status anzeigen) aus.

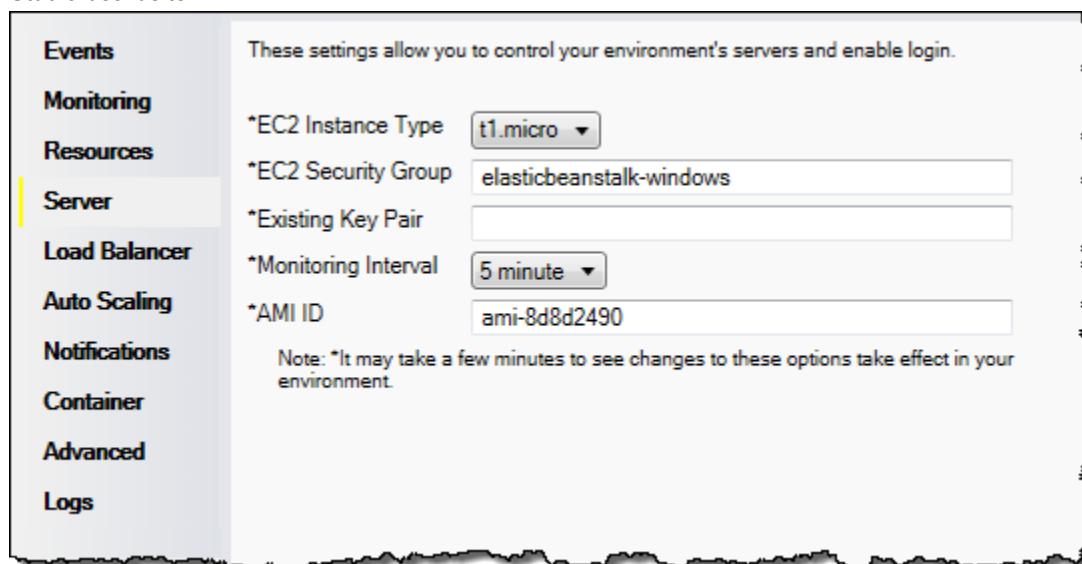
Sie können nun Einstellungen für Folgendes konfigurieren:

- herstellen
- Load Balancing
- Auto Scaling
- Benachrichtigungen
- Umgebungseigenschaften

Konfigurieren von EC2-Server-Instances mit dem AWS Toolkit for Visual Studio

Amazon Elastic Compute Cloud (Amazon EC2) ist ein Webservice, mit dem Sie Server-Instances in den Rechenzentren von Amazon starten und verwalten. Sie können Amazon EC2-Server-Instances zu jedem beliebigen Zeitpunkt, so lange wie nötig und für alle rechtlichen Zwecke verwenden. Instances sind in verschiedenen Größen und Konfigurationen verfügbar. Weitere Informationen erhalten Sie unter [Amazon EC2](#).

Sie können die Amazon EC2-Instance-Konfiguration der Elastic Beanstalk-Umgebung über die Registerkarte Server innerhalb der Registerkarte für die Anwendungsumgebung im AWS Toolkit for Visual Studio bearbeiten.



Amazon EC2-Instance-Typen

Instance type (Instance-Typ) zeigt die Instance-Typen an, die für die Elastic Beanstalk-Anwendung verfügbar sind. Ändern Sie den Instance-Typ, um einen Server mit den Merkmalen (einschließlich Speichergröße und CPU-Leistung) auszuwählen, die für Ihre Anwendung am besten geeignet sind. Anwendungen mit aufwändigen und lange dauernden Operationen erfordern mehr CPU oder Speicher.

Weitere Informationen über die für die Elastic Beanstalk-Anwendung verfügbaren Amazon EC2-Instance-Typen finden Sie unter [Instance-Typen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Amazon EC2-Sicherheitsgruppen

Sie können den Zugriff auf Ihre Elastic Beanstalk-Anwendung mit einer Amazon EC2-Sicherheitsgruppe steuern. Eine Sicherheitsgruppe definiert Firewall-Regeln für Ihre Instances. Diese Regeln legen fest, welcher eingehende Netzwerkverkehr an Ihre Instance übertragen werden soll. Anderer eingehender Datenverkehr wird verworfen. Sie können die Regeln für eine Gruppe jederzeit ändern. Die neuen Regeln werden automatisch für alle laufenden und künftigen Instances durchgesetzt.

Sie können Ihre Amazon EC2-Sicherheitsgruppen über die AWS-Managementkonsole oder mit dem AWS Toolkit for Visual Studio einrichten. Sie können angeben, welche Amazon EC2-Sicherheitsgruppen den Zugriff auf Ihre Elastic Beanstalk-Anwendung steuern, indem Sie die Namen von einer oder mehreren Amazon EC2-Sicherheitsgruppen (durch Kommas getrennt) in das Textfeld EC2 Security Groups (EC2-Sicherheitsgruppen) eingeben.

Note

Stellen Sie sicher, dass Port 80 (HTTP) über 0.0.0.0/0 als Quell-CIDR-Bereich zugreifbar ist, wenn Sie Zustandsprüfungen für Ihre Anwendung aktivieren möchten. Weitere Informationen zu Zustandsprüfungen finden Sie unter [Zustandsprüfungen \(p. 236\)](#).

So erstellen Sie eine Sicherheitsgruppe mit dem AWS Toolkit for Visual Studio

1. Erweitern Sie in Visual Studio unter AWS Explorer den Amazon EC2-Knoten und doppelklicken Sie dann auf Security Groups (Sicherheitsgruppen).
2. Klicken Sie auf Create Security Group (Sicherheitsgruppe erstellen) und geben Sie einen Namen und eine Beschreibung für Ihre Sicherheitsgruppe ein.
3. Klicken Sie auf OK.

Weitere Informationen zu Amazon EC2-Sicherheitsgruppen finden Sie unter [Verwenden von Sicherheitsgruppen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Amazon EC2-Schlüsselpaare

Mit einem Amazon EC2-Schlüsselpaar können Sie sich sicher an den für Ihre Elastic Beanstalk-Anwendung bereitgestellten Amazon EC2-Instances anmelden.

Important

Bevor Sie auf die von Elastic Beanstalk bereitgestellten Amazon EC2-Instances zugreifen können, müssen Sie zunächst ein Amazon EC2-Schlüsselpaar erstellen und die von Elastic Beanstalk bereitgestellten Amazon EC2-Instances zur Verwendung des Amazon EC2-Schlüsselpaares konfigurieren. Sie können ein Schlüsselpaar mit dem Assistenten Publish to AWS (In AWS veröffentlichen) im AWS Toolkit for Visual Studio erstellen, wenn Sie Ihre Anwendung für Elastic Beanstalk bereitstellen. Wenn Sie zusätzliche Schlüsselpaare mit dem Toolkit erstellen möchten, führen Sie die folgenden Schritte aus. Alternativ können Sie Ihre Amazon EC2-Schlüsselpaare mit der [AWS-Managementkonsole](#) einrichten. Anleitungen zum Erstellen eines Schlüsselpaares für Amazon EC2 finden Sie im [Amazon Elastic Compute Cloud-Handbuch „Erste Schritte“](#).

Im Textfeld Existing Key Pair (Vorhandenes Schlüsselpaar) können Sie den Namen eines Amazon EC2-Schlüsselpaares angeben, das Sie verwenden können, um sich sicher bei den Amazon EC2-Instances anzumelden, die Ihre Elastic Beanstalk-Anwendung ausführen.

So geben Sie den Namen eines Amazon EC2-Schlüsselpaares ein

1. Erweitern Sie den Amazon EC2-Knoten und doppelklicken Sie auf Key Pairs (Schlüsselpaare).
2. Klicken Sie auf Create Key Pair (Schlüsselpaar erstellen) und geben Sie den Namen des Schlüsselpaares ein.
3. Klicken Sie auf OK.

Weitere Informationen zu Amazon EC2-Schlüsselpaaren finden Sie unter [Verwenden von Amazon EC2-Anmeldeinformationen](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch. Weitere Information zum Verbinden mit Amazon EC2-Instances erhalten Sie unter [Auflisten von Server-Instances/Verbinden mit Server-Instances \(p. 242\)](#).

Überwachungsintervall

Standardmäßig werden nur grundlegende Amazon CloudWatch-Metriken aktiviert. Sie geben Daten in 5-Minuten-Intervallen zurück. Sie können noch genauere CloudWatch-Metriken in 1-Minuten-Intervallen aktivieren, indem Sie auf der Registerkarte Configuration (Konfiguration) im Abschnitt Server für den Wert Monitoring Interval (Überwachungsintervall) den Wert 1 minute (1 Minute) für Ihre Umgebung im AWS Toolkit for Eclipse auswählen.

Note

Für Metriken in 1-Minuten-Intervallen können Amazon CloudWatch-Servicegebühren anfallen. Weitere Informationen finden Sie unter [Amazon CloudWatch](#).

Benutzerdefinierte AMI-ID

Sie können das für Ihre Amazon EC2-Instances verwendete Standard-AMI mit einem eigenen, benutzerdefinierten AMI überschreiben. Geben Sie dazu den Bezeichner Ihres benutzerdefinierten AMI auf der Registerkarte Configuration (Konfiguration) im Abschnitt Server in das Feld Custom AMI ID (ID für benutzerdefiniertes AMI) für Ihre Umgebung im AWS Toolkit for Eclipse ein.

Important

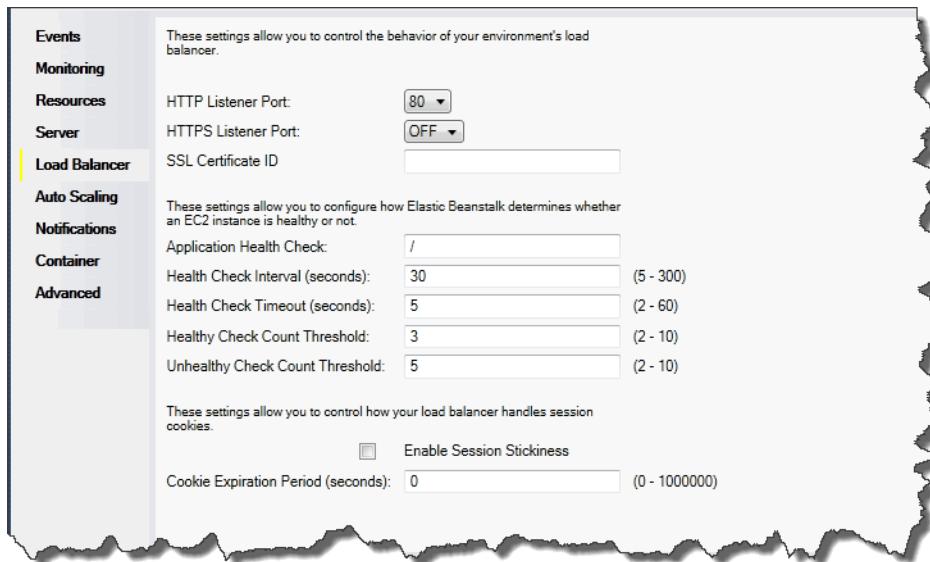
Die Verwendung eines eigenen AMI ist eine Aufgabe für Fortgeschrittene und sollte mit Bedacht ausgeführt werden. Wenn Sie ein benutzerdefiniertes AMI benötigen, empfehlen wir, das standardmäßige Elastic Beanstalk-AMI als Ausgangspunkt zu nehmen und es zu ändern. Für einen ordnungsgemäßen Zustand erwartet Elastic Beanstalk, dass die Amazon EC2-Instances eine Reihe von Anforderungen erfüllen, einschließlich Vorhandensein eines laufenden Hostmanagers. Werden diese Anforderungen nicht erfüllt, funktioniert Ihre Umgebung möglicherweise nicht ordnungsgemäß.

Konfigurieren von Elastic Load Balancing mit dem AWS Toolkit for Visual Studio

Elastic Load Balancing ist ein Amazon-Webservice, der Ihnen hilft, die Verfügbarkeit und Skalierbarkeit Ihrer Anwendung zu verbessern. Dieser Service macht es für Sie einfacher, Anwendungslasten auf zwei oder mehr Amazon EC2-Instances zu verteilen. Elastic Load Balancing ermöglicht die Verfügbarkeit durch Redundanz und unterstützt wachsenden Datenverkehr für Ihre Anwendung.

Mit Elastic Load Balancer können Sie den eingehenden Datenverkehr der Anwendung automatisch auf sämtliche Instances verteilen, die Sie ausführen. Mit dem Service ist es auch ganz einfach, neue Instances hinzuzufügen, wenn Sie die Kapazität Ihrer Anwendung erhöhen.

Elastic Beanstalk stellt Elastic Load Balancing automatisch bereit, wenn Sie eine Anwendung bereitstellen. Sie können die Amazon EC2-Instance-Konfiguration der Elastic Beanstalk-Umgebung über die Registerkarte Load Balancer innerhalb der Registerkarte für die Anwendungsumgebung im AWS Toolkit for Visual Studio bearbeiten.



In den folgenden Abschnitten werden die Elastic Load Balancing-Parameter beschrieben, die Sie für Ihre Anwendung konfigurieren können.

Ports

Der Load Balancer, der für die Verarbeitung von Anforderungen für Ihre Elastic Beanstalk-Anwendung bereitgestellt wurde, sendet Anforderungen an Amazon EC2-Instances, auf denen Ihre Anwendung ausgeführt wird. Der bereitgestellte Load Balancer kann auf Anforderungen über HTTP- und HTTPS-Ports warten und diese an Amazon EC2-Instances in Ihrer AWS Elastic Beanstalk-Anwendung weiterleiten. Standardmäßig verarbeitet der Load Balancer Anfragen auf dem HTTP-Port. Mindestens einer der Ports (HTTP oder HTTPS) muss aktiviert werden.



Important

Stellen Sie sicher, dass der von Ihnen angegebene Port nicht gesperrt ist. Andernfalls können die Benutzer keine Verbindung zu Ihrer Elastic Beanstalk-Anwendung herstellen.

Steuern des HTTP-Ports

Um den HTTP-Port auszuschalten, wählen Sie OFF (AUS) für HTTP Listener Port aus. Zum Aktivieren des HTTP-Ports wählen Sie einen HTTP-Port (z. B. 80) aus der Liste aus.

Note

Für den Zugriff auf Ihre Umgebung mit einem anderen als dem Standardport 80, z. B. Port 8080, fügen Sie dem vorhandenen Load Balancer einen Listener hinzu und konfigurieren Sie den neuen Listener für die Überwachung des entsprechenden Ports.

Geben Sie beispielsweise unter Verwendung der [AWS CLI für Classic Load Balancer](#) den folgenden Befehl ein, wobei Sie `LOAD_BALANCER_NAME` durch den Namen des Load Balancers für Elastic Beanstalk ersetzen.

```
aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
--listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
InstancePort=80"
```

Geben Sie beispielsweise unter Verwendung der [AWS CLI für Application Load Balancer](#) den folgenden Befehl ein, wobei Sie `LOAD_BALANCER_ARN` durch den ARN des Load Balancers für Elastic Beanstalk ersetzen.

```
aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP --
port 8080
```

Wenn Elastic Beanstalk Ihre Umgebung überwachen soll, entfernen Sie den Listener für Port 80 nicht.

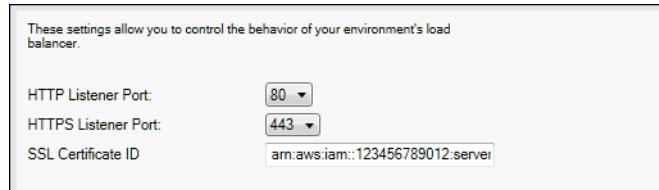
Steuern des HTTPS-Ports

Elastic Load Balancing unterstützt das HTTPS/TLS-Protokoll, um die Verschlüsselung des Datenverkehrs für Client-Verbindungen zum Load Balancer zu aktivieren. Verbindungen vom Load Balancer mit EC2-Instances verwenden eine Klartext-Verschlüsselung. Standardmäßig ist der HTTPS-Port deaktiviert.

So aktivieren Sie den HTTPS-Port

1. Erstellen Sie ein neues Zertifikat mit AWS Certificate Manager (ACM) oder laden Sie ein Zertifikat und einen Schlüssel in AWS Identity and Access Management (IAM) hoch. Weitere Informationen zum Anfordern von ACM-Zertifikaten finden Sie unter [Anfordern eines Zertifikats](#) im AWS Certificate Manager-Benutzerhandbuch. Weitere Informationen zum Importieren von Zertifikaten von Drittanbietern in ACM finden Sie unter [Importieren von Zertifikaten](#) im AWS Certificate Manager-Benutzerhandbuch. Wenn ACM in Ihrer Region nicht verfügbar ist, verwenden Sie AWS Identity and Access Management (IAM), um ein Zertifikat von einem Drittanbieter hochzuladen. Die ACM- und IAM-Services speichern das Zertifikat und stellen einen Amazon-Ressourcennamen (ARN) für das SSL-Zertifikat bereit. Weitere Informationen zum Erstellen und Hochladen von Zertifikaten in IAM finden Sie unter [Arbeiten mit Serverzertifikaten](#) im IAM-Benutzerhandbuch.

2. Geben Sie den HTTPS-Port an, indem Sie einen Port für HTTPS Listener Port auswählen.



3. Geben Sie für SSL Certificate ID (SSL-Zertifikat-ID) den Amazon-Ressourcennamen (ARN) Ihres SSL-Zertifikats ein. Zum Beispiel `arn:aws:iam::123456789012:server-certificate/abc/certs/build` oder `arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`. Verwenden Sie das SSL-Zertifikat, das Sie in Schritt 1 erstellt oder hochgeladen haben.

Um den HTTPS-Port auszuschalten, wählen Sie OFF (AUS) für HTTPS Listener Port.

Zustandsprüfungen

Die Definition der Zustandsprüfung enthält einen URL, die für den Instance-Zustand abgefragt wird. Standardmäßig nutzt Elastic Beanstalk TCP:80 für Nonlegacy-Container und HTTP:80 für Legacy-

Container. Sie können die Standard-URL überschreiben, sodass sie mit einer vorhandenen Ressource der Anwendung übereinstimmt (z. B. `/myapp/default.aspx`). Dazu geben Sie diese im Feld Application Health Check URL (Zustandsprüfungs-URL für die Anwendung) ein. Wenn Sie die Standard-URL überschreiben, verwendet Elastic Beanstalk HTTP zum Abfragen der Ressource. Wenn Sie überprüfen möchten, ob Sie einen Legacy-Container verwenden, erhalten Sie Informationen unter [the section called "Warum sind einige Plattformversionen als Legacy gekennzeichnet?" \(p. 502\)](#).

Sie können die Einstellungen für die Zustandsprüfung im Abschnitt EC2 Instance Health Check (Zustandsprüfung für EC2-Instances) des Bereichs Load Balancing steuern.

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)

Die Definition der Zustandsprüfung enthält einen URL, die für den Instance-Zustand abgefragt wird. Überschreiben Sie die Standard-URL, sodass sie mit einer vorhandenen Ressource der Anwendung übereinstimmt (z. B. `/myapp/index.jsp`). Dazu geben Sie diese im Feld Application Health Check URL (Zustandsprüfungs-URL für die Anwendung) ein.

In der folgenden Liste werden die Zustandsprüfungs-Parameter beschrieben, die Sie für Ihre Anwendung festlegen können.

- Für Health Check Interval (seconds) (Zustandsprüfungsintervall (Sekunden)) geben Sie die Anzahl der Sekunden ein, die Elastic Load Balancing zwischen Zustandsprüfungen für die Amazon EC2-Instances Ihrer Anwendung wartet.
- Für Health Check Timeout (seconds) (Zustandsprüfungs-Timeout (Sekunden)) geben Sie die Anzahl der Sekunden ein, die Elastic Load Balancing auf eine Antwort wartet, bevor die Instance als nicht reagierend eingestuft wird.
- Für Healthy Check Count Threshold (Schwellenwert Anzahl erfolgreicher Zustandsprüfungen) und Unhealthy Check Count Threshold (Schwellenwert Anzahl fehlgeschlagener Zustandsprüfungen) geben Sie die Anzahl der aufeinanderfolgenden erfolgreichen oder nicht erfolgreichen URL-Proben ein, bevor Elastic Load Balancing den Status des Instance-Zustands ändert. Beispiel: Die Angabe von 5 für Unhealthy Check Count Threshold (Schwellenwert Anzahl fehlgeschlagener Zustandsprüfungen) bedeutet, dass die URL fünfmal hintereinander eine Fehlermeldung oder eine Zeitüberschreitung zurückgeben müssten, bevor Elastic Load Balancing die Zustandsprüfung als fehlgeschlagen einstufen würde.

Sitzungen

Standardmäßig leitet ein Load Balancer jede Anforderung getrennt an die Server-Instance mit der kleinsten Last weiter. Im Vergleich dazu bindet eine Sticky Session eine Benutzersitzung an eine bestimmte Server-Instance, sodass alle vom Benutzer während der Sitzung gesendeten Anforderungen an die gleiche Server-Instance übermittelt werden.

Elastic Beanstalk nutzt von Load Balancer generierte HTTP-Cookies, wenn Sticky Sessions für eine Anwendung aktiviert sind. Der Load Balancer verwendet ein vom Load-Balancer generiertes spezielles Cookie, um die Anwendungs-Instance für jede Anforderung nachzuverfolgen. Wenn der Load Balancer eine Anforderung empfängt, prüft er zunächst, ob dieses Cookie in der Anforderung vorhanden ist. Wenn ja, wird die Anforderung an die im Cookie angegebene Anwendungs-Instance gesendet. Wenn kein Cookie vorhanden ist, wählt der Load Balancer eine Anwendungs-Instance basierend auf dem vorhandenen Load Balancing-Algorithmus aus. Ein Cookie wird in die Antwort eingefügt, um nachfolgende Anforderungen von demselben Benutzer an diese Anwendungs-Instance zu binden. Mit der Konfiguration der Richtlinie wird ein Cookie-Ablauf definiert, der die Dauer der Gültigkeit für jedes Cookie festlegt.

Sie können über den Abschnitt Sessions (Sitzungen) auf der Registerkarte Load Balancer angeben, ob der Load Balancer für Ihre Anwendung "Sticky Sessions" erlaubt.

These settings allow you to control how your load balancer handles session cookies.

Enable Session Stickiness

Cookie Expiration Period (seconds): (0 - 1000000)

Weitere Informationen zu Elastic Load Balancing finden Sie im [Elastic Load Balancing-Entwicklerhandbuch](#).

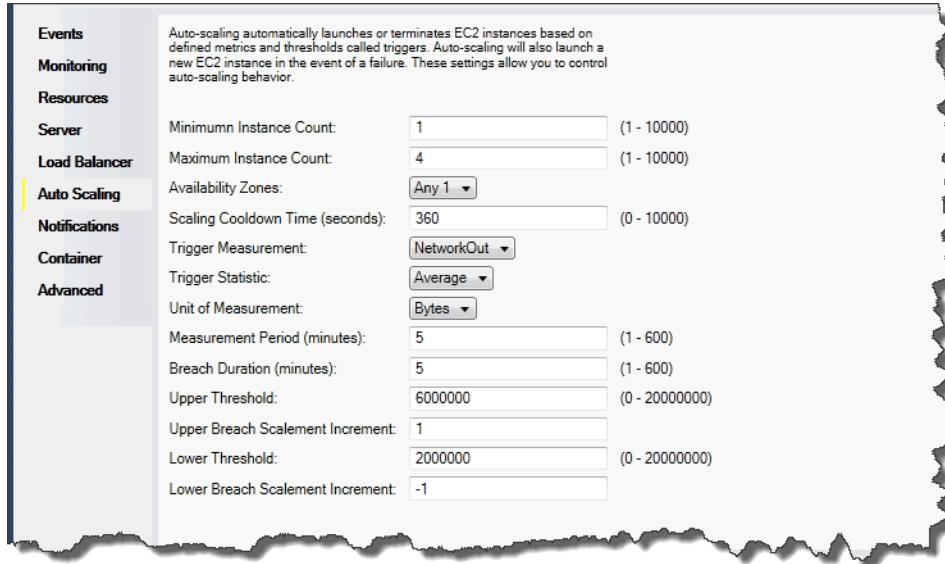
Konfigurieren von Auto Scaling mit dem AWS Toolkit for Visual Studio

Amazon EC2 Auto Scaling ist ein Amazon-Webservice, mit dem Amazon EC2-Instances basierend auf benutzerdefinierten Auslösern automatisch gestartet oder beendet werden. Benutzer können Auto Scaling-Gruppen einrichten und diesen Gruppen Auslöser zuweisen, um Datenverarbeitungsressourcen automatisch basierend auf Metriken, wie Bandbreitennutzung oder CPU-Auslastung, zu skalieren. Amazon EC2 Auto Scaling ruft mithilfe von Amazon CloudWatch Metriken für die Server-Instances ab, auf denen die Anwendung ausgeführt wird.

Mit Amazon EC2 Auto Scaling können Sie eine Gruppe von Amazon EC2-Instances nehmen und verschiedene Parameter festlegen, damit diese Gruppe automatisch größer oder kleiner wird. Mit Amazon EC2 Auto Scaling können Amazon EC2-Instances zur Gruppe hinzugefügt oder daraus entfernt werden, damit sie nahtlos auf Datenverkehrsänderungen an Ihrer Anwendung eingehen können.

Des Weiteren wird mit Amazon EC2 Auto Scaling auch die Integrität der einzelnen gestarteten Amazon EC2-Instances überwacht. Falls eine Instance unerwartet beendet wird, erkennt Amazon EC2 Auto Scaling dies und startet eine andere Instance als Ersatz. Mit dieser Funktion wird automatisch stets die von Ihnen festgelegte Anzahl an Amazon EC2-Instances ausgeführt.

Elastic Beanstalk stellt Amazon EC2 Auto Scaling für Ihre Anwendung bereit. Sie können die Amazon EC2-Instance-Konfiguration der Elastic Beanstalk-Umgebung über die Registerkarte Auto Scaling innerhalb der Registerkarte für die Anwendungsumgebung im AWS Toolkit for Visual Studio bearbeiten.



Im folgenden Abschnitt wird beschrieben, wie Sie Auto Scaling-Parameter für die Anwendung konfigurieren.

Starten der Konfiguration

Sie können die Startkonfiguration so bearbeiten, dass gesteuert wird, wie Ihre Elastic Beanstalk-Anwendung Amazon EC2 Auto Scaling-Ressourcen bereitstellt.

In den Feldern Minimum Instance Count (Minimale Instance-Anzahl) und Maximum Instance Count (Maximale Instance-Anzahl) geben Sie die minimale sowie die maximale Größe der Auto Scaling-Gruppe an, die von der Elastic Beanstalk-Anwendung genutzt wird.

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="button" value="Any"/>
Scaling Cooldown Time (seconds):	<input type="text" value="360"/> (0 - 10000)

Note

Um die Anzahl der Amazon EC2-Instances als Festwert festzulegen, geben Sie in den Feldern Minimum Instance Count (Minimale Instance-Anzahl) und Maximum Instance Count (Maximale Instance-Anzahl) denselben Wert ein.

Im Feld Availability Zones geben Sie die Anzahl der Availability Zones an, in denen sich Ihre Amazon EC2-Instances befinden sollen. Es ist wichtig, diese Zahl festzulegen, wenn Sie fehlertolerante Anwendungen erstellen möchten. Falls eine Availability Zone ausfällt, werden die Instances in den anderen Availability Zones weiterhin ausgeführt.

Note

Derzeit ist es nicht möglich anzugeben, in welcher Availability Zone sich Ihre Instance befinden wird.

Auslöser

Bei einem Auslöser handelt es sich um einen Amazon EC2 Auto Scaling-Mechanismus, mit dem Sie festlegen, wann die Instance-Anzahl im System erhöht (hochskaliert) bzw. verringert (herunterskaliert) werden soll. Sie können die Auslöser so konfigurieren, dass Sie bei allen Metriken, die in Amazon CloudWatch veröffentlicht werden, ausgelöst werden, wie z. B. die CPU-Auslastung. Zudem wird ermittelt, ob die vorgegebenen Bedingungen erfüllt sind. Wird der obere oder untere Schwellenwert der Metrikbedingungen für einen bestimmten Zeitraum überschritten, startet der Auslöser den anhaltenden Prozess der Skalierung.

Sie können einen Skalierungsauslöser für Ihre Elastic Beanstalk-Anwendung mit dem AWS Toolkit for Visual Studio definieren.

Trigger Measurement:	<input type="button" value="NetworkOut"/>
Trigger Statistic:	<input type="button" value="Average"/>
Unit of Measurement:	<input type="button" 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="200000"/> (0 - 20000000)
Lower Breach Scalement Increment:	<input type="text" value="-1"/>

Die Amazon EC2 Auto Scaling-Auslöser überwachen eine angegebene Amazon CloudWatch-Metrik für eine Instance. Zu den Auslösern zählen CPU-Auslastung, Netzwerkdatenverkehr und Datenträgeraktivität. Mit der Einstellung Trigger Measurement (Auslösermessung) wählen Sie eine Metrik für den Auslöser aus.

In der folgenden Liste werden die Auslöserparameter beschrieben, die Sie mit der AWS-Managementkonsole konfigurieren können.

- Sie können angeben, welche Statistik der Auslöser verwenden soll. Sie können Minimum, Maximum, Sum (Summe) oder Average (Durchschnitt) für Trigger Statistic (Auslöserstatistik) auswählen.
- Für Unit of Measurement (Maßeinheit) geben Sie die Maßeinheit für den Auslöser an.
- Der Wert im Feld Measurement Period (Messzeitraum) gibt an, wie häufig Amazon CloudWatch die Metriken für Ihren Auslöser misst. Mit Breach Duration (Verletzungsdauer) wird der Zeitraum angegeben, für den der Grenzwert einer Metrik überschritten werden kann (wie unter Upper Threshold (Oberer Schwellenwert) und Lower Threshold (Unterer Schwellenwert) definiert), bevor der Auslöser ausgelöst wird.
- Unter Upper Breach Scale Increment (Oberes Skalierungsinkrement für eine Verletzung) und Lower Breach Scale Increment (Unteres Skalierungsinkrement für eine Verletzung) geben Sie an, wie viele Amazon EC2-Instances im Rahmen der Skalierung hinzugefügt oder entfernt werden sollen.

Weitere Informationen zu Amazon EC2 Auto Scaling finden Sie im Abschnitt Amazon EC2 Auto Scaling in der [Dokumentation zu Amazon Elastic Compute Cloud](#).

Konfigurieren von Benachrichtigungen mit dem AWS Toolkit for Visual Studio

Elastic Beanstalk verwendet den Amazon Simple Notification Service (Amazon SNS) für Benachrichtigungen zu wichtigen Ereignissen in Ihrer Anwendung. Um Amazon SNS-Benachrichtigungen zu aktivieren, geben Sie einfach Ihre E-Mail-Adresse in das Feld Email Address ein. Um diese Benachrichtigungen zu deaktivieren, entfernen Sie Ihre E-Mail-Adresse aus dem Feld.



Konfigurieren von .NET-Containern mit dem AWS Toolkit for Visual Studio

Über das Feld Container/.NET Options (Container/.NET-Optionen) können Sie das Verhalten Ihrer Amazon EC2-Instances optimieren und die Amazon S3-Protokollrotation aktivieren oder deaktivieren. Sie können das AWS Toolkit for Visual Studio verwenden, um Ihre Container-Informationen zu konfigurieren.

Note

Sie können Ihre Konfigurationseinstellungen ändern, ohne eine Ausfallzeit in Kauf nehmen zu müssen, indem Sie den CNAME für Ihre Umgebungen austauschen. Weitere Informationen finden Sie unter [Blau/Grün-Bereitstellungen mit AWS Elastic Beanstalk \(p. 481\)](#).

Wenn Sie möchten, können Sie die Anzahl der Parameter erhöhen. Weitere Informationen über das Erhöhen der Parameteranzahl finden Sie unter [Optionseinstellungen \(p. 724\)](#).

So greifen Sie auf den Container/.NET-Optionsbereich für Ihre Elastic Beanstalk-Anwendung zu

1. In AWS Toolkit for Visual Studio erweitern Sie den Elastic Beanstalk-Knoten und Ihren Anwendungsknoten.

2. Doppelklicken Sie im AWS Explorer auf Ihre Elastic Beanstalk-Umgebung.
3. Klicken Sie unten im Bereich Overview (Übersicht) auf die Registerkarte Configuration (Konfiguration).
4. Konfigurieren Sie im Bereich Container die Containeroptionen.

The screenshot shows a configuration dialog for a .NET application. At the top, a note says "These properties are passed into the application as environment variables." Below this are several input fields:

- AWS_ACCESS_KEY_ID: [Input Field]
- AWS_SECRET_KEY_ID: [Input Field]
- PARAM1: [Input Field]
- PARAM2: [Input Field]
- PARAM3: [Input Field]
- PARAM4: [Input Field]
- PARAM5: [Input Field]

Below these are two dropdown menus:

- Target Runtime: 4.0
- Enable 32-bit Applications: False

.NET-Container-Optionen

Sie können die Version von .NET-Framework für Ihre Anwendung wählen. Wählen Sie entweder 2.0 oder 4.0 für Target runtime (Ziellaufzeit) aus. Wählen Sie Enable 32-bit Applications (32-Bit-Anwendungen aktivieren) aus, wenn Sie 32-Bit-Anwendungen aktivieren möchten.

Anwendungseinstellungen

Unter Application Settings (Anwendungseinstellungen) geben Sie Umgebungsvariablen an, die aus dem Anwendungscode ausgelesen werden.

The screenshot shows a configuration dialog for a .NET application under Application Settings. At the top, a note says "These properties are passed into the application as environment variables." Below this are several input fields:

- AWS_ACCESS_KEY_ID: [Input Field]
- AWS_SECRET_KEY_ID: [Input Field]
- PARAM1: [Input Field]
- PARAM2: [Input Field]
- PARAM3: [Input Field]
- PARAM4: [Input Field]
- PARAM5: [Input Field]

Verwalten von Konten

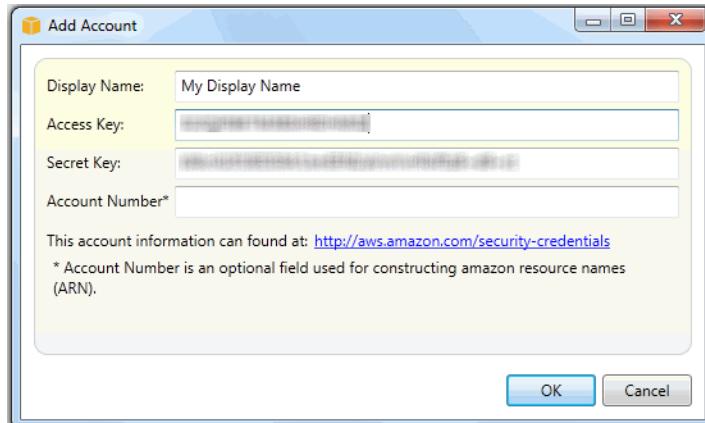
Verwenden Sie das AWS Toolkit for Visual Studio zum Hinzufügen, Bearbeiten und Löschen von Konten, wenn Sie verschiedene AWS-Konten einrichten möchten, um unterschiedliche Aufgaben auszuführen, wie z. B. Tests, Staging und Produktion.

So verwalten Sie mehrere Konten

1. Klicken Sie in Visual Studio im Menü View (Anzeigen) auf AWS Explorer.
2. Neben der Liste Account (Konto) klicken Sie auf die Schaltfläche Add Account (Konto hinzufügen).



Das Dialogfeld Add Account (Konto hinzufügen) wird angezeigt.



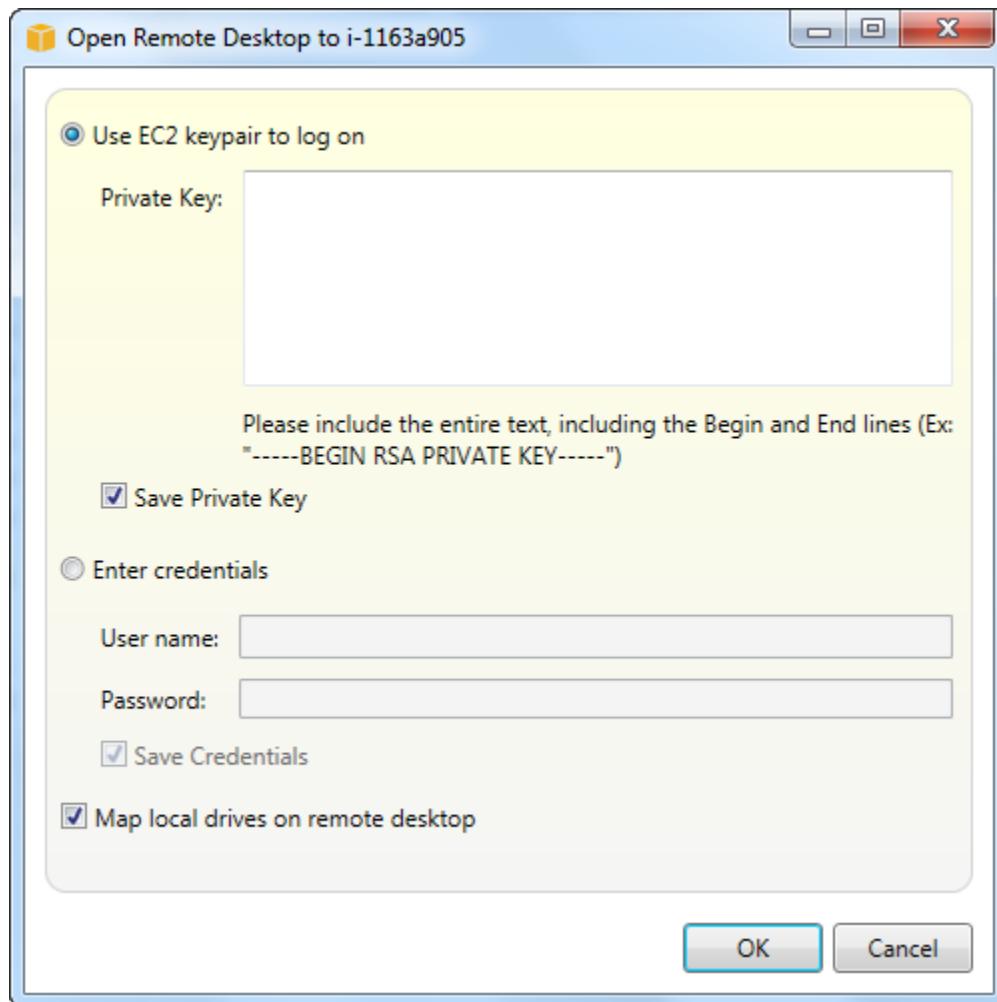
3. Geben Sie die erforderlichen Informationen ein.
4. Ihre Kontoinformationen werden nun auf der Registerkarte AWS Explorer angezeigt. Wenn Sie eine Veröffentlichung in Elastic Beanstalk durchführen, können Sie auswählen, welches Konto Sie verwenden möchten.

Auflisten von Server-Instances/Verbinden mit Server-Instances

Sie können im AWS Toolkit for Visual Studio oder in der AWS-Managementkonsole eine Liste der Amazon EC2-Instances anzeigen, auf denen die Elastic Beanstalk-Anwendungsumgebung ausgeführt wird. Die Verbindung zu diesen Instances können Sie mit Remote Desktop Connection herstellen. Weitere Informationen zum Auflisten von Server-Instances und zum Herstellen einer Verbindung zu diesen mit der AWS Management Console finden Sie unter [Auflisten von Server-Instances/Verbinden mit Server-Instances \(p. 866\)](#). Der folgende Abschnitt führt Sie durch das Anzeigen und Herstellen einer Verbindung mit Ihren Server-Instances mit dem AWS Toolkit for Visual Studio.

So zeigen Sie Amazon EC2-Instances für eine Umgebung an und stellen eine Verbindung zu diesen her

1. Erweitern Sie in Visual Studio unter AWS Explorer den Amazon EC2-Knoten und doppelklicken Sie dann auf Instances.
2. Klicken Sie mit der rechten Maustaste auf die Instance-ID der Amazon EC2-Instance, die vom Load Balancer Ihrer Anwendung in der Spalte Instance verwendet wird, und wählen Sie Open Remote Desktop (Remote Desktop öffnen) im Kontextmenü aus.



3. Wählen Sie Use EC2 keypair to log on (EC2-Schlüsselpaar für die Anmeldung verwenden) aus und fügen Sie den Inhalt der Datei mit dem privaten Schlüssel, die Sie für die Bereitstellung der Anwendung verwendet haben, in das Feld Private key (Privater Schlüssel) ein. Alternativ geben Sie Ihren Benutzernamen und Ihr Passwort in die Textfelder User name (Benutzername) und Password (Passwort) ein.

Note

Wenn das Schlüsselpaar im Toolkit gespeichert ist, wird das Textfeld nicht angezeigt.

4. Klicken Sie auf OK.

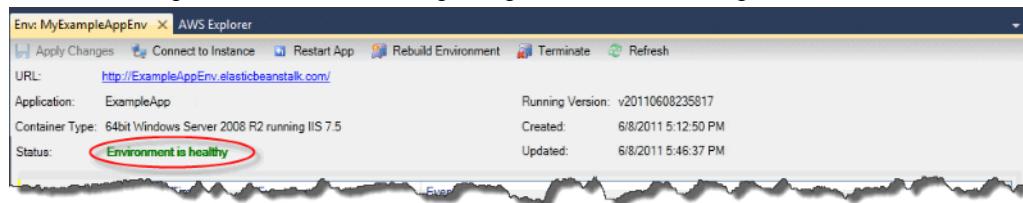
Überwachen des Anwendungsstatus

Bei Ausführung einer Produktionswebsite ist es wichtig zu wissen, dass die Anwendung verfügbar ist und auf Anforderungen reagiert. Für die Überwachung der Reaktionsfähigkeit der Anwendung können Sie die Elastic Beanstalk-Funktionen nutzen, mit denen Sie Anwendungsstatistiken überwachen und Warnungen erstellen können, die bei einer Überschreitung der Schwellenwerte ausgelöst werden.

Weitere Informationen über die Integritätsüberwachung von Elastic Beanstalk finden Sie unter [Grundlegende Zustandsberichte \(p. 821\)](#).

Sie können auf operative Informationen zu Ihrer Anwendung zugreifen, indem Sie das AWS Toolkit for Visual Studio oder die AWS-Managementkonsole verwenden.

Das Toolkit zeigt den Zustand Ihrer Umgebung und den Anwendungsstatus im Feld Status an.



So überwachen Sie den Anwendungsstatus

1. Erweitern Sie im AWS Toolkit for Visual Studio in AWS Explorer den Elastic Beanstalk-Knoten und erweitern Sie dann Ihren Anwendungsknoten.
2. Klicken Sie mit der rechten Maustaste auf Ihre Elastic Beanstalk-Umgebung und anschließend auf View Status (Status anzeigen).
3. Klicken Sie auf der Registerkarte der Anwendungsumgebung auf Monitoring (Überwachung).

Der Bereich Monitoring (Überwachung) enthält eine Reihe von Diagrammen, die die Ressourcennutzung für Ihre bestimmte Anwendungsumgebung zeigen.



Note

Standardmäßig ist der Zeitraum auf der letzten Stunde gesetzt. Ändern Sie diese Einstellung, indem Sie in der Liste Time Range (Zeitraum) auf einen anderen Zeitraum klicken.

Sie können das AWS Toolkit for Visual Studio oder die AWS-Managementkonsole verwenden, um Ereignisse im Zusammenhang mit Ihrer Anwendung anzuzeigen.

So zeigen Sie Anwendungsereignisse an

1. Erweitern Sie im AWS Toolkit for Visual Studio in AWS Explorer den Elastic Beanstalk-Knoten und Ihren Anwendungsknoten.
2. Klicken Sie in AWS Explorer mit der rechten Maustaste auf Ihre Elastic Beanstalk-Umgebung und anschließend auf View Status (Status anzeigen).
3. Klicken Sie in der Registerkarte der Anwendungsumgebung auf Events (Ereignisse).

Events	Filter:			
	Event Time	Event Type	Version Label	Event Details
Monitoring	12/2/2011 3:43:19 PM	INFO	v20111202232102	Environment update completed successfully.
Resources	12/2/2011 3:43:19 PM	INFO	v20111202232102	New application version was deployed to running EC2 instances.
Server	12/2/2011 3:43:06 PM	INFO	v20111202232102	Waiting for 2 seconds while EC2 instances download the updated application version.
Load Balancer	12/2/2011 3:43:04 PM	INFO	v20111202232102	Deploying version v20111202232102 to 1 instance(s).
	12/2/2011 3:42:59 PM	INFO	v2011120223009	Environment update is starting.
	12/2/2011 3:40:38 PM	INFO	v2011120223009	Environment health has transitioned from RED to GREEN
Auto Scaling	12/2/2011 3:29:37 PM	WARN	v2011120223009	Environment health has been set to RED
Notifications	12/2/2011 3:28:35 PM	INFO	v2011120223009	Launched environment: MyExampleAppEnv. However, there were issues during launch. See event log for details.
Container	12/2/2011 3:19:13 PM	INFO	v2011120223009	Exceeded maximum amount of time to wait for the application to become available. Setting environment Ready.
Advanced	12/2/2011 3:18:50 PM	INFO	v2011120223009	Added EC2 instance i-93d6e5f0 to Auto Scaling Group awseb-MyExampleAppEnv-5y33OGVvOm.
	12/2/2011 3:18:47 PM	INFO	v2011120223009	An EC2 instance has been launched. Waiting for it to be added to Auto Scaling...
	12/2/2011 3:18:34 PM	INFO	v2011120223009	Waiting for an EC2 instance to be launched...
	12/2/2011 3:18:33 PM	INFO	v2011120223009	Adding Auto Scaling Group 'awseb-MyExampleAppEnv-5y33OGVvOm' to your environment.
	12/2/2011 3:18:33 PM	INFO	v2011120223009	Added URLCheck healthcheck for 'http://MyExampleAppEnv.elasticbeanstalk.com:80'
	12/2/2011 3:18:31 PM	INFO	v2011120223009	Created Auto Scaling trigger named: awseb-MyExampleAppEnv-5y33OGVvOm.
	12/2/2011 3:18:30 PM	INFO	v2011120223009	Created Auto Scaling group named: awseb-MyExampleAppEnv-5y33OGVvOm.
	12/2/2011 3:18:30 PM	INFO	v2011120223009	Created Auto Scaling launch configuration named: awseb-MyExampleAppEnv-JDwosdTJA.
	12/2/2011 3:18:29 PM	INFO	v2011120223009	Created load balancer named: awseb-MyExampleAppEnv.
	12/2/2011 3:18:28 PM	INFO	v2011120223009	Created security group named: elasticbeanstalk-windows.
	12/2/2011 3:18:28 PM	INFO	v2011120223009	Using elasticbeanstalk-us-east-1-049020475370 as Amazon S3 storage bucket for environment data.

Bereitstellen von Elastic Beanstalk-Anwendungen in .NET mit dem Bereitstellungstool

Das AWS Toolkit for Visual Studio enthält ein Bereitstellungstool in Form eines Befehlszeilen-Tools, das dieselbe Funktionalität bietet wie der Bereitstellungsassistent im AWS Toolkit. Sie können das Bereitstellungstool in Ihrer Build-Pipeline oder in anderen Skripts verwenden, um Bereitstellungen in Elastic Beanstalk zu automatisieren.

Das Bereitstellungstool unterstützt sowohl erste als auch erneute Bereitstellungen. Wenn Sie zuvor das Bereitstellungstool für die Anwendungsbereitstellung verwendet haben, können Sie diese mit dem Bereitstellungsassistenten von Visual Studio erneut bereitstellen. Ebenso gilt, wenn Sie den Assistenten für die Bereitstellung genutzt haben, können Sie das Bereitstellungstool für die erneute Bereitstellung nutzen.

Note

Das Bereitstellungstool übernimmt keine [empfohlenen Werte \(p. 648\)](#) für Konfigurationsoptionen (im Gegensatz zur Konsole oder der EB CLI). Verwenden Sie [Konfigurationsdateien \(p. 722\)](#), um sicherzustellen, dass alle erforderlichen Einstellungen beim Umgebungsstart konfiguriert sind.

In diesem Kapitel werden die Schritte für die Bereitstellung einer .NET-Beispielanwendung in Elastic Beanstalk mithilfe des Bereitstellungstools sowie für die erneute Anwendungsbereitstellung mit einer inkrementellen Bereitstellung geschrieben. Eine umfassende Beschreibung des Bereitstellungstools einschließlich der Parameteroptionen finden Sie unter [Deployment Tool \(Bereitstellungstool\)](#).

Voraussetzungen

Um das Bereitstellungstool nutzen zu können, müssen Sie das AWS Toolkit for Visual Studio installieren. Weitere Informationen zu Voraussetzungen sowie die Installationsanleitung finden Sie unter [AWS Toolkit for Microsoft Visual Studio](#).

Das Bereitstellungstool wird in der Regel in einem der folgenden Windows-Verzeichnisse installiert:

32-Bit	64-Bit
C:\Program Files\AWS Tools \Deployment Tool\awsdeploy.exe	C:\Program Files (x86)\AWS Tools\Deployment Tool \awsdeploy.exe

Bereitstellen auf Elastic Beanstalk

Um die Beispielanwendung mit dem Bereitstellungstool in Elastic Beanstalk bereitzustellen, müssen Sie zunächst die Konfigurationsdatei `ElasticBeanstalkDeploymentSample.txt` im Verzeichnis `Samples` ändern. In dieser Konfigurationsdatei sind die erforderlichen Informationen für die Anwendungsbereitstellung enthalten, darunter der Anwendungsname, die Anwendungsversion, der Umgebungsname und die AWS-Anmeldeinformationen. Nachdem Sie die Konfigurationsdatei geändert haben, stellen Sie Beispielanwendung über die Befehlszeile bereit. Die Web Deploy-Datei wird in Amazon S3 hochgeladen und als neue Anwendungsversion in Elastic Beanstalk registriert. Die Anwendungsbereitstellung dauert einige Minuten. Sobald die Umgebung stabil läuft, gibt das Bereitstellungstool eine URL für die ausgeführte Anwendung aus.

So stellen Sie eine .NET-Anwendung in Elastic Beanstalk bereit

1. Öffnen Sie im Unterverzeichnis `Samples`, in dem das Bereitstellungstool installiert ist, die Datei `ElasticBeanstalkDeploymentSample.txt` und geben Sie Ihren AWS-Zugriffsschlüssel und den geheimen AWS-Schlüssel wie in folgendem Beispiel ein.

```
### AWS Access Key and Secret Key used to create and deploy the application instance
AWSAccessKey = AKIAIOSFODNN7EXAMPLE
AWSSecretKey = wJalrXUtnFEMI/K7MDENG/bPxRfCYEXAMPLEKEY
```

Note

Für den API-Zugriff benötigen Sie eine Zugriffsschlüssel-ID und einen geheimen Zugriffsschlüssel. Verwenden Sie IAM-Benutzerzugriffsschlüssel anstelle von Root-Benutzerzugriffsschlüsseln für das AWS-Konto. Weitere Information über die Erstellung von Zugriffsschlüsseln finden Sie unter [Verwalten von Zugriffsschlüsseln für IAM-Benutzer](#) im IAM-Benutzerhandbuch.

2. Geben Sie in der Befehlszeile Folgendes ein:

```
C:\Program Files (x86)\AWS Tools\Deployment Tool>awsdeploy.exe /w Samples
\ElasticBeanstalkDeploymentSample.txt
```

Die Anwendungsbereitstellung dauert einige Minuten. Bei einer erfolgreichen Bereitstellung wird die Nachricht `Application deployment completed; environment health is Green` angezeigt.

Note

Wird der folgende Fehler angezeigt, ist der CNAME bereits vorhanden.

```
[Error]: Deployment to AWS Elastic Beanstalk failed with exception: DNS name
(MyAppEnv.elasticbeanstalk.com) is not available.
```

Da der CNAME eindeutig sein muss, ist eine Änderung von `Environment.CNAME` in `ElasticBeanstalkDeploymentSample.txt` erforderlich.

3. Rufen Sie im Webbrowser die URL der ausgeführten Anwendung auf. Die URL weist das Format `<CNAME.elasticbeanstalk.com>` (z. B. **MyAppEnv.elasticbeanstalk.com**) auf.

Migrieren Ihrer lokalen .NET-Anwendung zu Elastic Beanstalk

Wenn Sie darüber nachdenken, Ihre .NET-Anwendung von lokalen Servern auf Amazon Web Services (AWS) zu migrieren, könnte der .NET Migration Assistant for AWS Elastic Beanstalk für Sie nützlich sein. Der Assistent ist ein interaktives PowerShell-Dienstprogramm, das eine .NET-Anwendung von Windows Server auf AWS Elastic Beanstalk migriert, wobei IIS lokal ausgeführt wird. Der Assistent kann eine ganze Website auf Elastic Beanstalk migrieren, wobei nur minimale oder gar keine Änderungen vorgenommen werden müssen.

Weitere Informationen zum .NET-Migrationsassistenten für AWS Elastic Beanstalk und darüber, wie er heruntergeladen wird, finden Sie im Repository <https://github.com/awslabs/windows-web-app-migration-assistant> auf GitHub.

Wenn Ihre Anwendung Microsoft SQL Server-Datenbanken enthält, enthält die Dokumentation des Assistenten auf GitHub mehrere Optionen für die Migration.

Ressourcen

Es gibt zahlreiche Möglichkeiten, um zusätzliche Unterstützung bei der Entwicklung von .NET-Anwendungen zu erhalten:

Ressource	Beschreibung
.NET Development Forum	Hier können Sie Fragen stellen und Feedback erhalten.
.NET Developer Center	Umfangreiches Kit mit Beispiel-Code, Dokumentation, Tools und zusätzlichen Ressourcen.
AWS SDK für .NET – Dokumentation	Lesen Sie über das Einrichten des SDK und Ausführen der Codebeispiele, Funktionen des SDK und detaillierte Informationen über API-Vorgänge für das SDK.

Bereitstellen von Node.js-Anwendungen in Elastic Beanstalk

Themen

- [Erste Schritte mit Node.js in Elastic Beanstalk \(p. 248\)](#)
- [Einrichten Ihrer Node.js-Entwicklungsumgebung \(p. 249\)](#)
- [Verwenden der Elastic Beanstalk-Node.js-Plattform \(p. 250\)](#)
- [Bereitstellen einer Express-Anwendung für Elastic Beanstalk \(p. 260\)](#)
- [Bereitstellen einer Express-Anwendung mit Clustering in Elastic Beanstalk \(p. 264\)](#)
- [Bereitstellen einer Node.js-Anwendung mit DynamoDB in Elastic Beanstalk \(p. 273\)](#)
- [Hinzufügen einer Amazon RDS-DB-Instance zur Node.js-Anwendungsumgebung \(p. 282\)](#)
- [Ressourcen \(p. 285\)](#)

AWS Elastic Beanstalk for Node.js vereinfacht das Bereitstellen, Verwalten und Skalieren Ihrer Node.js-Webanwendungen mithilfe von Amazon Web Services. AWS Elastic Beanstalk for Node.js ist für jede Person verfügbar, der eine Webanwendung mit Node.js bereitstellt oder hostet. Dieses Kapitel enthält Schritt-für-Schritt-Anleitungen für das Bereitstellen Ihrer Node.js-Webanwendung in Elastic Beanstalk.

mithilfe der Elastic Beanstalk-Managementkonsole und bietet Anleitungen für gängige Aufgaben, wie z. B. Datenbankintegration und Arbeiten mit dem Express-Framework.

Nach der Bereitstellung der Elastic Beanstalk-Anwendung können Sie weiter die EB-CLI für die Verwaltung von Anwendung und Umgebung nutzen oder die Elastic Beanstalk-Konsole, die AWS-CLI oder die APIs verwenden.

In den Themen in diesem Kapitel wird davon ausgegangen, dass Sie über einige Kenntnisse der Elastic Beanstalk-Umgebungen verfügen. Wenn Sie Elastic Beanstalk noch nicht verwendet haben, machen Sie sich im [Tutorial "Erste Schritte" \(p. 3\)](#) mit den Grundlagen vertraut.

Erste Schritte mit Node.js in Elastic Beanstalk

Für die ersten Schritte mit Node.js-Anwendungen in AWS Elastic Beanstalk benötigen Sie nur das [Quell-Bundle \(p. 412\)](#) einer Anwendung, damit Sie die erste Anwendungsversion hochladen und in einer Umgebung bereitstellen können. Im Rahmen der Umgebungserstellung werden von Elastic Beanstalk alle erforderlichen AWS-Ressourcen für eine hoch skalierbare Webanwendung zugewiesen.

Starten einer Umgebung mit einer Node.js-Beispielanwendung

Elastic Beanstalk bietet für jede Plattform sowohl einfache Beispielanwendungen mit einer Seite als auch komplexe Beispiele, in denen die Nutzung weiterer AWS-Ressourcen (wie z. B. Amazon RDS; sowie sprach- oder plattformspezifische Funktionen und APIs) veranschaulicht wird.

Beispiele

Umgebungstyp	Quell-Bundle	Beschreibung	
Webserver	nodejs.zip	Anwendung mit einer Seite. Verwenden Sie das Verfahren unter Create an Example Application (p. 3) , um dieses Beispiel zu starten.	
Webserver mit Amazon RDS	nodejs-express-hiking-v1.zip	Hiking-Protokollanwendung, die das Express-Framework und eine RDS-Datenbank nutzt. Praktische Anleitung (p. 260)	
Webserver mit Amazon ElastiCache	nodejs-example-express-elasticache.zip	Express-Webanwendung, die Amazon ElastiCache für das Clustering verwendet. Mit Clustering lassen sich Hochverfügbarkeit, Leistung und Sicherheit Ihrer Webanwendung verbessern. Praktische Anleitung (p. 264)	
Webserver mit DynamoDB, Amazon SNS und Amazon SQS	eb-node-express-sample-v1.0.zip Clone the repo at GitHub.com (Das Repository klonen unter GitHub.com)	Express-Website, die Kontaktinformationen der Benutzer für eine neue Marketingkampagne des Unternehmens sammelt. Verwendet das AWS SDK für JavaScript in Node.js, um Einträge in eine DynamoDB-Tabelle zu schreiben, und Elastic Beanstalk-Konfigurationsdateien zum Erstellen von Ressourcen in DynamoDB, Amazon SNS und Amazon SQS. Praktische Anleitung (p. 273)	

Nächste Schritte

Nachdem Sie eine Umgebung erstellt haben, in der eine Anwendung ausgeführt wird, können Sie jederzeit eine neue Version der Anwendung oder eine gänzlich andere Anwendung bereitstellen. Das Bereitstellen einer neuen Anwendungsversion geht sehr schnell, da keine EC2-Instances bereitgestellt oder neu gestartet werden müssen. Weitere Informationen zur Anwendungsbereitstellung finden Sie unter [Bereitstellen einer neuen Version Ihrer Anwendung \(p. 7\)](#).

Wenn Sie ein oder zwei Beispianwendungen bereitgestellt haben, können Sie mit der lokalen Entwicklung und Ausführung von Node.js-Anwendungen beginnen. Die entsprechenden Informationen zum Einrichten einer Node.js-Entwicklungsumgebung und zu allen benötigten Tools finden Sie im [nächsten Abschnitt \(p. 249\)](#).

Einrichten Ihrer Node.js-Entwicklungsumgebung

Richten Sie eine Node.js-Entwicklungsumgebung ein, um Ihre Anwendung vor der Bereitstellung in AWS Elastic Beanstalk lokal zu testen. In diesem Thema finden Sie die Schritte zum Einrichten der Entwicklungsumgebung sowie Links zu den Installationsseiten nützlicher Tools.

Allgemeine Einrichtungsschritte und Tools für alle Sprachen finden Sie unter [Konfigurieren des Entwicklungscomputers \(p. 1005\)](#).

Themen

- [Installieren von Node.js \(p. 249\)](#)
- [Installieren von npm \(p. 249\)](#)
- [Installieren des AWS SDK for Node.js \(p. 249\)](#)
- [Installieren von Express \(p. 249\)](#)

Installieren von Node.js

Installieren Sie Node.js, um Node.js-Anwendungen lokal auszuführen. Falls Sie keine Version bevorzugen, wählen Sie die neueste, von Elastic Beanstalk unterstützte Version aus. Eine Liste der unterstützten Versionen finden Sie unter [Node.js](#) im Dokument AWS Elastic Beanstalk-Plattformen.

Laden Sie Node.js unter [nodejs.org](#) herunter.

Installieren von npm

Node.js verwendet einen npm Package Manager, mit dem Sie Tools und Frameworks für die Verwendung in Ihrer Anwendung installieren können. Laden Sie npm unter [npmjs.com](#) herunter.

Installieren des AWS SDK for Node.js

Wenn Sie die AWS-Ressourcen aus der Anwendung heraus verwalten möchten, installieren Sie das AWS SDK for JavaScript in Node.js. Installieren Sie das SDK mit npm:

```
$ npm install aws-sdk
```

Besuchen Sie die Homepage [AWS SDK for JavaScript in Node.js](#), um weitere Informationen zu erhalten.

Installieren von Express

Express ist ein Framework für Webanwendungen, das auf Node.js ausgeführt wird. Richten Sie Express ein und erstellen Sie die Projektstruktur, um sie zu verwenden. In den folgenden Schritten wird erläutert, wie Sie Express auf einem Linux-Betriebssystem einrichten.

Note

Je nach Berechtigungsstufe für die Systemverzeichnisse müssen Sie möglicherweise einige dieser Befehle mit dem Präfix `sudo` versehen.

So richten Sie die Express-Entwicklungsumgebung auf dem lokalen Computer ein

1. Erstellen Sie ein Verzeichnis für die Express-Anwendung.

```
~$ mkdir node-express  
~$ cd node-express
```

2. Installieren Sie Express global, sodass Sie Zugriff auf den `express`-Befehl haben.

```
~/node-express$ npm install -g express-generator
```

3. Je nach Betriebssystem müssen Sie möglicherweise einen Pfad zur Ausführung des `express`-Befehls festlegen. Ist das der Fall, ziehen Sie die Ausgabe des vorherigen Schritts zur Express-Installation heran. Im Folgenden wird ein Beispiel gezeigt.

```
~/node-express$ export PATH=$PATH:/usr/local/share/npm/bin/express
```

4. Führen Sie den Befehl `express` aus. Dadurch werden `package.json`, `app.js` und ein paar Verzeichnisse generiert.

```
~/node-express$ express
```

Wenn Sie gefragt werden, ob Sie fortfahren möchten, geben Sie `y` ein.

5. Richten Sie lokale Abhängigkeiten ein.

```
~/node-express$ npm install
```

6. Überprüfen Sie, ob dies funktioniert.

```
~/node-express$ npm start
```

Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

```
> nodejs@0.0.0 start /home/local/user/node-express  
> node ./bin/www
```

Der Server wird standardmäßig auf Port 3000 ausgeführt. Um ihn zu testen, führen Sie `curl http://localhost:3000` in einem anderen Terminal aus, oder öffnen Sie einen Browser auf dem lokalen Computer und navigieren Sie zu `http://localhost:3000`.

Drücken Sie Strg+C, um den Server zu stoppen.

Verwenden der Elastic Beanstalk-Node.js-Plattform

Important

Amazon Linux 2-Plattformversionen unterscheiden sich grundlegend von Amazon Linux AMI-Plattformversionen (Vorgängerversion von Amazon Linux 2). Diese unterschiedlichen Plattformgenerationen sind auf verschiedene Arten inkompatibel. Wenn Sie zu einer Amazon

Linux 2-Plattformversion migrieren, müssen Sie die Informationen in [the section called "Aktualisieren auf Amazon Linux 2" \(p. 503\)](#) lesen.

Die AWS Elastic Beanstalk-Node.js-Plattform besteht aus einer Reihe von [Plattformversionen](#) für Node.js-Webanwendungen, die hinter einem nginx-Proxy-Server ausgeführt werden.

Elastic Beanstalk bietet [Konfigurationsoptionen \(p. 646\)](#), mit denen sich die auf den EC2-Instances in der Elastic Beanstalk-Umgebung ausgeführte Software anpassen lässt. Sie können die für Ihre Anwendung erforderlichen Umgebungsvariablen konfigurieren, die Protokollrotation an Amazon S3 aktivieren und die Ordner in der Anwendungsquelle, die statische Dateien enthalten, den vom Proxy-Server verwendeten Pfaden zuordnen.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das [Ändern der Konfiguration einer ausgeführten Umgebung verfügbar \(p. 658\)](#). Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie [gespeicherte Konfigurationen \(p. 764\)](#) verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie [Konfigurationsdateien \(p. 722\)](#) einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Sie können eine [Datei Package.json in Ihr Quell-Bundle aufnehmen \(p. 256\)](#), um Pakete während der Bereitstellung zu installieren, einen Startbefehl bereitzustellen und die Node.js-Version anzugeben, die Ihre Anwendung verwenden soll. Sie können eine [-Datei npm-shrinkwrap.json \(p. 257\)](#) aufnehmen, um Abhängigkeitsversionen zu sperren.

Die Node.js-Plattform umfasst einen Proxy-Server, um statische Komponenten zu verarbeiten, Datenverkehr an Ihre Anwendung weiterzuleiten und Antworten zu komprimieren. Sie können die für erweiterte Szenarien [Standard-Proxy-Konfiguration erweitern oder überschreiben \(p. 257\)](#).

Sie können Ihrem Quellpaket ein [Procfile](#) hinzufügen, um den Befehl anzugeben, mit dem die Anwendung gestartet wird, wie im folgenden Beispiel gezeigt. Diese Funktion ersetzt die ältere Option `NodeCommand` im Namespace `aws:elasticbeanstalk:container:nodejs`.

Example Procfile

```
web: node index.js
```

Details zur [Procfile-Verwendung erhalten Sie, indem Sie den Abschnitt Buildfile und Procfile in the section called "Erweitern von Linux-Plattformen" \(p. 34\) erweitern.](#)

Wenn Sie kein [Procfile](#) bereitstellen, führt Elastic Beanstalk `npm start` aus, wenn Sie eine `package.json`-Datei verfügbar machen. Wenn Sie auch dies nicht angeben, sucht Elastic Beanstalk die Dateien `app.js` oder `server.js` (in dieser Reihenfolge) und führt sie aus.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, wenn vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe [Konfigurationsoptionen \(p. 646\)](#).

Details zu den verschiedenen Möglichkeiten zum Erweitern einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter [the section called "Erweitern von Linux-Plattformen" \(p. 34\)](#).

Konfigurieren der Node.js-Umgebung

Mit den Einstellungen der Node.js-Plattform können Sie das Verhalten Ihrer Amazon EC2-Instances optimieren. Sie können die Konfiguration der Amazon EC2-Instance der Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-Konsole bearbeiten.

Mithilfe der Elastic Beanstalk-Konsole aktivieren Sie die Protokollrotation an Amazon S3 und konfigurieren Variablen, die die Anwendung aus der Umgebung auslesen kann.

So konfigurieren Sie Ihre Node.js -Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.

Containeroptionen

Sie können die folgenden plattformspezifischen Optionen angeben:

- Proxy server (Proxy-Server) – Der Proxy-Server, der in Ihren Umgebungs-Instances verwendet werden soll. Standardmäßig wird nginx verwendet.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) –Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren): gibt an, ob die Protokolldateien für die Amazon EC2-Instances der Anwendung in den zugeordneten Amazon S3-Bucket kopiert werden sollen.

Statische Dateien

Um die Leistung zu verbessern, können Sie im Abschnitt Static files (Statische Dateien) den Proxy-Server so konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zur Verzeichniszuordnung fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, statt die Anfrage an Ihre Anwendung weiterzuleiten.

Weitere Informationen zum Konfigurieren statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter [the section called “Statische Dateien” \(p. 776\)](#).

Umgebungseigenschaften

Im Bereich Environment Properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Diese Einstellungen werden als Schlüssel-Wert-Paare an die Anwendung weitergeleitet.

Innerhalb der Node.js-Umgebung, die in AWS Elastic Beanstalk ausgeführt wird, können Sie die Umgebungsvariablen mit `process.env.ENV_VARIABLE` wie im folgenden Beispiel hinzufügen.

```
var endpoint = process.env.API_ENDPOINT
```

Die Node.js-Plattform legt die PORT-Umgebungsvariable auf den Port fest, an den der Proxy-Server den Datenverkehr weiterleitet. Siehe [Konfigurieren des Proxy-Servers \(p. 257\)](#).

Weitere Informationen finden Sie unter [Umgebungseigenschaften und andere Softwareeinstellungen \(p. 621\)](#).

Konfigurieren eines Amazon Linux AMI Node.js-Umgebung (Vorgängerversion von Amazon Linux 2)

Die folgenden Konfigurationskategorien für Konsolensoftware werden nur in einer Elastic Beanstalk Node.js-Umgebung unterstützt, die eine Amazon Linux AMI-Plattformversion (vor Amazon Linux 2) verwendet.

Containeroptionen

Geben Sie auf der Konfigurationsseite Folgendes ein:

- Proxy Server (Proxy-Server) – Gibt den Webserver an, der für Proxy-Verbindungen zu Node.js verwendet werden soll. Standardmäßig wird nginx verwendet. Wenn Sie none auswählen, sind statische Dateizuordnungen nicht wirksam und die gzip-Komprimierung ist deaktiviert.
- Node.js version (Version von Node.js) – Gibt die Version von Node.js an. Eine Liste der unterstützten Versionen von Node.js finden Sie unter [Node.js](#) im Handbuch AWS Elastic Beanstalk-Plattformen.
- gzip compression (gzip-Komprimierung) – Gibt an, ob die gzip-Komprimierung aktiviert ist. Die gzip-Komprimierung ist standardmäßig aktiviert.
- Node Command (Knotenbefehl) – Hier können Sie den Befehl eingeben, der zum Starten der Node.js-Anwendung verwendet wird. Eine leere Zeichenfolge (Standard) bedeutet, dass Elastic Beanstalk erst `app.js`, dann `server.js` und dann `npm start` verwenden wird.

Node.js-Konfigurations-Namespace

Mithilfe einer [Konfigurationsdatei \(p. 722\)](#) können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Konfigurationsoptionen können durch den Elastic Beanstalk-Service oder die verwendete Plattform definiert und in Namespaces organisiert werden.

Sie können mittels des Namespace `aws:elasticbeanstalk:environment:proxy` den Proxy auswählen, der für die Instances Ihrer Umgebung verwendet werden soll. Im folgenden Beispiel wird Ihre Umgebung für die Verwendung des Apache HTTPD-Proxy-Servers konfiguriert.

Example `.ebextensions/nodejs-settings.config`

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

Sie können den Proxy so konfigurieren, dass statische Dateien mittels des Namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` bereitgestellt werden. Einzelheiten und ein Beispiel finden Sie unter [the section called “Statische Dateien” \(p. 776\)](#).

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Die Amazon Linux AMI Node.js-Plattform (vor Amazon Linux 2)

Wenn Ihre Elastic Beanstalk Node.js Umgebung eine Amazon Linux AMI-Plattformversion verwendet (Vorgängerversion von Amazon Linux 2), lesen Sie die zusätzlichen Informationen in diesem Abschnitt.

Node.js-plattformspezifische Konfigurationsoptionen

Elastic Beanstalk unterstützt einige plattformspezifische Konfigurationsoptionen für Amazon Linux AMI-Node.js-Plattformversionen. Sie können wählen, welchen Proxy-Server Sie vor Ihrer Anwendung ausführen möchten, eine bestimmte Version von Node.js für die Ausführung verwenden und den Befehl wählen, um Ihre Anwendung auszuführen.

Für Proxy-Server können Sie zusätzlich zu nginx den Apache-Proxy-Server wählen. Darüber hinaus können Sie den Wert `none` auf die Option `ProxyServer` setzen. In diesem Fall führt Elastic Beanstalk Ihre Anwendung als eigenständige Anwendung aus und nicht hinter einem Proxy-Server. Wenn in Ihrer Umgebung eine eigenständige Anwendung ausgeführt wird, aktualisieren Sie den Code so, dass dem Port zugehört wird, an den nginx den Datenverkehr weiterleitet.

```
var port = process.env.PORT || 8080;  
  
app.listen(port, function() {  
    console.log('Server running at http://127.0.0.1:%s', port);  
});
```

Node.js-Sprachversionen

In Bezug auf die unterstützte Sprachversion unterscheidet sich die Amazon Linux AMI-Node.js-Plattform geringfügig von anderen von Elastic Beanstalk verwalteten Plattformen. Jede Node.js-Plattformversion unterstützt verschiedene Node.js-Sprachversionen. Eine Liste der unterstützten Versionen von Node.js finden Sie unter [Node.js](#) im Handbuch AWS Elastic Beanstalk-Plattformen.

Sie können eine plattformspezifische Konfigurationsoption verwenden, um die Sprachversion festzulegen. Details dazu finden Sie unter [the section called “Konfigurieren der Node.js-Umgebung” \(p. 251\)](#). Alternativ können Sie –wie im folgenden Verfahren gezeigt – die Elastic Beanstalk-Konsole verwenden, um die Node.js-Version zu aktualisieren, die Ihre Umgebung als Teil der Aktualisierung Ihrer Plattformversion verwendet.

Note

Wenn die von Ihnen verwendete Version von Node.js von der Plattform nicht mehr unterstützt wird, müssen Sie die Versionseinstellung vor der Ausführung einer [Plattformaktualisierung \(p. 491\)](#) ändern oder löschen. Dieser Fall kann eintreten, wenn eine Sicherheitslücke für eine oder mehrere Versionen von Node.js identifiziert wird. In diesem Fall tritt bei dem Versuch, ein Upgrade auf eine neue Plattformversion, die die konfigurierte [NodeVersion \(p. 716\)](#) nicht unterstützt, ein Fehler auf. Um keine neue Umgebung erstellen zu müssen, ändern Sie die Konfigurationsoption `NodeVersion` in eine Node.js-Version, die sowohl von der alten als auch von der neuen Plattformversion unterstützt wird, oder [entfernen Sie die Optionseinstellung \(p. 658\)](#) und führen Sie dann die Plattformaktualisierung aus.

So konfigurieren Sie die Node.js-Version Ihrer Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Seite „Umgebungsübersicht“ unter Platform (Plattform) die Option Change (Ändern) aus.
4. Wählen Sie im Dialogfeld Update platform version (Plattformversion aktualisieren) eine Node.js-Version aus.

Update platform version

Availability warning

This operation replaces your instances; your application is unavailable during the update. If you have one instance in service during the update, enable rolling updates. Another option is to clone the environment, which creates a newer version of the platform, and then swap the CNAME of the clone when you are ready to deploy the clone. Learn more at [Updating AWS Elastic Beanstalk Environments Using Rolling Updates and Deploying Version with Zero Downtime](#).

Platform branch	Current Node.js version
Node.js running on 64bit Amazon Linux	12.14.0
Current platform version	New Node.js version
4.13.0	12.14.1
New platform version	
4.13.0 (Recommended)	

5. Wählen Sie Save (Speichern) aus.

Node.js-Konfigurations-Namespace

Die Node.js-Amazon Linux AMI-Plattform definiert zusätzliche Optionen in den Namespaces `aws:elasticbeanstalk:container:nodejs:staticfiles` und `aws:elasticbeanstalk:container:nodejs`.

Die folgende Konfigurationsdatei weist Elastic Beanstalk an, die Anwendung mit `npm start` auszuführen, setzt den Proxytyp auf Apache, aktiviert die Komprimierung und konfiguriert den Proxy für die Bereitstellung statischer Dateien aus zwei Quellverzeichnissen: HTML-Dateien über den `html`-Pfad unter dem Stammverzeichnis der Website aus dem `statichtml`-Quellverzeichnis und Bilddateien über den `images`-Pfad unter dem Stammverzeichnis der Website aus dem `staticimages`-Quellverzeichnis.

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
```

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Installieren der Abhängigkeiten Ihrer Anwendung

Ihre Anwendung verfügt möglicherweise über Abhängigkeiten von einigen Node.js-Modulen, wie denen, die Sie in `require()`-Anweisungen angeben. Sie können diese Abhängigkeiten mithilfe einer `package.json`-Datei angeben. Alternativ können Sie die Abhängigkeiten Ihrer Anwendung in das Quell-Bundle einschließen und sie mit der Anwendung bereitstellen. Auf dieser Seite werden diese beiden Möglichkeiten erläutert.

Angeben von Node.js-Abhängigkeiten mit einer `package.json`-Datei

Fügen Sie eine `package.json`-Datei in das Stammverzeichnis Ihrer Projektquelle ein, um Abhängigkeitsspakete anzugeben (verwenden Sie das Schlüsselwort `dependencies`) und einen Startbefehl bereitzustellen (verwenden Sie das Schlüsselwort `scripts`). Mit dem Schlüsselwort `scripts` kann die Legacy-Option `NodeCommand` im Namespace `aws:elasticbeanstalk:container:nodejs` ersetzt werden.

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"
  }
}
```

Wenn eine `package.json`-Datei vorhanden ist, führt Elastic Beanstalk `npm install` aus, um Abhängigkeiten zu installieren. Darüber hinaus wird der Befehl `start` verwendet, um Ihre Anwendung zu starten.

Standardmäßig installiert Elastic Beanstalk Abhängigkeiten im Produktionsmodus (`npm install --production`). Wenn Sie Entwicklungsabhängigkeiten in Ihren Umgebungs-Instances installieren möchten, setzen Sie die [Umgebungseigenschaft \(p. 623\)](#) `NPM_USE_PRODUCTION` auf `false`.

Sie können auch das Schlüsselwort `engines` in der Datei `package.json` verwenden, um die Node.js-Version anzugeben, die Ihre Anwendung verwenden soll, wie im folgenden Beispiel gezeigt. Diese Funktion ersetzt die ältere Option `NodeVersion` im Namespace `aws:elasticbeanstalk:container:nodejs`. Beachten Sie, dass Sie jeweils nur eine Node.js-Version angeben können, die Ihrem Plattformzweig entspricht. Wenn Sie beispielsweise den Plattformzweig Node.js 12 verwenden, können Sie nur eine 12.x.y Node.js-Version angeben. Gültige Versionen von Node.js für die einzelnen Plattformvarianten finden Sie unter [Node.js](#) im Handbuch AWS Elastic Beanstalk-Plattformen.

Example package.json – Node.js-Version

```
{
```

```
...  
  "engines": { "node" : "12.16.1" }  
}
```

Note

Wenn die von Ihnen verwendete Version von Node.js von der Plattform nicht mehr unterstützt wird, müssen Sie die Node.js-Versionseinstellung vor der Ausführung einer [Plattformaktualisierung \(p. 491\)](#) ändern oder löschen. Dieser Fall kann eintreten, wenn eine Sicherheitslücke für eine oder mehrere Versionen von Node.js identifiziert wird. In diesem Fall schlägt ein Updateversuch auf eine neue Plattformversion, die die konfigurierte Node.js-Version nicht unterstützt, fehl. Um zu vermeiden, dass eine neue Umgebung erstellt werden muss, ändern Sie die Node.js-Versionseinstellung in `package.json` in eine Node.js-Version, die sowohl von der alten als auch von der neuen Version unterstützt wird. Sie können die angegebene Node.js-Version auch entfernen und dann das neue Quellpaket bereitstellen. Führen Sie erst dann die Plattform-Aktualisierung durch.

Einschließen von Node.js-Abhängigkeiten in ein node_modules-Verzeichnis

Wenn Sie Abhängigkeitspakte zusammen mit Ihrem Anwendungscode in Umgebungsinstances bereitstellen möchten, fügen Sie sie in das Verzeichnis `node_modules` im Stammverzeichnis der Projektquelle ein. Node.js sucht nach Abhängigkeiten in diesem Verzeichnis. Weitere Informationen finden Sie unter [Laden aus node_modules-Ordnern](#) in der Node.js-Dokumentation.

Note

Wenn Sie ein `node_modules`-Verzeichnis auf einer Amazon Linux 2-Node.js-Plattformversion bereitstellen, geht Elastic Beanstalk davon aus, dass Sie Ihre eigenen Abhängigkeitspakte bereitstellen, und vermeidet die Installation von Abhängigkeiten, die in einer [package.json \(p. 256\)](#)-Datei angegeben sind.

Sperren von Abhängigkeiten mit npm shrinkwrap

Die Node.js-Plattform führt `npm install` bei jeder Bereitstellung aus. Wenn neue Versionen Ihrer Abhängigkeiten verfügbar sind, werden sie installiert, wenn Sie Ihre Anwendung bereitstellen. Dadurch kann die Bereitstellung möglicherweise sehr lange dauern.

Sie können ein Upgrade Ihrer Abhängigkeiten vermeiden, indem Sie eine `npm-shrinkwrap.json`-Datei erstellen, die die Abhängigkeiten Ihrer Anwendung auf die aktuelle Version sperrt.

```
$ npm install  
$ npm shrinkwrap  
wrote npm-shrinkwrap.json
```

Fügen Sie diese Datei in Ihr Quell-Bundle ein, um sicherzustellen, dass Abhängigkeiten nur einmal installiert werden.

Konfigurieren des Proxy-Servers

Elastic Beanstalk verwendet nginx oder Apache HTTPD als Reverse-Proxy-Server, um die Anwendung dem Elastic Load Balancing Load Balancer auf Port 80 zuzuweisen. Standardmäßig wird nginx verwendet. Elastic Beanstalk bietet eine Proxy-Standardkonfiguration, die Sie entweder erweitern oder gänzlich mit einer eigenen Konfiguration überschreiben können.

Standardmäßig wird der Proxy von Elastic Beanstalk für die Weiterleitung von Anforderungen an die Anwendung auf Port 8080 konfiguriert. Diese Port-Standardeinstellung können Sie überschreiben. Legen Sie dazu die [PORT-Umgebungseigenschaft \(p. 251\)](#) auf den Überwachungsport der Hauptanwendung fest.

Hinweise

- Der Überwachungsport der Anwendung kollidiert nicht mit dem Überwachungsport des nginx-Servers, von dem dieser die Anforderungen des Load Balancers empfängt.
- Auf Amazon Linux AMI-Node.js Plattformversionen (vor Amazon Linux 2) konfiguriert Elastic Beanstalk den Proxy-Server so, dass Anforderungen an die Anwendung auf Port 8081 weitergeleitet werden. Weitere Informationen zur Proxy-Konfiguration auf diesen Node.js-Plattformversionen finden Sie unter [the section called “Konfigurieren des Proxys auf Amazon Linux AMI \(Vorgängerversion von Amazon Linux 2\)” \(p. 258\)](#) auf dieser Seite.

Alle Amazon Linux 2-Plattformen unterstützen eine einheitliche Proxy-Konfigurationsfunktion. Details zum Konfigurieren des Proxy-Servers auf den neuen Amazon Corretto-Plattformversionen, die Amazon Linux 2 ausführen, erhalten Sie, wenn Sie den Abschnitt Reverse Proxy-Konfiguration unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#) erweitern.

Konfigurieren des Proxys auf Amazon Linux AMI (Vorgängerversion von Amazon Linux 2)

Wenn die Elastic Beanstalk-Node.js-Umgebung eine Amazon Linux AMI-Plattformversion verwendet (vor Amazon Linux 2), lesen Sie die Informationen in diesem Abschnitt.

Erweitern und Überschreiben der Standard-Proxykonfiguration

Die Node.js-Plattform verwendet einen Reverse-Proxy-Server, um Anforderungen von Port 80 auf der Instance an die Anwendung weiterzuleiten, die Port 8081 überwacht. Elastic Beanstalk bietet eine Proxy-Standardkonfiguration, die Sie entweder erweitern oder gänzlich mit einer eigenen Konfiguration überschreiben können.

Um die Standardkonfiguration zu erweitern, fügen Sie .conf-Dateien /etc/nginx/conf.d mit einer Konfigurationsdatei hinzu. Ein Beispiel finden Sie unter [Beenden von HTTPS auf EC2-Instances, auf denen Node.js ausgeführt wird \(p. 793\)](#).

Die Node.js-Plattform legt die PORT-Umgebungsvariable auf den Port fest, an den der Proxy-Server den Datenverkehr weiterleitet. Lesen Sie diese Variable in Ihrem Code, um den Port Ihrer Anwendung zu konfigurieren.

```
var port = process.env.PORT || 3000;  
  
var server = app.listen(port, function () {  
    console.log('Server running at http://127.0.0.1:' + port + '/');  
});
```

Die standardmäßige nginx-Konfiguration leitet Datenverkehr an einen Upstream-Server mit dem Namen nodejs auf 127.0.0.1:8081 weiter. Sie können die Standardkonfiguration entfernen und Ihre eigene in einer [Konfigurationsdatei \(p. 722\)](#) bereitstellen.

Example .ebextensions/proxy.config

Im folgenden Beispiel wird die Standard-Konfiguration entfernt und eine benutzerdefinierte Konfiguration hinzugefügt, die Datenverkehr an Port 5000 anstelle von 8081 weiterleitet.

```
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"

```

Die Beispielkonfiguration `/etc/nginx/conf.d/proxy.conf` verwendet die Standardkonfiguration unter `/etc/nginx/conf.d/00_elastic_beanstalk_proxy.conf` als Basis, um den Standard-Serverblock mit Komprimierungs- und Protokolleinstellungen und eine statische Dateizuordnung einzuschließen.

Der `removeconfig`-Befehl entfernt die Standardkonfiguration des Containers, um sicherzustellen, dass der Proxy-Server die benutzerdefinierte Konfiguration verwendet. Elastic Beanstalk erstellt die Standardkonfiguration während jeder Konfigurationsbereitstellung neu. Um dies zu berücksichtigen, fügt das Beispiel einen Bereitstellungs-Hook nach der Konfiguration, `/opt/elasticbeanstalk/hooks/configdeploy/post/99_kill_default_nginx.sh`, hinzu, durch die die Standardkonfiguration entfernt und der Proxy-Server neu gestartet wird.

Note

Die Standardkonfiguration kann sich in späteren Versionen der Node.js-Plattform ändern. Verwenden Sie die neueste Version der Konfiguration als Grundlage für Ihre Anpassungen, um Kompatibilität sicherzustellen.

Wenn Sie die Standardkonfiguration überschreiben, müssen Sie alle statischen Dateizuordnungen und die gzip-Komprimierung definieren, da die Plattform die [Standardeinstellungen \(p. 253\)](#) nicht anwenden kann.

Bereitstellen einer Express-Anwendung für Elastic Beanstalk

In diesem Abschnitt wird beschrieben, wie Sie eine Beispielanwendung in Elastic Beanstalk mithilfe der Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI) und Git bereitstellen und anschließend die Anwendung zur Verwendung des [Express](#)-Frameworks aktualisieren.

Voraussetzungen

Dieses Tutorial erfordert die Node.js-Sprache, ihren Package Manager mit der Bezeichnung npm und das Express-Framework für Webanwendungen. Weitere Informationen zur Installation dieser Komponenten und zur Einrichtung Ihrer lokalen Entwicklungsumgebung finden Sie unter [Einrichten Ihrer Node.js-Entwicklungsumgebung \(p. 249\)](#).

Note

Für dieses Tutorial müssen Sie das AWS SDK for Node.js nicht installieren. Dies wird auch in [Einrichten Ihrer Node.js-Entwicklungsumgebung \(p. 249\)](#) erwähnt.

Das Tutorial erfordert auch die Elastic Beanstalk Command Line Interface (EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter [Installieren der EB CLI \(p. 1009\)](#) und [Konfigurieren der EB CLI \(p. 1017\)](#).

Git initialisieren

Die vorausgesetzte Einrichtung von Node.js und Express führt zu einer Express-Projektstruktur im `node-express`-Ordner. Wenn Sie nicht bereits ein Express-Projekt generiert haben, führen Sie den folgenden Befehl aus. Weitere Informationen finden Sie unter [Installieren von Express \(p. 249\)](#).

```
~/node-express$ express && npm install
```

Lassen Sie uns nun ein Git-Repository in diesem Ordner einrichten.

Einrichten eines Git-Repositories

1. Initialisieren Sie das Git-Repository. Wenn Sie Git nicht installiert haben, laden Sie es auf der [Git downloads site](#) herunter.

```
~/node-express$ git init
```

2. Erstellen Sie eine Datei namens `.gitignore` und fügen Sie die folgenden Dateien und Verzeichnisse zu hinzu. Diese Dateien werden ausgeschlossen, damit sie nicht zum Repository hinzugefügt werden. Dieser Schritt ist nicht erforderlich, wird jedoch empfohlen.

```
node-express/.gitignore
```

```
node_modules/  
.gitignore
```

.elasticbeanstalk/

Erstellen einer Elastic Beanstalk-Umgebung

Konfigurieren Sie ein EB CLI-Repository für Ihre Anwendung und erstellen Sie eine Elastic Beanstalk-Umgebung mit der Node.js-Plattform.

1. Erstellen Sie ein Repository mit dem Befehl eb init.

```
~/node-express$ eb init --platform node.js --region us-east-2
Application node-express has been created.
```

Mit diesem Befehl wird eine Konfigurationsdatei in einem Ordner mit dem Namen .elasticbeanstalk generiert, die Einstellungen für die Erstellung von Umgebungen für Ihre Anwendung angibt und eine Elastic Beanstalk-Anwendung mit dem Namen des aktuellen Ordners erstellt.

2. Erstellen Sie eine Umgebung mit einer laufenden Beispielanwendung mit dem eb create-Befehl.

```
~/node-express$ eb create --sample node-express-env
```

Mit diesem Befehl wird eine Umgebung mit Lastausgleich, den Standardeinstellungen für die Node.js-Plattform und den folgenden Ressourcen erstellt:

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die so konfiguriert ist, dass Web-Apps auf der von Ihnen ausgewählten Plattform ausgeführt werden.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance security group (Instance-Sicherheitsgruppe): Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer) – Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe): Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – CloudWatch-Alarne, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen

zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.

- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form `subdomain.region.elasticbeanstalk.com` weiterleitet.
3. Wenn die Umgebungserstellung abgeschlossen ist, verwenden Sie den `eb open`-Befehl, um die URL der Umgebung im Standard-Browser zu öffnen.

```
~/node-express$ eb open
```

Aktualisieren der Anwendung

Nachdem Sie eine Umgebung mit einer Beispielanwendung erstellt haben, können Sie diese mit Ihrer eigenen Anwendung aktualisieren. In diesem Schritt wird die Beispielanwendung zur Nutzung des Express-Frameworks aktualisiert.

So aktualisieren Sie die Anwendung zur Nutzung von Express

1. Erstellen Sie auf dem lokalen Computer ein `.ebextensions`-Verzeichnis im Verzeichnis der obersten Ebene Ihres Quell-Bundles. In diesem Beispiel verwenden wir `node-express/.ebextensions`.
2. Fügen Sie eine Konfigurationsdatei hinzu, mit der der Knoten-Befehl "npm start" festgelegt wird:

```
node-express/.ebextensions/nodecommand.config
```

```
option_settings:  
  aws:elasticbeanstalk:container:nodejs:  
    NodeCommand: "npm start"
```

Weitere Informationen finden Sie unter [Erweiterte Umgebungsanpassung mit Konfigurationsdateien \(.ebextensions\) \(p. 722\)](#).

3. Legen Sie Stufen für die Dateien fest:

```
~/node-express$ git add .  
~/node-express$ git commit -m "First express app"
```

4. Stellen Sie die Änderungen bereit:

```
~/node-express$ eb deploy
```

5. Sobald die Umgebung grün und einsatzbereit ist, aktualisieren Sie die URL, um die ordnungsgemäße Funktionsweise zu prüfen. Sie sollten eine Webseite mit dem Text Welcome to Express sehen.

Aktualisieren wir nun die Express-Anwendung für die Bereitstellung statischer Dateien und das Hinzufügen einer neuen Seite.

So konfigurieren Sie statische Dateien und fügen eine neue Seite zu Ihrer Express-Anwendung hinzu

1. Fügen Sie eine zweite Konfigurationsdatei mit dem folgenden Inhalt hinzu:

```
node-express/.ebextensions/staticfiles.config
```

```
option_settings:  
  aws:elasticbeanstalk:container:nodejs:staticfiles:
```

```
/public: /public
```

Diese Einstellung konfiguriert den Proxy-Server so, dass er Dateien im public-Ordner auf dem Pfad /public der Anwendung bereitstellt. [Das statische Bereitstellen von Dateien \(p. 253\)](#) vom Proxy-Server reduziert die Verarbeitungslast Ihrer Anwendung.

2. Kommentieren Sie die statische Zuordnung in node-express/app.js. Dieser Schritt ist nicht erforderlich, aber es ist ein guter Test, um zu bestätigen, dass statische Zuordnungen korrekt konfiguriert sind.

```
// app.use(express.static(path.join(__dirname, 'public')));
```

3. Fügen Sie Ihre aktualisierten Dateien zum lokalen Repository hinzu und übergeben Sie die Änderungen.

```
~/node-express$ git add .ebextensions/ app.js
~/node-express$ git commit -m "Serve stylesheets statically with nginx."
```

4. Add node-express/routes/hike.js. Geben Sie Folgendes ein:

```
exports.index = function(req, res) {
  res.render('hike', {title: 'My Hiking Log'});
};

exports.add_hike = function(req, res) {
};
```

5. Aktualisieren Sie node-express/app.js, um drei neue Zeilen hinzuzufügen.

Fügen Sie zuerst die folgende Zeile hinzu, um einen require für diese Route hinzuzufügen:

```
hike = require('./routes/hike');
```

Die Datei sollte wie im folgenden Ausschnitt aussehen:

```
var express = require('express');
var path = require('path');
var hike = require('./routes/hike');
```

Fügen Sie anschließend die folgenden beiden Zeilen zu node-express/app.js nach var app = express(); hinzu.

```
app.get('/hikes', hike.index);
app.post('/add_hike', hike.add_hike);
```

Die Datei sollte wie im folgenden Ausschnitt aussehen:

```
var app = express();
app.get('/hikes', hike.index);
app.post('/add_hike', hike.add_hike);
```

6. Kopieren Sie node-express/views/index.jade in node-express/views/hike.jade.

```
~/node-express$ cp views/index.jade views/hike.jade
```

7. Fügen Sie die Dateien zum lokalen Repository hinzu, übergeben Sie die Änderungen und stellen Sie die aktualisierte Anwendung bereit.

```
~/node-express$ git add .
~/node-express$ git commit -m "Add hikes route and template."
~/node-express$ eb deploy
```

8. Die Umgebung wird nach einigen Minuten aktualisiert. Nachdem Ihre Umgebung grün und bereit ist, überprüfen Sie das Funktionieren durch Aktualisieren Ihres Browsers und Anfügen von **hikes** am Ende der URL (z. B. <http://node-express-env-syyptcz2q.elasticbeanstalk.com/hikes>).

Sie sollten eine Webseite mit dem Titel My Hiking Log sehen.

Bereinigen

Falls Sie nicht länger mit Elastic Beanstalk arbeiten möchten, können Sie die Umgebung beenden.

Mit dem eb terminate-Befehl können Sie Umgebung und alle darin enthaltenen Ressourcen beenden.

```
~/node-express$ eb terminate
The environment "node-express-env" and all associated instances will be terminated.
To confirm, type the environment name: node-express-env
INFO: terminateEnvironment is starting.
...
```

Bereitstellen einer Express-Anwendung mit Clustering in Elastic Beanstalk

In diesem Tutorial wird beschrieben, wie Sie eine Beispielanwendung in Elastic Beanstalk mithilfe der Elastic Beanstalk Command Line Interface (EB CLI) bereitstellen und anschließend die Anwendung zur Verwendung des [Express](#)-Frameworks und [Amazon ElastiCache](#) mit Clustering aktualisieren. Mit Clustering lassen sich Hochverfügbarkeit, Leistung und Sicherheit Ihrer Webanwendung verbessern. Weitere Informationen über Amazon ElastiCache finden Sie unter [Was ist Amazon ElastiCache for Memcached?](#) im Benutzerhandbuch zu Amazon ElastiCache for Memcached.

Note

Mit diesem Beispiel werden AWS-Ressourcen erstellt, für die möglicherweise Kosten anfallen. Für weitere Informationen zu AWS-Preisen siehe <http://aws.amazon.com/pricing/>. Einige Services sind im Umfang des kostenlosen Nutzungskontingents für AWS enthalten. Als Neukunde können Sie diese Services kostenlos testen. Weitere Informationen finden Sie unter <http://aws.amazon.com/free/>.

Voraussetzungen

Dieses Tutorial erfordert die Node.js-Sprache, ihren Package Manager mit der Bezeichnung npm und das Express-Framework für Webanwendungen. Weitere Informationen zur Installation dieser Komponenten und zur Einrichtung Ihrer lokalen Entwicklungsumgebung finden Sie unter [Einrichten Ihrer Node.js-Entwicklungsumgebung \(p. 249\)](#).

Note

Für dieses Tutorial müssen Sie das AWS SDK for Node.js nicht installieren. Dies wird auch in [Einrichten Ihrer Node.js-Entwicklungsumgebung \(p. 249\)](#) erwähnt.

Das Tutorial erfordert auch die Elastic Beanstalk Command Line Interface (EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter [Installieren der EB CLI \(p. 1009\)](#) und [Konfigurieren der EB CLI \(p. 1017\)](#).

Erstellen einer Elastic Beanstalk-Umgebung

Konfigurieren Sie ein EB CLI-Repository für Ihre Anwendung und erstellen Sie eine Elastic Beanstalk-Umgebung mit der Node.js-Plattform.

1. Erstellen Sie ein Repository mit dem Befehl eb init.

```
~/node-express$ eb init --platform node.js --region us-east-2
Application node-express has been created.
```

Mit diesem Befehl wird eine Konfigurationsdatei in einem Ordner mit dem Namen `.elasticbeanstalk` generiert, die Einstellungen für die Erstellung von Umgebungen für Ihre Anwendung angibt und eine Elastic Beanstalk-Anwendung mit dem Namen des aktuellen Ordners erstellt.

2. Erstellen Sie eine Umgebung mit einer laufenden Beispielanwendung mit dem eb create-Befehl.

```
~/node-express$ eb create --sample node-express-env
```

Mit diesem Befehl wird eine Umgebung mit Lastausgleich, den Standardeinstellungen für die Node.js-Plattform und den folgenden Ressourcen erstellt:

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die so konfiguriert ist, dass Web-Apps auf der von Ihnen ausgewählten Plattform ausgeführt werden.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance security group (Instance-Sicherheitsgruppe): Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer) – Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe): Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – CloudWatch-Alarne, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.

- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form `subdomain.region.elasticbeanstalk.com` weiterleitet.
3. Wenn die Umgebungserstellung abgeschlossen ist, verwenden Sie den eb open-Befehl, um die URL der Umgebung im Standard-Browser zu öffnen.

```
~/node-express$ eb open
```

Aktualisieren der Anwendung

Aktualisieren Sie die Beispielanwendung in der Elastic Beanstalk-Umgebung zur Verwendung des Express-Frameworks.

Sie können den endgültigen Quellcode von [nodejs-example-express-elasticache.zip](#) herunterladen.

Note

Die vorausgesetzte Einrichtung der Entwicklungsumgebung führt zu einer Express-Projektstruktur im node-express-Ordner. Wenn Sie nicht bereits ein Express-Projekt generiert haben, führen Sie den folgenden Befehl aus. Weitere Informationen finden Sie unter [Installieren von Express \(p. 249\)](#).

```
~/node-express$ express && npm install
```

So aktualisieren Sie die Anwendung zur Nutzung von Express

1. Benennen Sie node-express/app.js um zu node-express/express-app.js.

```
node-express$ mv app.js express-app.js
```

2. Aktualisieren Sie die Zeile `var app = express();` in node-express/express-app.js folgendermaßen:

```
var app = module.exports = express();
```

3. Erstellen Sie auf dem lokalen Computer eine Datei namens node-express/app.js mit dem folgenden Code.

```
var cluster = require('cluster'),
    app = require('./express-app');

var workers = {},
    count = require('os').cpus().length;

function spawn(){
    var worker = cluster.fork();
    workers[worker.pid] = worker;
    return worker;
}

if (cluster.isMaster) {
    for (var i = 0; i < count; i++) {
        spawn();
    }
    cluster.on('death', function(worker) {
        console.log('worker ' + worker.pid + ' died. spawning a new process...');
        delete workers[worker.pid];
        spawn();
    });
}
```

```
    } else {
      app.listen(process.env.PORT || 5000);
    }
}
```

4. Stellen Sie die aktualisierte Anwendung bereit.

```
node-express$ eb deploy
```

5. Die Umgebung wird nach einigen Minuten aktualisiert. Sobald die Umgebung grün und einsatzbereit ist, aktualisieren Sie die URL, um die ordnungsgemäße Funktionsweise zu prüfen. Sie sollten eine Webseite mit dem Text „Welcome to Express“ sehen.

Sie können auf die Protokolle für Ihre EC2 Instances, die Ihre Anwendung ausführen, zugreifen. Anleitungen zum Zugreifen auf Ihre Protokolle finden Sie unter [Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen \(p. 869\)](#).

Im nächsten Schritt wird die Express-Anwendung für die Verwendung von Amazon ElastiCache aktualisiert.

Aktualisieren der Express-Anwendung für die Verwendung von Amazon ElastiCache

1. Erstellen Sie auf dem lokalen Computer ein `.ebextensions`-Verzeichnis im Verzeichnis der obersten Ebene Ihres Quell-Bundles. In diesem Beispiel verwenden wir `node-express/.ebextensions`.
2. Erstellen Sie eine Konfigurationsdatei `node-express/.ebextensions/elasticache-iam-with-script.config` mit dem folgenden Snippet. Weitere Informationen zur Konfigurationsdatei finden Sie unter [Node.js-Konfigurations-Namespace \(p. 253\)](#). Damit wird ein IAM-Benutzer mit den benötigten Berechtigungen zum Erkennen der ElastiCache-Knoten erstellt. Zudem wird jede Veränderung im Cache in einer Datei erfasst. Sie können die Datei auch aus [nodejs-example-express-elasticache.zip](#) kopieren. Weitere Informationen zu den ElastiCache-Eigenschaften finden Sie unter [Beispiel: ElastiCache \(p. 752\)](#).

Note

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

```
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 :
    ElasticCacheConfig :
      CacheName :
        Ref : MyElasticCache
      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" ] }`

MyElasticCacheName:
  Description: Name of the elasticache
  Value:
    Ref : MyElasticCache

NumCacheNodes:
  Description: Number of cache nodes in MyElasticCache
  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

```

- Erstellen Sie auf dem lokalen Computer die Konfigurationsdatei `node-express/.ebextensions/elasticache_settings.config` mit dem folgenden Snippet, um ElastiCache zu konfigurieren.

```
option_settings:
  "aws:elasticbeanstalk:customoption":
    CacheNodeType: cache.t2.micro
    NumCacheNodes: 1
    Engine: memcached
    NodeListPath: /var/nodelist
```

4. Ersetzen Sie auf Ihrem lokalen Computer `node-express/express-app.js` durch den folgenden Snippet. Diese Datei liest die Knoten-Liste vom Datenträger (`/var/nodelist`) und konfiguriert Express so, dass memcached als Sitzungsspeicher verwendet wird, wenn Knoten vorhanden sind. Ihre Datei sollte wie folgt aussehen.

/* *

```
* 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);
});
```

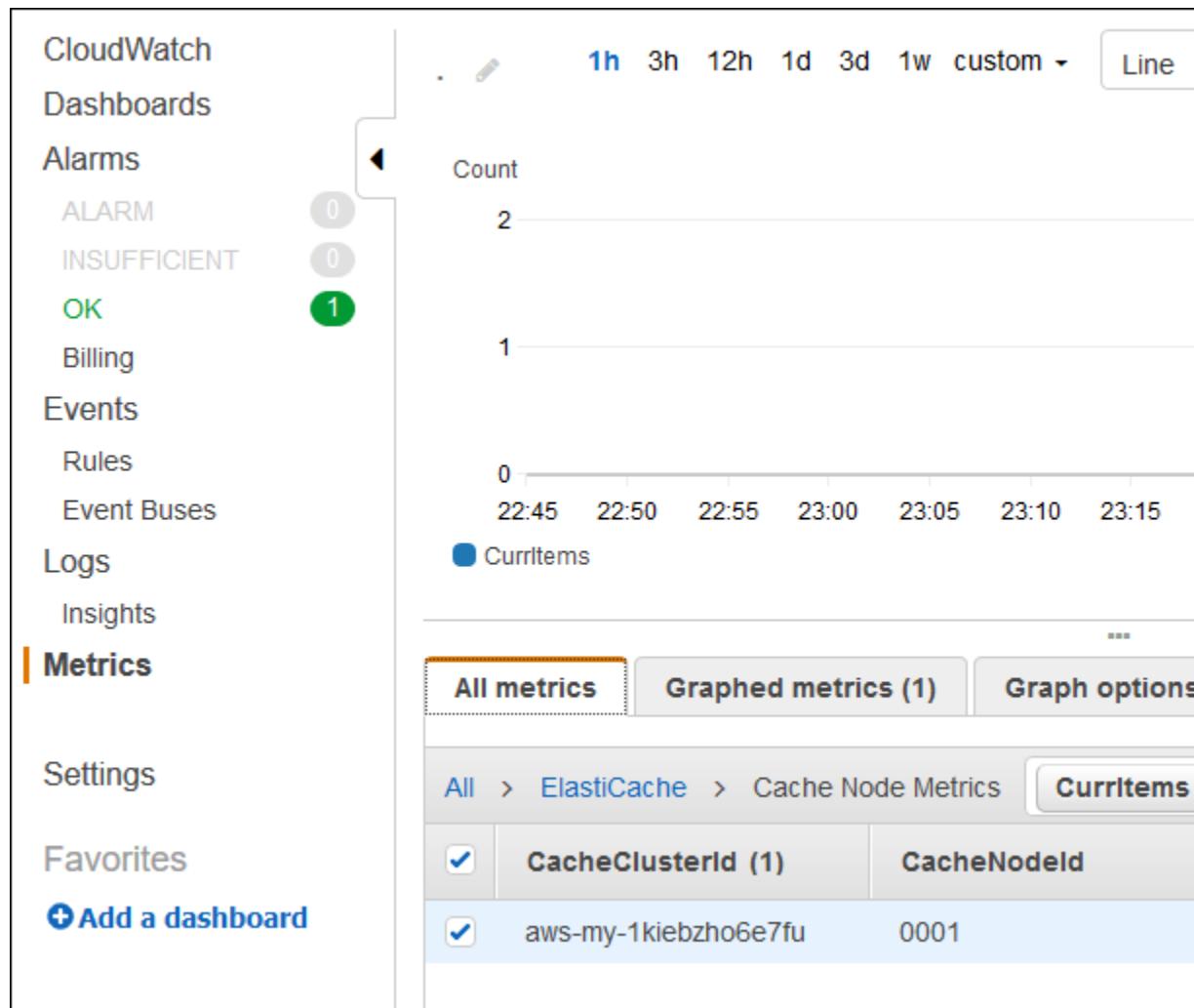
5. Aktualisieren Sie auf dem lokalen Computer `node-express/package.json`, um vier Abhängigkeiten hinzuzufügen.

```
{
  "name": "node-express",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "start": "node ./bin/www"
  },
  "dependencies": {
    "cookie-parser": "*",
    "debug": "~2.6.9",
    "express": "~4.16.0",
    "http-errors": "~1.6.2",
    "jade": "~1.11.0",
    "morgan": "~1.9.0",
    "connect-memcached": "*",
    "express-session": "*",
    "body-parser": "*",
    "method-override": "*"
  }
}
```

6. Stellen Sie die aktualisierte Anwendung bereit.

```
node-express$ eb deploy
```

7. Die Umgebung wird nach einigen Minuten aktualisiert. Wenn die Umgebung grün und einsatzbereit ist, überprüfen Sie, ob der Code funktioniert hat.
 - a. Zeigen Sie die ElastiCache-Metriken in der [Amazon CloudWatch-Konsole](#) an. Zum Anzeigen Ihrer ElastiCache-Metriken wählen Sie im linken Bereich Metrics (Metriken) aus und suchen Sie dann nach CurrItems. Wählen Sie ElastiCache > Cache Node Metrics (ElastiCache > Cache-Knoten-Metriken) und dann Ihren Cache-Knoten aus, um die Anzahl der Elemente im Cache anzuzeigen.



Note

Achten Sie darauf, dass Sie in der Region suchen, in der Sie die Anwendung bereitgestellt haben.

Wenn Sie die Anwendungs-URL kopieren, in einen anderen Webbrowser einfügen und die Seite aktualisieren, sollte die CurrItem-Anzahl nach fünf Minuten steigen.

- b. Erstellen Sie einen Snapshot Ihrer Protokolle. Weitere Informationen über das Abrufen von Protokollen finden Sie unter [Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen \(p. 869\)](#).
- c. Überprüfen Sie die Datei `/var/log/nodejs/nodejs.log` im Protokoll-Bundle. Die Ausgabe sollte wie folgt aussehen:

```
Using memcached store nodes:  
[ 'aws-my-1oys9co8ztluo.liwtrn.0001.use1.cache.amazonaws.com:11211' ]
```

Bereinigen

Wenn Sie Ihre Anwendung nicht mehr ausführen möchten, können Sie Ihr System durch Beenden Ihrer Umgebung und Löschen Ihrer Anwendung bereinigen.

Verwenden Sie den Befehl `eb terminate`, um Ihre Umgebung zu beenden, und den Befehl `eb delete` zum Löschen Ihrer Anwendung.

So beenden Sie Ihre Umgebung

Führen Sie in dem Verzeichnis, in dem Sie Ihr lokales Repository erstellt haben, `eb terminate` aus.

```
$ eb terminate
```

Dieser Vorgang kann einige Minuten dauern. Elastic Beanstalk zeigt eine Meldung an, sobald die Umgebung erfolgreich beendet wurde.

Bereitstellen einer Node.js-Anwendung mit DynamoDB in Elastic Beanstalk

Dieses Tutorial und die [Beispielanwendung](#) führen Sie durch den Prozess der Bereitstellung einer Node.js-Anwendung, die das AWS SDK for JavaScript in Node.js verwendet, um mit Amazon DynamoDB zu interagieren. Sie erstellen eine DynamoDB-Tabelle, die außerhalb der AWS Elastic Beanstalk-Umgebung liegt, und konfigurieren die Anwendung so, dass diese externe Tabelle verwendet wird, anstatt eine in der Umgebung zu erstellen. In einer Produktionsumgebung behalten Sie die Tabelle unabhängig von der Elastic Beanstalk-Umgebung zum Schutz vor versehentlichem Datenverlust und damit Sie [Blau/Grün-Bereitstellungen \(p. 481\)](#) durchführen können.

Die Tutorial-Beispielanwendung verwendet eine DynamoDB-Tabelle zum Speichern der vom Benutzer bereitgestellten Textdaten. Die Beispielanwendung verwendet [Konfigurationsdateien \(p. 722\)](#), um die Tabelle und ein Amazon Simple Notification Service-Thema zu erstellen. Außerdem erfahren Sie, wie Sie eine [package.json-Datei \(p. 256\)](#) zum Installieren von Paketen während der Bereitstellung verwenden.

Abschnitte

- [Voraussetzungen \(p. 273\)](#)
- [Starten einer Elastic Beanstalk-Umgebung \(p. 274\)](#)
- [Hinzufügen von Berechtigungen zu den Instances Ihrer Umgebung \(p. 276\)](#)
- [Bereitstellen der Beispielanwendung \(p. 276\)](#)
- [Erstellen einer DynamoDB-Tabelle \(p. 278\)](#)
- [Aktualisieren der Konfigurationsdateien der Anwendung \(p. 279\)](#)
- [Konfigurieren Ihrer Umgebung für hohe Verfügbarkeit \(p. 281\)](#)
- [Bereinigen \(p. 281\)](#)
- [Nächste Schritte \(p. 282\)](#)

Voraussetzungen

- Bevor Sie beginnen, laden Sie das Beispielanwendungs-Quell-Bundle von GitHub herunter: [eb-node-express-sample-v1.1.zip](#)
- Außerdem benötigen Sie ein Befehlszeilen-Terminal oder eine -Shell für die Ausführung der Befehle in den Verfahren. Beispielbefehlen sind ein Eingabeaufforderungs-Symbol (\$) und der Namen des aktuellen Verzeichnisses vorangestellt (bei Bedarf):

```
~/eb-project$ this is a command
```

```
this is output
```

Note

Sie können alle Befehle in diesem Tutorial auf einer virtuellen Linux-Maschine, einem OS X-Computer oder einer Amazon EC2-Instance mit Amazon Linux ausführen. Wenn Sie eine Entwicklungsumgebung benötigen, können Sie eine Elastic Beanstalk-Umgebung mit einer einzelnen Instance starten und sie mit SSH verbinden.

- In diesem Tutorial wird ein ZIP-Befehlszeilenprogramm zum Erstellen eines Quell-Bundles verwendet, das Sie in Elastic Beanstalk bereitstellen können. Um den Befehl `zip` in Windows zu verwenden, installieren Sie [UnxUtils](#), eine Lightweight-Kollektion nützlicher Befehlszeilenprogramme wie z. B. `zip` und `ls`. (Alternativ können Sie den [Windows-Explorer \(p. 414\)](#) oder ein beliebiges anderes ZIP-Programm für die Erstellung von Quell-Bundle-Archiven nutzen.)

So installieren Sie UnxUtils

1. Herunterladen [UnxUtils](#).
2. Extrahieren Sie das Archiv in ein lokales Verzeichnis. Beispiel, `C:\Program Files (x86)`
3. Fügen Sie den Pfad zu den binären Dateien zu Ihrer Windows PATH-Benutzervariable hinzu.
Beispiel, `C:\Program Files (x86)\UnxUtils\usr\local\wbin`.
 - a. Betätigen Sie die Windows-Taste und geben Sie anschließend **environment variables** ein.
 - b. Wählen Sie Edit environment variables for your account (Umgebungsvariablen für Ihr Konto bearbeiten).
 - c. Wählen Sie PATH (PFAD) und Edit (Bearbeiten) aus.
 - d. Fügen Sie im Feld Variable value (Variablenwert) Pfade hinzu, getrennt durch Semikola.
Beispiel: `c:\item1\path;c:\item2\path`
 - e. Klicken Sie zweimal auf OK, um die neuen Einstellungen anzuwenden.
 - f. Schließen Sie alle laufenden Eingabeaufforderungsfenster und öffnen Sie dann erneut ein Eingabeaufforderungsfenster.
4. Öffnen Sie ein neues Eingabeaufforderungsfenster und führen Sie den `zip`-Befehl aus, um sicherzustellen, dass es funktioniert.

```
> zip -h
Copyright (C) 1990-1999 Info-ZIP
Type 'zip "-L"' for software license.
...
```

Starten einer Elastic Beanstalk-Umgebung

Sie verwenden die Elastic Beanstalk-Konsole, um eine Elastic Beanstalk-Umgebung zu starten. Wählen Sie die Plattform Node.js aus und übernehmen Sie die Standardeinstellungen und den Beispielcode. Nachdem Sie die Berechtigungen der Umgebung konfiguriert, stellen Sie die Beispielanwendung bereit, die Sie von GitHub heruntergeladen haben.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.

4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus, und wenn Sie bereit sind, wählen Sie Create App (App erstellen) aus.

Elastic Beanstalk benötigt etwa fünf Minuten, um die Umgebung mit den folgenden Ressourcen zu erstellen:

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die so konfiguriert ist, dass Web-Apps auf der von Ihnen ausgewählten Plattform ausgeführt werden.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance security group (Instance-Sicherheitsgruppe): Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer) – Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe): Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – CloudWatch-Alarne, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form **subdomain.region.elasticbeanstalk.com** weiterleitet.

Elastic Beanstalk verwaltet all diese Ressourcen. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Note

Der von Elastic Beanstalk erstellte &S3;-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#).

Hinzufügen von Berechtigungen zu den Instances Ihrer Umgebung

Ihre Anwendung führt eine oder mehrere EC2-Instances hinter einem Load Balancer aus, wobei HTTP-Anfragen aus dem Internet verarbeitet werden. Wenn eine Anforderung empfangen wird, die die Verwendung von AWS-Services erfordert, verwendet die Anwendung die Berechtigungen der Instance, auf der sie ausgeführt wird, für den Zugriff auf diese Services.

Die Beispielanwendung verwendet Instance-Berechtigungen, um Daten in eine DynamoDB-Tabelle zu schreiben und Benachrichtigungen an das Amazon SNS-Thema mit dem SDK for JavaScript in Node.js zu senden. Fügen Sie die folgenden verwalteten Richtlinien zum standardmäßigen [Instance-Profil \(p. 22\)](#) hinzu, um den EC2-Instances in Ihrer Umgebungsberechtigung den Zugriff auf DynamoDB und Amazon SNS zu gewähren:

- `AmazonDynamoDBFullAccess`
- `AmazonSNSFullAccess`

So fügen Sie Richtlinien zum Instance-Standardprofil hinzu

1. Öffnen Sie die Seite [Roles \(Rollen\)](#) in der IAM-Konsole.
2. Wählen Sie `aws-elasticbeanstalk-ec2-role` aus.
3. Wählen Sie auf der Registerkarte `Permissions (Berechtigungen)` die Option `Attach policies (Richtlinien anfügen)` aus.
4. Wählen Sie die verwaltete Richtlinie für die zusätzlichen Services aus, die von der Anwendung verwendet werden. Zum Beispiel `AmazonSNSFullAccess` oder `AmazonDynamoDBFullAccess`.
5. Wählen Sie `Attach policy` aus.

Weitere Informationen zum Verwalten von Instance-Profilen finden Sie unter [Elastic Beanstalk Instance-Profile verwalten \(p. 904\)](#).

Bereitstellen der Beispielanwendung

Jetzt können Sie in Ihrer Umgebung die Beispielanwendung bereitstellen und ausführen.

Note

Laden Sie das Quell-Bundle von GitHub herunter, falls Sie es noch nicht getan haben: [eb-node-express-sample-v1.1.zip](#).

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option `Upload and deploy (Hochladen und bereitstellen)`.
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie `Deploy (Bereitstellen)` aus.

6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Die Website sammelt Benutzerkontaktinformationen und verwendet eine DynamoDB-Tabelle zum Speichern der Daten. Zum Hinzufügen eines Eintrags wählen Sie Sign up today (Jetzt registrieren), geben Sie einen Namen und eine E-Mail-Adresse ein, und wählen Sie dann Sign Up! (Registrieren) aus. Die Webanwendung schreibt die Formularinhalte in die Tabelle und löst eine Amazon SNS-E-Mail-Benachrichtigung aus.

The screenshot shows a website with the title "A New Startup" in the top left corner. In the top right corner, there are three buttons: "Home" (dark blue), "About" (light blue), and "Blog" (light blue). The main content area features a large, bold, dark blue text block that reads "The next big thing is coming...". Below this, in a smaller dark blue font, is the text "We're pretty thrilled to unveil our latest creation. Sign up below to be notified when we officially launch!". At the bottom of this section is a teal-colored button with white text that says "Sign up today". At the very bottom of the page, in a small gray font, is the copyright notice "© A New Startup 2013".

Aktuell ist das Amazon SNS-Thema mit einer Platzhalter-E-Mail für Benachrichtigungen konfiguriert. Sie werden die Konfiguration in Kürze aktualisieren, aber in der Zwischenzeit können Sie die DynamoDB-Tabelle und das Amazon SNS;-Thema in der AWS-Managementkonsole überprüfen.

So zeigen Sie die Tabelle an

1. Öffnen Sie die Seite [Tables \(Tabellen\)](#) in der DynamoDB-Konsole.
2. Suchen Sie die Tabelle, die die Anwendung erstellt hat. Der Name beginnt mit awseb und enthält StartupSignupsTable.
3. Wählen Sie die Tabelle aus, klicken Sie auf Items und anschließend auf Start search, um alle Elemente in der Tabelle anzuzeigen.

Die Tabelle enthält einen Eintrag für jede E-Mail-Adresse, die auf der Anmeldungs-Website übermittelt wurde. Zusätzlich zum Schreiben in die Tabelle sendet die Anwendung eine Nachricht an ein Amazon SNS-Thema, das zwei Abonnements hat: eines für E-Mail-Benachrichtigungen an Sie und ein anderes für eine Amazon Simple Queue Service-Warteschlange, aus der eine Worker-Anwendung lesen kann, um Anfragen zu verarbeiten und E-Mails an interessierte Kunden zu senden.

So zeigen Sie das Thema an

1. Öffnen Sie die Seite [Topics \(Themen\)](#) in der Amazon SNS-Konsole.
2. Suchen Sie das Thema, das die Anwendung erstellt hat. Der Name beginnt mit awseb und enthält NewSignupTopic.
3. Wählen Sie das Thema, um seine Abonnements anzuzeigen.

Die Anwendung ([app.js](#)) definiert zwei Routen. Der Stammpfad (/) gibt eine Webseite zurück, die aus einer Embedded JavaScript(EJS)-Vorlage mit einem Formular gerendert wurde, das der Benutzer ausfüllt, um seinen Namen und seine E-Mail-Adresse zu registrieren. Durch das Übermitteln des Formulars wird eine POST-Anforderung mit den Formulardaten an das /signup-Routing gesendet, das einen Eintrag in die DynamoDB-Tabelle schreibt und eine Nachricht im Amazon SNS-Thema veröffentlicht, um den Eigentümer des Themas zu benachrichtigen.

Die Beispieldatenbank enthält [Konfigurationsdateien \(p. 722\)](#), mit der die DynamoDB-Tabelle, das Amazon SNS-Thema und die Amazon SQS-Warteschlange, die von der Anwendung verwendet werden, erstellt werden. Dadurch können Sie eine neue Umgebung erstellen und die Funktionalität sofort testen. Der Nachteil ist, dass die DynamoDB-Tabelle an die Umgebung gebunden ist. Bei einer Produktionsumgebung sollten Sie die DynamoDB-Tabelle außerhalb Ihrer Umgebung erstellen, um zu vermeiden, dass Sie sie verlieren, wenn Sie die Umgebung beenden oder ihre Konfiguration aktualisieren.

Erstellen einer DynamoDB-Tabelle

Um eine externe DynamoDB-Tabelle mit einer Anwendung zu verwenden, die in Elastic Beanstalk ausgeführt wird, erstellen Sie zuerst eine Tabelle in DynamoDB. Wenn Sie eine Tabelle außerhalb von Elastic Beanstalk erstellen, ist diese vollständig von Elastic Beanstalk und Ihren Elastic Beanstalk-Umgebungen unabhängig und wird nicht von Elastic Beanstalk beendet.

Verwenden Sie die folgenden Einstellungen, um eine Tabelle zu erstellen:

- Tabellenname – **nodejs-tutorial**
- Primärschlüssel – **email**
- Primary key type (Primärschlüsseltyp) – Zeichenfolge

So erstellen Sie eine DynamoDB-Tabelle

1. Öffnen Sie die Seite [Tables \(Tabellen\)](#) in der DynamoDB-Managementkonsole.
2. Wählen Sie Create table aus.
3. Geben Sie einen Table name (Tabellennamen) und Primary key (Primärschlüssel) ein.

4. Wählen Sie den Typ des Primärschlüssels.
5. Wählen Sie Create (Erstellen) aus.

Aktualisieren der Konfigurationsdateien der Anwendung

Aktualisieren Sie die [Konfigurationsdateien](#) (p. 722) in der Anwendungsquelle zur Verwendung der nodejs-tutorial-Tabelle, statt eine neue zu erstellen.

So aktualisieren Sie die Beispielanwendung für die Produktionsnutzung

1. Extrahieren Sie die Projektdateien aus dem Quell-Bundle:

```
~$ mkdir nodejs-tutorial
~$ cd nodejs-tutorial
~/nodejs-tutorial$ unzip ~/Downloads/eb-node-express-sample-v1.0.zip
```

2. Öffnen Sie `.ebextensions/options.config` und ändern Sie die Werte der folgenden Einstellungen:

- NewSignupEmail – Ihre E-Mail-Adresse.
- 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"
```

Dies konfiguriert die Anwendung für die Verwendung der nodejs-tutorial-Tabelle anstelle der, die in `.ebextensions/create-dynamodb-table.config` erstellt wurde, und legt die E-Mail-Adresse fest, die das Amazon SNS-Thema für Benachrichtigungen verwendet.

3. Remove `.ebextensions/create-dynamodb-table.config`.

```
~/nodejs-tutorial$ rm .ebextensions/create-dynamodb-table.config
```

Wenn Sie das nächste Mal die Anwendung bereitstellen, wird die Tabelle, die von dieser Konfigurationsdatei erstellt wurde, gelöscht.

4. Erstellen Sie ein Quell-Bundle aus dem geänderten Code.

```
~/nodejs-tutorial$ zip nodejs-tutorial.zip -r *.[^.*]
adding: LICENSE (deflated 65%)
adding: README.md (deflated 56%)
adding: app.js (deflated 63%)
adding: iam_policy.json (deflated 47%)
adding: misc/ (stored 0%)
adding: misc/theme-flow.png (deflated 1%)
adding: npm-shrinkwrap.json (deflated 87%)
adding: package.json (deflated 40%)
adding: static/ (stored 0%)
adding: static/bootstrap/ (stored 0%)
adding: static/bootstrap/css/ (stored 0%)
adding: static/bootstrap/css/jumbotron-narrow.css (deflated 59%)
adding: static/bootstrap/css/theme/ (stored 0%)
adding: static/bootstrap/css/theme/united/ (stored 0%)
adding: static/bootstrap/css/theme/united/bootstrap.css (deflated 86%)
adding: static/bootstrap/css/theme/amelia/ (stored 0%)
adding: static/bootstrap/css/theme/amelia/bootstrap.css (deflated 86%)
adding: static/bootstrap/css/theme/slate/ (stored 0%)
adding: static/bootstrap/css/theme/slate/bootstrap.css (deflated 87%)
adding: static/bootstrap/css/theme/default/ (stored 0%)
adding: static/bootstrap/css/theme/default/bootstrap.css (deflated 86%)
adding: static/bootstrap/css/theme/flatly/ (stored 0%)
adding: static/bootstrap/css/theme/flatly/bootstrap.css (deflated 86%)
adding: static/bootstrap/LICENSE (deflated 65%)
adding: static/bootstrap/js/ (stored 0%)
adding: static/bootstrap/js/bootstrap.min.js (deflated 74%)
adding: static/bootstrap/fonts/ (stored 0%)
adding: static/bootstrap/fonts/glyphicons-halflings-regular.eot (deflated 1%)
adding: static/bootstrap/fonts/glyphicons-halflings-regular.svg (deflated 73%)
adding: static/bootstrap/fonts/glyphicons-halflings-regular.woff (deflated 1%)
adding: static/bootstrap/fonts/glyphicons-halflings-regular.ttf (deflated 44%)
adding: static/jquery/ (stored 0%)
adding: static/jquery/jquery-1.11.3.min.js (deflated 65%)
adding: static/jquery/MIT-LICENSE.txt (deflated 41%)
adding: views/ (stored 0%)
adding: views/index.ejs (deflated 67%)
adding: .ebextensions/ (stored 0%)
adding: .ebextensions/options.config (deflated 47%)
adding: .ebextensions/create-sns-topic.config (deflated 56%)
```

Stellen Sie das nodejs-tutorial.zip-Quell-Bundle für Ihre Umgebung bereit.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.

6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Bei der Bereitstellung aktualisiert Elastic Beanstalk die Konfiguration des Amazon SNS-Themas und löscht die DynamoDB-Tabelle, die erstellt wurde, als Sie die erste Version der Anwendung bereitgestellt haben.

Wenn Sie die Umgebung jetzt beenden, wird die nodejs-tutorial-Tabelle nicht gelöscht. Dadurch können Sie blaue/grüne Bereitstellungen durchführen, Konfigurationsdateien ändern oder Ihre Website offline nehmen, ohne Datenverlust zu riskieren.

Öffnen Sie Ihre Webseite in einem Browser und überprüfen Sie, ob das Formular wie erwartet funktioniert. Erstellen Sie einige Einträge und prüfen Sie die DynamoDB-Konsole, um die Tabelle zu verifizieren.

So zeigen Sie die Tabelle an

1. Öffnen Sie die Seite [Tables \(Tabellen\)](#) in der DynamoDB-Konsole.
2. Suchen Sie die nodejs-tutorial-Tabelle.
3. Wählen Sie die Tabelle aus, klicken Sie auf Items und anschließend auf Start search, um alle Elemente in der Tabelle anzuzeigen.

Sie können auch sehen, dass Elastic Beanstalk die zuvor erstellte Tabelle gelöscht hat.

Konfigurieren Ihrer Umgebung für hohe Verfügbarkeit

Konfigurieren Sie abschließend für die Auto Scaling-Gruppe der Umgebung eine höhere Instance-Mindestanzahl. Führen Sie jederzeit mindestens zwei Instances aus, damit kein Webserver in der Umgebung eine einzelne Fehlerquelle darstellt und damit Sie Änderungen ohne Ausfallzeit der Website bereitstellen können.

So konfigurieren Sie die Auto Scaling-Gruppe Ihrer Umgebung für hohe Verfügbarkeit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Legen Sie im Abschnitt Auto Scaling Group (Auto Scaling-Gruppe) die Option Min instances (Min. Instances) auf **2** fest.
6. Wählen Sie Apply (Anwenden) aus.

Bereinigen

Falls Sie nicht länger mit Elastic Beanstalk arbeiten möchten, können Sie die Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 555\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie auf Environment actions (Umgebungsaktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Sie können auch die durch Sie erstellten, externen DynamoDB-Tabellen löschen.

So löschen Sie eine DynamoDB-Tabelle

1. Öffnen Sie die Seite [Tables \(Tabellen\)](#) in der DynamoDB-Konsole.
2. Wählen Sie eine Tabelle aus.
3. Wählen Sie die Option Actions (Aktionen) und anschließend Delete table (Tabelle löschen) aus.
4. Wählen Sie Delete.

Nächste Schritte

Im weiteren Verlauf der Anwendungsentwicklung möchten Sie vermutlich Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die [Elastic Beanstalk Command Line Interface \(p. 1008\)](#) (EB CLI) stellt benutzerfreundliche Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk-Umgebungen über die Befehlszeile bereit.

Die Beispielanwendung verwendet Konfigurationsdateien zum Konfigurieren von Softwareeinstellungen und Erstellen von AWS-Ressourcen als Teil Ihrer Umgebung. Weitere Informationen über Konfigurationsdateien und Ihre Verwendung finden Sie unter [Erweiterte Umgebungsanpassung mit Konfigurationsdateien \(.ebextensions\) \(p. 722\)](#).

Die Beispielanwendung für dieses Tutorial verwendet das Express-Web-Framework für Node.js. Weitere Informationen zu Express finden Sie in der offiziellen Dokumentation unter [expressjs.com](#).

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, [konfigurieren Sie einen benutzerdefinierten Domänennamen \(p. 644\)](#) für die Umgebung und [aktivieren Sie HTTPS \(p. 778\)](#) für sichere Verbindungen.

Hinzufügen einer Amazon RDS-DB-Instance zur Node.js-Anwendungsumgebung

Sie können eine DB-Instance von Amazon Relational Database Service (Amazon RDS) verwenden, um Daten zu speichern, die von Ihrer Anwendung gesammelt und geändert wurden. Die Datenbank kann Ihrer Umgebung angefügt und von Elastic Beanstalk verwaltet werden oder sie kann extern erstellt und verwaltet werden.

Bei der ersten Verwendung von Amazon RDS [fügen Sie eine DB-Instance \(p. 283\)](#) zu einer Testumgebung über die Elastic Beanstalk-Managementkonsole hinzu und prüfen, ob die Anwendung eine Verbindung zu dieser herstellen kann.

Für die Verbindungserstellung zu einer Datenbank [fügen Sie einen Treiber zur Anwendung hinzu \(p. 284\)](#), laden den Treiber in den Code und [erstellen ein Verbindungsobjekt \(p. 285\)](#) mit den von Elastic Beanstalk bereitgestellten Umgebungseigenschaften. Die Konfiguration und der Verbindungscode hängen von der Datenbank-Engine und dem Framework ab, die Sie nutzen.

Note

Zu Lernzwecken oder für Testumgebungen können Sie mit Elastic Beanstalk eine DB-Instance hinzufügen.

Für Produktionsumgebungen können Sie eine DB-Instance außerhalb Ihrer Elastic Beanstalk-Umgebung erstellen, um Ihre Umgebungsressourcen von Ihren Datenbankressourcen zu entkoppeln. Auf diese Weise wird beim Beenden Ihrer Umgebung die DB-Instance nicht gelöscht. Mit einer externen DB-Instance können Sie von mehreren Umgebungen aus eine Verbindung mit derselben Datenbank herstellen und [Blau/Grün-Bereitstellungen](#) durchführen. Anleitungen hierzu finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon RDS \(p. 973\)](#).

Abschnitte

- [Hinzufügen einer DB-Instance zu Ihrer Umgebung \(p. 283\)](#)
- [Herunterladen eines Treibers \(p. 284\)](#)
- [Verbinden mit einer Datenbank \(p. 285\)](#)

Hinzufügen einer DB-Instance zu Ihrer Umgebung

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie Apply (Anwenden) aus.

Das Hinzufügen einer DB-Instance dauert ca. 10 Minuten. Sobald die Umgebungsaktualisierung abgeschlossen ist, stehen der Hostname der DB-Instance und andere Verbindungsinformationen über die folgenden Umgebungseigenschaften zur Verfügung:

Eigenschaftenname	Beschreibung	Eigenschaftenwert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).

Eigenschaftenname	Beschreibung	Eigenschaftswert
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbankname, ebdb	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: DB Name (DB-Name).
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	In der Amazon RDS-Konsole nicht als Referenz verfügbar.

Weitere Informationen zum Konfigurieren einer internen DB-Instance finden Sie unter [Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung \(p. 609\)](#).

Herunterladen eines Treibers

Fügen Sie den Datenbanktreiber zur [package.json-Datei \(p. 256\)](#) unter dependencies hinzu.

Example **package.json** – Ausdrücken mit 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"
  }
}
```

Gängige Treiberpakete für Node.js

- MySQL – `mysql`
- PostgreSQL – `pg`
- SQL Server – `mssql`
- Oracle – `oracle` oder `oracledb`

Oracle-Paket und -Version hängen von der Node.js-Version ab, die Sie verwenden:

- Node.js 6.x, 8.x – Verwenden Sie die neueste Version von `oracledb`.
- Node.js 4.x – Verwenden Sie die `oracledb`-Version 2.2.0.

- Node.js 5.x, 7.x – Verwenden Sie die neueste Version von oracle. Das oracledb-Paket unterstützt die folgenden Node.js-Versionen nicht.

Verbinden mit einer Datenbank

Elastic Beanstalk stellt Verbindungsinformationen für zugehörige DB-Instances in Umgebungseigenschaften bereit. Mit `process.env.VARIABLE` können Sie die Eigenschaften lesen und eine Datenbankverbindung konfigurieren.

Example app.js – MySQL-Datenbankverbindung

```
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();
```

Weitere Informationen über das Erstellen einer Verbindungszeichenfolge mithilfe von node-mysql finden Sie unter npmjs.org/package/mysql.

Ressourcen

Es gibt zahlreiche Möglichkeiten, um zusätzliche Unterstützung bei der Entwicklung von Node.js-Anwendungen zu erhalten:

Ressource	Beschreibung
GitHub	Installieren Sie das AWS SDK for Node.js mit GitHub.
Node.js-Entwicklungsforum	Hier können Sie Fragen stellen und Feedback erhalten.
AWS SDK for Node.js (Developer-Vorschau)	Umfangreiches Kit mit Beispiel-Code, Dokumentation, Tools und zusätzlichen Ressourcen.

Erstellen und Bereitstellen von PHP-Anwendungen in Elastic Beanstalk

AWS Elastic Beanstalk for PHP erleichtert die Bereitstellung, Verwaltung und Skalierung Ihrer PHP-Webanwendungen mit Amazon Web Services. Elastic Beanstalk for PHP steht jedem zur Verfügung,

der eine Webanwendung mit PHP entwickelt oder hostet. Dieser Abschnitt enthält Anweisungen für die Bereitstellung Ihrer PHP-Webanwendung für Elastic Beanstalk. Sie können Ihre Anwendung in wenigen Minuten mit der Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI) oder über die Elastic Beanstalk-Managementkonsole bereitstellen. Außerdem bietet sie schrittweise Anleitungen zu den gängigen Frameworks, wie CakePHP und Symfony.

In den Themen in diesem Kapitel wird davon ausgegangen, dass Sie über einige Kenntnisse der Elastic Beanstalk-Umgebungen verfügen. Wenn Sie Elastic Beanstalk noch nicht verwendet haben, machen Sie sich im [Tutorial „Erste Schritte“ \(p. 3\)](#) mit den Grundlagen vertraut.

Wenn Sie Hilfe bei der Entwicklung der PHP-Anwendung benötigen, gibt es verschiedene Ressourcen:

Ressource	Beschreibung
GitHub	Installieren Sie das AWS SDK for PHP mit GitHub.
PHP Development Forum	Hier können Sie Fragen stellen und Feedback erhalten.
PHP Developer Center	Umfangreiches Kit mit Beispiel-Code, Dokumentation, Tools und zusätzlichen Ressourcen.
AWS SDK for PHP FAQs	Hier erhalten Sie Antworten auf häufig gestellte Fragen.

Themen

- [Einrichten der PHP-Entwicklungsumgebung \(p. 286\)](#)
- [Verwenden der Elastic Beanstalk-PHP-Plattform \(p. 288\)](#)
- [Bereitstellen einer Laravel-Anwendung in Elastic Beanstalk \(p. 293\)](#)
- [Bereitstellen einer CakePHP-Anwendung in Elastic Beanstalk \(p. 301\)](#)
- [Bereitstellen einer Symfony-Anwendung in Elastic Beanstalk \(p. 308\)](#)
- [Bereitstellen einer PHP-Anwendung mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank zu Elastic Beanstalk \(p. 313\)](#)
- [Bereitstellen einer WordPress-Website mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank zu Elastic Beanstalk \(p. 322\)](#)
- [Bereitstellen einer Drupal-Website mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank zu Elastic Beanstalk \(p. 336\)](#)
- [Hinzufügen einer Amazon RDS-DB-Instance zur PHP-Anwendungsumgebung \(p. 350\)](#)

Einrichten der PHP-Entwicklungsumgebung

Richten Sie eine PHP-Entwicklungsumgebung ein, um Ihre Anwendung vor der Bereitstellung in AWS Elastic Beanstalk lokal zu testen. In diesem Thema finden Sie die Schritte zum Einrichten der Entwicklungsumgebung sowie Links zu den Installationsseiten nützlicher Tools.

Allgemeine Einrichtungsschritte und Tools für alle Sprachen finden Sie unter [Konfigurieren des Entwicklungscomputers \(p. 1005\)](#).

Abschnitte

- [Installieren von PHP \(p. 287\)](#)
- [Composer installieren \(p. 287\)](#)
- [Installieren des AWS SDK for PHP \(p. 288\)](#)
- [Installieren einer IDE oder eines Texteditors \(p. 288\)](#)

Installieren von PHP

Installieren Sie PHP und einige gängige Erweiterungen. Falls Sie keine Version bevorzugen, wählen Sie die neueste aus. Je nach Plattform und verfügbarem Paketmanager variieren die Schritte.

Verwenden Sie auf Amazon Linux yum:

```
$ sudo yum install php  
$ sudo yum install php-mbstring  
$ sudo yum install php-intl
```

Note

Zum Abrufen bestimmter PHP-Paketversionen, die der Version auf Ihrer Elastic Beanstalk-**PHP-Plattformversion** entsprechen, verwenden Sie den Befehl `yum search php`, um verfügbare Paketversionen zu finden, wie etwa `php72`, `php72-mbstring` und `php72-intl`. Verwenden Sie dann `sudo yum install package`, um diese zu installieren.

Verwenden Sie unter Ubuntu apt:

```
$ sudo apt install php-all-dev  
$ sudo apt install php-intl  
$ sudo apt install php-mbstring
```

Verwenden Sie auf OS-X brew:

```
$ brew install php  
$ brew install php-intl
```

Note

Zum Abrufen bestimmter PHP-Paketversionen, die der Version auf Ihrer Elastic Beanstalk-**PHP-Plattformversion** entsprechen, suchen Sie unter [Homebrew Formulae](#) verfügbare PHP-Versionen, wie etwa `php@7.2`. Verwenden Sie dann `brew install package`, um sie zu installieren. Je nach Version ist `php-intl` möglicherweise im Haupt-PHP-Paket enthalten und nicht als separates Paket vorhanden.

Installieren Sie unter Windows 10 das [Windows-Subsystem für Linux](#), um Ubuntu zu erhalten. Installieren Sie dann PHP mit apt. Besuchen Sie für frühere Versionen die Downloadseite unter windows.php.net, um PHP herunterzuladen, und lesen Sie [diese Seite](#), um Informationen über Erweiterungen zu erhalten.

Nach dem Installieren von PHP öffnen Sie Ihr Terminal erneut und führen Sie `php --version` aus, um sicherzustellen, dass die neue Version installiert wurde und die Standardeinstellung ist.

Composer installieren

Composer ist ein Manager zur Verwaltung von Abhängigkeiten für PHP. Sie können mit dem Tool Bibliotheken installieren, die Abhängigkeiten von Anwendungen verfolgen und Projekte für populäre PHP-Frameworks generieren.

Installieren Sie Composer mit dem PHP-Skript von getcomposer.org.

```
$ curl -s https://getcomposer.org/installer | php
```

Das Installationsprogramm generiert eine PHAR-Datei im aktuellen Verzeichnis. Verschieben Sie diese Datei in ein Verzeichnis, das in der Umgebungsvariablen PATH enthalten ist, damit es als ausführbare Datei genutzt werden kann.

```
$ mv composer.phar ~/.local/bin/composer
```

Installieren Sie Bibliotheken mit dem Befehl `require`.

```
$ composer require twig/twig
```

Composer fügt Bibliotheken hinzu, die Sie lokal in der Datei [composer.json](#) (p. 291) des Projekts installieren. Wenn Sie den Projektcode bereitstellen, verwendet Elastic Beanstalk Composer, um die in dieser Datei aufgelisteten Bibliotheken auf den Anwendungs-Instances der Umgebung zu installieren.

Wenn bei der Composer-Installation Probleme auftreten, schlagen Sie die [Composer-Dokumentation](#) nach.

Installieren des AWS SDK for PHP

Wenn Sie die AWS-Ressourcen aus der Anwendung heraus verwalten möchten, installieren Sie das AWS SDK for PHP. Sie können mit dem SDK for PHP beispielsweise Amazon DynamoDB (DynamoDB) zum Speichern von Benutzer- und Sitzungsinformationen verwenden, ohne eine relationale Datenbank zu erstellen.

Installieren Sie das SDK for PHP mit Composer.

```
$ composer require aws/aws-sdk-php
```

Weitere Informationen und Installationsanleitungen finden Sie auf der [AWS SDK for PHP-Homepage](#).

Installieren einer IDE oder eines Texteditors

Integrierte Entwicklungsumgebungen (IDEs) bieten zahlreiche Funktionen, mit denen die Anwendungsentwicklung vereinfacht wird. Falls Sie noch keine IDE für die PHP-Entwicklung eingesetzt haben, testen Sie Eclipse und PHPStorm, um die für Sie beste Lösung auszuwählen.

- [Installieren von Eclipse](#)
- [Installieren von PhpStorm](#)

Note

Mit einer IDE werden dem Projektordner möglicherweise Dateien hinzugefügt, die Sie unter Umständen nicht an die Quellüberwachung übergeben möchten. Damit diese Dateien nicht an die Quellüberwachung übergeben werden, setzen Sie `.gitignore` oder die Entsprechung Ihres Quellüberwachungstools ein.

Wenn Sie gerade mit dem Programmieren beginnen und nicht alle Funktionen einer IDE benötigen, können Sie [Sublime Text installieren](#).

Verwenden der Elastic Beanstalk-PHP-Plattform

Important

Amazon Linux 2-Plattformversionen unterscheiden sich grundlegend von Amazon Linux AMI-Plattformversionen (vor Amazon Linux 2). Diese unterschiedlichen Plattformgenerationen sind auf verschiedene Arten inkompatibel. Wenn Sie zu einer Amazon Linux 2-Plattformversion migrieren,

müssen Sie die Informationen in [the section called “Aktualisieren auf Amazon Linux 2” \(p. 503\)](#) lesen.

AWS Elastic Beanstalk unterstützt zahlreiche Plattformen für verschiedene Versionen der PHP-Programmiersprache. Diese Plattformen unterstützen PHP-Webanwendungen, die sowohl eigenständig als auch mit Composer ausgeführt werden können. Erfahren Sie mehr unter [PHP](#) im Dokument zu AWS Elastic Beanstalk-Plattformen.

Elastic Beanstalk bietet [Konfigurationsoptionen \(p. 646\)](#), mit denen sich die auf den EC2-Instances in der Elastic Beanstalk-Umgebung ausgeführte Software anpassen lässt. Sie können die von Ihrer Anwendung erforderlichen Umgebungsvariablen konfigurieren, die Protokollrotation an Amazon S3 aktivieren, Ordner in der Anwendungsquelle mit statischen Dateien zu vom Proxy-Server verwendeten Pfaden zuordnen und gemeinsame PHP-Initialisierungseinstellungen festlegen.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das [Ändern der Konfiguration einer ausgeführten Umgebung \(p. 658\)](#) verfügbar. Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie [gespeicherte Konfigurationen \(p. 764\)](#) verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie [Konfigurationsdateien \(p. 722\)](#) einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Wenn Sie Composer verwenden, können Sie durch das [Einbinden einer composer.json-Datei \(p. 291\)](#) in das Quell-Bundle die Pakete während der Bereitstellung installieren.

Für eine erweiterte PHP-Konfiguration und PHP-Einstellungen, die nicht als Konfigurationsoptionen verfügbar sind, können Sie [mit Konfigurationsdateien eine INI-Datei bereitstellen \(p. 293\)](#), mit der die Standardeinstellungen von Elastic Beanstalk erweitert und überschrieben werden, oder Sie können zusätzliche Erweiterungen installieren.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, wenn vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe [Konfigurationsoptionen \(p. 646\)](#).

Details zu den verschiedenen Möglichkeiten für das Erweitern einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#).

Konfigurieren Ihrer PHP-Umgebung

Mit der Elastic Beanstalk-Konsole können Sie die Protokollrotation für Amazon S3 aktivieren, Variablen konfigurieren, die die Anwendung in der Umgebung lesen kann, und die PHP-Einstellungen ändern.

So konfigurieren Sie Ihre PHP-Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.

PHP-Einstellungen

- Proxy server (Proxy-Server) – Der Proxy-Server, der in Ihren Umgebungs-Instances verwendet werden soll. Standardmäßig wird nginx verwendet.
- Document root (Dokumentenstamm) – In diesem Ordner ist die Standardseite Ihrer Website enthalten. Falls sich die Willkommenseite nicht im Stamm des Quell-Bundles befindet, geben Sie den entsprechenden Ordner relativ zum Stammpfad an. Beispielsweise /public, wenn die Willkommenseite im Ordner public ist.
- Memory limit (Speicherlimit) – Der maximale Speicher, der von einem Skript zugeordnet werden kann. Beispiel, 512M.
- Zlib output compression (Zlib-Ausgabekomprimierung) – Legen Sie den Wert auf on fest, um Antworten zu komprimieren.
- Allow URL fopen (fopen für URLs zulassen) – Legen Sie den Wert auf off fest, damit Skripts keine Dateien von Remote-Speicherorten herunterladen können.
- Display errors (Fehler anzeigen) – Legen Sie den Wert auf on fest, damit interne Fehlermeldungen zum Debuggen angezeigt werden.
- Max execution time (Max. Ausführungszeit) – Gibt die maximale Zeitspanne (in Sekunden) an, die ein Skript ausgeführt werden kann, bevor es von der Umgebung beendet wird.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren) – Gibt an, ob die Protokolldateien für die Amazon EC2-Instances der Anwendung in den der Anwendung zugeordneten Amazon S3-Bucket kopiert werden sollen.

Statische Dateien

Um die Leistung zu verbessern, können Sie im Abschnitt Static files (Statische Dateien) den Proxy-Server so konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zur Verzeichniszuordnung fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, statt die Anfrage an Ihre Anwendung weiterzuleiten.

Weitere Informationen zum Konfigurieren statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter [the section called “Statische Dateien” \(p. 776\)](#).

Umgebungseigenschaften

Im Bereich Environment Properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Diese Einstellungen werden als Schlüssel-Wert-Paare an die Anwendung weitergeleitet.

Ihr Anwendungscode kann mithilfe der Funktion `$_SERVER` oder `get_cfg_var` auf Umgebungseigenschaften zugreifen.

```
$endpoint = $_SERVER['API_ENDPOINT'];
```

Weitere Informationen finden Sie unter [Umgebungseigenschaften und andere Softwareeinstellungen \(p. 621\)](#).

Der Namespace aws:elasticbeanstalk:container:php:phpini

Mithilfe einer [Konfigurationsdatei \(p. 722\)](#) können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Konfigurationsoptionen können durch den Elastic Beanstalk-Service oder die verwendete Plattform definiert und in Namespaces organisiert werden.

Sie können mithilfe des Namespace `aws:elasticbeanstalk:environment:proxy` den Proxy-Server der Umgebung auswählen.

Sie können mithilfe des Namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` den Umgebungsproxy so konfigurieren, dass statische Dateien bereitgestellt werden. Sie definieren Zuordnungen von virtuellen Pfaden zu Anwendungsverzeichnissen.

Die PHP-Plattform definiert Optionen im `aws:elasticbeanstalk:container:php:phpini`-Namespace, darunter auch eine Option, die in der Elastic Beanstalk-Konsole nicht verfügbar ist. `composer_options` legt fest, welche benutzerdefinierten Optionen verwendet werden sollen, wenn Sie mit Composer Abhängigkeiten über `composer.phar install` installieren. Weitere Informationen sowie die verfügbaren Optionen finden Sie unter <http://getcomposer.org/doc/03-cli.md#install>.

Die folgende [Beispielkonfigurationsdatei \(p. 722\)](#) gibt eine Option für statische Dateien an, mit der ein Verzeichnis mit dem Namen `staticimages` dem Pfad `/images` zugeordnet wird, und zeigt die Einstellungen jeder der im Namespace `aws:elasticbeanstalk:container:php:phpini` verfügbaren Optionen an:

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

Der Namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` ist auf Amazon Linux AMI-PHP-Plattformvarianten (vor Amazon Linux 2) nicht definiert.

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Installieren der Abhängigkeiten Ihrer Anwendung

Ihre Anwendung verfügt möglicherweise über Abhängigkeiten von anderen PHP-Paketen. Sie können Ihre Anwendung so konfigurieren, dass diese Abhängigkeiten auf Amazon Elastic Compute Cloud (Amazon EC2)-Instances der Umgebung installiert werden. Alternativ können Sie die Abhängigkeiten Ihrer Anwendung in das Quell-Bundle einschließen und sie mit der Anwendung bereitstellen. Im folgenden Abschnitt werden beide Möglichkeiten vorgestellt.

Verwenden einer Composer-Datei zum Installieren von Abhängigkeiten auf Instances

Mit einer `composer.json`-Datei im Stammverzeichnis der Projektquelle können Sie die von der Anwendung benötigten Pakete über Composer auf den Amazon EC2-Instances Ihrer Umgebung installieren.

Example `composer.json`

```
{  
    "require": {  
        "monolog/monolog": "1.0.*"  
    }  
}
```

Wenn eine `composer.json`-Datei vorhanden ist, führt Elastic Beanstalk `composer.phar install` aus, um Abhängigkeiten zu installieren. Sie können Optionen für den Befehl anfügen, indem Sie die [Option `composer_options` \(p. 291\)](#) im Namespace `aws:elasticbeanstalk:container:php:phpini` festlegen.

Einschließen von Abhängigkeiten in Quell-Bundles

Wenn Ihre Anwendung eine große Anzahl von Abhängigkeiten besitzt, kann deren Installation einige Zeit in Anspruch nehmen. Dies kann die Bereitstellung und Skalierung von Operationen erhöhen, da Abhängigkeiten auf jeder neuen Instance installiert sind.

Um negative Auswirkungen auf die Bereitstellungszeit zu vermeiden, verwenden Sie Composer in Ihrer Entwicklungsumgebung, um Abhängigkeiten aufzulösen und sie im Ordner `vendor` zu installieren.

So schließen Sie Abhängigkeiten im Quell-Bundle Ihrer Anwendung ein

1. Führen Sie den folgenden Befehl aus:

```
% composer install
```

2. Fügen Sie den generierten Ordner `vendor` zum Stamm des Quell-Bundles der Anwendung hinzu.

Wenn Elastic Beanstalk einen Ordner `vendor` auf der Instance findet, wird die Datei `composer.json` ignoriert (selbst, wenn sie vorhanden ist). Ihre Anwendung verwendet dann Abhängigkeiten aus dem Ordner `vendor`.

Aktualisieren von Composer

[PHP-Plattformversionen](#) werden mit der neuesten Composer-Version ausgeliefert, die zum Veröffentlichungszeitpunkt verfügbar ist. Um Composer-, PHP- und andere Bibliotheken auf dem neuesten Stand zu halten, sollten Sie bei jeder verfügbaren Plattformaktualisierung ein [Upgrade der Umgebung \(p. 491\)](#) ausführen.

Zwischen den Plattformaktualisierungen können Sie Composer auf den Instances der Umgebung mithilfe einer [Konfigurationsdatei \(p. 722\)](#) aktualisieren. Möglicherweise ist eine Composer-Aktualisierung erforderlich, wenn bei der Paketinstallation über eine Composer-Datei Fehler angezeigt werden oder Sie die neueste Plattformversion nicht nutzen können.

Example `.ebextensions/composer.config`

```
commands:
```

```
01updateComposer:  
  command: export COMPOSER_HOME=/root && /usr/bin/composer.phar self-update 1.4.1  
  
option_settings:  
  - namespace: aws:elasticbeanstalk:application:environment  
    option_name: COMPOSER_HOME  
    value: /root
```

Mit dieser Konfigurationsdatei wird Composer für die automatische Aktualisierung auf die Version 1.4.1 konfiguriert. Die neueste Version finden Sie auf der Composer-[Versionsseite](#) unter GitHub.

Note

Falls Sie die Versionsnummer aus dem Befehl `composer.phar self-update` weglassen, wird Composer jedes Mal auf die neueste verfügbare Version aktualisiert, wenn Sie neuen Quellcode bereitstellen und neue Instances von der Auto Scaling-Funktion zur Verfügung gestellt werden. Das kann zu fehlerhaften Skalierungen und Bereitstellungen führen, falls eine Composer-Version veröffentlicht wird, die mit Ihrer Anwendung inkompatibel ist.

Erweitern von "php.ini"

Verwenden Sie eine Konfigurationsdatei mit einem `files`-Block, um eine `.ini`-Datei zu `/etc/php.d/` auf den Instances der Umgebung hinzuzufügen. Von der Hauptkonfigurationsdatei `php.ini` werden die Einstellungen aus den Dateien in diesem Ordner in alphabetischer Reihenfolge abgerufen. Viele Erweiterungen werden standardmäßig über die Dateien in diesem Ordner aktiviert.

Example `.ebextensions/mongo.config`

```
files:  
  "/etc/php.d/99mongo.ini" :  
    mode: "000755"  
    owner: root  
    group: root  
    content: |  
      extension=mongo.so
```

Bereitstellen einer Laravel-Anwendung in Elastic Beanstalk

Bei Laravel handelt es sich um ein Open-Source-Framework mit Model-View-Controller (MVC) für PHP. In diesem Tutorial wird der Prozess für das Erstellen einer Laravel-Anwendung, dessen Bereitstellung in einer AWS Elastic Beanstalk-Umgebung und die Konfiguration zum Herstellen einer Verbindung mit einer Amazon Relational Database Service (Amazon RDS)-Datenbank-Instance erklärt.

Abschnitte

- [Voraussetzungen \(p. 294\)](#)
- [Starten einer Elastic Beanstalk-Umgebung \(p. 294\)](#)
- [Installieren von Laravel und Erstellen einer Website \(p. 295\)](#)
- [Bereitstellen Ihrer Anwendung \(p. 296\)](#)
- [Konfigurieren von Composer-Einstellungen \(p. 296\)](#)
- [Hinzufügen einer Datenbank zu Ihrer Umgebung \(p. 297\)](#)
- [Bereinigen \(p. 299\)](#)
- [Nächste Schritte \(p. 300\)](#)

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie über Kenntnisse über die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole verfügen. Sofern noch nicht geschehen, befolgen Sie die Anweisungen in [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#), um die erste Elastic Beanstalk-Umgebung zu starten.

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine -Shell für die Ausführung der Befehle. Befehle werden in Auflistungen mit einem vorangestellten Eingabeaufforderungs-Symbol (\$) und dem Namen des aktuellen Verzeichnisses angezeigt, falls angebracht.

```
~/eb-project$ this is a command  
this is output
```

Verwenden Sie auf Linux und macOS Ihren bevorzugten Shell- und Paket-Manager. Sie können unter Windows 10 das [Windows-Subsystem für Linux](#) installieren, um eine Windows-Version von Ubuntu und Bash zu erhalten.

Laravel erfordern PHP 5.5.9 oder höher und die mbstring-Erweiterung für PHP. In diesem Tutorial verwenden wir PHP 7.0 und die entsprechende Elastic Beanstalk-Plattformversion. Installieren Sie PHP und Composer mithilfe der Anleitungen unter [Einrichten der PHP-Entwicklungsumgebung \(p. 286\)](#).

Starten einer Elastic Beanstalk-Umgebung

Verwenden Sie die Elastic Beanstalk-Konsole, um eine Elastic Beanstalk-Umgebung zu erstellen. Wählen Sie die Plattform PHP und übernehmen Sie die Standardeinstellungen und den Beispielcode.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced.
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus, und wenn Sie bereit sind, wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2) zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.

- Load balancer (Load Balancer) – Ein Elastic Load Balancing-Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn sie beendet wird oder nicht verfügbar ist.
- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – Zwei CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form `subdomain.region.elasticbeanstalk.com` weiterleitet.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Note

Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#).

Installieren von Laravel und Erstellen einer Website

Composer kann Laravel installieren und ein Arbeitsprojekt mit einem Befehl erstellen:

```
~$ composer create-project --prefer-dist laravel/laravel eb-laravel
Installing laravel/laravel (v5.5.28)
- Installing laravel/laravel (v5.5.28): Downloading (100%)
Created project in eb-laravel
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies (including require-dev)

Package operations: 70 installs, 0 updates, 0 removals
- Installing symfony/thanks (v1.0.7): Downloading (100%)
- Installing hamcrest/hamcrest-php (v2.0.0): Downloading (100%)
- Installing mockery/mockery (1.0): Downloading (100%)
- Installing vlucas/phpdotenv (v2.4.0): Downloading (100%)
- Installing symfony/css-selector (v3.4.8): Downloading (100%)
- Installing tijsverkoyen/css-to-inline-styles (2.2.1): Downloading (100%)
...
...
```

Composer installiert Laravel und seine Abhängigkeiten und generiert ein Standardprojekt.

Wenn Probleme bei der Installation von Laravel auftreten, gehen Sie zum Installationsthema in der offiziellen Dokumentation: <https://laravel.com/docs/5.2>

Bereitstellen Ihrer Anwendung

Erstellen Sie ein [Quell-Bundle \(p. 412\)](#) mit den Dateien, die von Composer erstellt wurden. Der folgende Befehl erstellt ein Quell-Bundle namens `laravel-default.zip`. Die Dateien im Ordner `vendor` werden ausgeschlossen, weil sie viel Platz benötigen und zum Bereitstellen der Anwendung für Elastic Beanstalk nicht erforderlich sind.

```
~/eb-laravel$ zip ..../laravel-default.zip -r *.[^.]* -x "vendor/*"
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um Laravel in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Note

Um das Quell-Bundle weiter zu optimieren, initialisieren Sie ein Git-Repository und verwenden den Befehl [git archive \(p. 413\)](#), um das Quell-Bundle zu erstellen. Das Laravel-Standardprojekt enthält eine `.gitignore`-Datei, die Git anweist, den Ordner `vendor` sowie andere Dateien auszuschließen, die für die Bereitstellung nicht benötigt werden.

Konfigurieren von Composer-Einstellungen

Klicken Sie nach Abschluss der Bereitstellung auf die URL, um die Laravel-Anwendung im Browser zu öffnen:

Forbidden

You don't have permission to access / on this server.

Was ist das? Standardmäßig bedient Elastic Beanstalk das Stammverzeichnis Ihres Projekts auf dem Stammpfad der Website. In diesem Fall liegt die Standardseite (`index.php`) jedoch eine Ebene weiter unten im `public`-Ordner. Sie können dies überprüfen, indem Sie `/public` zur URL hinzufügen. Beispiel, `http://laravel.us-east-2.elasticbeanstalk.com/public`.

Damit die Anwendung Laravel vom Stammverzeichnis aus genutzt werden kann, konfigurieren Sie mit der Elastic Beanstalk-Konsole das Dokumentenstammverzeichnis für die Website.

So konfigurieren Sie das Stammverzeichnis Ihrer Website

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Geben Sie für Document root (Basisverzeichnis) **/public** ein.
6. Wählen Sie Apply (Anwenden) aus.
7. Wenn die Aktualisierung abgeschlossen ist, klicken Sie auf die URL, um Ihre Website erneut im Browser zu öffnen.

Laravel

[DOCUMENTATION](#)

[LARACASTS](#)

[NEWS](#)

[FORGE](#)

[GITHUB](#)

So weit, so gut. Als Nächstes fügen Sie eine Datenbank zu Ihrer Umgebung hinzu und konfigurieren Laravel so, dass eine Verbindung hergestellt wird.

Hinzufügen einer Datenbank zu Ihrer Umgebung

Starten Sie eine RDS-DB-Instance in Ihrer Elastic Beanstalk-Umgebung. Sie können MySQL-, SQLServer- oder PostgreSQL-Datenbanken mit Laravel auf Elastic Beanstalk verwenden. In diesem Beispiel wird MySQL verwendet.

Note

Das Ausführen einer Amazon RDS-Instance in Ihrer Elastic Beanstalk-Umgebung eignet sich hervorragend für Entwicklung und Tests, aber sie bindet den Lebenszyklus Ihrer Datenbank an Ihre Umgebung. Anweisungen zum Herstellen einer Verbindung mit einer Datenbank, die außerhalb der Umgebung ausgeführt wird, finden Sie unter [Starten von und Verbinden mit einer externen Amazon RDS-Instance in einer Standard-VPC \(p. 974\)](#).

So fügen Sie Ihrer Elastic Beanstalk-Umgebung eine RDS-DB-Instance hinzu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie für Engine mysql aus.
6. Geben Sie einen Master-username (Benutzernamen) und ein Master-password (Passwort) ein. Elastic Beanstalk stellt diese Werte für Ihre Anwendung mithilfe von Umgebungseigenschaften bereit.
7. Wählen Sie Apply (Anwenden) aus.

Das Erstellen einer Datenbank-Instance dauert ca. zehn Minuten. In der Zwischenzeit können Sie Ihren Quell-Code aktualisieren, um Verbindungsinformationen aus der Umgebung zu lesen. Elastic Beanstalk bietet Verbindungsdetails mithilfe von Umgebungsvariablen, wie z. B. `RDS_HOSTNAME`, worauf Sie von Ihrer Anwendung aus zugreifen können.

Die Laravel-Datenbankkonfiguration ist in der Datei `database.php` im config-Ordner in Ihrem Projektcode gespeichert. Suchen Sie den Eintrag `mysql` und ändern Sie die Variablen `host`, `database`, `username` und `password` so, dass sie die entsprechenden Werte aus Elastic Beanstalk lesen:

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,
        ],
    ...
}
```

Um zu überprüfen, ob die Datenbankverbindung ordnungsgemäß konfiguriert ist, fügen Sie den Code zu `index.php` hinzu, um eine Verbindung zur Datenbank herzustellen und Code zur Standardantwort hinzuzufügen:

Example `~/Eb-laravel/public/index.php`

```
...
if(DB::connection()->getDatabaseName())
{
    echo "Connected to database ".DB::connection()->getDatabaseName();
```

```
}
```

```
$response->send();
```

```
...
```

Wenn die DB-Instance den Start abgeschlossen hat, bündeln Sie die aktualisierte Anwendung und stellen Sie sie in Ihrer Umgebung bereit.

So aktualisieren Sie Ihre Elastic Beanstalk-Umgebung

1. Erstellen Sie ein neues Quell-Bundle:

```
~/eb-laravel$ zip ..../laravel-v2-rds.zip -r * .[^.]* -x "vendor/*"
```

2. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
3. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

4. Klicken Sie auf die Schaltfläche Upload and Deploy (Hochladen und Bereitstellen).
5. Wählen Sie Browse und laden Sie `laravel-v2-rds.zip` hoch.
6. Wählen Sie Deploy (Bereitstellen) aus.

Die Bereitstellung einer neuen Version Ihrer Anwendung dauert weniger als eine Minute. Wenn die Bereitstellung abgeschlossen ist, aktualisieren Sie die Webseite erneut, um zu überprüfen, ob die Datenbankverbindung erfolgreich war:

Connected to database `ebdb`

Laravel

DOCUMENTATION

LARACASTS

NEWS

FORGE

GITHUB

Bereinigen

Wenn Sie nicht mehr mit Elastic Beanstalk arbeiten, können Sie Ihre Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 554\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) und dann Terminate environment (Umgebung beenden) aus.
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Außerdem können Sie Datenbankressourcen beenden, die außerhalb Ihrer Elastic Beanstalk-Umgebung erstellt wurden. Beim Beenden einer Amazon RDS-DB-Instance können Sie einen Snapshot erstellen und die Daten zu einem späteren Zeitpunkt in einer anderen Instance wiederherstellen.

So beenden Sie Ihre RDS DB-Instance

1. Öffnen Sie die [Amazon RDS-Konsole](#).
2. Wählen Sie Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus.
4. Wählen Sie Actions und anschließend Delete.
5. Wählen Sie, ob Sie einen Snapshot erstellen, und wählen Sie anschließend Delete (Löschen).

Nächste Schritte

Weitere Informationen zu Laravel finden Sie im Tutorial unter [laravel.com](#).

Im weiteren Verlauf der Anwendungsentwicklung möchten Sie vermutlich Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die [Elastic Beanstalk Command Line Interface \(p. 1008\)](#) (EB CLI) stellt benutzerfreundliche Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk-Umgebungen über die Befehlszeile bereit.

In diesem Tutorial haben Sie die Elastic Beanstalk-Konsole zum Konfigurieren von Composer-Optionen verwendet. Um diese Konfiguration zum Teil der Anwendungsquelle zu machen, können Sie eine Konfigurationsdatei wie die folgende verwenden.

Example .ebextensions/composer.config

```
option_settings:  
  aws:elasticbeanstalk:container:php:phpini:  
    document_root: /public
```

Weitere Informationen finden Sie unter [Erweiterte Umgebungsanpassung mit Konfigurationsdateien \(.ebextensions\) \(p. 722\)](#).

Das Ausführen einer Amazon RDS-DB-Instance in Ihrer Elastic Beanstalk-Umgebung eignet sich hervorragend für Entwicklung und Tests, aber sie bindet den Lebenszyklus Ihrer Datenbank an Ihre Umgebung. Anweisungen zum Herstellen einer Verbindung mit einer Datenbank, die außerhalb der Umgebung ausgeführt wird, finden Sie unter [Hinzufügen einer Amazon RDS-DB-Instance zur PHP-Anwendungsumgebung \(p. 350\)](#).

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, [konfigurieren Sie einen benutzerdefinierten Domänennamen \(p. 644\)](#) für die Umgebung und [aktivieren Sie HTTPS \(p. 778\)](#) für sichere Verbindungen.

Bereitstellen einer CakePHP-Anwendung in Elastic Beanstalk

CakePHP ist ein Open-Source-MVC-Framework für PHP. In diesem Tutorial wird der Prozess für das Erstellen eines CakePHP-Projekts, dessen Bereitstellung in einer Elastic Beanstalk-Umgebung und die Konfiguration zum Herstellen einer Verbindung mit einer Amazon RDS-Datenbank-Instance erklärt.

Abschnitte

- [Voraussetzungen \(p. 301\)](#)
- [Starten einer Elastic Beanstalk-Umgebung \(p. 301\)](#)
- [Installieren von CakePHP und Erstellen einer Website \(p. 302\)](#)
- [Bereitstellen Ihrer Anwendung \(p. 303\)](#)
- [Hinzufügen einer Datenbank zu Ihrer Umgebung \(p. 305\)](#)
- [Bereinigen \(p. 307\)](#)
- [Nächste Schritte \(p. 307\)](#)

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie über Kenntnisse über die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole verfügen. Sofern noch nicht geschehen, befolgen Sie die Anweisungen in [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#), um die erste Elastic Beanstalk-Umgebung zu starten.

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine -Shell für die Ausführung der Befehle. Befehle werden in Auflistungen mit einem vorangestellten Eingabeaufforderungs-Symbol (\$) und dem Namen des aktuellen Verzeichnisses angezeigt, falls angebracht.

```
~/eb-project$ this is a command
this is output
```

Verwenden Sie auf Linux und macOS Ihren bevorzugten Shell- und Paket-Manager. Sie können unter Windows 10 das [Windows-Subsystem für Linux](#) installieren, um eine Windows-Version von Ubuntu und Bash zu erhalten.

CakePHP erfordert PHP 5.5.9 oder neuer und die mbstring- und intl-Erweiterungen für PHP. In diesem Tutorial verwenden wir PHP 7.0 und die entsprechende Elastic Beanstalk-Plattformversion. Installieren Sie PHP und Composer mithilfe der Anleitungen unter [Einrichten der PHP-Entwicklungsumgebung \(p. 286\)](#).

Starten einer Elastic Beanstalk-Umgebung

Verwenden Sie die Elastic Beanstalk-Konsole, um eine Elastic Beanstalk-Umgebung zu erstellen. Wählen Sie die Plattform PHP und übernehmen Sie die Standardeinstellungen und den Beispielcode.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced.
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.

3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus, und wenn Sie bereit sind, wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2) zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer) – Ein Elastic Load Balancing-Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn sie beendet wird oder nicht verfügbar ist.
- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – Zwei CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form **subdomain.region.elasticbeanstalk.com** weiterleitet.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Note

Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#).

Installieren von CakePHP und Erstellen einer Website

Composer kann CakePHP installieren und ein Arbeitsprojekt mit einem Befehl erstellen:

```
~$ composer create-project --prefer-dist cakephp/app eb-cake
Installing cakephp/app (3.6.0)
- Installing cakephp/app (3.6.0): Loading from cache
Created project in eb-cake2
Loading composer repositories with package information
Updating dependencies (including require-dev)

Package operations: 46 installs, 0 updates, 0 removals
- Installing cakephp/plugin-installer (1.1.0): Loading from cache
- Installing aura/intl (3.0.0): Loading from cache
...
...
```

Composer installiert CakePHP und etwa 20 Abhängigkeiten und generiert ein Standardprojekt.

Wenn Probleme beim Installieren von CakePHP auftreten, besuchen Sie das Thema zur Installation in der offiziellen Dokumentation: <http://book.cakephp.org/3.0/en/installation.html>

Bereitstellen Ihrer Anwendung

Erstellen Sie ein [Quell-Bundle \(p. 412\)](#) mit den Dateien, die von Composer erstellt wurden. Der folgende Befehl erstellt ein Quell-Bundle namens `cake-default.zip`. Die Dateien im Ordner `vendor` werden ausgeschlossen, weil sie viel Platz benötigen und zum Bereitstellen der Anwendung für Elastic Beanstalk nicht erforderlich sind.

```
eb-cake zip ../cake-default.zip -r * .[^.]* -x "vendor/*"
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um CakePHP in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Note

Um das Quell-Bundle weiter zu optimieren, initialisieren Sie ein Git-Repository und verwenden den Befehl [git archive \(p. 413\)](#), um das Quell-Bundle zu erstellen. Das Symfony-Standardprojekt enthält eine `.gitignore`-Datei, die Git anweist, den Ordner `vendor` sowie andere Dateien auszuschließen, die für die Bereitstellung nicht benötigt werden.

Wenn der Vorgang abgeschlossen ist, klicken Sie auf die URL, um Ihre CakePHP-Anwendung im Browser zu öffnen:

The screenshot shows the CakePHP 3.6.1 Red Velvet welcome page. At the top is a red header with the CakePHP logo and the text "Welcome to CakePHP 3.6.1 Red Velvet. Build fast. Grow solid.". Below the header is a yellow warning message: "⚠ Please be aware that this page will not be shown if you turn off debug mode unless you replace src/Template/Pages/home.ctp with". The main content area is divided into several sections:

- Environment**
 - ��色感叹号图标: Your version of PHP is 5.6.0 or higher (detected 7.1.13).
 - 绿色感叹号图标: Your version of PHP has the mbstring extension loaded.
 - 绿色感叹号图标: Your version of PHP has the openssl extension loaded.
 - 绿色感叹号图标: Your version of PHP has the intl extension loaded.
- Filesystem**
 - 绿色感叹号图标: Your tmp directory is writable.
 - 绿色感叹号图标: Your logs directory is writable.
 - 绿色感叹号图标: The FileEngine is being used for core caching. edit config/app.php
- Database**
 - 红色感叹号图标: CakePHP is NOT able to connect to the database.
Connection to database could not be established: SQLSTATE[HY000]
[2002] No such file or directory
- DebugKit**
 - 绿色感叹号图标: DebugKit is loaded.
- Editing this Page**
 - 红色叉号图标: To change the content of this page, edit: src/Template/Pages/home.ctp.
 - 红色叉号图标: You can also add some CSS styles for your pages at: webroot/css/.
- Getting Started**
 - 书籍图标: [CakePHP 3.0 Docs](#)
 - 书籍图标: [The 15 min Bookmarker Tutorial](#)
 - 书籍图标: [The 15 min Blog Tutorial](#)
 - 书籍图标: [The 15 min CMS Tutorial](#)
- More about Cake**

So weit, so gut. Als Nächstes fügen Sie eine Datenbank zu Ihrer Umgebung hinzu und konfigurieren CakePHP so, dass eine Verbindung hergestellt wird.

Hinzufügen einer Datenbank zu Ihrer Umgebung

Starten Sie eine Amazon RDS-Datenbank-Instance in Ihrer Elastic Beanstalk-Umgebung. Sie können MySQL-, SQLServer- oder PostgreSQL-Datenbanken mit CakePHP auf Elastic Beanstalk verwenden. In diesem Beispiel wird PostgreSQL verwendet.

So fügen Sie Ihrer Elastic Beanstalk-Umgebung eine Amazon RDS-DB-Instance hinzu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie unter Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie für DB engine postgres aus.
6. Geben Sie einen Master-username (Benutzernamen) und ein Master-password (Passwort) ein. Elastic Beanstalk stellt diese Werte für Ihre Anwendung mithilfe von Umgebungseigenschaften bereit.
7. Wählen Sie Apply (Anwenden) aus.

Das Erstellen einer Datenbank-Instance dauert ca. zehn Minuten. In der Zwischenzeit können Sie Ihren Quell-Code aktualisieren, um Verbindungsinformationen aus der Umgebung zu lesen. Elastic Beanstalk bietet Verbindungsdetails mithilfe von Umgebungsvariablen, wie z. B. `RDS_HOSTNAME`, worauf Sie von Ihrer Anwendung aus zugreifen können.

Die CakePHP-Datenbankkonfiguration befindet sich in der Datei `app.php` im `config`-Ordner in Ihrem Projektcode. Öffnen Sie diese Datei und fügen Sie einige Code hinzu, der die Umgebungsvariablen aus `$_SERVER` liest und sie den lokalen Variablen zuweist. Fügen Sie die hervorgehobenen Zeilen im untenstehenden Beispiel nach der ersten Zeile ein (`<?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 [
    ...
]
```

Die Datenbankverbindung wird weiter unten in `app.php` konfiguriert. Suchen Sie den folgenden Abschnitt und ändern Sie die Standard-Datenquellenkonfiguration mit dem Namen des Treibers für Ihre Datenbank-Engine (Mysql, Sqlserver oder Postgres) und legen Sie die `host`-, `username`-, `password`- und `database`-Variablen so fest, dass sie die entsprechenden Werte aus Elastic Beanstalk lesen:

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,  
    ...
```

Wenn die DB-Instance den Start abgeschlossen hat, bündeln Sie die aktualisierte Anwendung und stellen Sie sie in Ihrer Umgebung bereit:

So aktualisieren Sie Ihre Elastic Beanstalk-Umgebung

1. Erstellen Sie ein neues Quell-Bundle:

```
~/eb-cake$ zip ../cake-v2-rds.zip -r * .[^.]* -x "vendor/*"
```

2. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
3. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

4. Klicken Sie auf die Schaltfläche Upload and Deploy (Hochladen und Bereitstellen).
5. Wählen Sie Browse und laden Sie `cake-v2-rds.zip` hoch.
6. Wählen Sie Deploy (Bereitstellen) aus.

Die Bereitstellung einer neuen Version Ihrer Anwendung dauert weniger als eine Minute. Wenn die Bereitstellung abgeschlossen ist, aktualisieren Sie die Webseite erneut, um zu überprüfen, ob die Datenbankverbindung erfolgreich war:

Database



CakePHP is able to connect to the database.

Bereinigen

Wenn Sie nicht mehr mit Elastic Beanstalk arbeiten, können Sie Ihre Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 531\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) und dann Terminate environment (Umgebung beenden) aus.
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Außerdem können Sie Datenbankressourcen beenden, die außerhalb Ihrer Elastic Beanstalk-Umgebung erstellt wurden. Beim Beenden einer Amazon RDS-DB-Instance können Sie einen Snapshot erstellen und die Daten zu einem späteren Zeitpunkt in einer anderen Instance wiederherstellen.

So beenden Sie Ihre RDS DB-Instance

1. Öffnen Sie die [Amazon RDS-Konsole](#).
2. Wählen Sie Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus.
4. Wählen Sie Actions und anschließend Delete.
5. Wählen Sie, ob Sie einen Snapshot erstellen, und wählen Sie anschließend Delete (Löschen).

Nächste Schritte

Weitere Informationen zu CakePHP finden Sie im Buch unter book.cakephp.org.

Im weiteren Verlauf der Anwendungsentwicklung möchten Sie vermutlich Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die [Elastic Beanstalk Command Line Interface \(p. 1008\)](#) (EB CLI) stellt

benutzerfreundliche Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk-Umgebungen über die Befehlszeile bereit.

Das Ausführen einer Amazon RDS-DB-Instance in Ihrer Elastic Beanstalk-Umgebung eignet sich hervorragend für Entwicklung und Tests, aber sie bindet den Lebenszyklus Ihrer Datenbank an Ihre Umgebung. Anweisungen zum Herstellen einer Verbindung mit einer Datenbank, die außerhalb der Umgebung ausgeführt wird, finden Sie unter [Hinzufügen einer Amazon RDS-DB-Instance zur PHP-Anwendungsumgebung \(p. 350\)](#).

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, [konfigurieren Sie einen benutzerdefinierten Domänennamen \(p. 644\)](#) für die Umgebung und [aktivieren Sie HTTPS \(p. 778\)](#) für sichere Verbindungen.

Bereitstellen einer Symfony-Anwendung in Elastic Beanstalk

[Symfony](#) ist ein Open-Source-Framework für die Entwicklung dynamischer PHP-Webanwendungen. Dieses Tutorial erläutert den Prozess zum Generieren einer Symfony-Anwendung und zum Bereitstellen dieser Anwendung in einer AWS Elastic Beanstalk-Umgebung.

Abschnitte

- [Voraussetzungen \(p. 308\)](#)
- [Starten einer Elastic Beanstalk-Umgebung \(p. 309\)](#)
- [Installieren von Symfony und Generieren einer Website \(p. 310\)](#)
- [Bereitstellen Ihrer Anwendung \(p. 310\)](#)
- [Konfigurieren von Composer-Einstellungen \(p. 311\)](#)
- [Bereinigen \(p. 312\)](#)
- [Nächste Schritte \(p. 313\)](#)

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie über Kenntnisse über die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole verfügen. Sofern noch nicht geschehen, befolgen Sie die Anweisungen in [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#), um die erste Elastic Beanstalk-Umgebung zu starten.

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine -Shell für die Ausführung der Befehle. Befehle werden in Auflistungen mit einem vorangestellten Eingabeaufforderungs-Symbol (\$) und dem Namen des aktuellen Verzeichnisses angezeigt, falls angebracht.

```
~/eb-project$ this is a command
this is output
```

Verwenden Sie auf Linux und macOS Ihren bevorzugten Shell- und Paket-Manager. Sie können unter Windows 10 das [Windows-Subsystem für Linux](#) installieren, um eine Windows-Version von Ubuntu und Bash zu erhalten.

Symfony 4.3 setzt PHP ab Version 7.1 und die Erweiterung `intl` für PHP voraus. In diesem Tutorial verwenden wir PHP 7.2 und die entsprechende Elastic Beanstalk-Plattformversion. Installieren Sie PHP und Composer mithilfe der Anleitungen unter [Einrichten der PHP-Entwicklungsumgebung \(p. 286\)](#).

Starten einer Elastic Beanstalk-Umgebung

Verwenden Sie die Elastic Beanstalk-Konsole, um eine Elastic Beanstalk-Umgebung zu erstellen. Wählen Sie die Plattform PHP und übernehmen Sie die Standardeinstellungen und den Beispielcode.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced.
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus, und wenn Sie bereit sind, wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2) zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer) – Ein Elastic Load Balancing-Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn sie beendet wird oder nicht verfügbar ist.
- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – Zwei CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form `subdomain.region.elasticbeanstalk.com` weiterleitet.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Note

Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#).

Installieren von Symfony und Generieren einer Website

Composer kann Symfony installieren und ein Arbeitsprojekt mit einem Befehl erstellen:

```
~$ composer create-project symfony/website-skeleton eb-symfony
Installing symfony/website-skeleton (v4.3.99)
- Installing symfony/website-skeleton (v4.3.99): Downloading (100%)
Created project in eb-symfony
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Installing symfony/flex (v1.4.5): Downloading (100%)
Symfony operations: 1 recipe (539d006017ad5ef71beab4a2e2870e9a)
- Configuring symfony/flex (>=1.0): From github.com/symfony/recipes:master
Loading composer repositories with package information
Updating dependencies (including require-dev)
Restricting packages listed in "symfony/symfony" to "4.3.*"
Package operations: 103 installs, 0 updates, 0 removals
- Installing ocreamius/package-versions (1.4.0): Loading from cache
...
...
```

Composer installiert Symfony und seine Abhängigkeiten und generiert ein Standardprojekt.

Wenn bei der Installation von Symfony Probleme auftreten, rufen Sie das Installationsthema in der offiziellen Dokumentation auf: symfony.com/doc/3.4/setup.html

Bereitstellen Ihrer Anwendung

Rufen Sie das Projektverzeichnis auf.

```
~$ cd eb-symfony
```

Erstellen Sie ein [Quell-Bundle \(p. 412\)](#) mit den Dateien, die von Composer erstellt wurden. Der folgende Befehl erstellt ein Quell-Bundle namens `symfony-default.zip`. Die Dateien im Ordner `vendor` werden ausgeschlossen, weil sie viel Platz benötigen und zum Bereitstellen der Anwendung für Elastic Beanstalk nicht erforderlich sind.

```
eb-symfony$ zip ../symfony-default.zip -r * .[^.]* -x "vendor/*"
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um Symfony in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Note

Um das Quell-Bundle weiter zu optimieren, initialisieren Sie ein Git-Repository und verwenden den Befehl [git archive \(p. 413\)](#), um das Quell-Bundle zu erstellen. Das Symfony-Standardprojekt enthält eine `.gitignore`-Datei, die Git anweist, den Ordner `vendor` sowie andere Dateien auszuschließen, die für die Bereitstellung nicht benötigt werden.

Konfigurieren von Composer-Einstellungen

Klicken Sie nach Abschluss der Bereitstellung auf die URL, um die Symfony-Anwendung im Browser zu öffnen:

Forbidden

You don't have permission to access / on this server.

Was ist das? Standardmäßig bedient Elastic Beanstalk das Stammverzeichnis Ihres Projekts auf dem Stammpfad der Website. In diesem Fall liegt die Standardseite (`app.php`) jedoch eine Ebene weiter unten im web-Ordner. Sie können dies überprüfen, indem Sie `/public` zur URL hinzufügen. Beispiel, <http://symfony.us-east-2.elasticbeanstalk.com/public>.

Damit die Anwendung Symfony vom Stammverzeichnis aus genutzt werden kann, konfigurieren Sie mit der Elastic Beanstalk-Konsole das Dokumentenstammverzeichnis für die Website.

So konfigurieren Sie das Stammverzeichnis Ihrer Website

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Geben Sie für Document root (Basisverzeichnis) `/public` ein.
6. Wählen Sie Apply (Anwenden) aus.
7. Wenn die Aktualisierung abgeschlossen ist, klicken Sie auf die URL, um Ihre Website erneut im Browser zu öffnen.

Welcome to Symfony 4.3.4



Your application is now ready. You can start working on it at:
`/var/app/current/`

What's next?



Read the documentation to learn
[How to create your first page in Symfony](#)

You're seeing this page because debug mode is enabled and you haven't configured any home pages.

Bereinigen

Wenn Sie nicht mehr mit Elastic Beanstalk arbeiten, können Sie Ihre Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 554\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) und dann Terminate environment (Umgebung beenden) aus.
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Nächste Schritte

Weitere Informationen über Symfony finden Sie unter [symfony.com](#) in [What is Symfony?](#).

Im weiteren Verlauf der Anwendungsentwicklung möchten Sie vermutlich Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die [Elastic Beanstalk Command Line Interface \(p. 1008\)](#) (EB CLI) stellt benutzerfreundliche Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk-Umgebungen über die Befehlszeile bereit.

In diesem Tutorial haben Sie die Elastic Beanstalk-Konsole zum Konfigurieren von Composer-Optionen verwendet. Um diese Konfiguration zum Teil der Anwendungsquelle zu machen, können Sie eine Konfigurationsdatei wie die folgende verwenden.

Example .ebextensions/composer.config

```
option_settings:  
  aws:elasticbeanstalk:container:php:phpini:  
    document_root: /public
```

Weitere Informationen finden Sie unter [Erweiterte Umgebungsanpassung mit Konfigurationsdateien \(.ebextensions\) \(p. 722\)](#).

Symfony verwendet seine eigenen Konfigurationsdateien zum Konfigurieren von Datenbankverbindungen. Anleitungen zum Herstellen einer Verbindung zwischen einer Datenbank und Symfony finden Sie unter [Verbinden mit einer Datenbank mit Symfony \(p. 352\)](#).

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, [konfigurieren Sie einen benutzerdefinierten Domänennamen \(p. 644\)](#) für die Umgebung und [aktivieren Sie HTTPS \(p. 778\)](#) für sichere Verbindungen.

Bereitstellen einer PHP-Anwendung mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank zu Elastic Beanstalk

In diesem Tutorial wird beschrieben, wie Sie außerhalb von AWS Elastic Beanstalk [eine RDS-DB-Instance starten \(p. 973\)](#). Zudem erfahren Sie, wie Sie eine Umgebung mit hoher Verfügbarkeit (High Availability, HA) konfigurieren und die Verbindung über eine ausgeführte PHP-Anwendung herstellen. Wenn Sie eine DB-Instance außerhalb von Elastic Beanstalk ausführen, wird die Datenbank vom Lebenszyklus Ihrer Umgebung entkoppelt. Auf diese Weise können Sie eine Verbindung mit der gleichen Datenbank aus mehreren Umgebungen herstellen, eine Datenbank gegen eine andere austauschen oder eine Blau/Grün-Bereitstellung, ohne dass dies Auswirkungen auf Ihre Datenbank hat.

Das Tutorial verwendet eine [PHP-Beispielanwendung](#) mit einer MySQL-Datenbank zum Speichern der vom Benutzer bereitgestellten Textdaten. In der Beispielanwendung werden [Konfigurationsdateien \(p. 722\)](#) verwendet, um [PHP-Einstellungen \(p. 291\)](#) zu konfigurieren und eine Tabelle in der Datenbank zu erstellen. Außerdem erfahren Sie, wie Sie eine [Composer-Datei \(p. 291\)](#) zum Installieren von Paketen während der Bereitstellung verwenden.

Abschnitte

- [Voraussetzungen \(p. 314\)](#)
- [Starten einer DB-Instance in Amazon RDS \(p. 314\)](#)
- [Erstellen einer Elastic Beanstalk-Umgebung \(p. 316\)](#)

- Konfigurieren von Sicherheitsgruppen, Umgebungseigenschaften und Skalierung (p. 317)
- Bereitstellen der Beispielanwendung (p. 320)
- Bereinigen (p. 321)
- Nächste Schritte (p. 322)

Voraussetzungen

Bevor Sie beginnen, laden Sie das Beispielanwendungs-Quell-Bundle von GitHub herunter: [eb-demo-php-simple-app-1.3.zip](#)

Die Verfahren in diesem Tutorial für Amazon Relational Database Service (Amazon RDS)-Aufgaben gehen davon aus, dass Sie Ressourcen in einer standardmäßigen [Amazon Virtual Private Cloud](#) (Amazon VPC) starten. Alle neuen Konten umfassen eine Standard-VPC in jeder Region. Wenn Sie nicht über eine Standard-VPC verfügen, variieren die Verfahren. Weitere Anweisungen für EC2-Classic und benutzerdefinierte VPC-Plattformen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon RDS \(p. 973\)](#).

Starten einer DB-Instance in Amazon RDS

Um eine externe Datenbank mit einer Anwendung zu verwenden, die in Elastic Beanstalk ausgeführt wird, starten Sie zuerst eine DB-Instance mit Amazon RDS. Wenn Sie eine Instance mit Amazon RDS starten, ist diese gänzlich unabhängig von Elastic Beanstalk und den Elastic Beanstalk-Umgebungen. Sie wird von Elastic Beanstalk weder beendet noch überwacht.

Verwenden Sie die Amazon RDS-Konsole, um eine DB-Instance mit MySQL und Multi-AZ-Bereitstellung zu starten. Eine Multi-AZ-Bereitstellung gewährleistet die Ausführung von Failovern für Ihre Datenbank, sodass sie auch bei einem Ausfall der DB-Quell-Instance verfügbar bleibt.

So starten Sie eine RDS DB-Instance in einer Standard-VPC

1. Öffnen Sie die [RDS-Konsole](#).
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie Create database (Datenbank erstellen) aus.
4. Wählen Sie Standard Create (Standarderstellung).

Important

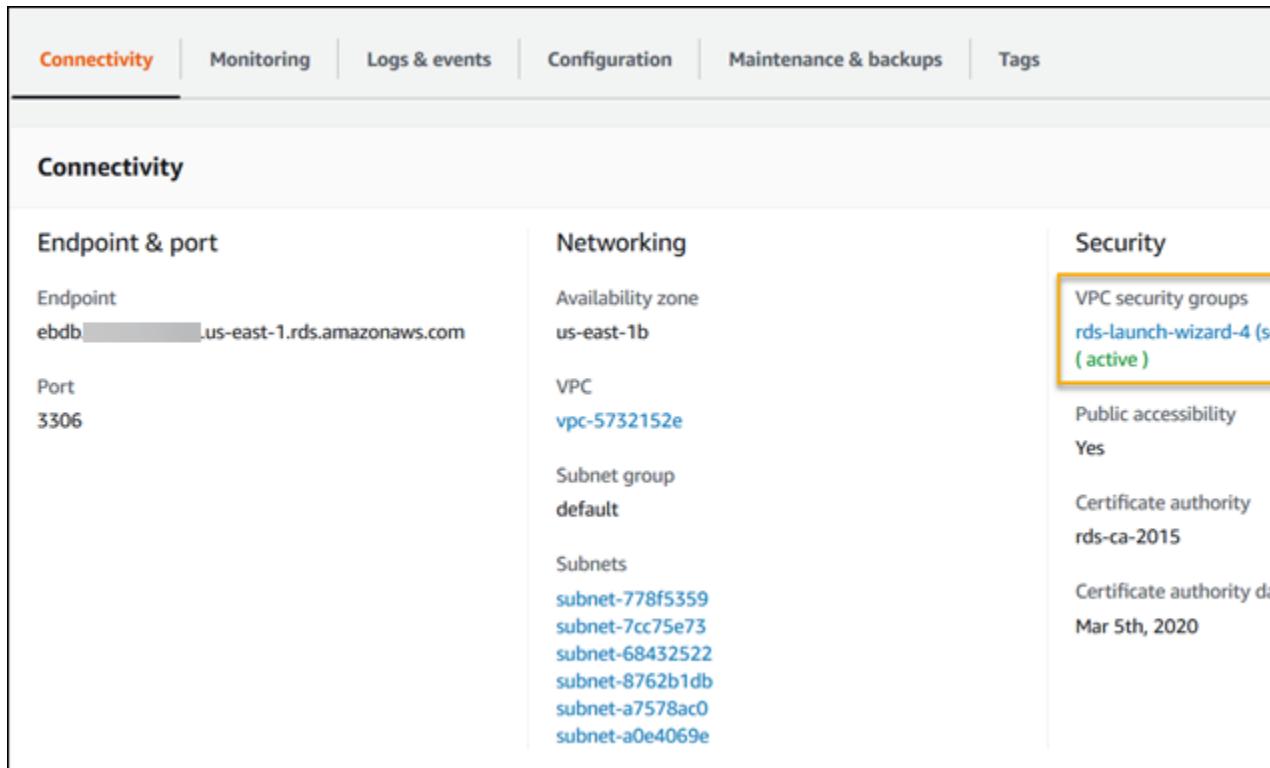
Wählen Sie nicht Easy Create (Einfache Erstellung) aus. Sie können nicht die erforderlichen Einstellungen zum Start dieser RDS-DB konfigurieren.

5. Geben Sie unter Additional configuration (Zusätzliche Konfiguration) für Initial database name (Anfangsdatenbankname) „**ebdb**“ ein.
6. Überprüfen Sie die Standardeinstellungen sorgfältig und passen Sie sie bei Bedarf an. Achten Sie auf die folgenden Optionen:
 - DB instance class (DB-Instance-Klasse) – Wählen Sie eine Instance-Größe, die eine geeignete Speichermenge und CPU-Leistung für den Workload bereitstellt.
 - Multi-AZ deployment (Multi-AZ-Bereitstellung) – Um hohe Verfügbarkeit zu gewährleisten, legen Sie dies auf Create an Aurora Replica/Reader node in a different AZ (Einen Aurora-Replikats-/Reader-Knoten in einer anderen AZ erstellen) fest.
 - Master username (Master-Benutzername) und Master password (Master-Passwort) – Der Benutzername und das Passwort für die Datenbank. Notieren Sie sich diese Einstellungen, um sie später zu verwenden.
7. Überprüfen Sie die Standardeinstellungen für die übrigen Optionen und wählen Sie dann Create database (Datenbank starten) aus.

Als Nächstes bearbeiten Sie die Sicherheitsgruppe für die DB-Instance, damit eingehender Datenverkehr über den entsprechenden Port zulässig ist. Es handelt sich um die gleiche Sicherheitsgruppe, die später mit der Elastic Beanstalk-Umgebung verknüpft wird. Die von Ihnen hinzugefügte Regel gewährt damit Eingangsberechtigungen für andere Ressourcen in derselben Sicherheitsgruppe.

So ändern Sie die Eingangsregeln für die Sicherheitsgruppe der RDS-Instance

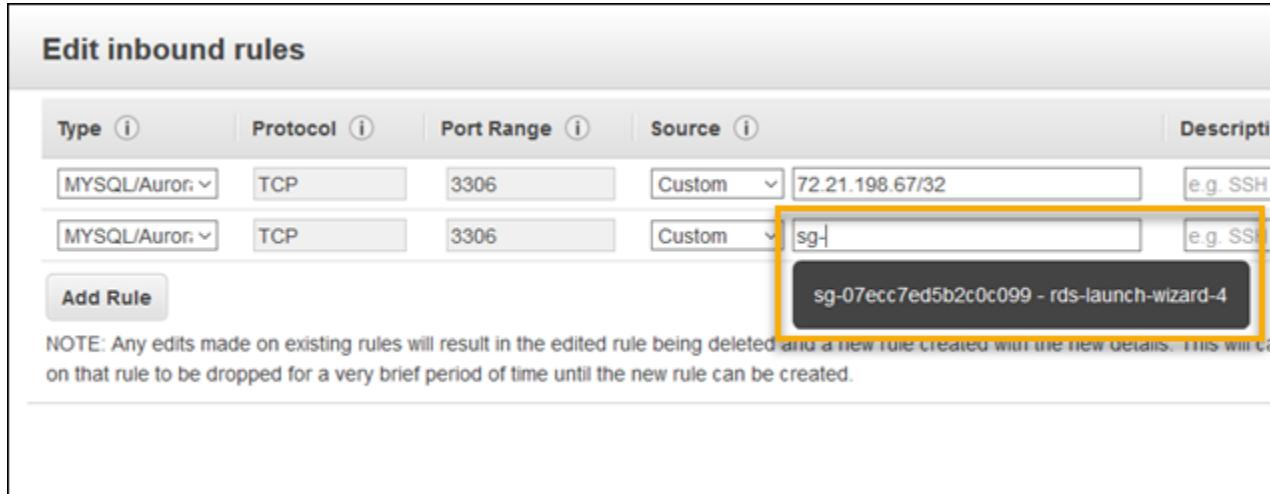
1. Öffnen Sie die [Amazon RDS-Konsole](#).
2. Wählen Sie Databases (Datenbanken) aus.
3. Wählen Sie den Namen der DB-Instance, um deren Details anzuzeigen.
4. Notieren Sie im Abschnitt Connectivity (Konnektivität) die auf dieser Seite angezeigten Subnets (Subnetze), Security groups (Sicherheitsgruppen) und den Endpoint (Endpunkt), damit Sie diese Informationen später verwenden können.
5. In Security (Sicherheit) wird die mit der DB-Instance verknüpfte Sicherheitsgruppe angezeigt. Öffnen Sie den Link, um die Sicherheitsgruppe in der Amazon EC2-Konsole aufzurufen.



The screenshot shows the AWS RDS console with the 'Connectivity' tab selected. It displays the following information:

Endpoint & port	Networking	Security
Endpoint ebdb.us-east-1.rds.amazonaws.com	Availability zone us-east-1b	VPC security groups rds-launch-wizard-4 (active)
Port 3306	VPC vpc-5732152e	Public accessibility Yes
	Subnet group default	Certificate authority rds-ca-2015
	Subnets subnet-778f5359 subnet-7cc75e73 subnet-68432522 subnet-8762b1db subnet-a7578ac0 subnet-a0e4069e	Certificate authority date Mar 5th, 2020

6. Klicken Sie in den Details der Sicherheitsgruppe auf Inbound (Eingehend).
7. Wählen Sie Edit (Bearbeiten) aus.
8. Klicken Sie auf Add Rule.
9. Wählen Sie unter Type (Typ) die von der Anwendung genutzte DB-Engine aus.
10. Geben Sie für Source (Quelle) **sg-** ein, um eine Liste der verfügbaren Sicherheitsgruppen anzuzeigen. Wählen Sie die Sicherheitsgruppe aus, die mit der Auto Scaling-Gruppe einer Elastic Beanstalk-Umgebung verknüpft ist, damit Amazon EC2-Instances in der Umgebung Zugriff auf die Datenbank haben.



11. Wählen Sie Save (Speichern) aus.

Das Erstellen einer DB-Instance dauert ca. 10 Minuten. In der Zwischenzeit erstellen Sie die Elastic Beanstalk-Umgebung.

Erstellen einer Elastic Beanstalk-Umgebung

Verwenden Sie die Elastic Beanstalk-Konsole, um eine Elastic Beanstalk-Umgebung zu erstellen. Wählen Sie die Plattform PHP und übernehmen Sie die Standardeinstellungen und den Beispielcode. Nachdem Sie die Umgebung gestartet haben, können Sie die Umgebung für die Verbindung mit der Datenbank konfigurieren. Stellen Sie dann die Beispielanwendung bereit, die Sie von GitHub heruntergeladen haben.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced.
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus, und wenn Sie bereit sind, wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2) zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-

Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.

- Load balancer (Load Balancer) – Ein Elastic Load Balancing-Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn sie beendet wird oder nicht verfügbar ist.
- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – Zwei CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form **subdomain.region.elasticbeanstalk.com** weiterleitet.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet. Die RDS-DB-Instance wurde von Ihnen außerhalb der Umgebung gestartet, daher müssen Sie die Verwaltung ihres Lebenszyklus übernehmen.

Note

Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#).

Konfigurieren von Sicherheitsgruppen, Umgebungseigenschaften und Skalierung

Fügen Sie die Sicherheitsgruppe der DB-Instance zur ausgeführten Umgebung hinzu. Dadurch stellt Elastic Beanstalk alle Instances in der Umgebung mit der zusätzlich zugeordneten Sicherheitsgruppe erneut bereit.

So fügen Sie eine Sicherheitsgruppe zur Umgebung hinzu

- Führen Sie eine der folgenden Aufgaben aus:
 - So fügen Sie eine Sicherheitsgruppe mit der Elastic Beanstalk-Konsole hinzu
 - a. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
 - b. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

- c. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
- d. Wählen Sie in der Konfigurationskategorie Instances die Option Edit (Bearbeiten).
- e. Wählen Sie unter EC2 security groups (EC2-Sicherheitsgruppen) neben der Instance-Sicherheitsgruppe, die Elastic Beanstalk erstellt, die Sicherheitsgruppe aus, die an die Instances angehängt werden soll.
- f. Wählen Sie Apply (Anwenden) aus.
- g. Lesen Sie die Warnmeldung und klicken Sie dann auf Bestätigen.
- Um eine Sicherheitsgruppe mit einer [Konfigurationsdatei \(p. 722\)](#) hinzuzufügen, verwenden Sie die Beispieldatei [securitygroup-addexisting.config](#).

Anschließend werden die Verbindungsinformationen mithilfe von Umgebungseigenschaften an die Umgebung übergeben. In der Beispielanwendung werden Standardeigenschaften verwendet, die denen entsprechen, die von Elastic Beanstalk im Rahmen der Datenbankbereitstellung in der Umgebung konfiguriert werden.

So konfigurieren Sie Umgebungseigenschaften für eine Amazon RDS-DB-Instance

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Definieren Sie im Abschnitt Environment properties (Umgebungseigenschaften) die Variablen, die Ihre Anwendung liest, um eine Verbindungszeichenfolge zu konstruieren. Für die Kompatibilität mit Umgebungen, die eine integrierte RDS-DB-Instance haben, verwenden Sie die folgenden Namen und Werte. Sie können alle Werte außer Ihrem Passwort in der [RDS-Konsole](#) finden.

Eigenschaftenname	Beschreibung	Eigenschaftswert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbankname, ebdb	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: DB Name (DB-Name).
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole:

Eigenschaftenname	Beschreibung	Eigenschaftswert
		Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	In der Amazon RDS-Konsole nicht als Referenz verfügbar.

Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	ebdb
RDS_HOSTNAME	webapp-db.jxccb5mpan
RDS_PORT	5432
RDS_USERNAME	webapp-admin
RDS_PASSWORD	kUj5uKxmWDMYc403

[Cancel](#)

6. Wählen Sie Apply (Anwenden) aus.

Konfigurieren Sie abschließend für die Auto Scaling-Gruppe der Umgebung eine höhere Instance-Mindestanzahl. Führen Sie jederzeit mindestens zwei Instances aus, damit kein Webserver in der Umgebung eine einzelne Fehlerquelle darstellt und damit Sie Änderungen ohne Ausfallzeit der Website bereitstellen können.

So konfigurieren Sie die Auto Scaling-Gruppe Ihrer Umgebung für hohe Verfügbarkeit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Legen Sie im Abschnitt Auto Scaling Group (Auto Scaling-Gruppe) die Option Min instances (Min. Instances) auf **2** fest.
6. Wählen Sie Apply (Anwenden) aus.

Bereitstellen der Beispielanwendung

Ihre Umgebung ist jetzt bereit für die Ausführung der Beispielanwendung und zum Herstellen einer Verbindung mit Amazon RDS. Stellen Sie die Beispielanwendung in Ihrer Umgebung bereit.

Note

Laden Sie das Quell-Bundle von GitHub herunter: [eb-demo-php-simple-app-1.3.zip](#)

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Die Website sammelt Benutzerkommentare und verwendet die MySQL-Datenbank zum Speichern der Daten. Zum Hinzufügen eines Kommentars wählen Sie Share Your Thought, geben Sie einen Kommentar ein und wählen Sie dann Submit Your Thought. Die Web-App schreibt den Kommentar in die Datenbank, sodass alle Instances in der Umgebung ihn lesen können und er nicht verloren geht, wenn Instances außer Betrieb sind.

Your Thoughts

 Share Your Thought

Elastic Beanstalk makes it easy to get up and running with web development or — Beanstalk User

Bereinigen

Wenn Sie nicht mehr mit Elastic Beanstalk arbeiten, können Sie Ihre Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 531\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) und dann Terminate environment (Umgebung beenden) aus.
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Außerdem können Sie Datenbankressourcen beenden, die außerhalb Ihrer Elastic Beanstalk-Umgebung erstellt wurden. Beim Beenden einer Amazon RDS-DB-Instance können Sie einen Snapshot erstellen und die Daten zu einem späteren Zeitpunkt in einer anderen Instance wiederherstellen.

So beenden Sie Ihre RDS DB-Instance

1. Öffnen Sie die [Amazon RDS-Konsole](#).
2. Wählen Sie Databases (Datenbanken) aus.

3. Wählen Sie die DB-Instance aus.
4. Wählen Sie Actions und anschließend Delete.
5. Wählen Sie, ob Sie einen Snapshot erstellen, und wählen Sie anschließend Delete (Löschen).

Nächste Schritte

Im weiteren Verlauf der Anwendungsentwicklung möchten Sie vermutlich Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die [Elastic Beanstalk Command Line Interface \(p. 1008\)](#) (EB CLI) stellt benutzerfreundliche Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk-Umgebungen über die Befehlszeile bereit.

In der Beispielanwendung werden Konfigurationsdateien verwendet, um PHP-Einstellungen zu konfigurieren und eine Tabelle in der Datenbank zu erstellen (sofern nicht bereits vorhanden). Mit einer Konfigurationsdatei können Sie zudem während der Umgebungserstellung die Sicherheitsgruppeneinstellungen der Instance konfigurieren und so zeitaufwendige Konfigurationsaktualisierungen verhindern. Weitere Informationen finden Sie unter [Erweiterte Umgebungsanpassung mit Konfigurationsdateien \(.ebextensions\) \(p. 722\)](#).

Für die Entwicklung sowie für Tests können Sie die Elastic Beanstalk-Funktionalität nutzen und so eine verwaltete DB-Instance direkt in der Umgebung hinzufügen. Detaillierte Anweisungen zum Einrichten einer Datenbank in der Umgebung finden Sie unter [Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung \(p. 609\)](#).

Wenn Sie eine Datenbank mit hoher Verfügbarkeit benötigen, können Sie [Amazon Aurora](#) verwenden. Amazon Aurora ist eine mit MySQL kompatible Datenbank-Engine, die kommerzielle Datenbankfunktionen bei geringen Kosten bietet. Um die Verbindung der Anwendung mit einer anderen Datenbank herzustellen, wiederholen Sie die Schritte der [Sicherheitsgruppenkonfiguration \(p. 314\)](#) und [aktualisieren Sie die RDS-bezogenen Umgebungseigenschaften \(p. 317\)](#).

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, [konfigurieren Sie einen benutzerdefinierten Domänennamen \(p. 644\)](#) für die Umgebung und [aktivieren Sie HTTPS \(p. 778\)](#) für sichere Verbindungen.

Bereitstellen einer WordPress-Website mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank zu Elastic Beanstalk

In diesem Tutorial wird beschrieben, wie Sie [eine Amazon RDS-DB-Instance starten \(p. 973\)](#), die für AWS Elastic Beanstalk extern ist, und anschließend eine Hochverfügbarkeitsumgebung konfigurieren, auf der eine WordPress-Website ausgeführt wird, um eine Verbindung damit herzustellen. Die Website verwendet Amazon Elastic File System (Amazon EFS) als gemeinsamen Speicher für hochgeladene Dateien.

Wenn Sie eine DB-Instance außerhalb von Elastic Beanstalk ausführen, wird die Datenbank vom Lebenszyklus Ihrer Umgebung entkoppelt. Auf diese Weise können Sie eine Verbindung mit der gleichen Datenbank aus mehreren Umgebungen herstellen, eine Datenbank gegen eine andere austauschen oder eine [Blau/Grün-Bereitstellung \(p. 481\)](#), ohne dass dies Auswirkungen auf Ihre Datenbank hat.

Dieses Tutorial wurde mit WordPress Version 4.9.5 und PHP 7.0 entwickelt.

Themen

- [Voraussetzungen \(p. 323\)](#)

- [Starten einer DB-Instance in Amazon RDS \(p. 323\)](#)
- [Herunterladen von WordPress \(p. 326\)](#)
- [Starten einer Elastic Beanstalk-Umgebung \(p. 326\)](#)
- [Konfigurieren von Sicherheitsgruppen und Umgebungseigenschaften \(p. 328\)](#)
- [Konfigurieren und Bereitstellen Ihrer Anwendung \(p. 330\)](#)
- [Installation von WordPress \(p. 332\)](#)
- [Aktualisieren von Schlüsseln und Salts \(p. 332\)](#)
- [Entfernen von Zugriffseinschränkungen \(p. 333\)](#)
- [Konfigurieren der Auto Scaling-Gruppe \(p. 333\)](#)
- [Upgraden von WordPress \(p. 334\)](#)
- [Bereinigen \(p. 334\)](#)
- [Nächste Schritte \(p. 335\)](#)

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie über Kenntnisse über die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole verfügen. Sofern noch nicht geschehen, befolgen Sie die Anweisungen in [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#), um die erste Elastic Beanstalk-Umgebung zu starten.

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine -Shell für die Ausführung der Befehle. Befehle werden in Auflistungen mit einem vorangestellten Eingabeaufforderungs-Symbol (\$) und dem Namen des aktuellen Verzeichnisses angezeigt, falls angebracht.

```
~/eb-project$ this is a command  
this is output
```

Verwenden Sie auf Linux und macOS Ihren bevorzugten Shell- und Paket-Manager. Sie können unter Windows 10 das [Windows-Subsystem für Linux](#) installieren, um eine Windows-Version von Ubuntu und Bash zu erhalten.

Standard-VPC

Die Verfahren für Amazon Relational Database Service (Amazon RDS) in diesem Tutorial gehen davon aus, dass Sie Ressourcen in einer standardmäßigen [Amazon Virtual Private Cloud](#) (Amazon VPC) starten. Alle neuen Konten umfassen eine Standard-VPC in jeder AWS-Region. Wenn Sie nicht über eine Standard-VPC verfügen, variieren die Verfahren. Weitere Anweisungen für EC2-Classic und benutzerdefinierte VPC-Plattformen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon RDS \(p. 973\)](#).

AWS-Regionen

Die Beispielanwendung verwendet Amazon EFS. Dies funktioniert nur in AWS-Regionen, die Amazon EFS unterstützen. Weitere Informationen zu unterstützten AWS-Regionen finden Sie unter [Amazon Elastic File System Endpunkte und Kontingente](#) in der Allgemeinen AWS-Referenz.

Starten einer DB-Instance in Amazon RDS

Wenn Sie eine Instance mit Amazon RDS starten, ist diese gänzlich unabhängig von Elastic Beanstalk und den Elastic Beanstalk-Umgebungen. Sie wird von Elastic Beanstalk weder beendet noch überwacht.

In den folgenden Schritten verwenden Sie die Amazon RDS-Konsole für Folgendes:

- Starten Sie eine Datenbank mit der MySQL-Engine.
- Aktivieren Sie eine Multi-AZ-Bereitstellung. Dadurch wird ein Standby-Modus in einer anderen Availability Zone (AZ) erstellt, um Datenredundanz bereitzustellen, das Einfrieren von Ein-/Ausgaben zu vermeiden und Latenzspitzen während Systemsicherungen zu minimieren.

So starten Sie eine RDS DB-Instance in einer Standard-VPC

1. Öffnen Sie die [RDS-Konsole](#).
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie Create database (Datenbank erstellen) aus.
4. Wählen Sie Standard Create (Standarderstellung).

Important

Wählen Sie nicht Easy Create (Einfache Erstellung) aus. Sie können nicht die erforderlichen Einstellungen zum Start dieser RDS-DB konfigurieren.

5. Geben Sie unter Additional configuration (Zusätzliche Konfiguration) für Initial database name (Anfangsdatenbankname) „**ebdb**“ ein.
6. Überprüfen Sie die Standardeinstellungen sorgfältig und passen Sie sie bei Bedarf an. Achten Sie auf die folgenden Optionen:
 - DB instance class (DB-Instance-Klasse) – Wählen Sie eine Instance-Größe, die eine geeignete Speichermenge und CPU-Leistung für den Workload bereitstellt.
 - Multi-AZ deployment (Multi-AZ-Bereitstellung) – Um hohe Verfügbarkeit zu gewährleisten, legen Sie dies auf Create an Aurora Replica/Reader node in a different AZ (Einen Aurora-Replikats-/Reader-Knoten in einer anderen AZ erstellen) fest.
 - Master username (Master-Benutzername) und Master password (Master-Passwort) – Der Benutzername und das Passwort für die Datenbank. Notieren Sie sich diese Einstellungen, um sie später zu verwenden.
7. Überprüfen Sie die Standardeinstellungen für die übrigen Optionen und wählen Sie dann Create database (Datenbank starten) aus.

Nachdem Ihre DB-Instance erstellt wurde, ändern Sie die ihr zugewiesene Sicherheitsgruppe, um eingehenden Datenverkehr auf dem entsprechenden Port zuzulassen.

Note

Es handelt sich um die gleiche Sicherheitsgruppe, die später der Elastic Beanstalk-Umgebung zugewiesen wird. Die von Ihnen jetzt hinzugefügte Regel gewährt damit Eingangsberechtigungen für andere Ressourcen in derselben Sicherheitsgruppe.

So ändern Sie die Eingangsregeln für die Sicherheitsgruppe der RDS-Instance

1. Öffnen Sie die [Amazon RDS-Konsole](#).
2. Wählen Sie Databases (Datenbanken) aus.
3. Wählen Sie den Namen der DB-Instance, um deren Details anzuzeigen.
4. Notieren Sie im Abschnitt Connectivity (Konnektivität) die auf dieser Seite angezeigten Subnets (Subnetze), Security groups (Sicherheitsgruppen) und den Endpoint (Endpunkt), damit Sie diese Informationen später verwenden können.
5. In Security (Sicherheit) wird die mit der DB-Instance verknüpfte Sicherheitsgruppe angezeigt. Öffnen Sie den Link, um die Sicherheitsgruppe in der Amazon EC2-Konsole aufzurufen.

Connectivity	Monitoring	Logs & events	Configuration	Maintenance & backups	Tags
Connectivity					
Endpoint & port	Networking	Security			
Endpoint ebdb[REDACTED].us-east-1.rds.amazonaws.com	Availability zone us-east-1b	VPC security groups rds-launch-wizard-4 (active)			
Port 3306	VPC vpc-5732152e	Public accessibility Yes			
	Subnet group default	Certificate authority rds-ca-2015			
	Subnets subnet-778f5359 subnet-7cc75e73 subnet-68432522 subnet-8762b1db subnet-a7578ac0 subnet-a0e4069e	Certificate authority date Mar 5th, 2020			

6. Klicken Sie in den Details der Sicherheitsgruppe auf Inbound (Eingehend).
7. Wählen Sie Edit (Bearbeiten) aus.
8. Klicken Sie auf Add Rule.
9. Wählen Sie unter Type (Typ) die von der Anwendung genutzte DB-Engine aus.
10. Geben Sie für Source (Quelle) **sg-** ein, um eine Liste der verfügbaren Sicherheitsgruppen anzuzeigen. Wählen Sie die Sicherheitsgruppe aus, die mit der Auto Scaling-Gruppe einer Elastic Beanstalk-Umgebung verknüpft ist, damit Amazon EC2-Instances in der Umgebung Zugriff auf die Datenbank haben.

Edit inbound rules

Type	Protocol	Port Range	Source	Description
MySQL/Aurora	TCP	3306	Custom	72.21.198.67/32
MySQL/Aurora	TCP	3306	Custom	sg-[REDACTED]
<input type="button" value="Add Rule"/> <small>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 to be dropped for a very brief period of time until the new rule can be created.</small>				

11. Wählen Sie Save (Speichern) aus.

Das Erstellen einer DB-Instance dauert ca. 10 Minuten. In der Zwischenzeit können Sie WordPress herunterladen und die Elastic Beanstalk-Umgebung erstellen.

Herunterladen von WordPress

Um die Bereitstellung von WordPress mit AWS Elastic Beanstalk vorzubereiten, müssen Sie die WordPress-Dateien auf Ihren Computer kopieren und die korrekten Konfigurationsinformationen angeben.

So erstellen Sie ein WordPress-Projekt

1. Laden Sie WordPress von wordpress.org herunter.

```
~$ curl https://wordpress.org/wordpress-4.9.5.tar.gz -o wordpress.tar.gz
```

2. Laden Sie die Konfigurationsdateien aus dem Beispielrepository herunter.

```
~$ wget https://github.com/aws-samples/eb-php-wordpress/releases/download/v1.1/eb-php-wordpress-v1.zip
```

3. Extrahieren Sie WordPress und ändern Sie den Namen des Ordners.

```
~$ tar -xvf wordpress.tar.gz
~$ mv wordpress wordpress-beanstalk
~$ cd wordpress-beanstalk
```

4. Extrahieren Sie die Konfigurationsdateien über die WordPress-Installation.

```
~/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
```

Starten einer Elastic Beanstalk-Umgebung

Verwenden Sie die Elastic Beanstalk-Konsole, um eine Elastic Beanstalk-Umgebung zu erstellen. Nachdem Sie die Umgebung gestartet haben, können Sie sie für die Verbindung zur Datenbank konfigurieren und danach den WordPress-Code in der Umgebung bereitstellen.

In den folgenden Schritten verwenden Sie die Elastic Beanstalk-Konsole für Folgendes:

- Erstellen Sie eine Elastic Beanstalk-Anwendung mit der verwalteten PHP-Plattform.
- Übernehmen Sie die Standardeinstellungen und den Beispiel-Code.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced.
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.

3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus, und wenn Sie bereit sind, wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt.

Elastic Beanstalk hat Ressourcen erstellt

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2) zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer) – Ein Elastic Load Balancing-Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn sie beendet wird oder nicht verfügbar ist.
- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – Zwei CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form **subdomain.region.elasticbeanstalk.com** weiterleitet.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Da die Amazon RDS-Instance von Ihnen außerhalb der Umgebung gestartet wurde, müssen Sie die Verwaltung ihres Lebenszyklus übernehmen.

Note

Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#).

Konfigurieren von Sicherheitsgruppen und Umgebungseigenschaften

Fügen Sie die Sicherheitsgruppe der DB-Instance zur ausführten Umgebung hinzu. Dadurch stellt Elastic Beanstalk alle Instances in der Umgebung mit der zusätzlich zugeordneten Sicherheitsgruppe erneut bereit.

So fügen Sie eine Sicherheitsgruppe zur Umgebung hinzu

- Führen Sie eine der folgenden Aufgaben aus:
 - So fügen Sie eine Sicherheitsgruppe mit der Elastic Beanstalk-Konsole hinzu
 - a. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
 - b. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

- c. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
 - d. Wählen Sie in der Konfigurationskategorie Instances die Option Edit (Bearbeiten).
 - e. Wählen Sie unter EC2 security groups (EC2-Sicherheitsgruppen) neben der Instance-Sicherheitsgruppe, die Elastic Beanstalk erstellt, die Sicherheitsgruppe aus, die an die Instances angehängt werden soll.
 - f. Wählen Sie Apply (Anwenden) aus.
 - g. Lesen Sie die Warnmeldung und klicken Sie dann auf Bestätigen.
- Um eine Sicherheitsgruppe mit einer [Konfigurationsdatei \(p. 722\)](#) hinzuzufügen, verwenden Sie die Beispieldatei [securitygroup-addexisting.config](#).

Anschließend werden die Verbindungsinformationen mithilfe von Umgebungseigenschaften an die Umgebung übergeben.

In der WordPress-Anwendung werden Standardeigenschaften verwendet, die denen entsprechen, die von Elastic Beanstalk im Rahmen der Datenbankbereitstellung in der Umgebung konfiguriert werden.

So konfigurieren Sie Umgebungseigenschaften für eine Amazon RDS-DB-Instance

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Definieren Sie im Abschnitt Environment properties (Umgebungseigenschaften) die Variablen, die Ihre Anwendung liest, um eine Verbindungszeichenfolge zu konstruieren. Für die Kompatibilität mit Umgebungen, die eine integrierte RDS-DB-Instance haben, verwenden Sie die folgenden Namen und Werte. Sie können alle Werte außer Ihrem Passwort in der [RDS-Konsole](#) finden.

Eigenschaftenname	Beschreibung	Eigenschaftenwert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbankname, ebdb	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: DB Name (DB-Name).
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	In der Amazon RDS-Konsole nicht als Referenz verfügbar.

Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	ebdb
RDS_HOSTNAME	webapp-db.jxccb5mpan
RDS_PORT	5432
RDS_USERNAME	webapp-admin
RDS_PASSWORD	kUj5uKxmWDMYc403

[Cancel](#)

6. Wählen Sie Apply (Anwenden) aus.

Konfigurieren und Bereitstellen Ihrer Anwendung

Überprüfen Sie wie angezeigt, ob die Struktur des Ordners `wordpress-beanstalk` korrekt ist.

```
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
```

Die angepasste Datei `wp-config.php` aus dem Projektrepository verwendet die Umgebungsvariablen, die Sie im vorherigen Schritt zum Konfigurieren der Datenbankverbindung definiert haben. Der Ordner `.ebextensions` enthält Konfigurationsdateien, die weitere Ressourcen in Ihrer Elastic Beanstalk-Umgebung erstellen.

Die Konfigurationsdateien müssen für die Arbeit mit dem Konto geändert werden. Ersetzen Sie die Platzhalterwerte in den Dateien durch die entsprechenden IDs und erstellen Sie ein Quell-Bundle.

So aktualisieren Sie die Konfigurationsdateien und erstellen ein Quell-Bundle

1. Ändern Sie die Konfigurationsdateien wie folgt.

- `.ebextensions/dev.config` – Beschränkt den Zugriff auf die Umgebung, um sie während der WordPress-Installation zu schützen. Ersetzen Sie die Platzhalter-IP-Adresse oben in der Datei durch die öffentliche IP-Adresse des Computers, mit dem Sie auf die Website Ihrer Umgebung zugreifen, um Ihre WordPress-Installation abzuschließen.

Note

Abhängig von Ihrem Netzwerk benötigen Sie möglicherweise eine IP-Adressblock.

- `.ebextensions/efs-create.config` – Erstellt ein EFS-Dateisystem und Mountingpunkte in allen Availability Zones/Subnetzen der VPC. Ermitteln Sie die IDs von Standard-VPC und Subnetz in der [Amazon VPC-Konsole](#).

2. Erstellen Sie ein [Quell-Bundle \(p. 412\)](#) mit den Dateien im Projektordner. Der folgende Befehl erstellt ein Quell-Bundle namens `wordpress-beanstalk.zip`.

```
~/eb-wordpress$ zip .../wordpress-beanstalk.zip -r *.[^.]*
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um WordPress in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Installation von WordPress

So schließen Sie die WordPress-Installation ab

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie die Umgebungs-URL aus, um die Website in einem Browser zu öffnen. Sie werden zu einem WordPress-Installationsassistenten umgeleitet, da Sie die Website noch nicht konfiguriert haben.
4. Führen Sie eine Standardinstallation aus. Die Datei `wp-config.php` ist bereits im Quellcode vorhanden und so konfiguriert, dass sie die Informationen für die Datenbankverbindung aus der Umgebung ausliest. Sie sollten nicht zur Verbindungskonfiguration aufgefordert werden.

Die Installation dauert etwa eine Minute.

Aktualisieren von Schlüsseln und Salts

Von der WordPress-Konfigurationsdatei `wp-config.php` werden auch Werte für Schlüssel und Salts aus den Umgebungseigenschaften ausgelesen. Zurzeit werden diese Eigenschaften alle von der Datei `test` im Ordner `wordpress.config` auf den Wert `.ebextensions` festgelegt.

Der Hash-Salt kann jeder Wert sein, der die [Anforderungen an Umgebungseigenschaften \(p. 623\)](#) erfüllt, Sie sollten ihn jedoch nicht in der Versionskontrolle speichern. Legen Sie diese Eigenschaften mithilfe der Elastic Beanstalk-Konsole direkt in der Umgebung fest.

So aktualisieren Sie die Umgebungseigenschaften

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie unter Software die Option Edit (Bearbeiten).
5. Ändern Sie für `Environment properties` die folgenden Eigenschaften:
 - `AUTH_KEY` – Der gewählte Wert für `AUTH_KEY`.
 - `SECURE_AUTH_KEY` – Der gewählte Wert für `SECURE_AUTH_KEY`.
 - `LOGGED_IN_KEY` – Der gewählte Wert für `LOGGED_IN_KEY`.
 - `NONCE_KEY` – Der gewählte Wert für `NONCE_KEY`.
 - `AUTH_SALT` – Der gewählte Wert für `AUTH_SALT`.
 - `SECURE_AUTH_SALT` – Der gewählte Wert für `SECURE_AUTH_SALT`.
 - `LOGGED_IN_SALT` – Der gewählte Wert für `LOGGED_IN_SALT`.
 - `NONCE_SALT` – Der gewählte Wert für `NONCE_SALT`.

6. Wählen Sie Apply (Anwenden) aus.

Note

Wenn Sie die Eigenschaften direkt in der Umgebung festlegen, werden die Werte in `wordpress.config` überschrieben.

Entfernen von Zugriffseinschränkungen

Das Beispielprojekt enthält die Konfigurationsdatei „`loadbalancer-sg.config`“. Es erstellt eine Sicherheitsgruppe und weist sie dem Load Balancer der Umgebung unter Verwendung der IP-Adresse zu, die Sie in `dev.config` konfiguriert haben. Sie schränkt den HTTP-Zugriff auf Port 80 auf Verbindungen von Ihrem Netzwerk ein. Andernfalls könnte eine externe Partei potenziell eine Verbindung zur Website herstellen, bevor Sie WordPress installiert und das Administratorkonto konfiguriert haben.

Nachdem Sie WordPress nun installiert haben, entfernen Sie die Konfigurationsdatei, um die Site für die Welt verfügbar zu machen.

So entfernen Sie die Beschränkung und aktualisieren die Umgebung

1. Löschen Sie die Datei `.ebextensions/loadbalancer-sg.config` aus dem Projektverzeichnis.

```
~/wordpress-beanstalk$ rm .ebextensions/loadbalancer-sg.config
```

2. Erstellen Sie ein Quell-Bundle.

```
~/eb-wordpress$ zip ../wordpress-beanstalk-v2.zip -r *.[^.]*
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um WordPress in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Konfigurieren der Auto Scaling-Gruppe

Konfigurieren Sie abschließend für die Auto Scaling-Gruppe der Umgebung eine höhere Instance-Mindestanzahl. Führen Sie stets mindestens zwei Instances aus, um zu verhindern, dass die Webserver in Ihrer Umgebung eine einzelne Fehlerquelle darstellen. Damit können Sie Änderungen bereitstellen, ohne Ihre Website außer Betrieb zu nehmen.

So konfigurieren Sie die Auto Scaling-Gruppe Ihrer Umgebung für hohe Verfügbarkeit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Legen Sie im Abschnitt Auto Scaling Group (Auto Scaling-Gruppe) die Option Min instances (Min. Instances) auf **2** fest.
6. Wählen Sie Apply (Anwenden) aus.

Das Beispielprojekt erstellt mit Amazon EFS ein freigegebenes Dateisystem, um das Hochladen von Inhalten über mehrere Instances zu unterstützen. Erstellen Sie einen Beitrag auf der Site und laden Sie Inhalt hoch, der im freigegebenen Dateisystem gespeichert werden soll. Betrachten Sie den Beitrag und aktualisieren Sie die Seite mehrmals, damit Instances genutzt werden und zugleich verifiziert wird, dass das freigegebene Dateisystem funktioniert.

Upgraden von WordPress

Um das Upgrade auf eine neue Version von WordPress vorzunehmen, sichern Sie die Site und stellen Sie sie in einer neuen Umgebung bereit.

Important

Verwenden Sie nicht die Aktualisierungsfunktionalität von WordPress und aktualisieren Sie nicht die Quelldateien, damit sie eine neue Version verwenden. Beide Aktionen können dazu führen, dass die Beitrags-URLs den Fehler 404 zurückgeben, obwohl sie sich noch in der Datenbank und im Dateisystem befinden.

So nehmen Sie ein Upgraden für WordPress vor

1. Verwenden Sie das Export-Tool in der WordPress-Administratorkonsole, um Ihre Posts in eine XML-Datei zu exportieren.
2. Stellen Sie die neue Version von WordPress auf Elastic Beanstalk bereit und installieren Sie sie mit denselben Schritten, die Sie auch zum Installieren der vorherigen Version verwendet haben. Um Ausfallzeiten zu vermeiden, können Sie eine Umgebung mit der neuen Version erstellen.
3. Installieren Sie unter der neuen Version das WordPress-Import-Tool in der Administratorkonsole und importieren Sie mit dem Tool die XML-Datei mit Ihren Beiträgen. Wenn die Beiträge vom Benutzer "admin" mit der alten Version erstellt wurden, weisen Sie sie dem Benutzer "admin" der neuen Site zu, statt den Benutzer "admin" zu importieren.
4. Wenn Sie die neue Version in einer separaten Umgebung bereitgestellt haben, führen Sie einen [CNAME-Austausch \(p. 481\)](#) durch, um Benutzer von der alten zur neuen Site umzuleiten.

Bereinigen

Wenn Sie nicht mehr mit Elastic Beanstalk arbeiten, können Sie Ihre Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 532\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) und dann Terminate environment (Umgebung beenden) aus.
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Außerdem können Sie Datenbankressourcen beenden, die außerhalb Ihrer Elastic Beanstalk-Umgebung erstellt wurden. Beim Beenden einer Amazon RDS-DB-Instance können Sie einen Snapshot erstellen und die Daten zu einem späteren Zeitpunkt in einer anderen Instance wiederherstellen.

So beenden Sie Ihre RDS DB-Instance

1. Öffnen Sie die [Amazon RDS-Konsole](#).
2. Wählen Sie Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus.
4. Wählen Sie Actions und anschließend Delete.
5. Wählen Sie, ob Sie einen Snapshot erstellen, und wählen Sie anschließend Delete (Löschen).

Nächste Schritte

Im weiteren Verlauf der Anwendungsentwicklung möchten Sie vermutlich Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die [Elastic Beanstalk Command Line Interface \(p. 1008\)](#) (EB CLI) stellt benutzerfreundliche Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk-Umgebungen über die Befehlszeile bereit.

In der Beispieldatenbank werden Konfigurationsdateien verwendet, um PHP-Einstellungen zu konfigurieren und eine Tabelle in der Datenbank zu erstellen (sofern nicht bereits vorhanden). Mit einer Konfigurationsdatei können Sie zudem während der Umgebungserstellung die Sicherheitsgruppeneinstellungen der Instance konfigurieren und so zeitaufwendige Konfigurationsaktualisierungen verhindern. Weitere Informationen finden Sie unter [Erweiterte Umgebungsanpassung mit Konfigurationsdateien \(.ebextensions\) \(p. 722\)](#).

Für die Entwicklung sowie für Tests können Sie die Elastic Beanstalk-Funktionalität nutzen und so eine verwaltete DB-Instance direkt in der Umgebung hinzufügen. Detaillierte Anweisungen zum Einrichten einer Datenbank in der Umgebung finden Sie unter [Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung \(p. 609\)](#).

Wenn Sie eine Datenbank mit hoher Verfügbarkeit benötigen, können Sie [Amazon Aurora](#) verwenden. Amazon Aurora ist eine mit MySQL kompatible Datenbank-Engine, die kommerzielle Datenbankfunktionen bei geringen Kosten bietet. Um die Verbindung der Anwendung mit einer anderen Datenbank herzustellen, wiederholen Sie die Schritte der [Sicherheitsgruppenkonfiguration \(p. 314\)](#) und aktualisieren Sie die RDS-bezogenen Umgebungseigenschaften (p. 317).

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, [konfigurieren Sie einen benutzerdefinierten Domänennamen \(p. 644\)](#) für die Umgebung und [aktivieren Sie HTTPS \(p. 778\)](#) für sichere Verbindungen.

Bereitstellen einer Drupal-Website mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank zu Elastic Beanstalk

In diesem Tutorial erfahren Sie, wie Sie außerhalb von AWS Elastic Beanstalk eine [RDS-DB-Instance starten \(p. 973\)](#). Anschließend wird beschrieben, wie Sie eine hochverfügbare Umgebung konfigurieren, in der eine Drupal-Website ausgeführt wird, um eine Verbindung zu ihr herzustellen. Die Website verwendet Amazon Elastic File System (Amazon EFS) als gemeinsamen Speicher für hochgeladene Dateien. Wenn eine DB-Instance außerhalb von Elastic Beanstalk ausgeführt wird, ist die Datenbank vom Lebenszyklus der Umgebung unabhängig. Das heißt, Sie können von mehreren Umgebungen aus auf die Datenbank zugreifen, eine Datenbank mit einer anderen austauschen oder eine Blau/Grün-Bereitstellung ausführen, ohne dass die Datenbank dadurch beeinträchtigt wird.

Abschnitte

- [Voraussetzungen \(p. 336\)](#)
- [Starten einer DB-Instance in Amazon RDS \(p. 337\)](#)
- [Starten einer Elastic Beanstalk-Umgebung \(p. 339\)](#)
- [Konfigurieren von Sicherheitseinstellungen und Umgebungseigenschaften \(p. 340\)](#)
- [Konfigurieren und Bereitstellen Ihrer Anwendung \(p. 343\)](#)
- [Installation von Drupal \(p. 345\)](#)
- [Aktualisieren der Drupal-Konfiguration und Entfernen von Zugriffsbeschränkungen \(p. 345\)](#)
- [Konfigurieren der Auto Scaling-Gruppe \(p. 348\)](#)
- [Bereinigen \(p. 348\)](#)
- [Nächste Schritte \(p. 349\)](#)

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie über Kenntnisse über die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole verfügen. Sofern noch nicht geschehen, befolgen Sie die Anweisungen in [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#), um die erste Elastic Beanstalk-Umgebung zu starten.

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine -Shell für die Ausführung der Befehle. Befehle werden in Auflistungen mit einem vorangestellten Eingabeaufforderungs-Symbol (\$) und dem Namen des aktuellen Verzeichnisses angezeigt, falls angebracht.

```
~/eb-project$ this is a command
this is output
```

Verwenden Sie auf Linux und macOS Ihren bevorzugten Shell- und Paket-Manager. Sie können unter Windows 10 das [Windows-Subsystem für Linux](#) installieren, um eine Windows-Version von Ubuntu und Bash zu erhalten.

Die Verfahren in diesem Tutorial für Amazon Relational Database Service (Amazon RDS)-Aufgaben gehen davon aus, dass Sie Ressourcen in einer standardmäßigen [Amazon Virtual Private Cloud](#) (Amazon VPC) starten. Alle neuen Konten umfassen eine Standard-VPC in jeder Region. Wenn Sie nicht über

eine Standard-VPC verfügen, variieren die Verfahren. Weitere Anweisungen für EC2-Classic und benutzerdefinierte VPC-Plattformen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon RDS \(p. 973\)](#).

Die Beispielanwendung verwendet Amazon EFS. Es funktioniert nur in AWS-Regionen, die Amazon EFS unterstützen. Weitere Informationen zu unterstützten AWS-Regionen finden Sie unter [Amazon Elastic File System Endpunkte und Kontingente](#) in der allgemeinen AWS-Referenz.

Dieses Tutorial wurde mit Drupal Version 8.5.3 und PHP 7.0 entwickelt.

Starten einer DB-Instance in Amazon RDS

Um eine externe Datenbank mit einer Anwendung zu verwenden, die in Elastic Beanstalk ausgeführt wird, starten Sie zuerst eine DB-Instance mit Amazon RDS. Wenn Sie eine Instance mit Amazon RDS starten, ist diese gänzlich unabhängig von Elastic Beanstalk und den Elastic Beanstalk-Umgebungen. Sie wird von Elastic Beanstalk weder beendet noch überwacht.

Verwenden Sie die Amazon RDS-Konsole, um eine DB-Instance mit MySQL und Multi-AZ-Bereitstellung zu starten. Eine Multi-AZ-Bereitstellung gewährleistet die Ausführung von Failovern für Ihre Datenbank, sodass sie auch bei einem Ausfall der DB-Quell-Instance verfügbar bleibt.

So starten Sie eine RDS DB-Instance in einer Standard-VPC

1. Öffnen Sie die [RDS-Konsole](#).
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie Create database (Datenbank erstellen) aus.
4. Wählen Sie Standard Create (Standarderstellung).

Important

Wählen Sie nicht Easy Create (Einfache Erstellung) aus. Sie können nicht die erforderlichen Einstellungen zum Start dieser RDS-DB konfigurieren.

5. Geben Sie unter Additional configuration (Zusätzliche Konfiguration) für Initial database name (Anfangsdatenbankname) „**ebdb**“ ein.
6. Überprüfen Sie die Standardeinstellungen sorgfältig und passen Sie sie bei Bedarf an. Achten Sie auf die folgenden Optionen:
 - DB instance class (DB-Instance-Klasse) – Wählen Sie eine Instance-Größe, die eine geeignete Speichermenge und CPU-Leistung für den Workload bereitstellt.
 - Multi-AZ deployment (Multi-AZ-Bereitstellung) – Um hohe Verfügbarkeit zu gewährleisten, legen Sie dies auf Create an Aurora Replica/Reader node in a different AZ (Einen Aurora-Replikats-/Reader-Knoten in einer anderen AZ erstellen) fest.
 - Master username (Master-Benutzername) und Master password (Master-Passwort) – Der Benutzername und das Passwort für die Datenbank. Notieren Sie sich diese Einstellungen, um sie später zu verwenden.
7. Überprüfen Sie die Standardeinstellungen für die übrigen Optionen und wählen Sie dann Create database (Datenbank starten) aus.

Als Nächstes bearbeiten Sie die Sicherheitsgruppe für die DB-Instance, damit eingehender Datenverkehr über den entsprechenden Port zulässig ist. Es handelt sich um die gleiche Sicherheitsgruppe, die später mit der Elastic Beanstalk-Umgebung verknüpft wird. Die von Ihnen hinzugefügte Regel gewährt damit Eingangsberechtigungen für andere Ressourcen in derselben Sicherheitsgruppe.

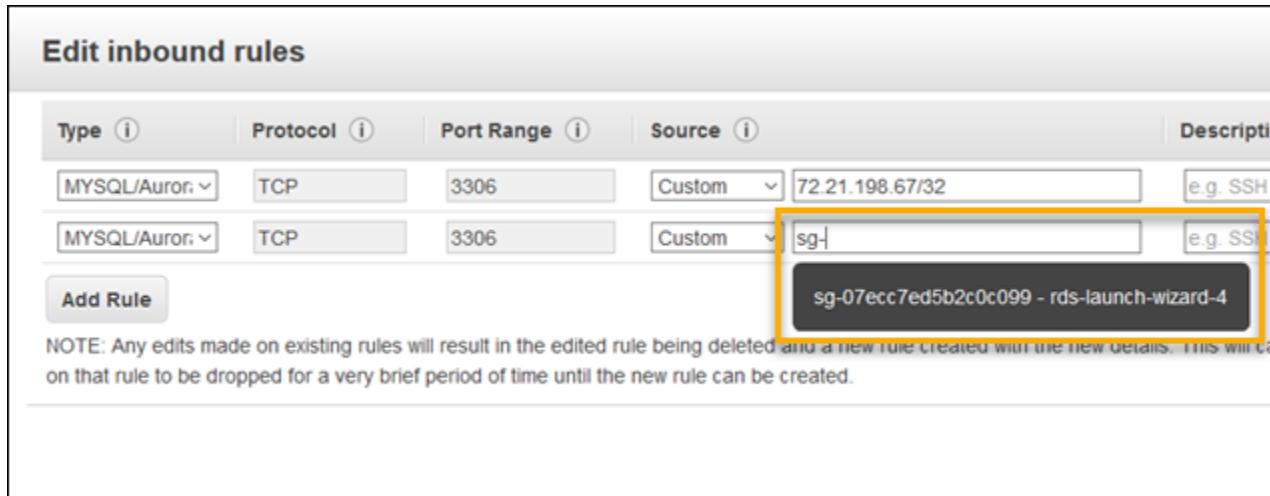
So ändern Sie die Eingangsregeln für die Sicherheitsgruppe der RDS-Instance

1. Öffnen Sie die [Amazon RDS-Konsole](#).

2. Wählen Sie Databases (Datenbanken) aus.
3. Wählen Sie den Namen der DB-Instance, um deren Details anzuzeigen.
4. Notieren Sie im Abschnitt Connectivity (Konnektivität) die auf dieser Seite angezeigten Subnets (Subnetze), Security groups (Sicherheitsgruppen) und den Endpoint (Endpunkt), damit Sie diese Informationen später verwenden können.
5. In Security (Sicherheit) wird die mit der DB-Instance verknüpfte Sicherheitsgruppe angezeigt. Öffnen Sie den Link, um die Sicherheitsgruppe in der Amazon EC2-Konsole aufzurufen.

Connectivity	Monitoring	Logs & events	Configuration	Maintenance & backups	Tags
Connectivity					
Endpoint & port					
Endpoint			Networking		Security
ebdb...us-east-1.rds.amazonaws.com			Availability zone	VPC security groups	
Port			us-east-1b	rds-launch-wizard-4 (s)	
3306			VPC	(active)	
			vpc-5732152e		
			Subnet group	Public accessibility	
			default	Yes	
			Subnets	Certificate authority	
			subnet-778f5359	rds-ca-2015	
			subnet-7cc75e73		
			subnet-68432522	Certificate authority d	
			subnet-8762b1db	Mar 5th, 2020	
			subnet-a7578ac0		
			subnet-a0e4069e		

6. Klicken Sie in den Details der Sicherheitsgruppe auf Inbound (Eingehend).
7. Wählen Sie Edit (Bearbeiten) aus.
8. Klicken Sie auf Add Rule.
9. Wählen Sie unter Type (Typ) die von der Anwendung genutzte DB-Engine aus.
10. Geben Sie für Source (Quelle) **sg-** ein, um eine Liste der verfügbaren Sicherheitsgruppen anzuzeigen. Wählen Sie die Sicherheitsgruppe aus, die mit der Auto Scaling-Gruppe einer Elastic Beanstalk-Umgebung verknüpft ist, damit Amazon EC2-Instances in der Umgebung Zugriff auf die Datenbank haben.



11. Wählen Sie Save (Speichern) aus.

Das Erstellen einer DB-Instance dauert ca. 10 Minuten. In der Zwischenzeit starten Sie die Elastic Beanstalk-Umgebung.

Starten einer Elastic Beanstalk-Umgebung

Verwenden Sie die Elastic Beanstalk-Konsole, um eine Elastic Beanstalk-Umgebung zu erstellen. Wählen Sie die Plattform PHP und übernehmen Sie die Standardeinstellungen und den Beispielcode. Nachdem Sie die Umgebung gestartet haben, können Sie die Umgebung für die Verbindung zur Datenbank konfigurieren und danach den Drupal-Code in der Umgebung bereitstellen.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced.
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus, und wenn Sie bereit sind, wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2) zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-

Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.

- Load balancer (Load Balancer) – Ein Elastic Load Balancing-Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn sie beendet wird oder nicht verfügbar ist.
- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – Zwei CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form **subdomain.region.elasticbeanstalk.com** weiterleitet.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet. Die RDS-DB-Instance wurde von Ihnen außerhalb der Umgebung gestartet, daher müssen Sie die Verwaltung ihres Lebenszyklus übernehmen.

Note

Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#).

Konfigurieren von Sicherheitseinstellungen und Umgebungseigenschaften

Fügen Sie die Sicherheitsgruppe der DB-Instance zur ausgeführten Umgebung hinzu. Dadurch stellt Elastic Beanstalk alle Instances in der Umgebung mit der zusätzlich zugeordneten Sicherheitsgruppe erneut bereit.

So fügen Sie eine Sicherheitsgruppe zur Umgebung hinzu

- Führen Sie eine der folgenden Aufgaben aus:
 - So fügen Sie eine Sicherheitsgruppe mit der Elastic Beanstalk-Konsole hinzu
 - a. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
 - b. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

- c. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
- d. Wählen Sie in der Konfigurationskategorie Instances die Option Edit (Bearbeiten).
- e. Wählen Sie unter EC2 security groups (EC2-Sicherheitsgruppen) neben der Instance-Sicherheitsgruppe, die Elastic Beanstalk erstellt, die Sicherheitsgruppe aus, die an die Instances angehängt werden soll.
- f. Wählen Sie Apply (Anwenden) aus.
- g. Lesen Sie die Warnmeldung und klicken Sie dann auf Bestätigen.
- Um eine Sicherheitsgruppe mit einer [Konfigurationsdatei \(p. 722\)](#) hinzuzufügen, verwenden Sie die Beispieldatei [securitygroup-addexisting.config](#).

Anschließend werden die Verbindungsinformationen mithilfe von Umgebungseigenschaften an die Umgebung übergeben. In der Beispielanwendung werden Standardeigenschaften verwendet, die denen entsprechen, die von Elastic Beanstalk im Rahmen der Datenbankbereitstellung in der Umgebung konfiguriert werden.

So konfigurieren Sie Umgebungseigenschaften für eine Amazon RDS-DB-Instance

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Definieren Sie im Abschnitt Environment properties (Umgebungseigenschaften) die Variablen, die Ihre Anwendung liest, um eine Verbindungszeichenfolge zu konstruieren. Für die Kompatibilität mit Umgebungen, die eine integrierte RDS-DB-Instance haben, verwenden Sie die folgenden Namen und Werte. Sie können alle Werte außer Ihrem Passwort in der [RDS-Konsole](#) finden.

Eigenschaftenname	Beschreibung	Eigenschaftswert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbankname, ebdb	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: DB Name (DB-Name).
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole:

Eigenschaftenname	Beschreibung	Eigenschaftswert
		Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	In der Amazon RDS-Konsole nicht als Referenz verfügbar.

Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	ebdb
RDS_HOSTNAME	webapp-db.jxccb5mpan
RDS_PORT	5432
RDS_USERNAME	webapp-admin
RDS_PASSWORD	kUj5uKxmWDMYc403

[Cancel](#)

6. Wählen Sie Apply (Anwenden) aus.

Nach dem Installieren von Drupal müssen Sie über SSH eine Verbindung zur Instance herstellen, um einige Konfigurationsdaten abzurufen. Weisen Sie den Instances der Umgebung einen SSH-Schlüssel zu.

So konfigurieren Sie SSH

1. Wenn Sie nicht bereits ein Schlüsselpaar erstellt haben, öffnen Sie die [Schlüsselpaar-Seite](#) der Amazon EC2-Konsole und befolgen Sie die Anleitungen zum Erstellen eines Schlüsselpaares.
2. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
3. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

4. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
5. Wählen Sie unter Security (Sicherheit) die Option Edit (Bearbeiten).
6. Wählen Sie unter EC2 key pair (EC2-Schlüsselpaar) das betreffende Schlüsselpaar aus.
7. Wählen Sie Apply (Anwenden) aus.

Konfigurieren und Bereitstellen Ihrer Anwendung

Zum Erstellen eines Drupal-Projekts für Elastic Beanstalk laden Sie den Drupal-Quellcode herunter und kombinieren ihn mit den Dateien im Repository [aws-samples/eb-php-drupal](https://github.com/aws-samples/eb-php-drupal) auf GitHub.

So erstellen Sie ein Drupal-Projekt

1. Laden Sie Drupal von <https://www.drupal.org/download> herunter.

```
~$ curl https://ftp.drupal.org/files/projects/drupal-8.5.3.tar.gz -o drupal.tar.gz
```

2. Laden Sie die Konfigurationsdateien aus dem Beispielrepository herunter:

```
~$ wget https://github.com/aws-samples/eb-php-drupal/releases/download/v1.1/eb-php-drupal-v1.zip
```

3. Extrahieren Sie Drupal und ändern Sie den Namen des Ordners.

```
~$ tar -xvf drupal.tar.gz
~$ mv drupal-8.5.3 drupal-beanstalk
~$ cd drupal-beanstalk
```

4. Extrahieren Sie die Konfigurationsdateien über die Drupal-Installation.

```
~/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
```

Überprüfen Sie wie angezeigt, ob die Struktur des Ordners drupal-beanstalk korrekt ist.

```
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
```

Die Datei `beanstalk-settings.php` aus dem Projektrepository verwendet die Umgebungsvariablen, die Sie im vorherigen Schritt zum Konfigurieren der Datenbankverbindung definiert haben. Der Ordner `.ebextensions` enthält Konfigurationsdateien, die weitere Ressourcen in Ihrer Elastic Beanstalk-Umgebung erstellen.

Die Konfigurationsdateien müssen für die Arbeit mit dem Konto geändert werden. Ersetzen Sie die Platzhalterwerte in den Dateien durch die entsprechenden IDs und erstellen Sie ein Quell-Bundle.

Zum Aktualisieren der Konfigurationsdateien und Erstellen eines Quell-Bundles.

1. Ändern Sie die Konfigurationsdateien wie folgt.
 - `.ebextensions/dev.config` – Beschränkt den Zugriff auf die Umgebung auf Ihre IP-Adresse, um die Umgebung während der Drupal-Installation zu schützen. Geben Sie anstelle des Platzhalters für die IP-Adresse oben in der Datei Ihre öffentliche IP-Adresse ein.
 - `.ebextensions/efs-create.config` – Erstellt ein EFS-Dateisystem und Mountingpunkte in allen Availability Zones/Subnetzen der VPC. Ermitteln Sie die IDs von Standard-VPC und Subnetz in der [Amazon VPC-Konsole](#).
2. Erstellen Sie ein [Quell-Bundle \(p. 412\)](#) mit den Dateien im Projektordner. Der folgende Befehl erstellt ein Quell-Bundle namens `drupal-beanstalk.zip`. Die Dateien im Ordner `vendor` werden ausgeschlossen, weil sie viel Platz benötigen und zum Bereitstellen der Anwendung für Elastic Beanstalk nicht erforderlich sind.

```
~/eb-drupal$ zip ../drupal-beanstalk.zip -r *.[^.]* -x "vendor/*"
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um Drupal in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Installation von Drupal

So schließen Sie die Drupal-Installation ab

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie die Umgebungs-URL aus, um die Website in einem Browser zu öffnen. Sie werden zu einem Drupal-Installationsassistenten umgeleitet, da die Website noch nicht konfiguriert ist.
4. Führen Sie eine Standardinstallation mit den folgenden Einstellungen für die Datenbank durch:
 - Datenbankname – Der DB Name (DB-Name), der in der Amazon RDS-Konsole angezeigt wird.
 - Benutzername und Passwort der Datenbank – Die Werte für Master Username (Master-Benutzername) und Master Password (Master-Passwort), die Sie beim Erstellen Ihrer Datenbank eingegeben haben.
 - Erweiterte Optionen > Host – Der Endpoint (Endpunkt) der DB-Instance, die in der Amazon RDS-Konsole gezeigt wird.

Die Installation dauert etwa eine Minute.

Aktualisieren der Drupal-Konfiguration und Entfernen von Zugriffsbeschränkungen

Im Rahmen der Drupal-Installation wurde eine Datei namens `settings.php` im Ordner `sites/default` in der Instance erstellt. Sie benötigen diese Datei im Quellcode, damit Ihre Site bei späteren Bereitstellungen nicht zurückgesetzt wird. Derzeit enthält die Datei aber Geheimnisse und sollte deshalb nicht in die Quelle geschrieben werden. Stellen Sie eine Verbindung zur Anwendungs-Instance her, um Daten aus der Einstellungsdatei abzurufen.

So verbinden Sie die Anwendungs-Instance mit SSH

1. Öffnen Sie die [Instances-Seite](#) der Amazon EC2-Konsole.
2. Wählen Sie die Anwendungs-Instance. Es handelt sich um die, die den Namen der Elastic Beanstalk-Umgebung trägt.
3. Wählen Sie Connect (Verbinden) aus.

4. Befolgen Sie die Anweisungen, um die Instance mit SSH zu verbinden. Der Befehl ähnelt dem folgenden.

```
$ ssh -i ~/.ssh/mykey ec2-user@ec2-00-55-33-222.us-west-2.compute.amazonaws.com
```

Entnehmen Sie die ID des Synchronisierungsverzeichnisses der letzten Zeile der Einstellungsdatei.

```
[ec2-user ~]$ tail -n 1 /var/app/current/sites/default/settings.php
$config_directories['sync'] = 'sites/default/files/';
$config_4ccfx2sPQm79p1mk5IbUq9S_FokcENO4mxyC-L18-4g_xKj_7j9ydn31kDOYOgnzMu071Tvc4Q/sync';
```

Die Datei enthält außerdem den aktuellen Hash-Schlüssel der Site. Sie können den aktuellen Wert aber ignorieren und einen eigenen Wert verwenden.

Weisen Sie Sync-Verzeichnispfad und Hash-Schlüssel den Umgebungseigenschaften zu. Die benutzerdefinierte Einstellungsdatei aus dem Projektrepository liest diese Eigenschaften, um die Site im Rahmen der Bereitstellung sowie die zuvor eingerichteten Eigenschaften der Datenbankverbindung zu konfigurieren.

Drupal-Konfigurationseigenschaften

- **SYNC_DIR** – Der Pfad zum Sync-Verzeichnis.
- **HASH_SALT** – Jeder Zeichenfolgenwert, der die [Anforderungen an Umgebungseigenschaften \(p. 623\)](#) erfüllt.

So konfigurieren Sie Umgebungseigenschaften in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Geben Sie unter Environment properties (Umgebungseigenschaften) Schlüssel/Wert-Paare ein.

Name	Value
JDBC_CONNECTION_STRING	[redacted]
XRAY_ENABLED	{ "Fn::GetOptionSetting" : {"N:
	[redacted]

6. Wählen Sie Apply (Anwenden) aus.

Schließlich enthält das Beispielprojekt eine Konfigurationsdatei (`loadbalancer-sg.config`), mit der eine Sicherheitsgruppe erstellt und dem Load Balancer der Umgebung zugeordnet wird. Dabei wird die IP-Adresse verwendet, die Sie in der Datei `dev.config` konfiguriert haben, um den HTTP-Zugriff über Port 80 auf Verbindungen aus Ihrem Netzwerk zu beschränken. Andernfalls könnte eine externe Partei potenziell eine Verbindung zur Website herstellen, bevor Sie Drupal installiert und das Administratorkonto konfiguriert haben.

So aktualisieren Sie die Drupal-Konfiguration und entfernen Zugriffsbeschränkungen

1. Löschen Sie die Datei `.ebextensions/loadbalancer-sg.config` aus dem Projektverzeichnis.

```
~/drupal-beanstalk$ rm .ebextensions/loadbalancer-sg.config
```

2. Kopieren Sie die benutzerdefinierte Datei `settings.php` in den Ordner der Site.

```
~/drupal-beanstalk$ cp beanstalk-settings.php sites/default/settings.php
```

3. Erstellen Sie ein Quell-Bundle.

```
~/eb-drupal$ zip ../drupal-beanstalk-v2.zip -r *.[^.*] -x "vendor/*"
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um Drupal in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Konfigurieren der Auto Scaling-Gruppe

Konfigurieren Sie abschließend für die Auto Scaling-Gruppe der Umgebung eine höhere Instance-Mindestanzahl. Führen Sie jederzeit mindestens zwei Instances aus, damit kein Webserver in der Umgebung eine einzelne Fehlerquelle darstellt und damit Sie Änderungen ohne Ausfallzeit der Website bereitstellen können.

So konfigurieren Sie die Auto Scaling-Gruppe Ihrer Umgebung für hohe Verfügbarkeit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Legen Sie im Abschnitt Auto Scaling Group (Auto Scaling-Gruppe) die Option Min instances (Min. Instances) auf **2** fest.
6. Wählen Sie Apply (Anwenden) aus.

Das Beispielprojekt erstellt mit Amazon Elastic File System ein freigegebenes Dateisystem, um das Hochladen von Inhalten über mehrere Instances zu unterstützen. Erstellen Sie einen Beitrag auf der Site und laden Sie Inhalt hoch, der im freigegebenen Dateisystem gespeichert werden soll. Betrachten Sie den Beitrag und aktualisieren Sie die Seite mehrmals, damit Instances genutzt werden und zugleich verifiziert wird, dass das freigegebene Dateisystem funktioniert.

Bereinigen

Wenn Sie nicht mehr mit Elastic Beanstalk arbeiten, können Sie Ihre Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 555\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) und dann Terminate environment (Umgebung beenden) aus.
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Außerdem können Sie Datenbankressourcen beenden, die außerhalb Ihrer Elastic Beanstalk-Umgebung erstellt wurden. Beim Beenden einer Amazon RDS-DB-Instance können Sie einen Snapshot erstellen und die Daten zu einem späteren Zeitpunkt in einer anderen Instance wiederherstellen.

So beenden Sie Ihre RDS DB-Instance

1. Öffnen Sie die [Amazon RDS-Konsole](#).
2. Wählen Sie Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus.
4. Wählen Sie Actions und anschließend Delete.
5. Wählen Sie, ob Sie einen Snapshot erstellen, und wählen Sie anschließend Delete (Löschen).

Nächste Schritte

Im weiteren Verlauf der Anwendungsentwicklung möchten Sie vermutlich Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die [Elastic Beanstalk Command Line Interface \(p. 1008\)](#) (EB CLI) stellt benutzerfreundliche Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk-Umgebungen über die Befehlszeile bereit.

In der Beispielanwendung werden Konfigurationsdateien verwendet, um PHP-Einstellungen zu konfigurieren und eine Tabelle in der Datenbank zu erstellen (sofern nicht bereits vorhanden). Mit einer Konfigurationsdatei können Sie zudem während der Umgebungserstellung die Sicherheitsgruppeneinstellungen der Instance konfigurieren und so zeitaufwendige Konfigurationsaktualisierungen verhindern. Weitere Informationen finden Sie unter [Erweiterte Umgebungsanpassung mit Konfigurationsdateien \(.ebextensions\) \(p. 722\)](#).

Für die Entwicklung sowie für Tests können Sie die Elastic Beanstalk-Funktionalität nutzen und so eine verwaltete DB-Instance direkt in der Umgebung hinzufügen. Detaillierte Anweisungen zum Einrichten einer Datenbank in der Umgebung finden Sie unter [Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung \(p. 609\)](#).

Wenn Sie eine Datenbank mit hoher Verfügbarkeit benötigen, können Sie [Amazon Aurora](#) verwenden. Amazon Aurora ist eine mit MySQL kompatible Datenbank-Engine, die kommerzielle Datenbankfunktionen bei geringen Kosten bietet. Um die Verbindung der Anwendung mit einer anderen Datenbank herzustellen, wiederholen Sie die Schritte der [Sicherheitsgruppenkonfiguration \(p. 314\)](#) und [aktualisieren Sie die RDS-bezogenen Umgebungseigenschaften \(p. 317\)](#).

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, [konfigurieren Sie einen benutzerdefinierten Domänennamen \(p. 644\)](#) für die Umgebung und [aktivieren Sie HTTPS \(p. 778\)](#) für sichere Verbindungen.

Hinzufügen einer Amazon RDS-DB-Instance zur PHP-Anwendungsumgebung

Sie können eine DB-Instance von Amazon Relational Database Service (Amazon RDS) verwenden, um Daten zu speichern, die von Ihrer Anwendung gesammelt und geändert wurden. Die Datenbank kann Ihrer Umgebung angefügt und von Elastic Beanstalk verwaltet werden oder sie kann extern erstellt und verwaltet werden.

Bei der ersten Verwendung von Amazon RDS [fügen Sie eine DB-Instance \(p. 350\)](#) zu einer Testumgebung über die Elastic Beanstalk-Konsole hinzu und prüfen, ob die Anwendung eine Verbindung zu dieser herstellen kann.

Für die Verbindungserstellung zu einer Datenbank [fügen Sie den Treiber zur Anwendung hinzu \(p. 351\)](#), laden die Treiberklasse in den Code und [erstellen ein Verbindungsobjekt \(p. 351\)](#) mit den von Elastic Beanstalk bereitgestellten Umgebungseigenschaften. Die Konfiguration und der Verbindungscode hängen von der Datenbank-Engine und dem Framework ab, die Sie nutzen.

Note

Zu Lernzwecken oder für Testumgebungen können Sie mit Elastic Beanstalk eine DB-Instance hinzufügen.

Für Produktionsumgebungen können Sie eine DB-Instance außerhalb Ihrer Elastic Beanstalk-Umgebung erstellen, um Ihre Umgebungsressourcen von Ihren Datenbankressourcen zu entkoppeln. Auf diese Weise wird beim Beenden Ihrer Umgebung die DB-Instance nicht gelöscht. Mit einer externen DB-Instance können Sie von mehreren Umgebungen aus eine Verbindung mit derselben Datenbank herstellen und [Blau/Grün-Bereitstellungen](#) durchführen. Anleitungen hierzu finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon RDS \(p. 973\)](#).

Abschnitte

- [Hinzufügen einer DB-Instance zu Ihrer Umgebung \(p. 350\)](#)
- [Herunterladen eines Treibers \(p. 351\)](#)
- [Verbinden mit einer Datenbank mit einem PDO oder MySQLi \(p. 351\)](#)
- [Verbinden mit einer Datenbank mit Symfony \(p. 352\)](#)

Hinzufügen einer DB-Instance zu Ihrer Umgebung

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie Apply (Anwenden) aus.

Das Hinzufügen einer DB-Instance dauert ca. 10 Minuten. Sobald die Umgebungsaktualisierung abgeschlossen ist, stehen der Hostname der DB-Instance und andere Verbindungsinformationen über die folgenden Umgebungseigenschaften zur Verfügung:

Eigenschaftenname	Beschreibung	Eigenschaftswert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbankname, ebdb	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: DB Name (DB-Name).
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	In der Amazon RDS-Konsole nicht als Referenz verfügbar.

Weitere Informationen zum Konfigurieren einer internen DB-Instance finden Sie unter [Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung \(p. 609\)](#).

Herunterladen eines Treibers

Zur Verwendung von PHP Data Objects (PDO), um eine Verbindung mit der Datenbank herzustellen, installieren Sie den Treiber für die Datenbank-Engine, die Sie gewählt haben.

- MySQL – [PDO_MYSQL](#)
- PostgreSQL – [PDO_PGSQ](#)
- Oracle – [PDO_OCI](#)
- SQL Server – [PDO_SQLSRV](#)

Weitere Informationen finden Sie unter <http://php.net/manual/en/pdo.installation.php>.

Verbinden mit einer Datenbank mit einem PDO oder MySQLi

Sie können mit `$_SERVER['VARIABLE']` die Verbindungsinformationen in der Umgebung lesen.

Erstellen Sie für ein PDO einen Data Source Name (DSN) aus dem Host, Port und Namen. Geben Sie den DSN an den [Konstruktor für das PDO](#) mit dem Benutzernamen und Passwort der Datenbank weiter.

Example Verbinden mit einer RDS-Datenbank mit 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);
?>
```

Für andere Treiber ersetzen Sie `mysql` durch den Namen Ihres Treibers – `pgsql`, `oci` oder `sqlsrv`.

Übergeben Sie für MySQLi den Hostnamen, den Benutzernamen, das Passwort, den Datenbanknamen und den Port an den `mysqli`-Konstruktör.

Example Verbinden mit einer RDS-Datenbank mit `mysqli_connect()`

```
$link = new mysqli($_SERVER['RDS_HOSTNAME'], $_SERVER['RDS_USERNAME'],
$_SERVER['RDS_PASSWORD'], $_SERVER['RDS_DB_NAME'], $_SERVER['RDS_PORT']);
```

Verbinden mit einer Datenbank mit Symfony

Für Symfony ab Version 3.2 können Sie mit `%env(PROPERTY_NAME)%` Datenbankparameter in einer Konfigurationsdatei basierend auf den Umgebungseigenschaften von Elastic Beanstalk festlegen.

Example 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)%'
```

Unter [Externe Parameter \(Symfony 3.4\)](#) finden Sie weitere Informationen.

In früheren Versionen von Symfony konnte auf Umgebungsvariablen nur zugegriffen werden, wenn der Name mit `SYMFONY__` begann. Dies bedeutet, dass die Elastic Beanstalk-definierten Umgebungseigenschaften nicht zugänglich sind und Sie eigene Umgebungseigenschaften definieren müssen, um die Verbindungsdaten an Symfony zu übermitteln.

Zum Verbinden einer Datenbank mit Symfony 2 erstellen Sie eine Umgebungseigenschaft (p. 290) für jeden Parameter. Anschließend verwenden Sie `%property.name%`, um in einer Konfigurationsdatei auf die für Symfony umgewandelte Variable zuzugreifen. Beispiel: Auf eine Umgebungseigenschaft namens `SYMFONY__DATABASE__USER` kann als `database.user` zugegriffen werden.

```
database_user:      "%database.user%"
```

Unter [Externe Parameter \(Symfony 2.8\)](#) finden Sie weitere Informationen.

Verwenden von Python

In diesem Abschnitt finden Sie Tutorials und Informationen darüber, wie Sie Python-Anwendungen mit AWS Elastic Beanstalk bereitstellen.

In den Themen in diesem Kapitel wird davon ausgegangen, dass Sie über einige Kenntnisse der Elastic Beanstalk-Umgebungen verfügen. Wenn Sie Elastic Beanstalk noch nicht verwendet haben, machen Sie sich im [Tutorial "Erste Schritte" \(p. 3\)](#) mit den Grundlagen vertraut.

Themen

- [Einrichten der Python-Entwicklungsumgebung \(p. 353\)](#)
- [Verwenden der Elastic Beanstalk-Python-Plattform \(p. 355\)](#)
- [Bereitstellen einer Flask-Anwendung in Elastic Beanstalk \(p. 361\)](#)
- [Bereitstellen einer Django-Anwendung in Elastic Beanstalk \(p. 367\)](#)
- [Hinzufügen einer Amazon RDS-DB-Instance zur Python-Anwendungsumgebung \(p. 378\)](#)
- [Python-Tools und -Ressourcen \(p. 380\)](#)

Einrichten der Python-Entwicklungsumgebung

Richten Sie eine Python-Entwicklungsumgebung ein, um Ihre Anwendung vor der Bereitstellung in AWS Elastic Beanstalk lokal zu testen. In diesem Thema finden Sie die Schritte zum Einrichten der Entwicklungsumgebung sowie Links zu den Installationsseiten nützlicher Tools.

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine -Shell für die Ausführung der Befehle. Befehle werden in Auflistungen mit einem vorangestellten Eingabeaufforderungs-Symbol (\$) und dem Namen des aktuellen Verzeichnisses angezeigt, falls angebracht.

```
~/eb-project$ this is a command
this is output
```

Verwenden Sie auf Linux und macOS Ihren bevorzugten Shell- und Paket-Manager. Sie können unter Windows 10 das [Windows-Subsystem für Linux](#) installieren, um eine Windows-Version von Ubuntu und Bash zu erhalten.

Allgemeine Einrichtungsschritte und Tools für alle Sprachen finden Sie unter [Konfigurieren des Entwicklungscomputers \(p. 1005\)](#).

Abschnitte

- [Installieren von Python und Pip \(p. 353\)](#)
- [Verwenden einer virtuellen Umgebung \(p. 354\)](#)
- [Konfigurieren eines Python-Projekts für Elastic Beanstalk \(p. 354\)](#)

Installieren von Python und Pip

Für alle Python-Anwendungen, die Sie mit Elastic Beanstalk bereitstellen, gelten diese Voraussetzungen:

1. Eine Python-Version, die der Elastic Beanstalk-Python-Plattformversion entspricht, die Ihre Anwendung verwendet.
2. Das pip-Dienstprogramm für die jeweilige Python-Version. Damit werden Abhängigkeiten für das Projekt installiert und aufgelistet, sodass Elastic Beanstalk erkennt, wie die Anwendungsumgebung eingerichtet werden soll.
3. Das virtualenv-Paket. Damit wird eine Umgebung für die Anwendungsentwicklung sowie für Anwendungstests erstellt, sodass die Umgebung von Elastic Beanstalk repliziert werden kann, ohne dass weitere Pakete installiert werden, die die Anwendung nicht benötigt.

4. Das `awsebcli`-Paket. Damit wird die Anwendung mit den Dateien, die für die Bereitstellung mit Elastic Beanstalk erforderlich sind, initialisiert.
5. Eine aktive `ssh`-Installation. Damit wird die Verbindung zu den ausgeführten Instances hergestellt, um eine Bereitstellung zu prüfen oder zu debuggen.

Anweisungen zum Installieren von Python, pip und der EB CLI finden Sie unter [Installieren der EB CLI \(p. 1009\)](#).

Verwenden einer virtuellen Umgebung

Sofern die Voraussetzungen erfüllt sind, können Sie eine virtuelle Umgebung mit `virtualenv` einrichten und die Anwendungsabhängigkeiten installieren. Mithilfe einer virtuellen Umgebung können Sie genau erkennen, welche Pakete die Anwendung benötigt, sodass die erforderlichen Pakete auf den EC2-Instances installiert werden, auf denen die Anwendung ausgeführt wird.

So richten Sie eine virtuelle Umgebung ein

1. Öffnen Sie ein Befehlszeilenfenster und geben Sie Folgendes ein:

```
$ virtualenv /tmp/eb_python_app
```

Ersetzen Sie `eb_python_app` durch einen sinnvollen Namen für die Anwendung (z. B. eignet sich der Name der Anwendung gut dafür). Der Befehl `virtualenv` erstellt für Sie eine virtuelle Umgebung im angegebenen Verzeichnis und druckt die Ergebnisse seiner Aktionen:

```
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. Sobald die virtuelle Umgebung bereit ist, starten Sie diese mithilfe des Skripts `activate`, das Sie im Verzeichnis `bin` der Umgebung finden. Um die im vorigen Schritt erstellte Umgebung `eb_python_app` zu starten, geben Sie beispielsweise Folgendes ein:

```
$ source /tmp/eb_python_app/bin/activate
```

In der virtuellen Umgebung wird der Name (z. B. `(eb_python_app)`) am Anfang jeder Eingabeaufforderung angezeigt, damit Sie wissen, dass es sich um eine virtuelle Python-Umgebung handelt.

3. Führen Sie den Befehl `deactivate` aus, um die Verwendung Ihrer virtuellen Umgebung zu beenden und zum standardmäßigen Python-Interpreter des Systems mit allen installierten Bibliotheken zurückzukehren.

```
(eb_python_app) $ deactivate
```

Note

Nach der Erstellung können Sie die virtuelle Umgebung jederzeit starten, indem Sie das Skript `activate` erneut ausführen.

Konfigurieren eines Python-Projekts für Elastic Beanstalk

Sie können die Elastic Beanstalk CLI verwenden, um Ihre Python-Anwendungen auf die Bereitstellung mit Elastic Beanstalk vorzubereiten.

So konfigurieren Sie eine Python-Anwendung für die Bereitstellung mit Elastic Beanstalk

1. Kehren Sie in der [virtuellen Umgebung \(p. 354\)](#) nach oben zum Verzeichnisbaum des Projekts zurück (`python_eb_app`) und geben Sie Folgendes ein:

```
pip freeze >requirements.txt
```

Mit diesem Befehl werden die Namen und Versionen der Pakete, die in der virtuellen Umgebung installiert sind, in die Datei "requirements.txt" kopiert. Wenn beispielsweise das Paket PyYAML in der Version 3.11 in der virtuellen Umgebung installiert ist, enthält die Datei die folgende Zeile:

```
PyYAML==3.11
```

Auf diese Weise verwendet Elastic Beanstalk dieselben Pakete und Versionen, mit denen Sie die Anwendung entwickelt und getestet haben, um die Python-Umgebung der Anwendung zu replizieren.

2. Konfigurieren Sie das Repository der EB CLI mit dem Befehl `eb init`. Folgen Sie den Anweisungen und wählen Sie die Region, die Plattform und andere Optionen aus. Detaillierte Anweisungen finden Sie im [Elastic Beanstalk-Umgebungen mit der EB CLI verwalten \(p. 1020\)](#).

Standardmäßig sucht Elastic Beanstalk nach einer Datei namens `application.py`, um die Anwendung zu starten. Falls diese im erstellten Python-Projekt nicht vorhanden ist, sind bestimmte Anpassungen an der Anwendungsumgebung erforderlich. Außerdem müssen Sie Umgebungsvariablen festlegen, damit die Anwendungsmodule geladen werden können. Weitere Informationen finden Sie unter [Verwenden der Elastic Beanstalk-Python-Plattform \(p. 355\)](#).

Verwenden der Elastic Beanstalk-Python-Plattform

Important

Amazon Linux 2-Plattformversionen unterscheiden sich grundlegend von Amazon Linux AMI-Plattformversionen (Vorgängerversion von Amazon Linux 2). Diese unterschiedlichen Plattformgenerationen sind auf verschiedene Arten inkompatibel. Wenn Sie zu einer Amazon Linux 2-Plattformversion migrieren, müssen Sie die Informationen in [the section called "Aktualisieren auf Amazon Linux 2" \(p. 503\)](#) lesen.

Die Python-Umgebung von AWS Elastic Beanstalk umfasst [Plattformversionen](#) für Python-Webanwendungen, die hinter einem Proxy-Server mit WSGI ausgeführt werden können. Jede Plattformvariante entspricht einer Version von Python, wie z. B. Python 3.4.

Beginnend mit den Amazon Linux 2-Plattformzweigen bietet Elastic Beanstalk [Gunicorn](#) als Standard-WSGI-Server an.

Sie können ein [Procfile](#) zu Ihrem Quellpaket hinzufügen, um den WSGI-Server für Ihre Anwendung anzugeben und zu konfigurieren. Details dazu finden Sie unter [the section called "Procfile" \(p. 359\)](#).

Sie können die von Pipenv erstellten [Pipfile](#)- und [Pipfile.lock](#)-Dateien verwenden, um Python-Paketabhängigkeiten und andere Anforderungen anzugeben. Details zur Angabe von Abhängigkeiten finden Sie unter [the section called "Angeben von Abhängigkeiten" \(p. 360\)](#).

Elastic Beanstalk bietet [Konfigurationsoptionen \(p. 646\)](#), mit denen sich die auf den EC2-Instances in der Elastic Beanstalk-Umgebung ausgeführte Software anpassen lässt. Sie können die für Ihre Anwendung erforderlichen Umgebungsvariablen konfigurieren, die Protokollrotation an Amazon S3 aktivieren und die Ordner in der Anwendungsquelle, die statische Dateien enthalten, den vom Proxy-Server verwendeten Pfaden zuordnen.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das [Ändern der Konfiguration einer ausgeführten Umgebung verfügbar \(p. 658\)](#). Um zu verhindern, dass die Umgebungskonfiguration

beim Beenden verloren geht, können Sie [gespeicherte Konfigurationen \(p. 764\)](#) verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie [Konfigurationsdateien \(p. 722\)](#) einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, wenn vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe [Konfigurationsoptionen \(p. 646\)](#).

Bei Python-Paketen, die über `pip` bereitgestellt werden, können Sie eine Anforderungsdatei in den Stamm des Anwendungsquellcodes aufnehmen. Elastic Beanstalk installiert alle Abhängigkeitspakte, die während der Bereitstellung in einer Anforderungsdatei angegeben sind. Details dazu finden Sie unter [the section called “Angeben von Abhängigkeiten” \(p. 360\)](#).

Details zu den verschiedenen Möglichkeiten zum Erweitern einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#).

Konfigurieren Ihrer Python-Umgebung

Mit den Einstellungen der Python-Plattform können Sie das Verhalten Ihrer Amazon EC2-Instances optimieren. Sie können die Konfiguration der Amazon EC2-Instance der Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-Konsole bearbeiten.

Verwenden Sie die Elastic Beanstalk-Konsole, um die Python-Prozesseinstellungen zu konfigurieren, um AWS X-Ray und die Protokollrotation für Amazon S3 zu aktivieren und um Variablen zu konfigurieren, die die Anwendung aus der Umgebung lesen kann.

So konfigurieren Sie Ihre Python-Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.

Python-Einstellungen

- Proxy server (Proxy-Server) – Der Proxy-Server, der in Ihren Umgebungs-Instances verwendet werden soll. Standardmäßig wird nginx verwendet.
- WSGI-Pfad – Der Name oder Pfad zu Ihrer Hauptanwendungsdatei. Beispiel: `application.py` oder `django/wsgi.py`.
- NumProcesses – Die Anzahl der Prozesse, die auf jeder Anwendungs-Instance ausgeführt werden sollen.
- NumThreads – Die Anzahl der Threads, die in jedem Prozess ausgeführt werden sollen.

AWS X-Ray-Einstellungen

- X-Ray Daemon – Führt den AWS X-Ray-Daemon für die Verarbeitung von Ablaufverfolgungsdaten aus dem [AWS X-Ray SDK for Python](#) aus.

Protokolloptionen

Im Abschnitt mit den Protokolloptionen sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren) – Gibt an, ob die Protokolldateien für die Amazon EC2-Instances der Anwendung in den zugeordneten Amazon S3-Bucket kopiert werden sollen.

Statische Dateien

Um die Leistung zu verbessern, können Sie im Abschnitt Static files (Statische Dateien) den Proxy-Server so konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zur Verzeichniszuordnung fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, statt die Anfrage an Ihre Anwendung weiterzuleiten.

Weitere Informationen zum Konfigurieren statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter [the section called “Statische Dateien” \(p. 776\)](#).

Standardmäßig stellt der Proxy-Server in einer Python-Umgebung alle Dateien im Ordner static unter dem Pfad /static bereit. Wenn beispielsweise die Anwendungsquelle eine Datei mit dem Namen logo.png im Ordner static enthält, wird diese vom Proxy-Server unter [subdomain.elasticbeanstalk.com/static/logo.png](#) an die Benutzer weitergeleitet. Sie können zusätzliche Zuweisungen konfigurieren, wie in diesem Abschnitt erklärt.

Umgebungseigenschaften

Mithilfe von Umgebungseigenschaften stellen Sie Informationen für die Anwendung bereit und konfigurieren Umgebungsvariablen. Beispielsweise können Sie eine Umgebungseigenschaft mit dem Namen CONNECTION_STRING mit einer Verbindungszeichenfolge erstellen, über die Ihre Anwendung eine Verbindung zu einer Datenbank herstellen kann.

Innerhalb der in Elastic Beanstalk ausgeführten Python-Umgebung kann über das Python-Wörterbuch os.environ auf diese Werte zugegriffen werden. Weitere Informationen finden Sie unter [http://docs.python.org/library/os.html](#).

Für den Zugriff auf Schlüssel und Parameter sollte der verwendete Code wie folgt aussehen:

```
import os
endpoint = os.environ['API_ENDPOINT']
```

Umgebungseigenschaften können auch Informationen für ein Framework bereitstellen. Sie können beispielsweise eine Eigenschaft mit dem Namen DJANGO_SETTINGS_MODULE erstellen und so Django zur Verwendung eines bestimmten Einstellungsmoduls konfigurieren. Abhängig von der Umgebung könnte der Wert z. B. development.settings oder production.settings lauten.

Weitere Informationen finden Sie unter [Umgebungseigenschaften und andere Softwareeinstellungen \(p. 621\)](#).

Python-Konfigurations-Namespaces

Mithilfe einer [Konfigurationsdatei \(p. 722\)](#) können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Konfigurationsoptionen können durch den Elastic Beanstalk-Service oder die verwendete Plattform definiert und in Namespaces organisiert werden.

Die Python-Plattform definiert Optionen in den Namespaces

`aws:elasticbeanstalk:environment:proxy,`
`aws:elasticbeanstalk:environment:proxy:staticfiles` und
`aws:elasticbeanstalk:container:python.`

In der folgenden Beispielkonfigurationsdatei werden Einstellungen für Konfigurationsoptionen angegeben, mit denen die Umgebungseigenschaft `DJANGO_SETTINGS_MODULE`, zwei Optionen für statische Dateien, die das Verzeichnis `statichtml` dem Pfad `/html` und das Verzeichnis `staticimages` dem Pfad `/images` zuordnen, sowie weitere Einstellungen im Namespace `aws:elasticbeanstalk:container:python (p. 719)` erstellt und der Apache-Proxy-Server ausgewählt werden. Dieser Namespace enthält Optionen, mit denen Sie den Speicherort des WSGI-Skripts im Quellcode und die Anzahl der Threads und Prozesse, die in WSGI ausgeführt werden sollen, angeben können.

```
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
```

Hinweise

- Wenn Sie eine Amazon Linux AMI Python-Plattformversion (Vorgängerversion von Amazon Linux 2) verwenden, ersetzen Sie den Wert für `WSGIPath` durch `ebdjango/wsgi.py`. Der Wert im Beispiel funktioniert mit dem Unicorn WSGI-Server, der von Amazon Linux AMI-Plattformversionen nicht unterstützt wird.
- Darüber hinaus verwenden diese älteren Plattformversionen einen anderen Namespace für die Konfiguration statischer Dateien: `aws:elasticbeanstalk:container:python:staticfiles`. Er verfügt über die gleichen Optionsnamen und die gleiche Semantik wie der standardmäßige Namespace für statische Dateien.

Konfigurationsdateien unterstützen außerdem zahlreiche Schlüssel für die weitere [Anpassung der Software auf den Umgebungs-Instances \(p. 725\)](#). In diesem Beispiel wird der Schlüssel `packages (p. 726)` für die Memcached-Installation mit `yum` und [Container-Befehlen \(p. 734\)](#) verwendet, um Befehle für die Konfiguration des Servers während der Bereitstellung auszuführen.

```
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"
```

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Konfigurieren des WSGI-Servers mit einer Procfile

Important

Amazon Linux 2-Plattformversionen unterscheiden sich grundlegend von Amazon Linux AMI-Plattformversionen (Vorgängerversion von Amazon Linux 2). Diese unterschiedlichen Plattformgenerationen sind auf verschiedene Arten inkompatibel. Wenn Sie zu einer Amazon Linux 2-Plattformversion migrieren, müssen Sie die Informationen in [the section called “Aktualisieren auf Amazon Linux 2” \(p. 503\)](#) lesen.

Sie können ein `Procfile` zu Ihrem Quellpaket hinzufügen, um den WSGI-Server für Ihre Anwendung anzugeben und zu konfigurieren. Im folgenden Beispiel wird ein `Procfile` verwendet, um uWSGI als Server anzugeben und ihn zu konfigurieren.

Example Procfile

```
web: uwsgi --http :8000 --wsgi-file application.py --master --processes 4 --threads 2
```

Im folgenden Beispiel wird ein `Procfile` verwendet, um Gunicorn, den Standard-WSGI-Server, zu konfigurieren.

Example Procfile

```
web: gunicorn --bind :8000 --workers 3 --threads 2 project.wsgi:application
```

Hinweise

- Wenn Sie einen anderen WSGI-Server als Gunicorn konfigurieren, müssen Sie ihn auch als Abhängigkeit Ihrer Anwendung angeben, damit er auf Ihren Umgebungs-Instances installiert wird. Weitere Informationen zur Abhängigkeitsspezifikation finden Sie unter [the section called “Angeben von Abhängigkeiten” \(p. 360\)](#).
- Der Standardport für den WSGI-Server ist 8000. Wenn Sie in Ihrem `Procfile`-Befehl eine andere Portnummer angeben, legen Sie die [PORT-Umgebungseigenschaft \(p. 621\)](#) auch auf diese Portnummer fest.

Wenn Sie ein `Procfile` verwenden, überschreibt es `aws:elasticbeanstalk:container:python`-Namespaceoptionen, die Sie mithilfe von Konfigurationsdateien festlegen.

Weitere Informationen zur `Procfile`-Verwendung erhalten Sie, indem Sie den Abschnitt Buildfile und `Procfile` in [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#) erweitern.

Angeben von Abhängigkeiten mithilfe einer Anforderungsdatei

Eine typische Python-Anwendung weist Abhängigkeiten zu anderen Python-Paketen von Drittanbietern auf. Mit der Elastic Beanstalk-Python-Plattform haben Sie einige Möglichkeiten, Python-Pakete anzugeben, von denen Ihre Anwendung abhängt.

Verwendung von pip und requirements.txt

Das Standardwerkzeug für die Installation von Python-Paketen ist `pip`. Es verfügt über eine Funktion, mit der Sie alle benötigten Pakete (und deren Versionen) in einer einzelnen Anforderungsdatei spezifizieren können. Weitere Informationen zur Anforderungsdatei finden Sie unter [Requirements File Format](#).

Erstellen Sie eine Datei mit dem Namen `requirements.txt`, und speichern Sie diese im Top-Level-Verzeichnis Ihres Quell-Bundles. Das folgende Beispiel zeigt eine `requirements.txt`-Beispieldatei für Django.

```
Django==1.11.3
MySQL-python==1.2.5
```

In der Entwicklungsumgebung können Sie die Anforderungsdatei mit dem Befehl `pip freeze` generieren.

```
~/my-app$ pip freeze > requirements.txt
```

Damit sichergestellt ist, dass die Anforderungsdatei nur die tatsächlich von der Anwendung genutzten Pakete enthält, sollten Sie eine [virtuelle Umgebung \(p. 354\)](#) einsetzen, in der nur diese Pakete installiert sind. Außerhalb einer virtuellen Umgebung sind in der Ausgabe von `pip freeze` alle `pip`-Pakete enthalten, die auf dem Entwicklungskomputer installiert sind, darunter auch solche, die mit dem Betriebssystem ausgeliefert wurden.

Note

Bei Amazon Linux AMI-Python-Plattformversionen unterstützt Elastic Beanstalk Pipenv oder Pipfiles nicht nativ. Wenn Sie mit Pipenv Abhängigkeiten Ihrer Anwendung zu verwalten, führen Sie den folgenden Befehl aus, um eine `requirements.txt`-Datei zu generieren.

```
~/my-app$ pipenv lock -r > requirements.txt
```

Weitere Informationen finden Sie unter [Generating a requirements.txt](#) in der Pipenv-Dokumentation.

Verwendung von Pipenv und Pipfile

Pipenv ist ein moderneres Python-Verpackungs-Tool. Es kombiniert die Paketinstallation mit der Erstellung und Verwaltung einer Abhängigkeitsdatei und einem `virtualenv` für Ihre Anwendung. Pipenv verwaltet zwei Dateien: `Pipfile` enthält verschiedene Arten von Abhängigkeiten und Anforderungen, und `Pipfile.lock` ist ein Versions-Snapshot, der deterministische Builds ermöglicht. Weitere Informationen finden Sie unter [Pipenv: Python Dev Workflow for Humans](#).

Amazon Linux 2-Python-Plattformversionen unterstützen Pipenv-basierte Anforderungsdateien. Erstellen Sie diese in Ihrer Entwicklungsumgebung und fügen Sie sie in das Quellpaket ein, das Sie für Elastic Beanstalk bereitstellen.

Note

Amazon Linux AMI Python-Plattformversionen (Vorgängerversion von Amazon Linux 2) unterstützen weder Pipenv noch `Pipfile`.

Im folgenden Beispiel wird Pipenv verwendet, um Django und das Django REST-Framework zu installieren.

```
~/my-app$ pipenv install django
~/my-app$ pipenv install djangorestframework
```

Diese Befehle erstellen die Dateien `Pipfile` und `Pipfile.lock`. Platzieren Sie `Pipfile` in das Verzeichnis der obersten Ebene Ihres Quellpaketes, um die neuesten Versionen von Abhängigkeitspaketen auf Ihren Umgebungs-Instances zu erhalten. Alternativ können Sie auch `Pipfile.lock` einschließen, um einen konstanten Satz von Paketversionen abzurufen, die Ihre Entwicklungsumgebung zum Zeitpunkt der Erstellung der Datei widerspiegeln.

Wenn Sie mehr als eine der hier beschriebenen Anforderungsdateien hinzufügen, verwendet Elastic Beanstalk nur eine davon. Die folgende Liste zeigt die Rangfolge in absteigender Reihenfolge.

1. `requirements.txt`
2. `Pipfile.lock`
3. `Pipfile`

Bereitstellen einer Flask-Anwendung in Elastic Beanstalk

Flask ist ein Open-Source-Framework für Webanwendungen für Python. Dieses Tutorial erläutert den Prozess zum Generieren einer Flask-Anwendung und zum Bereitstellen dieser Anwendung in einer AWS Elastic Beanstalk-Umgebung.

In diesem Tutorial führen Sie folgende Aufgaben aus:

- Einrichten einer virtuellen Python-Umgebung mit Flask (p. 362)
- Erstellen einer Flask-Anwendung (p. 362)
- Bereitstellen Ihrer Website mit der EB-CLI (p. 364)
- Bereinigen (p. 366)

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie über Kenntnisse über die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole verfügen. Sofern noch nicht geschehen, befolgen Sie die Anweisungen in [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#), um Ihre erste Elastic Beanstalk-Umgebung zu starten.

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine -Shell für die Ausführung der Befehle. Befehle werden in Auflistungen mit einem vorangestellten Eingabeaufforderungs-Symbol (\$) und dem Namen des aktuellen Verzeichnisses angezeigt, falls angebracht.

```
~/eb-project$ this is a command
this is output
```

Verwenden Sie auf Linux und macOS Ihren bevorzugten Shell- und Paket-Manager. Sie können unter Windows 10 das [Windows-Subsystem für Linux](#) installieren, um eine Windows-Version von Ubuntu und Bash zu erhalten.

Flask erfordert Python 2.7, 3.4 oder eine neuere Version. In diesem Tutorial verwenden wir Python 3.6 und die entsprechende Elastic Beanstalk-Plattformversion. Installieren Sie Python mithilfe der Anleitungen unter [Einrichten der Python-Entwicklungsumgebung \(p. 353\)](#).

Das [Flask](#)-Framework wird im Rahmen des Tutorials installiert.

Dieses Tutorial verwendet die Elastic Beanstalk Command Line Interface (EB CLI). Detaillierte Anweisungen zum Installieren und Konfigurieren der EB CLI finden Sie unter [Installieren der EB CLI \(p. 1009\)](#) und [Konfigurieren der EB CLI \(p. 1017\)](#).

Einrichten einer virtuellen Python-Umgebung mit Flask

Erstellen Sie ein Projektverzeichnis sowie eine virtuelle Umgebung für Ihre Anwendung, und installieren Sie Flask.

So richten Sie die Projektumgebung ein

1. Erstellen Sie ein Projektverzeichnis.

```
~$ mkdir eb-flask  
~$ cd eb-flask
```

2. Erstellen und aktivieren Sie eine virtuelle Umgebung mit dem Namen `virt`:

```
~/eb-flask$ virtualenv virt  
~$ source virt/bin/activate  
(virt) ~/eb-flask$
```

In der Eingabeaufforderung wird (`virt`) vorangestellt, sodass Sie wissen, dass Sie in einer virtuellen Umgebung arbeiten. Verwenden Sie die virtuelle Umgebung für den Rest dieses Tutorials.

3. Installieren Sie Flask mit `pip install`:

```
(virt)~/eb-flask$ pip install flask==1.0.2
```

4. Zeigen Sie die installierten Bibliotheken mit `pip freeze` an:

```
(virt)~/eb-flask$ pip freeze  
click==6.7  
Flask==1.0.2  
itsdangerous==0.24  
Jinja2==2.10  
MarkupSafe==1.0  
Werkzeug==0.14.1
```

Mit diesem Befehl werden alle Pakete aufgeführt, die in der virtuellen Umgebung installiert sind. Da Sie sich in einer virtuellen Umgebung befinden, werden global installierte Pakete wie die EB CLI nicht angezeigt.

5. Speichern Sie die Ausgabe aus `pip freeze` als Datei mit dem Namen `requirements.txt`.

```
(virt)~/eb-flask$ pip freeze > requirements.txt
```

Diese Datei weist Elastic Beanstalk an, die Bibliotheken während der Bereitstellung zu installieren. Weitere Informationen finden Sie unter [Angeben von Abhängigkeiten mithilfe einer Anforderungsdatei \(p. 360\)](#).

Erstellen einer Flask-Anwendung

Als Nächstes erstellen Sie eine Anwendung, die Sie mit Elastic Beanstalk bereitstellen. Wir erstellen einen "Hello World"-RESTful-Webservice.

Erstellen Sie eine neue Textdatei in diesem Verzeichnis mit dem Namen `application.py` mit folgendem Inhalt:

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()
```

In diesem Beispiel wird eine benutzerdefinierte Begrüßung gedruckt, die je nach Pfad, der für den Zugriff auf den Service verwendet wurde, variiert.

Note

Indem Sie `application.debug = True` vor der Ausführung der Anwendung hinzufügen, ist das Debuggen der Ausgabe aktiviert, falls ein Problem auftritt. Es ist eine bewährte Methode für die Entwicklung, Sie sollten jedoch Debug-Anweisungen im Produktionscode entfernen, da eine Debug-Ausgabe interne Aspekte Ihrer Anwendung offenlegen kann.

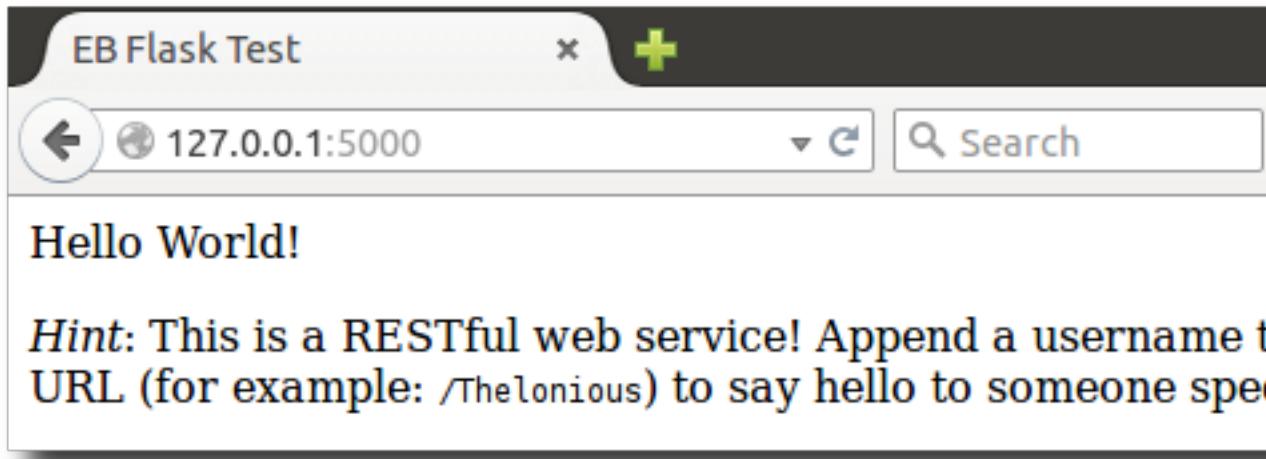
Verwenden Sie `application.py` als Dateinamen und geben Sie ein aufrufbares `application`-Objekt an (in diesem Fall das `Flask`-Objekt), damit Elastic Beanstalk den Code Ihrer Anwendung leichter finden kann.

Führen Sie `application.py` mit Python aus:

```
(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
```

Öffnen Sie <http://127.0.0.1:5000> in Ihrem Webbrowser. Sie sollten sehen, dass die Anwendung ausgeführt wird. Dabei wird folgende Indexseite angezeigt:



Rufen Sie die Ausgabe der Anforderung im Serverprotokoll auf. Mit Ctrl+C stoppen Sie den Server und kehren in die virtuelle Umgebung zurück.

Wenn Sie stattdessen eine Debug-Ausgabe erhalten, beheben Sie die Fehler und stellen Sie sicher, dass die Anwendung lokal ausgeführt wird, bevor Sie sie für Elastic Beanstalk konfigurieren.

Bereitstellen Ihrer Website mit der EB-CLI

Die Anwendung kann nun in Elastic Beanstalk bereitgestellt werden. Das Projektverzeichnis sollte wie folgt aussehen:

```
~/eb-flask/
|-- virt
|-- application.py
`-- requirements.txt
```

Der `virt`-Ordner ist jedoch für die Ausführung der Anwendung in Elastic Beanstalk nicht erforderlich. Bei der Bereitstellung erstellt Elastic Beanstalk eine neue virtuelle Umgebung auf den Server-Instances und installiert die in `requirements.txt` aufgeführten Bibliotheken. Um die Größe des Quell-Bundles zu minimieren, das Sie während der Bereitstellung hochladen, fügen Sie eine `.ebignore` (p. 1018)-Datei hinzu, die der EB CLI mitteilt, dass der Ordner `virt` ausgelassen werden soll.

Example `~/Eb-flask/.ebignore`

```
virt
```

Als Nächstes erstellen Sie die Anwendungsumgebung und stellen die konfigurierte Anwendung in Elastic Beanstalk bereit.

So erstellen Sie eine Umgebung und stellen Ihre Flask-Anwendung bereit:

1. Initialisieren Sie das Repository der EB CLI mit dem Befehl `eb init`:

```
~/eb-flask$ eb init -p python-3.6 flask-tutorial --region us-east-2
```

```
Application flask-tutorial has been created.
```

Mit diesem Befehl wird die neue Anwendung `flask-tutorial` erstellt und das lokale Repository wird zur Umgebungserstellung mit der neuesten Python 3.6-Plattformversion eingerichtet.

2. (Optional) Führen Sie `eb init` erneut aus, um ein Standardschlüsselpaar für die SSH-Verbindung zur EC2-Instance (auf der die Anwendung ausgeführt wird) zu konfigurieren:

```
~/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 ]
```

Wählen Sie ein vorhandenes Schlüsselpaar aus (sofern vorhanden) oder befolgen Sie die Anweisungen, um ein neues zu erstellen. Falls keine Anweisungen angezeigt werden oder Sie die Einstellungen später ändern möchten, führen Sie `eb init -i` aus.

3. Verwenden Sie `eb create`, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen:

```
~/eb-flask$ eb create flask-env
```

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2), die so konfiguriert ist, dass Web-Apps auf der von Ihnen ausgewählten Plattform ausgeführt werden.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance security group (Instance-Sicherheitsgruppe): Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer) – Ein Elastic Load Balancing Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe): Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen

zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.

- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form `subdomain.region.elasticbeanstalk.com` weiterleitet.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

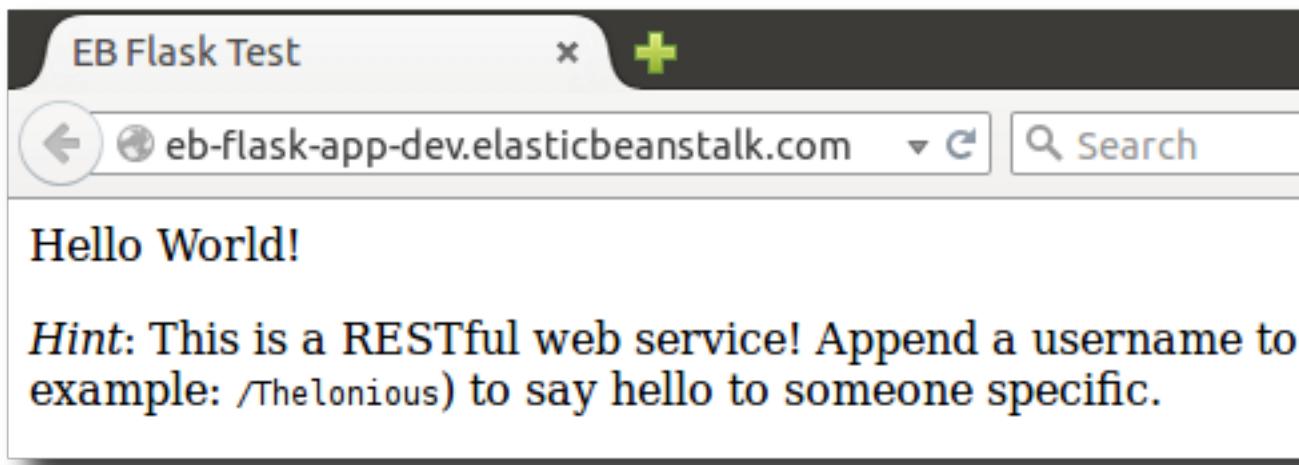
Note

Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#).

Wenn die Umgebungserstellung abgeschlossen ist, öffnen Sie die Website mit eb open:

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

Ein Browserfenster wird mit dem für die Anwendung erstellten Domänennamen geöffnet. Sie sollten dieselbe Flask-Website sehen, die Sie erstellt und lokal getestet haben.



Falls die Anwendung nicht ausgeführt wird oder Sie eine Fehlermeldung erhalten, finden Sie unter [Troubleshooting deployments \(p. 1111\)](#) weitere Informationen zur Ermittlung der Fehlerursache.

Wird die Anwendung ausgeführt, dann haben Sie Ihre erste Flask-Anwendung mit Elastic Beanstalk bereitgestellt. Herzlichen Glückwunsch!

Bereinigen

Falls Sie nicht länger mit Elastic Beanstalk arbeiten möchten, können Sie die Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 555\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie auf Environment actions (Umgebungsaktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Oder mit der EB CLI:

```
~/eb-flask$ eb terminate flask-env
```

Nächste Schritte

Mehr über Flask erfahren Sie unter flask.pocoo.org.

Ein weiteres Python-Web-Framework steht Ihnen unter [Bereitstellen einer Django-Anwendung in Elastic Beanstalk \(p. 367\)](#) zur Verfügung.

Bereitstellen einer Django-Anwendung in Elastic Beanstalk

In diesem Tutorial erfahren Sie, wie die Bereitstellung einer standardmäßigen, automatisch generierten [Django](#)-Website in einer AWS Elastic Beanstalk-Umgebung mit Python erfolgt. In diesem Tutorial erfahren Sie, wie Sie eine Python-Webanwendung in der Cloud mithilfe einer Elastic Beanstalk-Umgebung hosten.

In diesem Tutorial führen Sie folgende Aufgaben aus:

- [Einrichten einer virtuellen Python-Umgebung und Installieren von Django \(p. 368\)](#)
- [Erstellen eines Django-Projekts \(p. 369\)](#)
- [Konfigurieren der Django-Anwendung für Elastic Beanstalk \(p. 371\)](#)
- [Bereitstellen Ihrer Website mit der EB-CLI \(p. 372\)](#)
- [Aktualisieren der Anwendung \(p. 374\)](#)
- [Bereinigen \(p. 378\)](#)

Voraussetzungen

Zur Nutzung eines AWS-Service, einschließlich Elastic Beanstalk, benötigen Sie ein AWS-Konto und Anmeldeinformationen. Weitere Informationen und die Kontoanmeldung finden Sie unter <https://aws.amazon.com/>.

Damit Sie die Schritte in diesem Tutorial ausführen können, müssen alle [allgemeinen Voraussetzungen \(p. 353\)](#) für Python einschließlich der folgenden Pakete installiert sein:

- Python 3.6
- pip
- virtualenv
- awsebcli

Das [Django](#)-Framework wird im Rahmen des Tutorials installiert.

Note

Zum Erstellen von Umgebungen mit der EB-CLI ist eine [Service-Rolle \(p. 21\)](#) erforderlich. Eine Servicerolle generieren Sie, indem Sie eine Umgebung in der Elastic Beanstalk-Konsole erstellen. Falls keine Servicerolle vorhanden ist, versucht die EB-CLI eine Rolle zu erstellen, wenn Sie `eb create` ausführen.

Einrichten einer virtuellen Python-Umgebung und Installieren von Django

Erstellen Sie eine virtuelle Umgebung mit `virtualenv` und anschließend Django und die entsprechenden Abhängigkeiten. Durch die Verwendung einer virtuellen Umgebung wissen Sie genau, welche Pakete Ihre Anwendung benötigt, sodass die erforderlichen Pakete auf den Amazon EC2-Instances installiert werden, auf denen Ihre Anwendung ausgeführt wird.

Die folgenden Schritte zeigen die Befehle, die Sie für Unix-basierte Systeme und Windows eingeben müssen. Diese werden auf separaten Registerkarten angezeigt.

So richten Sie eine virtuelle Umgebung ein

1. Erstellen Sie eine virtuelle Umgebung mit dem Namen `eb-virt`.

Unix-based systems

```
~$ virtualenv ~/eb-virt
```

Windows

```
C:\> virtualenv %HOMEPATH%\eb-virt
```

2. Aktivieren Sie die virtuelle Umgebung.

Unix-based systems

```
~$ source ~/eb-virt/bin/activate  
(eb-virt) ~$
```

Windows

```
C:\>%HOMEPATH%\eb-virt\Scripts\activate  
(eb-virt) C:\>
```

In der Eingabeaufforderung wird `(eb-virt)` vorangestellt, sodass Sie wissen, dass Sie in einer virtuellen Umgebung arbeiten.

Note

In den restlichen Anleitungsschritten wird die Linux-Eingabeaufforderung im Stammverzeichnis `~$` angezeigt. Unter Windows ist das `C:\Users\<USERNAME>`, wobei `<USERNAME>` für Ihren Windows-Anmeldenamen steht.

3. Verwenden Sie `pip`, um Django zu installieren.

```
(eb-virt)~$ pip install django==2.1.1
```

Note

Die Django-Version, die Sie installieren, muss mit der Python-Version auf der Elastic Beanstalk-Python-Konfiguration, die Sie für die Bereitstellung Ihrer Anwendung auswählen, kompatibel sein. Weitere Informationen zur Bereitstellung finden Sie unter [??? \(p. 372\)](#) in diesem Thema.

Weitere Informationen zu aktuellen Python-Plattformversionen finden Sie unter [Python](#) im Dokument zu AWS Elastic Beanstalk-Plattformen.

Informationen zur Kompatibilität von Django-Versionen mit Python finden Sie unter [What Python version can I use with Django?](#) Django 2.2 ist nicht mit der Elastic Beanstalk Python 3.6-Plattform kompatibel. Die neueste kompatible Version ist Django 2.1.

- Um zu überprüfen, ob Django installiert ist, geben Sie Folgendes ein.

```
(eb-virt)~$ pip freeze
Django==2.1.1
...
...
```

Mit diesem Befehl werden alle Pakete aufgeführt, die in der virtuellen Umgebung installiert sind. Später verwenden Sie die Ausgabe dieses Befehls, um Ihr Projekt zur Verwendung mit Elastic Beanstalk zu konfigurieren.

Erstellen eines Django-Projekts

Nun können Sie in der virtuellen Umgebung ein Django-Projekt erstellen und auf dem Computer ausführen.

Note

In diesem Tutorial wird SQLite verwendet, eine Datenbank-Engine, die in Python enthalten ist. Die Datenbank wird mit Ihren Projektdateien bereitgestellt. Für Produktionsumgebungen wird empfohlen, Amazon Relational Database Service (Amazon RDS) zu verwenden und von der Umgebung zu trennen. Weitere Informationen finden Sie unter [Hinzufügen einer Amazon RDS-DB-Instance zur Python-Anwendungsumgebung \(p. 378\)](#).

So erstellen Sie eine Django-Anwendung

- Aktivieren Sie die virtuelle Umgebung.

Unix-based systems

```
~$ source ~/eb-virt/bin/activate
(eb-virt) ~$
```

Windows

```
C:\>%HOMEPATH%\eb-virt\Scripts\activate
(eb-virt) C:\>
```

In der Eingabeaufforderung wird das Präfix (eb-virt) vorangestellt, sodass Sie wissen, dass Sie in einer virtuellen Umgebung arbeiten.

Note

In den restlichen Anleitungsschritten werden die Linux-Eingabeaufforderung ~\$ in Ihrem Stammverzeichnis und das Linux-Stammverzeichnis ~/ angezeigt. Unter Windows ist das C:\Users\<USERNAME>, wobei <USERNAME> für Ihren Windows-Anmeldenamen steht.

2. Verwenden Sie den Befehl `django-admin startproject ebdjango`.

```
(eb-virt)~$ django-admin startproject ebdjango
```

Mit diesem Befehl wird eine Django-Standardwebsite mit dem Namen ebdjango und folgender Verzeichnisstruktur erstellt.

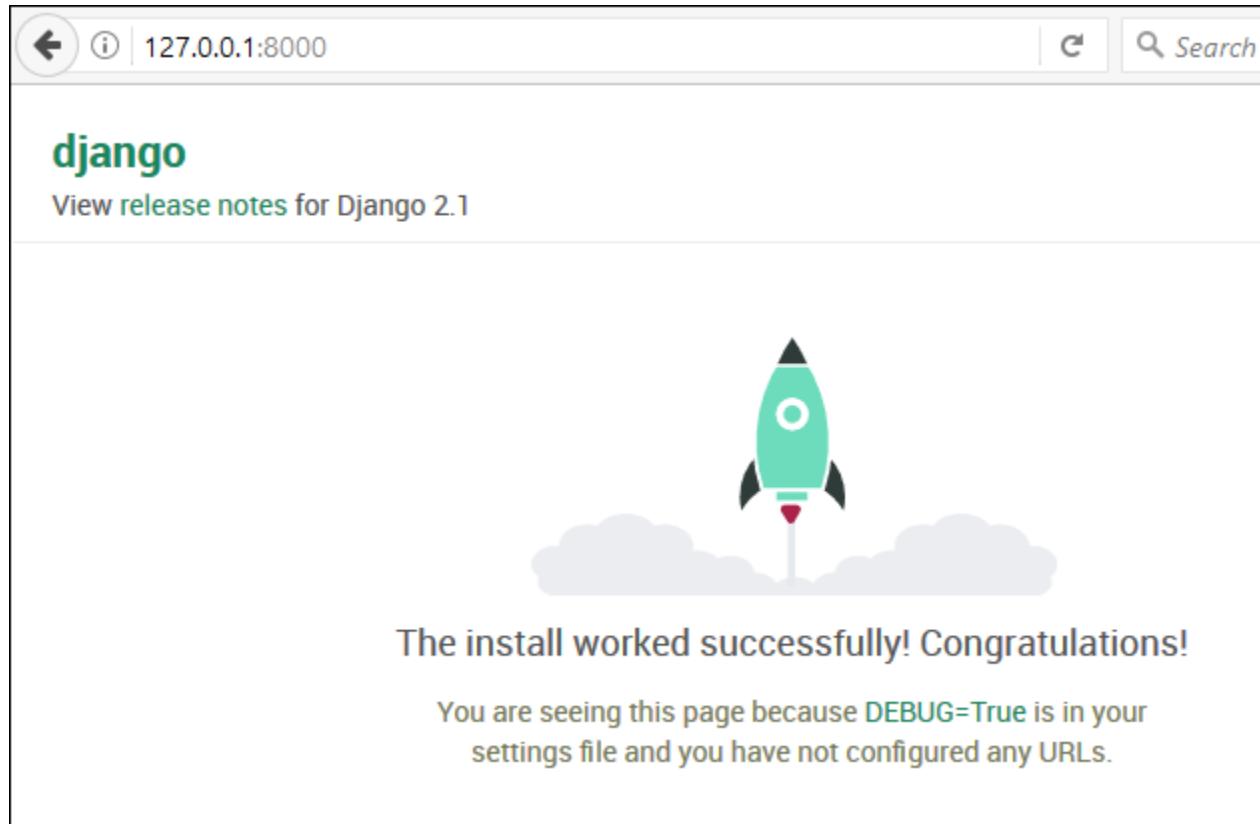
```
~/ebdjango
|-- ebdjango
|   |-- __init__.py
|   |-- settings.py
|   |-- urls.py
|   '-- wsgi.py
`-- manage.py
```

3. Führen Sie die Django-Website lokal mit `manage.py runserver` aus.

```
(eb-virt) ~$ cd ebdjango
```

```
(eb-virt) ~/ebdjango$ python manage.py runserver
```

4. Öffnen Sie zum Anzeigen der Website `http://127.0.0.1:8000` in einem Webbrowser.



5. Rufen Sie die Ausgabe der Anforderung im Serverprotokoll auf. Um den Webserver zu stoppen und zu Ihrer virtuellen Umgebung zurückzukehren, drücken Sie Strg+C.

```
Django version 2.1.1, 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
```

Konfigurieren der Django-Anwendung für Elastic Beanstalk

Sie verfügen nun auf Ihrem lokalen Computer über eine Website mit Django-Unterstützung und können diese für die Bereitstellung in Elastic Beanstalk konfigurieren.

Standardmäßig sucht Elastic Beanstalk nach einer Datei mit dem Namen `application.py`, um Ihre Anwendung zu starten. Da diese im Django-Projekt, das Sie erstellt haben, nicht vorhanden ist, müssen Sie einige Anpassungen an der Anwendungsumgebung vornehmen. Außerdem müssen Sie Umgebungsvariablen festlegen, damit die Anwendungsmodule geladen werden können.

So konfigurieren Sie Ihre Website für Elastic Beanstalk

1. Aktivieren Sie die virtuelle Umgebung.

Unix-based systems

```
~/ebdjango$ source ~/eb-virt/bin/activate
```

Windows

```
C:\Users\<USERNAME>\ebdjango>%HOMEPATH%\eb-virt\Scripts\activate
```

2. Führen Sie `pip freeze` aus und speichern Sie die Ausgabe in der Datei `requirements.txt`.

```
(eb-virt) ~/ebdjango$ pip freeze > requirements.txt
```

Elastic Beanstalk ermittelt anhand der Datei `requirements.txt`, welches Paket auf den EC2-Instances, die zur Anwendungsausführung verwendet werden, installiert werden muss.

3. Erstellen Sie ein Verzeichnis mit dem Namen `.ebextensions`.

```
(eb-virt) ~/ebdjango$ mkdir .ebextensions
```

4. Fügen Sie im Verzeichnis `.ebextensions` eine Konfigurationsdatei ([p. 722](#)) mit dem Namen `django.config` und folgendem Text hinzu.

Example `~/ebdjango/.ebextensions/django.config`

```
option_settings:  
aws:elasticbeanstalk:container:python:  
    WSGIPath: ebdjango.wsgi:application
```

Mit der Einstellung `WSGIPath` wird der Speicherort des WSGI-Skripts angegeben, das von Elastic Beanstalk für den Anwendungsstart verwendet wird.

Note

Wenn Sie eine Amazon Linux AMI Python-Plattformversion (Vorgängerversion von Amazon Linux 2) verwenden, ersetzen Sie den Wert für `WSGIPath` durch `ebdjango/wsgi.py`. Der Wert im Beispiel funktioniert mit dem Gunicorn WSGI-Server, der von Amazon Linux AMI-Plattformversionen nicht unterstützt wird.

5. Deaktivieren Sie die virtuelle Umgebung mit dem Befehl deactivate.

```
(eb-virt) ~/ebdjango$ deactivate
```

Sie können die virtuelle Umgebung jederzeit wieder aktivieren, um Pakete zur Anwendung hinzuzufügen oder diese lokal auszuführen.

Bereitstellen Ihrer Website mit der EB-CLI

Die Anwendung kann nun in Elastic Beanstalk bereitgestellt werden. Das Projektverzeichnis sollte wie folgt aussehen.

```
~/ebdjango/
|-- .ebextensions
|   '-- django.config
|-- ebdjango
|   |-- __init__.py
|   '-- settings.py
|   '-- urls.py
|   '-- wsgi.py
|-- db.sqlite3
|-- manage.py
`-- requirements.txt
```

Als Nächstes erstellen Sie die Anwendungsumgebung und stellen die konfigurierte Anwendung in Elastic Beanstalk bereit.

Unmittelbar nach der Bereitstellung bearbeiten Sie die Django-Konfiguration, um den Domänennamen, den Elastic Beanstalk der Anwendung zugewiesen hat, Djangos ALLOWED_HOSTS hinzuzufügen. Anschließend stellen Sie Ihre Anwendung erneut bereit. Dies ist eine Django-Sicherheitsanforderung, um HTTP-Host-Header-Angriffe zu verhindern. Weitere Informationen finden Sie unter [Host-Header-Validierung](#).

So erstellen Sie eine Umgebung und stellen die Django-Anwendung bereit

Note

Dieses Tutorial verwendet die EB-CLI als Bereitstellungs-Tool. Sie können jedoch auch die Elastic Beanstalk-Konsole zum Bereitstellen einer ZIP-Datei mit den Inhalten des Projekts verwenden.

1. Initialisieren Sie das EB CLI-Repository mit dem Befehl eb init.

```
~/ebdjango$ eb init -p python-3.6 django-tutorial
Application django-tutorial has been created.
```

Dieser Befehl erstellt eine Anwendung mit dem Namen django-tutorial. Außerdem wird Ihr lokales Repository so konfiguriert, dass Umgebungen mit der neuesten Python 3.6-Plattformversion erstellt werden.

2. (Optional) Führen Sie eb init erneut aus, um ein Standardschlüsselpaar für die SSH-Verbindung zur EC2-Instance (auf der die Anwendung ausgeführt wird) zu konfigurieren.

```
~/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 ]
```

Wählen Sie ein Schlüsselpaar aus (sofern vorhanden) oder befolgen Sie die Anweisungen, um eines zu erstellen. Falls keine Anweisungen angezeigt werden oder Sie die Einstellungen später ändern möchten, führen Sie eb init -i aus.

3. Verwenden Sie eb create, um eine Umgebung zu erstellen und die Anwendung darin bereitzustellen.

```
~/ebdjango$ eb create django-env
```

Note

Wenn eine Fehlermeldung angezeigt wird, dass eine Servicerolle erforderlich ist, führen Sie eb create interaktiv aus (ohne Angabe eines Umgebungsnamens). Die EB CLI erstellt die Rolle dann für Sie.

Mit diesem Befehl wird eine lastenverteilte Elastic Beanstalk-Umgebung namens django-env erstellt. Das Erstellen einer Umgebung dauert ca. 5 Minuten. Bei der zur Anwendungsausführung erforderlichen Ressourcenerstellung durch Elastic Beanstalk werden die ausgegebenen Informationsmeldungen von der EB CLI an den Terminal weitergeleitet.

4. Nachdem die Umgebung erstellt wurde, können Sie den Domänennamen der neuen Umgebung ermitteln, indem Sie eb status ausführen.

```
~/ebdjango$ eb status
Environment details for: django-env
  Application name: django-tutorial
  ...
  CNAME: eb-django-app-dev.elasticbeanstalk.com
  ...
```

Der Domänenname der Umgebung ist der Wert der Eigenschaft CNAME.

5. Öffnen Sie die settings.py-Datei im Verzeichnis ebdjango. Suchen Sie die ALLOWED_HOSTS-Einstellung und fügen Sie dann den Domänennamen Ihrer Anwendung, den Sie im vorherigen Schritt gefunden haben, zum Wert der Einstellung hinzu. Wenn Sie diese Einstellung nicht in der Datei finden, fügen Sie sie in einer neuen Zeile hinzu.

```
...
ALLOWED_HOSTS = ['eb-django-app-dev.elasticbeanstalk.com']
```

6. Speichern Sie die Datei und stellen Sie die Anwendung dann bereit, indem Sie eb deploy ausführen. Bei Ausführung von eb deploy werden die Inhalte des Projektverzeichnisses von der EB CLI gebündelt und in der Umgebung bereitgestellt.

```
~/ebdjango$ eb deploy
```

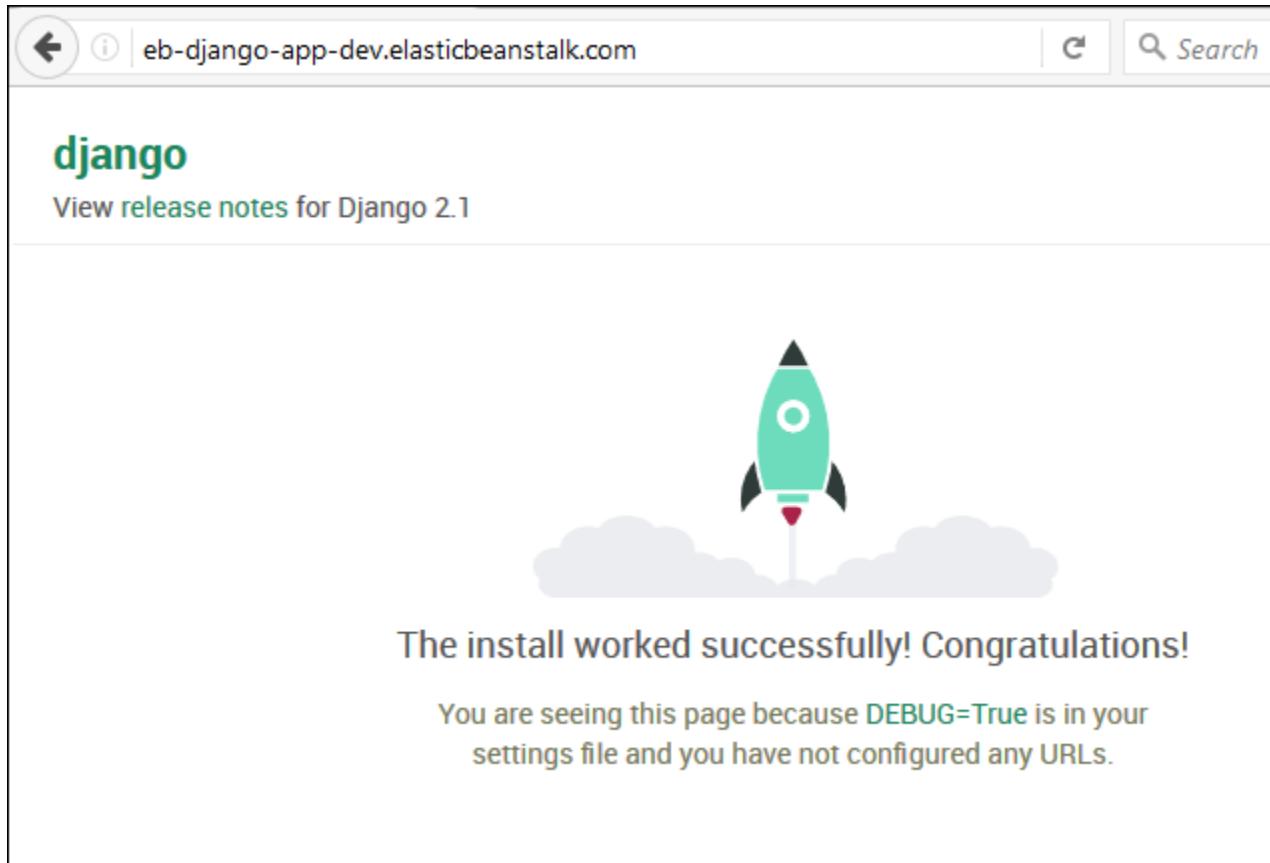
Note

Wenn Sie bei Ihrem Projekt Git verwenden, lesen Sie [Verwenden der EB CLI mit Git \(p. 1027\)](#).

7. Wenn die Umgebungsaktualisierung abgeschlossen ist, öffnen Sie die Website mit eb open.

```
~/ebdjango$ eb open
```

Ein Browserfenster wird mit dem für die Anwendung erstellten Domänennamen geöffnet. Sie sollten dieselbe Django-Website sehen, die Sie erstellt und lokal getestet haben.



Falls die Anwendung nicht ausgeführt wird oder Sie eine Fehlermeldung erhalten, finden Sie unter [Troubleshooting deployments \(p. 1111\)](#) weitere Informationen zur Ermittlung der Fehlerursache.

Wird die Anwendung ausgeführt, dann haben Sie Ihre erste Django-Anwendung mit Elastic Beanstalk bereitgestellt. Herzlichen Glückwunsch!

Aktualisieren der Anwendung

Ihre Anwendung wird nun in Elastic Beanstalk ausgeführt. Sie können die Anwendung oder deren Konfiguration aktualisieren und erneut bereitstellen. Elastic Beanstalk führt dabei die Aktualisierung der Instances aus und startet Ihre neue Anwendungsversion.

In diesem Beispiel aktivieren Sie die Django-Administrationskonsole und konfigurieren weitere Einstellungen.

Ändern der Website-Einstellungen

Standardmäßig erfolgt die Zeitangabe der Django-Website in koordinierter Weltzeit (UTC). Sie können diese Einstellung ändern, indem Sie eine Zeitzone in `settings.py` angeben.

So ändern Sie die Zeitzone der Website

1. Ändern Sie die Einstellung `TIME_ZONE` in `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
```

Eine Liste der Zeitzonen finden Sie [hier](#).

2. Stellen Sie die Anwendung in Ihrer Elastic Beanstalk-Umgebung bereit.

```
~/ebdjango/$ eb deploy
```

Erstellen eines Website-Administrators

Sie können einen Websiteadministrator für Ihre Django-Anwendung erstellen, um direkt von der Website auf die Administrationskonsole zuzugreifen. Die Anmeldeinformationen für den Administrator werden sicher im lokalen Datenbank-Image gespeichert, welches in das von Django generierte Standardprojekt eingebunden ist.

So erstellen Sie einen Websiteadministrator

1. Initialisieren Sie die lokale Datenbank der Django-Anwendung.

```
(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. Führen Sie zum Erstellen eines Administrators `manage.py createsuperuser` aus.

```
(eb-virt) ~/ebdjango$ python manage.py createsuperuser
Username: admin
Email address: me@mydomain.com
Password: *****
Password (again): *****
Superuser created successfully.
```

3. Um Django einen Speicherort für statische Dateien zuzuweisen, definieren Sie `STATIC_ROOT` in `settings.py`.

Example `~/ebdjango/ebdjango/settings.py`

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.1/howto/static-files/
```

```
STATIC_URL = '/static/'  
STATIC_ROOT = 'static'
```

4. Führen Sie `manage.py collectstatic` aus, um das Verzeichnis `static` mit statischen Komponenten (JavaScript, CSS und Images) für die Administrationswebsite zu füllen.

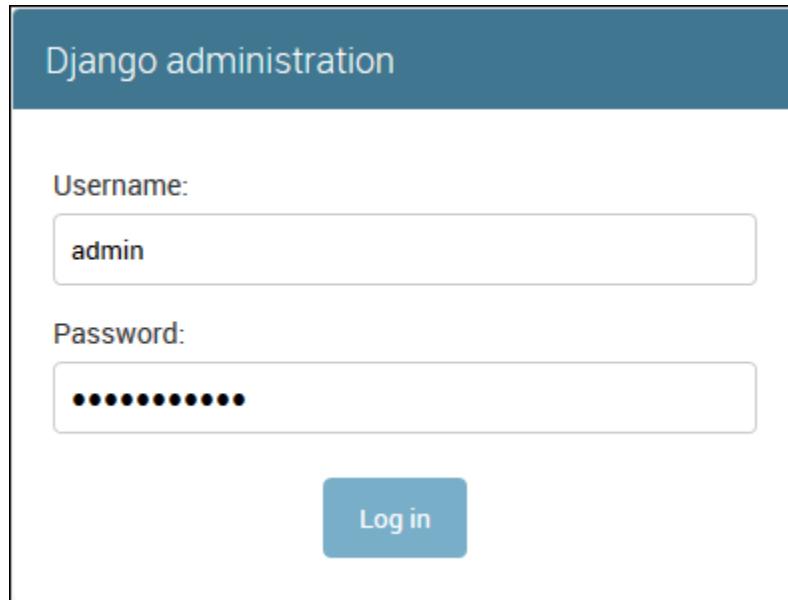
```
(eb-virt) ~/ebdjango$ python manage.py collectstatic  
119 static files copied to ~/ebdjango/static
```

5. Stellen Sie die Anwendung bereit.

```
~/ebdjango$ eb deploy
```

6. Rufen Sie die Administrationskonsole auf, indem Sie die Website im Browser öffnen und `/admin/` an die Website-URL anfügen, z. B. im Folgenden.

```
http://djang-env.p33kq46sfh.us-west-2.elasticbeanstalk.com/admin/
```



7. Melden Sie sich mit dem in Schritt 2 konfigurierten Benutzernamen und dem zugehörigen Passwort an.

The screenshot shows the Django administration interface. At the top, it says "Django administration" and "WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT". Below that, it says "Site administration". A blue header bar labeled "AUTHENTICATION AND AUTHORIZATION" contains two items: "Groups" with a "+ Add" and "Change" button, and "Users" with a "+ Add" and "Change" button. To the right, there's a sidebar titled "Recent actions" which is currently empty, and "My actions" which also says "None available".

Sie können ein ähnliches Verfahren der lokalen Aktualisierung/Tests gefolgt von eb deploy verwenden. Elastic Beanstalk führt die Aktualisierung der Live-Server aus, sodass Sie sich nicht um die Serveradministration kümmern müssen, sondern sich ganz auf die Anwendungsbereitstellung konzentrieren können.

Hinzufügen einer Konfigurationsdatei für die Datenbankmigration

Sie können Ihrem .ebextensions-Skript Befehle hinzufügen, die beim Aktualisieren der Website ausgeführt werden. Auf diese Weise können Sie automatisch Datenbankmigrationen generieren.

So fügen Sie bei der Anwendungsbereitstellung einen Migrationsschritt hinzu

1. Erstellen Sie eine Konfigurationsdatei (p. 722) namens db-migrate.config mit dem folgenden Inhalt.

Example ~/ebdjango/.ebextensions/db-migrate.config

```
container_commands:  
  01_migrate:  
    command: "django-admin.py migrate"  
    leader_only: true  
option_settings:  
  aws:elasticbeanstalk:application:environment:  
    DJANGO_SETTINGS_MODULE: ebdjango.settings
```

Diese Konfigurationsdatei führt den django-admin.py migrate-Befehl während des Bereitstellungsprozesses aus, bevor Ihre Anwendung gestartet wird. Da die Befehlausführung vor dem Start der Anwendung erfolgt, muss die Umgebungsvariable DJANGO_SETTINGS_MODULE explizit konfiguriert werden (in der Regel wird dies von wsgi.py während des Starts übernommen). Geben Sie leader_only: true im Befehl an, damit dieser auch bei einer Bereitstellung auf mehreren Instances nur einmal ausgeführt wird.

2. Stellen Sie die Anwendung bereit.

```
~/ebdjango$ eb deploy
```

Bereinigen

Um Instance-Stunden und weitere AWS-Ressourcen zwischen den Bereitstellungssitzungen einzusparen, beenden Sie die Elastic Beanstalk-Umgebung mit eb terminate.

```
~/ebdjango$ eb terminate django-env
```

Mit diesem Befehl werden die Umgebung und alle zugehörigen AWS-Ressourcen beendet. Die Anwendung wird jedoch nicht gelöscht, sodass Sie durch die erneute Ausführung von eb create jederzeit zusätzliche Umgebungen mit der gleichen Konfiguration erstellen können. Weitere Informationen zu EB CLI-Befehlen finden Sie unter [Elastic Beanstalk-Umgebungen mit der EB CLI verwalten \(p. 1020\)](#).

Sofern Sie die Beispielanwendung nicht mehr benötigen, können Sie auch den Projektordner sowie die virtuelle Umgebung entfernen.

```
~$ rm -rf ~/eb-virt
~$ rm -rf ~/ebdjango
```

Nächste Schritte

Weitere Informationen zu Django und ein ausführliches Tutorial finden Sie in der [offiziellen Dokumentation](#).

Ein weiteres Python-Web-Framework steht Ihnen unter [Bereitstellen einer Flask-Anwendung in Elastic Beanstalk \(p. 361\)](#) zur Verfügung.

Hinzufügen einer Amazon RDS-DB-Instance zur Python-Anwendungsumgebung

Sie können eine DB-Instance von Amazon Relational Database Service (Amazon RDS) verwenden, um Daten zu speichern, die von Ihrer Anwendung gesammelt und geändert wurden. Die Datenbank kann Ihrer Umgebung angefügt und von Elastic Beanstalk verwaltet werden oder sie kann extern erstellt und verwaltet werden.

Bei der ersten Verwendung von Amazon RDS [fügen Sie eine DB-Instance \(p. 379\)](#) zu einer Testumgebung über die Elastic Beanstalk-Managementkonsole hinzu und prüfen, ob die Anwendung eine Verbindung zu dieser herstellen kann.

Für die Verbindungserstellung zu einer Datenbank [fügen Sie einen Treiber zur Anwendung hinzu \(p. 380\)](#), laden den Treiber in den Code und [erstellen ein Verbindungsobjekt \(p. 380\)](#) mit den von Elastic Beanstalk bereitgestellten Umgebungseigenschaften. Die Konfiguration und der Verbindungscode hängen von der Datenbank-Engine und dem Framework ab, die Sie nutzen.

Note

Zu Lernzwecken oder für Testumgebungen können Sie mit Elastic Beanstalk eine DB-Instance hinzufügen.

Für Produktionsumgebungen können Sie eine DB-Instance außerhalb Ihrer Elastic Beanstalk-Umgebung erstellen, um Ihre Umgebungsressourcen von Ihren Datenbankressourcen zu entkoppeln. Auf diese Weise wird beim Beenden Ihrer Umgebung die DB-Instance nicht gelöscht. Mit einer externen DB-Instance können Sie von mehreren Umgebungen aus eine Verbindung mit

derselben Datenbank herstellen und [Blau/Grün-Bereitstellungen](#) durchführen. Anleitungen hierzu finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon RDS \(p. 973\)](#).

Abschnitte

- [Hinzufügen einer DB-Instance zu Ihrer Umgebung \(p. 379\)](#)
- [Herunterladen eines Treibers \(p. 380\)](#)
- [Verbinden mit einer Datenbank \(p. 380\)](#)

Hinzufügen einer DB-Instance zu Ihrer Umgebung

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie Apply (Anwenden) aus.

Das Hinzufügen einer DB-Instance dauert ca. 10 Minuten. Sobald die Umgebungsaktualisierung abgeschlossen ist, stehen der Hostname der DB-Instance und andere Verbindungsinformationen über die folgenden Umgebungseigenschaften zur Verfügung:

Eigenschaftenname	Beschreibung	Eigenschaftswert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbankname, ebdb	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: DB Name (DB-Name).
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).

Eigenschaftenname	Beschreibung	Eigenschaftswert
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	In der Amazon RDS-Konsole nicht als Referenz verfügbar.

Weitere Informationen zum Konfigurieren einer internen DB-Instance finden Sie unter [Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung \(p. 609\)](#).

Herunterladen eines Treibers

Fügen Sie den Datenbanktreiber zur [requirements-Datei \(p. 360\)](#) des Projekts hinzu.

Example Beispieldatei "requirements.txt" – Django mit MySQL

```
Django==2.0.3
mysqlclient==1.3.12
```

Gängige Treiberpakete für Python

- MySQL – MySQL-python (Python 2) , mysqlclient (Python 3)
- PostgreSQL – psycopg2
- Oracle – cx_Oracle
- SQL Server – adodbapi

Verbinden mit einer Datenbank

Elastic Beanstalk stellt Verbindungsinformationen für zugehörige DB-Instances in Umgebungseigenschaften bereit. Mit `os.environ['VARIABLE']` können Sie die Eigenschaften lesen und eine Datenbankverbindung konfigurieren.

Example Django-Einstellungsdatei – DATABASES Dictionary

```
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'],
        }
    }
```

Python-Tools und -Ressourcen

Es gibt zahlreiche Möglichkeiten, um zusätzliche Unterstützung bei der Entwicklung von Python-Anwendungen zu erhalten:

Ressource	Beschreibung
Boto (AWS SDK for Python)	Installieren Sie Boto mit GitHub.

Ressource	Beschreibung
Python-Entwicklungsforum	Hier können Sie Fragen stellen und Feedback erhalten.
Python-Entwicklerzentrum	Umfangreiches Kit mit Beispiel-Code, Dokumentation, Tools und zusätzlichen Ressourcen.

Erstellen und Bereitstellen von Ruby-Anwendungen in Elastic Beanstalk

Themen

- [Einrichten Ihrer Ruby-Entwicklungsumgebung \(p. 381\)](#)
- [Verwenden der Elastic Beanstalk-Ruby-Plattform \(p. 383\)](#)
- [Bereitstellen einer Rails-Anwendung in Elastic Beanstalk \(p. 388\)](#)
- [Bereitstellen einer Sinatra-Anwendung in Elastic Beanstalk \(p. 394\)](#)
- [Hinzufügen einer Amazon RDS-DB-Instance zur Ruby-Anwendungsumgebung \(p. 398\)](#)

AWS Elastic Beanstalk for Ruby vereinfacht das Bereitstellen, Verwalten und Skalieren Ihre Ruby-Webanwendungen mithilfe von Amazon Web Services. Elastic Beanstalk steht jedem zur Verfügung, der eine Webanwendung mit Ruby entwickelt oder hostet. In diesem Abschnitt werden Schritt-für-Schritt-Anleitungen für das Bereitstellen einer Beispielanwendung in Elastic Beanstalk mithilfe der Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI) bereitgestellt und die anschließende Aktualisierung der Anwendung zur Verwendung von [Rails](#)- und [Sinatra](#)-Frameworks für Webanwendungen.

In den Themen in diesem Kapitel wird davon ausgegangen, dass Sie über einige Kenntnisse der Elastic Beanstalk-Umgebungen verfügen. Wenn Sie Elastic Beanstalk noch nicht verwendet haben, machen Sie sich im [Tutorial „Erste Schritte“ \(p. 3\)](#) mit den Grundlagen vertraut.

Einrichten Ihrer Ruby-Entwicklungsumgebung

Richten Sie eine Ruby-Entwicklungsumgebung ein, um Ihre Anwendung vor der Bereitstellung in AWS Elastic Beanstalk lokal zu testen. In diesem Thema finden Sie die Schritte zum Einrichten der Entwicklungsumgebung sowie Links zu den Installationsseiten nützlicher Tools.

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine -Shell für die Ausführung der Befehle. Befehle werden in Auflistungen mit einem vorangestellten Eingabeaufforderungs-Symbol (\$) und dem Namen des aktuellen Verzeichnisses angezeigt, falls angebracht.

```
~/eb-project$ this is a command  
this is output
```

Verwenden Sie auf Linux und macOS Ihren bevorzugten Shell- und Paket-Manager. Sie können unter Windows 10 das [Windows-Subsystem für Linux](#) installieren, um eine Windows-Version von Ubuntu und Bash zu erhalten.

Allgemeine Einrichtungsschritte und Tools für alle Sprachen finden Sie unter [Konfigurieren des Entwicklungscomputers für die Verwendung mit Elastic Beanstalk \(p. 1005\)](#).

Abschnitte

- [Installieren von Ruby \(p. 382\)](#)

- [Installieren des AWS SDKs für Ruby \(p. 383\)](#)
- [Installieren einer IDE oder eines Texteditors \(p. 383\)](#)

Installieren von Ruby

Installieren Sie GCC, wenn Sie keinen C-Compiler haben. Unter Ubuntu verwenden Sie apt.

```
~$ sudo apt install gcc
```

Verwenden Sie auf Amazon Linux yum.

```
~$ sudo yum install gcc
```

Installieren Sie RVM, um Ruby-Sprachinstallationen auf Ihrem Rechner zu verwalten. Verwenden Sie die Befehle von [rvm.io](#), um die Projektschlüssel zu erhalten und das Installationsskript auszuführen.

```
~$ gpg --keyserver hkp://keys.gnupg.net --recv-keys key1 key2
~$ curl -ssl https://get.rvm.io | bash -s stable
```

Dieses Skript installiert RVM in einem Ordner namens `.rvm` in Ihrem Benutzerverzeichnis und ändert Ihr Shell-Profil, um ein Setup-Skript zu laden, wenn Sie ein neues Terminal öffnen. Laden Sie die Skripts manuell, um beginnen zu können.

```
~$ source ~/.rvm/scripts/rvm
```

Verwenden Sie `rvm get head`, um die neueste Version zu erhalten.

```
~$ rvm get head
```

Zeigen Sie die verfügbaren Versionen von Ruby an.

```
~$ rvm list known
# MRI Rubies
[ruby-]2.1[.10]
[ruby-]2.2[.10]
[ruby-]2.3[.7]
[ruby-]2.4[.4]
[ruby-]2.5[.1]
...
```

Lesen Sie im Dokument für AWS Elastic Beanstalk-Plattformen nach, um die neueste Version von [Ruby](#) zu finden, die für eine Elastic Beanstalk-Plattform verfügbar ist. Installieren Sie diese Version.

```
~$ rvm install 2.5.1
Searching for binary rubies, this might take some time.
Found remote file https://rvm_io.global.ssl.fastly.net/binaries/ubuntu/16.04/x86_64/
ruby-2.5.1.tar.bz2
Checking requirements for ubuntu.
Requirements installation successful.
ruby-2.5.1 - #configure
ruby-2.5.1 - #download
...
```

Testen Sie Ihre Ruby-Installation.

```
~$ ruby --version
ruby 2.5.1p57 (2018-03-29 revision 63029) [x86_64-linux]
```

Installieren des AWS SDKs für Ruby

Wenn Sie die AWS-Ressourcen aus der Anwendung heraus verwalten möchten, installieren Sie das AWS SDK for Ruby. Sie können mit dem SDK for Ruby beispielsweise Amazon DynamoDB (DynamoDB) zum Speichern von Benutzer- und Sitzungsinformationen verwenden, ohne eine relationale Datenbank zu erstellen.

Installieren Sie das SDK for Ruby und seine Abhängigkeiten mit dem `gem`-Befehl.

```
$ gem install aws-sdk
```

Weitere Informationen und Installationsanleitungen finden Sie auf der [AWS SDK for Ruby-Homepage](#).

Installieren einer IDE oder eines Texteditors

Integrierte Entwicklungsumgebungen (IDEs) bieten zahlreiche Funktionen, mit denen die Anwendungsentwicklung vereinfacht wird. Falls Sie noch keine IDE für die Ruby-Entwicklung eingesetzt haben, probieren Sie Aptana und RubyMine aus. Dann sehen Sie, welche IDE für Sie am besten geeignet ist.

- [Installieren von Aptana](#)
- [RubyMine](#)

Note

Mit einer IDE werden dem Projektordner möglicherweise Dateien hinzugefügt, die Sie unter Umständen nicht an die Quellüberwachung übergeben möchten. Damit diese Dateien nicht an die Quellüberwachung übergeben werden, setzen Sie `.gitignore` oder die Entsprechung Ihres Quellüberwachungstools ein.

Wenn Sie gerade mit dem Programmieren beginnen und nicht alle Funktionen einer IDE benötigen, können Sie [Sublime Text installieren](#).

Verwenden der Elastic Beanstalk-Ruby-Plattform

Important

Amazon Linux 2-Plattformversionen unterscheiden sich grundlegend von Amazon Linux AMI-Plattformversionen (vor Amazon Linux 2). Diese unterschiedlichen Plattformgenerationen sind auf verschiedene Arten inkompatibel. Wenn Sie zu einer Amazon Linux 2-Plattformversion migrieren, müssen Sie die Informationen in [the section called “Aktualisieren auf Amazon Linux 2” \(p. 503\)](#) lesen.

Bei der Ruby-Plattform von AWS Elastic Beanstalk handelt es sich um eine Reihe von [Umgebungskonfigurationen](#) für Ruby-Webanwendungen, die hinter einem nginx-Proxy-Server unter einem Puma-Anwendungsserver ausgeführt werden können. Jede Plattformvariante entspricht einer Version von Ruby.

Wenn Sie RubyGems verwenden, können Sie durch das [Einbinden einer Gemfile-Datei \(p. 386\)](#) in das Quell-Bundle die Pakete während der Bereitstellung installieren.

Ihre Anwendung läuft möglicherweise unter einem anderen Anwendungsserver, z. B. Passenger. Sie können eine `Procfile`-Datei verwenden, um einen anderen Anwendungsserver zu starten, und eine, `Gemfile`-Datei, um ihn zu installieren. Details dazu finden Sie unter [the section called “Procfile” \(p. 387\)](#).

Note

Wenn Sie eine Amazon Linux AMI-Ruby-Plattformversion (vor Amazon Linux 2) verwenden, beachten Sie bitte, dass Elastic Beanstalk zwei Varianten dieser Versionen anbietet – mit Puma und mit Passenger. Wenn die Anwendung den Passenger-Anwendungsserver benötigt, können Sie die entsprechende Passenger-Plattformversion verwenden und müssen keine zusätzliche Konfiguration vornehmen.

Elastic Beanstalk bietet [Konfigurationsoptionen \(p. 646\)](#), mit denen sich die auf den Amazon Elastic Compute Cloud (Amazon EC2)-Instances in der Elastic Beanstalk-Umgebung ausgeführte Software anpassen lässt. Sie können die für Ihre Anwendung erforderlichen Umgebungsvariablen konfigurieren, die Protokollrotation an Amazon S3 aktivieren und die Ordner in der Anwendungsquelle, die statische Dateien enthalten, den vom Proxy-Server verwendeten Pfaden zuordnen. Die Plattform verfügt auch über einige allgemeine Umgebungsvariablen für Rails und Rack, die eine einfache Erkennung und Nutzung ermöglichen.

In der Elastic Beanstalk-Konsole sind Konfigurationsoptionen für das [Ändern der Konfiguration einer ausgeführten Umgebung \(p. 658\)](#) verfügbar. Um zu verhindern, dass die Umgebungskonfiguration beim Beenden verloren geht, können Sie [gespeicherte Konfigurationen \(p. 764\)](#) verwenden, um Ihre Einstellungen zu speichern und sie später für eine andere Umgebung zu übernehmen.

Zum Speichern der Einstellungen im Quellcode können Sie [Konfigurationsdateien \(p. 722\)](#) einschließen. Die Einstellungen in Konfigurationsdateien werden jedes Mal verwendet, wenn Sie eine Umgebung erstellen oder Ihre Anwendung bereitstellen. Mit Konfigurationsdateien können Sie auch Pakete installieren, Skripts ausführen und andere Instance-Anpassungen bei Bereitstellungen vornehmen.

Die in der Elastic Beanstalk-Konsole angewendeten Einstellungen überschreiben die entsprechenden Einstellungen in Konfigurationsdateien, wenn vorhanden. So sind Standardeinstellungen in Konfigurationsdateien möglich, die Sie mit umgebungsspezifischen Einstellungen in der Konsole überschreiben können. Für weitere Informationen zur Rangfolge und zu anderen Methoden zum Ändern der Einstellungen siehe [Konfigurationsoptionen \(p. 646\)](#).

Details zu den verschiedenen Möglichkeiten für das Erweitern einer Linux-basierten Elastic Beanstalk-Plattform finden Sie unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#).

Konfigurieren Ihrer Ruby-Umgebung

Mit der Elastic Beanstalk-Konsole können Sie die Protokollrotation für Amazon S3 aktivieren und Variablen konfigurieren, die die Anwendung in der Umgebung lesen kann.

So greifen Sie auf die Softwarekonfigurationseinstellungen für die Umgebung zu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.

Protokolloptionen

Im Abschnitt Log options (Protokolloptionen) sind zwei Einstellungen vorhanden:

- Instance profile (Instance-Profil) – Gibt das Instance-Profil an, das zum Zugriff auf den entsprechenden Amazon S3-Bucket der Anwendung berechtigt ist.
- Enable log file rotation to Amazon S3 (Protokolldateirotation für Amazon S3 aktivieren) – Gibt an, ob die Protokolldateien für die Amazon EC2-Instances der Anwendung in den der Anwendung zugeordneten Amazon S3-Bucket kopiert werden sollen.

Statische Dateien

Um die Leistung zu verbessern, können Sie im Abschnitt Static files (Statische Dateien) den Proxy-Server so konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Legen Sie für jedes Verzeichnis den virtuellen Pfad zur Verzeichniszuordnung fest. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, statt die Anfrage an Ihre Anwendung weiterzuleiten.

Weitere Informationen zum Konfigurieren statischer Dateien mit der Elastic Beanstalk-Konsole finden Sie unter [the section called “Statische Dateien” \(p. 776\)](#).

Standardmäßig stellt der Proxyserver in einer Ruby-Umgebung alle Dateien im Ordner `public` auf dem Pfad `/public` und alle Dateien im Unterordner `public/assets` auf dem Pfad `/assets` bereit. Wenn beispielsweise die Anwendungsquelle eine Datei mit dem Namen `logo.png` im Ordner `public` enthält, wird diese vom Proxy-Server unter `subdomain.elasticbeanstalk.com/public/logo.png` an die Benutzer weitergeleitet. Wenn sich `logo.png` in einem Ordner mit dem Namen `assets` innerhalb des Ordners `public` befindet, stellt der Proxyserver ihn unter `subdomain.elasticbeanstalk.com/assets/logo.png` bereit. Sie können zusätzliche Zuweisungen konfigurieren, wie in diesem Abschnitt erklärt.

Umgebungseigenschaften

Im Bereich Environment Properties (Umgebungseigenschaften) können Sie die Einstellungen für die Umgebungskonfiguration der Amazon EC2-Instances angeben, auf denen die Anwendung ausgeführt wird. Umgebungseigenschaften werden als Schlüssel-Wert-Paare an die Anwendung weitergeleitet.

Die Ruby-Plattform definiert folgende Eigenschaften für die Umgebungskonfiguration:

- `BUNDLE_WITHOUT` – Eine durch Doppelpunkt getrennte Liste mit Gruppen, die bei der [Installation von Abhängigkeiten](#) aus einer `Gemfile`-Datei ignoriert werden sollen.
- `BUNDLER_DEPLOYMENT_MODE` – Auf `true` gesetzt (Standardeinstellung), um Abhängigkeiten im [Bereitstellungsmodus](#) mit Bundler zu installieren. Auf `false` gesetzt, um die `bundle install` im Entwicklungsmodus auszuführen.

Note

Diese Umgebungseigenschaft ist in den Versionen der Amazon Linux AMI-Ruby-Plattform (vor Amazon Linux 2) nicht definiert.

- `RAILS_SKIP_ASSET_COMPILATION` – Setzen Sie den Wert auf `true`, damit `rake assets:precompile` während der Bereitstellung nicht ausgeführt wird.
- `RAILS_SKIP.Migrations` – Setzen Sie den Wert auf `true`, damit `rake db:migrate` während der Bereitstellung nicht ausgeführt wird.
- `RACK_ENV` – Geben Sie die Umgebungsphase für Rack an. Beispiel: `development`, `production` oder `test`.

Innerhalb der Ruby-Umgebung, die in Elastic Beanstalk ausgeführt wird, können Sie über das Objekt `ENV` auf Umgebungsvariablen zugreifen. Beispielsweise können Sie mit folgendem Code eine Eigenschaft mit dem Namen `API_ENDPOINT` als Variable lesen:

```
endpoint = ENV['API_ENDPOINT']
```

Weitere Informationen finden Sie unter [Umgebungseigenschaften und andere Softwareeinstellungen \(p. 621\)](#).

Ruby-Konfigurations-Namespaces

Mithilfe einer [Konfigurationsdatei \(p. 722\)](#) können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Konfigurationsoptionen können durch den Elastic Beanstalk-Service oder die verwendete Plattform definiert und in Namespaces organisiert werden.

Sie können mithilfe des Namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` den Umgebungsproxy so konfigurieren, dass statische Dateien bereitgestellt werden. Sie definieren Zuordnungen von virtuellen Pfaden zu Anwendungsverzeichnissen.

Die Ruby-Plattform definiert keine plattformspezifischen Namespaces. Stattdessen werden Umgebungseigenschaften für häufig verwendete Rails- und Rack-Optionen definiert.

Die folgende Konfigurationsdatei gibt eine Option für statische Dateien an, mit der das Verzeichnis `staticimages` dem Pfad `/images` zuordnet wird und die einzelnen plattformdefinierten Umgebungseigenschaften und eine zusätzliche Umgebungseigenschaft mit dem Namen `LOGGING` festlegt werden.

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

Die Umgebungseigenschaft `BUNDLER_DEPLOYMENT_MODE` und der Namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` sind auf Amazon Linux AMI-Ruby-Plattformvarianten (vor Amazon Linux 2) nicht definiert.

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Installieren von Paketen mit einer Gemfile-Datei

Mit einer `Gemfile`-Datei im Stammverzeichnis der Projektquelle können Sie die von der Anwendung benötigten Pakete über RubyGems installieren.

Example `Gemfile`

```
source "https://rubygems.org"
gem 'sinatra'
gem 'json'
```

```
gem 'rack-parser'
```

Wenn eine `Gemfile`-Datei vorhanden ist, führt Elastic Beanstalk `bundle install` aus, um Abhängigkeiten zu installieren.

Konfigurieren der Anwendung mit einer Procfile-Datei

Um den Befehl anzugeben, mit dem Ihre Ruby-Anwendung gestartet wird, fügen Sie eine Datei namens `Procfile` in das Stammverzeichnis des Quell-Bundles ein.

Note

Elastic Beanstalk unterstützt diese Funktion nicht auf Amazon Linux AMI-Ruby-Plattformvarianten (vor Amazon Linux 2). Plattformzweige mit Namen, die mit Puma oder mit Passenger enthalten, haben unabhängig von ihrer Ruby-Version Vorrang vor Amazon Linux 2 und unterstützen die Funktion `Procfile` nicht.

Einzelheiten zum Schreiben und Verwenden eines `Procfile` finden Sie im Abschnitt `Buildfile` und `Procfile` unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#).

Wenn Sie keine `Procfile` angeben, generiert Elastic Beanstalk die folgende Standarddatei, die voraussetzt, dass Sie den vorinstallierten Puma-Anwendungsserver verwenden.

```
web: puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

Wenn Sie Ihren eigenen Puma-Server verwenden möchten, können Sie ihn mit einer [Gemfile \(p. 386\)](#) installieren. Die folgende Beispiel-`Procfile` zeigt, wie sie gestartet wird.

Example Procfile

```
web: bundle exec puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

Wenn Sie den Passenger-Anwendungsserver verwenden möchten, verwenden Sie die folgenden Beispieldateien, um Ihre Ruby-Umgebung für die Installation und Verwendung von Passenger zu konfigurieren.

1. Verwenden Sie diese Beispieldatei, um Passenger zu installieren.

Example Gemfile

```
source 'https://rubygems.org'  
gem 'passenger'
```

2. Verwenden Sie diese Beispieldatei, um Elastic Beanstalk anzuweisen, Passenger zu starten.

Example Procfile

```
web: bundle exec passenger start /var/app/current --socket /var/run/puma/my_app.sock
```

Note

Sie müssen in der Konfiguration des nginx-Proxy-Servers nichts ändern, um Passenger verwenden zu können. Um andere Anwendungsserver verwenden zu können, müssen Sie

möglicherweise die nginx-Konfiguration anpassen, um Anforderungen ordnungsgemäß an Ihre Anwendung weiterzuleiten.

Bereitstellen einer Rails-Anwendung in Elastic Beanstalk

Bei Rails handelt es sich um ein Open-Source-Framework mit Model-View-Controller (MVC) für Ruby. Dieses Tutorial erläutert den Prozess zum Generieren einer Rails-Anwendung und zum Bereitstellen dieser Anwendung in einer AWS Elastic Beanstalk-Umgebung.

Abschnitte

- [Voraussetzungen \(p. 388\)](#)
- [Starten einer Elastic Beanstalk-Umgebung \(p. 389\)](#)
- [Installieren von Rails und Generieren einer Website \(p. 390\)](#)
- [Konfigurieren von Rails-Einstellungen \(p. 392\)](#)
- [Bereitstellen Ihrer Anwendung \(p. 393\)](#)
- [Bereinigen \(p. 394\)](#)
- [Nächste Schritte \(p. 394\)](#)

Voraussetzungen

Grundkenntnisse zu Elastic Beanstalk

In diesem Tutorial wird davon ausgegangen, dass Sie über Kenntnisse über die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole verfügen. Sofern noch nicht geschehen, befolgen Sie die Anweisungen in [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#), um die erste Elastic Beanstalk-Umgebung zu starten.

Befehlszeile

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine -Shell für die Ausführung der Befehle. Befehle werden in Auflistungen mit einem vorangestellten Eingabeaufforderungs-Symbol (\$) und dem Namen des aktuellen Verzeichnisses angezeigt, falls angebracht.

```
~/eb-project$ this is a command  
this is output
```

Verwenden Sie auf Linux und macOS Ihren bevorzugten Shell- und Paket-Manager. Sie können unter Windows 10 das [Windows-Subsystem für Linux](#) installieren, um eine Windows-Version von Ubuntu und Bash zu erhalten.

Rails-Abhängigkeiten

Das Rails-Framework hat die folgenden Abhängigkeiten. Stellen Sie sicher, dass Sie alle installiert haben.

- Ruby 2.2.2 oder höher – Installationsanweisungen finden Sie unter [Einrichten Ihrer Ruby-Entwicklungsumgebung \(p. 381\)](#).

In diesem Tutorial verwenden wir Ruby 2.5.1 und die entsprechende Elastic Beanstalk-Plattformversion.

- Node.js – Anweisungen zur Installation finden Sie unter [Installieren von Node.js über den Package Manager](#).
- Yarn – Anweisungen zur Installation finden Sie unter [Installation](#) auf der Yarn-Website.

Starten einer Elastic Beanstalk-Umgebung

Verwenden Sie die Elastic Beanstalk-Konsole, um eine Elastic Beanstalk-Umgebung zu erstellen. Wählen Sie die Plattform Ruby und übernehmen Sie die Standardeinstellungen und den Beispielcode.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced.
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus, und wenn Sie bereit sind, wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2) zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer) – Ein Elastic Load Balancing-Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn sie beendet wird oder nicht verfügbar ist.
- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – Zwei CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form `subdomain.region.elasticbeanstalk.com` weiterleitet.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Note

Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#).

Installieren von Rails und Generieren einer Website

Installieren Sie Rails und seine Abhängigkeiten mit dem `gem`-Befehl.

```
~$ gem install rails
Fetching: concurrent-ruby-1.0.5.gem (100%)
Successfully installed concurrent-ruby-1.0.5
Fetching: i18n-1.0.1.gem (100%)
Successfully installed i18n-1.0.1
...
```

Testen Sie Ihre Rails-Installation.

```
~$ rails --version
Rails 5.2.0
```

Verwenden Sie `rails new` mit dem Namen der Anwendung, um ein neues Rails-Projekt zu erstellen.

```
~$ rails new ~/eb-rails
```

Rails erstellt ein Verzeichnis mit dem angegebenen Namen, generiert alle Dateien, die für eine lokale Ausführung des Beispielprojekts benötigt werden, und führt anschließend Bundler aus, um alle Abhängigkeiten ("Gems") zu installieren, die in der "Gemfile"-Datei des Projekts definiert sind.

Testen Sie die Rails-Installation, indem Sie das Standardprojekt lokal ausführen.

```
~$ cd eb-rails
eb-rails $ rails server
=> Booting Puma
=> Rails 5.2.0 application starting in development
=> Run `rails server -h` for more startup options
Puma starting in single mode...
* Version 3.11.4 (ruby 2.5.1-p57), codename: Love Song
* Min threads: 5, max threads: 5
* Environment: development
* Listening on tcp://0.0.0.0:3000
Use Ctrl+C to stop
...
```

Öffnen Sie <http://localhost:3000> in einem Webbrowser, um das Standardprojekt in Aktion zu sehen.



Yay! You're on Rails!



Diese Seite ist nur im Entwicklungsmodus sichtbar. Fügen Sie Inhalte auf der Titelseite der Anwendung hinzu, um die Produktionsbereitstellung in Elastic Beanstalk zu unterstützen. Erstellen Sie mit `rails generate` einen Controller, eine Route und eine Ansicht für die Willkommensseite.

```
~/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.
```

Dies gibt Ihnen alles, was Sie brauchen, um auf die Seite unter `/welcome_page/welcome` zuzugreifen. Bevor Sie die Änderungen veröffentlichen, sollten Sie die Inhalte der Ansicht ändern und eine Route hinzufügen, damit diese Seite als Top-Level-Objekt der Website angezeigt wird.

Ändern Sie die Inhalte in `app/views/welcome_page/welcome.html.erb` mit einem Texteditor. In diesem Beispiel werden die Inhalte der bestehenden Datei einfach mit `cat` überschrieben.

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>
```

Abschließend fügen Sie die folgende Route zu `config/routes.rb` hinzu:

Example `config/routes.rb`

```
Rails.application.routes.draw do
  get 'welcome_page/welcome'
  root 'welcome_page#welcome'
```

Damit wird Rails angewiesen, an den Website-Stamm gerichtete Anforderungen an die Willkommen-Methode des Controllers der Willkommenseite zu leiten, von der die Inhalte in der Willkommen-Ansicht dargestellt werden (`welcome.html.erb`).

Konfigurieren von Rails-Einstellungen

Verwenden Sie die Elastic Beanstalk-Konsole, um Rails mit Umgebungseigenschaften zu konfigurieren. Setzen Sie die `SECRET_KEY_BASE`-Umgebungseigenschaft auf eine Zeichenfolge mit bis zu 256 alphanumerischen Zeichen.

Rails verwendet diese Eigenschaft, um Schlüssel zu erstellen. Aus diesem Grund sollten Sie sie geheim halten und nicht als Klartext in der Quellsteuerung speichern. Übergeben Sie sie stattdessen an Rails-Code in Ihrer Umgebung über eine Umgebungseigenschaft.

So konfigurieren Sie Umgebungseigenschaften in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Geben Sie unter Environment properties (Umgebungseigenschaften) Schlüssel/Wert-Paare ein.

6. Wählen Sie Apply (Anwenden) aus.

Jetzt können Sie die Website in Ihrer Umgebung bereitstellen.

Bereitstellen Ihrer Anwendung

Erstellen Sie ein [Quell-Bundle \(p. 412\)](#) mit den Dateien, die von Rails erstellt wurden. Der folgende Befehl erstellt ein Quell-Bundle namens `rails-default.zip`.

```
~/eb-rails$ zip ..../rails-default.zip -r * .[^.]*
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um Rails in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Bereinigen

Wenn Sie nicht mehr mit Elastic Beanstalk arbeiten, können Sie Ihre Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 531\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) und dann Terminate environment (Umgebung beenden) aus.
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Nächste Schritte

Weitere Informationen über Rails finden Sie unter [rubyonrails.org](#).

Im weiteren Verlauf der Anwendungsentwicklung möchten Sie vermutlich Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die [Elastic Beanstalk Command Line Interface \(p. 1008\)](#) (EB CLI) stellt benutzerfreundliche Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk-Umgebungen über die Befehlszeile bereit.

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, [konfigurieren Sie einen benutzerdefinierten Domänennamen \(p. 644\)](#) für die Umgebung und [aktivieren Sie HTTPS \(p. 778\)](#) für sichere Verbindungen.

Bereitstellen einer Sinatra-Anwendung in Elastic Beanstalk

Diese schrittweise Anleitung zeigt, wie eine einfache [Sinatra](#)-Webanwendung in AWS Elastic Beanstalk erstellt wird.

Voraussetzungen

In diesem Tutorial wird davon ausgegangen, dass Sie über Kenntnisse über die grundlegenden Elastic Beanstalk-Operationen und die Elastic Beanstalk-Konsole verfügen. Sofern noch nicht geschehen, befolgen Sie die Anweisungen in [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#), um die erste Elastic Beanstalk-Umgebung zu starten.

Um die Verfahren in diesem Leitfaden zu befolgen, benötigen Sie ein Befehlszeilen-Terminal oder eine -Shell für die Ausführung der Befehle. Befehle werden in Auflistungen mit einem vorangestellten Eingabeaufforderungs-Symbol (\$) und dem Namen des aktuellen Verzeichnisses angezeigt, falls angebracht.

```
~/eb-project$ this is a command
this is output
```

Verwenden Sie auf Linux und macOS Ihren bevorzugten Shell- und Paket-Manager. Sie können unter Windows 10 das [Windows-Subsystem für Linux](#) installieren, um eine Windows-Version von Ubuntu und Bash zu erhalten.

Sinatra erfordert Ruby 1.9 oder höher. In diesem Tutorial verwenden wir Ruby 2.5.1 und die entsprechende Elastic Beanstalk-Plattformversion. Installieren Sie Ruby mithilfe der Anleitungen unter [Einrichten Ihrer Ruby-Entwicklungsumgebung \(p. 381\)](#).

Starten einer Elastic Beanstalk-Umgebung

Verwenden Sie die Elastic Beanstalk-Konsole, um eine Elastic Beanstalk-Umgebung zu erstellen. Wählen Sie die Plattform Ruby und übernehmen Sie die Standardeinstellungen und den Beispielcode.

So starten Sie eine Umgebung (Konsole)

1. Öffnen Sie die Elastic Beanstalk-Konsole mit diesem vorkonfigurierten Link: console.aws.amazon.com/elasticbeanstalk/home#/newApplication?applicationName=tutorials&environmentType=LoadBalanced.
2. Wählen Sie unter Platform (Plattform) die Plattform und die Plattformvariante aus, die mit der Sprache Ihrer Anwendung übereinstimmen.
3. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
4. Klicken Sie auf Review and launch (Überprüfen und starten).
5. Sehen Sie sich die verfügbaren Optionen an. Wählen Sie die gewünschte verfügbare Option aus, und wenn Sie bereit sind, wählen Sie Create App (App erstellen) aus.

Das Erstellen der Umgebung benötigt etwa fünf Minuten. Dabei werden die folgenden Ressourcen erstellt:

- EC2 instance (EC2-Instance) – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2) zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattform.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance security group (Instance-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load balancer (Load Balancer) – Ein Elastic Load Balancing-Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht mehr direkt für das Internet zugänglich gemacht werden.
- Load balancer security group (Load Balancer-Sicherheitsgruppe) – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling group (Auto Scaling-Gruppe) – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn sie beendet wird oder nicht verfügbar ist.
- Amazon S3 bucket (Amazon S3-Bucket) – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.

- Amazon CloudWatch alarms (Amazon CloudWatch-Alarme) – Zwei CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Wenn ein Alarm ausgelöst wird, wird Ihre Auto Scaling-Gruppe nach oben oder nach unten skaliert.
- AWS CloudFormation stack (AWS CloudFormation-Stack) – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domain name (Domänenname) – Ein Domänenname, der auf Ihre Webanwendung in der Form `subdomain.region.elasticbeanstalk.com` weiterleitet.

All diese Ressourcen werden von Elastic Beanstalk verwaltet. Wenn Sie die Umgebung beenden, werden alle darin enthaltenen Ressourcen von Elastic Beanstalk beendet.

Note

Der von Elastic Beanstalk erstellte Amazon S3-Bucket wird von den Umgebungen gemeinsam genutzt und beim Beenden einer Umgebung nicht gelöscht. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#).

Schreiben einer einfachen Sinatra-Website

So können Sie eine Sinatra-Anwendung erstellen und bereitstellen:

1. Erstellen Sie eine Konfigurationsdatei mit dem Namen config.ru und folgendem Inhalt.

Example config.ru

```
require './helloworld'  
run Sinatra::Application
```

2. Erstellen Sie eine Ruby-Codedatei mit dem Namen helloworld.rb und folgendem Inhalt.

Example helloworld.rb

```
require 'sinatra'  
get '/' do  
  "Hello World!"  
end
```

3. Erstellen Sie eine Gemfile-Datei mit folgendem Inhalt.

Example Gemfile

```
source 'https://rubygems.org'  
gem 'sinatra'
```

Bereitstellen Ihrer Anwendung

Erstellen Sie ein [Quell-Bundle \(p. 412\)](#) mit Ihren Quelldateien. Der folgende Befehl erstellt ein Quell-Bundle namens `sinatra-default.zip`.

```
~/eb-sinatra$ zip ../sinatra-default.zip -r *.[^.]*
```

Laden Sie das Quell-Bundle nach Elastic Beanstalk hoch, um Sinatra in der Umgebung bereitzustellen.

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Bereinigen

Wenn Sie nicht mehr mit Elastic Beanstalk arbeiten, können Sie Ihre Umgebung beenden. Elastic Beanstalk beendet alle AWS-Ressourcen der Umgebung, z. B. [Amazon EC2-Instances \(p. 531\)](#), [Datenbank-Instances \(p. 609\)](#), [Load Balancer \(p. 554\)](#), Sicherheitsgruppen und [Alarne \(p. 555\)](#).

So beenden Sie die Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) und dann Terminate environment (Umgebung beenden) aus.
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Mit Elastic Beanstalk können Sie ganz einfach und jederzeit eine neue Umgebung für Ihre Anwendung erstellen.

Nächste Schritte

Mehr über Sinatra erfahren Sie unter sinatrarb.com.

Im weiteren Verlauf der Anwendungsentwicklung möchten Sie vermutlich Umgebungen verwalten und Anwendungen bereitstellen, ohne manuell eine ZIP-Datei zu erstellen und diese in die Elastic Beanstalk-Konsole hochzuladen. Die [Elastic Beanstalk Command Line Interface \(p. 1008\)](#) (EB CLI) stellt benutzerfreundliche Befehle zum Erstellen, Konfigurieren und Bereitstellen von Anwendungen in Elastic Beanstalk-Umgebungen über die Befehlszeile bereit.

Sofern Sie planen, die Anwendung in einer Produktionsumgebung einzusetzen, [konfigurieren Sie einen benutzerdefinierten Domänennamen \(p. 644\)](#) für die Umgebung und [aktivieren Sie HTTPS \(p. 778\)](#) für sichere Verbindungen.

Hinzufügen einer Amazon RDS-DB-Instance zur Ruby-Anwendungsumgebung

Sie können eine DB-Instance von Amazon Relational Database Service (Amazon RDS) verwenden, um Daten zu speichern, die von Ihrer Anwendung gesammelt und geändert wurden. Die Datenbank kann Ihrer Umgebung angefügt und von Elastic Beanstalk verwaltet werden oder sie kann extern erstellt und verwaltet werden.

Bei der ersten Verwendung von Amazon RDS [fügen Sie eine DB-Instance \(p. 398\)](#) zu einer Testumgebung über die Elastic Beanstalk Management Console hinzu und prüfen, ob die Anwendung eine Verbindung zu dieser herstellen kann.

Für die Verbindungserstellung zu einer Datenbank [fügen Sie den Adapter \(p. 399\)](#) zur Anwendung hinzu und [konfigurieren eine Verbindung \(p. 399\)](#) mit den von Elastic Beanstalk bereitgestellten Umgebungseigenschaften. Die Konfiguration und der Verbindungscode hängen von der Datenbank-Engine und dem Framework ab, die Sie nutzen.

Note

Zu Lernzwecken oder für Testumgebungen können Sie mit Elastic Beanstalk eine DB-Instance hinzufügen.

Für Produktionsumgebungen können Sie eine DB-Instance außerhalb Ihrer Elastic Beanstalk-Umgebung erstellen, um Ihre Umgebungsressourcen von Ihren Datenbankressourcen zu entkoppeln. Auf diese Weise wird beim Beenden Ihrer Umgebung die DB-Instance nicht gelöscht. Mit einer externen DB-Instance können Sie von mehreren Umgebungen aus eine Verbindung mit derselben Datenbank herstellen und [Blau/Grün-Bereitstellungen](#) durchführen. Anleitungen hierzu finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon RDS \(p. 973\)](#).

Abschnitte

- [Hinzufügen einer DB-Instance zu Ihrer Umgebung \(p. 398\)](#)
- [Herunterladen eines Adapters \(p. 399\)](#)
- [Verbinden mit einer Datenbank \(p. 399\)](#)

Hinzufügen einer DB-Instance zu Ihrer Umgebung

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie Apply (Anwenden) aus.

Das Hinzufügen einer DB-Instance dauert ca. 10 Minuten. Sobald die Umgebungsaktualisierung abgeschlossen ist, stehen der Hostname der DB-Instance und andere Verbindungsinformationen über die folgenden Umgebungseigenschaften zur Verfügung:

Eigenschaftenname	Beschreibung	Eigenschaftswert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbankname, ebdb	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: DB Name (DB-Name).
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	In der Amazon RDS-Konsole nicht als Referenz verfügbar.

Weitere Informationen zum Konfigurieren einer internen DB-Instance finden Sie unter [Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung \(p. 609\)](#).

Herunterladen eines Adapters

Fügen Sie den Datenbankadapter zur [gem-Datei \(p. 386\)](#) des Projekts hinzu.

Example Gemfile – Rails mit MySQL

```
source 'https://rubygems.org'
gem 'puma'
gem 'rails', '4.1.8'
gem 'mysql2'
```

Gängige Adapter-Gems für Ruby

- MySQL – mysql2
- PostgreSQL – pg
- Oracle – activerecord-oracle_enhanced-adapter
- SQL Server – sql_server

Verbinden mit einer Datenbank

Elastic Beanstalk stellt Verbindungsinformationen für zugehörige DB-Instances in Umgebungseigenschaften bereit. Mit `ENV['VARIABLE']` können Sie die Eigenschaften lesen und eine Datenbankverbindung konfigurieren.

Example config/database.yml – Ruby on Rails-Datenbankkonfiguration (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'] %>
```

Tutorials und Beispiele

Framework- und sprachspezifische Tutorials finden Sie überall im AWS Elastic Beanstalk-Entwicklerhandbuch. Neue und aktualisierte Tutorials werden dieser Liste nach ihrer Veröffentlichung hinzugefügt. Die neuesten Aktualisierungen werden zuerst angezeigt.

Diese Tutorials richten sich an fortgeschrittene Benutzer und enthalten keine Anweisungen für grundlegende Schritte, wie z. B. die Anmeldung bei AWS. Wenn Sie AWS oder Elastic Beanstalk zum ersten Mal verwenden, können Sie Ihre erste Elastic Beanstalk-Umgebung mit der [Anleitung für die ersten Schritte \(p. 3\)](#) einrichten und ausführen.

- Ruby on Rails – [Bereitstellen einer Rails-Anwendung in Elastic Beanstalk \(p. 388\)](#)
- Ruby und Sinatra – [Bereitstellen einer Sinatra-Anwendung in Elastic Beanstalk \(p. 394\)](#)
- PHP und MySQL HA-Konfiguration – [Bereitstellen einer PHP-Anwendung mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank zu Elastic Beanstalk \(p. 313\)](#)
- PHP und Laravel – [Bereitstellen einer Laravel-Anwendung in Elastic Beanstalk \(p. 293\)](#)
- PHP und CakePHP – [Bereitstellen einer CakePHP-Anwendung in Elastic Beanstalk \(p. 301\)](#)
- PHP und Drupal HA-Konfiguration – [Bereitstellen einer Drupal-Website mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank zu Elastic Beanstalk \(p. 336\)](#)
- PHP und WordPress HA-Konfiguration – [Bereitstellen einer WordPress-Website mit hoher Verfügbarkeit und einer externen Amazon RDS-Datenbank zu Elastic Beanstalk \(p. 322\)](#)
- Node.js mit DynamoDB HA-Konfiguration – [Bereitstellen einer Node.js-Anwendung mit DynamoDB in Elastic Beanstalk \(p. 273\)](#)
- ASP.NET Core – [Tutorial: Bereitstellen einer ASP.NET Core-Anwendung mit Elastic Beanstalk \(p. 208\)](#)
- Python und Flask – [Bereitstellen einer Flask-Anwendung in Elastic Beanstalk \(p. 361\)](#)
- Python und Django – [Bereitstellen einer Django-Anwendung in Elastic Beanstalk \(p. 367\)](#)
- Node.js und Express – [Bereitstellen einer Express-Anwendung für Elastic Beanstalk \(p. 260\)](#)
- Docker, PHP und nginx – [Multicontainer Docker-Umgebungen mit der Elastic Beanstalk-Konsole \(p. 73\)](#)
- .NET Framework (IIS und ASP.NET) – [Tutorial: Erstellen und Bereitstellen einer .NET-Beispielanwendung mit Elastic Beanstalk. \(p. 201\)](#)

Unter den nachfolgenden Links können Sie die von Elastic Beanstalk verwendeten Beispielanwendungen herunterladen, um eine Umgebung ohne Quell-Bundle zu erstellen:

- Docker – [docker.zip](#)
- Multicontainer-Docker – [docker-multicontainer-v2.zip](#)
- Vorkonfigurierter Docker (Glassfish) – [docker-glassfish-v1.zip](#)
- Go – [go.zip](#)
- Corretto – [corretto.zip](#)
- Tomcat – [tomcat.zip](#)
- .NET Core unter Linux – [dotnet-core-linux.zip](#)
- .NET – [dotnet-asp-v1.zip](#)
- Node.js – [nodejs.zip](#)
- PHP – [php.zip](#)
- Python – [python.zip](#)
- Ruby – [ruby.zip](#)

Weitere Beispieldokumente, mit denen die Nutzung zusätzlicher Web-Frameworks, Bibliotheken und Tools veranschaulicht wird, stehen als Open Source-Projekte auf GitHub zur Verfügung:

- [Load-balanced WordPress \(Tutorial \(p. 322\)\)](#) – Konfigurationsdateien für die sichere Installation und Ausführung von WordPress in einer AWS Elastic Beanstalk-Umgebung mit Load Balancing.
- [Load-balanced Drupal \(Tutorial \(p. 336\)\)](#) – Konfigurationsdateien und Anweisungen für die sichere Installation und Ausführung von Drupal in einer AWS Elastic Beanstalk-Umgebung mit Load Balancing.
- [Scorekeep](#) – RESTful-Web-API, die das Spring-Framework und das AWS SDK for Java verwendet, um eine Schnittstelle zum Erstellen und Verwalten von Benutzern, Sitzungen und Spielen bereitzustellen. Die API ist mit einer Angular 1.5-Web-App gebündelt, von der die API über HTTP eingesetzt wird. Umfasst Verzweigungen, die die Integration mit Amazon Cognito, AWS X-Ray und Amazon Relational Database Service zeigen.

Die Anwendung nutzt Funktionen der Java SE-Plattform, um Abhängigkeiten herunterzuladen und auf Instances zu erstellen, und minimiert so die Größe des Quell-Bundles. Die Anwendung umfasst auch nginx-Konfigurationsdateien, mit denen die Standardkonfiguration überschrieben wird, sodass die Bereitstellung der Frontend-Webanwendung statisch auf Port 80 über den Proxy erfolgt und Anforderungen über Pfade unter /api an die auf localhost:5000 ausgeführte API weitergeleitet werden.

- [Does it Have Snakes?](#) – Tomcat-Anwendung, mit der die Verwendung von RDS in einer Java EE-Webanwendung in Elastic Beanstalk veranschaulicht wird. Das Projekt zeigt die Nutzung von Servlets, JSPs, Simple Tag Support, Tag Files, JDBC, SQL, Log4J, Bootstrap, Jackson und Elastic Beanstalk-Konfigurationsdateien.
- [Locust Load Generator](#) – Dieses Projekt zeigt die Verwendung von Java SE-Plattformfunktionen zur Installation und Ausführung von [Locust](#). Dieses Tool für die Lastgenerierung ist in Python geschrieben. Das Projekt umfasst Konfigurationsdateien zur Installation und Konfiguration von Locust, ein Build-Skript zur Konfiguration einer DynamoDB-Tabelle sowie eine Procfile-Datei für die Locust-Ausführung.
- [Share Your Thoughts \(Tutorial \(p. 313\)\)](#) – PHP-Anwendung, mit der die Nutzung von MySQL auf Amazon RDS, Composer und Konfigurationsdateien gezeigt wird.
- [A New Startup \(Tutorial \(p. 273\)\)](#) – Node.js-Beispieldokumentation, mit der die Verwendung von DynamoDB, dem AWS SDK for JavaScript in Node.js, npm Package Management und Konfigurationsdateien gezeigt wird.

Verwalten und Konfigurieren von Elastic Beanstalk-Anwendungen

Der erste Schritt mit AWS Elastic Beanstalk besteht darin, eine Anwendung zu erstellen, die Ihre Webanwendung in AWS darstellt. In Elastic Beanstalk dient eine Anwendung als Container für die Umgebungen, in denen die Web-App ausgeführt wird, sowie für Quellcodeversionen der Web-App, gespeicherte Konfigurationen, Protokolle und andere Artefakte, die Sie in Elastic Beanstalk erstellen.

So erstellen Sie eine Anwendung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Applications (Anwendungen) und anschließend Create a new application (Neue Anwendung erstellen).
3. Verwenden Sie das Formular auf dem Bildschirm, um einen Anwendungsnamen anzugeben.
4. Geben Sie optional eine Beschreibung ein und fügen Sie Tag-Schlüssel und -Werte hinzu.
5. Wählen Sie Create (Erstellen) aus.

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 a resource and is case-sensitive. [Learn more](#)

Key	Value	Remove
<input type="text"/>	<input type="text"/>	<input type="button" value="Remove"/>

Add tag

50 remaining

Ca

The screenshot shows the 'Create new application' wizard in the AWS Elastic Beanstalk console. In the 'Application information' section, there is a field for 'Application Name' with a note about character limits and forward slashes. Below it is a larger 'Description' field. In the 'Tags' section, users can apply up to 50 tags, each consisting of a 'Key' and a 'Value'. There is a 'Remove' button for each tag entry. An 'Add tag' button is available to add more entries. A note indicates that 50 tags remain available. The interface is clean with a light gray background and white input fields.

Nachdem die Anwendung generiert ist, werden Sie von der Konsole aufgefordert, eine Umgebung dafür zu erstellen. Detaillierte Informationen zu allen verfügbaren Optionen finden Sie unter [Erstellen einer Elastic Beanstalk-Umgebung \(p. 435\)](#).

Wenn Sie eine Anwendung nicht mehr benötigen, können Sie sie löschen.

Warning

Wenn Sie eine Anwendung löschen, werden alle zugeordneten Umgebungen und sämtliche Anwendungsversionen sowie gespeicherte Konfigurationen für die Anwendung gelöscht.

So löschen Sie eine Anwendung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Applications (Anwendungen) und wählen Sie dann Ihre Anwendung in der Liste aus.
3. Wählen Sie Actions (Aktionen) und anschließend Delete Application (Anwendung löschen).

Themen

- [Elastic Beanstalk-Anwendungsmanagement-Konsole \(p. 405\)](#)
- [Verwalten von Anwendungsversionen \(p. 406\)](#)
- [Erstellen des Quell-Bundles einer Anwendung. \(p. 412\)](#)
- [Markieren von Elastic Beanstalk-Anwendungsressourcen \(p. 420\)](#)

Elastic Beanstalk-Anwendungsmanagement-Konsole

Sie können die AWS Elastic Beanstalk-Konsole zur Verwaltung von Anwendungen, Anwendungsversionen und gespeicherten Konfigurationen verwenden.

So greifen Sie auf die Konsole zur Anwendungsverwaltung zu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.

Note

Wenn Sie viele Anwendungen haben, verwenden Sie die Suchleiste, um die Anwendungsliste zu filtern.

Die Anwendungsübersichtsseite zeigt eine Liste mit einer Übersicht über alle mit der Anwendung verknüpften Umgebungen.

Environment name	Health	Date created	Last modified	URL	Platform
GettingStartedApp-env	OK	2020-01-28 12:06:50 UTC-0800	2020-01-30 15:02:35 UTC-0800	GettingStartedApp-env.bn7dx222kw.us-east-2.elasticbeanstalk.com	Tomcat running Linux
GettingStartedApp-Windows	OK	2020-01-28 16:34:29 UTC-0800	2020-01-28 16:38:20 UTC-0800	GettingStartedApp-Windows.bn7dx222kw.us-east-2.elasticbeanstalk.com	IIS 10.0 Windows

3. Sie haben verschiedene Möglichkeiten, um fortzufahren:

- a. Wählen Sie das Drop-down-Menü Actions (Aktionen) und dann eine der Aktionen zur Anwendungsverwaltung. Um eine Umgebung in dieser Anwendung zu starten, können Sie direkt

Create a new environment (Neue Umgebung erstellen) wählen. Details dazu finden Sie unter [the section called “Erstellen von Umgebungen” \(p. 435\)](#).

- b. Wählen Sie einen Umgebungsnamen, um zur [Konsole zur Umgebungsverwaltung \(p. 426\)](#) für diese Umgebung zu wechseln, in der Sie die Umgebung konfigurieren, überwachen oder verwalten können.
- c. Wählen Sie im Navigationsbereich Application versions (Anwendungsversionen) hinter dem Anwendungsnamen, um die Anwendungsversionen für Ihre Anwendung anzuzeigen und zu verwalten.

Eine Anwendungsversion ist eine hochgeladene Version Ihres Anwendungscodes. Sie können neue Versionen hochladen, eine vorhandene Version auf einer beliebigen Umgebung der Anwendung bereitstellen oder alte Versionen löschen. Weitere Informationen finden Sie unter [Verwalten von Anwendungsversionen \(p. 406\)](#).

- d. Wählen Sie Saved configurations (Gespeicherte Konfigurationen) hinter dem Anwendungsnamen im Navigationsbereich, um Konfigurationen anzuzeigen und zu verwalten, die in ausgeführten Umgebungen gespeichert wurden.

Eine gespeicherte Konfiguration ist eine Sammlung von Einstellungen, die Sie verwenden können, um die Einstellungen einer Umgebung in einem vorherigen Status wiederherzustellen oder eine Umgebung mit den gleichen Einstellungen zu erstellen. Weitere Informationen finden Sie unter [Verwenden gespeicherter Elastic Beanstalk-Konfigurationen \(p. 764\)](#).

Verwalten von Anwendungsversionen

Elastic Beanstalk erstellt eine Anwendungsversion, wenn Sie Quellcode hochladen. Dies erfolgt in der Regel beim Erstellen einer Umgebung oder dem Hochladen und Bereitstellen von Code mit der [Environment Management Console \(p. 426\)](#) oder [EB CLI \(p. 1008\)](#). Elastic Beanstalk löscht diese Anwendungsversionen gemäß Lebenszyklus-Richtlinie der Anwendung, und wenn Sie die Anwendung löschen. Details zu den Lebenszyklusrichtlinien von Anwendungen finden Sie unter [Konfigurieren von Einstellungen für den Lebenszyklus der Anwendungsversion \(p. 408\)](#).

Sie können auch einen Quell-Bundle ohne Bereitstellung von der [Konsole zur Anwendungsverwaltung \(p. 405\)](#) hochladen. Elastic Beanstalk speichert Quell-Bundles in Amazon Simple Storage Service (Amazon S3) und löscht sie nicht automatisch.

Sie können eine Anwendungsversion mit Tags markieren, wenn Sie sie erstellen, und Tags von vorhandenen Anwendungsversionen bearbeiten. Details dazu finden Sie unter [Markieren von Anwendungsversionen \(p. 410\)](#).

So erstellen Sie eine neue Anwendungsversion

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.

Note

Wenn Sie viele Anwendungen haben, verwenden Sie die Suchleiste, um die Anwendungsliste zu filtern.

3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Application versions (Anwendungsversionen) aus.
4. Klicken Sie auf Upload. Verwenden Sie das Formular auf dem Bildschirm, um das [Quellpaket \(p. 412\)](#) Ihrer Anwendung hochzuladen.

Note

Die maximale Dateigröße des Quell-Bundles beträgt 512 MB.

5. Geben Sie optional eine kurze Beschreibung ein und fügen Sie Tag-Schlüssel und -Werte hinzu.
6. Klicken Sie auf Upload (Hochladen).

Die angegebene Datei ist mit Ihrer Anwendung verbunden. Sie können die Anwendungsversion in einer neuen oder vorhandenen Umgebung bereitstellen.

Im Lauf der Zeit kann Ihre Anwendung viele Anwendungsversionen ansammeln. Um Speicherplatz zu sparen und das [Kontingent für die Anwendungsversion](#) nicht auszureißen, sollten Sie nicht mehr benötigte Anwendungsversionen löschen.

Note

Das Löschen einer Anwendungsversion hat keine Auswirkungen auf aktuell ausgeführte Umgebungen dieser Version.

So löschen Sie eine Anwendungsversion

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.

Note

Wenn Sie viele Anwendungen haben, verwenden Sie die Suchleiste, um die Anwendungsliste zu filtern.

3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Application versions (Anwendungsversionen) aus.
4. Wählen Sie eine oder mehrere Anwendungsversionen aus, die Sie löschen möchten.

The screenshot shows the 'Application versions' section of the AWS Elastic Beanstalk console. At the top, the navigation path is: Elastic Beanstalk > Applications > getting-started-app > Application versions. Below this, a table lists five application versions:

Version label	Description	Date created	Source
Sample Application-3		2020-01-30T15:02:09-08:00	2020030W...v-java-tomcat-v3.zip
<input checked="" type="checkbox"/> Sample Application-2		2020-01-28T15:16:47-08:00	2020028J18-java-tomcat-v3.zip
<input checked="" type="checkbox"/> Sample Application-1		2019-12-06T14:56:36-08:00	2019340FRL-eb-demo-php-simple-app-v1.3.zip
Sample Application		2019-11-01T15:12:51-07:00	Sample Application

Two rows are highlighted with a yellow box and have a checked checkbox in the first column. In the top right corner of the table header, there is a 'Actions' button with a dropdown menu. The 'Delete' option in this menu is highlighted with a yellow box and has a hand cursor icon pointing at it. The full URL for the screenshot is: https://docs.aws.amazon.com/de-de/elasticbeanstalk/latest/dg/versions.html#delete-version

5. Wählen Sie Actions (Aktionen) und dann Delete (Löschen).
6. (Optional) Um das Programm-Quellbündel für diese Anwendungsversionen in Ihrem Amazon Simple Storage Service (Amazon S3)-Bucket zu belassen, deaktivieren Sie die option Delete versions from Amazon S3 (Versionen von Amazon S3 löschen).

Delete Application Versions

The following application versions will be deleted:

- Sample Application-1
- Sample Application-2

Delete versions from Amazon S3

7. Wählen Sie Delete.

Sie können Elastic Beanstalk auch so konfigurieren, dass alte Versionen automatisch gelöscht werden, indem Sie die Lebenszykluseinstellungen der Anwendungsversion konfigurieren. Wenn Sie diese Lebenszykluseinstellungen konfigurieren, werden sie angewendet, wenn Sie neue Anwendungsversionen erstellen. Wenn Sie maximal 25 Anwendungsversionen konfigurieren, löscht Elastic Beanstalk die älteste Version, wenn Sie eine 26. Version hochladen. Wenn Sie ein Höchstalter von 90 Tagen festlegen, werden alle Versionen, die älter sind als 90 Tage alt, gelöscht, sobald Sie eine neue Version hochladen. Details dazu finden Sie unter [the section called "Lebenszyklus der Version" \(p. 408\)](#).

Wenn Sie sich nicht dafür entscheiden, das Quell-Bundle von Amazon S3 zu löschen, löscht Elastic Beanstalk trotzdem die Version aus seinen Datensätzen. Das Quell-Bundle verbleibt jedoch in Ihrem [Elastic Beanstalk Storage-Bucket \(p. 985\)](#). Das Kontingent für die Anwendungsversion gilt nur für die Elastic Beanstalk-Versionen. Daher können Sie Versionen löschen, um das Kontingent nicht zu überschreiten und um alle Quell-Bundles in Amazon S3 zu erhalten.

Note

Das Anwendungsversionslimit gilt nicht für Quell-Bundles, aber es können Ihnen dennoch Amazon S3-Gebühren entstehen, und sie behalten möglicherweise personenbezogene Daten länger als notwendig. Elastic Beanstalk löscht Quell-Bundles niemals automatisch. Sie sollten Quell-Bundles löschen, wenn Sie sie nicht mehr benötigen.

Konfigurieren von Einstellungen für den Lebenszyklus der Anwendungsversion

Jedes Mal, wenn Sie eine neue Version Ihrer Anwendung mit der Elastic Beanstalk-Konsole oder der EB CLI hochladen, erstellt Elastic Beanstalk eine [Anwendungsversion \(p. 406\)](#). Falls Sie nicht länger benötigte Versionen nicht löschen, erreichen Sie irgendwann das [Anwendungsversionskontingent](#) und können keine neuen Versionen dieser Anwendung mehr erstellen.

Das Erreichen dieses Kontingents können Sie vermeiden, indem Sie den Anwendungen eine Lebenszyklusrichtlinie für die Anwendungsversion hinzufügen. Mit einer Lebenszyklus-Richtlinie wird Elastic Beanstalk angewiesen, Anwendungsversionen mit einem bestimmten Alter oder bei Überschreitung eines vorgegebenen Maximalwerts an Versionen zu löschen.

Elastic Beanstalk wendet die Lebenszyklus-Richtlinie der Anwendung bei jeder neu erstellten Anwendungsversion an und löscht dabei jedes Mal bis zu 100 Versionen. Elastic Beanstalk löscht alte Versionen nach dem Erstellen der neuen Version. Die neue Version wird nicht hinsichtlich der maximalen Anzahl an Versionen berücksichtigt, die Sie in der Richtlinie festgelegt haben.

Elastic Beanstalk löscht weder Anwendungsversionen, die derzeit in einer Umgebung ausgeführt werden, noch solche, die in Umgebungen bereitgestellt wurden, die weniger als zehn Wochen vor dem Inkrafttreten der Richtlinie beendet wurden.

Das Anwendungsversionskontingent gilt für alle Anwendungen in einer Region. Bei mehreren Anwendungen sollten Sie für jede eine angemessene Lebenszyklus-Richtlinie konfigurieren, damit das Kontingent nicht erreicht wird. Wenn Sie z. B. 10 Anwendungen in einer Region haben und das Kontingent 1.000 Anwendungsversionen beträgt, sollten Sie in Erwägung ziehen, eine Lebenszyklusrichtlinie mit einem Kontingent von 99 Anwendungsversionen für alle Anwendungen festzulegen oder andere Werte in jeder Anwendung festzulegen, solange die Summe weniger als 1.000 Anwendungsversionen beträgt. Elastic Beanstalk wendet die Richtlinie nur an, sofern die Anwendungsversion erfolgreich erstellt wurde. Falls das Limit also bereits erreicht ist, müssen Sie einige Versionen manuell löschen, bevor Sie eine neue Version erstellen können.

Standardmäßig belässt Elastic Beanstalk das [Quell-Bundle \(p. 412\)](#) der Anwendungsversion in Amazon S3, um Datenverlust zu verhindern. Sie können das Quell-Bundle löschen, um Speicherplatz freizusetzen.

Sie können die Einstellungen für den Lebenszyklus über die Elastic Beanstalk CLI und APIs festlegen. Weitere Informationen finden Sie unter [eb appversion \(p. 1044\)](#), [CreateApplication](#) (mit dem Parameter `ResourceLifecycleConfig`) und [UpdateApplicationResourceLifecycle](#).

Festlegen der Einstellungen für den Anwendungslebenszyklus in der Konsole

Sie können die Lebenszyklus-Einstellungen in der Elastic Beanstalk-Konsole festlegen.

So legen Sie die Einstellungen für den Anwendungslebenszyklus fest

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.

Note

Wenn Sie viele Anwendungen haben, verwenden Sie die Suchleiste, um die Anwendungsliste zu filtern.

3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Application versions (Anwendungsversionen) aus.
4. Wählen Sie Settings aus.
5. Verwenden Sie das Formular auf dem Bildschirm, um Einstellungen für den Anwendungslebenszyklus zu konfigurieren.
6. Wählen Sie Save (Speichern) aus.

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

Cancel **Save**

Auf der Einstellungsseite haben Sie folgende Möglichkeiten.

- Konfigurieren Sie Lebenszykluseinstellungen basierend auf der Gesamtzahl der Anwendungsversionen oder dem Alter der Anwendungsversionen.
- Legen Sie fest, ob das Quell-Bundle in S3 beim Löschen einer Anwendungsversion ebenfalls gelöscht werden soll.
- Wählen Sie die Servicerolle aus, unter der die Anwendungsversion gelöscht wird. Um alle erforderlichen Berechtigungen zum Löschen der Version aufzunehmen, wählen Sie die standardmäßige Elastic Beanstalk-Servicerolle, `aws-elasticbeanstalk-service-role`, oder eine andere Servicerolle unter Verwendung der von Elastic Beanstalk verwalteten Service-Richtlinien. Weitere Informationen finden Sie unter [Rollen von Elastic Beanstalk Service verwalten \(p. 909\)](#).

Markieren von Anwendungsversionen

Sie können Tags auf Ihre AWS Elastic Beanstalk-Anwendungsversionen anwenden. Tags sind mit AWS-Ressourcen verknüpfte Schlüssel-Wert-Paare. Weitere Informationen zum Elastic Beanstalk-

Ressourcen-Tagging, Anwendungsfälle, Einschränkungen für Tag-Schlüssel und -Werte sowie unterstützte Ressourcentypen finden Sie unter [Markieren von Elastic Beanstalk-Anwendungsressourcen \(p. 420\)](#).

Sie können Tags angeben, wenn Sie eine Anwendung erstellen. In einer vorhandenen Anwendungsversion können Sie Tags hinzufügen oder entfernen sowie die Werte von vorhandenen Tags aktualisieren. Sie können jeder Anwendungsversion bis zu 50 Tags hinzufügen.

Hinzufügen von Tags während der Anwendungsversionserstellung

Wenn Sie über die Elastic Beanstalk-Konsole [eine Umgebung erstellen \(p. 437\)](#) und eine Version des Anwendungscodes hochladen, können Sie Tag-Schlüssel und -Werte angeben, um sie der neuen Anwendungsversion zuzuordnen.

Die Elastic Beanstalk-Konsole können Sie auch verwenden, um [eine Anwendungsversion hochzuladen, \(p. 406\)](#) ohne sie sofort in einer Umgebung zu verwenden. Sie können Tag-Schlüssel und -Werte angeben, wenn Sie eine Anwendungsversion hochladen.

Fügen Sie Tags mit der AWS CLI oder anderen API-basierten Clients hinzu, indem Sie den `--tags`-Parameter mit dem Befehl `create-application-version` verwenden.

```
$ aws elasticbeanstalk create-application-version \
  --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
  --application-name my-app --version-label v1
```

Wenn Sie die EB CLI zum Erstellen oder Aktualisieren einer Umgebung verwenden, wird eine Anwendungsversion aus dem Code erstellt, den Sie bereitstellen. Es gibt keine direkte Methode, um eine Anwendungsversion bei der Erstellung über die EB CLI mit Tags zu markieren. Weitere Informationen zum Hinzufügen von Tags zu einer vorhandenen Anwendungsversion finden Sie im folgenden Abschnitt.

Verwalten von Tags einer vorhandenen Anwendungsversion

Sie können Tags in einer vorhandenen Elastic Beanstalk-Anwendungsversion hinzufügen, aktualisieren und löschen.

So verwalten Sie die Tags einer Anwendungsversion mit der Elastic Beanstalk-Konsole:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.

Note

Wenn Sie viele Anwendungen haben, verwenden Sie die Suchleiste, um die Anwendungsliste zu filtern.

3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Application versions (Anwendungsversionen) aus.
4. Wählen Sie die Anwendungsversion aus, die Sie verwalten möchten.
5. Klicken Sie auf Actions (Aktionen) und dann auf Manage Tags (Tags verwalten).
6. Verwenden Sie das Formular auf dem Bildschirm, um Tags hinzuzufügen, zu aktualisieren oder zu löschen.
7. Wählen Sie Apply (Anwenden) aus.

Wenn Sie Ihre Anwendungsversion mit der EB CLI aktualisieren, verwenden Sie den Befehl `eb tags (p. 1092)` zum Hinzufügen, Aktualisieren, Löschen oder Auflisten von Tags.

Beispiel: Der folgende Befehl listet die Tags in einer Anwendungsversion auf.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-version"
```

Mit dem folgenden Befehl wird das Tag `mytag1` aktualisiert und das Tag `mytag2` gelöscht.

```
~/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"
```

Eine umfassende Liste der Optionen sowie weitere Beispiele finden Sie unter [eb tags \(p. 1092\)](#).

Wenn Sie die AWS CLI oder andere API-basierte Clients verwenden, nutzen Sie den Befehl [list-tags-for-resource](#), um die Tags einer Anwendungsversion aufzulisten.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-version"
```

Verwenden Sie den Befehl [update-tags-for-resource](#) zum Hinzufügen, Aktualisieren oder Löschen von Tags in einer Anwendungsversion.

```
$ 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"
```

Geben Sie sowohl hinzuzufügende als auch zu aktualisierende Tags im Parameter `--tags-to-add` des Befehls `update-tags-for-resource` an. Wenn ein Tag nicht vorhanden ist, wird es hinzugefügt, andernfalls wird es aktualisiert.

Note

Um einige der EB CLI- und AWS CLI-Befehle mit einer Elastic Beanstalk-Anwendungsversion zu verwenden, benötigen Sie den ARN der Anwendungsversion. Verwenden Sie den folgenden Befehl, um den ARN abzurufen.

```
$ aws elasticbeanstalk describe-application-versions --application-name my-app --  
version-label my-version
```

Erstellen des Quell-Bundles einer Anwendung.

Wenn Sie mithilfe der AWS Elastic Beanstalk-Konsole eine neue Anwendung oder Anwendungsversion bereitstellen, müssen Sie ein Quell-Bundle hochladen. Das Quell-Bundle muss folgende Anforderungen erfüllen:

- Es muss aus einer einzigen ZIP- oder WAR-Datei bestehen (Sie können mehrere WAR-Dateien in eine ZIP-Datei einbinden).
- Es darf nicht größer sein als 512 MB.
- Es darf keinen übergeordneten Ordner bzw. kein Top-Level-Verzeichnis enthalten (Unterverzeichnisse sind in Ordnung).

Soll eine Worker-Anwendung zur Verarbeitung regelmäßiger Hintergrundaufgaben bereitgestellt werden, muss das Quell-Bundle der Anwendung zudem eine `cron.yaml`-Datei enthalten. Weitere Informationen finden Sie unter [Regelmäßige Aufgaben \(p. 519\)](#).

Sofern Sie die Anwendung mit der Elastic Beanstalk Command Line Interface (EB CLI), dem AWS Toolkit for Eclipse oder dem AWS Toolkit for Visual Studio bereitstellen, wird die ZIP- oder WAR-Datei automatisch richtig strukturiert. Weitere Informationen finden Sie unter [Verwenden des Elastic Beanstalk Command Line Interface \(EB CLI\) \(p. 1008\)](#), [Java-Anwendungen in Elastic Beanstalk erstellen und bereitstellen \(p. 109\)](#) und [Das AWS Toolkit for Visual Studio \(p. 220\)](#).

Abschnitte

- [Erstellen eines Quell-Bundles mit der Befehlszeile \(p. 413\)](#)
- [Erstellen eines Quell-Bundles mit Git \(p. 413\)](#)
- [Komprimieren von Dateien im Mac OS X Finder oder Windows-Explorer \(p. 414\)](#)
- [Erstellen eines Quell-Bundles für eine .NET-Anwendung \(p. 418\)](#)
- [Testen des Quell-Bundles \(p. 419\)](#)

Erstellen eines Quell-Bundles mit der Befehlszeile

Erstellen Sie ein Quell-Bundle mit dem Befehl `zip`. Um ausgeblendete Dateien und Ordner einzuschließen, verwenden Sie ein Muster wie das Folgende.

```
~/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%)
```

So wird sichergestellt, dass Elastic Beanstalk-Konfigurationsdateien (p. 722) und andere Dateien und Ordner, die mit einem Punkt beginnen, im Archiv eingeschlossen werden.

Verwenden Sie für Tomcat-Webanwendungen `jar`, um ein Web-Archiv zu erstellen.

```
~/myapp$ jar -cvf myapp.war .
```

Die obigen Befehle schließen ausgeblendete Dateien ein, wodurch Ihr Quell-Bundle unnötig vergrößert werden kann. Verwenden Sie für eine bessere Kontrolle ein detaillierteres Dateimuster oder [erstellen Sie Ihr Quell-Bundle mit Git \(p. 413\)](#).

Erstellen eines Quell-Bundles mit Git

Wenn Sie den Anwendungsquellcode mit Git verwalten, erstellen Sie das Quell-Bundle über den Befehl `git archive`.

```
$ git archive -v -o myapp.zip --format=zip HEAD
```

`git archive` schließt nur Dateien ein, die in Git gespeichert sind, und schließt ignorierte Dateien und Git-Dateien aus. Dies trägt dazu bei, dass Ihr Quell-Bundle so klein wie möglich ist. Weitere Informationen finden Sie unter [git-archive manual page](#).

Komprimieren von Dateien im Mac OS X Finder oder Windows-Explorer

Achten Sie beim Erstellen einer ZIP-Datei im Mac OS X Finder oder Windows-Explorer darauf, die Dateien und Unterordner selbst zu komprimieren (und nicht den übergeordneten Ordner).

Note

In der grafischen Benutzeroberfläche (GUI) von Mac OS X- und Linux-basierten Betriebssystemen werden keine Dateien und Ordner angezeigt, deren Namen mit einem Punkt (.) beginnen. Soll die ZIP-Datei einen versteckten Ordner wie z. B. .ebextensions enthalten, komprimieren Sie die Anwendung mit der Befehlszeile (anstelle der GUI). Informationen zu Befehlszeilenverfahren zum Erstellen einer ZIP-Datei auf einem Mac OS X- oder Linux-basierten Betriebssystem finden Sie unter [Erstellen eines Quell-Bundles mit der Befehlszeile \(p. 413\)](#).

Example

Angenommen, Sie haben einen Python-Projektordner mit dem Namen `myapp` und folgenden Dateien und Unterordnern:

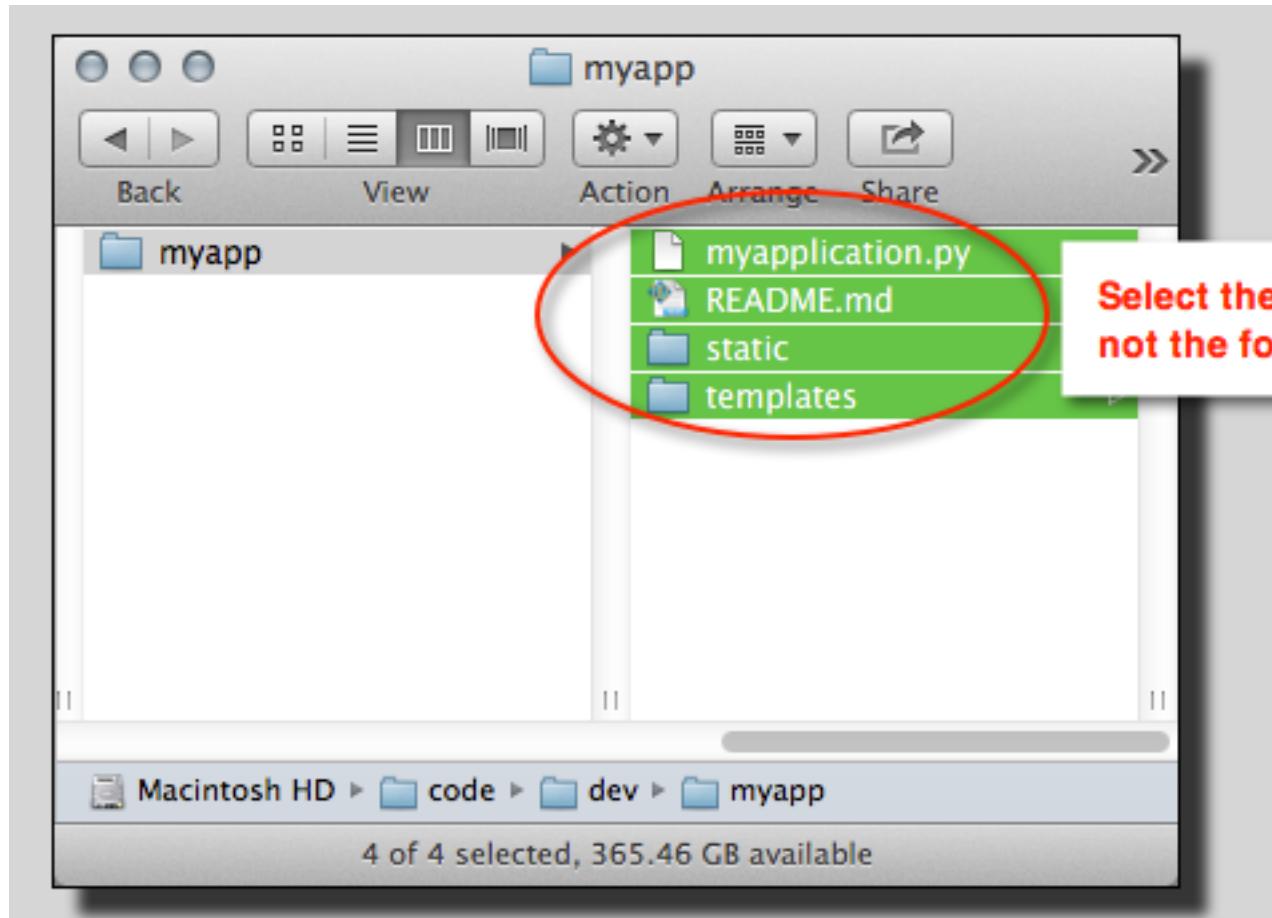
```
myapplication.py
README.md
static/
static/css
static/css/styles.css
static/img
static/img/favicon.ico
static/img/logo.png
templates/
templates/base.html
templates/index.html
```

Wie bereits oben in der Anforderungsliste dargelegt, muss das Quell-Bundle ohne übergeordneten Ordner komprimiert werden, sodass in der extrahierten Struktur kein zusätzliches Top-Level-Verzeichnis vorhanden ist. In diesem Beispiel soll bei der Dateiextraktion kein `myapp`-Ordner erstellt werden (bzw. in der Befehlszeile soll kein `myapp`-Segment zu den Dateipfaden hinzugefügt werden).

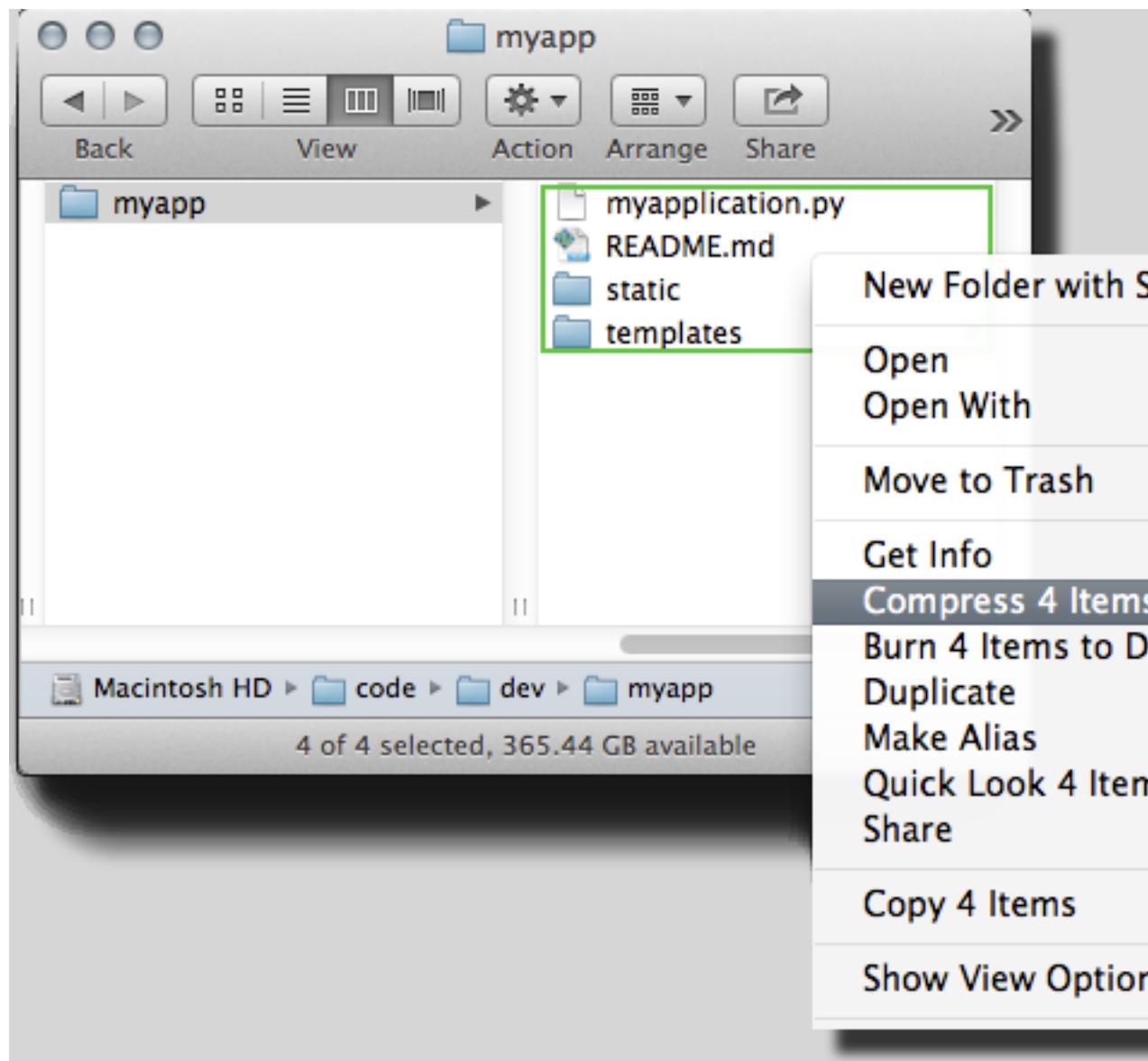
Dieses Dateistrukturbeispiel wird im ganzen Thema verwendet und veranschaulicht das Komprimieren von Dateien.

So komprimieren Sie Dateien im Mac OS X Finder

1. Öffnen Sie den Top-Level-Projektordner und wählen Sie alle Dateien und Unterordner darin aus. Wählen Sie nicht den Top-Level-Ordner selbst aus.

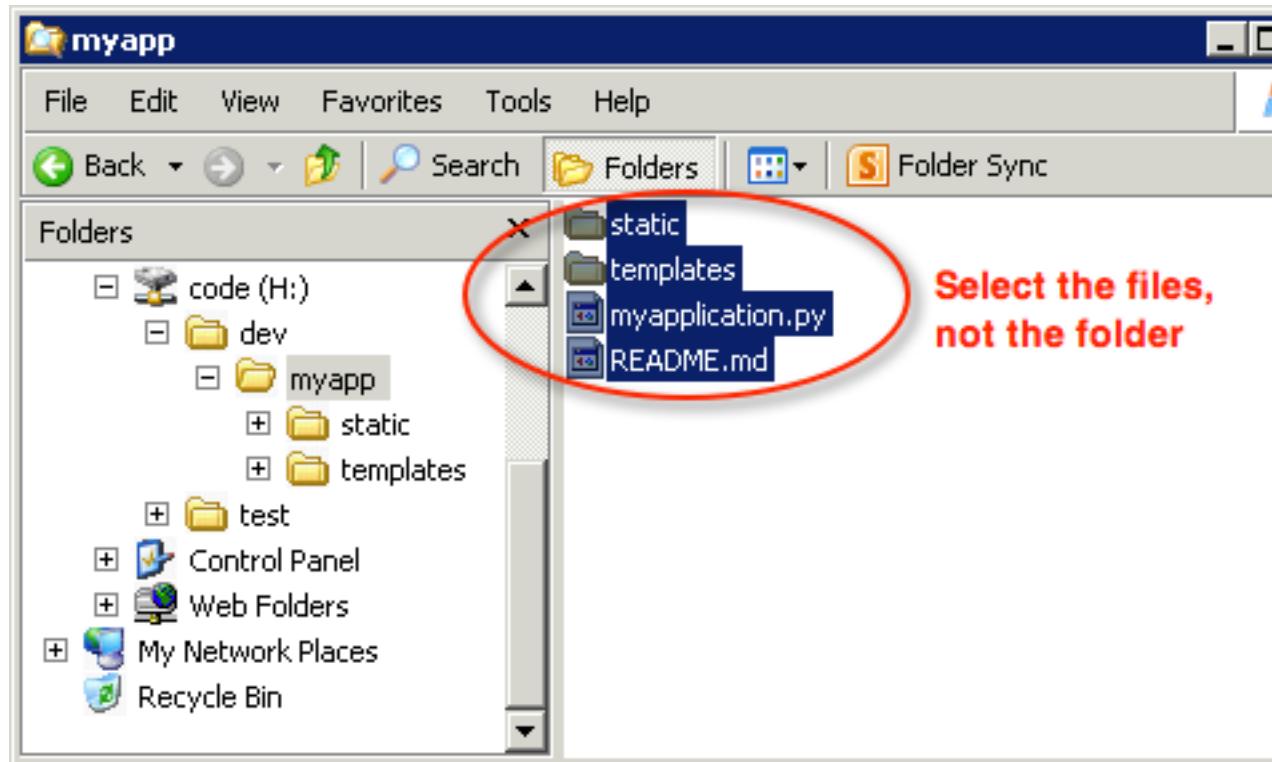


2. Klicken Sie mit der rechten Maustaste auf die ausgewählten Dateien und wählen Sie Compress X items aus (wobei X für die Anzahl der ausgewählten Dateien und Unterordner steht).

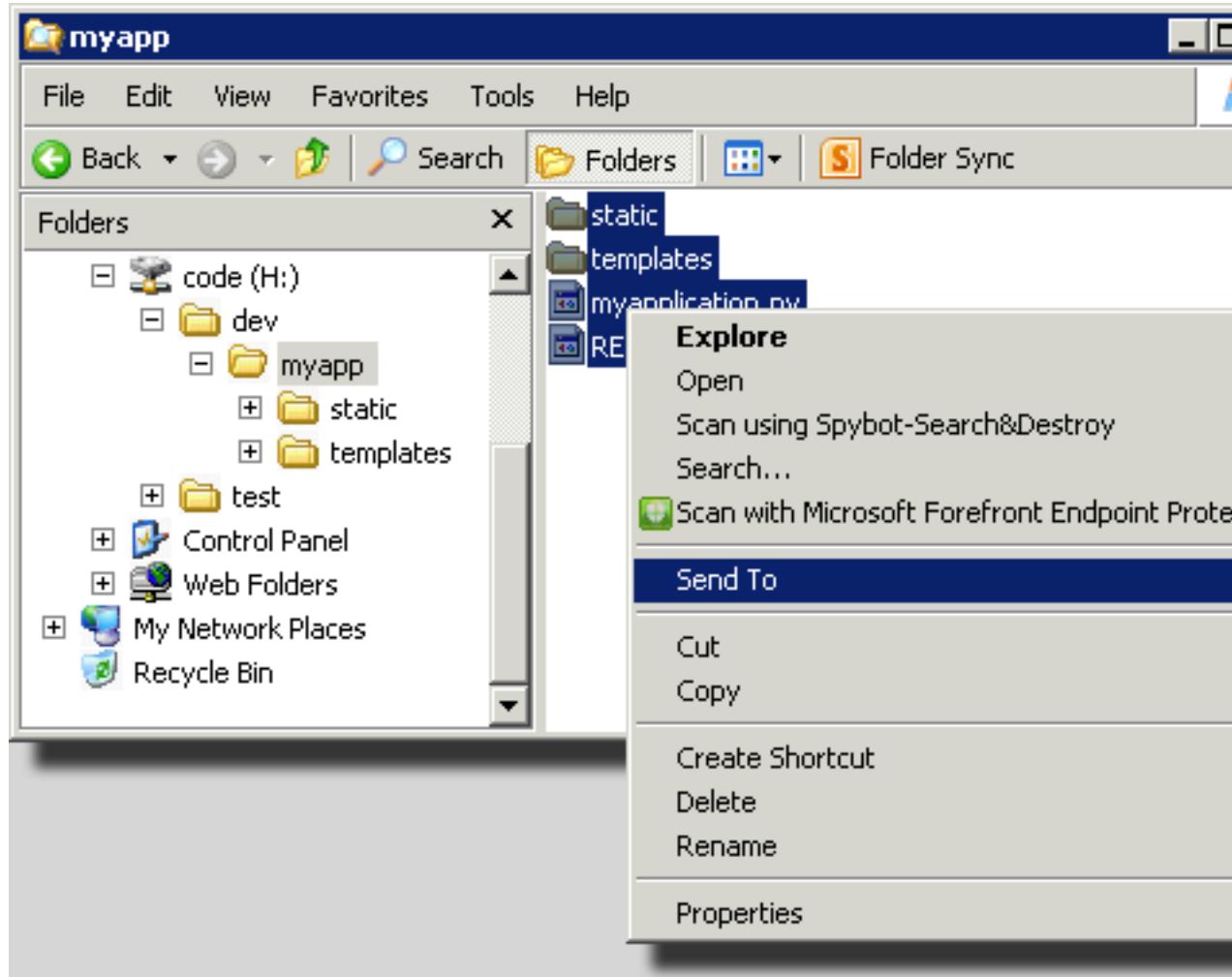


So komprimieren Sie Dateien im Windows-Explorer:

1. Öffnen Sie den Top-Level-Projektordner und wählen Sie alle Dateien und Unterordner darin aus. Wählen Sie nicht den Top-Level-Ordner selbst aus.



2. Klicken Sie mit der rechten Maustaste auf die ausgewählten Dateien, wählen Sie Senden an und dann ZIP-komprimierter Ordner aus.



Erstellen eines Quell-Bundles für eine .NET-Anwendung

Wenn Sie Visual Studio verwenden, können Sie das im AWS Toolkit for Visual Studio enthaltene Bereitstellungstool verwenden, um Ihre .NET-Anwendung in Elastic Beanstalk bereitzustellen. Weitere Informationen finden Sie unter [Bereitstellen von Elastic Beanstalk-Anwendungen in .NET mit dem Bereitstellungstool \(p. 245\)](#).

Falls das Quell-Bundle für die .NET-Anwendung manuell erstellt werden muss, können Sie nicht einfach eine ZIP-Datei mit enthaltenem Projektverzeichnis generieren. Dazu müssen Sie ein Webbereitstellungspaket für das Projekt erstellen, das für die Bereitstellung in Elastic Beanstalk geeignet ist. Es gibt mehrere Möglichkeiten, um ein Bereitstellungspaket zu erstellen:

- Erstellen Sie das Bereitstellungspaket mithilfe des Visual Studio-Assistenten Web veröffentlichen. Weitere Informationen finden Sie unter [Gewusst wie: Erstellen eines Webbereitstellungspakets in Visual Studio](#).

Important

Beim Erstellen eines Webbereitstellungspakets muss der Site name (Site-Name) mit `Default Web Site` beginnen.

- Bei einem .NET-Projekt können Sie das Bereitstellungspaket wie in folgendem Beispiel dargestellt mit dem Befehl msbuild erstellen.

Important

Der Parameter `DeployIisAppPath` muss mit `Default Web Site` beginnen.

```
C:/> msbuild <web_app>.csproj /t:Package /p:DeployIisAppPath="Default Web Site"
```

- Bei einem Website-Projekt können Sie das IIS Web Deploy-Tool zum Erstellen des Bereitstellungspakets nutzen. Weitere Informationen finden Sie unter [Packaging and Restoring a Web site](#).

Important

Der Parameter `apphostconfig` muss mit `Default Web Site` beginnen.

Sofern Sie mehrere Anwendungen oder eine ASP.NET Core-Anwendung bereitstellen, platzieren Sie den Ordner `.ebextensions` im Stamm des Quell-Bundles, parallel zu den Anwendungs-Bundles und der Manifestdatei:

```
~/workspace/source-bundle/
|-- .ebextensions
|   |-- environmentvariables.config
|   '-- healthcheckurl.config
|-- AspNetCore101HelloWorld.zip
|-- AspNetCoreHelloWorld.zip
|-- aws-windows-deployment-manifest.json
`-- VS2015AspNetWebApiApp.zip
```

Testen des Quell-Bundles

Möglicherweise möchten Sie das Quell-Bundle lokal testen, bevor Sie es in Elastic Beanstalk hochladen. Da die Dateien in Elastic Beanstalk im Grunde über die Befehlszeile extrahiert werden, ist es am besten, wenn Sie die Tests mit der Befehlszeile (anstatt mit einem GUI-Tool) ausführen.

So testen Sie die Dateiextraktion in Mac OS X oder Linux

- Öffnen Sie ein Terminalfenster (Mac OS X) oder stellen Sie eine Verbindung zum Linux-Server her. Navigieren Sie zu dem Verzeichnis, in dem sich das Quell-Bundle befindet.
- Extrahieren Sie das Archiv mit dem Befehl `unzip` oder `tar xf`.
- Stellen Sie sicher, dass die Dateien in denselben Ordner wie das Archiv selbst extrahiert werden (und nicht in einen neuen Top-Level-Ordner bzw. -Verzeichnis).

Note

Falls Sie das Archiv mit dem Mac OS X Finder extrahieren, wird – unabhängig von der eigentlichen Archivstruktur – ein neuer Top-Level-Ordner erstellt. Nutzen Sie die Befehlszeile, um beste Ergebnisse zu erzielen.

So testen Sie die Dateiextraktion in Windows

1. Sie sollten ein Programm herunterladen oder installieren, mit dem sich komprimierte Dateien über die Befehlszeile extrahieren lassen. Beispielsweise können Sie das kostenlose Programm "unzip.exe" unter <http://stahlforce.com/dev/index.php?tool=zipunzip> herunterladen.
2. Falls erforderlich, kopieren Sie die ausführbare Datei in das Verzeichnis mit dem Quell-Bundle. Falls ein systemweites Tool installiert wurde, können Sie diesen Schritt überspringen.
3. Extrahieren Sie das Archiv mit dem entsprechenden Befehl. Wenn Sie "unzip.exe" über den in Schritt 1 angegebenen Link heruntergeladen haben, lautet der Befehl `unzip <archive-name>`.
4. Stellen Sie sicher, dass die Dateien in denselben Ordner wie das Archiv selbst extrahiert werden (und nicht in einen neuen Top-Level-Ordner bzw. -Verzeichnis).

Markieren von Elastic Beanstalk-Anwendungsressourcen

Sie können Tags auf Ressourcen Ihrer AWS Elastic Beanstalk-Anwendungen anwenden. Tags sind mit AWS-Ressourcen verknüpfte Schlüssel-Wert-Paare. Mit Tags können Sie Ressourcen kategorisieren. Sie sind besonders nützlich, wenn Sie viele Ressourcen im Rahmen mehrerer AWS-Anwendungen verwalten.

Hier finden Sie einige Möglichkeiten zur Nutzung von Tags mit Elastic Beanstalk-Ressourcen:

- Bereitstellungsstufen – Identifizieren Sie Ressourcen in verschiedenen Stufen Ihrer Anwendung, wie z. B. Entwicklung, Beta und Produktion.
- Kostenzuordnung – Verwenden Sie Kostenzuordnungsberichte zum Nachverfolgen Ihrer Nutzung der AWS-Ressourcen in Verbindung mit verschiedenen Aufwandskonten. Die Berichte enthalten außerdem markierte und nicht markierte Ressourcen, aggregieren die Kosten jedoch gemäß den Tags. Weitere Informationen darüber, wie Kostenzuordnungsberichte Tags verwenden, finden Sie unter [Verwenden von Kostenzuordnungs-Tags für benutzerdefinierte Fakturierungsberichte](#) im Benutzerhandbuch für AWS-Fakturierungs- und Kostenverwaltung.
- Zugriffskontrolle – Verwenden Sie Tags zur Verwaltung der Berechtigungen für Anfragen und Ressourcen. Beispiel: Ein Benutzer, der nur Beta-Umgebungen erstellen und verwalten kann, sollte nur Zugriff auf Beta-Ressourcen haben. Details dazu finden Sie unter [Tags zur Steuerung des Zugriffs auf Elastic Beanstalk-Ressourcen verwenden \(p. 960\)](#).

Sie können bis zu 50 Tags für jede Ressource hinzufügen. Umgebungen unterscheiden sich geringfügig: Elastic Beanstalk fügt Umgebungen drei Standard-System-Tags hinzu. Diese Tags können Sie weder bearbeiten noch löschen. Neben den Standard-Tags können Sie jeder Umgebung bis zu 47 zusätzliche Tags hinzufügen.

Die folgenden Einschränkungen gelten für Tag-Schlüssel und -Werte:

- Schlüssel und Werte dürfen Buchstaben, Zahlen, Leerzeichen sowie die folgenden Symbole enthalten:
`_ . : / = + - @`
- Schlüssel können bis zu 127 Zeichen enthalten. Werte können bis zu 255 Zeichen enthalten.

Note

Diese Längenbeschränkungen gelten für Unicode-Zeichen in UTF-8. Für andere Multibyte-Kodierungen können niedrigere Limits gelten.

- Bei Schlüsseln muss die Groß- und Kleinschreibung beachtet werden.
- Schlüssel dürfen nicht mit `aws:` oder `elasticbeanstalk:` beginnen.

Ressourcen, die markiert werden können

Im Folgenden werden die Arten von Elastic Beanstalk-Ressourcen, die Sie markieren können, sowie Links zu bestimmten Themen zum Verwalten von Tags für die einzelnen Ressourcen aufgeführt:

- [Anwendungen \(p. 421\)](#)
- [Umgebungen \(p. 617\)](#)
- [Anwendungsversionen \(p. 410\)](#)
- [Gespeicherte Konfigurationen \(p. 769\)](#)
- [Benutzerdefinierte Plattformversionen \(p. 1127\)](#)

Markieren von Anwendungen

Sie können Tags auf Ihre AWS Elastic Beanstalk-Anwendungen anwenden. Tags sind mit AWS-Ressourcen verknüpfte Schlüssel-Wert-Paare. Weitere Informationen zum Elastic Beanstalk-Ressourcen-Tagging, zu Anwendungsfälle, Einschränkungen für Tag-Schlüssel und -Werte sowie zu unterstützten Ressourcentypen finden Sie unter [Markieren von Elastic Beanstalk-Anwendungsressourcen \(p. 420\)](#).

Sie können Tags angeben, wenn Sie eine Anwendung erstellen. In einer vorhandenen Anwendung können Sie Tags hinzufügen oder entfernen sowie die Werte von vorhandenen Tags aktualisieren. Sie können jeder Anwendung bis zu 50 Tags hinzufügen.

Hinzufügen von Tags während der Anwendungserstellung

Wenn Sie die Elastic Beanstalk-Konsole zum [Erstellen einer Anwendung \(p. 403\)](#) nutzen, können Sie Tag-Schlüssel und -Werte im Dialogfeld Create New Application (Neue Anwendung erstellen) angeben.

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 per resource and is case-sensitive. [Learn more](#)

Key	Value	Remove
<input type="text"/>	<input type="text"/>	<input type="button" value="Remove"/>

Add tag

50 remaining

Ca

Wenn Sie mit der EB CLI eine Anwendung erstellen, verwenden Sie die --tags-Option mit `eb init` (p. 1067), um Tags hinzuzufügen.

```
~/workspace/my-app$ eb init --tags mytag1=value1,mytag2=value2
```

Fügen Sie Tags mit der AWS CLI oder anderen API-basierten Clients hinzu, indem Sie den --tags-Parameter mit dem Befehl `create-application` verwenden.

```
$ aws elasticbeanstalk create-application \
--tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
```

```
--application-name my-app --version-label v1
```

Verwalten der Tags einer vorhandenen Anwendung

Sie können Tags in einer vorhandenen Elastic Beanstalk-Anwendung hinzufügen, aktualisieren und löschen.

So verwalten Sie die Tags einer Anwendung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.

Note

Wenn Sie viele Anwendungen haben, verwenden Sie die Suchleiste, um die Anwendungsliste zu filtern.

3. Klicken Sie auf Actions (Aktionen) und dann auf Manage Tags (Tags verwalten).
4. Verwenden Sie das Formular auf dem Bildschirm, um Tags hinzuzufügen, zu aktualisieren oder zu löschen.
5. Wählen Sie Apply (Anwenden) aus.

Wenn Sie Umgebungen mit der EB CLI aktualisieren, verwenden Sie zum Hinzufügen, Aktualisieren, Löschen oder Auflisten von Tags den Befehl [eb tags \(p. 1092\)](#).

Beispiel: Der folgende Befehl listet die Tags in einer Anwendung auf.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Mit dem folgenden Befehl wird das Tag mytag1 aktualisiert und das Tag mytag2 gelöscht.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
--resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Eine umfassende Liste der Optionen sowie weitere Beispiele finden Sie unter [eb tags \(p. 1092\)](#).

Wenn Sie die AWS CLI oder andere API-basierte Clients verwenden, listen Sie die Tags einer Anwendung mit dem Befehl [list-tags-for-resource](#) auf.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Verwenden Sie den Befehl [update-tags-for-resource](#) zum Hinzufügen, Aktualisieren und Löschen von Tags in einer Anwendung.

```
$ 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"
```

Geben Sie sowohl hinzuzufügende als auch zu aktualisierende Tags im Parameter --tags-to-add des Befehls [update-tags-for-resource](#) an. Wenn ein Tag nicht vorhanden ist, wird es hinzugefügt, andernfalls wird es aktualisiert.

Note

Um einige der EB CLI- und AWS CLI-Befehle mit einer Elastic Beanstalk-Anwendung zu nutzen, benötigen Sie den ARN der Anwendung. Verwenden Sie den folgenden Befehl, um den ARN abzurufen.

```
$ aws elasticbeanstalk describe-applications --application-names my-app
```

Verwalten von Umgebungen

Mit AWS Elastic Beanstalk ist es einfach, neue Umgebungen für Ihre Anwendung zu erstellen. Sie können eigene Umgebungen für Entwicklung, Tests und Produktion erstellen und verwalten und [jede Version \(p. 471\)](#) der Anwendung in jeder Umgebung bereitstellen. Umgebungen können lange oder temporär eingesetzt werden. Wenn Sie eine Umgebung beenden, können Sie deren Konfiguration für eine spätere erneute Erstellung speichern.

Im Entwicklungsverlauf der Anwendung wird diese häufig – und zu verschiedenen Zwecken in mehreren unterschiedlichen Umgebungen – bereitgestellt. Mit Elastic Beanstalk können Sie [konfigurieren, wie Bereitstellungen durchgeführt werden \(p. 474\)](#). Sie können die Bereitstellung für alle Instances der Umgebung gleichzeitig ausführen oder sie in Stapel für eine fortlaufende Bereitstellung unterteilen.

[Konfigurationsänderungen \(p. 483\)](#) werden getrennt von Bereitstellungen verarbeitet und haben einen eigenen Umfang. Wenn Sie beispielsweise den Typ der EC2-Instances ändern, auf denen die Anwendung ausgeführt wird, müssen alle Instances ersetzt werden. Ändern Sie hingegen die Konfiguration vom Load Balancer der Umgebung, kann diese Änderung direkt ohne Serviceunterbrechung oder Kapazitätseinbußen erfolgen. Sie können auch Konfigurationsänderungen in [fortlaufenden Konfigurationsaktualisierungen \(p. 484\)](#) ausführen, bei denen die Instances der Umgebung in Stapeln geändert werden.

Note

Ändern Sie die Ressourcen der Umgebung nur mit Elastic Beanstalk. Falls Sie die Ressourcen mit der Konsole, den CLI-Befehlen oder den SDKs eines anderen Service ändern, kann Elastic Beanstalk den Status dieser Ressourcen nicht akkurat überwachen. Außerdem können Sie weder die Konfiguration speichern noch die Umgebung zuverlässig erneut erstellen. Out-of-Band-Änderungen können zudem zu Problemen beim Aktualisieren oder Beenden einer Umgebung führen.

Beim Erstellen einer Umgebung wählen Sie eine Plattformversion aus. Wir aktualisieren die Plattformen regelmäßig mit neuen Plattformversionen und stellen damit Leistungsverbesserungen und neue Funktionen bereit. Sie können [Ihre Umgebung jederzeit auf die neueste Plattformversion aktualisieren \(p. 491\)](#).

Mit zunehmender Komplexität der Anwendung können Sie diese in mehrere Komponenten aufteilen, von denen jede in einer eigenen Umgebung ausgeführt wird. Für lange laufende Workloads können Sie [Worker-Umgebungen \(p. 516\)](#) starten, die Aufträge von einer Amazon Simple Queue Service (Amazon SQS)-Warteschlange verarbeiten.

Themen

- [Elastic Beanstalk-Umgebungsverwaltungskonsole verwenden \(p. 426\)](#)
- [Erstellen einer Elastic Beanstalk-Umgebung \(p. 435\)](#)
- [Anwendungen in Elastic Beanstalk-Umgebungen bereitstellen \(p. 471\)](#)
- [Konfigurationsänderungen \(p. 483\)](#)
- [Aktualisieren der Plattformversion für die Elastic Beanstalk-Umgebung \(p. 491\)](#)
- [Abbrechen von Umgebungskonfigurations-Updates und Anwendungsbereitstellungen \(p. 510\)](#)
- [Elastic Beanstalk-Umgebungen wiederherstellen \(p. 511\)](#)
- [Umgebungstypen \(p. 514\)](#)
- [AWS Elastic Beanstalk-Worker-Umgebungen \(p. 516\)](#)

- Verknüpfungen zwischen Elastic Beanstalk-Umgebungen erstellen (p. 524)

Elastic Beanstalk-Umgebungsverwaltungskonsole verwenden

Die Elastic Beanstalk-Konsole bietet eine Verwaltungskonsole für jede Ihrer AWS Elastic Beanstalk-Umgebungen. Auf dieser Seite können Sie die Konfiguration Ihrer Umgebung verwalten und allgemeine Aktionen, wie z. B. den Neustart der Webserver in Ihrer Umgebung, das Klonen der Umgebung oder das Erstellen von Grund auf, durchführen.

The screenshot shows the AWS Elastic Beanstalk Environment Management Console. On the left, a sidebar navigation bar lists 'Environments' and 'Applications', with 'getting-started-app' and 'GettingStartedApp-env' expanded. Under 'GettingStartedApp-env', options like 'Go to environment', 'Configuration', 'Logs', 'Health', 'Monitoring', 'Alarms', 'Managed updates', 'Events', and 'Tags' are visible. Below this is a section for 'Recent environments' with 'GettingStartedApp-env' listed again. The main content area displays the 'GettingStartedApp-env' environment details. It includes a 'Health' section with a green circle containing a checkmark and the status 'Ok'. To the right are buttons for 'Running version' (labeled 'Sample Application') and 'Upload and deploy'. At the bottom, there's a 'Recent events' table with the following data:

Time	Type	Details
2020-01-28 12:11:17 UTC-0800	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 9 seconds ago.
2020-01-28 12:11:15 UTC-0800	INFO	Successfully launched environment: GettingStartedApp-env
2020-01-28 12:11:13 UTC-0800	INFO	Application available at GettingStartedApp-env.bx7dx222kw.us-east-2.elasticbeanstalk.com
2020-01-28 12:10:32 UTC-0800	INFO	Created CloudWatch alarm named: airseb-e-cubmdjm6ga-stack-AWSEBCloudwatchAlarmHigh
2020-01-28 12:10:32 UTC-0800	INFO	Created CloudWatch alarm named: airseb-e-cubmdjm6ga-stack-AWSEBCloudwatchAlarmLow

Verwenden der Environment Management Console

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

Ihnen wird die Seite „Umgebungsübersicht“ angezeigt. Im Navigationsbereich der Konsole werden der Name der Anwendung, zu der die Umgebung gehört, verwandte Anwendungsverwaltungsseiten, der Name der Umgebung und Umgebungsverwaltungsseiten angezeigt.

Themen

- [Umgebungsübersicht \(p. 427\)](#)
- [Umgebungsaktionen \(p. 428\)](#)
- [Konfiguration \(p. 430\)](#)
- [Protokolle \(p. 431\)](#)
- [Integrität \(p. 431\)](#)
- [Überwachung \(p. 432\)](#)
- [Alarne \(p. 432\)](#)
- [Verwaltete Aktualisierungen \(p. 433\)](#)
- [Events \(p. 433\)](#)
- [Tags \(p. 434\)](#)

Umgebungsübersicht

Zur Anzeige der Seite „Umgebungsübersicht“ wählen Sie im Navigationsbereich den Namen der Umgebung aus, wenn Sie Informationen zur aktuellen Umgebung erhalten möchten. Alternativ können Sie über die Anwendungsseite oder die Hauptumgebungsliste auf der Seite Environments (Umgebungen) zur Umgebung navigieren.

Im oberen Bereich auf der Seite „Umgebungsübersicht“ werden Übersichtsinformationen zu Ihrer Umgebung angezeigt. Dazu zählen der Name, die URL, der aktuelle Integritätsstatus, der Name der zurzeit bereitgestellten Anwendungsversion und die Plattformversion, auf der die Anwendung ausgeführt wird. Unterhalb des Übersichtsbereichs werden Ihnen die letzten fünf Ereignisse in der Umgebung angezeigt.

Klicken Sie auf Refresh (Aktualisieren), um die angezeigten Informationen zu aktualisieren. Die Übersichtsseite enthält die folgenden Informationen und Optionen.

URL

Die URL der Umgebung befindet sich oben in der Übersicht unterhalb des Namens der Umgebung. Dies ist die URL der Webanwendung, die die Umgebung ausführt.

Integrität

Der Gesamtzustand der Umgebung. Wenn [Erweiterte Zustandsberichte und Überwachung \(p. 824\)](#) aktiviert ist, wird der Umgebungsstatus mit der Schaltfläche Causes (Ursachen) angezeigt. Klicken Sie darauf, um weitere Informationen über den aktuellen Status anzuzeigen.

Für [Grundlegende Zustandsberichte \(p. 821\)](#)-Umgebungen wird ein Link zur [Überwachungskonsole \(p. 817\)](#) angezeigt.

Ausgeführte Version

Der Name der in Ihrer Umgebung bereitgestellten und ausgeführten Anwendungsversion. Wählen Sie Upload and Deploy (Hochladen und bereitstellen) aus, um ein [Quell-Bundle \(p. 412\)](#) hochzuladen und es in Ihrer Umgebung bereitzustellen. Diese Option erstellt eine neue Anwendungsversion.

Plattform

Zeigt den Namen der Plattformversion an, die in Ihrer Umgebung ausgeführt wird. Dabei handelt es sich in der Regel um eine Kombination aus Architektur, Betriebssystem, Sprache und Anwendungsserver (zusammen Plattformzweig) mit einer spezifischen Plattformversionsnummer. Wählen Sie Change (Ändern) aus, um eine andere Plattformversion auszuwählen. Diese Option ist nur verfügbar, wenn eine andere Version des Plattformzweigs verfügbar ist.

Durch das Aktualisieren der Plattformversion mit dieser Option werden Instances, die in Ihrer Umgebung ausgeführt werden, durch neue Instances ersetzt.

Update platform version

Availability warning

This operation replaces your instances; your application is unavailable during the update. To keep one instance in service during the update, enable rolling updates. Another option is to clone the environment, which creates a newer version of the platform, and then swap the CNAME of the clone 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
Tomcat 8.5 with Java 8 running on 64bit Amazon Linux

Current platform version
3.3.1

New platform version
3.3.2 (Recommended) ▾

Canc

Note

Bei erstmaliger Verwendung von Elastic Beanstalk ist jeweils nur die neueste (empfohlene) Version der einzelnen Plattformzweige verfügbar. Change (Ändern) ist erst verfügbar, wenn eine neue Plattformversion für den Zweig veröffentlicht wird. Nach dem Upgrade können Sie zur vorherigen Version zurückwechseln.

Aktuelle Ereignisse

Im Abschnitt Recent Events (Aktuelle Ereignisse) auf der Seite „Umgebungsübersicht“ werden die neuesten Ereignisse angezeigt, die von Ihrer Umgebung ausgegeben wurden. Diese Liste wird in Echtzeit aktualisiert, wenn Ihre Umgebung aktualisiert wird.

Wählen Sie Show All (Alle anzeigen) aus, um die Seite Events (Ereignisse) zu öffnen.

Umgebungsaktionen

Die Seite „Umgebungsübersicht“ enthält das Menü Environment actions (Umgebungsaktionen), über das Sie allgemeine Operationen für Ihre Umgebung ausführen können. Das Menü wird auf der rechten

Seite des Umgebungs-Headers unter der Option Create New Environment (Neue Umgebung erstellen) angezeigt.

Note

Einige Aktionen sind nur unter bestimmten Bedingungen verfügbar und deaktiviert, es sei denn, diese Bedingungen werden erfüllt.

Laden der Konfiguration

Laden Sie eine zuvor gespeicherte Konfiguration. Konfigurationen werden in Ihrer Anwendung gespeichert und können von jeder zugehörigen Umgebung geladen werden. Wenn Sie Änderungen an der Konfiguration Ihrer Umgebung vorgenommen haben, können Sie eine gespeicherte Konfiguration laden, um diese Änderungen rückgängig zu machen. Sie können auch eine gespeicherte Konfiguration laden, die Sie aus einer anderen Umgebung mit derselben Anwendung gespeichert haben, um Konfigurationsänderungen zwischen ihnen zu propagieren.

Speichern der Konfiguration

Speichern Sie die aktuelle Konfiguration Ihrer Umgebung in Ihrer Anwendung. Bevor Sie Änderungen an der Konfiguration Ihrer Umgebung vornehmen, speichern Sie die aktuelle Konfiguration, damit Sie sie später bei Bedarf wiederherstellen können. Sie können auch eine gespeicherte Konfiguration anwenden, wenn Sie eine neue Umgebung starten.

Austauschen von Umgebungs-URLs

Tauschen Sie den CNAME der aktuellen Umgebung gegen eine neue Umgebung aus. Nach einem CNAME-Austausch wird der gesamte Datenverkehr zur Anwendung, die die Umgebungs-URL verwendet, an die neue Umgebung geleitet. Wenn Sie bereit sind, eine neue Version Ihrer Anwendung bereitzustellen, können Sie eine separate Umgebung unter der neuen Version starten. Wenn die neue Umgebung bereit ist, Anfragen zu bearbeiten, führen Sie einen CNAME-Austausch durch, um den Datenverkehr ohne Serviceunterbrechung an die neue Umgebung weiterzuleiten. Weitere Informationen finden Sie unter [Blau/Grün-Bereitstellungen mit AWS Elastic Beanstalk \(p. 481\)](#).

Klonen einer Umgebung

Starten Sie eine neue Umgebung mit der gleichen Konfiguration wie in Ihrer aktuell ausgeführten Umgebung.

Klonen mit der neuesten Plattform

Sie können Ihre aktuelle Umgebung mit der neuesten Version der jeweils verwendeten Elastic Beanstalk-Plattform klonen. Diese Option ist nur verfügbar, wenn eine neuere Version der aktuellen Umgebungsplattform zur Verfügung steht.

Abbrechen des aktuellen Vorgangs

Stoppen Sie ein laufendes Umgebungs-Update. Das Abbrechen eines Vorgangs kann dazu führen, dass einige der Instances in Ihrer Umgebung einen anderen Zustand haben als andere – je nachdem, wie weit der Prozess fortgeschritten ist. Diese Option ist nur verfügbar, wenn Ihre Umgebung aktualisiert wird.

Neustarten von App-Servern

Starten Sie den Webserver in den Instances Ihrer Umgebung neu. Durch diese Option werden keine AWS-Ressourcen beendet oder neu gestartet. Wenn Ihre Umgebung merkwürdiges Verhalten als Reaktion auf einige fehlerhafte Anfragen zeigt, kann durch einen Neustart des Anwendungsservers die Funktionalität vorübergehend wiederhergestellt werden, während Sie der Ursache auf den Grund gehen.

Neuerstellen der Umgebung

Beenden Sie alle Ressourcen in der laufenden Umgebung und erstellen Sie eine neue Umgebung mit den gleichen Einstellungen bereit. Dieser Vorgang dauert einige Minuten, entsprechend der Bereitstellung einer neuen Umgebung von Grund auf. Alle Amazon RDS-Instances, die in der Datenschicht Ihrer Umgebung ausgeführt werden, werden während einer Neuerstellung gelöscht. Wenn Sie die Daten benötigen, erstellen Sie einen Snapshot. Sie können einen Snapshot manuell [in der RDS-Konsole](#) erstellen oder die Richtlinie zum Löschen Ihrer Datenschicht so konfigurieren, dass vor dem Löschen der Instance automatisch ein Snapshot erstellt wird. (Dies ist die Standardeinstellung bei der Erstellung einer Datenschicht).

Beenden der Umgebung

Beenden Sie alle Ressourcen in der laufenden Umgebung und entfernen Sie die Umgebung aus der Anwendung. Wenn eine RDS-Instance in einer Datenschicht ausgeführt wird und Sie die Daten behalten müssen, stellen Sie sicher, dass der Snapshot vor dem Beenden Ihrer Umgebung erstellt wird. Sie können einen Snapshot manuell [in der RDS-Konsole](#) erstellen oder die Richtlinie zum Löschen Ihrer Datenschicht so konfigurieren, dass vor dem Löschen der Instance automatisch ein Snapshot erstellt wird. (Dies ist die Standardeinstellung bei der Erstellung einer Datenschicht).

Wiederherstellen der Umgebung

Wenn die Umgebung in der letzten Stunde beendet wurde, können Sie sie auf dieser Seite wiederherstellen. Nach einer Stunde können Sie sie [auf der Seite mit der Anwendungsübersicht wiederherstellen \(p. 512\)](#).

Konfiguration

Die Seite Configuration overview (Konfigurationsübersicht) zeigt die aktuelle Konfiguration Ihrer Umgebung und ihre Ressourcen, einschließlich Amazon EC2-Instances, Load Balancer, Benachrichtigungen und Einstellungen für die Zustandsüberwachung. Verwenden Sie die Einstellungen auf dieser Seite, um das Verhalten Ihrer Umgebung während Bereitstellungen anzupassen, zusätzliche Funktionen zu aktivieren und den Instance-Typ und andere Einstellungen zu ändern, die Sie während der Umgebungserstellung wählen.

The screenshot shows the 'Configuration overview' page of the AWS Elastic Beanstalk console. The page is divided into six main sections:

- Software:** Node.js version: 10.15.3, AWS X-Ray: disabled, Rotate logs: disabled (default), Log streaming: disabled (default), Environment properties: 0. A 'Modify' button is at the bottom.
- Instances:** EC2 instance type: t2.micro, EC2 image ID: ami-0159e246d939e78a0, Monitoring interval: 5 minute, Root volume type: container default, Root volume size (GB): container default, Root volume IOPS: container default, Security groups: sg-07a4598b8cfe0f9a0. A 'Modify' button is at the bottom.
- Capacity:** Environment type: load balancer, Availability Zones: Any, Instances: 1–4. A 'Modify' button is at the bottom.
- Load balancer:** Load balancer type: application, Listeners: 1, Processes: 1.
- Rolling updates and deployments:** Deployment policy: All at once, Rolling updates: disabled, Health check: enabled.
- Security:** Service role: aws-elasticbeanstalk-environment, Virtual machine key pair: Virtual machine instance.

At the top right of the page are 'Cancel' and 'Review changes' buttons.

Weitere Informationen finden Sie unter [Konfigurieren von Elastic Beanstalk-Umgebungen \(p. 526\)](#).

Protokolle

Auf der Seite Logs (Protokolle) können Sie Protokolle aus den EC2-Instances in Ihrer Umgebung abrufen. Wenn Sie Protokolle anfordern, sendet Elastic Beanstalk einen Befehl an die Instances, die dann Protokolle in Ihren Elastic Beanstalk Storage Bucket in Amazon S3 hochladen. Wenn Sie Protokolle auf dieser Seite anfordern, löscht Elastic Beanstalk diese nach 15 Minuten automatisch aus Amazon S3.

Sie können die Instances Ihrer Umgebung auch so konfigurieren, dass Protokolle zur dauerhaften Speicherung in Amazon S3 hochgeladen werden, nachdem sie lokal rotiert sind.

Elastic Beanstalk > Environments > GettingStartedApp-env > Logs

Logs

Click Request Logs to retrieve the last 100 lines of logs or the entire set of logs from each EC2 instance. [Learn more](#)

Log file	Time	EC2 instance

Click Request Logs to request and review log files for all your servers.

Request Logs

Last 100 Lines

Full Logs

Weitere Informationen finden Sie unter [Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen \(p. 869\)](#).

Integrität

Bei Aktivierung der erweiterten Integritätsüberwachung werden auf der Seite Enhanced Health Overview (Erweiterte Integritätsübersicht) Live-Informationen zur Integrität der einzelnen Instances in Ihrer Umgebung angezeigt. Die erweiterte Zustandsüberwachung ermöglicht Elastic Beanstalk, die Ressourcen in Ihrer Umgebung genau zu überwachen, sodass der Zustand Ihrer Anwendung genauer bewertet werden kann.

Wenn die erweiterte Zustandsüberwachung aktiviert ist, zeigt diese Seite Informationen über die Anfragen, die von den Instances in Ihrer Umgebung verarbeitet werden, und Metriken aus dem Betriebssystem, wie z. B. Latenz, Last und CPU-Auslastung.

Elastic Beanstalk > Environments > GettingStartedApp-env > Health

Enhanced health overview

Instances: 2 Total, 2 Ok

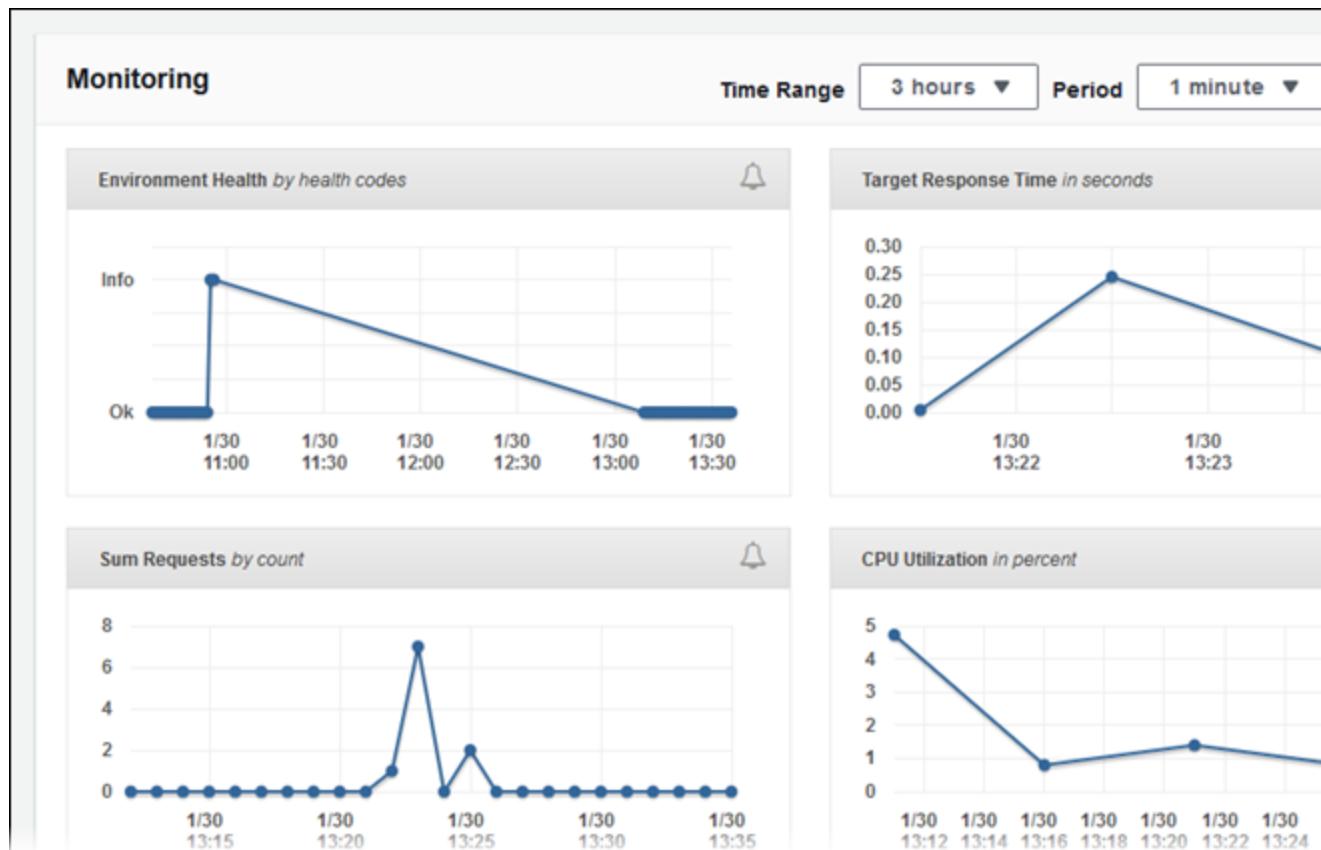
[Learn more](#) about enhanced health.

Instance ID	Status	Running	Deployment ID	Requests/sec	2xx Responses	3xx Responses	4xx Responses	5xx Responses	P99 Latency	P90 Latency
Overall	Ok	N/A	N/A	0.4	100%	0.0%	0.0%	0.0%	0.002	0.00
i-00227807c4c4a1334	Ok	2 hours	3	0.2	2	0	0	0	0.002	0.002
i-03280193ba1ba4171	Ok	19 days	3	0.2	2	0	0	0	0.001	0.001

Weitere Informationen finden Sie unter [Erweiterte Zustandsberichte und Überwachung \(p. 824\)](#).

Überwachung

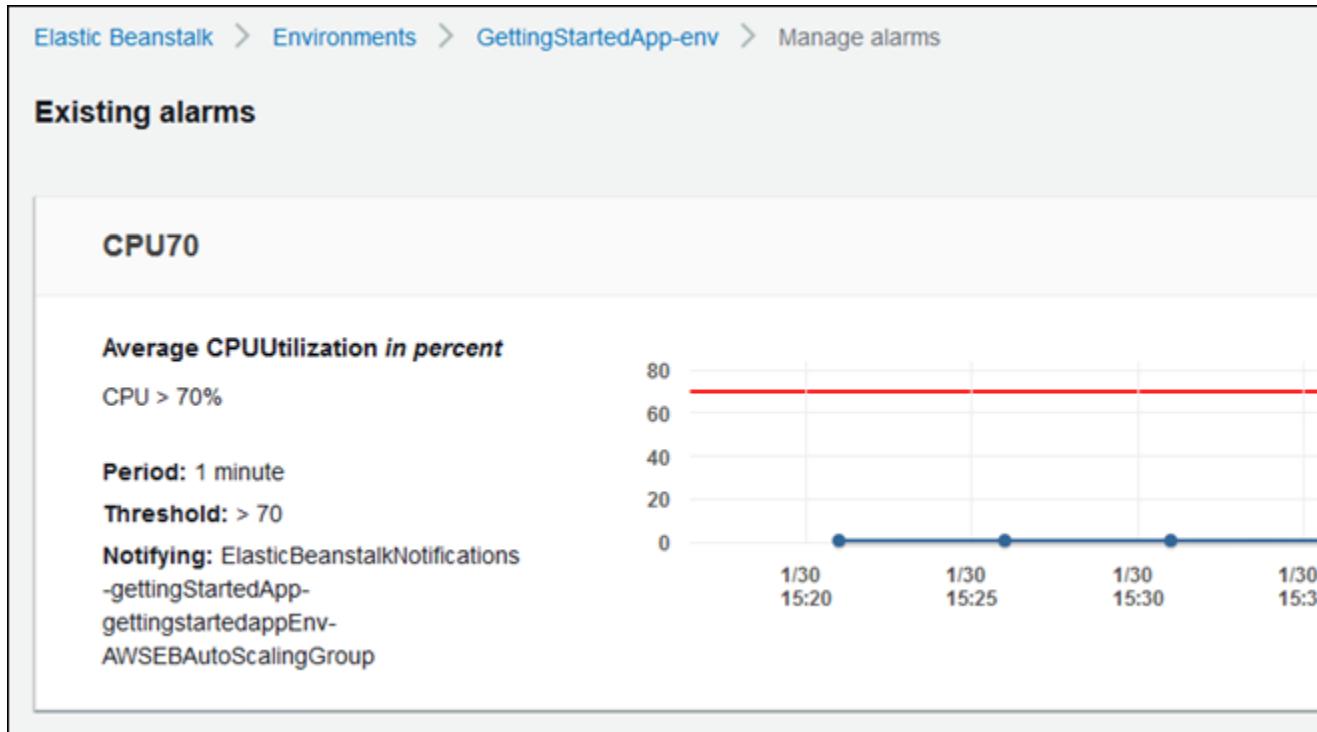
Die Seite Monitoring (Überwachung) zeigt eine Übersicht über die Zustandsdaten für Ihre Umgebung an. Dies umfasst die Standardmetriken von Elastic Load Balancing und Amazon EC2 sowie Diagramme, die veranschaulichen, wie sich der Zustand der Umgebung im Laufe der Zeit geändert hat. Sie können die Optionen auf dieser Seite verwenden, um zusätzliche Diagramme für ressourcenspezifische Metriken zu konfigurieren und Alarne für Metriken hinzuzufügen, die vom verwendeten System für Zustandsberichte unterstützt werden.



Weitere Informationen finden Sie unter [Überwachen des Umgebungszustands in der AWS Managementkonsole \(p. 817\)](#).

Alarne

Auf der Seite Existing Alarms (Vorhandene Alarne) werden Informationen zu allen Alarmen angezeigt, die Sie für Ihre Umgebung konfiguriert haben. Sie können die Optionen auf dieser Seite verwenden, um Alarne zu ändern oder zu löschen.



Weitere Informationen finden Sie unter [Verwalten von Alarmen \(p. 861\)](#).

Verwaltete Aktualisierungen

Auf der Seite Managed updates overview (Übersicht über verwaltete Updates) werden Informationen zu bevorstehenden und abgeschlossenen Updates und Instance-Ersetzungen auf der verwalteten Plattform angezeigt. Mit diesen Funktionen können Sie Ihre Umgebung so konfigurieren, dass während eines wöchentlichen Wartungsfensters, das Sie festlegen, automatisch auf die neueste Plattform aktualisiert wird.

Zwischen Plattformversionen können Sie festlegen, dass Ihre Umgebung alle Amazon EC2-Instances während des Wartungsfensters ersetzt. Dadurch können Probleme vermieden werden, die auftreten, wenn Ihre Anwendung für längere Zeiträume ausgeführt wird.

Weitere Informationen finden Sie unter [Verwaltete Plattformaktualisierungen \(p. 496\)](#).

Events

Die Seite Events (Ereignisse) zeigt den Event-Stream für Ihre Umgebung. Elastic Beanstalk gibt Ereignismeldungen aus, wenn Sie mit der Umgebung interagieren und wenn Ressourcen Ihrer Umgebung infolgedessen erstellt oder geändert werden.

Elastic Beanstalk > Environments > GettingStartedApp-env > Events

 Click the link to be routed to the previous Beanstalk Console
Switch to the previous console

Events

Severity	TRACE	<	1	2	3	4	5	6	7
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 update completed 85 seconds ago and took 15							
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 envi							

Weitere Informationen finden Sie unter [Ereignis-Stream einer Elastic Beanstalk-Umgebung anzeigen \(p. 864\)](#).

Tags

Auf der Seite Tags (Tags) werden die Tags angezeigt, die Elastic Beanstalk auf die Umgebung angewendet hat, als sie erstellt wurde, sowie alle von Ihnen hinzugefügten Tags. Sie können benutzerdefinierte Tags hinzufügen, bearbeiten und löschen. Sie können die Tags, die Elastic Beanstalk angewendet hat, nicht bearbeiten oder löschen.

Umgebungs-Tags werden auf alle Ressourcen angewendet, die Elastic Beanstalk zur Unterstützung Ihrer Anwendung erstellt.

The screenshot shows the 'Tags for GettingStartedApp-env' section of the AWS Elastic Beanstalk console. It displays a table of tags with columns 'Key' and 'Value'. There are three visible rows: 'elasticbeanstalk:environment-id' with value 'e-cubmdjm6ga', 'elasticbeanstalk:environment-name' with value 'GettingStartedApp-env', and 'Name' with value 'GettingStartedApp-env'. Below the table is an 'Add tag' button and a note indicating '47 remaining' tag slots. A 'Remove' button is located on the far right.

Weitere Informationen finden Sie unter [Ressourcen in Ihren Elastic Beanstalk-Umgebungen markieren \(p. 617\)](#).

Erstellen einer Elastic Beanstalk-Umgebung

Eine AWS Elastic Beanstalk-Umgebung ist eine Sammlung von AWS-Ressourcen, die eine Anwendungsversion ausführen. Sie können mehrere Umgebungen bereitstellen, wenn Sie mehrere Versionen einer Anwendung ausführen müssen. Sie können beispielsweise Entwicklungs-, Integrations- und Produktionsumgebungen haben.

Mit dem folgenden Verfahren wird eine neue Umgebung gestartet, in der die Standardanwendung ausgeführt wird. Diese Schritte sind vereinfacht, um Ihre Umgebung mit Standardoptionswerten schnell zum Laufen zu bringen. Ausführliche Anweisungen mit Beschreibungen der vielen Optionen, mit denen Sie die Ressourcen konfigurieren können, die Elastic Beanstalk in Ihrem Auftrag bereitstellt, finden Sie unter [Der Assistent zum Erstellen einer neuen Umgebung \(p. 437\)](#).

Hinweise

- Anweisungen zum Erstellen und Verwalten von Umgebungen mit der EB CLI finden Sie unter [Elastic Beanstalk-Umgebungen mit der EB CLI verwalten \(p. 1020\)](#).
- Das Erstellen einer Umgebung erfordert die Berechtigungen in der verwalteten Elastic Beanstalk-Richtlinie für vollständigen Zugriff. Details dazu finden Sie unter [Elastic Beanstalk-Benutzerrichtlinie \(p. 24\)](#).

So starten Sie eine Umgebung mit einer Beispielanwendung (Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.

2. Wählen Sie im Navigationsbereich Applications (Anwendungen) und anschließend in der Liste den Namen einer vorhandenen Anwendung aus. Sie können auch [eine Anwendung erstellen \(p. 403\)](#).
3. Wählen Sie auf der Anwendungsübersichtsseite die Option Create a new environment (Neue Umgebung erstellen).

Environment name	Health	Date created	Last modified	URL	Platform
GettingStartedApp-env	OK	2020-01-28 12:05:50 UTC-0800	2020-01-30 15:02:35 UTC-0800	GettingStartedApp-env.bn7dx222kw.us-east-2.elasticbeanstalk.com	Tomcat running on Linux
GettingStartedApp-Windows	OK	2020-01-28 16:34:29 UTC-0800	2020-01-28 16:38:20 UTC-0800	GettingStartedApp-Windows.bn7dx222kw.us-east-2.elasticbeanstalk.com	IIS 10.0 running on Windows

4. Wählen Sie die Web server environment (Webserverumgebung) oder die Worker environment (Worker-Umgebung) als [Umgebungsebene \(p. 14\)](#) aus. Nach der Erstellung lässt sich die Umgebungsebene nicht mehr ändern.

Note

Die .NET-Plattform auf Windows Server (p. 189) unterstützt die Worker-Umgebungsschicht nicht.

Select environment tier

AWS Elastic Beanstalk has two types of environment tiers to support different types of web applications. Web server environments run web applications that listen for and then process HTTP requests, typically over port 80. Workers are specialized applications that perform a background processing task that listens for messages on an Amazon SQS queue. Worker applications post their results back to the application by using HTTP.

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](#)

5. Wählen Sie in Platform (Plattform) die Plattform und den Plattformzweig aus, die/der mit der Sprache Ihrer Anwendung übereinstimmt.

Note

Elastic Beanstalk unterstützt mehrere [Versionen \(p. 32\)](#) für die meisten Plattformen, die aufgelistet sind. Standardmäßig wählt die Konsole die empfohlene Version für die gewählte Plattform und Plattformvariante. Wenn Ihre Anwendung eine andere Version erfordert, können Sie diese hier auswählen oder [Configure more options \(Weitere Optionen konfigurieren\)](#) auswählen wie in Schritt 7 beschrieben. Informationen zu unterstützten Plattformversionen finden Sie unter [the section called “Unterstützte Plattformen” \(p. 32\)](#).

6. Wählen Sie unter Application code (Anwendungscode) die Option Sample application (Beispielanwendung) aus.
7. Wählen Sie [Configure more options \(Weitere Optionen konfigurieren\)](#) aus, um Ihre Umgebung weiter anzupassen. Sie können die folgenden Optionen nur während des Erstellens der Umgebung festlegen:
 - Environment name
 - Domänenname
 - Plattformversion
 - VPC
 - Stufe

Sie können die folgenden Einstellungen nach der Umgebungserstellung zwar ändern, jedoch müssen dafür neue Instances oder andere Ressourcen bereitgestellt werden, deren Anwendung viel Zeit in Anspruch nehmen kann:

- Instance-Typ, Root-Volumen, Schlüsselpaar und AWS Identity and Access Management (IAM)-Rolle
- Interne Amazon RDS-Datenbank
- Load Balancer

Weitere Informationen zu allen verfügbaren Einstellungen finden Sie unter [Der Assistent zum Erstellen einer neuen Umgebung \(p. 437\)](#).

8. Wählen Sie [Create environment \(Umgebung erstellen\)](#) aus.

Während Elastic Beanstalk Ihre Umgebung erstellt, werden Sie zu [Elastic Beanstalk-Konsole \(p. 426\)](#) umgeleitet. Wenn der Umgebungsstatus grün angezeigt wird, wählen Sie die URL neben dem Umgebungsnamen, um die laufende Anwendung anzuzeigen. Diese URL ist in der Regel über das Internet zugänglich, es sei denn, Sie konfigurieren Ihre Umgebung so, dass eine [benutzerdefinierte VPC mit einem internen Load Balancer \(p. 452\)](#) verwendet wird.

Themen

- [Der Assistent zum Erstellen einer neuen Umgebung \(p. 437\)](#)
- [Elastic Beanstalk-Umgebung klonen \(p. 456\)](#)
- [Elastic Beanstalk-Umgebung terminieren \(p. 459\)](#)
- [Elastic Beanstalk-Umgebungen mit der AWS CLI erstellen \(p. 460\)](#)
- [Elastic Beanstalk-Umgebungen mit der API erstellen \(p. 461\)](#)
- [Erstellen einer Launch-Now-URL \(p. 464\)](#)
- [Gruppen von Elastic Beanstalk Umgebungen erstellen oder aktualisieren \(p. 469\)](#)

Der Assistent zum Erstellen einer neuen Umgebung

Im Thema [Erstellen einer Elastic Beanstalk-Umgebung \(p. 435\)](#) zeigen wir Ihnen, wie Sie den Assistenten [Create New Environment \(Erstellen einer neuen Umgebung\)](#) öffnen und schnell eine Umgebung erstellen

können. Wählen Sie Create environment (Umgebung erstellen) aus, um eine Umgebung mit dem Standard-Umgebungsnamen, der automatisch generierten Domäne, Beispielanwendungscode und den empfohlenen Einstellungen zu starten.

In diesem Thema werden der Assistent Create New Environment (Erstellen einer neuen Umgebung) und alle Möglichkeiten zur Konfiguration der Umgebung beschrieben, die Sie erstellen möchten.

Hauptseite des Assistenten

Die Hauptseite des Assistenten Create New Environment (Assistent zum Erstellen einer neuen Umgebung) beginnt mit Informationen zum Namen der neuen Umgebung. Legen Sie den Namen und die Subdomäne der Umgebung fest und erstellen Sie eine Beschreibung für Ihre Umgebung. Beachten Sie, dass diese Einstellungen nach der Umgebungserstellung nicht mehr geändert werden können.

The screenshot shows the 'Environment information' step of the 'Create New Environment' wizard. It includes fields for Application name, Environment name, Domain, and Description, along with a 'Check availability' button.

Environment information

Choose the name, subdomain, and description for your environment. These cannot be changed later.

Application name
getting-started-app

Environment name
GettingStartedApp-env-1

Domain
Leave blank for autogenerated value .us-east-2.elasticbeanstalk

Description

Check availability

- Name – Geben Sie einen Namen für die Umgebung ein. Das Formular stellt einen generierten Namen bereit.
- Domain – (Webserverumgebungen) Geben Sie einen eindeutigen Domänennamen für Ihre Umgebung ein. Der Standardname ist der Name der Umgebung. Sie können einen anderen Domänennamen verwenden. Elastic Beanstalk verwendet diesen Namen, um einen eindeutigen CNAME für die Umgebung zu erstellen. Um zu überprüfen, ob der gewünschte Domänenname verfügbar ist, klicken Sie auf Check Availability (Verfügbarkeit prüfen).
- Description – Geben Sie eine Beschreibung für diese Umgebung ein.

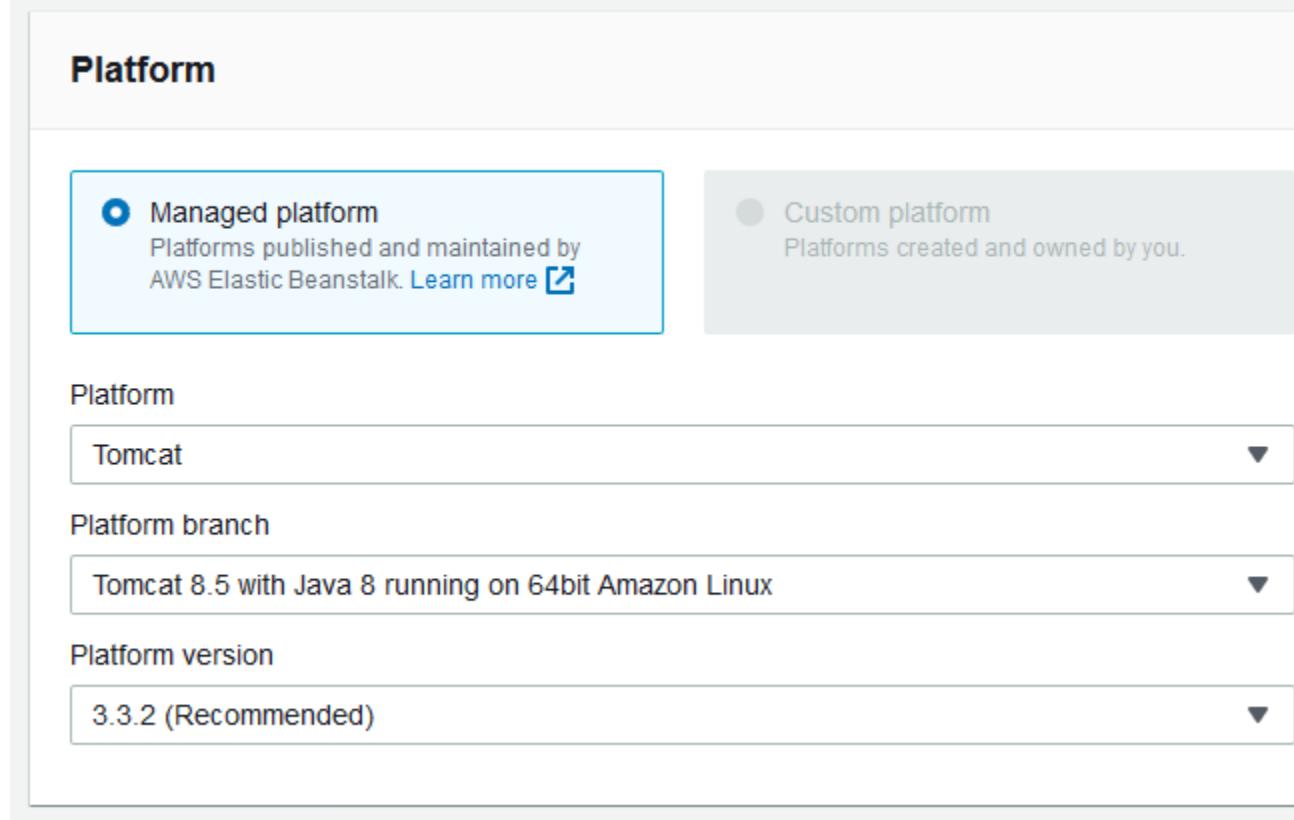
Auswählen der Plattform für die neue Umgebung

Sie können eine neue Umgebung aus zwei Arten von Plattformen erstellen:

- Verwaltete Plattform
- Benutzerdefinierte Plattform

Verwaltete Plattform

In den meisten Fällen verwenden Sie eine von Elastic Beanstalk verwaltete Plattform für Ihre neue Umgebung. Wenn der Assistent für die neue Umgebung gestartet wird, wählt dieser standardmäßig die Option Managed platform (Verwaltete Plattform) aus wie im folgenden Screenshot gezeigt.



Wählen Sie eine Plattform, einen Plattformzweig innerhalb dieser Plattform und im Zweig eine bestimmte Plattformversion aus. Wenn Sie einen Plattformzweig auswählen, wird standardmäßig die empfohlene Version innerhalb des Zweigs ausgewählt. Darüber hinaus können Sie eine Plattformversion auswählen, die Sie zuvor verwendet haben.

Note

Im Fall einer Produktionsumgebung sollten Sie eine Plattformversion in einem unterstützten Plattformzweig auswählen. Details zu den Statusarten von Plattformzweigen finden Sie in der Definition von Plattformzweig im [the section called “Glossar zu Plattformen” \(p. 27\)](#).

Benutzerdefinierte Plattform

Wenn eine vorgefertigte Plattform nicht Ihren Anforderungen entspricht, müssen Sie eine neue Umgebung aus einer benutzerdefinierten Plattform erstellen. Um eine benutzerdefinierte Plattform anzugeben, wählen Sie die Option Custom platform (Benutzerdefinierte Plattform) und anschließend eine der verfügbaren

benutzerdefinierten Plattformen aus. Wenn keine benutzerdefinierten Plattformen zur Verfügung stehen, ist diese Option nicht verfügbar.

Bereitstellen von Anwendungscode

Nachdem Sie die Plattform ausgewählt haben, stellen Sie im nächsten Schritt den Anwendungscode bereit.

The screenshot shows the 'Application code' section of the AWS Elastic Beanstalk configuration interface. It displays three options for providing code:

- Sample application**: Selected. Description: "Get started right away with sample code."
- Existing version**: Description: "Application versions that you have uploaded for getting-started-app."
- Upload your code**: Description: "Upload a source bundle from your computer or copy one from Amazon S3."

Sie haben mehrere Möglichkeiten:

- Sie können die Beispielanwendung verwenden, die Elastic Beanstalk für jede Plattform bereitstellt.
- Sie können Code verwenden, den Sie bereits für Elastic Beanstalk bereitgestellt haben. Wählen Sie Existing version (Bestehende Version) und die Anwendung im Bereich Application code (Anwendungscode) aus.
- Sie können neuen Code hochladen. Wählen Sie Upload your code (Code hochladen) und dann Upload (Hochladen) aus. Sie können den neuen Anwendungscode aus einer lokalen Datei hochladen oder die URL für den Amazon S3-Bucket mit Ihrem Anwendungscode angeben.

Note

Abhängig von der ausgewählten Plattformversion können Sie Ihre Anwendung in einem ZIP-basierten Quell-Bundle (p. 412), in einer WAR-Datei (p. 117) oder in einer Klartext-Docker-Konfiguration (p. 48) hochladen. Die maximale Dateigröße ist 512 MB.

Wenn Sie neuen Code hochladen möchten, können Sie auch Tags angeben, die mit dem neuen Code verknüpft werden sollen. Weitere Informationen zum Markieren einer Anwendungsversion finden Sie unter [the section called “Markieren von Anwendungsversionen” \(p. 410\)](#).

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

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 within the resource and is case-sensitive. [Learn more](#)

Key	Value	Remove
		<input type="button" value="Remove"/>

50 remaining

Um eine Umgebung schnell mithilfe von Standardkonfigurationsoptionen zu erstellen, können Sie jetzt Create environment (Umgebung erstellen) auswählen. Wählen Sie Configure more options (Weitere Optionen konfigurieren) aus, um zusätzliche Konfigurationsänderungen anzugeben, wie in den folgenden Abschnitten beschrieben.

Seite „Assistentenkonfiguration“

Bei Auswahl von Configure more options (Weitere Optionen konfigurieren) wird im Assistenten die Seite Configure (Konfigurieren) angezeigt. Auf dieser Seite können Sie eine voreingestellte Konfiguration auswählen, die Plattformversion ändern, die Ihre Umgebung verwenden soll, oder bestimmte Konfigurationsoptionen für die neue Umgebung auswählen.

Auswahl einer voreingestellten Konfiguration

Im Abschnitt Presets (Voreinstellungen) der Seite stellt Elastic Beanstalk verschiedene voreingestellte Konfigurationen für verschiedene Anwendungsfälle bereit. Jede Voreinstellung enthält empfohlene Werte für mehrere Konfigurationsoptionen (p. 646).

The screenshot shows the 'Configure GettingStartedApp-env-1' page. At the top, there's a breadcrumb navigation: 'Elastic Beanstalk > Applications > getting-started-app'. Below the title, the 'Presets' section is visible. It contains a heading 'Presets' and a sub-section 'Configuration presets' with the following options:

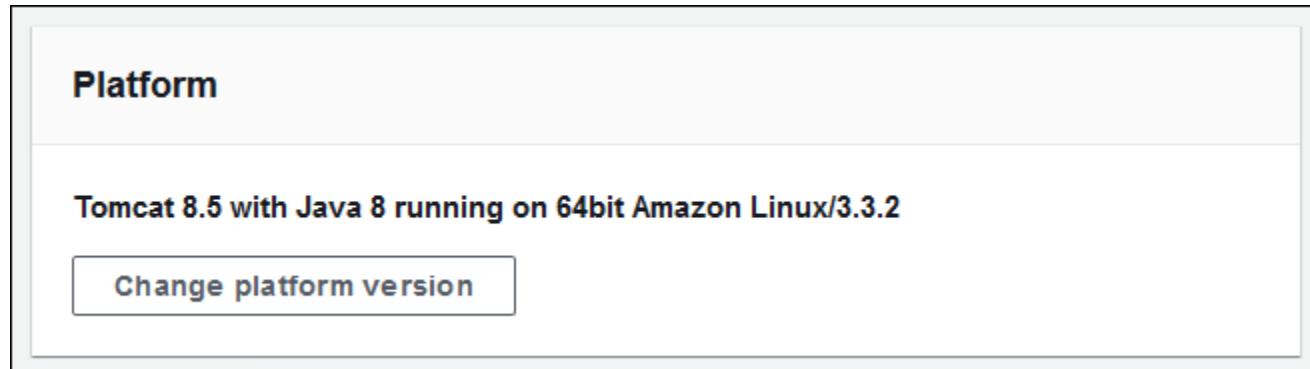
- Single instance (*Free Tier eligible*)
- Single instance (using Spot instance)
- High availability
- High availability (using Spot and On-Demand instances)
- Custom configuration

Die Voreinstellungen für High Availability (Hohe Verfügbarkeit) enthalten einen Load Balancer und werden für Produktionsumgebungen empfohlen. Wählen Sie diese aus, wenn Sie eine Umgebung wünschen, in der mehrere Instances ausgeführt werden können – für eine hohe Verfügbarkeit und Skalierbarkeit bei Lastenänderungen. Die Single Instance (Einzel-Instance)-Voreinstellungen werden in erster Linie für die Entwicklung empfohlen. Zwei der Voreinstellungen ermöglichen Spot-Instance-Anforderungen. Einzelheiten zur Kapazitätskonfiguration von Elastic Beanstalk finden Sie unter [Auto Scaling-Gruppe \(p. 540\)](#).

Die letzte Voreinstellung, Custom configuration (Benutzerdefinierte Konfiguration), entfernt alle empfohlenen Werte außer den Rolleneinstellungen und verwendet die API-Standardeinstellungen. Wählen Sie diese Option, wenn Sie ein Quell-Bundle mit [Konfigurationsdateien \(p. 722\)](#) bereitstellen, die Konfigurationsoptionen festlegen. Custom configuration (Benutzerdefinierte Konfiguration) ist auch automatisch ausgewählt, wenn Sie die Konfigurationsvoreinstellungen Low cost (Niedrige Kosten) oder High availability (Hohe Verfügbarkeit) ändern.

Änderung der Plattformversion

Im Abschnitt Platform (Plattform) der Seite können Sie die Plattformversion ändern, die Ihre neue Umgebung verwendet. Sie können in jedem Plattformzweig die empfohlene Plattformversion auswählen oder eine andere Plattformversion, die Sie früher verwendet haben.



Anpassen Ihrer Konfiguration

Zusätzlich zu (oder anstelle von) einer ausgewählten Konfigurationsvoreinstellung können Sie die [Konfigurationsoptionen \(p. 646\)](#) in Ihrer Umgebung feiner abstimmen. Der Assistent Configure (Konfigurieren) zeigt mehrere Konfigurationskategorien an. Jede Konfigurationskategorie zeigt eine Übersicht der Werte für eine Gruppe von Konfigurationseinstellungen an. Wählen Sie Edit (Bearbeiten) aus, um diese Einstellungsgruppe zu bearbeiten.

Konfigurationskategorien

- [Softwareeinstellungen \(p. 443\)](#)
- [Instances \(p. 444\)](#)
- [Kapazität \(p. 445\)](#)
- [Load Balancer \(p. 445\)](#)
- [Fortlaufende Updates und Bereitstellungen \(p. 447\)](#)
- [Sicherheit \(p. 448\)](#)
- [Überwachen \(p. 449\)](#)
- [Verwaltete Aktualisierungen \(p. 450\)](#)
- [Benachrichtigungen \(p. 451\)](#)
- [Netzwerk \(p. 452\)](#)
- [Datenbank \(p. 453\)](#)
- [Tags \(p. 454\)](#)
- [Worker-Umgebung \(p. 455\)](#)

Softwareeinstellungen

Verwenden Sie die Konfigurationsseite Modify software (Software ändern), um die Software auf den Instances von Amazon Elastic Compute Cloud (Amazon EC2) zu konfigurieren, auf denen Ihre Anwendung ausgeführt wird. Sie können Umgebungseigenschaften, AWS X-Ray-Debugging, das Speichern und Streamen von Instance-Protokollen sowie plattformspezifische Einstellungen konfigurieren. Details dazu finden Sie unter [the section called "Softwareeinstellungen" \(p. 621\)](#).

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

Instances

Auf der Modify Instances (Instanzen ändern) können Sie die Amazon EC2-Instances konfigurieren, die Ihre Anwendung ausführen. Details dazu finden Sie unter [the section called “Amazon EC2-Instances” \(p. 531\)](#).

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)

Kapazität

Auf der Konfigurationsseite Modify capacity (Kapazität ändern) können Sie die Rechenkapazität Ihrer Umgebung und die Einstellungen für Auto Scaling group (Auto Scaling-Gruppe konfigurieren, um Anzahl und Typ der von Ihnen verwendeten Instances zu optimieren. Sie können die Umgebungskapazität auch auf der Grundlage von Auslösern oder Zeitplänen ändern.

Eine Umgebung mit Lastausgleich kann mehrere Instances für hohe Verfügbarkeit ausführen und Ausfallzeiten während Konfigurations-Updates und Bereitstellungen verhindern. In einer Umgebung mit Lastausgleich entspricht der Domänenname dem Load Balancer. In einer Umgebung mit einer Instance entspricht er einer Elastic IP-Adresse auf der Instance.

Warning

Eine Umgebung mit einer einzelnen Instance ist nicht geeignet für die Produktion. Falls die Instance während der Bereitstellung instabil wird, oder Elastic Beanstalk beendet wird und die Instance während eines Konfigurations-Update neu gestartet wird, steht Ihre Anwendung möglicherweise längere Zeit nicht zur Verfügung. Umgebungen mit einer einzelnen Instanz können Sie für die Entwicklung, das Testen oder ein Staging nutzen. Für die Produktion verwenden Sie Umgebungen mit Load Balancing.

Weitere Informationen zu Umgebungskapazitätseinstellungen finden Sie unter [the section called “Auto Scaling-Gruppe” \(p. 540\)](#).

The screenshot shows the 'Modify capacity' configuration page for an environment named 'getting-started-app'. The top navigation bar shows 'Elastic Beanstalk > Applications > getting-started-app'. The main heading is 'Modify capacity'. Below it, a sub-section title 'Auto Scaling Group' is followed by the instruction 'Configure the compute capacity of your environment and Auto Scaling settings to optimize the number of instances used.' Under 'Auto Scaling Group', there are several configuration options:

- Environment type:** A dropdown menu set to 'Load balanced'.
- Instances:** A section with two input fields:
 - 'Min' is set to 1.
 - 'Max' is set to 2.
- Fleet composition:** A note stating 'Choose a mix of On-Demand and Spot Instances with multiple instance types. Spot Instances are automatically launched at the lowest available price.' Below this are two radio button options:
 - On-Demand instances
 - Combine purchase options and instances
- Maximum spot price:** A note stating '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 requests.' Below this is a link 'Custom instances'.
- Load Balancer:** A section with the text 'Wählen Sie auf der Konfigurationskarte Modify load Balancer (Load Balancer ändern) einen Load Balancer-Typ aus und konfigurieren Sie die zugehörigen Einstellungen. In einer Umgebung mit Load Balancing ist'.

der Load Balancer Ihrer Umgebung der Eintrittspunkt für den gesamten Datenverkehr, der Ihre Anwendung ansteuert. Elastic Beanstalk unterstützt verschiedene Arten von Load Balancern. Standardmäßig erstellt die Elastic Beanstalk-Konsole einen Application Load Balancer und konfiguriert ihn so, dass er HTTP-Datenverkehr auf Port 80 bedient.

Note

Sie können den Typ des Load Balancer Ihrer Umgebung nur während der Umgebungserstellung auswählen.

Weitere Informationen zu Typen und Einstellungen von Load Balancern finden Sie unter [the section called “Load Balancer” \(p. 554\)](#) und [the section called “HTTPS” \(p. 778\)](#).

Elastic Beanstalk > Applications > 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

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 specific protocol to one or more environment processes. By default, we've configured your load balancer with a standard web server on port 80.

Action

<input type="checkbox"/>	Port	Protocol	SSL certificate
<input type="checkbox"/>	80	HTTP	--

Processes

For each environment process, you can specify the protocol and port that the load balancer uses to route requests to it.

Fortlaufende Updates und Bereitstellungen

Auf der Konfigurationsseite Modify rolling updates and deployments (Rollierende Updates und Bereitstellungen) können Sie die Verarbeitung von Anwendungsbereitstellungen und Konfigurationsupdates für Ihre Umgebung durch Elastic Beanstalk konfigurieren.

Anwendungsbereitstellungen werden ausgeführt, wenn Sie ein aktuelles Anwendungsquellpaket hochladen und in Ihrer Umgebung bereitstellen. Weitere Informationen zur Konfigurierung von Bereitstellungen finden Sie unter [the section called “Optionen für die Bereitstellung” \(p. 474\)](#).

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

Falls mit Konfigurationsänderungen die [Startkonfiguration \(p. 668\)](#) oder die [VPC-Einstellungen \(p. 682\)](#) geändert werden, müssen alle Instances in der Umgebung beendet und ersetzt werden. Weitere Informationen zur Festlegung des Update-Typs und anderer Optionen finden Sie unter [the section called “Konfigurationsänderungen” \(p. 483\)](#).

Configuration updates

Changes to virtual machine settings and VPC configuration trigger rolling updates to replace the instances in your environment.

[Learn more](#)

Rolling update type

▼

Batch size

▲

The maximum number of instances to replace in each phase of the update.

Minimum capacity

▲

The minimum number of instances to keep in service at all times.

Pause time

Pause the update for up to an hour between each batch.

Sicherheit

Auf der Konfigurationsseite Modify security (Sicherheit ändern) können Sie die Sicherheitseinstellungen von Services und Instances konfigurieren.

Eine Beschreibung der Sicherheitskonzepte von Elastic Beanstalk finden Sie unter [Berechtigungen \(p. 21\)](#). Weitere Informationen zur Konfigurierung der Sicherheitseinstellungen von Umgebungen finden Sie unter [the section called “Sicherheit” \(p. 615\)](#).

Elastic Beanstalk > Applications > getting-started-app

Modify security

Service role

Service role

aws-elasticbeanstalk-service-role



Virtual machine permissions

EC2 key pair

-- Choose a key pair --



IAM instance profile

aws-elasticbeanstalk-ec2-role



Cancel

Überwachen

Auf der Konfigurationsseite Modify monitoring (Überwachung ändern) können Sie Integritätsberichte, Überwachungsregeln und das Integritätsereignisstreaming konfigurieren. Details hierzu finden Sie unter [the section called “Aktivieren der erweiterten Zustandsberichte” \(p. 832\)](#), [the section called “Regeln für den erweiterten Zustand” \(p. 845\)](#) und [the section called “Streamen des Umgebungszustands” \(p. 888\)](#).

Modify monitoring

Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and 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](#)



Verwaltete Aktualisierungen

Auf der Konfigurationsseite Modify managed updates (Verwaltete Aktualisierungen ändern) können Sie Updates für die verwaltete Plattform konfigurieren. Sie können festlegen, ob sie aktiviert werden sollen, den Zeitplan angeben und weitere Eigenschaften konfigurieren. Details dazu finden Sie unter [the section called "Verwaltete Updates" \(p. 496\)](#).

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 the 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

Benachrichtigungen

Auf der Konfigurationsseite Modify notifications (Benachrichtigungen ändern) können Sie eine E-Mail-Adresse angeben, um [E-Mail-Benachrichtigungen \(p. 633\)](#) für wichtige Ereignisse aus Ihrer Umgebung zu erhalten.

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

`user@example.com`

[Cancel](#)

[Save](#)

Netzwerk

Wenn Sie eine [benutzerdefinierte VPC](#) (p. 637) erstellt haben, können Sie Ihre Umgebung auf der Konfigurationsseite Modify network (Netzwerk ändern) für ihre Verwendung konfigurieren. Wenn Sie keine VPC auswählen, verwendet Elastic Beanstalk die Standard-VPC und Subnetze.

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.

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, choose 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

Datenbank

Verwenden Sie die Konfigurationsseite Modify database (Datenbank ändern), um eine Amazon Relational Database Service (Amazon RDS)-Datenbank zu Entwicklungs- und Testzwecken zu Ihrer Umgebung hinzuzufügen. Elastic Beanstalk bietet Verbindungsinformationen zu Ihren Instances, indem Umgebungseigenschaften für den Hostnamen, Benutzernamen, das Passwort, den Tabellennamen und den Port der Datenbank eingerichtet werden.

Details dazu finden Sie unter [the section called “Datenbank” \(p. 609\)](#).

Modify database

Add an Amazon RDS SQL database to your environment for development and testing. AWS Elastic Beanstalk provides information to your instances by setting environment properties for the database hostname, username, password, 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

Auf der Konfigurationsseite Modify tags (Tags ändern) können Sie den Ressourcen in Ihrer Umgebung Tags hinzufügen. Weitere Informationen zu Umgebungs-Tags finden Sie unter [Ressourcen in Ihren Elastic Beanstalk-Umgebungen markieren \(p. 617\)](#).

The screenshot shows the 'Modify tags' section of the AWS Elastic Beanstalk configuration interface. At the top, a breadcrumb navigation shows 'Elastic Beanstalk > Applications > getting-started-app'. Below it, the title 'Modify tags' is displayed. A sub-instruction reads 'Apply up to 50 tags to the resources in your environment in addition to the default tags.' A table is used for adding tags, with columns 'Key' and 'Value'. One row is shown: 'mytag1' in the Key column and 'value1' in the Value column. An 'Add tag' button is located below the table, and a note indicates '49 remaining' tags can still be added. On the far right, there are 'Remove' and 'Cancel' buttons.

Worker-Umgebung

Wenn Sie eine Worker-Tier-Umgebung erstellen, können Sie auf der Konfigurationsseite Modify Worker (Worker ändern) die Worker-Umgebung konfigurieren. Der Worker-Daemon auf den Instances in Ihrer Umgebung ruft Elemente aus einer Amazon Simple Queue Service (Amazon SQS)-Warteschlange ab und sendet sie als Post-Nachrichten an Ihre Worker-Anwendung. Sie können die Amazon SQS-Warteschlange auswählen, aus der der Worker-Daemon liest (automatisch generiert oder vorhanden). Sie können auch die Nachrichten konfigurieren, die der Worker-Daemon an Ihre Anwendung sendet.

Weitere Informationen finden Sie unter [the section called “Worker-Umgebungen” \(p. 516\)](#).

Modify worker

You can create a new Amazon SQS queue for your worker application or pull work items from an existing queue. The instances in your environment pulls an item from the queue and relays it in the body of a POST request to a local `localhost`.

Queue

Worker queue

Autogenerated queue



SQS queue from which to read work items.

Messages

HTTP path

Elastic Beanstalk-Umgebung klonen

Durch das Klonen einer vorhandenen Elastic Beanstalk-Umgebung können Sie diese als Grundlage für eine neue Umgebung einsetzen. Beispielsweise können Sie einen Klon erstellen, um eine neuere Version des ursprünglichen Lösungs-Stacks für die Umgebungsplattform zu verwenden. Elastic Beanstalk konfiguriert den Klon mit den gleichen Umgebungseinstellungen wie in der ursprünglichen Umgebung. Wenn Sie eine Umgebung klonen, anstatt eine neue zu erstellen, müssen Sie weder Optionseinstellungen noch Umgebungsvariablen oder andere Einstellungen manuell konfigurieren. Außerdem erstellt Elastic Beanstalk eine Kopie von jeder AWS-Ressource, die der ursprünglichen Umgebung zugeordnet ist. Während des Klonvorgangs kopiert Elastic Beanstalk jedoch keine Daten von Amazon RDS in den Klon. Nach der Erstellung der Klonumgebung können Sie die Einstellungen der Umgebungskonfiguration nach Bedarf anpassen.

Sie können eine Umgebung nur auf eine andere Plattformversion desselben Plattformzweigs klonen. Ein anderer Plattformzweig ist nicht garantiert kompatibel. Um einen anderen Plattformzweig zu verwenden, müssen Sie manuell eine neue Umgebung erstellen, Ihren Anwendungscode bereitstellen und alle erforderlichen Änderungen an Code und Optionen vornehmen, um sicherzustellen, dass Ihre Anwendung auf dem neuen Plattformzweig ordnungsgemäß funktioniert.

Note

Elastic Beanstalk bindet keine nicht verwalteten Ressourcenänderungen in den Klon ein. Änderungen an AWS-Ressourcen, die Sie mit anderen Tools als der Elastic Beanstalk-Konsole, den Befehlszeilen-Tools oder der API vorgenommen haben, gelten als nicht verwaltete Änderungen.

AWS Management Console

So klonen Sie eine Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Übersichtsseite der Umgebung Environment actions (Umgebungsaktionen) aus. Führen Sie dann eine der folgenden Aktionen aus:
 - Wählen Sie Clone Environment (Umgebung klonen) aus, um die Umgebung ohne Änderungen an der Plattformversion zu klonen.
 - Wählen Sie Clone with Latest Platform (Mit neuester Plattform klonen), um die Umgebung mit einer neueren Version des Plattformzweigs der ursprünglichen Umgebung zu klonen.
4. Überprüfen Sie auf der Seite Clone Environment (Umgebung klonen) anhand der Informationen unter Original Environment (Ursprüngliche Umgebung), ob Sie die korrekte Umgebung für die Klonerstellung ausgewählt haben.
5. Optional können Sie im Abschnitt New Environment (Neue Umgebung) die von Elastic Beanstalk automatisch auf Basis der ursprünglichen Umgebung festgelegten Werte für Environment name (Umgebungsname), Environment URL (Umgebungs-URL), Description (Beschreibung), Platform (Plattform) und Service role (Servicerolle) ändern.

Note

Wenn es sich bei der in der ursprünglichen Umgebung verwendeten Plattformversion nicht um die für die Verwendung in der Plattformvariante empfohlene Version handelt, werden Sie gewarnt, dass eine andere Plattformversion empfohlen wird. Wählen Sie Platform version (Plattformversion), damit die empfohlene Plattformversion in der Liste angezeigt wird, z. B.3.3.2 (Recommended) (3.3.2 (Empfohlen)).

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 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

[Check availability](#)

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.

458

6. Wenn Sie fertig sind, klicken Sie auf Clone (Klonen).

Elastic Beanstalk Command Line Interface (EB CLI)

Mit dem Befehl eb clone klonen Sie eine ausgeführte Umgebung, wie hier gezeigt.

```
~/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
```

Sie können im Klonbefehl den Namen der Quellumgebung angeben. Wenn Sie dies nicht tun, wird die Standardumgebung des aktuellen Projektordners geklont. Sie werden von der EB CLI aufgefordert, einen Namen und ein DNS-Präfix für die neue Umgebung anzugeben.

Standardmäßig wird die neue Umgebung von eb clone mit der neuesten verfügbaren Version der Quellumgebungsplattform erstellt. Mit der Option --exact erzwingen Sie – selbst wenn eine neuere Version vorhanden ist – in der EB CLI die Verwendung der gleichen Version.

```
~/workspace/my-app$ eb clone --exact
```

Weitere Informationen zu diesem Befehl finden Sie unter [eb clone \(p. 1047\)](#).

Elastic Beanstalk-Umgebung terminieren

Sie können eine laufende AWS Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-Konsole terminieren, um zu vermeiden, dass Gebühren für ungenutzte AWS-Ressourcen anfallen. Weitere Informationen zum Beenden einer Umgebung mit dem AWS Toolkit for Eclipse finden Sie unter [Beenden einer Umgebung \(p. 155\)](#).

Note

Sie können zu einem späteren Zeitpunkt problemlos eine neue Umgebung mit derselben Version starten. Sofern Sie Daten aus einer Umgebung behalten möchten, erstellen Sie einen Snapshot der aktuellen DB-Instance, bevor Sie die Umgebung beenden. Diesen können Sie später bei der Erstellung einer neuen Umgebung als Basis für die neue DB-Instance nutzen. Weitere Informationen finden Sie unter [Erstellen von DB-Snapshots im Amazon Relational Database Service-Benutzerhandbuch](#).

Elastic Beanstalk kann Ihre Umgebung möglicherweise nicht terminieren. Ein typischer Grund für diesen Fehler besteht darin, dass eine andere Sicherheitsgruppe der Umgebung von der Sicherheitsgruppe der Umgebung abhängig ist, die beendet werden soll. Eine Methode zur Vermeidung dieses Problems wird unter [Sicherheitsgruppen \(p. 535\)](#) auf der Seite EC2-Instances dieses Handbuchs beschrieben.

Elastic Beanstalk-Konsole

So beenden Sie eine Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Note

Nachdem die Umgebung beendet ist, steht ihr zugehöriger CNAME wieder zur Verfügung und kann von allen verwendet werden.

Es dauert einige Minuten, bis Elastic Beanstalk die AWS-Ressourcen beendet hat, die in der Umgebung ausgeführt werden.

AWS-CLI

So beenden Sie eine Umgebung

- Führen Sie den folgenden Befehl aus.

```
$ aws elasticbeanstalk terminate-environment --environment-name my-env
```

API

So beenden Sie eine Umgebung

- Rufen Sie `TerminateEnvironment` mit folgendem Parameter auf:

`EnvironmentName = SampleAppEnv`

```
https://elasticbeanstalk.us-west-2.amazonaws.com/?EnvironmentName=SampleAppEnv&Operation=TerminateEnvironment&AuthParams
```

Elastic Beanstalk-Umgebungen mit der AWS CLI erstellen

1. Überprüfen Sie, ob der CNAME für die Umgebung verfügbar ist.

```
$ aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
{
    "Available": true,
    "FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"
}
```

2. Stellen Sie sicher, dass die Anwendungsversion vorhanden ist.

```
$ aws elasticbeanstalk describe-application-versions --application-name my-app --version-label v1
```

Wenn Sie noch nicht über eine Anwendungsversion Ihrer Quelle verfügen, erstellen Sie sie. Der folgende Befehl erstellt beispielsweise eine Anwendungsversion aus einem Quell-Bundle im 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. Erstellen Sie eine Konfigurationsvorlage für die Anwendung.

```
$ 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. Erstellen Sie die Umgebung.

```
$ 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
```

Optionseinstellungen sind in der Datei options.txt definiert:

```
[  
 {  
     "Namespace": "aws:autoscaling:launchconfiguration",  
     "OptionName": "IamInstanceProfile",  
     "Value": "aws-elasticbeanstalk-ec2-role"  
 }  
]
```

Die oben genannte Optionseinstellung definiert das IAM-Instance-Profil. Sie können die ARN oder den Profilnamen angeben.

5. Stellen Sie fest, ob die neue Umgebung grün und einsatzbereit ist.

```
$ aws elasticbeanstalk describe-environments --environment-names my-env
```

Wenn die neue Umgebung nicht grün und einsatzbereit ist, sollten Sie entscheiden, ob Sie den Vorgang wiederholen oder die Umgebung im aktuellen Zustand zur Untersuchung lassen möchten. Achten Sie darauf, die Umgebung zu beenden, nachdem Sie fertig sind, und nicht verwendete Ressourcen zu bereinigen.

Note

Sie können die Zeitüberschreitung anpassen, wenn die Umgebung nicht in einem angemessenen Zeitraum startet.

Elastic Beanstalk-Umgebungen mit der API erstellen

1. Rufen Sie CheckDNSAvailability mit folgendem Parameter auf:

- CNAMEPrefix = SampleApp

Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?CNAMEPrefix=sampleapplication  
&Operation=CheckDNSAvailability  
&AuthParams
```

2. Rufen Sie DescribeApplicationVersions mit folgenden Parametern auf:

- ApplicationName = SampleApp
- VersionLabel = Version2

Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&VersionLabel=Version2
&Operation=DescribeApplicationVersions
&AuthParams
```

3. Rufen Sie CreateConfigurationTemplate mit folgenden Parametern auf:

- 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. Rufen Sie CreateEnvironment mit einem der folgenden Parametersätze auf.

- a. Verwenden Sie Folgendes für eine Webserverumgebungsebene:

- 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. Verwenden Sie Folgendes für eine Worker-Umgebungsebene:

- 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:sqsd
- OptionSettings.member.2.OptionName = WorkerQueueURL
- OptionSettings.member.2.Value = sqsd.elasticbeanstalk.us-east-2.amazonaws.com
- OptionSettings.member.3.Namespace = aws:elasticbeanstalk:sqsd
- OptionSettings.member.3.OptionName = HttpPath
- OptionSettings.member.3.Value = /
- OptionSettings.member.4.Namespace = aws:elasticbeanstalk:sqsd
- OptionSettings.member.4.OptionName = MimeType
- OptionSettings.member.4.Value = application/json
- OptionSettings.member.5.Namespace = aws:elasticbeanstalk:sqsd
- OptionSettings.member.5.OptionName = HttpConnections
- OptionSettings.member.5.Value = 75
- OptionSettings.member.6.Namespace = aws:elasticbeanstalk:sqsd
- OptionSettings.member.6.OptionName = ConnectTimeout
- OptionSettings.member.6.Value = 10
- OptionSettings.member.7.Namespace = aws:elasticbeanstalk:sqsd
- OptionSettings.member.7.OptionName = InactivityTimeout
- OptionSettings.member.7.Value = 10
- OptionSettings.member.8.Namespace = aws:elasticbeanstalk:sqsd
- OptionSettings.member.8.OptionName = VisibilityTimeout
- OptionSettings.member.8.Value = 60
- OptionSettings.member.9.Namespace = aws:elasticbeanstalk:sqsd
- 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
463
&OptionSettings.member.1.OptionName=IamInstanceProfile
&OptionSettings.member.1.Value=aws-elasticbeanstalk-ec2-role
```

```
&OptionSettings.member.2.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.2.OptionName=WorkerQueueURL
&OptionSettings.member.2.Value=sqsd.elasticbeanstalk.us-east-2.amazonaws.com
&OptionSettings.member.3.Namespace=aws%3Aelasticbeanstalk%3sqsd
&OptionSettings.member.3.OptionName=HttpPath
&OptionSettings.member.3.Value=%2F
&OptionSettings.member.4.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.4.OptionName=MimeType
&OptionSettings.member.4.Value=application%2Fjson
&OptionSettings.member.5.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.5.OptionName=HttpConnections
&OptionSettings.member.5.Value=75
&OptionSettings.member.6.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.6.OptionName=ConnectTimeout
&OptionSettings.member.6.Value=10
&OptionSettings.member.7.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.7.OptionName=InactivityTimeout
&OptionSettings.member.7.Value=10
&OptionSettings.member.8.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.8.OptionName=VisibilityTimeout
&OptionSettings.member.8.Value=60
&OptionSettings.member.9.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.9.OptionName=RetentionPeriod
&OptionSettings.member.9.Value=345600
&AuthParams
```

Erstellen einer Launch-Now-URL

Sie können eine benutzerdefinierte URL (Uniform Resource Locator) erstellen, damit jeder Benutzer eine vordefinierte Webanwendung schnell in AWS Elastic Beanstalk bereitstellen und ausführen kann. Diese URL wird als Launch-Now-URL bezeichnet. Mit einer Launch-Now-URL können Sie zum Beispiel eine Webanwendung demonstrieren, die zur Ausführung auf Elastic Beanstalk erstellt wurde. Mit einer Launch-Now-URL können Sie Parameter verwenden, um die erforderlichen Informationen zum Assistenten zum Erstellen von Anwendungen im Vorfeld hinzuzufügen. Wenn Sie dies tun, kann jeder den URL-Link verwenden, um eine Elastic Beanstalk-Umgebung mit Ihrer Webanwendungsquelle in nur wenigen Schritten zu starten. Das bedeutet, dass Benutzer den Speicherort des Anwendungs-Quell-Bundles nicht manuell hochladen oder angeben oder zusätzliche Eingaben für den Assistenten bereitstellen müssen.

Eine Launch-Now-URL liefert Elastic Beanstalk die minimalen Informationen, die erforderlich sind, um eine Anwendung zu erstellen: Anwendungsnamen, Lösungs-Stack, Instance-Typ und Umgebungstyp. Elastic Beanstalk nutzt Standardwerte für andere Konfigurationsdetails, die nicht explizit in Ihrer benutzerdefinierten Launch-Now-URL angegeben wurden.

Eine Launch-Now-URL verwendet die Standard-URL-Syntax. Weitere Informationen finden Sie unter [RFC 3986 - Uniform Resource Identifier \(URI\): Generic Syntax](#).

URL-Parameter

Die URL muss die folgenden Parameter enthalten, bei denen die Groß- und Kleinschreibung beachtet werden muss:

- **region** (Region) – Geben Sie eine AWS-Region an. Eine Liste der Regionen, die von Elastic Beanstalk unterstützt werden, finden Sie unter [AWS Elastic Beanstalk-Endpunkte und -Kontingente](#) in der Allgemeinen AWS-Referenz.
- **applicationName** – Geben Sie den Namen Ihrer Anwendung an. Elastic Beanstalk zeigt den Anwendungsnamen in der Elastic Beanstalk-Konsole an, um sie von anderen Anwendungen zu unterscheiden. Standardmäßig bildet der Name der Anwendung auch die Basis für den Umgebungsnamen und die Umgebungs-URL.

- platform (Plattform) – Geben Sie die Plattformversion für die Umgebung an. Nutzen Sie dazu eine der folgenden Methoden und codieren Sie Ihre Auswahl dann in einer URL:
 - Geben Sie einen Plattform-ARN ohne Version an. Elastic Beanstalk wählt die neueste Plattformversion der entsprechenden Plattform-Hauptversion aus. Um beispielsweise die neueste Python 3.6-Plattformversion auszuwählen, geben Sie Folgendes an:

```
Python 3.6 running on 64bit Amazon Linux
```

- Geben Sie den Plattformnamen an. Elastic Beanstalk wählt die neueste Plattformversion der neuesten Sprach-Laufzeitumgebung der Plattform aus. Beispiel:

```
Python
```

Eine Beschreibung aller verfügbaren Plattformen und deren Versionen finden Sie unter [Von Elastic Beanstalk unterstützte Plattformen \(p. 32\)](#).

Sie können die [AWS-Befehlszeilenschnittstelle](#) verwenden (AWS CLI), um eine Liste der verfügbaren Plattformversionen mit ihren jeweiligen ARNs zu erhalten. Der `list-platform-versions`-Befehl listet detaillierte Informationen über alle verfügbaren Plattformversionen auf. Das `--filters`-Argument gestattet Ihnen, die Liste einzuzgrenzen. Sie können beispielsweise alle Plattformversionen einer bestimmten Sprache auflisten.

Das folgende Beispiel fragt alle Python-Plattformversionen ab und übergibt die Ausgabe über eine Reihe von Befehlen. Das Ergebnis ist eine Liste der Plattformversions-ARNs (ohne den `/version`-Schwanz) in einem für Menschen lesbaren Format ohne URL-Kodierung.

```
$ 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
```

Das folgenden Beispiel fügt dem letzten Beispiel einen Perl-Befehl hinzu, um die Ausgabe als URL zu codieren.

```
$ 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
```

Eine Launch-Now-URL kann optional die folgenden Parameter enthalten. Wenn Sie die optionalen Parameter nicht in Ihre Launch-Now-URL einschließen, verwendet Elastic Beanstalk Standardwerte zum Erstellen und Ausführen Ihrer Anwendung. Wenn Sie den `sourceBundleUrl`-Parameter nicht einbeziehen, nutzt Elastic Beanstalk die Standard-Beispielanwendung für die angegebene `platform` (Plattform).

- `sourceBundleUrl` – Geben Sie den Speicherort Ihres Webanwendungs-Quell-Bundle im URL-Format an. Wenn Sie zum Beispiel Ihren Quell-Bundle in einen Amazon S3-Bucket hochgeladen haben, können Sie den Wert des `sourceBundleUrl`-Parameters als <https://mybucket.s3.amazonaws.com/myobject> angeben.

Note

Sie können den Wert des sourceBundleUrl-Parameters als eine HTTP-URL angeben, der Webbrowser des Benutzers konvertiert jedoch Zeichen nach Bedarf, indem HTML-URL-Codierung angewendet wird.

- environmentType – Gibt an, ob es sich um eine skalierbare Umgebung mit Lastausgleich oder um eine Single-Instance-Umgebung handelt. Weitere Informationen finden Sie unter [Umgebungstypen \(p. 514\)](#). Sie können entweder LoadBalancing oder SingleInstance als Parameterwert angeben.
- tierName – Geben Sie an, ob die Umgebung eine Webanwendung unterstützt, die Webanfragen oder eine Webanwendung unterstützt, die Hintergrundaufträge ausführt. Weitere Informationen finden Sie unter [AWS Elastic Beanstalk-Worker-Umgebungen \(p. 516\)](#). Sie können entweder WebServer oder Worker angeben.
- instanceType – Geben Sie einen Server mit den Merkmalen (einschließlich Speichergröße und CPU-Leistung) an, die für Ihre Anwendung am besten geeignet sind. Die Instance-Typen, die in Ihrer Elastic Beanstalk-Region verfügbar sind, finden Sie unter [InstanceType \(p. 670\)](#) im Thema [Konfigurationsoptionen \(p. 646\)](#). Detaillierte Spezifikationen für die einzelnen Amazon EC2-Instance-Typen finden Sie unter [Instance Types](#).
- withVpc – Geben Sie an, ob die Umgebung in einer Amazon VPC erstellt werden soll. Sie können entweder true oder false angeben. Weitere Informationen zum Verwenden von AWS Elastic Beanstalk mit Amazon VPC finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon VPC \(p. 988\)](#).
- withRds – Geben Sie an, ob eine Amazon RDS-Datenbank-Instance mit dieser Umgebung erstellt werden soll. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon RDS \(p. 973\)](#). Sie können entweder true oder false angeben.
- rdsDBEngine – Geben Sie die Datenbank-Engine an, die Sie für Ihre Amazon EC2-Instances in dieser Umgebung verwenden möchten. Sie können mysql, oracle-sel, sqlserver-ex, sqlserver-web oder sqlserver-se festlegen. Der Standardwert ist mysql.
- rdsDBAllocatedStorage – Geben Sie die zugewiesene Größe des Datenbankspeichers in Gigabyte an. Sie können die folgenden Werte angeben:
 - MySQL – 5 bis 1024. Der Standardwert ist 5.
 - Oracle – 10 to 1024. Der Standardwert ist 10.
 - Microsoft SQL Server Express Edition – 30.
 - Microsoft SQL Server Web Edition – 30.
 - Microsoft SQL Server Standard Edition – 200.
- rdsDBInstanceClass – Geben Sie den Datenbank-Instance-Typen an. Der Standardwert ist db.t2.micro (db.m1.large für eine Umgebung, die nicht in einer Amazon VPC ausgeführt wird). Eine Liste der von Amazon RDS unterstützten Datenbank-Instance-Klassen finden Sie unter [DB-Instance-Klasse](#) im Amazon Relational Database Service-Benutzerhandbuch.
- rdsMultiAZDatabase – Geben Sie an, ob Elastic Beanstalk die Datenbank-Instance über mehrere Availability Zones hinweg erstellen muss. Sie können entweder true oder false angeben. Weitere Informationen zu Multi-AZ-Bereitstellungen mit Amazon Relational Database Service (RDS), finden Sie unter [Regionen und Availability Zones](#) im Amazon Relational Database Service-Benutzerhandbuch.
- rdsDBDeletionPolicy – Geben Sie an, ob die Datenbank-Instance beim Beenden der Umgebung gelöscht wird oder ein Snapshot davon erstellt werden soll. Sie können entweder Delete oder Snapshot angeben.

Beispiel

Es folgt ein Beispiel für eine Launch-Now-URL. Nachdem Sie Ihre eigene erstellt haben, können Sie sie an Ihre Benutzer weitergeben. Sie können z. B. die URL auf einer Webseite oder in Schulungsmaterial einbetten. Wenn Benutzer eine Anwendung mit der Launch-Now-URL erstellen, benötigt der Elastic Beanstalk-Assistent zum Erstellen einer Anwendung keine zusätzlichen Eingaben.

```
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&withRds=t
```

Wenn ein Benutzer auf eine Launch-Now-URL klickt, zeigt Elastic Beanstalk eine Seite ähnlich der folgenden an.



Create a web app

Create a new application and environment with a sample application or your own code. By creating an environment, AWS Elastic Beanstalk manages AWS resources and permissions on your behalf. [Learn more](#)

Application information

Application name

YourCompanySampleApp

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

Yourcompanysampleapp-env

Domain

Leave blank for autogenerated value

.us-east-1.elasticbeanstalk.com

[Check availability](#)

Description

Base configuration

Tier Web Server ([Choose tier](#))

Platform

Preconfigured platform

Platforms published and maintained by AWS Elastic Beanstalk.

PHP

Custom platform NEW

Platforms created and owned by you. [Learn more](#)

-- Choose a custom platform --

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.

468

[Upload](#)

ZIP or WAR

So verwenden Sie die Launch-Now-URL

1. Wählen Sie die Launch-Now-URL aus.
2. Wenn die Elastic Beanstalk-Konsole geöffnet wird, wählen Sie auf der Seite Create a web app (Webanwendung erstellen) die Option Review and launch (Prüfen und starten) aus, um die Einstellungen anzuzeigen, die Elastic Beanstalk zum Erstellen der Anwendung und Starten der Umgebung für die Anwendungsausführung verwendet.
3. Wählen Sie auf der Seite Configure (Konfigurieren) die Option Create app (Anwendung erstellen) aus, um die Anwendung zu erstellen.

Gruppen von Elastic Beanstalk Umgebungen erstellen oder aktualisieren

Mit dem AWS Elastic Beanstalk Compose Environments API können Sie Gruppen von Elastic Beanstalk-Umgebungen innerhalb einer einzigen Anwendung erstellen und aktualisieren. Jede Umgebung in der Gruppe kann eine separate Komponente einer serviceorientierten Architekturanwendung ausführen. Die Compose Environments-API nimmt eine Liste der Anwendungsversionen und einen optionalen Gruppennamen. Elastic Beanstalk erstellt eine Umgebung für jede Anwendungsversion oder, wenn die Umgebungen bereits vorhanden ist, stellt die Anwendungsversionen darin bereit.

Erstellen Sie Links zwischen Elastic Beanstalk-Umgebungen zur Benennung einer Umgebung in Abhängigkeit von einer anderen. Beim Erstellen einer Gruppe von Umgebungen mit der Compose Environments-API erstellt Elastic Beanstalk abhängige Umgebungen nur, nachdem deren Abhängigkeiten aktiv sind. Weitere Informationen zu Umgebungs-Links finden Sie unter [Verknüpfungen zwischen Elastic Beanstalk-Umgebungen erstellen \(p. 524\)](#).

Die Compose Environments-API nutzt ein [Umgebungsmanifest \(p. 771\)](#) zum Speichern von Konfigurationsdetails, die von Gruppen von Umgebungen gemeinsam verwendet werden. Jede Komponentenanwendung muss über eine `env.yaml`-Konfigurationsdatei im Quell-Bundle der Anwendung verfügen, in der die Parameter angegeben werden, mit denen die Umgebung erstellt wird.

Compose Environments erfordert, dass `EnvironmentName` und `SolutionStack` im Umgebungsmanifest für jede Komponentenanwendung angegeben werden.

Sie können die Compose Environments-API mit der Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI), der AWS CLI oder einer SDK verwenden. EB CLI-Anweisungen finden Sie unter [Verwalten mehrerer Elastic Beanstalk-Umgebungen als Gruppe mit der EB CLI \(p. 1038\)](#).

Verwenden der Compose Environments-API

Beispielsweise könnten Sie eine Anwendung namens `Media Library` erstellen, mit der Benutzer Bilder und Videos hochladen und verwalten können, die in Amazon Simple Storage Service (Amazon S3) gespeichert sind. Die Anwendung verfügt über eine Frontend-Umgebung, `front`, für die Ausführung einer Webanwendung, mit der Benutzer einzelne Dateien hoch- und herunterladen, ihre Bibliothek anzeigen Batch-Verarbeitungsjobs initiieren können.

Anstatt die Jobs direkt zu verarbeiten, fügt die Frontend-Anwendung Jobs zu einer Amazon SQS-Warteschlange hinzu. Die zweite Umgebung, `worker` zieht Jobs aus der Warteschlange und verarbeitet sie. `worker` verwendet einen G2-Instance-Typ, der eine Hochleistungs-GPU hat, während `front` auf einem kosteneffizienteren, allgemeinen Instance-Type ausgeführt werden kann.

Sie organisieren den Projektordner `Media Library` in separate Verzeichnisse für jede Komponente, wobei jedes Verzeichnis eine Umgebungs-Definitionsdatei (`env.yaml`) mit den Quellcode enthält:

```
~/workspace/media-library
```

```
|-- front
|   '-- env.yaml
`-- worker
    '-- env.yaml
```

Die folgenden Auflistung zeigt die env.yaml-Datei für jede Komponentenanwendung.

~/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
```

Nach dem [Erstellen einer Anwendungsversion \(p. 406\)](#) für die Frontend- (front-v1) und Worker- (worker-v1) Anwendungskomponenten, die Sie mit der Compose Environments-API mit den Versionsnamen erstellen. In diesem Beispiel verwenden wir die AWS-Befehlszeilenschnittstelle zum Aufrufen der 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

```

Die dritte Anfrage erstellt zwei Umgebungen, `front-dev` und `worker-dev`. Die API erstellt die Namen der Umgebungen, indem der `EnvironmentName`, der in der `env.yaml`-Datei angegeben ist, mit der `group name`-Option im `Compose Environments`-Aufruf angegeben wird, getrennt durch einen Bindestrich. Die Gesamtlänge dieser beiden Optionen und der Bindestrich dürfen nicht größer als die maximal zulässige Umgebungsnamenlänge von 23 Zeichen.

Die Anwendung in der `front-dev`-Umgebung kann auf den Namen der Amazon SQS-Warteschlange zugreifen, die der `worker-dev`-Umgebung angefügt ist, indem die `WORKERQUEUE`-Variable gelesen wird. Weitere Informationen zu Umgebungs-Links finden Sie unter [Verknüpfungen zwischen Elastic Beanstalk-Umgebungen erstellen \(p. 524\)](#).

Anwendungen in Elastic Beanstalk-Umgebungen bereitstellen

Sie können die AWS Elastic Beanstalk-Konsole verwenden, um ein aktualisiertes [Quell-Bundle \(p. 412\)](#) hochzuladen und in Ihrer Elastic Beanstalk-Umgebung bereitzustellen oder eine zuvor hochgeladene Version erneut bereitzustellen.

Jede Bereitstellung wird mit einer Bereitstellungs-ID identifiziert. Bereitstellungs-IDs beginnen bei 1 und werden mit jeder Bereitstellung und Instance-Konfigurationsänderung um eins erhöht. Wenn Sie [erweiterte Zustandsberichte \(p. 824\)](#) aktiviert haben, zeigt Elastic Beanstalk die Bereitstellungs-ID in der [Zustandskonsole \(p. 835\)](#) und in der [EB CLI \(p. 1033\)](#) an, wenn der Instance-Zustand gemeldet wird. Mit der Bereitstellungs-ID können Sie den Zustand Ihrer Umgebung bestimmen, wenn ein fortlaufendes Update fehlschlägt.

Elastic Beanstalk stellt mehrere Bereitstellungsrichtlinien und -einstellungen zur Verfügung. Weitere Informationen zum Konfigurieren einer Richtlinie und zusätzlicher Einstellungen finden Sie unter [the section called “Optionen für die Bereitstellung” \(p. 474\)](#). In der folgenden Tabelle sind die Richtlinien und die Arten von Umgebungen aufgeführt, die sie unterstützen.

Unterstützte Bereitstellungsrichtlinien

Bereitstellungsrichtlinie	Umgebungen mit Load Balancing	Umgebungen mit einer Instance	Ältere Windows Server-Umgebungen†
Alle auf einmal	✓ Ja	✓ Ja	✓ Ja
Fortlaufend	✓ Ja	✗ Nein	✓ Ja
Fortlaufend mit zusätzlichem Stapel	✓ Ja	✗ Nein	✗ Nein
Unveränderlich	✓ Ja	✓ Ja	✗ Nein

Bereitstellungsrichtlinie	Umgebungen mit Load Balancing	Umgebungen mit einer Instance	Ältere Windows Server-Umgebungen†
Datenverkehrsaufteilung	✓ Ja (Application Load Balancer)	X Nein	X Nein

† In dieser Tabelle ist eine Legacy-Windows Server-Umgebung eine Umgebung, die auf einer [Windows Server-Plattformkonfiguration](#) mit einer früheren IIS-Version als IIS 8.5 basiert.

Warning

Einige Richtlinien ersetzen alle Instances während der Bereitstellung oder Aktualisierung. Dadurch gehen alle angesammelten [Amazon EC2-Steigerungsguthaben](#) verloren. Dies geschieht in den folgenden Fällen:

- Verwaltete Plattform-Updates mit aktiviertem Instance-Ersatz
- Unveränderliche Updates
- Bereitstellungen mit unveränderlichen Updates oder Datenverkehrsaufteilung aktiviert

Auswählen einer Bereitstellungsrichtlinie

Die Auswahl der richtigen Bereitstellungsrichtlinie für Ihre Anwendung ist eine Abstimmung einiger Überlegungen und hängt von Ihren speziellen Anforderungen ab. Die Seite [the section called “Optionen für die Bereitstellung” \(p. 474\)](#) enthält weitere Informationen zu den einzelnen Richtlinien und eine detaillierte Beschreibung der Funktionsweise einiger von ihnen.

Die folgende Liste bietet zusammenfassende Informationen zu den verschiedenen Bereitstellungsrichtlinien und zugehörige Überlegungen.

- **Alles auf einmal** – Die schnellste Bereitstellungsmethode. Geeignet, wenn Sie einen kurzen Serviceverlust akzeptieren können und eine schnelle Bereitstellung für Sie wichtig ist. Mit dieser Methode stellt Elastic Beanstalk die neue Anwendungsversion für jede Instance bereit. Anschließend muss der Webproxy oder der Anwendungsserver möglicherweise neu gestartet werden. Infolgedessen könnte Ihre Anwendung für kurze Zeit für Benutzer nicht verfügbar sein (oder eine geringe Verfügbarkeit haben).
- **Fortlaufend** – Vermeidet Ausfallzeiten und minimiert Einschränkungen bei der Verfügbarkeit auf Kosten einer längeren Bereitstellungszeit. Geeignet, wenn Sie keinerlei vollständigen Serviceverlust akzeptieren können. Mit dieser Methode wird Ihre Anwendung in Ihrer Umgebung in je einem Stapel von Instances bereitgestellt. Während der gesamten Bereitstellung wird die meiste Bandbreite beibehalten.
- **Fortlaufend mit zusätzlichem Stapel** – Vermeidet jegliche Verfügbarkeitseinschränkung auf Kosten einer noch längeren Bereitstellungszeit als bei der Methode Fortlaufend. Geeignet, wenn Sie während der gesamten Bereitstellung die gleiche Bandbreite beibehalten müssen. Mit dieser Methode startet Elastic Beanstalk einen zusätzlichen Stapel von Instances und führt dann eine fortlaufende Bereitstellung durch. Das Starten des zusätzlichen Stapels erfordert Zeit und stellt sicher, dass die gleiche Bandbreite während der gesamten Bereitstellung beibehalten wird.
- **Unveränderlich** – Eine langsamere Bereitstellungsmethode, mit der Ihre neue Anwendungsversion immer auf neuen Instances bereitgestellt wird, anstatt vorhandene Instances zu aktualisieren. Sie bietet darüber hinaus den Vorteil eines schnellen und sicheren Rollbacks für den Fall, dass die Bereitstellung fehlschlägt. Mit dieser Methode führt Elastic Beanstalk ein [unveränderliches Update \(p. 488\)](#) durch, um Ihre Anwendung bereitzustellen. In einem unveränderbaren Update wird eine zweite Auto Scaling-Gruppe in Ihrer Umgebung gestartet und die neue Version verarbeitet Datenverkehr zusammen mit der alten Version, bis die neue Instance die Zustandsprüfungen besteht.
- **Datenverkehrsaufteilung** – Eine Bereitstellungsmethode für Canary-Tests. Geeignet, wenn Sie die Integrität Ihrer neuen Anwendungsversion mit einem Teil des eingehenden Datenverkehrs testen möchten, während der Rest des Datenverkehrs, der von der alten Anwendungsversion bedient wird, beibehalten werden soll.

In der folgenden Tabelle werden die Eigenschaften der Bereitstellungsmethoden verglichen.

Bereitstellungsmethoden

Art	Auswirkungen einer fehlgeschlagenen Bereitstellung	Bereitstellungszeit	Ohne Ausfa	Keine DNS-Änder	Rollba	Vorga	Code bereitgestellt für
Alle auf einmal	Ausfallzeit	⌚	X Nein	✓ Ja	Erneut vorhandene manuelle Instances	Bereitstellung	
Fortlaufend	Einzelne Stapel außer Betrieb; alle erfolgreichen Stapel vor dem Ausfall werden mit neuer Anwendungsversion ausgeführt	⌚⌚†	✓ Ja	✓ Ja	Erneut vorhandene manuelle Instances	Bereitstellung	
Fortlaufend	Einmal, wenn der erste Stapel fehlschlägt, mit ansonsten ähnlich wie Rolling (Fortlaufend) zusätzlichem Stapel	⌚⌚⌚†	✓ Ja	✓ Ja	Erneut neue manuelle Instances	Bereitstellung	
Unverändert	Möglichkeit	⌚⌚⌚⌚	✓ Ja	✓ Ja	Neue Instances beenden	Neue Instances	
Datenverlust	Pausenzeit des Clientdatenverkehrs, der an die vorübergehend beeinträchtigte neue Version weitergeleitet wird	⌚⌚⌚⌚††	✓ Ja	✓ Ja	Datenverlust umleitete Instances und neue Instances beenden	Neue Instances	
Blau/Grün	Minimal	⌚⌚⌚⌚	✓ Ja	X Nein	URL austauschen	Neue Instances	

† Variiert je nach Stapelgröße.

†† Variiert je nach Einstellung der Option Auswertungszeit.

Bereitstellen einer neuen Anwendungsversion

Sie können Bereitstellungen über das Dashboard Ihrer Umgebung durchführen.

So stellen Sie eine neue Anwendungsversion in einer Elastic Beanstalk-Umgebung bereit:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie auf Upload and Deploy (Hochladen und Bereitstellen).
4. Verwenden Sie das Formular auf dem Bildschirm, um das Quellpaket der Anwendung hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.

Erneute Bereitstellung einer früheren Version

Sie können auch eine zuvor hochgeladene Version Ihrer Anwendung in einer der Umgebungen aus der Anwendungsversionsseite bereitstellen.

So stellen Sie eine vorhandene Anwendungsversion in einer vorhandenen Umgebung bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.

Note

Wenn Sie viele Anwendungen haben, verwenden Sie die Suchleiste, um die Anwendungsliste zu filtern.

3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Application versions (Anwendungsversionen) aus.
4. Wählen Sie die Anwendungsversion aus, die bereitgestellt werden soll.
5. Wählen Sie Actions (Aktionen) und anschließend Deploy (Bereitstellen).
6. Wählen Sie eine Umgebung und anschließend Deploy (Bereitstellen).

Weitere Möglichkeiten zur Bereitstellung Ihrer Anwendung

Wenn Sie häufig bereitstellen, sollten Sie die [Elastic Beanstalk-Befehlszeilenschnittstelle \(p. 1008\)](#) (EB CLI) zum Verwalten Ihrer Umgebungen verwenden. Die EB-CLI erstellt neben Ihrem Quellcode ein Repository. Sie kann außerdem ein Quellpaket erstellen, es nach Elastic Beanstalk hochladen und mit einem einzigen Befehl bereitstellen.

Für Bereitstellungen, die von Änderungen der Ressourcenkonfiguration oder einer neuen Version, die nicht neben der alten Version ausgeführt werden kann, abhängen, können Sie eine neue Umgebung mit der neuen Version starten und einen CNAME-Austausch für eine [Blau/Grün-Bereitstellung \(p. 481\)](#) durchführen.

Bereitstellungsrichtlinien und -einstellungen

AWS Elastic Beanstalk bietet mehrere Optionen für die Verarbeitung von [Bereitstellungen \(p. 471\)](#), darunter Bereitstellungsrichtlinien (All at once (Alle auf einmal), Rolling (Fortlaufend), Rolling with additional batch (Fortlaufend mit zusätzlichem Stapel), Immutable (Unveränderlich) und Traffic splitting (Datenverkehrsaufteilung)) und Optionen, mit denen Sie die Stapelgröße sowie das Zustandsprüfungsverhalten bei Bereitstellungen konfigurieren können. Standardmäßig stellt Ihre Umgebung alle Bereitstellungen gleichzeitig bereit. Wenn Sie die Umgebung über die EB CLI erstellt haben und es sich um eine skalierbare Umgebung handelt (d. h., Sie haben die Option --single nicht angegeben), verwendet sie fortlaufende Bereitstellungen.

Bei fortlaufenden Bereitstellungen teilt Elastic Beanstalk die Amazon EC2-Instances der Umgebung in Stapeln auf und stellt die neue Version der Anwendung jeweils in einem Stapel bereit. Auf den verbleibenden Instances in der Umgebung wird weiterhin die alte Version der Anwendung ausgeführt. Während der fortlaufenden Bereitstellung werden Anforderungen sowohl von Instances mit der alten Anwendungsversion als auch von Instances mit der neuen Anwendungsversion (in den bereits aktualisierten Stapeln) verarbeitet. Details dazu finden Sie unter [the section called “Funktionsweise fortlaufender Bereitstellungen” \(p. 478\)](#).

Um auch während der Bereitstellungen die volle Kapazität zu wahren, können Sie die Umgebung so konfigurieren, dass erst ein neuer Instance-Stapel gestartet wird, bevor andere Instances außer Betrieb genommen werden. Diese Option wird als fortlaufende Bereitstellung mit zusätzlichem Stapel bezeichnet. Nach erfolgter Bereitstellung beendet Elastic Beanstalk den zusätzlichen Instance-Stapel.

Bei unveränderlichen Bereitstellungen wird eine [unveränderliche Aktualisierung \(p. 488\)](#) ausgeführt. Dabei wird zusätzlich zu den Instances mit der alten Anwendungsversion eine ganze Gruppe neuer Instances mit der neuen Anwendungsversion in einer separaten Auto Scaling-Gruppe gestartet. Die Probleme, die von teilweise abgeschlossenen fortlaufenden Bereitstellungen verursacht werden können, treten bei unveränderlichen Bereitstellungen nicht auf. Falls die neuen Instances die Zustandsprüfungen nicht bestehen, werden sie von Elastic Beanstalk beendet – und die ursprünglichen Instances sind unverändert vorhanden.

Mit Datenverkehrsaufteilungs-Bereitstellungen können Sie Canary-Tests als Teil Ihrer Anwendungsbereitstellung durchführen. In einer Bereitstellung mit Datenverkehrsaufteilung startet Elastic Beanstalk eine vollständige Reihe neuer Instances, genau wie bei einer unveränderlichen Bereitstellung. Anschließend leitet er einen bestimmten Prozentsatz des eingehenden Clientdatenverkehrs für einen bestimmten Evaluierungszeitraum an die neue Anwendungsversion weiter. Wenn die neuen Instances fehlerfrei bleiben, leitet Elastic Beanstalk den gesamten Datenverkehr an sie weiter und beendet die alten Instances. Wenn die neuen Instances keine Zustandsprüfungen bestehen oder wenn Sie die Bereitstellung abbrechen, verschiebt Elastic Beanstalk den Datenverkehr zurück zu den alten Instances und beendet die neuen Instances. Der Service wird nie unterbrochen. Details dazu finden Sie unter [the section called "Funktionsweise von Bereitstellungen mit Datenverkehrsaufteilung" \(p. 479\)](#).

Warning

Einige Richtlinien ersetzen alle Instances während der Bereitstellung oder Aktualisierung. Dadurch gehen alle angesammelten [Amazon EC2-Steigerungsguthaben](#) verloren. Dies geschieht in den folgenden Fällen:

- Verwaltete Plattform-Updates mit aktiviertem Instance-Ersatz
- Unveränderliche Updates
- Bereitstellungen mit unveränderlichen Updates oder Datenverkehrsaufteilung aktiviert

Im Falle, dass die Anwendung nicht alle Zustandsprüfungen besteht, aber mit einem niedrigeren Status ordnungsgemäß funktioniert, können Sie die Option Healthy threshold ändern und so zulassen, dass Instances die Zustandsprüfungen auch mit einem niedrigeren Status (z. B. Warning) bestehen. Um bei fehlgeschlagenen Bereitstellungen aufgrund von nicht bestandenen Zustandsprüfungen eine Aktualisierung unabhängig vom Status zu erzwingen, aktivieren Sie die Option Ignore health check (Zustandsprüfung ignorieren).

Sofern Sie eine Stapelgröße für fortlaufende Aktualisierungen angeben, verwendet Elastic Beanstalk diesen Wert auch für den fortlaufenden Anwendungsneustart. Verwenden Sie fortlaufende Neustarts, wenn die Proxy- und Anwendungsserver, die auf den Instances der Umgebung ausgeführt werden, ohne Ausfallzeit neu gestartet werden müssen.

Konfigurieren von Anwendungsbereitstellungen

Aktivieren und konfigurieren Sie in der [Environment Management Console \(p. 426\)](#) die Stapelbereitstellung von Anwendungsversionen, indem Sie die Option Updates and Deployments (Updates und Bereitstellungen) auf der Seite Configuration (Konfiguration) der Umgebung ändern.

So konfigurieren Sie Bereitstellungen (Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Rolling updates and deployments (Fortlaufende Updates und Bereitstellungen) die Option Edit (Bearbeiten).
5. Wählen Sie unter Application Deployments (Anwendungsbereitstellungen) eine Option für Deployment policy (Bereitstellungsrichtlinie) sowie Stapeleinstellungen und Optionen für die Zustandsprüfung aus.
6. Wählen Sie Apply (Anwenden) aus.

Im Bereich Application deployments (Anwendungsbereitstellungen) der Seite Rolling updates and deployments (Fortlaufende Updates und Bereitstellungen) sind die folgenden Optionen für Anwendungsbereitstellungen verfügbar:

- Deployment policy (Bereitstellungsrichtlinie) – Wählen Sie eine der folgenden Bereitstellungsoptionen aus:
 - All at once (Alle auf einmal) – Die neue Version wird auf allen Instances gleichzeitig bereitgestellt. Während der Bereitstellung sind alle Instances der Umgebung kurzzeitig außer Betrieb.
 - Rolling (Fortlaufend) – Die neue Version wird in Stapeln bereitgestellt. Während der Bereitstellung ist der jeweilige Stapel außer Betrieb, sodass die Kapazität der Umgebung um die Anzahl der Instances in einem Stapel reduziert wird.
 - Rolling with additional batch (Fortlaufend mit zusätzlichem Stapel) – Die neue Version wird in Stapeln bereitgestellt, aber zuerst wird ein neuer Instance-Stapel gestartet, um die volle Kapazität auch während der Bereitstellung zu gewährleisten.
 - Immutable (Unveränderlich) – Die neue Version wird mithilfe einer [unveränderlichen Aktualisierung \(p. 488\)](#) auf einer neuen Instance-Gruppe bereitgestellt.
 - Datenverkehrsaufteilung – Stellen Sie die neue Version für eine neue Gruppe von Instances bereit und teilen Sie den eingehenden Clientdatenverkehr vorübergehend zwischen der vorhandenen und der neuen Anwendungsversion auf.

Für die Bereitstellungsrichtlinien Rolling (Fortlaufend) und Rolling with additional Batch (Fortlaufend mit zusätzlichem Stapel) können Sie Folgendes konfigurieren:

- Batch size (Stapelgröße) – Die Größe der Gruppe von Instances für die Bereitstellung in jedem Stapel.

Wählen Sie Percentage (Prozentsatz) aus, um einen Prozentsatz für die Gesamtanzahl der EC2-Instances in der Auto Scaling-Gruppe (bis zu 100 %) zu konfigurieren, oder wählen Sie Fixed (Fest) aus, um eine feste Anzahl von Instances zu konfigurieren (bis auf die maximale Anzahl der Instances in der Auto Scaling-Konfiguration Ihrer Umgebung).

Sie können für die Bereitstellungsrichtlinie Traffic Splitting (Datenverkehrsaufteilung) Folgendes konfigurieren:

- Traffic split (Datenverkehrsaufteilung) – Der anfängliche Prozentsatz des eingehenden Clientdatenverkehrs, den Elastic Beanstalk auf Umgebungs-Instances verlagert, auf denen die von Ihnen bereitgestellte neue Anwendungsversion ausgeführt wird.
- Auswertungszeit für Datenverkehrsaufteilung – Der Zeitraum (in Minuten), den Elastic Beanstalk nach einer anfänglich fehlerfreien Bereitstellung wartet, bevor der gesamte eingehende Clientdatenverkehr auf die von Ihnen bereitgestellte neue Anwendungsversion verschoben wird.

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

Der Abschnitt Deployment Preferences (Bereitstellungspräferenzen) enthält die Optionen für Zustandsprüfungen.

- Ignore health check (Zustandsprüfung ignorieren) – Verhindert den Rollback einer Bereitstellung, falls ein Stapel nicht innerhalb des unter Command timeout (Befehls-Timeout) angegebenen Werts als stabil eingestuft wird.
- Healthy threshold (Schwellenwert für den fehlerfreien Zustand) – Damit kann der Grenzwert, bei dem eine Instance als stabil eingestuft wird, für fortlaufende Bereitstellungen, fortlaufende Aktualisierungen und unveränderliche Aktualisierungen herabgesetzt werden.
- Command timeout (Befehls-Timeout) – Gibt an, wie viele Sekunden darauf gewartet wird, dass eine Instance als stabil eingestuft wird, bevor die Bereitstellung abgebrochen oder – bei aktivierter Ignore health check (Zustandsprüfung ignorieren)-Option – mit dem nächsten Stapel begonnen wird.

The screenshot shows the 'Deployment preferences' section of the AWS Elastic Beanstalk configuration interface. It includes fields for 'Ignore health check' (set to 'False'), 'Healthy threshold' (set to 'Ok'), and 'Command timeout' (set to '600'). Each field has a dropdown arrow icon to its right.

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.

Funktionsweise fortlaufender Bereitstellungen

Im Rahmen der Stapelverarbeitung trennt Elastic Beanstalk alle Instances im Stapel vom Load Balancer, stellt die neue Anwendungsversion bereit und ordnet die Instances wieder zu. Sofern [Connection Draining \(Verbindlungsausgleich\) \(p. 562\)](#) aktiviert ist, gleicht Elastic Beanstalk bestehende Verbindungen der Amazon EC2-Instances in jedem Stapel aus, bevor mit der Bereitstellung begonnen wird.

Nachdem die Instances im Stapel wieder dem Load Balancer zugeordnet wurden, wartet Elastic Load Balancing, bis die Instances die mindestens erforderlichen Elastic Load Balancing-Zustandsprüfungen (der Healthy check count threshold-Wert) bestanden haben, anschließend wird dann der Datenverkehr an sie weitergeleitet. Falls keine [URL für die Zustandsprüfung \(p. 563\)](#) konfiguriert ist, kann das sehr schnell gehen, da eine Instance die Zustandsprüfung bereits besteht, sobald sie eine TCP-Verbindung annimmt. Ist eine URL für die Zustandsprüfung definiert, leitet der Load Balancer den Datenverkehr erst an die aktualisierten Instances weiter, wenn diese als Antwort auf eine 200 OK-Anforderung den Statuscode `HTTP GET` an die URL für die Zustandsprüfung senden.

Elastic Beanstalk wartet, bis alle Instances in einem Stapel stabil sind, bevor mit dem nächsten Stapel begonnen wird. Beim [Basis-Health-Reporting \(p. 821\)](#) hängt der Status einer Instance vom Status der Elastic Load Balancing-Zustandsprüfung ab. Wenn alle Instances im Stapel genügend Zustandsprüfungen bestanden haben, um von Elastic Load Balancing als stabil eingestuft zu werden, ist der Stapel abgeschlossen. Sind [erweiterte Zustandsberichte \(p. 824\)](#) aktiviert, berücksichtigt Elastic Beanstalk noch einige weitere Faktoren, darunter das Ergebnis eingehender Anforderungen. Mit erweiterten Zustandsberichten müssen alle Instances 12 aufeinanderfolgende Zustandsprüfungen mit dem [Status "OK" \(p. 841\)](#) innerhalb von zwei Minuten für Webserverumgebungen und 18 Zustandsprüfungen innerhalb von drei Minuten für Worker-Umgebungen bestehen.

Die Bereitstellung schlägt fehl, wenn ein Instance-Stapel nicht innerhalb des [Timeout-Werts für Befehle \(p. 475\)](#) als stabil eingestuft wird. Nach einer fehlgeschlagenen Bereitstellung [überprüfen Sie den Zustand der Instances in Ihrer Umgebung \(p. 835\)](#) nach Informationen über die Ursache des Fehlers. Führen Sie dann einen Rollback mit einer anderen Bereitstellung mit einer fixen oder funktionierenden, guten Version Ihrer Anwendung durch.

Schlägt eine Bereitstellung nach dem erfolgreichen Abschluss von einem oder mehreren Stapeln fehl, wird auf den aktualisierten Stapeln die neue Anwendungsversion und auf ausstehenden Stapeln weiterhin die alte Version ausgeführt. Die auf den Umgebungs-Instances ausgeführte Version wird auf der [Zustandsprüfungsseite \(p. 836\)](#) in der Konsole angezeigt. Auf dieser Seite wird die Bereitstellungs-ID der zuletzt für die Instances der Umgebung ausgeführte Bereitstellung angegeben. Wenn Sie Instances aus der fehlgeschlagenen Bereitstellung beenden, ersetzt Elastic Beanstalk diese durch Instances, auf denen die Anwendungsversion der letzten erfolgreichen Bereitstellung ausgeführt wird.

Funktionsweise von Bereitstellungen mit Datenverkehrsaufteilung

Mit Datenverkehrsaufteilungs-Bereitstellungen können Sie Canary-Tests durchführen. Sie leiten einen Teil des eingehenden Clientdatenverkehrs an die neue Anwendungsversion weiter, um die Integrität der Anwendung zu überprüfen, bevor Sie die neue Version einsetzen und den gesamten Datenverkehr dahin weiterleiten.

Während einer Bereitstellung mit Datenverkehrsaufteilung erstellt Elastic Beanstalk einen neuen Satz von Instance in einer separaten temporären Auto Scaling-Gruppe. Elastic Beanstalk weist dann den Load Balancer an, einen bestimmten Prozentsatz des eingehenden Datenverkehrs Ihrer Umgebung an die neuen Instances weiterzuleiten. Anschließend verfolgt Elastic Beanstalk für einen festgelegten Zeitraum die Integrität dieser neuen Instances. Ist alles in Ordnung, verschiebt Elastic Beanstalk den verbleibenden Datenverkehr auf die neuen Instances und fügt sie der ursprünglichen Auto Scaling-Gruppe der Umgebung hinzu, wodurch die alten Instances ersetzt werden. Anschließend bereinigt Elastic Beanstalk die alten Instances und entfernt die temporäre Auto Scaling-Gruppe.

Note

Die Kapazität der Umgebung ändert sich während einer Bereitstellung mit Datenverkehrsaufteilung nicht. Elastic Beanstalk startet die gleiche Anzahl von Instances in der temporären Auto Scaling-Gruppe wie in der ursprünglichen Auto Scaling-Gruppe zum Zeitpunkt des Bereitstellungsbeginns. Für die Dauer der Bereitstellung wird dann eine konstante Anzahl von Instances in beiden Auto Scaling-Gruppen verwaltet. Berücksichtigen Sie diese Tatsache bei der Konfiguration der Evaluierungszeit für die Datenverkehrsaufteilung der Umgebung.

Das Rollback der Bereitstellung auf die vorherige Anwendungsversion erfolgt schnell und wirkt sich nicht auf den Dienst für den Clientdatenverkehr aus. Wenn die neuen Instances keine Zustandsprüfungen bestehen oder wenn Sie die Bereitstellung abbrechen, verschiebt Elastic Beanstalk den Datenverkehr zurück zu den alten Instances und beendet die neuen Instances. Sie können jede Bereitstellung abbrechen, indem Sie die Übersichtsseite der Umgebung in der Elastic Beanstalk-Konsole verwenden und unter Environment actions (Umgebungsaktionen) die Option Abort current operation (Aktuellen Vorgang abbrechen) auswählen. Sie können auch die [AbortEnvironmentUpdate](#)-API oder den entsprechenden AWS CLI-Befehl aufrufen.

Für die Bereitstellung zur Datenverkehrsaufteilung ist ein Application Load Balancer erforderlich. Elastic Beanstalk verwendet diesen Load Balancer-Typ standardmäßig, wenn Sie Ihre Umgebung mit der Elastic Beanstalk-Konsole oder der EB CLI erstellen.

Namespaces für die Bereitstellungsoption

Sie können die [Konfigurationsoptionen \(p. 646\)](#) im Namespace [aws:elasticbeanstalk:command \(p. 685\)](#) zum Konfigurieren Ihrer Bereitstellungen verwenden. Wenn Sie die Richtlinie für die Datenverkehrsaufteilung auswählen, stehen zusätzliche Optionen für diese Richtlinie im Namespace [aws:elasticbeanstalk:trafficsplitting \(p. 698\)](#) zur Verfügung.

Mit der Option `DeploymentPolicy` legen Sie den Bereitstellungstyp fest. Folgende Werte werden unterstützt:

- `AllAtOnce` – Deaktiviert fortlaufende Bereitstellungen; die Bereitstellung erfolgt stets auf allen Instances gleichzeitig.

- **Rolling** – Aktiviert standardmäßig fortlaufende Bereitstellungen.
- **RollingWithAdditionalBatch** – Startet vor der Bereitstellung einen zusätzlichen Instance-Stapel, um die volle Kapazität zu gewährleisten.
- **Immutable** – Führt bei jeder Bereitstellung eine [unveränderliche Aktualisierung \(p. 488\)](#) aus.
- **TrafficSplitting** – Führt Bereitstellungen mit Datenverkehrsaufteilung aus, um Ihre Anwendungsbereitstellungen einem Canary-Test zu unterziehen.

Wenn Sie fortlaufende Bereitstellungen aktivieren, legen Sie die Optionen `BatchSize` und `BatchSizeType` fest, um die Größe der einzelnen Stapel zu definieren. Um beispielsweise 25 % aller Instances in den einzelnen Stapeln bereitzustellen, geben Sie folgende Optionen und Werte an.

Example `.ebextensions/rolling-updates.config`

```
option_settings:  
  aws:elasticbeanstalk:command:  
    DeploymentPolicy: Rolling  
    BatchSizeType: Percentage  
    BatchSize: 25
```

Um unabhängig von der Anzahl ausgeführter Instances jeweils fünf Instances in den einzelnen Stapeln bereitzustellen und einen zusätzlichen Stapel mit fünf Instances, auf denen bereits die neue Version ausgeführt wird, vor der Außerbetriebnahme einer anderen Instance bereitzustellen, geben Sie folgende Optionen und Werte an.

Example `.ebextensions/rolling-additionalbatch.config`

```
option_settings:  
  aws:elasticbeanstalk:command:  
    DeploymentPolicy: RollingWithAdditionalBatch  
    BatchSizeType: Fixed  
    BatchSize: 5
```

Um für jede Bereitstellung mit dem Zustandsprüfungs-Grenzwert `Warning` (Warnung) eine unveränderliche Aktualisierung auszuführen und die Bereitstellung fortzusetzen, auch wenn die Instances im Stapel die Zustandsprüfungen nicht innerhalb von 15 Minuten bestanden haben, geben Sie folgende Optionen und Werte an.

Example `.ebextensions/immutable-ignorehealth.config`

```
option_settings:  
  aws:elasticbeanstalk:command:  
    DeploymentPolicy: Immutable  
    HealthCheckSuccessThreshold: Warning  
    IgnoreHealthCheck: true  
    Timeout: "900"
```

Geben Sie die folgenden Optionen und Werte an, um Bereitstellungen mit Datenverkehrsaufteilung durchzuführen, 15 Prozent des Clientdatenverkehrs an die neue Anwendungsversion weiterzuleiten und die Integrität für 10 Minuten zu bewerten.

Example `.ebextensions/traffic-splitting.config`

```
option_settings:  
  aws:elasticbeanstalk:command:
```

```
DeploymentPolicy: TrafficSplitting
aws:elasticbeanstalk:trafficsplitting:
  NewVersionPercent: "15"
  EvaluationTime: "10"
```

Die EB CLI und Elastic Beanstalk-Konsole wenden die empfohlenen Werte für die vorhergehenden Optionen an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter [Empfohlene Werte \(p. 648\)](#).

Blau/Grün-Bereitstellungen mit AWS Elastic Beanstalk

Da AWS Elastic Beanstalk ein Vor-Ort-Update durchführt, wenn Sie Ihre Anwendungsversionen aktualisieren, kann Ihre Anwendung für einen kurzen Zeitraum nicht mehr für Benutzer verfügbar sein. Sie können diese Ausfallzeit verhindern, indem Sie eine Blau/Grün-Bereitstellung durchführen. Bei dieser wird die neue Version auf einer separaten Umgebung bereitgestellt und die CNAMEs der beiden Umgebungen werden ausgetauscht, um den Datenverkehr sofort an die neue Version weiterzuleiten.

Eine Blau/Grün-Bereitstellung ist auch erforderlich, wenn Sie eine Umgebung auf eine inkompatible Plattformversion aktualisieren möchten. Weitere Informationen finden Sie unter [the section called "Plattformaktualisierungen" \(p. 491\)](#).

Für Blau/Grün-Bereitstellungen muss Ihre Umgebung unabhängig von Ihrer Produktionsdatenbank ausgeführt werden, wenn Ihre Anwendung eine verwendet. Wenn Ihre Umgebung eine Amazon RDS DB-Instance angefügt hat, werden die Daten nicht in die zweite Umgebung übertragen und gehen verloren, wenn Sie die ursprüngliche Umgebung beenden.

Weitere Informationen zur Konfiguration Ihrer Anwendung für eine Verbindung mit einer externen (nicht von Elastic Beanstalk verwalteten) Amazon RDS-Instance finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon RDS \(p. 973\)](#).

So führen Sie eine Blau/Grün-Bereitstellung durch

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. [Klonen Sie Ihre aktuelle Umgebung \(p. 456\)](#) oder starten Sie eine neue Umgebung mit der gewünschten Plattformversion.
3. [Stellen Sie die neue Anwendungsversion \(p. 473\)](#) in der neuen Umgebung bereit.
4. Testen Sie die neue Version auf der neuen Umgebung.
5. Wählen Sie auf der Umgebungsübersichtsseite Environment actions (Umgebungsaktionen) und dann Swap environment URLs (Umgebungs-URLs austauschen).
6. Wählen Sie unter Environment name (Umgebungsname) die aktuelle Umgebung aus.

Swap environment URLs

When you swap an environment's URL with another environment's URL, you can deploy versions with no downtime.

⚠️ Swapping the environment URL will modify the Route 53 DNS configuration, which may take a few minutes to propagate. 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-2mvwbhpfcs)

Environment URL:

prod-env.bx7dx222kw.us-east-2.elasticbeanstalk.com

7. Klicken Sie auf Swap (Austauschen).

Elastic Beanstalk tauscht die CNAME-Datensätze der alten und neuen Umgebungen aus und leitet so den Datenverkehr von der alten auf die neue Version und umgekehrt um.

Nachdem Elastic Beanstalk den Austausch abgeschlossen hat, stellen Sie sicher, dass die neue Umgebung reagiert, wenn Sie versuchen, eine Verbindung zur alten Umgebungs-URL herzustellen.

Beenden Sie Ihre alte Umgebung jedoch nicht, bis die DNS-Änderungen übertragen werden und Ihre alten DNS-Datensätze ablaufen. DNS-Server löschen nicht unbedingt alte Datensätze aus dem Cache basierend auf der TTL (Time to Live, Gültigkeitsdauer), die Sie für Ihre DNS-Datensätze festlegen.

Konfigurationsänderungen

Wenn Sie die Einstellungen der Konfigurationsoptionen im Abschnitt Configuration (Konfiguration) der [Environment Management Console \(p. 426\)](#) ändern, werden die Änderungen von AWS Elastic Beanstalk an alle betroffenen Ressourcen übermittelt. Zu diesen Ressourcen zählen der Load Balancer (der den Datenverkehr auf die Amazon EC2-Instances verteilt, auf denen die Anwendung ausgeführt wird), die Auto Scaling-Gruppe (von der diese Instances verwaltet werden) sowie die EC2-Instances selbst.

Viele Konfigurationsänderungen können in eine ausgeführte Umgebung übernommen werden, ohne vorhandene Instances zu ersetzen. Beispielsweise löst das Festlegen einer [URL für die Zustandsprüfung \(p. 563\)](#) eine Umgebungsaktualisierung aus, um die Load Balancer-Einstellungen zu ändern. Dies verursacht jedoch keine Ausfallzeit, weil die Instances, auf denen die Anwendung ausgeführt wird, während des Aktualisierungsvorgangs weiterhin Anforderungen verarbeiten.

Falls mit Konfigurationsänderungen die [Startkonfiguration \(p. 668\)](#) oder die [VPC-Einstellungen \(p. 682\)](#) geändert werden, müssen alle Instances in der Umgebung beendet und ersetzt werden. Wenn Sie beispielsweise den Instance-Typ oder die SSH-Schlüsseleinstellung für die Umgebung ändern, müssen die EC2-Instances beendet und ersetzt werden. Elastic Beanstalk bietet verschiedene Richtlinien, die festlegen, wie dieser Ersatz durchgeführt wird.

- Rolling updates (Fortlaufende Aktualisierungen) – Elastic Beanstalk wendet Ihre Konfigurationsänderungen in Stapeln an, wobei eine Mindestanzahl von Instances jederzeit ausgeführt wird und für Datenverkehr verfügbar ist. Dies verhindert Ausfallzeiten während des Aktualisierungsvorgangs. Details dazu finden Sie unter [Fortlaufende Updates \(p. 484\)](#).
- Immutable updates (Unveränderliche Aktualisierungen) – Elastic Beanstalk startet eine temporäre Auto Scaling-Gruppe außerhalb Ihrer Umgebung mit einem separaten Satz von Instances, die in der neuen Konfiguration ausgeführt werden. Anschließend platziert Elastic Beanstalk diese Instances hinter dem Load Balancer Ihrer Umgebung. Alte und neue Instances dienen beide dem Datenverkehr, bis die neuen Instances die Zustandsprüfungen bestehen. Elastic Beanstalk verschiebt zu diesem Zeitpunkt die neuen Instances in die Auto Scaling-Gruppe Ihrer Umgebung und beendet die temporäre Gruppe und alte Instances. Details dazu finden Sie unter [Unveränderliche Updates \(p. 488\)](#).
- Disabled (Deaktiviert) – Elastic Beanstalk macht keinen Versuch, Ausfallzeiten zu vermeiden. Es beendet die vorhandenen Instances Ihrer Umgebung und ersetzt sie durch neue Instances, die mit der neuen Konfiguration ausgeführt werden.

Warning

Einige Richtlinien ersetzen alle Instances während der Bereitstellung oder Aktualisierung. Dadurch gehen alle angesammelten [Amazon EC2-Steigerungsguthaben](#) verloren. Dies geschieht in den folgenden Fällen:

- Verwaltete Plattform-Updates mit aktiviertem Instance-Ersatz
- Unveränderliche Updates
- Bereitstellungen mit unveränderlichen Updates oder Datenverkehrsaufteilung aktiviert

Unterstützte Aktualisierungs-Typen

Einstellung für fortlaufende Aktualisierung	Umgebungen mit Load Balancing	Umgebungen mit einer Instance	Ältere Windows Server-Umgebungen†
Disabled	✓ Ja	✓ Ja	✓ Ja
Rolling Based on Health	✓ Ja	✗ Nein	✓ Ja
Rolling Based on Time	✓ Ja	✗ Nein	✓ Ja
Immutable (Unveränderlich)	✓ Ja	✓ Ja	✗ Nein

† Für die Zwecke dieser Tabelle ist eine Legacy-Windows Server-Umgebung eine Umgebung, die auf einer [Windows Server-Plattformkonfiguration](#) mit einer früheren IIS-Version als IIS 8.5 basiert.

Themen

- [Fortlaufende Updates der Elastic Beanstalk-Umgebungskonfiguration \(p. 484\)](#)
- [Unveränderliche Umgebungsaktualisierungen \(p. 488\)](#)

Fortlaufende Updates der Elastic Beanstalk-Umgebungskonfiguration

Wenn eine [Konfigurationsänderung vorsieht, dass Instances ersetzt werden \(p. 483\)](#), kann Elastic Beanstalk das Update in Stapeln durchführen, um Ausfallzeiten zu vermeiden, während die Änderung verbreitet wird. Während eines fortlaufenden Updates wird die Kapazität nur um die Größe eines einzigen Stapels reduziert, den Sie konfigurieren können. Elastic Beanstalk schaltet einen Stapel an Instances offline, beendet diese und startet dann einen Stapel mit der neuen Konfiguration. Nachdem der neue Stapel beginnt, Anfragen zu bedienen, fährt Elastic Beanstalk mit dem nächsten Stapel fort.

Fortlaufende Konfigurations-Update-Stapel können in regelmäßigen Abständen (zeitbasiert) verarbeitet werden – mit einer Verzögerung zwischen jedem Stapel oder basierend auf dem Zustand. Für zeitbasierte fortlaufende Updates können Sie die Dauer konfigurieren, die Elastic Beanstalk wartet, nachdem der Start eines Instance-Stapels abgeschlossen ist, bevor mit dem nächsten Stapel fortgefahren wird. Durch diese Pausenzeit kann Ihre Anwendung einen Bootsrap durchführen und mit dem Verarbeiten von Anforderungen beginnen.

Bei zustandsbasierten fortlaufenden Updates wartet Elastic Beanstalk, bis Instances in einem Stapel die Zustandsprüfungen bestanden haben, bevor mit dem nächsten Stapel fortgefahren wird. Der Zustand einer Instance wird durch das System für Zustandsberichte bestimmt. Dieses kann grundlegend oder erweitert sein. Beim [grundlegenden Zustandssystem \(p. 821\)](#) wird ein Stapel als stabil betrachtet, wenn alle Instances darin die Elastic Load Balancing (ELB)-Zustandsprüfungen bestehen.

Bei [erweiterten Zustandsberichten \(p. 824\)](#) müssen alle Instances in einem Stapel mehrere aufeinanderfolgende Zustandsprüfungen bestehen, bevor Elastic Beanstalk mit dem nächsten Stapel fortfährt. Zusätzlich zu den ELB-Zustandsprüfungen, bei denen nur Ihre Instances geprüft werden, werden bei der erweiterten Zustandsprüfung Anwendungsprotokolle und der Status der anderen Ressourcen Ihrer Umgebung überprüft. In einer Webserverumgebung mit erweiterten Zustandsberichten müssen alle Instances im Laufe von zwei Minuten zwölf Zustandsprüfungen bestehen (18 Prüfungen über drei Minuten für Worker-Umgebungen). Wenn eine Instance die Zustandsprüfung nicht besteht, wird die Anzahl zurückgesetzt.

Wenn ein Stapel während des Timeouts des fortlaufenden Updates keinen stabilen Zustand erreicht (Standard: 30 Minuten), wird das Update abgebrochen. Das

Timeout beim fortlaufenden Update ist eine [Konfigurationsoption \(p. 646\)](#), die im `aws:autoscaling:updatepolicy:rollingupdate (p. 487)`-Namespace verfügbar ist. Wenn Ihre Anwendung Zustandsprüfungen nicht mit dem Ok-Status besteht, sie auf einer anderen Stufe jedoch stabil ist, können Sie die `HealthCheckSuccessThreshold`-Option im `aws:elasticbeanstalk:healthreporting:system (p. 692)`-Namespace so festlegen, dass die Stufe geändert wird, auf der Elastic Beanstalk eine Instance als stabil betrachtet.

Wenn das fortlaufende Update fehlschlägt, startet Elastic Beanstalk ein weiteres, um zur vorherigen Konfiguration zurückzukehren. Ein fortlaufendes Update kann aufgrund fehlerhafter Zustandsprüfungen fehlschlagen oder wenn Sie durch das Starten neuer Instances die Kontingente auf Ihrem Konto überschreiten. Wenn Sie beispielsweise ein Kontingent für die Anzahl der Amazon EC2-Instances erreichen, kann das fortlaufende Update fehlschlagen, wenn versucht wird, einen Stapel an neuen Instances bereitzustellen. In diesem Fall schlägt das Rollback ebenfalls fehl.

Bei einem fehlgeschlagenen Rollback wird der Aktualisierungsvorgang beendet und Ihre Umgebung verbleibt in einem instabilen Zustand. Nicht verarbeitete Stapel führen nach wie vor Instances mit der alten Konfiguration aus, während erfolgreich abgeschlossene Stapel über die neue Konfiguration verfügen. Um eine Umgebung nach einem fehlgeschlagenen Rollback zu reparieren, beheben Sie zuerst das zugrundeliegende Problem, durch das das Update fehlschlägt, und initiieren Sie anschließend ein weiteres Umgebungs-Update.

Eine alternative Methode ist, die neue Version Ihrer Anwendung für eine andere Umgebung bereitzustellen und dann einen CNAME-Austausch durchzuführen, um den Datenverkehr ohne Ausfallzeit weiterzuleiten. Weitere Informationen finden Sie unter [Blau/Grün-Bereitstellungen mit AWS Elastic Beanstalk \(p. 481\)](#).

Fortlaufende Updates im Vergleich zu fortlaufenden Bereitstellungen

Fortlaufende Updates erfolgen, wenn Sie die Einstellungen ändern, die vorsehen, dass neue Amazon EC2-Instances für Ihre Umgebung bereitgestellt werden. Dazu gehören Änderungen an der Auto Scaling-Gruppenkonfiguration, wie Instance-Typ- und Schlüsselpaar-Einstellungen, sowie Änderungen an den VPC-Einstellungen. Bei einem fortlaufenden Update wird jeder Instance-Stapel beendet, bevor ein neuer Stapel bereitgestellt wird, der den alten ersetzen soll.

[Fortlaufende Bereitstellungen \(p. 474\)](#) treten auf, wenn Sie Ihre Anwendung bereitstellen. Sie können in der Regel durchgeführt werden, ohne dass Instances in Ihrer Umgebung ersetzt werden. Elastic Beanstalk schaltet jeden Stapel offline, stellt die neue Anwendungsversion bereit und nimmt sie dann wieder in Betrieb.

Eine Ausnahme liegt dann vor, wenn Sie Einstellungen ändern, bei denen der Instance-Austausch gleichzeitig mit der Bereitstellung einer neuen Anwendungsversion erfolgen muss. Wenn Sie beispielsweise die [Schlüsselnamen \(p. 668\)](#)-Einstellungen in einer [Konfigurationsdatei \(p. 722\)](#) in Ihrem Quell-Bundle ändern und es in Ihrer Umgebung bereitstellen, lösen Sie ein fortlaufendes Update aus. Anstatt Ihre neue Anwendungsversion für jeden Stapel an vorhandenen Instances bereitzustellen, wird ein neuer Instance-Stapel mit der neuen Konfiguration bereitgestellt. In diesem Fall erfolgt keine separate Bereitstellung, da die neuen Instances mit der neuen Anwendungsversion aufgerufen werden.

Jedes Mal, wenn neue Instances als Teil eines Umgebungs-Updates bereitgestellt werden, gibt es eine Bereitstellungsphase, in der der Quellcode Ihrer Anwendung für die neuen Instances bereitgestellt wird. Darüber hinaus werden alle Konfigurationseinstellungen, die das Betriebssystem oder die Software auf den Instances ändern, angewendet. [Einstellungen der Bereitstellungs-Zustandsprüfung \(p. 475\)](#) (Ignore health check (Zustandsprüfung ignorieren)), Healthy threshold (Zulässige Schwelle) Command timeout (Befehls-Timeout)) gelten auch für zustandsbasierte fortlaufende Updates und unveränderliche Updates während der Bereitstellungsphase.

Konfigurieren von fortlaufendes Updates

Sie können fortlaufende Updates in der Elastic Beanstalk-Konsole aktivieren und konfigurieren.

So aktivieren Sie fortlaufende Updates

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Rolling updates and deployments (Fortlaufende Updates und Bereitstellungen) die Option Edit (Bearbeiten).
5. Wählen Sie im Bereich Configuration updates (Konfigurationsaktualisierungen) für Rolling update type (Typ der fortlaufenden Aktualisierung) und wählen Sie eine der Optionen unter Rolling (Fortlaufend) aus.

The screenshot shows the 'Configuration updates' section of the AWS Elastic Beanstalk console. It includes fields for 'Rolling update type' (set to 'Rolling based on Health'), 'Batch size' (set to '1'), 'Minimum capacity' (set to '1'), and 'Pause time' (set to 'hh:mm:ss'). Each field has a descriptive tooltip below it.

Configuration updates
Changes to virtual machine settings and VPC configuration trigger rolling updates to replace the instances in your environment.
[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. Wählen Sie die Einstellungen Batch size (Stapelgröße), Minimum capacity (Mindestkapazität) und Pause time (Pausendauer).
7. Wählen Sie Apply (Anwenden) aus.

Im Bereich Configuration updates (Konfigurationsaktualisierungen) der Seite Updates and Deployments (Updates und Bereitstellungen) sind die folgenden Optionen für fortlaufende Updates verfügbar:

- Rolling update type (Art des fortlaufenden Updates) – Elastic Beanstalk wartet, nachdem das Aktualisieren eines Instance-Stapels abgeschlossen wurde, bevor mit dem nächsten Stapel fortgefahren

wird, damit diese Instances das Bootstrapping abschließen und mit dem Bereitstellen von Datenverkehr beginnen können. Wählen Sie aus den folgenden Optionen aus:

- Rolling based on Health (Fortlaufend basierend auf dem Zustand) – Warten Sie, bis Instances im aktuellen Stapel stabil sind, bevor Sie Instances in Betrieb nehmen und den nächsten Stapel starten.
- Rolling based on Time (Fortlaufend basierend auf Zeit) – Geben Sie eine Dauer ein, die zwischen dem Starten neuer Instances und deren Inbetriebnahme gewartet wird, bevor der nächste Stapel gestartet wird.
- Immutable (Unveränderlich) – Wenden Sie die Konfigurationsänderungen auf eine neue Gruppe von Instances an, indem Sie ein [unveränderliches Update \(p. 488\)](#) durchführen.
- Batch size (Stapelgröße) – Die Anzahl der Instances, die in jedem Stapel ersetzt werden sollen, zwischen **1** und **10000**. Standardmäßig ist dieser Wert ein Drittel der Mindestgröße der Auto Scaling-Gruppe, auf eine Ganzzahl aufgerundet.
- Minimum capacity (Mindestkapazität) – Die minimale Anzahl der Instances, die weiterhin ausgeführt werden, während andere Instances aktualisiert werden, zwischen **0** und **9999**). Der Standardwert ist entweder die Mindestgröße der Auto Scaling-Gruppe oder eine Instance weniger als die maximale Größe der Auto Scaling-Gruppe – je nachdem, welche Zahl die niedrigere ist.
- Pause time (Pausendauer) (nur zeitbasiert) – Die Zeit, die gewartet wird, nachdem ein Stapel aktualisiert wurde, bevor mit dem nächsten Stapel fortgefahren wird, sodass Ihre Anwendung mit dem Empfang von Datenverkehr beginnen kann. Zwischen 0 Sekunden und einer Stunde.

Der aws:autoscaling:updatepolicy:rollingupdate-Namespace

Sie können auch die [Konfigurationsoptionen \(p. 646\)](#) im `aws:autoscaling:updatepolicy:rollingupdate (p. 677)`-Namespace zum Konfigurieren fortlaufender Bereitstellungen verwenden.

Verwenden Sie die `RollingUpdateEnabled`-Option, um fortlaufende Updates zu aktualisieren, und `RollingUpdateType`, um den Update-Typ auszuwählen. Folgende Werte werden für `RollingUpdateType` unterstützt:

- `Health` – Warten Sie, bis Instances im aktuellen Stapel stabil sind, bevor Sie Instances in Betrieb nehmen und den nächsten Stapel starten.
- `Time` – Geben Sie eine Dauer ein, die zwischen dem Starten neuer Instances und deren Inbetriebnahme gewartet wird, bevor der nächste Stapel gestartet wird.
- `Immutable` – Wenden Sie die Konfigurationsänderungen auf eine neue Gruppe von Instances an, indem Sie ein [unveränderliches Update \(p. 488\)](#) durchführen.

Wenn Sie fortlaufende Updates aktivieren, legen Sie die Optionen `MaxBatchSize` und `MinInstancesInService` fest, um die Größe der einzelnen Stapel zu definieren. Für zeitbasierte und zustandsbasierte fortlaufende Updates können Sie auch `PauseTime` bzw. `Timeout` konfigurieren.

Um beispielsweise bis zu fünf Instances gleichzeitig zu starten, während mindestens zwei Instances in Betrieb bleiben, und um fünf Minuten und 30 Sekunden zwischen Stapeln zu warten, geben Sie die folgenden Optionen und Werte an.

Example .ebextensions/timebased.config

```
option_settings:  
  aws:autoscaling:updatepolicy:rollingupdate:  
    RollingUpdateEnabled: true  
    MaxBatchSize: 5  
    MinInstancesInService: 2  
    RollingUpdateType: Time
```

PauseTime: PT5M30S

Zum Aktivieren zustandsbasierter fortlaufender Updates mit einem 45-Minuten-Timeout für jeden Stapel geben Sie die folgenden Optionen und Werte an.

Example .ebextensions/healthbased.config

```
option_settings:  
  aws:autoscaling:updatepolicy:rollingupdate:  
    RollingUpdateEnabled: true  
    MaxBatchSize: 5  
    MinInstancesInService: 2  
    RollingUpdateType: Health  
    Timeout: PT45M
```

Timeout- und PauseTime-Werte müssen in [ISO8601-Dauer](#) angegeben werden: PT#H#M#S, wobei jedes #Zeichen für die Anzahl der Stunden, Minuten beziehungsweise Sekunden steht.

Die EB CLI und Elastic Beanstalk-Konsole wenden die empfohlenen Werte für die vorhergehenden Optionen an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter [Empfohlene Werte \(p. 648\)](#).

Unveränderliche Umgebungsaktualisierungen

Unveränderliche Umgebungsaktualisierungen stellen eine Alternative zu [fortlaufenden Updates \(p. 484\)](#) dar. Unveränderliche Umgebungsaktualisierungen sorgen dafür, dass Konfigurationsänderungen, bei denen ein Ersetzen von Instances erforderlich ist, effizient und sicher angewendet werden. Wenn ein unveränderliches Umgebungs-Update fehlschlägt, erfordert der Rollbackvorgang nur das Beenden einer Auto Scaling-Gruppe. Ein fehlgeschlagenes fortlaufendes Update erfordert hingegen die Durchführung eines zusätzlichen fortlaufenden Updates, um die Änderungen rückgängig zu machen.

Um ein unveränderliches Umgebungs-Update durchzuführen, erstellt Elastic Beanstalk eine zweite, temporäre Auto Scaling-Gruppe hinter dem Load Balancer Ihrer Umgebung für die neuen Instances. Zunächst startet Elastic Beanstalk eine einzelne Instance mit der neuen Konfiguration in der neuen Gruppe. Diese Instance stellt Datenverkehr zusammen mit allen Instances in der ursprünglichen Auto Scaling-Gruppe bereit, auf denen die vorherige Konfiguration ausgeführt wird.

Wenn die erste Instance die Zustandsprüfungen besteht, startet Elastic Beanstalk zusätzliche Instances mit der neuen Konfiguration, die der Anzahl der Instances, die in der ursprünglichen Auto Scaling-Gruppe ausgeführt werden, entsprechen. Wenn alle neuen Instances Zustandsprüfungen bestehen, überträgt Elastic Beanstalk sie an die ursprüngliche Auto Scaling-Gruppe und beendet die temporäre Auto Scaling-Gruppe und alte Instances.

Note

Während eines unveränderlichen Umgebungs-Update verdoppelt sich die Kapazität Ihrer Umgebung für eine kurze Zeit, wenn die Instances in der neuen Auto Scaling-Gruppe Anforderungen verarbeiten und bevor die Instances der ursprünglichen Auto Scaling-Gruppe beendet werden. Wenn Ihre Umgebung viele Instances hat oder Sie ein niedriges [On-Demand-Instance-Kontingent](#) haben, stellen Sie sicher, dass Sie genügend Kapazität haben, um ein unveränderliches Umgebungs-Update durchzuführen. Wenn Sie in der Nähe der Kontingentgrenze liegen, sollten Sie stattdessen fortlaufende Updates verwenden.

Unveränderliche Updates erfordern [erweiterte Zustandsberichte \(p. 824\)](#) zum Auswerten des Zustands Ihrer Umgebung während des Updates. Erweiterte Zustandsberichte kombinieren Standard-Load Balancer-

Zustandsprüfungen mit Instance-Überwachung, um sicherzustellen, dass die Instances, auf denen die neue Konfiguration ausgeführt wird, [Anforderungen erfolgreich verarbeitet \(p. 827\)](#).

Sie können auch unveränderliche Updates für die Bereitstellung neuer Versionen Ihrer Anwendung verwenden, als Alternative zu fortlaufenden Bereitstellungen. Wenn Sie [Elastic Beanstalk so konfigurieren, dass unveränderliche Updates für die Anwendungsbereitstellung verwendet werden \(p. 474\)](#), werden alle Instances in Ihrer Umgebung jedes Mal ersetzt, wenn Sie eine neue Version Ihrer Anwendung bereitstellen. Wenn eine unveränderliche Anwendungsbereitstellung fehlschlägt, kehrt Elastic Beanstalk die Änderungen sofort zurück, indem die neue Auto Scaling-Gruppe beendet wird. Dies kann eine teilweise Flottenbereitstellungen verhindern, die auftreten kann, wenn eine fortlaufende Bereitstellung fehlschlägt, nachdem einige Stapel bereits abgeschlossen wurden.

Warning

Einige Richtlinien ersetzen alle Instances während der Bereitstellung oder Aktualisierung. Dadurch gehen alle angesammelten [Amazon EC2-Steigerungsguthaben](#) verloren. Dies geschieht in den folgenden Fällen:

- Verwaltete Plattform-Updates mit aktiviertem Instance-Ersatz
- Unveränderliche Updates
- Bereitstellungen mit unveränderlichen Updates oder Datenverkehrsaufteilung aktiviert

Wenn ein unveränderliches Update fehlschlägt, laden die neuen Instances [Bundle-Protokolle \(p. 869\)](#) auf Amazon S3 hoch, bevor Elastic Beanstalk sie beendet. Elastic Beanstalk hinterlässt eine Stunde lang Protokolle eines fehlerhaften unveränderlichen Updates in Amazon S3 bevor sie gelöscht werden (anstatt der standardmäßigen 15 Minuten für Bundle- und Tail-Protokolle).

Note

Wenn Sie unveränderliche Updates für die Bereitstellung von Anwendungsversionen verwenden aber nicht für die Konfiguration, erhalten Sie möglicherweise eine Fehlermeldung, wenn Sie versuchen, eine Anwendungsversion bereitzustellen, die Konfigurationsänderungen enthält, die in der Regel ein fortlaufendes Update auslösen würden (beispielsweise Konfigurationen, die den Instance-Typ ändern). Um dies zu vermeiden, machen Sie die Konfigurationsänderung in einem separaten Update oder konfigurieren Sie unveränderliche Updates für Bereitstellungs- und Konfigurationsänderungen.

Es ist nicht möglich, ein unveränderliches Update in Verbindung mit Konfigurationsänderungen durchzuführen. Beispiel: Sie können keine [-Einstellungen ändern, die eine Instance-Ersatzung erfordern \(p. 483\)](#), während auch andere Einstellungen aktualisiert werden, oder eine unveränderliche Bereitstellung mit Konfigurationsdateien durchführen, die Konfigurationseinstellungen oder zusätzliche Ressourcen in Ihrem Quellcode ändern. Wenn Sie versuchen, Ressourceneinstellungen zu ändern (z. B. Load Balancer-Einstellungen) und gleichzeitig ein unveränderliches Update durchführen, gibt Elastic Beanstalk einen Fehler zurück.

Wenn Ihre Ressourcenkonfigurationsänderungen nicht abhängig von Ihrer Quellcodeänderung oder Instance-Konfiguration sind, führen Sie sie in zwei Updates hinzu. Wenn sie abhängig sind, führen Sie stattdessen eine [blaue/grüne Bereitstellung \(p. 481\)](#) durch.

Konfigurieren unveränderlicher Updates

Sie können unveränderliche Updates in der Elastic Beanstalk-Konsole aktivieren und konfigurieren.

So aktivieren Sie unveränderliche Updates (Konsole)

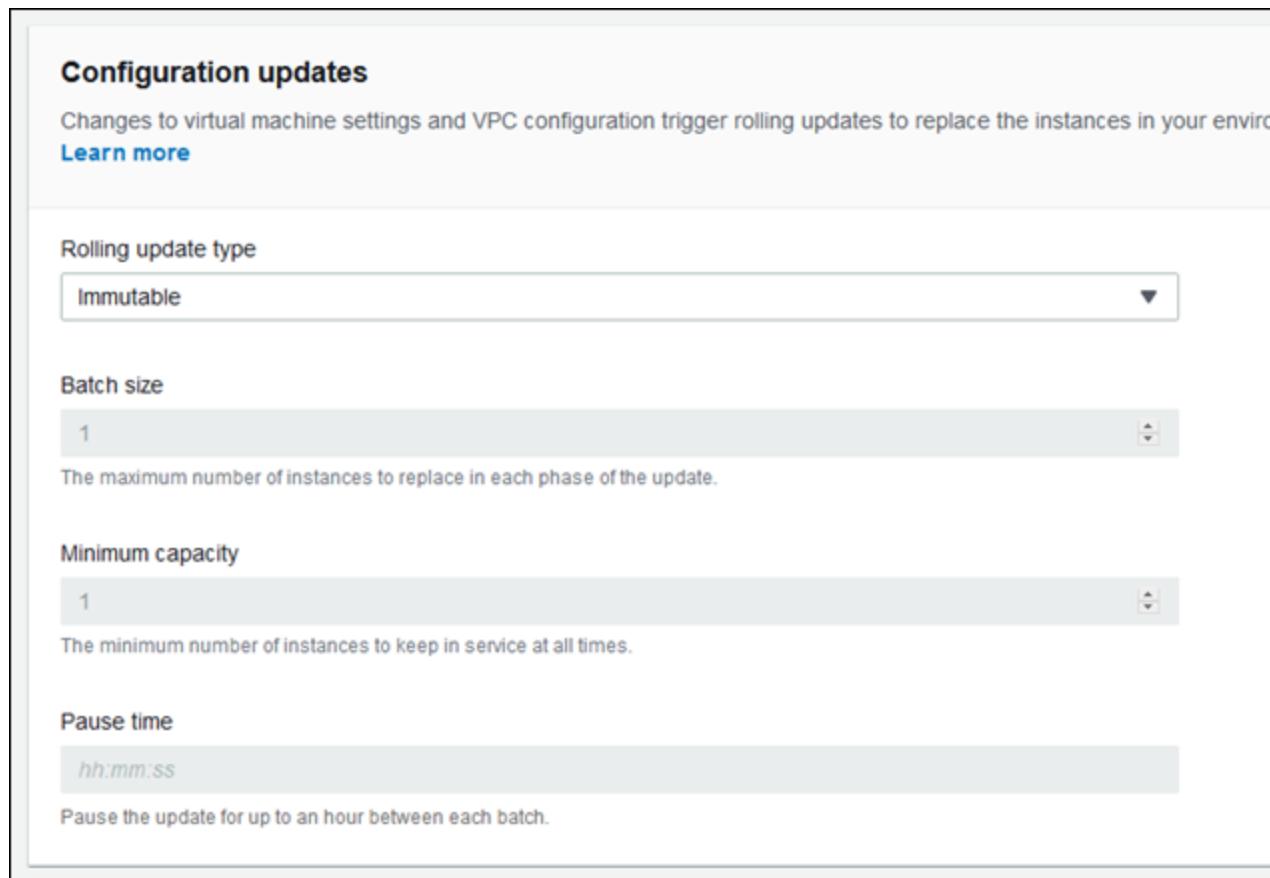
1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.

2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Rolling updates and deployments (Fortlaufende Updates und Bereitstellungen) die Option Edit (Bearbeiten).
5. Legen Sie im Abschnitt Configuration Updates Rolling update type auf Immutable (Unveränderlich) fest.



6. Wählen Sie Apply (Anwenden) aus.

Der aws:autoscaling:updatepolicy:rollingupdate-Namespace

Sie können auch die Optionen im `aws:autoscaling:updatepolicy:rollingupdate`-Namespace zum Konfigurieren unveränderlicher Updates verwenden. Die folgende Beispiel-Konfigurationsdatei (p. 722) ermöglicht unveränderliche Updates für Konfigurationsänderungen.

Example `.ebextensions/immutable-updates.config`

```
option_settings:  
  aws:autoscaling:updatepolicy:rollingupdate:  
    RollingUpdateType: Immutable
```

Die folgende Beispiel-Konfigurationsdatei ermöglicht unveränderliche Updates für Konfigurationsänderungen und Bereitstellungen.

Example .ebextensions/immutable-all.config

```
option_settings:  
  aws:autoscaling:updatepolicy:rollingupdate:  
    RollingUpdateType: Immutable  
  aws:elasticbeanstalk:command:  
    DeploymentPolicy: Immutable
```

Die EB CLI und Elastic Beanstalk-Konsole wenden die empfohlenen Werte für die vorhergehenden Optionen an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter [Empfohlene Werte \(p. 648\)](#).

Aktualisieren der Plattformversion für die Elastic Beanstalk-Umgebung

Elastic Beanstalk veröffentlicht regelmäßig neue Plattformversionen, um alle Linux-basierte und Windows Server-basierte [Plattformen \(p. 32\)](#) zu aktualisieren. Neue Plattformversionen bieten Aktualisierungen der vorhandenen Softwarekomponenten und Unterstützung für neue Funktionen und Konfigurationsoptionen. Weitere Informationen zu Plattformen und Plattformversionen finden Sie unter [Elastic Beanstalk-Plattformen – Glossar \(p. 27\)](#).

Sie können mit der Elastic Beanstalk-Konsole oder der EB CLI die Plattformversion der Umgebung aktualisieren. Abhängig von der Plattformversion, die Sie aktualisieren möchten, empfiehlt Elastic Beanstalk eine von zwei Methoden zum Ausführen von Plattformaktualisierungen.

- [Methode 1 – Aktualisieren der Plattformversion Ihrer Umgebung \(p. 493\)](#) aus. Wir empfehlen diese Methode, wenn Sie auf die neueste Plattformversion in einer Plattformvariante aktualisieren – ohne Änderung an Laufzeit, Webserver, Anwendungsserver, Betriebssystem oder Hauptplattformversion. Dies ist die häufigste und routinemäßige Plattformaktualisierung.
- [Methode 2 – Ausführen einer Blau/Grün-Bereitstellung \(p. 495\)](#) aus. Wir empfehlen diese Methode, wenn Sie auf eine Plattformversion in einer anderen Plattformvariante aktualisieren – mit anderer Laufzeit, anderem Webserver, Anwendungsserver oder Betriebssystem oder einer anderen Hauptplattformversion. Dies ist ein guter Ansatz, wenn Sie die Vorteile der neuen Laufzeitfunktionen oder die neueste Elastic Beanstalk-Funktionalität nutzen möchten oder wenn Sie von einer veralteten oder ausgemusterten Plattformvariante weg wechseln möchten.

Für die [Migration von einer älteren Plattformversion \(p. 502\)](#) ist eine Blau/Grün-Bereitstellung erforderlich, da diese Plattformversionen mit derzeit unterstützten Versionen nicht kompatibel sind.

Für die [Migration einer Linux-Anwendung auf Amazon Linux 2 \(p. 503\)](#) ist eine Blau/Grün-Bereitstellung erforderlich, da Amazon Linux 2-Plattformversionen mit früheren Amazon Linux AMI-Plattformversionen nicht kompatibel sind.

Weitere Unterstützung bei der Auswahl der optimalen Methode zur Plattformaktualisierung , erweitern Sie den Abschnitt für die Plattform Ihrer Umgebung.

Docker

Verwenden Sie [Methode 1 \(p. 493\)](#) zum Ausführen von Plattformaktualisierungen.

Multicontainer-Docker

Verwenden Sie [Methode 1 \(p. 493\)](#) zum Ausführen von Plattformaktualisierungen.

Vorkonfigurierte Docker-Plattform

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere Plattform migrieren, z. B. von Go 1.4 (Docker) auf Go 1.11 oder von Python 3.4 (Docker) auf Python 3.6, verwenden Sie [Methode 2 \(p. 495\)](#).
- Wenn Sie Ihre Anwendung für eine andere Docker-Container-Version migrieren, z. B. von Glassfish 4.1 (Docker) auf Glassfish 5.0 (Docker), verwenden Sie [Methode 2 \(p. 495\)](#).
- Wenn Sie auf eine aktuelle Plattformversion ohne Änderung der Container-Version oder Hauptversion aktualisieren, verwenden Sie [Methode 1 \(p. 493\)](#).

Go

Verwenden Sie [Methode 1 \(p. 493\)](#) zum Ausführen von Plattformaktualisierungen.

Java SE

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere Java-Laufzeitversion migrieren, z. B. von Java 7 auf Java 8, verwenden Sie [Methode 2 \(p. 495\)](#).
- Wenn Sie auf eine aktuelle Plattformversion ohne Änderung der Laufzeitversion aktualisieren, verwenden Sie [Methode 1 \(p. 493\)](#).

Java mit Tomcat

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere Java-Laufzeitversion oder Tomcat-Anwendungsserver-Version aktualisieren, z. B. von Java 7 mit Tomcat 7 auf Java 8 mit Tomcat 8.5, verwenden Sie [Methode 2 \(p. 495\)](#).
- Wenn Sie Ihre Anwendung auf größeren Java mit Tomcat-Plattformversionen (v1.x.x, v2.x.x, und v3.x.x) migrieren, verwenden Sie [Methode 2 \(p. 495\)](#).
- Wenn Sie auf eine aktuelle Plattformversion ohne Änderung der Laufzeitversion, Anwendungsserverversion oder Hauptversion aktualisieren, verwenden Sie [Methode 1 \(p. 493\)](#).

.NET on Windows Server mit IIS

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere Windows-Betriebssystemversion migrieren, z. B. von Windows Server 2008 R2 auf Windows Server 2016, verwenden Sie [Methode 2 \(p. 495\)](#).
- Wenn Sie Ihre Anwendung auf großen Windows Server-Plattformversionen migrieren, finden Sie weitere Informationen unter [Migration von früheren Hauptversionen der Windows Server-Plattform \(p. 196\)](#). Verwenden Sie [Methode 2 \(p. 495\)](#).
- Wenn Ihre Anwendung derzeit auf einer Windows Server-Plattform V2.x.x ausgeführt wird und Sie auf eine aktuelle Plattformversion aktualisieren, verwenden Sie [Methode 1 \(p. 493\)](#).

Note

[Windows Server-Plattformversionen](#) vor v2 werden nicht semantisch versioniert. Sie können nur jeweils die neueste Version einer jeden dieser Windows Server-Plattformhauptversionen starten. Ein Rollback nach einem Upgrade ist nicht möglich.

Node.js

Verwenden Sie [Methode 2 \(p. 495\)](#) zum Ausführen von Plattformaktualisierungen.

PHP

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere PHP-Laufzeitversion migrieren, z. B. von PHP 5.6 auf PHP 7.2, verwenden Sie [Methode 2 \(p. 495\)](#).
- Wenn Sie Ihre Anwendung auf große PHP-Plattformversionen (v1.x.x und v2.x.x) migrieren, verwenden Sie [Methode 2 \(p. 495\)](#).
- Wenn Sie auf eine aktuelle Plattformversion ohne Änderung der Laufzeit- oder -version aktualisieren, verwenden Sie [Methode 1 \(p. 493\)](#).

Python

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere Python-Laufzeitversion migrieren, z. B. von Python 2.7 auf Python 3.6, verwenden Sie [Methode 2 \(p. 495\)](#).
- Wenn Sie Ihre Anwendung auf große Python-Plattformversionen (v1.x.x und v2.x.x) migrieren, verwenden Sie [Methode 2 \(p. 495\)](#).
- Wenn Sie auf eine aktuelle Plattformversion ohne Änderung der Laufzeit- oder -version aktualisieren, verwenden Sie [Methode 1 \(p. 493\)](#).

Ruby

Beachten Sie folgende Fälle:

- Wenn Sie Ihre Anwendung auf eine andere Ruby-Laufzeitversion oder Anwendungsserverversion migrieren, z. B. von Ruby 2.3 mit Puma auf Ruby 2.6 mit Puma migrieren, verwenden Sie [Methode 2 \(p. 495\)](#).
- Wenn Sie Ihre Anwendung auf große Ruby-Plattformversionen (v1.x.x und v2.x.x) migrieren, verwenden Sie [Methode 2 \(p. 495\)](#).
- Wenn Sie auf eine aktuelle Plattformversion ohne Änderung der Laufzeitversion, Anwendungsserverversion oder Hauptversion aktualisieren, verwenden Sie [Methode 1 \(p. 493\)](#).

Methode 1 – Aktualisieren der Plattformversion Ihrer Umgebung

Verwenden Sie diese Methode zum Aktualisieren auf die aktuelle Plattformvariante Ihrer Umgebung. Falls Sie zuvor eine Umgebung mit einer älteren Plattformversion erstellt oder Ihre Umgebung von einer älteren Version aktualisiert haben, können Sie mit dieser Methode eine vorherige Plattformversion wiederherstellen, sofern sich diese in derselben Plattformvariante befindet.

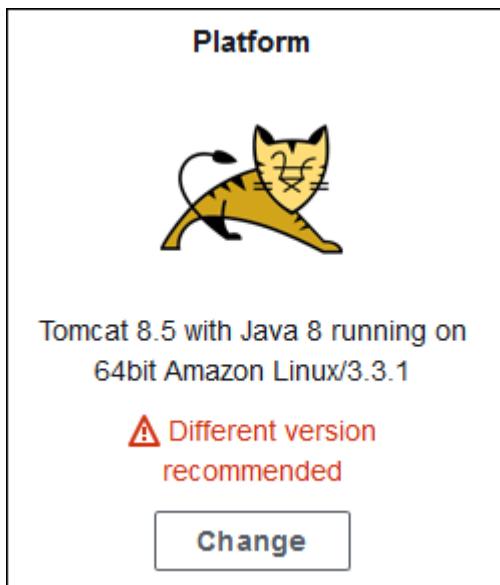
So aktualisieren Sie die Plattformversion Ihrer Umgebung:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite unter Platform (Plattform) die Option Change (Ändern) aus.



4. Wählen Sie im Dialogfeld Update platform version (Plattformversion aktualisieren) eine Plattformversion aus. Die neueste (empfohlene) Plattformversion in der Variante wird automatisch ausgewählt. Sie können auf jede Version aktualisieren, die Sie in der Vergangenheit verwendet haben.

The screenshot shows the 'Update platform version' configuration interface. It includes a warning message about replacing instances during the update, the current platform branch (Tomcat 8.5 with Java 8 running on 64bit Amazon Linux), the current version (3.3.1), and a dropdown menu for the new version (3.3.2 (Recommended)).

Platform branch
Tomcat 8.5 with Java 8 running on 64bit Amazon Linux

Current platform version
3.3.1

New platform version
3.3.2 (Recommended) ▾

5. Wählen Sie Save (Speichern) aus.

Lassen Sie die Plattformaktualisierungen von Elastic Beanstalk verwalten, um den Vorgang zu vereinfachen. Sie können die Umgebung so konfigurieren, dass während eines definierbaren wöchentlichen Wartungsfensters automatisch Neben- und Patch-Versionsaktualisierungen durchgeführt werden. Elastic Beanstalk führt verwaltete Aktualisierungen ohne Ausfallzeit oder Kapazitätseinbußen aus. Falls Instances, auf denen die Anwendung mit der neuen Version ausgeführt wird, die Zustandsprüfungen nicht bestehen, wird die Aktualisierung unverzüglich abgebrochen. Details dazu finden Sie unter [Verwaltete Plattformaktualisierungen \(p. 496\)](#).

Methode 2 – Ausführen einer Blau/Grün-Bereitstellung

Verwenden Sie diese Methode, um auf eine andere Plattformvariante zu aktualisieren – mit anderer Laufzeit, anderem Webserver, Anwendungsserver oder Betriebssystem oder einer anderen Hauptplattformversion. Dies ist in der Regel erforderlich, wenn Sie die Vorteile der neuen Laufzeitfunktionen oder der aktuellen Elastic Beanstalk-Funktionalität nutzen möchten. Es ist auch erforderlich, wenn Sie von einer veralteten oder ausgemusterten Plattformvariante weg migrieren.

Wenn Sie auf größere Plattformversionen oder auf Plattformversionen mit wichtigen Komponentenaktualisierungen aktualisieren möchten, ist die Wahrscheinlichkeit größer, dass Ihre Anwendung oder einige ihrer Aspekte, auf der neuen Plattformversion nicht wie erwartet funktionieren und möglicherweise Änderungen erfordern.

Aktualisieren Sie Ihren lokalen Entwicklungscomputer vor dem Ausführen der Migration auf neuere Laufzeitversionen und andere Komponenten der Plattform, die Sie migrieren wollen. Stellen Sie sicher, dass Ihre Anwendung weiterhin wie erwartet funktioniert und nehmen Sie bei Bedarf Codekorrekturen und -änderungen vor. Verwenden Sie dann die folgende bewährte Methode zur sicheren Migration Ihrer Umgebung auf die neue Plattformversion.

So migrieren Sie Ihre Umgebung auf eine Plattformversion mit wichtigen Aktualisierungen

1. [Erstellen Sie eine neue Umgebung \(p. 435\)](#) unter Verwendung der neuen Plattformzielversion und stellen Sie für sie Ihren Anwendungscode bereit. Die neue Umgebung sollte sich in der Elastic Beanstalk-Anwendung befinden, die die Umgebung enthält, in die Sie migrieren. Beenden Sie die vorhandene Umgebung noch nicht.
2. Verwenden Sie die neue Umgebung, um Ihre Anwendung zu migrieren. Insbesondere:
 - Suchen und beheben Sie Probleme mit der Anwendungskompatibilität, die Sie während der Entwicklungsphase nicht erkennen konnten.
 - Stellen Sie sicher, dass alle Anpassungen, die Ihre Anwendung mithilfe von [Konfigurationsdateien \(p. 722\)](#) vornimmt, in der neuen Umgebung korrekt funktionieren. Hierbei kann es sich z. B. um Optionseinstellungen, zusätzliche installierte Pakete, benutzerdefinierte Sicherheitsrichtlinien und Skripts oder auf Umgebungs-Instances installierte Konfigurationsdateien handeln.
 - Wenn Ihre Anwendung ein benutzerdefiniertes Amazon Machine Image (AMI) verwendet, erstellen Sie ein neues benutzerdefiniertes AMI basierend auf dem AMI der neuen Plattformversion. Weitere Informationen hierzu finden Sie unter [Verwenden eines benutzerdefinierten Amazon Machine Image \(AMI\) \(p. 773\)](#). Dies ist insbesondere erforderlich, wenn Ihre Anwendung die Windows Server-Plattform mit einem benutzerdefinierten AMI nutzt und Sie eine Migration auf eine Windows Server V2-Plattformversion vornehmen. Weitere Informationen über diesen Fall finden Sie auch unter [Migration von früheren Hauptversionen der Windows Server-Plattform \(p. 196\)](#).

Führen Sie so lange Tests durch und stellen Sie Ihre Korrekturen bereit, bis Sie mit der Anwendung in der neuen Umgebung zufrieden sind.

3. Machen Sie die neue Umgebung zu Ihrer Produktionsumgebung, indem Sie ihren CNAME mit dem CNAME der vorhandenen Produktionsumgebung vertauschen. Details dazu finden Sie unter [Blau/Grün-Bereitstellungen mit AWS Elastic Beanstalk \(p. 481\)](#).
4. Wenn Sie mit dem Status Ihrer neuen Umgebung in der Produktion zufrieden sind, schließen Sie die alte Umgebung. Details dazu finden Sie unter [Elastic Beanstalk-Umgebung terminieren \(p. 459\)](#).

Verwaltete Plattformaktualisierungen

AWS Elastic Beanstalk veröffentlicht regelmäßig [Plattformaktualisierungen \(p. 491\)](#) mit Fehlerbehebungen, Softwareupdates und neuen Funktionen. Mit verwalteten Plattformaktualisierungen können Sie die Umgebung so konfigurieren, dass während eines geplanten [Wartungsfensters \(p. 499\)](#) automatisch eine Aktualisierung auf die neueste Version einer Plattform durchgeführt wird. Die Anwendung bleibt während der Aktualisierung aktiv, und zwar ohne Kapazitätseinbußen. Verwaltete Aktualisierungen stehen sowohl für Umgebungen mit einer Instance als auch mit Load Balancing zur Verfügung.

Note

Diese Funktion steht bei [Windows Server-Plattformversionen](#) vor Version 2 (v2) nicht zur Verfügung.

Sie können die Umgebung zur automatischen Ausführung von [Patch-Versionsaktualisierungen \(p. 499\)](#) oder sowohl Patch- als auch Nebenversionsaktualisierungen konfigurieren. Verwaltete Plattformaktualisierungen unterstützen keine Aktualisierungen über Plattformvarianten hinweg (Aktualisierungen für verschiedene Hauptversionen von Plattformkomponenten wie beispielsweise

Betriebssystem, Laufzeit oder Elastic Beanstalk Komponenten), da diese zu nicht abwärtskompatiblen Änderungen führen können.

Sie können Elastic Beanstalk auch so konfigurieren, dass alle Instances der Umgebung während des Wartungsfensters ersetzt werden, auch wenn keine Plattformaktualisierung verfügbar ist. Das Ersetzen aller Instances der Umgebung ist sinnvoll, wenn die Anwendung über einen langen Zeitraum ausgeführt wird und Fehler oder Speicherprobleme auftreten.

In Umgebungen, die am 25. November 2019 oder später mit der Elastic Beanstalk-Konsole erstellt wurden, werden verwaltete Updates standardmäßig aktiviert (sofern möglich). Verwaltete Updates erfordern die Aktivierung einer [verbesserten Integrität \(p. 824\)](#). Die erweiterte Integrität ist standardmäßig aktiviert, wenn Sie eine der [Konfigurationsvoreinstellungen \(p. 442\)](#) auswählen, und deaktiviert, wenn Sie Custom configuration (Benutzerdefinierte Konfiguration) auswählen. Die Konsole kann verwaltete Updates nicht für ältere Plattformversionen aktivieren, die den erweiterten Status nicht unterstützen, oder wenn die erweiterte Integrität deaktiviert ist. Wenn die Konsole verwaltete Updates für eine neue Umgebung aktiviert, wird das Weekly update window (Wöchentliche Aktualisierungsfenster) auf einen zufälligen Zeitpunkt an einem zufälligen Wochentag eingestellt. Der Wert für Update level (Aktualisierungsstufe) ist auf Minor und patch (Nebenstufe und Patch) eingestellt und Instance replacement (Instance-Ersetzung) ist deaktiviert. Sie können verwaltete Updates vor dem letzten Schritt zur Umgebungserstellung deaktivieren oder neu konfigurieren.

Für eine vorhandene Umgebung können Sie verwaltete Plattformaktualisierungen jederzeit mit der Elastic Beanstalk-Konsole konfigurieren.

So konfigurieren Sie verwaltete Plattformaktualisierungen

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Kategorie Managed Updates (Verwaltete Updates) die Option Edit (Bearbeiten) aus.
5. Deaktivieren oder aktivieren Sie Managed updates (Verwaltete Updates).
6. Wenn verwaltete Updates aktiviert sind, wählen Sie ein Wartungsfenster aus und danach eine Einstellung unter Update level (Aktualisierungsstufe) aus.
7. (Optional) Wählen Sie Instance replacement (Instance-Ersetzung) aus, um den wöchentlichen Instance-Ersatz zu aktivieren.

The screenshot shows the 'Modify managed updates' configuration page in the AWS Elastic Beanstalk console. The navigation path is: Elastic Beanstalk > Environments > GettingStartedApp-env > Configuration. The main section is titled 'Managed platform updates'. It includes a description: 'Enable managed platform updates to apply platform updates automatically during a weekly maintenance window while your application stays available during the update process.' Below this, there are settings for 'Managed updates' (checkbox 'Enabled' checked), 'Weekly update window' (set to Tuesday at 12:00 UTC), and 'Update level' (dropdown set to 'Minor and patch'). Under 'Instance replacement', it says 'If enabled, an instance replacement will be scheduled if no other updates are available.' with a checkbox 'Enabled' unchecked. At the bottom right of the configuration panel is a 'Cancel' button.

8. Wählen Sie Apply (Anwenden) aus.

Verwaltete Plattformaktualisierungen erfordern [erweiterte Zustandsberichte \(p. 824\)](#), um zu bestimmen, ob die Anwendung stabil genug ist und die Plattformaktualisierung als erfolgreich eingestuft werden kann. Anweisungen finden Sie unter [Aktivieren der erweiterten Elastic Beanstalk-Integritätsberichte \(p. 832\)](#).

Abschnitte

- [Zur Durchführung von verwalteten Plattformaktualisierungen erforderliche Berechtigungen \(p. 499\)](#)
- [Wartungsfenster für verwaltete Aktualisierung \(p. 499\)](#)
- [Neben- und Patch-Versionsaktualisierungen \(p. 499\)](#)
- [Unveränderliche Umgebungsaktualisierungen \(p. 500\)](#)
- [Verwalten von verwalteten Aktualisierungen \(p. 500\)](#)
- [Namespaces mit der Option für verwaltete Aktionen \(p. 501\)](#)

Zur Durchführung von verwalteten Plattformaktualisierungen erforderliche Berechtigungen

Elastic Beanstalk benötigt die Berechtigung, um in Ihrem Namen eine Plattformaktualisierung zu initiieren. Um diese Berechtigungen zu erhalten, übernimmt Elastic Beanstalk die Servicerolle für verwaltete Aktualisierungen. Wenn Sie die standardmäßige [Servicerolle \(p. 909\)](#) für Ihre Umgebung verwenden, verwendet die Elastic Beanstalk-Konsole diese auch als Servicerolle für verwaltete Aktualisierungen. Die Konsole weist der Servicerolle die [AWSelasticBeanstalkService \(p. 914\)](#)-verwaltete Richtlinie zu. Diese Richtlinie verfügt über alle Berechtigungen, die Elastic Beanstalk zum Ausführen verwalteter Plattformaktualisierungen benötigt.

Details zu weiteren Möglichkeiten für die Einrichtung einer Servicerolle für verwaltete Updates finden Sie unter [the section called “Servicerollen” \(p. 909\)](#).

Note

Wenn Sie [Konfigurationsdateien \(p. 722\)](#) verwenden, um die Umgebung zu erweitern und zusätzliche Ressourcen einzufügen, müssen Sie der Servicerolle für verwaltete Updates in Ihrer Umgebung möglicherweise zusätzliche Berechtigungen hinzufügen. In der Regel ist das Hinzufügen von Berechtigungen erforderlich, wenn namentlich in anderen Abschnitten oder Dateien auf diese Ressourcen verwiesen wird.

Wenn eine Aktualisierung nicht erfolgreich ist, können Sie den Grund auf der Seite [Managed Updates \(Verwaltete Updates\) \(p. 500\)](#) finden.

Wartungsfenster für verwaltete Aktualisierung

Sobald AWS eine neue Version für die Plattform Ihrer Umgebung veröffentlicht, wird von Elastic Beanstalk eine verwaltete Plattformaktualisierung im nächsten wöchentlichen Wartungsfenster geplant. Ein Wartungsfenster überspannt einen Zeitraum von zwei Stunden. Elastic Beanstalk startet während des Wartungsfensters eine zeitgesteuerte Aktualisierung. Die Aktualisierung ist möglicherweise erst nach dem Ende des Fensters abgeschlossen.

Note

In den meisten Fällen plant Elastic Beanstalk Ihre verwaltete Aktualisierung für das nächste wöchentliche Wartungsfenster ein. Das System berücksichtigt die verschiedenen Aspekte der Sicherheit der Aktualisierung und der Serviceverfügbarkeit beim Planen verwalteter Updates. In seltenen Fällen kann eine Aktualisierung nicht für das nächste anstehende Wartungsfenster geplant werden. In diesem Fall versucht es das System während des nächsten Wartungsfensters erneut. Zur manuellen Anwendung der verwalteten Aktualisierung wählen Sie **Apply now (Jetzt anwenden)** wie unter [Verwalten von verwalteten Aktualisierungen \(p. 500\)](#) auf dieser Seite erläutert.

Neben- und Patch-Versionsaktualisierungen

Sie können verwaltete Plattformaktualisierungen so einstellen, dass nur Patch-Versionsaktualisierungen oder sowohl Neben- als auch Patch-Versionsaktualisierungen angewendet werden. Patch-Versionsaktualisierungen sorgen für Fehlerbehebungen sowie Leistungsverbesserungen und können geringfügige Konfigurationsänderungen der Software, Skripts und Konfigurationsoptionen für Instances umfassen. Nebenversionsaktualisierungen liefern die Unterstützung für neue Elastic Beanstalk-Funktionen. Mit verwalteten Plattformaktualisierungen können Sie keine Hauptversionsaktualisierungen ausführen, da mit diesen möglicherweise abwärts inkompatible Änderungen eingeführt werden.

Bei der Plattformversionsnummer steht die zweite Ziffer für die Nebenversionsaktualisierung und die dritte Ziffer bezeichnet die Patch-Version. Beispielsweise gibt die Plattformversion 2.0.7 die Nebenversion 0 und die Patch-Version 7 an.

Unveränderliche Umgebungsaktualisierungen

Verwaltete Plattformaktualisierungen führen [unveränderliche Umgebungsaktualisierungen \(p. 488\)](#) aus, um die Umgebung auf eine neue Plattformversion zu aktualisieren. Bei unveränderlichen Aktualisierungen wird die Umgebung ohne Instance-Ausfallzeit und Umgebungsänderungen aktualisiert, bevor bestätigt ist, dass die Instances mit der neuen Version die Zustandsprüfungen bestehen.

Elastic Beanstalk stellt bei einer unveränderlichen Aktualisierung die neue Plattformversion auf allen zu dem Zeitpunkt ausgeführten Instances bereit. Die neuen Instances verarbeiten die Anforderungen gemeinsam mit den alten Instances. Sofern die neuen Instances alle Zustandsprüfungen bestehen, beendet Elastic Beanstalk die alten Instances, sodass nur Instances mit der neuen Version ausgeführt werden.

Verwaltete Plattformaktualisierungen führen immer unveränderliche Aktualisierungen aus, auch wenn Sie diese außerhalb des Wartungsfensters anwenden. Wenn Sie die Plattformversion im Dashboard ändern, übernimmt Elastic Beanstalk die ausgewählte Richtlinie für Konfigurationsaktualisierungen.

Warning

Einige Richtlinien ersetzen alle Instances während der Bereitstellung oder Aktualisierung. Dadurch gehen alle angesammelten [Amazon EC2-Steigerungsguthaben](#) verloren. Dies geschieht in den folgenden Fällen:

- Verwaltete Plattform-Updates mit aktiviertem Instance-Ersatz
- Unveränderliche Updates
- Bereitstellungen mit unveränderlichen Updates oder Datenverkehrsaufteilung aktiviert

Verwalten von verwalteten Aktualisierungen

Die Elastic Beanstalk-Konsole zeigt detaillierte Informationen zu verwalteten Updates auf der Seite Managed Updates overview (Übersicht über verwaltete Aktualisierungen) an.

So zeigen Sie Informationen zu verwalteten Aktualisierungen an (Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie auf Managed Updates (Verwaltete Updates).

Im Abschnitt Managed Updates overview (Übersicht über die verwalteten Aktualisierungen) finden Sie Informationen zu geplanten und ausstehenden verwalteten Aktualisierungen. Unter History (Verlauf) werden erfolgreiche Aktualisierungen und Fehlversuche aufgeführt.

Sie können eine geplante Aktualisierung auch sofort ausführen (anstatt auf das nächste Wartungsfenster zu warten).

So übernehmen Sie eine verwaltete Plattformaktualisierung sofort (Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.

2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie auf Managed Updates (Verwaltete Updates).
4. Wählen Sie Apply now (Jetzt anwenden) aus.
5. Überprüfen Sie die Aktualisierungsdetails und wählen Sie dann Apply (Übernehmen).

Bei einer verwalteten Plattformaktualisierung außerhalb des Wartungsfensters führt Elastic Beanstalk eine unveränderliche Aktualisierung aus. Wenn Sie die Umgebungsplattform im [Dashboard \(p. 427\)](#) oder mit einem anderen Client aktualisieren, verwendet Elastic Beanstalk den für [Konfigurationsänderungen \(p. 483\)](#) ausgewählten Aktualisierungstyp.

Falls keine verwaltete Aktualisierung geplant ist, wird die Umgebung vermutlich bereits mit der neuesten Version ausgeführt. Es gibt auch andere Gründe dafür, dass keine Aktualisierung geplant ist:

- Eine [Nebenversionsaktualisierung \(p. 499\)](#) ist verfügbar, aber die Umgebung ist nur für die automatische Ausführung von Patch-Versionsaktualisierungen konfiguriert.
- Die Umgebung wurde seit der Veröffentlichung der Aktualisierung nicht geprüft. Elastic Beanstalk prüft in der Regel einmal pro Stunde, ob eine Aktualisierung vorhanden ist.
- Eine Aktualisierung steht aus oder wird gerade ausgeführt.

Mit dem Beginn des Wartungsfensters oder wenn Sie auf Apply now (Jetzt anwenden) klicken, wechseln geplante Aktualisierungen vor der Ausführung in den Status "Pending (Anstehend)".

Namespaces mit der Option für verwaltete Aktionen

Mit den [Konfigurationsoptionen \(p. 646\)](#) in den Namespaces `aws:elasticbeanstalk:managedactions (p. 693)` und `aws:elasticbeanstalk:managedactions:platformupdate (p. 694)` können Sie verwaltete Plattformaktualisierungen aktivieren und konfigurieren.

Mithilfe der Option `ManagedActionsEnabled` werden verwaltete Plattformaktualisierungen aktiviert. Legen Sie für diese Option den Wert `true` fest, um verwaltete Plattformaktualisierungen zu aktivieren, und konfigurieren Sie mit den anderen Optionen das Aktualisierungsverhalten.

Mit `PreferredStartTime` geben Sie den Beginn des wöchentlichen Wartungsfensters im Format `Tag:Stunde:Minute` an.

Legen Sie `UpdateLevel` auf `minor` oder `patch` fest, damit sowohl Neben- als auch Patch-Versionsaktualisierungen bzw. nur Patch-Versionsaktualisierungen ausgeführt werden.

Wenn verwaltete Plattformaktualisierungen aktiviert sind, können Sie den Instance-Ersatz aktivieren. Wählen Sie dazu für die Option `InstanceRefreshEnabled` den Wert `true` aus. Ist diese Einstellung aktiviert, führt Elastic Beanstalk einmal pro Woche eine unveränderliche Aktualisierung Ihrer Umgebung aus, und zwar unabhängig davon, ob eine neue Plattformversion zur Verfügung steht.

Im folgenden Beispiel einer [Konfigurationsdatei \(p. 722\)](#) werden verwaltete Plattformaktualisierungen für Patch-Versionsaktualisierungen in einem Wartungsfenster, das jeden Dienstag um 9:00 Uhr koordinierter Weltzeit (UTC) beginnt, aktiviert.

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
```

Migration Ihrer Anwendung von einer Legacy-Plattformversion

Wenn Sie eine Elastic Beanstalk-Anwendung bereitgestellt haben, die eine Legacy-Plattformversion verwendet, sollten Sie Ihre Anwendung in eine neue Umgebung mit einer nicht veralteten Plattformversion migrieren, sodass Sie Zugriff auf neue Funktionen erhalten. Wenn Sie sich nicht sicher sind, ob Sie Ihre Anwendung mit einer alten Plattformversion ausführen, können Sie dies in der Elastic Beanstalk-Konsole überprüfen. Detaillierte Anweisungen finden Sie unter [So prüfen Sie, ob Sie eine ältere Plattformversion verwenden \(p. 502\)](#).

Welche neuen Funktionen fehlen in älteren Plattformversionen?

Ältere Plattformen unterstützen die folgenden Funktionen nicht:

- Konfigurationsdateien, wie im Thema [Erweiterte Umgebungsanpassung mit Konfigurationsdateien \(.ebextensions\) \(p. 722\)](#) beschrieben
- ELB-Zustandsprüfungen, wie im Thema [Grundlegende Zustandsberichte \(p. 821\)](#) beschrieben
- Instance-Profile, wie im Thema [Elastic Beanstalk Instance-Profile verwalten \(p. 904\)](#) beschrieben
- VPCs, wie im Thema [Verwenden von Elastic Beanstalk mit Amazon VPC \(p. 988\)](#) beschrieben
- Datenschichten, wie im Thema [Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung \(p. 609\)](#) beschrieben
- Worker-Schichten, wie im Thema [Worker-Umgebungen \(p. 16\)](#) beschrieben
- Umgebungen mit einer Instance, wie im Thema [Umgebungstypen \(p. 514\)](#) beschrieben
- Tags, wie im Thema [Ressourcen in Ihren Elastic Beanstalk-Umgebungen markieren \(p. 617\)](#) beschrieben
- Fortlaufende Updates, wie im Thema [Fortlaufende Updates der Elastic Beanstalk-Umgebungskonfiguration \(p. 484\)](#) beschrieben

Warum sind einige Plattformversionen als Legacy gekennzeichnet?

Von einigen älteren Plattformversionen werden die neuesten Elastic Beanstalk-Funktionen nicht unterstützt. Diese Versionen sind als auf der Umgebungsübersichtsseite in der Elastic Beanstalk-Konsole mit (legacy) ((veraltet)) gekennzeichnet.

So prüfen Sie, ob Sie eine ältere Plattformversion verwenden

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Zeigen Sie auf der Umgebungsübersichtsseite den Namen der Platform (Plattform) an.
Ihre Anwendung verwendet eine ältere Plattformversion, wenn neben dem Plattformnamen (legacy) (veraltert) angezeigt wird.

So migrieren Sie Ihre Anwendung

1. Stellen Sie Ihre Anwendung in einer neuen Umgebung bereit. Anweisungen finden Sie unter [Erstellen einer Elastic Beanstalk-Umgebung \(p. 435\)](#).
2. Wenn Sie eine Amazon RDS-DB-Instance haben, aktualisieren Sie die Sicherheitsgruppe Ihrer Datenbank, um den Zugriff auf Ihre EC2-Sicherheitsgruppe für Ihre neue Umgebung zuzulassen. Anweisungen zum Suchen des Namens Ihrer EC2-Sicherheitsgruppe mit der AWS Management Console finden Sie unter [Sicherheitsgruppen \(p. 535\)](#). Weitere Informationen zum Konfigurieren Ihrer EC2-Sicherheitsgruppe finden Sie im Abschnitt „Autorsieren von Netzwerkzugriff auf eine Amazon EC2-Sicherheitsgruppe“ unter [Arbeiten mit DB-Sicherheitsgruppen](#) im Amazon Relational Database Service-Benutzerhandbuch.
3. Tauschen Sie Ihre Umgebungs-URL aus. Anweisungen finden Sie unter [Blau/Grün-Bereitstellungen mit AWS Elastic Beanstalk \(p. 481\)](#).
4. Beenden Sie Ihre alte Umgebung. Anweisungen finden Sie unter [Elastic Beanstalk-Umgebung terminieren \(p. 459\)](#).

Note

Wenn Sie AWS Identity and Access Management (IAM) verwenden, müssen Sie Ihre Richtlinien aktualisieren, damit sie AWS CloudFormation und Amazon RDS umfassen (falls zutreffend).

Weitere Informationen finden Sie unter [Elastic Beanstalk mit AWS Identity and Access Management verwenden \(p. 903\)](#).

Migrieren Ihrer Elastic Beanstalk Linux-Anwendung auf Amazon Linux 2

AWS bietet zwei Versionen von Amazon Linux: [Amazon Linux 2](#) und [Amazon Linux AMI](#). AWS Elastic Beanstalk verwaltet Plattformzweige mit beiden Amazon Linux-Versionen. Details zu Linux-Plattformen finden Sie unter [the section called “Linux-Plattformen” \(p. 33\)](#).

Wenn Ihre Elastic Beanstalk-Anwendung auf einem Amazon Linux AMI-Plattformzweig basiert und die Plattform über neuere Amazon Linux 2-Plattformzweige verfügt, erfahren Sie auf dieser Seite, wie Sie die Umgebungen Ihrer Anwendung auf Amazon Linux 2 migrieren. Für die beiden Plattformgenerationen wird nicht garantiert, dass sie mit Ihrer bestehenden Anwendung abwärtskompatibel sind. Selbst wenn Ihr Anwendungscode auf der neuen Plattformversion erfolgreich bereitgestellt wird, kann er sich aufgrund von Betriebssystem- und Laufzeitunterschieden möglicherweise anders verhalten oder abweichen. Obwohl Amazon Linux AMI und Amazon Linux 2 den gleichen Linux-Kernel teilen, unterscheiden sie sich in ihrem Initialisierungssystem, ihren libc-Versionen, der Compiler-Toolkette und verschiedenen Paketen. Wir haben auch plattformspezifische Versionen von Laufzeit, Build-Tools und anderen Abhängigkeiten aktualisiert. Daher empfehlen wir Ihnen, sich Zeit zu nehmen, Ihre Anwendung gründlich in einer Entwicklungsumgebung zu testen und notwendige Anpassungen vorzunehmen.

Wenn Sie für die Produktion bereit sind, erfordert Elastic Beanstalk eine Blau/Grün-Bereitstellung, um das Upgrade durchzuführen. Details zu Plattformaktualisierungsstrategien finden Sie unter [the section called “Plattformaktualisierungen” \(p. 491\)](#).

Überlegungen für alle Linux-Plattformen

In der folgenden Tabelle werden Überlegungen erläutert, die Sie bei der Planung einer Anwendungsmigration auf Amazon Linux 2 beachten sollten. Diese Überlegungen gelten für alle

Elastic Beanstalk Linux-Plattformen, unabhängig von bestimmten Programmiersprachen oder Anwendungsservern.

Area	Änderungen und Informationen
Konfigurationsdateien	<p>Amazon Linux 2-Plattformen können Sie Konfigurationsdateien (p. 722) wie gewohnt verwenden und alle Abschnitte funktionieren auf die gleiche Weise. Bestimmte Einstellungen funktionieren jedoch möglicherweise nicht so wie bei früheren Amazon Linux AMI-Plattformen. Zum Beispiel:</p> <ul style="list-style-type: none"> • Einige Softwarepakete, die Sie mit einer Konfigurationsdatei installieren, sind unter Umständen auf Amazon Linux 2 nicht verfügbar, oder ihre Namen haben sich geändert. • Einige plattformspezifische Konfigurationsoptionen wurden von ihren plattformspezifischen Namespaces in andere plattformunabhängige Namespaces verschoben. • Proxy-Konfigurationsdateien, die im Verzeichnis <code>.ebextensions/nginx</code> bereitgestellt werden, sollten in das Plattform-Hooks-Verzeichnis <code>.platform/nginx</code> verschoben werden. Für Einzelheiten erweitern Sie den Abschnitt Reverse-Proxy-Konfiguration unter the section called “Erweitern von Linux-Plattformen” (p. 34). <p>Wir raten zur Verwendung von Plattform-Hooks für die Ausführung von benutzerdefiniertem Code auf Ihren Umgebungs-Instances. Sie können in <code>.ebextensions</code>-Konfigurationsdateien weiterhin Befehle und Containerbefehle verwenden, aber sie sind nicht mehr so einfach in der Verwendung. Beispielsweise kann das Schreiben von Befehlsskripten in einer YAML-Datei umständlich und schwer zu testen sein.</p> <p>Sie müssen weiterhin <code>.ebextensions</code>-Konfigurationsdateien für jedes Skript verwenden, das einen Verweis auf eine AWS CloudFormation-Ressource benötigt.</p>
Plattform-Hooks	<p>Amazon Linux 2-Plattformen bieten eine neue Möglichkeit, die Plattform Ihrer Umgebung zu erweitern, indem ausführbare Dateien zu Hook-Verzeichnissen auf den Instances der Umgebung hinzugefügt werden. Bei früheren Linux-Plattformversionen haben Sie möglicherweise benutzerdefinierte Plattform-Hooks (p. 1123) verwendet. Diese Hooks wurden nicht für verwaltete Plattformen entwickelt und wurden nicht unterstützt, konnten aber in einigen Fällen auf nützliche Weise funktionieren. Bei Amazon Linux 2-Plattformversionen funktionieren keine benutzerdefinierte Plattform-Hooks. Sie sollten alle Hooks auf die neuen Plattform-Hooks migrieren. Details erhalten Sie im Abschnitt Plattform-Hooks in the section called “Erweitern von Linux-Plattformen” (p. 34).</p>
Unterstützte Proxy-Server	<p>Amazon Linux 2-Plattformversionen unterstützen die gleichen Reverse-Proxy-Server wie jede Plattform, die in ihren Amazon Linux-AMI Plattformversionen unterstützt wird. Alle Amazon Linux 2-Plattformversionen verwenden nginx als standardmäßigen Reverse-Proxy-Server. Die Plattformen Tomcat, Node.js, PHP und Python unterstützen ebenfalls Apache HTTPD als Alternative. Alle Plattformen ermöglichen eine einheitliche Konfiguration des Proxy-Servers wie in diesem Abschnitt beschrieben. Die Konfiguration des Proxy-Servers ist jedoch ein wenig anders als auf dem Amazon Linux-AMI. Dies sind die Unterschiede für alle Plattformen:</p> <ul style="list-style-type: none"> • Der Standard ist nginx – Der Standard-Proxy-Server für alle Amazon Linux 2-Plattformversionen ist nginx. Bei Amazon Linux-AMI-Plattformversionen von Tomcat, PHP und Python war der Standard-Proxy-Server Apache HTTPD. • Konsistenter Namespace – Alle Amazon Linux 2-Plattformversionen verwenden den Namespace <code>aws:elasticbeanstalk:environment:proxy</code>, um den Proxy-Server zu konfigurieren. Bei Amazon Linux-AMI-Plattformversionen war dies eine Entscheidung pro Plattform und Node.js verwendete einen anderen Namespace.

Area	Änderungen und Informationen
	<ul style="list-style-type: none"> Speicherort der Konfigurationsdatei – Sie sollten Proxy-Konfigurationsdateien unter allen Amazon Linux 2-Plattformversionen in den Verzeichnissen <code>.platform/nginx</code> und <code>.platform/httpd</code> ablegen. Auf Amazon Linux-AMI Plattformversionen waren diese Speicherorte <code>.ebextensions/nginx</code> bzw. <code>.ebextensions/httpd</code>. <p>Informationen zu plattformspezifischen Proxy-Konfigurationsänderungen finden Sie unter the section called “Plattformspezifische Überlegungen” (p. 505). Weitere Informationen zur Proxy-Konfiguration auf Amazon Linux 2-Plattformen finden Sie im Abschnitt Reverse-Proxy-Konfiguration unter the section called “Erweitern von Linux-Plattformen” (p. 34).</p>
Instance-Profil	Amazon Linux 2-Plattformen erfordern, dass ein Instance-Profil konfiguriert wird. Die Umgebung wird ohne Profil möglicherweise zunächst erfolgreich erstellt, kurz nach der Erstellung können in der Umgebung aber Fehler auftreten, wenn Aktionen, die ein Instance-Profil erfordern, fehlschlagen. Details dazu finden Sie unter the section called “Instance-Profile” (p. 904) .
Erweiterter Zustand	Amazon Linux 2-Plattformversionen verbessern standardmäßig die Integrität. Dies ist eine Änderung, wenn Sie nicht die Elastic Beanstalk-Konsole zum Erstellen von Umgebungen verwenden. Die Konsole ermöglicht standardmäßig möglichst eine erweiterte Integrität, unabhängig von der Plattformversion. Details dazu finden Sie unter the section called “Erweiterte Zustandsberichte und Überwachung” (p. 824) .
Benutzerdefinierte AMI	Wenn Ihre Umgebung ein benutzerdefiniertes AMI (p. 773) verwendet, erstellen Sie mithilfe einer Elastic Beanstalk Amazon Linux 2-Plattform ein neues AMI für Ihre neue Umgebung basierend auf Amazon Linux 2.
Benutzerdefinierte Plattformen	Die verwalteten AMIs der Amazon Linux 2-Plattformversionen unterstützen keine benutzerdefinierten Plattformen (p. 1117) .

Plattformspezifische Überlegungen

In diesem Abschnitt werden Migrationsüberlegungen behandelt, die für bestimmte Elastic Beanstalk Linux-Plattformen spezifisch sind.

Docker

In der folgenden Tabelle sind Migrationsinformationen für die Amazon Linux 2-Plattformversionen in der [Docker-Plattform \(p. 46\)](#) aufgeführt.

Area	Änderungen und Informationen
Speicher	<p>Elastic Beanstalk konfiguriert Docker so, dass Speichertreiber zum Speichern von Docker-Images und Containerdaten verwendet werden. Auf Amazon Linux AMI verwendete Elastic Beanstalk den Device Mapper storage driver (Device Mapper-Speichertreiber). Um die Leistung zu verbessern, stellte Elastic Beanstalk ein zusätzliches Amazon EBS-Volume bereit. Bei Amazon Linux 2-Docker-Plattformversionen verwendet Elastic Beanstalk den OverlayFS-Speichertreiber und erzielt eine noch bessere Leistung, ohne dass ein weiteres separates Volume benötigt wird.</p> <p>Wenn Sie mit Amazon Linux AMI die Option <code>BlockDeviceMappings</code> des Namespace <code>aws:autoscaling:launchconfiguration</code> verwendet haben, um benutzerdefinierte Speicher-Volumes zu einer Docker-Umgebung hinzuzufügen, haben wir Ihnen empfohlen, auch das Amazon EBS-Volume <code>/dev/xvdcz</code> hinzuzufügen, das durch Elastic Beanstalk bereitgestellt wurde. Elastic Beanstalk stellt dieses Volume nicht mehr bereit, daher sollten Sie es aus Ihren Konfigurationsdateien entfernen. Details dazu finden Sie unter AWS CloudFormation-Stacks für Docker-Container (p. 1118).</p>

Area	Änderungen und Informationen
	the section called “Docker-Konfiguration auf Amazon Linux AMI (Vorgängerversion von Amazon Linux 2)” (p. 92).
Private Repository-Authentifizierung	Wenn Sie eine Docker-generierte Authentifizierungsdatei bereitstellen, um eine Verbindung mit einem privaten Repository herzustellen, müssen Sie sie nicht mehr in das ältere Format konvertieren, das für die Amazon Linux-AMI Docker-Plattformversionen erforderlich ist. Amazon Linux-Docker 2-Plattformversionen unterstützen das neue Format. Details dazu finden Sie unter the section called “Verwenden von Images aus einem privaten Repository” (p. 90).
Proxy-Server	Amazon Linux 2-Docker-Plattformversionen unterstützen keine eigenständigen Container, die nicht hinter einem Proxy-Server ausgeführt werden. Bei Amazon Linux-AMI Docker-Plattformversionen war dies früher durch den Wert <code>none</code> der Option <code>ProxyServer</code> im Namespace <code>aws:elasticbeanstalk:environment:proxy</code> möglich.

Go

In der folgenden Tabelle werden Migrationsinformationen für die Amazon Linux 2-Plattformversionen in der [Go-Plattform \(p. 101\)](#) aufgelistet.

Area	Änderungen und Informationen
Übergeben eines Portwertes	Elastic Beanstalk übergibt auf Amazon Linux 2-Plattformen keinen Portwert über die <code>PORT</code> -Umgebungsvariable an Ihren Anwendungsprozess. Sie können dieses Verhalten für Ihren Prozess simulieren, indem Sie eine <code>PORT</code> -Umgebungseigenschaft selbst konfigurieren. Wenn Sie jedoch mehrere Prozesse haben und darauf zählen, dass Elastic Beanstalk inkrementelle Portwerte an Ihre Prozesse übergibt (5000, 5100, 5200 usw.), sollten Sie Ihre Implementierung ändern. Für Einzelheiten erweitern Sie den Abschnitt Reverse-Proxy-Konfiguration unter the section called “Erweitern von Linux-Plattformen” (p. 34).

Amazon Corretto

In der folgenden Tabelle werden Migrationsinformationen für die Corretto-Plattformzweige in der [Java SE-Plattform \(p. 128\)](#) aufgelistet.

Area	Änderungen und Informationen
Corretto vs. OpenJDK	Zur Implementierung von Java Platform, Standard Edition (Java SE), verwenden Amazon Linux 2-Plattformzweige Amazon Corretto , eine AWS-Verteilung des Open Java Development Kit (OpenJDK). Frühere Elastic Beanstalk Java SE-Plattformzweige verwenden die in Amazon Linux AMI enthaltenen OpenJDK-Pakete.
Build-Tools	Amazon Linux 2-Plattformen haben neuere Versionen der Build-Tools: <code>gradle</code> , <code>maven</code> und <code>ant</code> .
JAR-Dateibehandlung	Wenn Ihr Quell-Bundle (ZIP-Datei) auf Amazon Linux 2-Plattformen eine einzelne JAR-Datei und keine anderen Dateien enthält, benennt Elastic Beanstalk die JAR-Datei nicht mehr in <code>application.jar</code> um. Die Umbenennung erfolgt nur, wenn Sie eine JAR-Datei alleine und nicht innerhalb einer ZIP-Datei übermitteln.
Übergeben eines Portwertes	Elastic Beanstalk übergibt auf Amazon Linux 2-Plattformen keinen Portwert über die <code>PORT</code> -Umgebungsvariable an Ihren Anwendungsprozess. Sie können dieses Verhalten für Ihren Prozess simulieren, indem Sie eine <code>PORT</code> -Umgebungseigenschaft selbst

Area	Änderungen und Informationen
	konfigurieren. Wenn Sie jedoch mehrere Prozesse haben und darauf zählen, dass Elastic Beanstalk inkrementelle Portwerte an Ihre Prozesse übergibt (5000, 5100, 5200 usw.), sollten Sie Ihre Implementierung ändern. Für Einzelheiten erweitern Sie den Abschnitt Reverse-Proxy-Konfiguration unter the section called “Erweitern von Linux-Plattformen” (p. 34) .
Java 7	Elastic Beanstalk unterstützt keinen Zweig der Amazon Linux 2 Java 7-Plattform. Wenn Sie eine Java 7-Anwendung haben, migrieren Sie sie auf Corretto 8 oder Corretto 11.

Tomcat

In der folgenden Tabelle werden Migrationsinformationen für die Amazon Linux 2-Plattformversionen in der [Tomcat-Plattform \(p. 117\)](#) aufgelistet.

Area	Änderungen und Informationen						
Konfigurationsoptionen	<p>Bsp: Amazon Linux 2-Plattformversionen unterstützt Elastic Beanstalk nur eine Teilmenge der Konfigurationsoptionen und Optionswerte im Namespace <code>aws:elasticbeanstalk:environment:proxy</code>. Im Folgenden finden Sie Informationen zur Migration für die einzelnen Optionen.</p> <table border="1"> <thead> <tr> <th>Option</th><th>Informationen zur Migration</th></tr> </thead> <tbody> <tr> <td>GzipCompression</td><td>Nicht unterstützt auf Amazon Linux 2-Plattformversionen.</td></tr> <tr> <td>ProxyServer</td><td>Amazon Linux 2-Tomcat-Plattformversionen unterstützen sowohl Proxy-Server mit nginx als auch mit Apache HTTPD Version 2.4. Apache Version 2.2 wird jedoch nicht unterstützt. Bei Amazon Linux AMI-Plattformversionen war der Standard-Proxy Apache 2.4. Wenn Sie die Standard-Proxy-Einstellung verwendet und benutzerdefinierte Proxy-Konfigurationsdateien hinzugefügt haben, sollte Ihre Proxy-Konfiguration auf Amazon Linux 2 weiterhin funktionieren. Wenn Sie jedoch den Optionswert <code>apache/2.2</code> verwendet haben, müssen Sie nun Ihre Proxy-Konfiguration auf Apache Version 2.4 migrieren.</td></tr> </tbody> </table> <p>Die Option <code>xx:MaxPermSize</code> im Namespace <code>aws:elasticbeanstalk:container:tomcat:jvmoptions</code> wird auf Amazon Linux 2-Plattformversionen nicht unterstützt. Die JVM-Einstellung zum Ändern der Größe der permanenten Generation gilt nur für Java 7 und früher und ist daher nicht für Amazon Linux 2-Plattformversionen anwendbar.</p>	Option	Informationen zur Migration	GzipCompression	Nicht unterstützt auf Amazon Linux 2-Plattformversionen.	ProxyServer	Amazon Linux 2-Tomcat-Plattformversionen unterstützen sowohl Proxy-Server mit nginx als auch mit Apache HTTPD Version 2.4. Apache Version 2.2 wird jedoch nicht unterstützt. Bei Amazon Linux AMI-Plattformversionen war der Standard-Proxy Apache 2.4. Wenn Sie die Standard-Proxy-Einstellung verwendet und benutzerdefinierte Proxy-Konfigurationsdateien hinzugefügt haben, sollte Ihre Proxy-Konfiguration auf Amazon Linux 2 weiterhin funktionieren. Wenn Sie jedoch den Optionswert <code>apache/2.2</code> verwendet haben, müssen Sie nun Ihre Proxy-Konfiguration auf Apache Version 2.4 migrieren.
Option	Informationen zur Migration						
GzipCompression	Nicht unterstützt auf Amazon Linux 2-Plattformversionen.						
ProxyServer	Amazon Linux 2-Tomcat-Plattformversionen unterstützen sowohl Proxy-Server mit nginx als auch mit Apache HTTPD Version 2.4. Apache Version 2.2 wird jedoch nicht unterstützt. Bei Amazon Linux AMI-Plattformversionen war der Standard-Proxy Apache 2.4. Wenn Sie die Standard-Proxy-Einstellung verwendet und benutzerdefinierte Proxy-Konfigurationsdateien hinzugefügt haben, sollte Ihre Proxy-Konfiguration auf Amazon Linux 2 weiterhin funktionieren. Wenn Sie jedoch den Optionswert <code>apache/2.2</code> verwendet haben, müssen Sie nun Ihre Proxy-Konfiguration auf Apache Version 2.4 migrieren.						
Anwendungspfad	Auf Amazon Linux 2-Plattformen lautet der Pfad zum Verzeichnis der Anwendung auf Amazon EC2-Instances Ihrer Umgebung <code>/var/app/current</code> . Auf Amazon Linux AMI-Plattformen lautete er <code>/var/lib/tomcat8/webapps</code> .						

Node.js

In der folgenden Tabelle werden Migrationsinformationen für die Amazon Linux 2-Plattformversionen in der [Node.js-Plattform \(p. 250\)](#) aufgelistet.

Area	Änderungen und Informationen												
Installierte Node.js-Versionen	<p>Elastic Beanstalk verwaltet auf Amazon Linux 2-Plattformen mehrere Node.js-Plattformzweige und installiert auf den einzelnen Plattformversionen nur die jeweils neueste Version der Node.js-Hauptversion für den Plattformzweig. Beispielsweise wird für jede Plattformversion im Node.js 12-Plattformzweig standardmäßig nur Node.js 12.x.y installiert. Im Fall von Amazon Linux AMI-Plattformversionen sind auf jeder Plattformversion die verschiedenen Versionen der verschiedenen Node.js-Versionen installiert und es wird nur ein einziger Plattformzweig gewartet.</p> <p>Wählen Sie den Node.js-Plattformzweig für die Node.js-Hauptversion aus, die Ihre Anwendung benötigt.</p>												
Apache HTTPD-Protokolldateien	<p>Wenn Sie auf Amazon Linux 2-Plattformen den Apache-HTTPD-Proxy-Server verwenden, heißen die HTTPD-Protokolldateien <code>access_log</code> und <code>error_log</code>, was mit allen anderen Plattformen konsistent ist, die Apache HTTPD unterstützen. Auf Amazon Linux AMI-Plattformversionen heißen diese Protokolldateien <code>access.log</code> bzw. <code>error.log</code>.</p> <p>Weitere Informationen zu Namen und Speicherorten von Protokolldateien für alle Plattformen finden Sie unter the section called "Wie Elastic Beanstalk CloudWatch Logs einrichtet" (p. 883).</p>												
Konfigurationsoptionen	<p>Amazon Linux 2-Plattformen unterstützt Elastic Beanstalk die Konfigurationsoptionen im Namespace <code>aws:elasticbeanstalk:container:nodejs</code> nicht. Für einige Optionen gibt es Alternativen. Im Folgenden finden Sie Informationen zur Migration für die einzelnen Optionen.</p> <table border="1"> <thead> <tr> <th>Option</th><th>Informationen zur Migration</th></tr> </thead> <tbody> <tr> <td><code>NodeCommand</code></td><td>Verwenden Sie ein <code>Procfile</code> oder das Schlüsselwort <code>scripts</code> in einer <code>package.json</code>-Datei, um das Startskript anzugeben.</td></tr> <tr> <td><code>NodeVersion</code></td><td>Verwenden Sie in einer <code>package.json</code>-Datei das Schlüsselwort <code>engines</code>, um die Node.js-Version anzugeben. Beachten Sie, dass Sie jeweils nur eine Node.js-Version angeben können, die Ihrem Plattformzweig entspricht. Wenn Sie beispielsweise den Plattformzweig Node.js 12 verwenden, können Sie nur eine 12.x.y Node.js-Version angeben. Details dazu finden Sie unter the section called "Angeben von Node.js-Abhängigkeiten mit einer package.json-Datei" (p. 256).</td></tr> <tr> <td><code>GzipCompression</code></td><td>Nicht unterstützt auf Amazon Linux 2-Plattformversionen.</td></tr> <tr> <td><code>ProxyServer</code></td><td> <p>Bei Amazon Linux 2-Node.js-Plattformversionen wurde diese Option in den Namespace <code>aws:elasticbeanstalk:environment:proxy</code> verschoben. Sie können zwischen <code>nginx</code> (Standardeinstellung) und <code>apache</code> wählen.</p> <p>Darüber hinaus unterstützen Amazon Linux 2-Node.js-Plattformversionen keine eigenständigen Anwendungen, die nicht hinter einem Proxy-Server ausgeführt werden. Bei Amazon Linux-AMI Node.js-Plattformversionen war dies früher durch den Wert <code>none</code> der Option <code>ProxyServer</code> im Namespace <code>aws:elasticbeanstalk:container:nodejs</code> möglich. Wenn in Ihrer Umgebung eine eigenständige Anwendung ausgeführt wird, aktualisieren Sie den Code so, dass dem Port zugehört wird, an den der Proxy-Server (<code>nginx</code> oder <code>Apache</code>) den Datenverkehr weiterleitet.</p> </td></tr> <tr> <td></td><td><pre>var port = process.env.PORT 8080;</pre></td></tr> </tbody> </table>	Option	Informationen zur Migration	<code>NodeCommand</code>	Verwenden Sie ein <code>Procfile</code> oder das Schlüsselwort <code>scripts</code> in einer <code>package.json</code> -Datei, um das Startskript anzugeben.	<code>NodeVersion</code>	Verwenden Sie in einer <code>package.json</code> -Datei das Schlüsselwort <code>engines</code> , um die Node.js-Version anzugeben. Beachten Sie, dass Sie jeweils nur eine Node.js-Version angeben können, die Ihrem Plattformzweig entspricht. Wenn Sie beispielsweise den Plattformzweig Node.js 12 verwenden, können Sie nur eine 12.x.y Node.js-Version angeben. Details dazu finden Sie unter the section called "Angeben von Node.js-Abhängigkeiten mit einer package.json-Datei" (p. 256) .	<code>GzipCompression</code>	Nicht unterstützt auf Amazon Linux 2-Plattformversionen.	<code>ProxyServer</code>	<p>Bei Amazon Linux 2-Node.js-Plattformversionen wurde diese Option in den Namespace <code>aws:elasticbeanstalk:environment:proxy</code> verschoben. Sie können zwischen <code>nginx</code> (Standardeinstellung) und <code>apache</code> wählen.</p> <p>Darüber hinaus unterstützen Amazon Linux 2-Node.js-Plattformversionen keine eigenständigen Anwendungen, die nicht hinter einem Proxy-Server ausgeführt werden. Bei Amazon Linux-AMI Node.js-Plattformversionen war dies früher durch den Wert <code>none</code> der Option <code>ProxyServer</code> im Namespace <code>aws:elasticbeanstalk:container:nodejs</code> möglich. Wenn in Ihrer Umgebung eine eigenständige Anwendung ausgeführt wird, aktualisieren Sie den Code so, dass dem Port zugehört wird, an den der Proxy-Server (<code>nginx</code> oder <code>Apache</code>) den Datenverkehr weiterleitet.</p>		<pre>var port = process.env.PORT 8080;</pre>
Option	Informationen zur Migration												
<code>NodeCommand</code>	Verwenden Sie ein <code>Procfile</code> oder das Schlüsselwort <code>scripts</code> in einer <code>package.json</code> -Datei, um das Startskript anzugeben.												
<code>NodeVersion</code>	Verwenden Sie in einer <code>package.json</code> -Datei das Schlüsselwort <code>engines</code> , um die Node.js-Version anzugeben. Beachten Sie, dass Sie jeweils nur eine Node.js-Version angeben können, die Ihrem Plattformzweig entspricht. Wenn Sie beispielsweise den Plattformzweig Node.js 12 verwenden, können Sie nur eine 12.x.y Node.js-Version angeben. Details dazu finden Sie unter the section called "Angeben von Node.js-Abhängigkeiten mit einer package.json-Datei" (p. 256) .												
<code>GzipCompression</code>	Nicht unterstützt auf Amazon Linux 2-Plattformversionen.												
<code>ProxyServer</code>	<p>Bei Amazon Linux 2-Node.js-Plattformversionen wurde diese Option in den Namespace <code>aws:elasticbeanstalk:environment:proxy</code> verschoben. Sie können zwischen <code>nginx</code> (Standardeinstellung) und <code>apache</code> wählen.</p> <p>Darüber hinaus unterstützen Amazon Linux 2-Node.js-Plattformversionen keine eigenständigen Anwendungen, die nicht hinter einem Proxy-Server ausgeführt werden. Bei Amazon Linux-AMI Node.js-Plattformversionen war dies früher durch den Wert <code>none</code> der Option <code>ProxyServer</code> im Namespace <code>aws:elasticbeanstalk:container:nodejs</code> möglich. Wenn in Ihrer Umgebung eine eigenständige Anwendung ausgeführt wird, aktualisieren Sie den Code so, dass dem Port zugehört wird, an den der Proxy-Server (<code>nginx</code> oder <code>Apache</code>) den Datenverkehr weiterleitet.</p>												
	<pre>var port = process.env.PORT 8080;</pre>												

Area	Änderungen und Informationen	
	Option	Informationen zur Migration
		<pre>app.listen(port, function() { console.log('Server running at http://127.0.0.1:%s', port); });</pre>

PHP

In der folgenden Tabelle werden Migrationsinformationen für die Amazon Linux 2-Plattformversionen in der [PHP-Plattform \(p. 288\)](#) aufgelistet.

Area	Änderungen und Informationen
PHP-Dateiverarbeitung	Auf Amazon Linux 2-Plattformen werden PHP-Dateien mit PHP-FPM (einem CGI-Prozessmanager) verarbeitet. Auf Amazon Linux-AMI-Plattformen wird mod_php (ein Apache-Modul) verwendet.
Proxy-Server	Amazon Linux 2-PHP-Plattformversionen unterstützen sowohl Proxy-Server mit nginx als auch Apache HTTPD. Standardmäßig wird nginx verwendet. Amazon Linux AMI PHP-Plattformversionen unterstützen nur Apache HTTPD. Wenn Sie benutzerdefinierte Apache-Konfigurationsdateien hinzugefügt haben, können Sie die Option <code>ProxyServer</code> im Namespace <code>aws:elasticbeanstalk:environment:proxy</code> auf <code>apache</code> festlegen.

Python

In der folgenden Tabelle werden Migrationsinformationen für die Amazon Linux 2-Plattformversionen in der [Python-Plattform \(p. 355\)](#) aufgelistet.

Area	Änderungen und Informationen
WSGI-Server	Auf Amazon Linux 2-Plattformen ist Gunicorn der Standard-WSGI-Server. Standardmäßig überwacht Gunicorn Port 8000. Der Port kann sich von dem unterscheiden, den Ihre Anwendung auf der Amazon Linux-AMI-Plattform verwendet hat. Wenn Sie die <code>WSGIPath</code> -Option des aws:elasticbeanstalk:container:python (p. 719) -Namespace festlegen, ersetzen Sie den Wert durch die Syntax von Gunicorn. Details dazu finden Sie unter the section called “Python-Konfigurations-Namespaces” (p. 358) . Alternativ können Sie einen <code>Procfile</code> verwenden, um den WSGI-Server anzugeben und zu konfigurieren. Details dazu finden Sie unter the section called “Procfile” (p. 359) .
Anwendungspfad	Auf Amazon Linux 2-Plattformen lautet der Pfad zum Verzeichnis der Anwendung auf Amazon EC2-Instances Ihrer Umgebung <code>/var/app/current</code> . Auf Amazon Linux AMI-Plattformen lautete er <code>/opt/python/current/app</code> .
Proxy-Server	Amazon Linux 2-Python-Plattformversionen unterstützen sowohl Proxy-Server mit nginx als auch Apache HTTPD. Standardmäßig wird nginx verwendet. Amazon Linux AMI Python-Plattformversionen unterstützen nur Apache HTTPD. Wenn Sie benutzerdefinierte Apache-Konfigurationsdateien hinzugefügt haben, können Sie die Option <code>ProxyServer</code> im Namespace <code>aws:elasticbeanstalk:environment:proxy</code> auf <code>apache</code> festlegen.

Ruby

In der folgenden Tabelle werden Migrationsinformationen für die Amazon Linux 2-Plattformversionen in der [Ruby-Plattform \(p. 383\)](#) aufgelistet.

Area	Änderungen und Informationen
Installierte Ruby-Versionen	<p>Auf Amazon Linux 2-Plattformen installiert Elastic Beanstalk auf den einzelnen Plattformversionen nur die neueste Version einer einzelnen Ruby-Version für den betreffenden Plattformzweig. Beispielsweise ist auf den einzelnen Plattformversionen im Ruby 2.6-Plattformzweig nur Ruby 2.6.x installiert. Auf Amazon Linux AMI-Plattformversionen sind die neuesten Versionen mehrerer Ruby-Versionen installiert, z. B. 2.4.x, 2.5.x und 2.6.x.</p> <p>Wenn Ihre Anwendung eine Ruby-Version verwendet, die nicht dem von Ihnen verwendeten Plattformzweig entspricht, sollten Sie zu einem Plattformzweig wechseln, auf dem die richtige Ruby-Version für Ihre Anwendung installiert ist.</p>
Anwendungsse	<p>Auf Amazon Linux 2-Plattformen installiert Elastic Beanstalk auf allen Ruby-Plattformversionen ausschließlich den Puma-Anwendungsserver. Sie können eine <code>Procfile</code>-Datei verwenden, um einen anderen Anwendungsserver zu starten, und eine <code>Gemfile</code>-Datei, um ihn zu installieren.</p> <p>Auf der Amazon Linux AMI-Plattform werden zwei Varianten von Plattformzweigen für jede Ruby-Version unterstützt – Variante mit dem Puma-Anwendungsserver und eine Variante mit dem Passenger-Anwendungsserver. Wenn Ihre Anwendung Passenger verwendet, können Sie Ihre Ruby-Umgebung für die Installation und Verwendung von Passenger konfigurieren.</p> <p>Weitere Informationen und Beispiele finden Sie unter the section called “Die Ruby-Plattform” (p. 383).</p>

Abbrechen von Umgebungskonfigurations-Updates und Anwendungsbereitstellungen

Sie können angefangene Updates abbrechen, die durch Änderungen der Umgebungskonfiguration ausgelöst wurden. Sie können auch die laufende Bereitstellung einer neuen Anwendungsversion abbrechen. Sie können beispielsweise eine Aktualisierung abbrechen, wenn Sie sich dazu entscheiden, die bestehende Umgebungskonfiguration weiter zu verwenden, anstatt neue Einstellungen für die Umgebungskonfiguration anzuwenden. Oder Sie stellen möglicherweise fest, dass bei Ihrer Anwendungsversion Probleme vorliegen, die dazu führen, dass sie nicht gestartet oder ordnungsgemäß ausgeführt werden. Durch das Abbrechen einer Umgebungsaktualisierung oder Anwendungsversion können verhindern, dass Sie warten müssen, bis das Update oder der Bereitstellungsprozess durchgeführt wurde, bevor Sie erneut versuchen, die Umgebung oder Anwendungsversion zu aktualisieren.

Note

Während der Bereinigungsphase, in der alte Ressourcen entfernt werden, die nicht mehr benötigt werden, können Sie das Update nicht mehr abbrechen, nachdem der letzte Stapel an Instances aktualisiert wurde.

Elastic Beanstalk führt das Rollback genau wie das letzte erfolgreiche Update durch. Wenn Sie beispielsweise zeitbasierte fortlaufende Updates in Ihrer Umgebung aktiviert haben, wartet Elastic

Beanstalk die angegebene Pausenzeit zwischen dem Zurücksetzen der Änderungen auf einem Stapel von Instances, bevor die Änderungen auf dem nächsten Stapel zurückgesetzt werden. Wenn Sie vor Kurzem fortlaufende Updates aktiviert haben, die letzte erfolgreiche Aktualisierung Ihrer Einstellungen der Umgebungskonfiguration jedoch ohne fortlaufende Updates durchgeführt wurde, führt Elastic Beanstalk das Rollback auf allen Instances gleichzeitig durch.

Sie können Elastic Beanstalk nicht daran hindern, zur vorherigen Umgebungskonfiguration zurückzukehren, sobald mit dem Abbrechen des Updates begonnen wurde. Der Rollbackvorgang wird fortgesetzt, bis alle Instances der Umgebung wieder die vorherige Umgebungskonfiguration aufweisen oder bis im Rollback Fehler auftreten. Bei Bereitstellungen von Anwendungsversionen wird durch das Abbrechen der Bereitstellung diese nur gestoppt. Einige Instances haben die neue Anwendungsversion und andere führen weiterhin die vorhandene Anwendungsversion aus. Sie können die gleiche oder eine andere Anwendungsversion zu einem späteren Zeitpunkt bereitstellen.

Weitere Informationen zu fortlaufenden Aktualisierungen finden Sie unter [Fortlaufende Updates der Elastic Beanstalk-Umgebungskonfiguration \(p. 484\)](#). Weitere Informationen zu Bereitstellungen der Anwendungsversion in Stapeln finden Sie unter [Bereitstellungsrichtlinien und -einstellungen \(p. 474\)](#).

So brechen Sie ein Update ab

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite Environment actions (Umgebungsaktionen) und dann Abort current operation (Abbrechen des aktuellen Vorgangs) aus.

Elastic Beanstalk-Umgebungen wiederherstellen

Ihre AWS Elastic Beanstalk-Umgebung kann unbrauchbar werden, wenn Sie nicht die Elastic Beanstalk-Funktionalität verwenden, um die der Umgebung zugrundeliegenden AWS Ressourcen zu modifizieren oder zu terminieren. Wenn dies der Fall ist, können Sie die Umgebung erneut erstellen, um zu versuchen, sie wieder in einen betriebsbereiten Zustand zu versetzen. Durch die Neuerstellung einer Umgebung werden alle ihre Ressourcen beendet und durch neue Ressourcen mit derselben Konfiguration ersetzt.

Sie können beendete Umgebungen auch innerhalb von sechs Wochen (42 Tage) nach ihrer Beendigung erneut erstellen. Bei der Neuerstellung versucht Elastic Beanstalk, eine neue Umgebung mit dem gleichen Namen, der gleichen ID und Konfiguration zu erstellen.

Neuerstellung einer laufenden Umgebung

Sie können eine Umgebung über die Elastic Beanstalk-Konsole oder mit Hilfe der `RebuildEnvironment` API neu erstellen.

So erstellen Sie eine laufende Umgebung erneut (Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) aus. Wählen Sie dann Rebuild environment (Umgebung neu erstellen) aus.
4. Wählen Sie Rebuild (Neu erstellen).

Durch die Neuerstellung einer laufenden Umgebung werden neue Ressourcen erstellt, die die gleiche Konfiguration wie die alten Ressourcen haben. Die Ressourcen-IDs unterscheiden sich allerdings und Daten auf den alten Ressourcen werden nicht wiederhergestellt. Beispiel: Bei der Neuerstellung einer Umgebung mit einer Amazon RDS-Datenbank-Instance wird eine neue Datenbank mit der gleichen Konfiguration erstellt, es wird aber kein Snapshot auf die neue Datenbank angewendet.

Zur Neuerstellung einer laufenden Umgebung mit der Elastic Beanstalk-API verwenden Sie die [RebuildEnvironment](#)-Aktion mit der AWS-CLI oder dem AWS-SDK.

```
$ aws elasticbeanstalk rebuild-environment --environment-id e-vdnftxubwq
```

Neuerstellung einer beendeten Umgebung

Sie können eine Umgebung erneut erstellen oder eine beendete Umgebung wiederherstellen, indem Sie die Elastic Beanstalk-Konsole, die EB-CLI oder die [RebuildEnvironment](#)-API verwenden.

Note

Sofern Sie nicht Ihren eigenen benutzerdefinierten Domänennamen für Ihre beendete Umgebung verwenden, verwendet die Umgebung eine Subdomäne von elasticbeanstalk.com. Diese Unterdomänen werden innerhalb einer Elastic Beanstalk-Region gemeinsam genutzt. Daher können sie von jeder Umgebung verwendet werden, die von einem beliebigen Kunden in derselben Region erstellt wurden. Wenn Ihre Umgebung beendet wurde, kann eine andere Umgebung ihre Subdomäne verwenden. In diesem Fall würde die Neuerstellung fehlschlagen. Sie können dieses Problem vermeiden, indem Sie eine benutzerdefinierte Domäne verwenden. Details dazu finden Sie unter [Der Domänenname Ihrer Elastic Beanstalk-Umgebung \(p. 644\)](#).

Kürzlich beendete Umgebungen erscheinen im Überblick der Anwendung bis zu einer Stunde lang. Während dieser Zeit können Sie Ereignisse für die Umgebung in deren [Dashboard \(p. 426\)](#) anzeigen und die [Restore environment](#) (Umgebung wiederherstellen)-[Action \(Aktion\) \(p. 428\)](#) zur Neuerstellung verwenden.

Zur Neuerstellung einer Umgebung, die nicht mehr sichtbar ist, verwenden Sie die Option [Restore terminated environment](#) (Beendete Umgebung wiederherstellen) auf der Anwendungsseite.

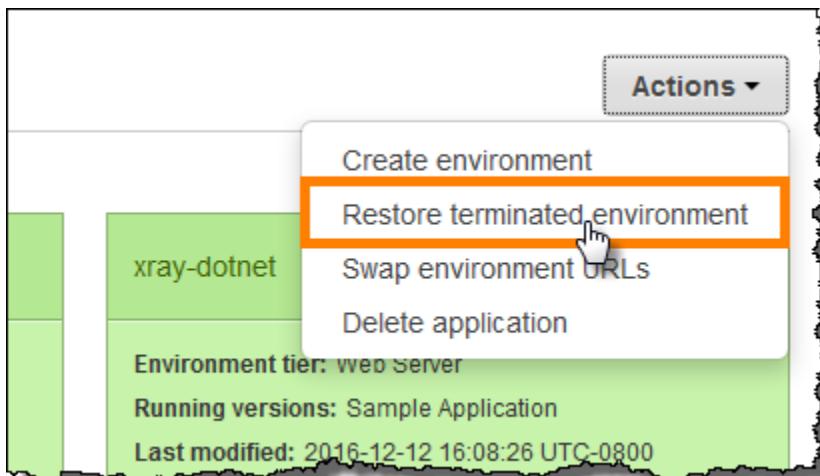
So erstellen Sie eine beendete Umgebung erneut (Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.

Note

Wenn Sie viele Anwendungen haben, verwenden Sie die Suchleiste, um die Anwendungsliste zu filtern.

3. Klicken Sie auf Actions und dann auf [Restore terminated environment](#) (Beendete Umgebung wiederherstellen).



4. Wählen Sie eine beendete Umgebung.
5. Wählen Sie Restore.

The screenshot shows the "Terminated environments" page. It displays a table with one row of data. The columns are "Name", "Date terminated", "Platform configuration", and "App...". The single entry is "beta-java" with a termination date of "2016-12-12 10:50:49 UTC-0800", a platform configuration of "64bit Amazon Linux 2016.03 v2.1.6 running Java 8", and an application name starting with "app-".

Bei der Neuerstellung versucht Elastic Beanstalk, eine neue Umgebung mit dem gleichen Namen, der gleichen ID und Konfiguration zu erstellen. Wenn eine Umgebung mit dem gleichen Namen oder der gleichen URL vorhanden ist, wenn Sie versuchen, eine Neuerstellung durchzuführen, schlägt diese fehl. Das Löschen der Anwendungsversion, die für die Umgebung bereitgestellt wurde, verursacht ebenfalls einen Fehler bei der Neuerstellung.

Wenn Sie die EB CLI für die Verwaltung Ihrer Umgebung verwenden, können Sie über den `eb restore`-Befehl eine beendete Umgebung erneut erstellen.

```
$ eb restore e-vdnftxubwq
```

Weitere Informationen finden Sie unter [eb restore \(p. 1085\)](#).

Zur Neuerstellung einer beendeten Umgebung mit der Elastic Beanstalk-API verwenden Sie die [RebuildEnvironment](#)-Aktion mit der AWS-CLI oder dem AWS-SDK.

```
$ aws elasticbeanstalk rebuild-environment --environment-id e-vdnftxubwq
```

Umgebungstypen

In AWS Elastic Beanstalk können Sie eine skalierbare Umgebung mit Lastausgleich oder eine Single-Instance-Umgebung erstellen. Der benötigte Umgebungstyp hängt von der bereitgestellten Anwendung ab. Beispielsweise können Sie eine Anwendung in einer Single-Instance-Umgebung entwickeln und testen, um Kosten zu sparen. Wenn die Anwendung für die Produktion bereit ist, können Sie ein Upgrade auf eine skalierbare Umgebung mit Lastausgleich ausführen.

Note

Die Worker-Umgebungsebene einer Webanwendung, die Hintergrundaufgaben ausführt, verfügt nicht über einen Load Balancer. Allerdings kann in einer Worker-Umgebung effektiv skaliert werden, sodass bei entsprechendem Workload weitere Instances für die Datenverarbeitung der Amazon SQS-Warteschlange zur Auto Scaling-Gruppe hinzufügt werden.

Skalierbare Umgebung mit Lastausgleich

Eine Umgebung mit Load Balancing- und Auto Scaling-Funktion nutzt die Elastic Load Balancing- und Amazon EC2 Auto Scaling-Services zur Bereitstellung der Amazon EC2-Instances, die von der bereitgestellten Anwendung benötigt werden. Amazon EC2 Auto Scaling startet automatisch weitere Instances, um die Datenverarbeitung der Anwendung auch bei einer erhöhten Workload zu gewährleisten. Sinkt der Workload wieder, werden einige Instances von Amazon EC2 Auto Scaling gestoppt. Die von Ihnen angegebene Mindestanzahl an Instances wird jedoch immer ausgeführt. Wenn die Anwendung Skalierbarkeit mit einer Option für die Ausführung in mehreren Availability Zones erfordert, sollten Sie eine skalierbare Umgebung mit Lastausgleich verwenden. Falls Sie nicht sicher sind, welcher Umgebungstyp der richtige ist, können Sie einfach einen auswählen und diesen später bei Bedarf ändern.

Single-Instance-Umgebung

Eine Umgebung mit einer Instance enthält eine Amazon EC2-Instance mit einer Elastic IP-Adresse. Eine Single-Instance-Umgebung besitzt keinen Load Balancer. Dies kann helfen, die Kosten im Vergleich zu einer skalierbaren Umgebung mit Lastausgleich zu senken. Obwohl eine Single-Instance-Umgebung den Amazon EC2 Auto Scaling-Service verwendet, sind sämtliche Einstellungen für die Mindestanzahl der Instances, die maximale Anzahl an Instances sowie die gewünschte Kapazität auf den Wert 1 festgelegt. Folglich werden keine neuen Instances gestartet, mit denen die Datenverarbeitung der Anwendung auch bei einer erhöhten Workload gewährleistet werden könnte.

Verwenden Sie eine Umgebung mit einer Instance, wenn Sie davon ausgehen, dass die Produktionsanwendung mit geringem Workload arbeitet, oder wenn Sie eine Remote-Entwicklung ausführen. Falls Sie nicht sicher sind, welcher Umgebungstyp der richtige ist, können Sie einfach einen auswählen und diesen später ändern. Weitere Informationen finden Sie unter [Ändern des Umgebungstyps \(p. 514\)](#).

Ändern des Umgebungstyps

Sie können den Typ Ihrer Umgebung in eine Single-Instance-Umgebung oder eine skalierbare Umgebung mit Lastausgleich ändern, indem Sie die Konfiguration Ihrer Umgebung bearbeiten. In einigen Fällen ist es möglicherweise erforderlich, von einem Umgebungstyp zum anderen zu wechseln. Das kann beispielsweise der Fall sein, wenn Sie eine Anwendung in einer Umgebung mit einer Instance entwickelt und getestet haben, um Kosten zu sparen. Wenn die Anwendung für die Produktion bereit ist, können Sie den Umgebungstyp in eine skalierbare Umgebung mit Lastausgleich ändern, damit sie entsprechend den Anforderungen Ihrer Kunden skaliert werden kann.

So ändern Sie den Umgebungstyp

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.

2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Kategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Wählen Sie aus der Liste Environment Type (Umgebungstyp) den gewünschten Umgebungstyp aus.

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

Max

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.

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 find instances.

6. Wählen Sie Save (Speichern) aus.

Es kann mehrere Minuten dauern, bis die Umgebung aktualisiert ist und Elastic Beanstalk die AWS-Ressourcen bereitgestellt hat.

Falls die Umgebung in einer VPC ist, wählen Sie Subnetze aus, in denen Sie Elastic Load Balancing und Amazon EC2-Instances platzieren. Jede Availability Zone, in der die Anwendung ausgeführt wird, muss über beides verfügen. Details dazu finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon VPC \(p. 988\)](#).

AWS Elastic Beanstalk-Worker-Umgebungen

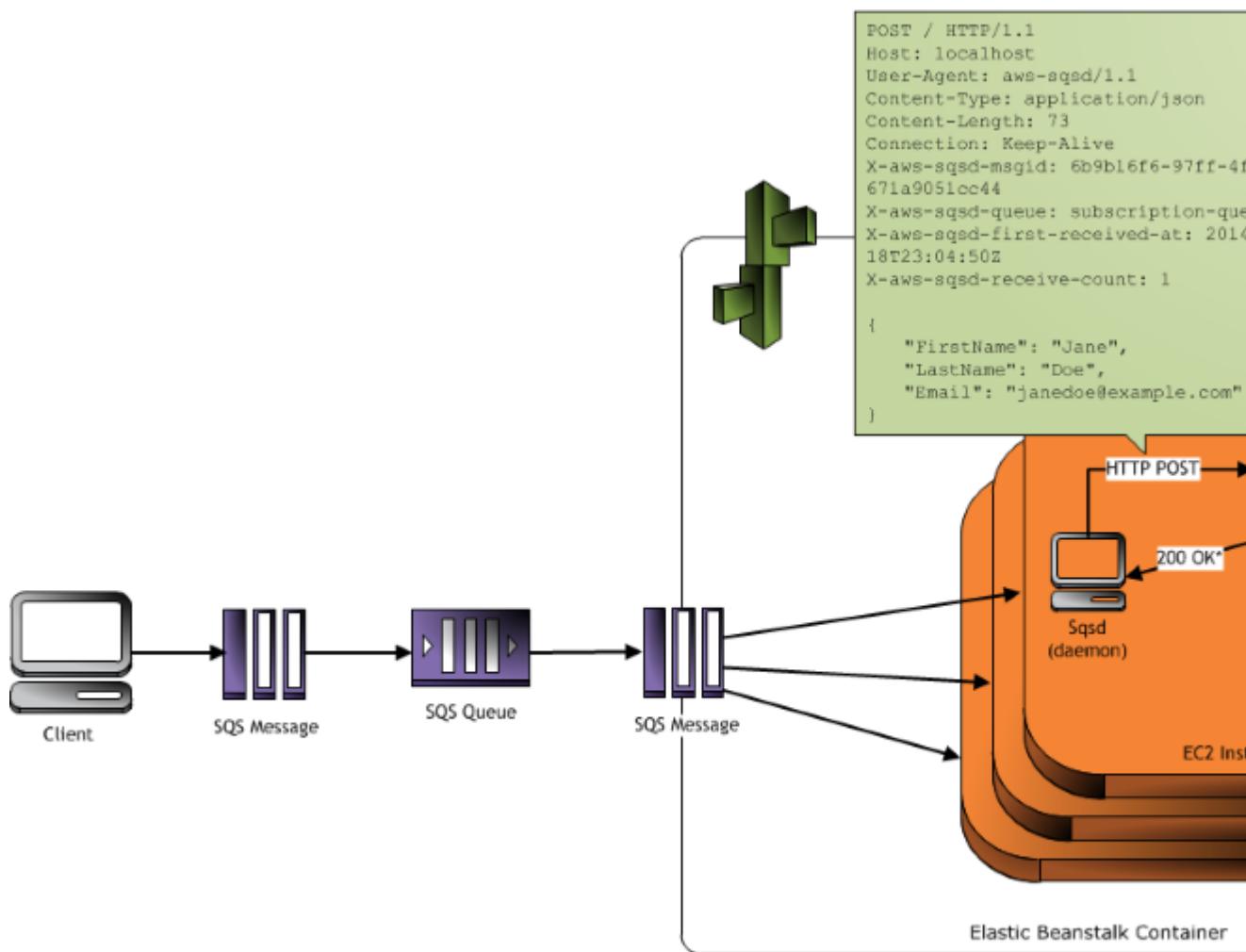
Wenn die AWS Elastic Beanstalk-Anwendung zeitaufwendige Operationen oder Workflows ausführt, können Sie diese Aufgaben in eine spezielle Worker-Umgebung auslagern. Das Frontend der Webanwendung von einem Prozess zu entkoppeln, der blockierende Vorgänge ausführt, ist ein gängiger Weg, um die Reaktionsfähigkeit der Anwendung auch bei Auslastung sicherzustellen.

Zu einer zeitaufwendigen Aufgabe zählen alle Vorgänge, die eine erhebliche Verlängerung der Anforderungsverarbeitung verursachen, also z. B. Bild- oder Videobearbeitung, Versenden von E-Mails oder Generieren eines ZIP-Archivs. Diese Vorgänge nehmen nur ein oder zwei Sekunden in Anspruch, jedoch ist eine Verzögerung von wenigen Sekunden ein hoher Wert für eine Webanforderung, die ansonsten in weniger als 500 ms verarbeitet wird.

Eine Möglichkeit besteht darin, einen Worker-Prozess lokal zu erstellen, erfolgreiche Antworten zurückzugeben und die Aufgabe asynchron zu verarbeiten. Das funktioniert, wenn die Instance alle gesendeten Aufgaben verarbeiten kann. Bei hoher Auslastung kann die Instance jedoch durch die Hintergrundaufgaben überlastet werden und daher nicht mehr auf Anforderungen mit höherer Priorität reagieren. Wenn einzelne Benutzer mehrere Aufgaben generieren können, entspricht die gestiegene Auslastung möglicherweise nicht einer höheren Benutzeranzahl, sodass keine effiziente Skalierung der Webserverebene mehr möglich ist.

Sie können die lokale Ausführung zeitaufwendiger Aufgaben verhindern, indem Sie diese – mithilfe des AWS SDKs für Ihre Programmiersprache – an eine Amazon Simple Queue Service (Amazon SQS)-Warteschlange senden, und den Prozess der Aufgabenverarbeitung auf separaten Instances ausführen. Die Worker-Instances verarbeiten Elemente aus der Warteschlange nur, wenn genügend Kapazität für deren Ausführung vorhanden ist. Auf diese Weise wird eine Überlastung verhindert.

Elastic Beanstalk Worker-Umgebungen vereinfachen diesen Prozess mithilfe der Amazon SQS-Warteschlangenverwaltung und der Ausführung eines [Daemon-Prozesses \(p. 518\)](#) auf jeder Instance, der Warteschlangenelemente liest. Wenn der Daemon ein Element aus der Warteschlange abruft, sendet er eine lokale HTTP POST-Anforderung auf Port 80 mit dem Textinhalt der Warteschlangennachricht an `http://localhost/`. Dann muss die Anwendung nur noch die zeitaufwendige Aufgabe als Reaktion auf die POST-Anforderung ausführen. Sie können den [Daemon konfigurieren \(p. 521\)](#), um die POST-Anforderung an einen anderen Pfad zu senden, einen anderen MIME-Typ als Anwendung/JSON zu verwenden, die Verbindung zu einer vorhandenen Warteschlange herzustellen oder Verbindungen (maximale gleichzeitige Anforderungen), Timeouts und Wiederholungen einzurichten.



* HTTP Response of 200 OK = delete the message
Any other HTTP Response = retry the message after the specified interval
No response = retry the message after the Inactivity timeout

Mit [regelmäßigen Aufgaben](#) (p. 519) können Sie den Worker-Daemon so konfigurieren, dass er Nachrichten auf Basis eines Cron-Plans in die Warteschlange stellt. Jede periodische Aufgabe kann die POST-Anforderung an einen anderen Pfad senden. Aktivieren Sie periodische Aufgaben, indem Sie eine YAML-Datei, die den Plan und den Pfad für die einzelnen Aufgaben definiert, in den Quellcode einbinden.

Note

Die [.NET-Plattform auf Windows Server](#) (p. 189) unterstützt keine Worker-Umgebungen.

Abschnitte

- [SQS-Daemon in der Worker-Umgebung](#) (p. 518)
- [Warteschlangen für unzustellbare Nachrichten](#) (p. 519)
- [Regelmäßige Aufgaben](#) (p. 519)
- [Amazon CloudWatch für die automatische Skalierung auf Worker-Umgebungs-Ebene verwenden](#) (p. 520)
- [Konfigurieren von Worker-Umgebungen](#) (p. 521)

SQS-Daemon in der Worker-Umgebung

In Worker-Umgebungen wird ein von Elastic Beanstalk bereitgestellter Daemon-Prozess ausgeführt. Dieser Daemon wird regelmäßig mit neuen Funktionen und Fehlerbehebungen aktualisiert. Für die neueste Daemon-Version aktualisieren Sie auf die neueste [Plattformversion \(p. 32\)](#).

Wenn die Anwendung in der Worker-Umgebung als Antwort 200 OK zurückgibt und damit den Empfang und die erfolgreiche Verarbeitung der Anforderung bestätigt, sendet der Daemon den Aufruf DeleteMessage an die Amazon SQS-Warteschlange, damit die Nachricht aus der Warteschlange gelöscht wird. Gibt die Anwendung eine andere Antwort als 200 OK zurück, stellt Elastic Beanstalk die Nachricht nach Ablauf des konfigurierten ErrorVisibilityTimeout-Zeitraums wieder in die Warteschlange. Falls es gar keine Antwort gibt, platziert Elastic Beanstalk die Nachricht nach Verstreichen des konfigurierten InactivityTimeout-Zeitraums wieder in die Warteschlange, damit sie anderweitig verarbeitet werden kann.

Note

Die Eigenschaften der Amazon SQS-Warteschlangen (Reihenfolge der Nachrichten, mindestens einmalige Zustellung und Nachrichten-Sampling) können die Entwicklung der Webanwendung in einer Worker-Umgebung beeinflussen. Weitere Informationen finden Sie unter [Eigenschaften verteilter Warteschlangen](#) im [Amazon Simple Queue Service-Entwicklerhandbuch](#).

Amazon SQS löscht automatisch Nachrichten, die sich länger als der konfigurierte RetentionPeriod-Zeitraum in der Warteschlange befinden.

Der Daemon legt folgende HTTP-Header fest.

HTTP-Header	
Name	Wert
User-Agent	aws-sqsd
	aws-sqsd/1.11
X-Aws-Sqsd-Msgid	SQS-Nachrichten-ID, mit der Nachrichtenfluten (ungewöhnlich hohe Anzahl neuer Nachrichten) erkannt werden.
X-Aws-Sqsd-Queue	Gibt den Namen der SQS-Warteschlange an.
X-Aws-Sqsd-First-Received-At	Gibt den Zeitpunkt, an dem die Meldung zum ersten Mal empfangen wurde, in koordinierter Weltzeit (UTC) im Format ISO 8601 an.
X-Aws-Sqsd-Receive-Count	Gibt die Anzahl der empfangenen SQS-Nachrichten an.
X-Aws-Sqsd-Attr- <i>message-attribute-name</i>	Benutzerdefinierte Attribute, die der gerade verarbeiteten Nachricht zugeordnet sind. Der tatsächliche Name des Nachrichtenattributs wird mit <i>message-attribute-name</i> angegeben. Alle Nachrichtenattribute im Zeichenfolgen- oder Ziffernformat werden zum Header hinzugefügt. Binary-Attribute werden verworfen und nicht in den Header aufgenommen.
Content-Type	Gibt die Mime-Typkonfiguration an, standardmäßig application/json.

Warteschlangen für unzustellbare Nachrichten

Elastic Beanstalk-WorkerUmgebungen unterstützen Amazon Simple Queue Service (Amazon SQS) Dead-Letter-Warteschlangen. Andere (Quell-) Warteschlangen können Nachrichten, die aus irgendeinem Grund nicht erfolgreich verarbeitet werden konnten, an die Warteschlange für unzustellbare Nachrichten senden. Eine solche Warteschlange für unzustellbare Nachrichten bietet den Vorteil, dass nicht erfolgreich verarbeitete Nachrichten ausgesondert und isoliert werden können. Anschließend können Sie alle Nachrichten analysieren, die an die Warteschlange für unzustellbare Nachrichten gesendet wurden, und ermitteln, warum die Verarbeitung nicht erfolgreich war.

Für eine Worker-Umgebung wird standardmäßig eine Warteschlange für unzustellbare Nachrichten aktiviert, wenn Sie bei der Erstellung der Worker-Umgebungsebene eine automatisch generierte Amazon SQS-Warteschlange angeben. Bei Auswahl einer vorhandenen SQS-Warteschlange für die Worker-Umgebung müssen Sie mithilfe von SQS eine Warteschlange für unzustellbare Nachrichten separat konfigurieren. Weitere Informationen zur Verwendung von SQS zum Konfigurieren einer Warteschlange für unzustellbare Nachrichten finden Sie unter [Using Amazon SQS Dead Letter Queues](#).

Warteschlangen für unzustellbare Nachrichten können nicht deaktiviert werden. Nicht zustellbare Nachrichten werden letztlich immer an eine Warteschlange für unzustellbare Nachrichten gesendet. Sie können diese Funktion jedoch effektiv deaktivieren, indem Sie die Option **MaxRetries** auf den maximal zulässigen Wert von 100 festlegen.

Wenn keine Warteschlange für unzustellbare Nachrichten für die Amazon SQS-Warteschlange Ihrer Arbeitsumgebung konfiguriert ist, werden Nachrichten in der Warteschlange von Amazon SQS aufbewahrt, bis der Aufbewahrungszeitraum abgelaufen ist. Weitere Informationen zum Konfigurieren des Aufbewahrungszeitraums finden Sie unter [the section called “Konfigurieren von Worker-Umgebungen” \(p. 521\)](#).

Note

Die Elastic Beanstalk **MaxRetries**-Option entspricht der SQS **MaxReceiveCount**-Option. Sofern in der Worker-Umgebung keine automatisch generierte SQS-Warteschlange verwendet wird, lässt sich die Warteschlange für unzustellbare Nachrichten effektiv mit der SQS-Option **MaxReceiveCount** deaktivieren. Weitere Informationen finden Sie unter [Using Amazon SQS Dead Letter Queues](#).

Weitere Informationen zum Lebenszyklus einer SQS-Nachricht finden Sie unter [Message Lifecycle](#).

Regelmäßige Aufgaben

Sie können periodische Aufgaben in einer `cron.yaml`-Datei im Quell-Bundle definieren, damit in regelmäßigen Abständen automatisch Aufträge zur Warteschlange der Worker-Umgebung hinzugefügt werden.

Die folgende `cron.yaml`-Datei erstellt beispielsweise zwei periodische Aufgaben. Die erste läuft alle 12 Stunden und die zweite läuft täglich um 23:00 Uhr UTC.

Example `cron.yaml`

```
version: 1
cron:
  - name: "backup-job"
    url: "/backup"
    schedule: "0 */12 * * *"
  - name: "audit"
    url: "/audit"
```

```
schedule: "0 23 * * *"
```

Die Eingabe für **name** muss für jede Aufgabe eindeutig sein. Die URL ist der Pfad, an den die POST-Anforderung gesendet wird, um den Auftrag auszulösen. Die Planung erfolgt über einen **CRON-Ausdruck**, mit dem der Ausführungszeitpunkt der Aufgabe festgelegt wird.

Wenn eine Aufgabe ausgeführt wird, sendet der Daemon eine Nachricht mit einem Header, in dem der auszuführende Auftrag beschrieben wird, an die SQS-Warteschlange der Umgebung. Jede Instance der Umgebung kann die Nachricht annehmen und den Auftrag verarbeiten.

Note

Bei Konfiguration der Worker-Umgebung mit einer vorhandenen SQS-Warteschlange müssen Sie eine [Amazon SQS-FIFO-Warteschlange](#) wählen, regelmäßige Aufgaben werden nicht unterstützt.

Elastic Beanstalk bestimmt anhand der Leader-Wahl, welche Instance der Worker-Umgebung die periodische Aufgabe in die Warteschlange stellt. Jede Instance versucht, durch Schreiben in eine Amazon DynamoDB-Tabelle zum Leader zu werden. Die erste erfolgreiche Instance fungiert als Leader und muss weiterhin in die Tabelle schreiben, um den Leader-Status zu behalten. Wenn der Leader ausfällt, übernimmt rasch eine andere Instance.

Bei regelmäßigen Aufgaben legt der Worker-Daemon die folgenden zusätzlichen Header fest.

HTTP-Header	
Name	Wert
X-Aws-Sqs-Taskname	Bei regelmäßigen Aufgaben ist dies der Name der auszuführenden Aufgabe.
X-Aws-Sqs-Scheduled-At	Gibt den Zeitpunkt an, für den die regelmäßige Aufgabe geplant war.
X-Aws-Sqs-Sender-Id	Gibt die AWS-Kontonummer des Absenders der Nachricht an.

Amazon CloudWatch für die automatische Skalierung auf Worker-Umgebungs-Ebene verwenden

Gemeinsam überwachen Amazon EC2 Auto Scaling und CloudWatch die CPU-Auslastung der laufenden Instanzen in der Worker-Umgebung. Mit der Konfiguration des Auto Scaling-Limits für die CPU-Kapazität geben Sie an, wie viele Instances von der Auto Scaling-Gruppe zur Verarbeitung des Nachrichtendurchsatzes in der Amazon SQS-Warteschlange ausgeführt werden. Jede EC2-Instance veröffentlicht ihre CPU-Auslastungsmetriken anCloudWatch. Mit der Amazon EC2 Auto Scaling-Funktion wird die durchschnittliche CPU-Auslastung aller Instances der Worker-Umgebung vonCloudWatch abgerufen. Sie konfigurieren den oberen und den unteren Grenzwert sowie die Anzahl der Instances, die entsprechend der CPU-Kapazität hinzugefügt oder beendet werden sollen. Wenn von der Amazon EC2 Auto Scaling-Funktion erkannt wird, dass der angegebene obere Grenzwert für die CPU-Kapazität erreicht ist, erstellt Elastic Beanstalk neue Instances in der Worker-Umgebung. Sinkt die CPU-Auslastung wieder unter den Grenzwert, werden die Instances gelöscht.

Note

Sofern beim Beenden einer Instance noch nicht verarbeitete Nachrichten vorhanden sind, werden diese an die Warteschlange zurückgegeben und können von einem anderen Daemon auf einer ausgeführten Instance verarbeitet werden.

Sie können bei Bedarf auch andere CloudWatch-Alarme mithilfe der AWS-Managementkonsole, der CLI oder der Optionsdatei einrichten. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon CloudWatch \(p. 879\)](#) und [Erstellen einer Auto Scaling-Gruppe mit Richtlinien für die schrittweise Skalierung](#).

Konfigurieren von Worker-Umgebungen

Die Verwaltung der Worker-Umgebungskonfiguration erfolgt über die Bearbeitung der Kategorie Worker auf der Seite Configuration (Konfiguration) in der [Environment Management Console \(p. 426\)](#).

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 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 the environment.

Queue

Worker queue

Autogenerated 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

application/json



Change the MIME type of the POST requests that the worker daemon sends to your application.

HTTP connections

50

Maximum number of concurrent connections to the application.

Visibility timeout

300

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

10

522

Maximum number of retries after which the message is discarded.

Connection timeout

Note

Sie können die URL-Pfad für die Veröffentlichung von Worker-Warteschlangen-Nachrichten konfigurieren, nicht aber den IP-Port. Elastic Beanstalk sendet Worker-Warteschlangennachrichten immer über Port 80. Die Anwendung der Worker-Umgebung oder ihr Proxy müssen Port 80 überwachen.

So konfigurieren Sie den Worker-Daemon

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Worker die Option Edit (Bearbeiten).

Die Konfigurationsseite Modify worker (Worker ändern) verfügt über die folgenden Optionen.

Im Abschnitt Queue (Warteschlange):

- Worker queue – Geben Sie die Amazon SQS-Warteschlange an, die vom Daemon gelesen wird. Sofern verfügbar, können Sie eine vorhandene Warteschlange auswählen. Wenn Sie Autogenerated queue (Automatisch generierte Warteschlange) auswählen, erstellt Elastic Beanstalk eine neue Amazon SQS-Warteschlange und eine entsprechende Worker-Warteschlangen-URL.

Note

Wenn Sie Autogenerated queue (Automatisch generierte Warteschlange) auswählen, erstellt Elastic Beanstalk eine Amazon SQS-[Standard](#)-Warteschlange. Wenn Sie eine vorhandene Warteschlange auswählen, können Sie entweder eine Standard- oder eine [FIFO](#)--Amazon SQS-Warteschlange angeben. Beachten Sie, dass bei Angabe einer FIFO-Warteschlange [regelmäßige Aufgaben](#) (p. 519) nicht unterstützt werden.

- Worker queue URL (URL für Worker-Warteschlange) – Wenn Sie eine vorhandene Worker queue (Worker-Warteschlange) auswählen, wird in dieser Einstellung die URL für diese Amazon SQS-Warteschlange angezeigt.

Im Abschnitt Messages (Nachrichten):

- HTTP path – Geben Sie den relativen Pfad zur Anwendung an, die Daten von der Amazon SQS-Warteschlange empfängt. Die Daten werden in den Nachrichtentext einer HTTP POST-Nachricht eingebunden. Der Standardwert ist /.
- MIME type (MIME-Typ) – Geben Sie den von der HTTP POST-Nachricht verwendeten MIME-Typ an. Der Standardwert ist application/json. Allerdings sind alle Werte gültig, da Sie einen eigenen MIME-Typ erstellen und dann angeben können.
- HTTP connections (HTTP-Verbindungen) – Geben Sie die maximale Anzahl gleichzeitiger Verbindungen an, die vom Daemon zu jeder Anwendung auf einer Amazon EC2-Instance hergestellt werden kann. Der Standardwert ist 50. Sie können 1 bis 100 eingeben.
- Visibility timeout – Geben Sie die Zeitspanne, die eine eingehende Nachricht aus der Amazon SQS-Warteschlange zur Verarbeitung gesperrt ist, in Sekunden an. Nach Ablauf des konfigurierten Zeitraums wird die Nachricht in der Warteschlange wieder sichtbar und kann von einem anderen Daemon gelesen werden. Wählen Sie einen Wert aus, der über der erwarteten Verarbeitungszeit der Nachricht durch die Anwendung liegt, bis max. **43200** Sekunden.

- Error visibility timeout – Geben Sie die zu verstreichende Zeitspanne, bevor Elastic Beanstalk eine Nachricht nach einem fehlgeschlagenen Verarbeitungsversuch mit einem expliziten Fehler an die Amazon SQS-Warteschlange zurückgibt, in Sekunden an. Sie können **0** bis **43200** Sekunden eingeben.

Führen Sie im Abschnitt Advanced options (Erweiterte Optionen) Folgendes durch:

- Max retries – Geben Sie die maximale Anzahl der Wiederholversuche für Elastic Beanstalk vor, um die Nachricht an die Amazon SQS-Warteschlange zu senden, bevor die Nachricht in die [Warteschlange für unzustellbare Nachrichten \(p. 519\)](#) verschoben wird. Der Standardwert ist **10**. Sie können **1** bis **100** eingeben.

Note

Die Option Max retries (Max. Wiederholungen) gilt nicht, wenn die Amazon SQS-Warteschlangen Ihrer Arbeitsumgebung nicht über eine konfigurierte Warteschlange für unzustellbare Nachrichten verfügt. In einem solchen Fall werden Nachrichten in der Warteschlange von Amazon SQS aufbewahrt und verarbeitet, bis der durch die Option Retention period (Aufbewahrungszeitraum) angegebene Zeitraum abgelaufen ist.

- Connection timeout (Zeitbeschränkung für Verbindung) – Geben Sie die Zeitspanne, die auf erfolgreiche Verbindungserstellungen zu einer Anwendung gewartet wird, in Sekunden an. Der Standardwert ist **5**. Sie können **1** bis **60** Sekunden eingeben.
- Inactivity timeout (Zeitbeschränkung für Inaktivität) – Geben Sie die Zeitspanne, die bei einer bestehenden Verbindung zu einer Anwendung auf eine Antwort gewartet wird, in Sekunden an. Der Standardwert ist **180**. Sie können **1** bis **36000** Sekunden eingeben.
- Retention period (Aufbewahrungszeitraum) – Geben Sie die Zeitspanne, die eine Nachricht gültig ist und aktiv verarbeitet wird, in Sekunden an. Der Standardwert ist **345600**. Sie können **60** bis **1209600** Sekunden eingeben.

Bei Verwendung einer vorhandenen Amazon SQS-Warteschlange kann zwischen den Einstellungen, die Sie bei der Worker-Umgebungserstellung konfiguriert haben, und den direkt von Ihnen in Amazon SQS konfigurierten Einstellungen ein Konflikt entstehen. Wenn beispielsweise der für eine Worker-Umgebung konfigurierte `RetentionPeriod`-Wert über dem in Amazon SQS festgelegten `MessageRetentionPeriod`-Wert ist, wird die Nachricht bei einer Überschreitung des `MessageRetentionPeriod`-Werts von Amazon SQS gelöscht.

Umgekehrt gilt dasselbe: Wenn der in den Worker-Umgebungseinstellungen konfigurierte `RetentionPeriod`-Wert niedriger ist als der in Amazon SQS definierte `MessageRetentionPeriod`-Wert, wird die Nachricht vom Daemon gelöscht, bevor Amazon SQS den Löschvorgang ausführt. Der für den Daemon in den Worker-Umgebungseinstellungen konfigurierte Wert setzt den in Amazon SQS eingestellten `VisibilityTimeout`-Wert für `VisibilityTimeout` außer Kraft. Stellen Sie sicher, dass Nachrichten angemessen gelöscht werden, indem Sie Ihre Elastic Beanstalk-Einstellungen mit Ihren Amazon SQS-Einstellungen vergleichen.

Verknüpfungen zwischen Elastic Beanstalk-Umgebungen erstellen

Mit zunehmender Anwendungsgröße und -komplexität kann es sinnvoll sein, sie in Komponenten mit unterschiedlichen Entwicklungs- und Betriebslebenszyklen zu untergliedern. Durch die Ausführung von kleineren Services, die über eine gut definierten Schnittstelle miteinander interagieren, können Teams unabhängig voneinander arbeiten, und die Bereitstellungen erfolgen mit einem geringeren Risiko. Mit AWS Elastic Beanstalk können Sie Ihre Umgebungen verknüpfen, um Informationen zwischen Komponenten zu teilen, die voneinander abhängig sind.

Note

Derzeit unterstützt Elastic Beanstalk Umgebungslinks für alle Plattformen (außer Multicontainer Docker).

Mithilfe von Umgebungslinks können Sie die Verbindungen zwischen den Umgebungen der Anwendungskomponenten als namentliche Referenzen angeben. Wenn Sie eine Umgebung mit einer Linkdefinition erstellen, legt Elastic Beanstalk eine Umgebungsvariable mit dem gleichen Namen des Links fest. Der Variablenwert ist der Endpunkt, über den Sie die Verbindung zur anderen Komponente herstellen; dabei kann es sich um eine Webserver- oder Worker-Umgebung handeln.

Angenommen, Ihre Anwendung besteht aus einem Frontend zur Erfassung von E-Mail-Adressen und einem Worker, der Begrüßungs-E-Mails an die vom Frontend gesammelten E-Mail-Adressen sendet. Nun können Sie in Ihrem Frontend einen Link zum Worker einrichten, damit das Frontend automatisch den Endpunkt (Warteschlangen-URL) des Workers erkennt.

Definieren Sie Links zu anderen Umgebungen in einem [Umgebungsmanifest \(p. 771\)](#). Dies ist eine Datei im YAML-Format mit dem Namen `env.yaml` im Stamm der Anwendungsquelle. In folgendem Manifest wird ein Link zu einer Umgebung mit dem Namen "worker" definiert:

`~/workspace/my-app/frontend/env.yaml`

```
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentLinks:
  "WORKERQUEUE": "worker"
```

Wenn Sie eine Umgebung mit einer Anwendungsversion erstellen, in der obiges Umgebungsmanifest enthalten ist, sucht Elastic Beanstalk nach einer Umgebung mit dem Namen `worker`, die zu dieser Anwendung gehört. Wenn diese Umgebung vorhanden ist, erstellt Elastic Beanstalk eine Umgebungseigenschaft namens `WORKERQUEUE`. Der Wert von `WORKERQUEUE` ist die URL der Amazon SQS-Warteschlange. Die Frontend-Anwendung kann diese Eigenschaft auf dieselbe Weise lesen wie eine Umgebungsvariable. Details dazu finden Sie unter [Umgebungsmanifest \(`env.yaml`\) \(p. 771\)](#).

Zum Verwenden von Umgebungslinks fügen Sie ein Umgebungsmanifest zu Ihrer Anwendungsquelle hinzu und laden diese mit der EB CLI, der AWS CLI oder einem SDK hoch. Wenn Sie die AWS CLI oder ein SDK nutzen, setzen Sie das Flag `process` beim Aufruf von `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"
```

Mit dieser Option wird Elastic Beanstalk angewiesen, das Umgebungsmanifest und die Konfigurationsdateien im Quell-Bundle zu validieren, wenn Sie die Anwendungsversion erstellen. In der EB CLI wird das Flag automatisch gesetzt, sobald sich ein Umgebungsmanifest im Projektverzeichnis befindet.

Erstellen Sie die Umgebungen ganz regulär mit einem beliebigen Client. Wenn Umgebungen beendet werden müssen, beenden Sie zuerst die Umgebung mit dem Link. Falls eine Umgebung von einer anderen Umgebung verlinkt ist, verhindert Elastic Beanstalk, dass die verlinkte Umgebung beendet wird. Diese Schutzmaßnahme setzen Sie mit dem Flag `ForceTerminate` außer Kraft. In der AWS CLI ist dieser Parameter als `--force-terminate` verfügbar:

```
$ aws elasticbeanstalk terminate-environment --force-terminate --environment-name worker
```

Konfigurieren von Elastic Beanstalk-Umgebungen

AWS Elastic Beanstalk stellt eine breite Palette an Optionen zum Anpassen der Ressourcen in Ihrer Umgebung, des Verhaltens von AWS Elastic Beanstalk; und der Platormeinstellungen bereit. Wenn Sie eine Webserver-Umgebung erstellen, erstellt Elastic Beanstalk mehrere Ressourcen, die für den Betrieb Ihrer Anwendung erforderlich sind.

- EC2-Instance – Eine virtuelle Maschine von Amazon Elastic Compute Cloud (Amazon EC2) zum Ausführen von Webanwendungen auf der von Ihnen ausgewählten Plattformen.

Jede Plattform führt eine spezifische Gruppe von Software, Konfigurationsdateien und Skripts zur Unterstützung einer bestimmten Sprachversion, eines Frameworks, Webcontainers oder einer Kombination daraus aus. Die meisten Plattformen verwenden entweder Apache oder NGINX als Reverse-Proxy, der vor der Webanwendung platziert ist, Anforderungen weiterleitet, statische Komponenten bedient und Zugriffs- und Fehlerprotokolle generiert.

- Instance-Sicherheitsgruppe – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Mit dieser Ressource kann HTTP-Datenverkehr vom Load Balancer die EC2-Instance mit Ihrer Webanwendung erreichen. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Load Balancer – Ein Elastic Load Balancing-Load Balancer ist für die Verteilung von Anforderungen an die Instances, auf denen Ihre Anwendung ausgeführt wird, konfiguriert. Mit einem Load Balancer müssen Ihre Instances nicht direkt für das Internet zugänglich gemacht werden.
- Load Balancer-Sicherheitsgruppe – Eine Amazon EC2-Sicherheitsgruppe, die so konfiguriert ist, dass eingehender Datenverkehr auf Port 80 erlaubt ist. Diese Ressource ermöglicht, dass HTTP-Datenverkehr aus dem Internet den Load Balancer erreicht. Standardmäßig ist Datenverkehr auf anderen Ports nicht zulässig.
- Auto Scaling-Gruppe – Eine Auto Scaling-Gruppe, die so konfiguriert ist, dass sie eine Instance ersetzt, wenn diese beendet wird oder nicht verfügbar ist.
- Amazon S3-Bucket – Ein Speicherort für Ihren Quellcode, Protokolle und andere Artefakte, die bei der Verwendung von Elastic Beanstalk erstellt werden.
- Amazon CloudWatch-Alarme – Zwei CloudWatch-Alarme, die die Verarbeitungslast auf den Instances in Ihrer Umgebung überwachen und ausgelöst werden, wenn die Last zu hoch oder zu niedrig ist. Beim Auslösen eines Alarms wird Ihre Auto Scaling-Gruppe nach oben oder unten skaliert.
- AWS CloudFormation Stack – Elastic Beanstalk verwendet AWS CloudFormation, um die Ressourcen in Ihrer Umgebung zu starten und Konfigurationsänderungen zu verbreiten. Die Ressourcen werden in einer Vorlage definiert, die Sie in der [AWS CloudFormation-Konsole](#) anzeigen können.
- Domänenname – Ein Domänenname, der auf Ihre Webanwendung in der Form `subdomain.region.elasticbeanstalk.com` weiterleitet.

In diesem Thema geht es um die Ressourcen-Konfigurationsoptionen in der Elastic Beanstalk-Konsole. In den folgenden Themen wird erläutert, wie Sie Ihre Umgebung in der Konsole konfigurieren. Außerdem werden die zugrunde liegenden Namespaces beschrieben, die den Konsoleoptionen für Konfigurationsdateien oder API-Konfigurationsoptionen entsprechen. Weitere Informationen zu erweiterten Konfigurationsmethoden finden Sie unter [Konfigurieren von Umgebungen \(Erweitert\)](#) (p. 646).

Themen

- Umgebungskonfiguration mit der [Elastic Beanstalk-Konsole](#) (p. 527)
- Die Amazon EC2-Instances Ihrer Elastic Beanstalk-Umgebung (p. 531)
- Auto Scaling-Gruppe für Ihre Elastic Beanstalk-Umgebung (p. 540)
- Load Balancer Ihrer AWS Elastic Beanstalk-Umgebung (p. 554)
- Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung (p. 609)
- Sicherheit Ihrer AWS Elastic Beanstalk-Umgebung (p. 615)
- Ressourcen in Ihren Elastic Beanstalk-Umgebungen markieren (p. 617)
- Umgebungseigenschaften und andere Softwareeinstellungen (p. 621)
- Elastic Beanstalk-Umgebungsbenachrichtigungen mit Amazon SNS (p. 633)
- Amazon Virtual Private Cloud (Amazon VPC) mit Elastic Beanstalk konfigurieren (p. 637)
- Der Domänenname Ihrer Elastic Beanstalk-Umgebung (p. 644)

Umgebungskonfiguration mit der Elastic Beanstalk-Konsole

Über die Elastic Beanstalk-Konsole können Sie viele [Konfigurationsoptionen](#) (p. 646) einer Umgebung und ihrer Ressourcen anzeigen und ändern. Sie können das Verhalten Ihrer Umgebung während Bereitstellungen anpassen, zusätzliche Funktionen aktivieren und den Instance-Typ und andere Einstellungen ändern, die Sie während der Umgebungserstellung ausgewählt haben.

So zeigen Sie eine Zusammenfassung Ihrer Umgebungskonfiguration an

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.

Seite “Konfigurationsübersicht“

Auf der Seite Configuration overview (Konfigurationsübersicht) wird eine Gruppe von Konfigurationskategorien angezeigt. Jede Konfigurationskategorie fasst den aktuellen Status einer Gruppe verwandter Optionen zusammen.

Sie können zwei Ansichten für diese Seite auswählen. Aktivieren Sie Table View (Tabellenansicht), um eine Liste mit nach Kategorie gruppierten Optionen anzuzeigen.

Configuration overview

Cancel Review changes

Search for an option name or value

Category	Options
Software	Rotate logs: disabled Node command: Gzip compression: true Proxy server: nginx Node.js version: 10.15.3 Log streaming: disabled X-Ray daemon: disabled
Instances	AMI ID: ami-0159e246d939e78a0 Instance type: t2.micro Monitoring interval: 5 minute IOPS: container default Size: container default Root volume type: container default EC2 security groups: load balancing, auto scaling
Capacity	Max: 4 Min: 1 Scale down increment: -1 Lower threshold: 2000000 Period: 5 Scale up increment: 1 Upper threshold: 6000000 Availability Zones: Any Breach duration: 5

Sie können eine Option nach ihrem Namen oder ihrem Wert suchen, indem Sie Suchbegriffe in ein Suchfeld eingeben. Während der Eingabe wird die Liste auf die Optionen verkürzt, die mit Ihren Suchbegriffen übereinstimmen.

Configuration overview

Cancel Review changes

Q log X

Category	Options
Software	Lifecycle: Keep logs after terminating environment Rotate logs: disabled Log streaming: disabled
Monitoring	Log group: logStream:group=/aws/elasticbeanstalk/Node-env/environment-health.log target="_blank">/aws/elasticbeanstalk/Node-env/environment-health.log

Aktivieren Sie Table View (Tabellenansicht), um die einzelnen Kategorien in einem separaten Frame (Konfigurationskategorie) anzuzeigen.

Configuration overview

Cancel Review changes

Software Node.js version: 10.15.3 AWS X-Ray: disabled Rotate logs: disabled (default) Log streaming: disabled (default) Environment properties: 0 Modify	Instances EC2 instance type: t2.micro EC2 Image ID: ami-0159e246d939e78a0 Monitoring interval: 5 minute Root volume type: container default Root volume size (GB): container default Root volume IOPS: container default Security groups: sg-07a4598b8cfe0f9a0 Modify	Capacity Environment type: load balancer Availability Zones: Any Instances: 1–4 Modify
Load balancer Load balancer type: application Listeners: 1 Processes: 1 Edit	Rolling updates and deployments Deployment policy: All at once Rolling updates: disabled Health check: enabled Edit	Security Service role: aws-elasticbeanstalk-environment Virtual machine key pair: Virtual machine instance: Edit

Wählen Sie in einer Konfigurationskategorie Edit (Bearbeiten), um eine zugehörige Konfigurationsseite aufzurufen, auf der die vollständigen Optionswerte angezeigt werden und Änderungen vorgenommen werden können. Nach dem Anzeigen und Ändern von Optionen können Sie eine der folgenden Aktionen auswählen:

- Cancel (Abbrechen) – Gehen Sie zurück zum Dashboard Ihrer Umgebung, ohne die Konfigurationsänderungen zu übernehmen. Wenn Sie Cancel (Abbrechen) auswählen, gehen ausstehende Änderungen, die Sie an einer Konfigurationskategorie vorgenommen haben, in der Konsole verloren.

Sie können Ihre Konfigurationsänderungen auch abbrechen, indem Sie eine andere Konsolenseite auswählen, z. B. Dashboard oder Logs (Protokolle). Dabei werden Sie im Falle ausstehender Konfigurationsänderungen von der Konsole aufgefordert, zu bestätigen, dass Sie mit dem Verlust der Änderungen einverstanden sind.

- Review changes (Änderungen überprüfen) – Rufen Sie eine Zusammenfassung aller ausstehenden Änderungen ab, die Sie an einer Konfigurationskategorie vorgenommen haben. Details dazu finden Sie unter [Seite "Änderungen überprüfen" \(p. 530\)](#).
- Apply configuration (Konfiguration übernehmen) – Übernehmen Sie die Änderungen, die Sie an einer Konfigurationskategorie in Ihrer Umgebung vorgenommen haben. In einigen Fällen werden Sie dazu aufgefordert, zu bestätigen, dass Sie mit den Konsequenzen Ihrer Konfigurationsentscheidungen einverstanden sind.

Seite "Änderungen überprüfen"

Auf der Seite Review changes (Änderungen überprüfen) wird eine Tabelle mit allen ausstehenden Optionsänderungen angezeigt, die Sie an einer Konfigurationskategorie vorgenommen und noch nicht in Ihrer Umgebung übernommen haben. Wenn Sie Optionen entfernt haben (zum Beispiel eine benutzerdefinierte Umgebungseigenschaft), wird eine zweite Tabelle mit den entfernten Optionen angezeigt.

In beiden Tabellen werden die jeweiligen Optionen in Kombination aus Namespace und Option Name (Optionsname) aufgelistet, anhand dessen diese in Elastic Beanstalk identifiziert werden. Details dazu finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Sie können beispielsweise folgende Konfigurationsänderungen vornehmen:

- In der Kategorie Capacity (Kapazität): Änderung von Instances (Min) von 1 auf 2 und Instances (Max) von 2 auf 4. Diese Änderung entspricht auf der Liste Changed Options (Geänderte Optionen) zwei Änderungen im aws:autoscaling:asg-Namespace.
- In der Kategorie Software:
 - Aktivieren der Option Rotate logs (Protokolle rotieren). Diese Änderung entspricht auf der Liste Changed Options (Geänderte Optionen) einer Änderung im aws:elasticbeanstalk:hostmanager-Namespace.
 - Entfernen der MY_ENV_PROPERTY-Umgebungseigenschaft. Diese Änderung entspricht auf der Liste Removed Options (Entfernte Optionen) einem Einzeleintrag für den aws:elasticbeanstalk:application:environment-Namespace.
- In der Kategorie Managed updates (Verwaltete Aktualisierungen): Aktivieren der Option Managed updates (Verwaltete Aktualisierungen). Diese einzelne Konfigurationsänderung entspricht auf der Liste Changed Options (Geänderte Optionen) drei Optionsänderungen in zwei Namespaces (die letzten drei Elemente der Liste).

Im folgenden Bild werden Ihre Konfigurationsänderungen auf der Seite Review changes (Änderungen überprüfen) aufgelistet.

Review Changes

Changed Options

Namespace ▾	Option Name	Old Value	New Value
aws:autoscaling:asg	MaxSize	2	4
aws:autoscaling:asg	MinSize	1	2
aws:elasticbeanstalk:hostmanager	LogPublicationControl	false	true
aws:elasticbeanstalk:managedactions	ManagedActionsEnabled	false	true
aws:elasticbeanstalk:managedactions	PreferredStartTime		WED:12:00
aws:elasticbeanstalk:managedactions :platformupdate	UpdateLevel		minor

Removed Options

Namespace	Option Name
aws:elasticbeanstalk:application:environment	MY_ENV_PROPERTY

Nach dem Überprüfen Ihrer Änderungen können Sie eine der folgenden Aktionen auswählen:

- Continue (Weiter) – Gehen Sie zur Seite Configuration overview (Konfigurationsübersicht) zurück. Anschließend können Sie weitere Änderungen vornehmen oder ausstehende Änderungen übernehmen.
- Apply configuration (Konfiguration übernehmen) – Übernehmen Sie die Änderungen, die Sie an einer Konfigurationskategorie in Ihrer Umgebung vorgenommen haben. In einigen Fällen werden Sie dazu aufgefordert, zu bestätigen, dass Sie mit den Konsequenzen Ihrer Konfigurationsentscheidungen einverstanden sind.

Die Amazon EC2-Instances Ihrer Elastic Beanstalk-Umgebung

Wenn Sie eine Webserver-Umgebung erstellen, erstellt AWS Elastic Beanstalk eine oder mehrere virtuelle Maschinen von Amazon Elastic Compute Cloud (Amazon EC2), die als Instancesbezeichnet werden.

Die Instances in Ihrer Umgebung sind so konfiguriert, dass Web-Apps auf der von Ihnen ausgewählten Plattform ausgeführt werden. Sie können Änderungen an verschiedenen Eigenschaften und Verhaltensweisen der Instances Ihrer Umgebung während der Umgebungserrstellung, nach der Erstellung in einer laufenden Umgebung oder als Teil des Quellcodes vornehmen, den Sie in der Umgebung bereitstellen. Details dazu finden Sie unter [the section called “Konfigurationsoptionen” \(p. 646\)](#).

Note

Die [Auto Scaling-Gruppe \(p. 540\)](#) in Ihrer Umgebung verwaltet die Amazon EC2-Instances, auf denen Ihre Anwendung ausgeführt wird. Wenn Sie Konfigurationsänderungen vornehmen, die auf dieser Seite beschrieben werden, ändert sich die Startkonfiguration (entweder eine Amazon EC2-Startvorlage oder eine Auto Scaling-Gruppenstart-Konfigurationsressource). Diese Änderung erfordert den [Austausch aller Instances \(p. 483\)](#) und löst ein [fortlaufende Aktualisierung \(p. 484\)](#) oder eine [unveränderliche Aktualisierung \(p. 488\)](#) aus, je nachdem, welche Version konfiguriert ist.

Während der Umgebungserstellung wählen Sie einen [Instance-Typ](#) aus, um die Hardware des Host-Computers zu bestimmen, der zur Ausführung Ihrer Instances verwendet wird. Elastic Beanstalk unterstützt neue Instance-Typen kurz nachdem Amazon EC2 sie eingeführt hat, in der Regel durch das nächste [Plattform-Update \(p. 491\)](#).

Elastic Beanstalk unterstützt mehrere [Kaufoptionen für Amazon EC2-Instances](#): On-Demand-Instances, Reserved Instances und Spot-Instances. Eine On-Demand-Instance ist eine nutzungsabhängige Ressource, bei der keine langfristige Verpflichtung erforderlich ist. Eine Reserved Instance ist ein vorab erworbener Abrechnungsrabatt, der automatisch auf übereinstimmende On-Demand-Instances in Ihrer Umgebung angewendet wird. Eine Spot-Instance ist eine ungenutzte Amazon EC2-Instance, die für weniger als den On-Demand-Preis erhältlich ist. Sie können Spot-Instances in Ihrer Umgebung aktivieren, indem Sie eine einzige Option festlegen. Sie können die Verwendung von Spot-Instances, einschließlich der Mischung aus On-Demand-Instances und Spot-Instances, mithilfe zusätzlicher Optionen konfigurieren. Weitere Informationen finden Sie unter [Auto Scaling-Gruppe \(p. 540\)](#).

Abschnitte

- [Konfigurieren der Amazon EC2-Instances Ihrer Umgebung \(p. 532\)](#)
- [Der Namespace aws: autoscaling: launchconfiguration \(p. 537\)](#)
- [Konfigurieren des Instance Metadata Service für die Instances Ihrer Umgebung \(p. 538\)](#)

Konfigurieren der Amazon EC2-Instances Ihrer Umgebung

Sie können die Konfiguration der Amazon EC2-Instance Ihrer Elastic Beanstalk-Umgebung in der Elastic Beanstalk-Konsole ändern.

So konfigurieren Sie Amazon EC2-Instances in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Instances die Option Edit (Bearbeiten). Nehmen Sie Änderungen an den Einstellungen in dieser Kategorie vor und wählen Sie dann Apply (Übernehmen). Informationen zum Einstellen von Beschreibungen finden Sie im Abschnitt [the section called "Instance-Kategorieeinstellungen" \(p. 533\)](#) auf dieser Seite.
5. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten). Nehmen Sie Änderungen an den Einstellungen in dieser Kategorie vor und wählen Sie dann Continue

(Weiter). Informationen zum Einstellen von Beschreibungen finden Sie im Abschnitt [the section called “Kapazitätskategorieeinstellungen” \(p. 536\)](#) auf dieser Seite.

Instance-Kategorieeinstellungen

Die folgenden Einstellungen in Bezug auf Amazon EC2-Instances sind in der Konfigurationskategorie Instances verfügbar.

Optionen

- [Überwachungsintervall \(p. 535\)](#)
- [Stamm-Volume \(Startgerät\) \(p. 535\)](#)
- [Instance-Metadatenservice \(p. 535\)](#)
- [Sicherheitsgruppen \(p. 535\)](#)

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.



IOPS

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"/> awseb-e-2mwbhpfc-stack-AWSEBLoadBalancerSecurityGroup-1DY3U98Q97M6V	sg-09d293ed9dde904a2	Getting
<input type="checkbox"/> awseb-e-2mwbhpfc-stack-AWSEBSecurityGroup-119YW6B19SFNK	sg-06e3ba42f5667c3a5	Getting
<input type="checkbox"/> awseb-e-cubmdjm6ga-stack-AWSEBLoadBalancerSecurityGroup-1GCJKWYQZ8MHY	sg-0ae9c16e639f7494e	Getting
<input checked="" type="checkbox"/> awseb-e-cubmdjm6ga-stack-AWSEBSecurityGroup-1HV9R4OGAAAXK	sg-0ca3299707080312a	Getting
<input type="checkbox"/> default	sg-1d9e5075	

Überwachungsintervall

Standardmäßig senden die Instances in Ihrer Umgebung [grundlegende Zustandsmetriken \(p. 821\)](#) in 5-Minuten-Intervallen an Amazon CloudWatch. Dabei entstehen für Sie keine zusätzlichen Kosten.

Für Detailberichte legen Sie den Wert von Monitoring interval (Überwachungsintervall) auf 1 Minute fest und erhöhen Sie so die Häufigkeit, mit der die Ressourcen der Umgebung die [grundlegenden Zustandsmetriken \(p. 823\)](#) an CloudWatch veröffentlichen. Für Metriken in 1-Minuten-Intervallen fallen Amazon CloudWatch-Servicegebühren an. Weitere Informationen finden Sie unter [Amazon CloudWatch](#).

Stamm-Volume (Startgerät)

Jede Instance der Umgebung ist mit einem Stamm-Volume konfiguriert. Das Stamm-Volume ist das Amazon EBS-Blockgerät, das der Instance zugeordnet ist und auf dem das Betriebssystem, Bibliotheken, Skripts und der Anwendungsquellcode gespeichert sind. Standardmäßig verwenden alle Plattformen allgemeine SSD-Blockgeräte für die Speicherung.

Sie können den Root volume type (Stamm-Volume-Typ) so ändern, dass Magnetspeicher oder bereitgestellte IOPS-SSD-Volumetypen verwendet werden und bei Bedarf die Volume-Größe erhöhen. Bei bereitgestellten IOPS-Volumes müssen Sie auch die Anzahl der bereitzustellenden IOPS angeben. Wählen Sie den Volume-Typ aus, der Ihren Leistungsanforderungen und Kostenvorstellungen entspricht.

Weitere Informationen finden Sie unter [Amazon EBS-Volume-Typen](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances und in den [Amazon EBS-Produktdetails](#).

Instance-Metadatenservice

Der Instance-Metadatenservice (IMDS) ist eine On-Instance-Komponente, die von Code auf der Instance verwendet wird, um sicher auf Instance-Metadaten zuzugreifen. Code kann auf Instance-Metadaten von einer ausgeführten Instance mit einer von zwei Methoden zugreifen: Instance Metadata Service Version 1 (IMDSv1) oder Instance Metadata Service Version 2 (IMDSv2). IMDSv2 ist sicherer. Deaktivieren Sie IMDSv1, um IMDSv2 zu erzwingen. Details dazu finden Sie unter [the section called “IMDS” \(p. 538\)](#).

Note

Der IMDS-Abschnitt auf dieser Konfigurationsseite wird nur für Plattformversionen angezeigt, die IMDSv2 unterstützen.

Sicherheitsgruppen

Mithilfe der Sicherheitsgruppen, die den Instances zugeordnet sind, wird der zulässige eingehende und ausgehende Datenverkehr für die Instances festgelegt. Elastic Beanstalk erstellt eine Sicherheitsgruppe, die den Datenverkehr vom Load Balancer über die Standardports für HTTP (80) und HTTPS (443) zulässt.

Sie können weitere Sicherheitsgruppen erstellen und spezifizieren, um den Datenverkehr über andere Ports oder von anderen Quellen zu ermöglichen. Beispielsweise können Sie eine Sicherheitsgruppe für den SSH-Zugriff erstellen, mit der eingehender Datenverkehr über Port 22 aus einem eingeschränkten IP-Adressbereich oder – für mehr Sicherheit – von einem Bastion-Host, auf den nur Sie zugreifen können, zulässig ist.

Note

Um den Datenverkehr zwischen Instances der Umgebung A und Instances der Umgebung B zu erlauben, können Sie eine Regel zur Sicherheitsgruppe hinzufügen, die Elastic Beanstalk an Umgebung B angefügt hat, und die Sicherheitsgruppe bestimmen, die Elastic Beanstalk an Umgebung A angefügt hat. Dies erlaubt den eingehenden und ausgehenden Datenverkehr für Instances der Umgebung A. Beachten Sie jedoch, dass dadurch eine Abhängigkeit zwischen den beiden Sicherheitsgruppen entsteht. Wenn Sie später versuchen, Umgebung A zu beenden,

wird Elastic Beanstalk die Sicherheitsgruppe der Umgebung nicht löschen können, da die Sicherheitsgruppe der Umgebung B von ihr abhängt.
Sicherer wäre es, eine separate Sicherheitsgruppe zu erstellen, sie der Umgebung A anzufügen und sie in einer Regel der Sicherheitsgruppe der Umgebung B anzugeben.

Weitere Informationen zu Amazon EC2-Sicherheitsgruppen finden Sie in den [Amazon EC2-Sicherheitsgruppen](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Kapazitätskategorieeinstellungen

Die folgenden Einstellungen in Bezug auf Amazon EC2-Instances sind in der Konfigurationskategorie Capacity (Kapazität) verfügbar.

Optionen

- [Instance-Typ \(p. 536\)](#)
- [AMI-ID \(p. 537\)](#)

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

Instance type: t2.micro

AMI ID: ami-0c0f37f83da1542ca

Instance-Typ

Mit der Einstellung Instance type (Instance-Typ) wird der Typ der Amazon EC2-Instance ermittelt, die für die Anwendungsausführung gestartet wurde. Wählen Sie eine Instance aus, die einerseits leistungsfähig genug ist, um die Anwendung auch bei voller Auslastung auszuführen, die andererseits aber keine ungenutzte Überkapazität darstellt. Für Entwicklungszwecke bietet die t2-Instance-Familie generell eine moderate Leistung sowie die Möglichkeit einer Leistungssteigerung für kurze Zeiträume.

Setzen Sie bei umfangreichen Anwendungen mit hoher Verfügbarkeit einen Instance-Pool ein, damit die Kapazität auch im Falle, dass eine Instance ausfällt, möglichst wenig beeinträchtigt wird. Beginnen Sie mit einem Instance-Typ, mit dem Sie fünf Instances bei moderater Auslastung während des normalen Betriebs ausführen können. Falls eine Instance ausfällt, können die verbleibenden Instances den restlichen Datenverkehr verarbeiten. Der Kapazitätspuffer bietet zudem genügend Zeit, um die Umgebung zur Bewältigung des steigenden Datenverkehr zu Spitzenzeiten entsprechend nach oben zu skalieren.

Weitere Informationen zu verfügbaren Amazon EC2 Instance-Familien und -Typen finden Sie unter [Instance-Typen](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Wenn Sie Spot-Instance-Anforderungen für Ihre Umgebung aktivieren, wird auf dieser Konfigurationsseite eine Liste von Instance types (Instance-Typen) anstelle einer einzelnen Einstellung angezeigt. Sie können einen oder mehrere Instance-Typen für Ihre Spot-Instance auswählen. Details dazu finden Sie unter [the section called "Spot-Instance-Unterstützung" \(p. 541\)](#).

AMI-ID

Das Amazon Machine Image (AMI) ist das Image der Amazon Linux- oder Windows Server-Maschine, das von Elastic Beanstalk zum Starten von Amazon EC2-Instances in der Umgebung verwendet wird. Elastic Beanstalk stellt Maschinen-Images mit den für die Anwendungsausführung benötigten Tools und Ressourcen bereit.

Elastic Beanstalk wählt basierend auf der Region, der Plattform und dem Instance-Typ eine Standard-AMI für Ihre Umgebung aus. Wenn Sie ein [benutzerdefiniertes AMI \(p. 773\)](#) erstellt haben, ersetzen Sie die Standard-AMI-ID durch Ihre eigene.

Der Namespace aws: autoscaling: launchconfiguration

Sie können die [Konfigurationsoptionen \(p. 646\)](#) im [aws:autoscaling:launchconfiguration \(p. 668\)](#)-Namespace zur Konfiguration Ihrer Umgebungs-Instances verwenden, einschließlich zusätzlicher Optionen, die in der Konsole nicht verfügbar sind.

Mit der folgenden Beispiel-Konfigurationsdatei (p. 722) werden die Grundoptionen in diesem Thema, die Option DisableIMDSv1, die in [IMDS \(p. 538\)](#) erläutert wird, die Optionen EC2KeyName und IamInstanceProfile, die in [Sicherheit \(p. 615\)](#) erläutert werden, sowie eine zusätzliche Option, BlockDeviceMappings, die nicht in der Konsole verfügbar ist, konfiguriert.

```
option_settings:
  aws:autoscaling:launchconfiguration:
    InstanceType: m1.small
    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"
```

Über BlockDeviceMappings können Sie weitere Blockgeräte für die Instances konfigurieren. Weitere Informationen finden Sie unter [Blockgerät-Zuweisung](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Note

Die Option InstanceType ist veraltet. Sie wird durch die neuere und leistungsfähigere Option InstanceTypes im [aws:ec2:instances \(p. 680\)](#)-Namespace ersetzt. Mit der neuen Option können Sie eine Liste einer oder mehrerer Instance-Typen für Ihre Umgebung angeben. Der erste Wert auf dieser List entspricht dem Wert der Option InstanceType, die in dem hier beschriebenen aws:autoscaling:launchconfiguration-Namespace enthalten ist. Als Methode zur Angabe von Instance-Typen wird die neue Option empfohlen. Wenn die neue Option angegeben wird, hat sie vor der alten Vorrang. Weitere Informationen finden Sie unter [the section called "Der Namespace aws:ec2:instances" \(p. 546\)](#).

Die EB CLI und Elastic Beanstalk-Konsole wenden die empfohlenen Werte für die vorhergehenden Optionen an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter [Empfohlene Werte \(p. 648\)](#).

Konfigurieren des Instance Metadata Service für die Instances Ihrer Umgebung

Instance-Metadaten sind Daten über eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance, mit denen Anwendungen die ausgeführte Instance konfigurieren oder verwalten können. Der Instance-Metadatenservice (IMDS) ist eine On-Instance-Komponente, die von Code auf der Instance verwendet wird, um sicher auf Instance-Metadaten zuzugreifen. Bei diesem Code kann es sich um Elastic Beanstalk-Plattformcode in Ihren Umgebungs-Instances, um das AWS-SDK, das Ihre Anwendung möglicherweise verwendet, und sogar um den eigenen Code Ihrer Anwendung handeln. Weitere Informationen finden Sie unter [Instance-Metadaten und Benutzerdaten](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Code kann auf Instance-Metadaten von einer ausgeführten Instance mit einer von zwei Methoden zugreifen: Instance Metadata Service Version 1 (IMDSv1) oder Instance Metadata Service Version 2 (IMDSv2). IMDSv2 verwendet sitzungsorientierte Anfragen und mildert verschiedene Arten von Sicherheitsschwachstellen, über die versucht werden kann, auf das IMDS zuzugreifen. Weitere Informationen zu diesen beiden Methoden finden Sie unter [Konfigurieren des Instance-Metadatendienstes](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Abschnitte

- [Plattformunterstützung für IMDS \(p. 538\)](#)
- [Auswahl der IMDS-Methoden \(p. 538\)](#)
- [Konfigurieren von IMDS mit der Elastic Beanstalk-Konsole \(p. 539\)](#)
- [Der Namespace aws: autoscaling: launchconfiguration \(p. 539\)](#)

Plattformunterstützung für IMDS

Ältere Versionen der Elastic Beanstalk-Plattform haben IMDSv1 unterstützt. Neuere Elastic Beanstalk-Plattformversionen (alle [Amazon Linux 2-Plattformversionen \(p. 503\)](#)) unterstützen sowohl IMDSv1 als auch IMDSv2. Sie können Ihre Umgebung so konfigurieren, dass beide Methoden unterstützt werden (Standardeinstellung) oder IMDSv1 deaktiviert wird.

Note

Die Deaktivierung von IMDSv1 erfordert die Verwendung von Amazon EC2-Startvorlagen. Wenn Sie diese Funktion während der Erstellung oder Aktualisierung der Umgebung konfigurieren, versucht Elastic Beanstalk, Ihre Umgebung so zu konfigurieren, dass Amazon EC2-Startvorlagen verwendet werden (sofern die Umgebung sie nicht bereits verwendet). In diesem Fall schlägt die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Ihre Benutzerrichtlinie nicht die erforderlichen Berechtigungen besitzt. Daher empfehlen wir, unsere Richtlinie für verwaltete Benutzer zu verwenden oder die erforderlichen Berechtigungen zu Ihren benutzerdefinierten Richtlinien hinzuzufügen. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [the section called “Erstellen einer benutzerdefinierten Benutzerrichtlinie” \(p. 929\)](#).

Auswahl der IMDS-Methoden

Berücksichtigen Sie bei der Entscheidung über die IMDS-Methoden, die von Ihrer Umgebung unterstützt werden sollen, folgende Fälle:

- AWS-SDK – Wenn Ihre Anwendung ein AWS-SDK für eine beliebige Sprache verwendet, stellen Sie sicher, dass Sie eine aktuelle Version des SDK verwenden. Die AWS-SDKs führen IMDS-Aufrufe durch und neuere SDK-Versionen verwenden IMDSv2, wenn möglich. Wenn Sie IMDSv1 deaktivieren und Ihre Anwendung eine alte SDK-Version verwendet, schlagen IMDS-Aufrufe möglicherweise fehl.

- Ihr Anwendungscode – Wenn Ihre Anwendung IMDS-Aufrufe durchführt, sollten Sie das AWS-SDK verwenden, um die Aufrufe durchzuführen, anstatt direkte HTTP-Anfragen durchzuführen. Auf diese Weise müssen Sie keine Codeänderungen vornehmen, um zwischen IMDS-Methoden zu wechseln. Das AWS-SDK verwendet nach Möglichkeit IMDSv2.
- Elastic Beanstalk-Plattformcode – Unser Code führt IMDS-Aufrufe über das AWS-SDK durch und verwendet daher IMDSv2 auf allen unterstützenden Plattformversionen. Wenn Ihr Code ein aktuelles AWS-SDK verwendet und alle IMDS-Aufrufe über das SDK durchführt, können Sie IMDSv1 sicher deaktivieren.

Konfigurieren von IMDS mit der Elastic Beanstalk-Konsole

Sie können die Konfiguration der Amazon EC2-Instance Ihrer Elastic Beanstalk-Umgebung in der Elastic Beanstalk-Konsole ändern.

So konfigurieren Sie IMDS auf Ihren Amazon EC2-Instances in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Instances die Option Edit (Bearbeiten).



5. Stellen Sie Disable IMDSv1 (IMDSv1 deaktivieren) ein, um IMDSv2 zu erzwingen. Löschen Sie Disable IMDSv1 (IMDSv1 deaktivieren), um IMDSv1 und IMDSv2 zu aktivieren.
6. Wählen Sie Apply (Anwenden) aus.

Der Namespace aws: autoscaling: launchconfiguration

Sie können eine [Konfigurationsoption \(p. 646\)](#) im [aws:autoscaling:launchconfiguration \(p. 668\)](#)-Namespace verwenden, um IMDS für die Instances Ihrer Umgebung zu konfigurieren.

Im folgenden Beispiel für die [Konfigurationsdatei \(p. 722\)](#) wird IMDSv1 mit der DisableIMDSv1-Option deaktiviert.

```
option_settings:  
  aws:autoscaling:launchconfiguration:  
    DisableIMDSv1: true
```

Auto Scaling-Gruppe für Ihre Elastic Beanstalk-Umgebung

Ihre AWS Elastic Beanstalk-Umgebung enthält eine Auto Scaling-Gruppe, die die [Amazon EC2-Instances \(p. 531\)](#) in Ihrer Umgebung verwaltet. In einer Umgebung mit nur einer Instance stellt die Auto Scaling Gruppe sicher, dass immer eine Instance ausgeführt wird. In einer Umgebung mit Lastausgleich konfigurieren Sie die Gruppe mit einem auszuführenden Instance-Bereich, und Auto Scaling fügt Instances je nach Auslastung hinzu bzw. entfernt diese.

Die Auto Scaling-Gruppe wendet auch die Startkonfiguration für die Instances in Ihrer Umgebung an. Sie können die [Startkonfiguration anpassen \(p. 531\)](#), um den Instance-Typ, das Schlüsselpaar, den Amazon Elastic Block Store (Amazon EBS)-Speicher und andere Einstellungen zu ändern, die nur beim Starten einer Instance konfiguriert werden können.

Die Auto Scaling-Gruppe verwendet zwei Amazon CloudWatch-Alarne, um Skalierungsvorgänge auszulösen. Mit den Standardauslösern wird die Skalierung durchgeführt, wenn der durchschnittliche ausgehende Netzwerkdatenverkehr einer Instance innerhalb von 5 Minuten 6 MiB über- oder 2 MiB unterschreitet. Um Auto Scaling effektiv nutzen zu können, müssen Sie [Auslöser konfigurieren \(p. 547\)](#), die für Ihre Anwendung, Ihren Instance-Typ und die Serviceanforderungen geeignet sind. Sie können die Skalierung anhand unterschiedlicher Statistiken wie Latenz, Festplatten-E/A-Vorgänge, CPU-Auslastung und Anforderungsanzahl ausführen.

Damit die Amazon EC2-Instances Ihrer Umgebung während erwarteter Spitzenlasten beim Datenverkehr optimal genutzt werden, sollten Sie Ihre Auto Scaling-Gruppe so konfigurieren, dass die Anzahl der Instances nach einem festen Plan angepasst wird (p. 550). Sie können Änderungen an der Konfiguration der Gruppe auf Tags- oder Wochenbasis planen oder einmalige Änderungen planen, um Marketing-Ereignisse abzufangen, durch die viel Datenverkehr auf Ihre Website ausgelöst wird.

Optional kann Elastic Beanstalk On-Demand- und [Spot \(p. 541\)](#)-Instances für Ihre Umgebung kombinieren.

Des Weiteren wird mit Auto Scaling auch der Zustand der einzelnen gestarteten Amazon EC2-Instances überwacht. Falls eine Instance unerwartet beendet wird, erkennt Auto Scaling dies und startet eine andere Instance als Ersatz. Weitere Informationen dazu, wie Sie die Gruppe so konfigurieren, dass die Zustandsprüfung des Load Balancers verwendet wird, finden Sie unter [Zustandsprüfungseinstellung für Auto Scaling \(p. 554\)](#).

Sie können Auto Scaling mit der [Elastic Beanstalk-Konsole \(p. 543\)](#), der [EB CLI \(p. 546\)](#) oder mit [Konfigurationsoptionen \(p. 546\)](#) für die jeweilige Umgebung konfigurieren.

Themen

- [Spot-Instance-Unterstützung \(p. 541\)](#)
- [Auto Scaling-Gruppenkonfiguration mit der Elastic Beanstalk-Konsole \(p. 543\)](#)
- [Auto Scaling-Gruppenkonfiguration mit der EB CLI \(p. 546\)](#)
- [Konfigurationsoptionen \(p. 546\)](#)
- [Auto Scaling-Auslöser \(p. 547\)](#)
- [Geplante Auto Scaling-Aktionen \(p. 550\)](#)
- [Zustandsprüfungseinstellung für Auto Scaling \(p. 554\)](#)

Spot-Instance-Unterstützung

Um Amazon EC2 [Spot-Instances](#) nutzen zu können, können Sie eine Spot-Option für Ihre Umgebung aktivieren. Die Auto Scaling-Gruppe Ihrer Umgebung kombiniert dann Amazon EC2-Kaufoptionen und verwaltet eine Mischung aus On-Demand- und Spot-Instances.

Important

Die Nachfrage nach Spot-Instances kann sich von einem Moment zum anderen erheblich unterscheiden und die Verfügbarkeit von Spot-Instances kann sich abhängig von der Verfügbarkeit ungenutzter Amazon EC2-Instances ebenfalls erheblich unterscheiden. Zudem besteht immer die Möglichkeit, dass Ihre Spot-Instance unterbrochen wird. Aus diesem Grund müssen Sie sicherstellen, dass Ihre Anwendung auf eine Spot-Instance-Unterbrechung vorbereitet ist. Weitere Informationen finden Sie unter Benachrichtigungen über [Spot-Instance-Unterbrechungen](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Elastic Beanstalk bietet verschiedene Konfigurationsoptionen zur Unterstützung der Spot-Funktion. Sie werden in den folgenden Abschnitten zum Konfigurieren Ihrer Auto Scaling-Gruppe erläutert. Zwei dieser Optionen verdienen besondere Aufmerksamkeit: `SpotFleetOnDemandBase` und `SpotFleetOnDemandAboveBasePercentage` (beide im `aws:ec2:instances`-Namespace). Überlegen Sie, wie sie sich auf die `MinSize`-Option (im `aws:autoscaling:asg`-Namespace) beziehen. Nur `MinSize` bestimmt die anfängliche Kapazität Ihrer Umgebungen – die Anzahl der Instances, die als Minimum ausgeführt werden sollten. `SpotFleetOnDemandBase` wirkt sich nicht auf die anfängliche Kapazität aus. Wenn die Spot-Funktion aktiviert ist, bestimmt diese Option nur, wie viele On-Demand-Instances bereitgestellt werden, bevor Spot-Instances berücksichtigt werden. Wenn `SpotFleetOnDemandBase` kleiner als `MinSize` ist, erhalten Sie immer noch exakt `MinSize` Instances als Anfangskapazität. Mindestens `SpotFleetOnDemandBase` davon müssen On-Demand-Instances sein. Wenn `SpotFleetOnDemandBase` größer als `MinSize` ist, erhalten Sie bei der Skalierung Ihrer Umgebung mindestens `SpotFleetOnDemandBase - MinSize` zusätzliche Instances von Typ On-Demand, damit der Anforderung `SpotFleetOnDemandBase` entsprochen wird.

In Produktionsumgebungen sind Spot-Instances in skalierbaren Umgebungen mit Lastausgleich besonders nützlich. Wir empfehlen nicht, Spot-Instances in Umgebungen mit einer einzigen Instance zu verwenden. Wenn keine Spot-Instances verfügbar sind, verlieren Sie möglicherweise die gesamte Kapazität (eine einzelne Instance) Ihrer Umgebung. Möglicherweise möchten Sie dennoch eine Spot-Instance in einer Umgebung mit einer einzelnen Instance für die Entwicklung oder das Testen verwenden. Stellen Sie in diesem Fall sicher, dass Sie sowohl `SpotFleetOnDemandBase` als auch `SpotFleetOnDemandAboveBasePercentage` auf Null setzen. Alle anderen Einstellungen führen zu einer On-Demand-Instance.

Hinweise

- Einige ältere AWS-Konten stellen möglicherweise Standard-Instance-Typen für Elastic Beanstalk bereit, die keine Spot-Instances unterstützen (z. B. `t1.micro`). Wenn Sie beim Aktivieren von Spot-Instance-Anforderungen die Fehlermeldung `None of the instance types you specified supports Spot` (Keiner der angegebenen Instance-Typen unterstützt Spot) sehen, müssen Sie Instance-Typen konfigurieren, die die Spot-Funktion unterstützen. Verwenden Sie zur Auswahl von Spot-Instance-Typen [Spot Instance Advisor](#).
- Das Aktivieren von Spot-Instance-Anforderungen erfordert die Verwendung von Amazon EC2-Startvorlagen. Wenn Sie diese Funktion während der Erstellung oder Aktualisierung der Umgebung konfigurieren, versucht Elastic Beanstalk, Ihre Umgebung so zu konfigurieren, dass Amazon EC2-Startvorlagen verwendet werden (sofern die Umgebung sie nicht bereits verwendet). In diesem Fall schlägt die Erstellung oder Aktualisierung der Umgebung möglicherweise fehl, wenn Ihre Benutzerrichtlinie nicht die erforderlichen Berechtigungen besitzt. Daher empfehlen wir, unsere Richtlinie für verwaltete Benutzer zu verwenden oder die erforderlichen Berechtigungen zu Ihren benutzerdefinierten Richtlinien hinzuzufügen. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter [the section called "Erstellen einer benutzerdefinierten Benutzerrichtlinie"](#) (p. 929).

Die folgenden Beispiele veranschaulichen verschiedene Szenarien zum Festlegen der verschiedenen Skalierungsoptionen. Alle Beispiele gehen von einer Umgebung mit Lastausgleich aus, in der Spot-Instance-Anforderungen aktiviert wurden.

Example 1: On-Demand- und Spot-Instances als Teil der Anfangskapazität

Optioneneinstellungen

Option	Namespace	Value (Wert)
MinSize	aws:autoscaling:asg	10
MaxSize	aws:autoscaling:asg	24
SpotFleetOnDemandBase	aws:ec2:instances	4
SpotFleetOnDemandAboveBasePercentage	aws:ec2:instances	50

In diesem Beispiel wird in der Umgebung mit zehn Instances begonnen, von denen sieben On-Demand-Instances (vier Basis-Instances und 50 % der sechs oberhalb der Basis) und drei Spot-Instances sind. Die Umgebung kann bis zu 24 Instances skalieren. Bei der Skalierung wird der Anteil an On-Demand-Instances in dem Flottenteil oberhalb der vier On-Demand-Basis-Instances bei 50 % gehalten, bis hin zu einem Maximum von insgesamt 24 Instances, von denen 14 On-Demand-Instances (vier On-Demand-Basis-Instances und 50 % der 20 oberhalb der Basis) und zehn Spot-Instances sind.

Example 2: Gesamte On-Demand-Anfangskapazität

Optioneneinstellungen

Option	Namespace	Value (Wert)
MinSize	aws:autoscaling:asg	4
MaxSize	aws:autoscaling:asg	24
SpotFleetOnDemandBase	aws:ec2:instances	4
SpotFleetOnDemandAboveBasePercentage	aws:ec2:instances	50

In diesem Beispiel wird in der Umgebung mit vier Instances begonnen, wovon alle On-Demand-Instances sind. Die Umgebung kann bis zu 24 Instances skalieren. Bei der Skalierung wird der Anteil an On-Demand-Instances in dem Flottenteil oberhalb der vier On-Demand-Basis-Instances bei 50 % gehalten, bis hin zu einem Maximum von insgesamt 24 Instances, von denen 14 On-Demand-Instances (vier On-Demand-Basis-Instances und 50 % der 20 oberhalb der Basis) und zehn Spot-Instances sind.

Example 3: Zusätzliche On-Demand-Basis über die anfängliche Kapazität hinaus

Optioneneinstellungen

Option	Namespace	Value (Wert)
MinSize	aws:autoscaling:asg	3
MaxSize	aws:autoscaling:asg	24
SpotFleetOnDemandBase	aws:ec2:instances	4

Option	Namespace	Value (Wert)
SpotFleetOnDemandAboveBasePercentage	aws:ec2:instances	50

In diesem Beispiel wird in der Umgebung mit drei Instances begonnen, wovon alle On-Demand-Instances sind. Die Umgebung kann bis zu 24 Instances skalieren. Die erste zusätzliche Instance oberhalb der ersten drei ist On-Demand, um die vier On-Demand-Basis-Instances zu vervollständigen. Bei der weiteren Skalierung wird der Anteil an On-Demand-Instances in dem Flottenteil oberhalb der vier On-Demand-Basis-Instances bei 50 % gehalten, bis hin zu einem Maximum von insgesamt 24 Instances, von denen 14 On-Demand-Instances (vier On-Demand-Basis-Instances und 50 % der 20 oberhalb der Basis) und zehn Spot-Instances sind.

Auto Scaling-Gruppenkonfiguration mit der Elastic Beanstalk-Konsole

Sie können die Funktionsweise von Auto Scaling konfigurieren, indem Sie die Capacity (Kapazität) auf der Seite Configuration (Konfiguration) der Umgebung in der [Elastic Beanstalk-Konsole \(p. 426\)](#) bearbeiten.

So konfigurieren Sie Ihre Auto Scaling-Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Konfigurieren Sie im Abschnitt Auto Scaling Group (Auto Scaling-Gruppe) die folgenden Einstellungen.
 - Environment type (Umgebungstyp) – Wählen Sie Load balanced (Lastverteilt) aus.
 - Min instances (Mindestanzahl Instances) – Die Mindestanzahl an EC2-Instances, die die Gruppe stets enthalten sollte. Die Gruppe beginnt mit der Mindestanzahl und fügt Instances hinzu, wenn eine Hochskalierung ausgelöst wird.
 - Max instances (Höchstanzahl Instances) – Die Höchstanzahl an EC2-Instances, die die Gruppe stets enthalten sollte.

Note

Wenn Sie mit fortlaufenden Updates arbeiten, stellen Sie sicher, dass die Höchstanzahl an Instances über der Einstellung [Minimum instances in service \(Mindestanzahl ausgeführter Instances\) \(p. 485\)](#) für fortlaufende Updates liegt.

- Fleet composition (Flottenzusammensetzung) – Wählen Sie Combined purchase options and instance types (Kombinierte Kaufoptionen und Instance-Typen) aus, um Spot-Instance-Anforderungen zu aktivieren.

Wählen Sie in diesem Beispiel On-Demand Instances (On-Demand-Instances) aus.

- Instance type (Instance-Typ) – Der Typ von Amazon EC2-Instance, der zur Ausführung Ihrer Anwendung gestartet wurde. Details dazu finden Sie unter [the section called “Instance-Typ” \(p. 536\)](#).

- AMI ID (AMI-ID) – Das Maschinenabbild, das von Elastic Beanstalk zum Starten von Amazon EC2-Instances in Ihrer Umgebung verwendet wird. Details dazu finden Sie unter [the section called “AMI-ID” \(p. 537\)](#).
- Availability Zones (Verfügbarkeitszonen) – Legen Sie die Anzahl an Availability Zones fest, auf die die Instances der Umgebung verteilt werden. Standardmäßig startet die Auto Scaling-Gruppe Instances gleichmäßig über alle nutzbaren Zonen verteilt. Um Instances auf weniger Zonen zu verteilen, legen Sie die Anzahl der zu verwendenden Zonen fest. Verwenden Sie für Produktionsumgebungen mindestens zwei Zonen, um sicherzustellen, dass Ihre Anwendung auch beim Ausfall einer Availability Zone weiterhin verfügbar ist.
- Placement (Platzierung) (optional) – Wählen Sie die zu verwendenden Availability Zones aus. Verwenden Sie diese Einstellung, wenn Ihre Instances auf Ressourcen in bestimmten Zonen zugreifen müssen oder wenn Sie zonspezifische [Reserved Instances \(Vorbehaltene Instances\)](#) gekauft haben. Wenn Sie bei Ruhephase: '720' Benutzerdefinierte Availability Zones: 'us-west-2a, us-west-2b' MaxSize: '4' die Anzahl der Zonen festlegen, müssen Sie mindestens so viele benutzerdefinierte Zonen auswählen.

Wenn Sie Ihre Umgebung in einer benutzerdefinierten VPC starten, können Sie diese Option nicht konfigurieren. In einer benutzerdefinierten VPC wählen Sie Availability Zones für die Subnetze aus, die Sie der Umgebung zuweisen.

- Scaling cooldown (Ruhephase für die Skalierung) – Die Wartezeit (in Sekunden) für das Starten oder Beenden von Instances nach einer Skalierung, bevor Auslöser weiter ausgewertet werden. Weitere Informationen finden Sie unter [Skalieren von Ruhephasen](#).

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.

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 find 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 %

Instance type

t2.micro

AMI ID

ami-0c0f37f83da1542ca

Availability Zones

Number of Availability Zones (AZs) to use.

Any

545

Placement

Specify Availability Zones (AZs) to use.

-- Choose Availability Zones (AZs) --

6. Wählen Sie Apply (Anwenden) aus.

Auto Scaling-Gruppenkonfiguration mit der EB CLI

Wenn Sie mit dem Befehl [eb create \(p. 1053\)](#) eine Umgebung erstellen, können Sie einige Optionen bezüglich der Auto Scaling-Gruppe Ihrer Umgebung angeben. Mit diesen Optionen können Sie die Kapazität der Umgebung steuern.

--single

Eine Option zum Erstellen der Umgebung mit einer einzelnen Amazon EC2-Instance und ohne Load Balancer. Ohne diese Option wird eine Umgebung mit Lastenausgleich erstellt.

--instance-types

Eine Liste von Amazon EC2-Instance-Typen, die in der Umgebung verwendet werden sollen.

--enable-spot, --spot-max-price

Optionen für das Aktivieren und Konfigurieren von Spot-Instance-Anforderungen.

Im folgenden Beispiel wird eine Umgebung erstellt und die Auto Scaling-Gruppe so konfiguriert, dass Spot-Instance-Anforderungen für die neue Umgebung aktiviert werden. Dabei sind drei Instance-Typen möglich.

```
$ eb create --enable-spot --instance-types "t2.micro,t3.micro,t3.small"
```

Konfigurationsoptionen

Elastic Beanstalk stellt Konfigurationsoptionen ([p. 646](#)) für die Auto Scaling-Einstellungen in zwei Namespaces bereit: [aws:autoscaling:asg \(p. 667\)](#) und [aws:ec2:instances](#).

Der Namespace aws:autoscaling:asg

Der [aws:autoscaling:asg \(p. 667\)](#)-Namespace bietet Optionen für die allgemeine Skalierung und Verfügbarkeit.

Im folgenden Beispiel einer [Konfigurationsdatei \(p. 722\)](#) wird die Auto Scaling-Gruppe so konfiguriert, dass sie zwei bis vier Instances, bestimmte Availability Zones und eine Ruhephase von 12 Minuten (720 Sekunden) verwendet.

```
option_settings:
  aws:autoscaling:asg:
    Availability Zones: Any
    Cooldown: '720'
    Custom Availability Zones: 'us-west-2a,us-west-2b'
    MaxSize: '4'
    MinSize: '2'
```

Der Namespace aws:ec2:instances

Der [aws:ec2:instances \(p. 680\)](#) Namespace bietet Optionen, die sich auf die Instances Ihrer Umgebung beziehen, einschließlich der Spot-Instance-Verwaltung. Ergänzt werden [aws:autoscaling:launchconfiguration \(p. 668\)](#) und [aws:autoscaling:asg \(p. 667\)](#).

Wenn Sie die Umgebungskonfiguration aktualisieren und einen oder mehrere Instance-Typen aus der InstanceTypes-Option entfernen, beendet Elastic Beanstalk alle Amazon EC2-Instances, die auf einem der entfernten Instance-Typen ausgeführt werden. Die Auto Scaling-Gruppe Ihrer Umgebung startet dann

nach Bedarf neue Instances, um die gewünschte Kapazität zu vervollständigen, unter Verwendung der aktuell angegebenen Instance-Typen.

Im folgenden Beispiel einer [Konfigurationsdatei \(p. 722\)](#) wird die Auto Scaling-Gruppe so konfiguriert, dass Spot-Instance-Anforderungen für Ihre Umgebung mit drei möglichen Instance-Typen, mit mindestens einer On-Demand-Instance als Baseline-Kapazität und mit stetigen 33 % an On-Demand-Instances für zusätzliche Kapazität aktiviert werden.

```
option_settings:  
  aws:ec2:instances:  
    EnableSpot: true  
    InstanceTypes: 't2.micro,t3.micro,t3.small'  
    SpotFleetOnDemandBase: '1'  
    SpotFleetOnDemandAboveBasePercentage: '33'
```

Verwenden Sie zur Auswahl von Spot-Instance-Typen [Spot Instance Advisor](#).

Auto Scaling-Auslöser

Die Auto Scaling-Gruppe in Ihrer Elastic Beanstalk-Umgebung verwendet zwei Amazon CloudWatch-Alarne, um Skalierungsvorgänge auszulösen. Mit den Standardauslösern wird die Skalierung durchgeführt, wenn der durchschnittliche ausgehende Netzwerksdatenverkehr von der jeweiligen Instance innerhalb von 5 Minuten 6 MB über- oder 2 MB unterschreitet. Um Amazon EC2 Auto Scaling effektiv nutzen zu können, müssen Sie Auslöser konfigurieren, die für Ihre Anwendung, Ihren Instance-Typ und die Serviceanforderungen geeignet sind. Sie können die Skalierung anhand unterschiedlicher Statistiken wie Latenz, Festplatten-E/A-Vorgänge, CPU-Auslastung und Anforderungsanzahl ausführen.

Weitere Informationen zu CloudWatch Metriken und Alarne finden Sie unter [Amazon CloudWatch-Konzepte](#) im Amazon CloudWatch-Benutzerhandbuch.

Konfigurieren von Auto Scaling-Auslöser

Sie können die Auslöser konfigurieren, über die die Anzahl der Instances in der Auto Scaling-Gruppe Ihrer Umgebung in der Elastic Beanstalk-Konsole angepasst werden.

So konfigurieren Sie Auslöser in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Konfigurieren Sie im Abschnitt Scaling triggers (Skalierungsauslöser) die folgenden Einstellungen:
 - Metric (Metrik) – Metrik für Ihren Auto Scaling-Auslöser.
 - Statistic (Statistik) – Vom Auslöser zu verwendende Statistikberechnung, z. B. Average.
 - Unit (Einheit) – Einheit für die Metrik des Auslösers, z. B. Bytes.
 - Period (Intervall) – Gibt an, wie häufig die Metriken für den Auslöser von Amazon CloudWatch gemessen werden.
 - Breach duration (Verletzungsdauer) – Zeitdauer (in Minuten), die eine Metrik außerhalb der Ober- und Untergrenzen liegen kann, bevor eine Skalierungsoperation ausgelöst wird.

- Upper threshold (Obergrenze) – Falls die Metrik diesen Wert der Verletzungsdauer überschreitet, wird eine Skalierungsoperation ausgelöst.
- Scale up increment (Skalierung aufwärts) – Die Anzahl von Amazon EC2-Instances, die bei einer Skalierung hinzugefügt werden.
- Lower threshold (Untergrenze) – Wenn die Metrik diesen Wert der Verletzungsdauer unterschreitet, wird eine Skalierungsoperation ausgelöst.
- Scale down increment (Skalierung abwärts) – Die Anzahl von Amazon EC2-Instances, die bei einer Skalierung entfernt werden.

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. Wählen Sie Apply (Anwenden) aus.

Der Namespace aws:autoscaling:trigger

Elastic Beanstalk stellt Konfigurationsoptionen ([p. 646](#)) für die Auto Scaling-Einstellungen im Namespace `aws:autoscaling:trigger` ([p. 675](#)) bereit. Die Einstellungen in diesem Namespace sind nach Ressourcen gruppiert, auf die sie angewendet werden.

```
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
```

Geplante Auto Scaling-Aktionen

Damit die Amazon EC2-Instances Ihrer Umgebung während erwarteter Spitzenlasten beim Datenverkehr optimal genutzt werden, sollten Sie Ihre Amazon EC2 Auto Scaling-Gruppe so konfigurieren, dass die Anzahl der Instances nach einem festen Plan angepasst wird. Sie können Ihre Umgebung mit einer wiederkehrenden Aktion so konfigurieren, dass sie jeden Morgen hochskaliert und nachts bei geringem Datenverkehr herunterskaliert wird. Wenn Sie beispielsweise während eines Marketing-Ereignisses für einen begrenzten Zeitraum ein erhöhtes Datenvolumen erwarten, können Sie ein einmaliges Ereignis planen, um die Umgebung zu Beginn des Marketing-Ereignisses hochzuskalieren und an dessen Ende wieder herunterzuskalieren.

Es können bis zu 120 aktive geplante Aktionen pro Umgebung definiert werden. Elastic Beanstalk behält darüber hinaus bis zu 150 abgelaufene geplante Aktionen bei, die Sie zum Aktualisieren Ihrer Einstellungen wiederverwenden können.

Konfigurieren von geplanten Aktionen

Sie können geplante Aktionen für die Auto Scaling-Gruppe Ihrer Umgebung in der Elastic Beanstalk-Konsole erstellen.

So konfigurieren Sie geplante Aktionen in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Wählen Sie im Abschnitt Time-based Scaling (Zeitbasierte Skalierung) die Option Add scheduled action (Geplante Aktion hinzufügen) aus.

The screenshot shows the 'Time-based Scaling' configuration page. At the top, it says 'Current status: 1 instance(s) in service, Min: 1, Max: 2'. Below that, 'Time zone' is set to 'UTC'. A table below lists scheduled actions, with a note 'No scheduled actions'.

<input type="checkbox"/>	Name	Min	Max	Desired	Next occurrence (U)
No scheduled actions					

6. Geben Sie die folgenden Einstellungen für die geplante Aktion ein:

- Name – Geben Sie einen eindeutigen Namen mit bis zu 255 alphanumerischen Zeichen ohne Leerzeichen ein.
- Instances – Wählen Sie die minimale und maximale Instance-Anzahl für die Auto Scaling-Gruppe aus.
- Desired capacity (Gewünschte Kapazität) (optional) – Legen Sie die gewünschte Anfangskapazität für die Auto Scaling-Gruppe fest. Nachdem die geplante Aktion angewendet wurde, wird die gewünschte Kapazität basierend auf den Einstellungen angepasst.
- Occurrence (Häufigkeit) – Wählen Sie Recurring (Wiederkehrend) aus, um die Skalierungsaktion nach einem festen Zeitplan zu wiederholen.
- Start time (Startzeit) – Wählen Sie für einmalige Aktionen das Datum und die Uhrzeit aus, zu der die Aktion ausgeführt werden soll.

Für wiederkehrende Aktionen ist eine Startzeit optional. Geben Sie dies an, um den frühesten Zeitpunkt der Aktionsausführung auszuwählen. Ab diesem Zeitpunkt wird die Aktion entsprechend dem Ausdruck Recurrence (Wiederholung) wiederholt.

- Recurrence (Wiederholung) – Geben Sie in einem Cron-Ausdruck an, wie häufig eine geplante Aktion ausgeführt werden soll. Mit der Option 30 6 * * 2 wird die Aktion beispielsweise jeden Dienstag um 6:30 Uhr UTC ausgeführt.
- Endtime (Endzeit) (optional) – Optional für wiederkehrende Aktionen. Bei Angabe wird die Aktion gemäß dem Ausdruck Recurrence (Wiederholung) wiederholt und ab diesem Zeitpunkt nicht mehr ausgeführt.

Wenn eine geplante Aktion endet, wird Auto Scaling nicht automatisch auf die vorherigen Einstellungen zurückgesetzt. Konfigurieren Sie daher gegebenenfalls eine zweite geplante Aktion, damit Auto Scaling zu den ursprünglichen Einstellungen zurückkehrt.

7. Wählen Sie Add aus.
8. Wählen Sie Apply (Anwenden) aus.

Note

Geplante Aktionen werden erst gespeichert, nachdem sie angewendet wurden.

Der Namespace aws:autoscaling:scheduledaction

Wenn Sie zahlreiche geplante Aktionen konfigurieren müssen, können Sie [Konfigurationsdateien \(p. 722\)](#) oder die [Elastic Beanstalk-API \(p. 665\)](#) verwenden, um Änderungen an den Konfigurationsoptionen über eine YAML- oder JSON-Datei vorzunehmen. Mit diesen Methoden können Sie auch auf die [Suspend-Option \(p. 674\)](#) zugreifen, um eine wiederkehrende geplante Aktion vorübergehend zu deaktivieren.

Note

Wenn Sie außerhalb der Konsole an den Konfigurationsoptionen von geplanten Aktionen arbeiten, verwenden Sie das ISO 8601-Zeitformat, um Start- und Endzeiten in UTC anzugeben. Beispiel: 2015-04-28T04:07:02Z Weitere Informationen zum ISO 8601-Zeitformat finden Sie unter [Date and Time Formats](#). Die Datums- und Uhrzeitangaben müssen für alle geplanten Aktionen eindeutig sein.

Elastic Beanstalk stellt Konfigurationsoptionen für die Einstellungen von geplanten Aktionen im Namespace [aws:autoscaling:scheduledaction \(p. 674\)](#) bereit. Verwenden Sie das Feld `resource_name`, um den Namen der geplanten Aktion anzugeben.

Example Scheduled-scale-up-specific-time-long.config

Diese Konfigurationsdatei weist Elastic Beanstalk an, um 2015-12-12T00:00:00Z von fünf Instances auf 10 Instances hochzuskalieren.

```
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

Um die Syntax-Kurznotation mit der EB CLI oder Konfigurationsdateien zu verwenden, stellen Sie den Ressourcennamen und den Namespace voran.

```
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

Diese Konfigurationsdatei weist Elastic Beanstalk an, um 2015-12-12T07:00:00Z herunterzuskalieren.

```
option_settings:  
  ScheduledScaleDownSpecificTime.aws:autoscaling:scheduledaction:  
    MinSize: '1'  
    MaxSize: '1'  
    DesiredCapacity: '1'  
    StartTime: '2015-12-12T07:00:00Z'
```

Example Scheduled-periodic-scale-up.config

Diese Konfigurationsdatei weist Elastic Beanstalk an, jeden Tag um 9 Uhr morgens hochzuskalieren. Für die Aktion ist als Startzeit der 14. Mai 2015 und als Endzeit der 12. Januar 2016 geplant.

```
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

Diese Konfigurationsdatei weist Elastic Beanstalk an, jeden Tag um 18 Uhr auf eine nicht laufende Instance herunterzuskalieren. Wenn Sie wissen, dass die Anwendung außerhalb der Geschäftszeiten normalerweise ohne Last arbeitet, können Sie eine entsprechend geplante Aktion erstellen. Wenn die Anwendung außerhalb der Geschäftszeiten heruntergefahren werden muss, ändern Sie MaxSize auf 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

Diese Konfigurationsdatei weist Elastic Beanstalk an, jeden Freitag um 18 Uhr herunterzuskalieren. Wenn Sie wissen, dass für die Anwendung am Wochenende nicht sehr viel Datenverkehr entsteht, können Sie eine ähnliche Aktion planen.

```
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
```

Zustandsprüfungseinstellung für Auto Scaling

Mit Amazon EC2 Auto Scaling wird der Zustand der einzelnen gestarteten Amazon Elastic Compute Cloud (Amazon EC2)-Instances überwacht. Falls eine Instance unerwartet beendet wird, erkennt Auto Scaling dies und startet eine andere Instance als Ersatz. Standardmäßig verwendet die für Ihre Umgebung erstellte Auto Scaling-Gruppe [Amazon EC2-Statusprüfungen](#). Falls eine Amazon EC2-Statusprüfung für eine Instance der Umgebung fehlschlägt, wird diese von Auto Scaling heruntergefahren und ersetzt.

Amazon EC2-Statusprüfungen beziehen sich ausschließlich auf den Instance-Zustand – nicht auf den Zustand der Anwendung, des Servers oder der Docker-Container, die auf der Instance ausgeführt werden. Falls die Anwendung ausfällt, aber die Instance, auf der die Anwendung ausgeführt wird, noch stabil ist, wird sie vermutlich aus dem Load Balancer genommen, aber nicht automatisch von Auto Scaling ersetzt. Das Standardverhalten ist sinnvoll für die Fehlerbehebung. Falls die Instance von der Auto Scaling-Funktion direkt nach dem Absturz der Anwendung ersetzt wird, bemerken Sie den Absturz möglicherweise gar nicht, auch wenn dies direkt nach dem Start geschehen ist.

Sollen die Instances, deren Anwendung nicht mehr reagiert, von Auto Scaling ersetzt werden, können Sie die Auto Scaling-Gruppe mithilfe einer [Konfigurationsdatei \(p. 722\)](#) zur Verwendung der Elastic Load Balancing-Zustandsprüfungen konfigurieren. Im folgenden Beispiel wird die Gruppe zur Verwendung der Load Balancer-Zustandsprüfungen, zusätzlich zur Amazon EC2-Statusprüfung, festgelegt, um den Instance-Zustand zu bestimmen.

Example .ebextensions/autoscaling.config

```
Resources:  
  AWSEBAutoScalingGroup:  
    Type: "AWS::AutoScaling::AutoScalingGroup"  
    Properties:  
      HealthCheckType: ELB  
      HealthCheckGracePeriod: 300
```

Weitere Informationen zu den Eigenschaften `HealthCheckType` und `HealthCheckGracePeriod` finden Sie unter [AWS::AutoScaling::AutoScalingGroup](#) im AWS CloudFormation-Benutzerhandbuch und [Zustandsprüfungen für Auto Scaling-Instances](#) im Amazon EC2 Auto Scaling-Benutzerhandbuch.

Standardmäßig ist die Elastic Load Balancing-Zustandsprüfung so konfiguriert, dass über Port 80 eine TCP-Verbindung zur Instance hergestellt wird. Dies bestätigt, dass der in der Instance ausgeführte Webserver Verbindungen akzeptiert. Sie können jedoch auch die [Load Balancer-Zustandsprüfung anpassen \(p. 554\)](#), um sicherzustellen, dass sich Ihre Anwendung – und nicht nur der Webserver – in einem guten Zustand befindet. Mit der Einstellung für die Übergangsfrist wird in Sekunden festgelegt, wie lange die Zustandsprüfung einer Instance fehlschlagen darf, bevor diese beendet und ersetzt wird. Instances können, auch wenn sie aus dem Load Balancer genommen wurden, wiederhergestellt werden, daher sollten Sie einen für die Anwendung angemessenen Zeitraum vorgeben.

Load Balancer Ihrer AWS Elastic Beanstalk-Umgebung

Ein Load Balancer verteilt den Datenverkehr auf die Instances Ihrer Umgebung. Wenn Sie [Load-Balancing aktivieren \(p. 514\)](#), erstellt AWS Elastic Beanstalk einen [Elastic Load Balancing](#)-Load Balancer, der speziell auf Ihre Umgebung zugeschnitten ist. Elastic Beanstalk verwaltet diesen Load Balancer vollständig und kümmert sich um die Sicherheitseinstellungen und um die Terminierung des Load Balancers, wenn Sie Ihre Umgebung terminieren.

Alternativ können Sie einen Load Balancer für mehrere Elastic Beanstalk-Umgebungen freigeben. Mit einem freigegebenen Load Balancer sparen Sie Betriebskosten, da nicht für jede Umgebung ein dedizierter

Load Balancer erstellt wird. Sie übernehmen auch einen größere Verantwortung für die Verwaltung des freigegebenen Load Balancer, den Ihre Umgebungen verwenden.

Elastic Load Balancing verfügt über diese Load Balancer-Typen:

- [Classic Load Balancer](#) – Der Load Balancer der vorherigen Generation. Leitet den Datenverkehr aus HTTP-, HTTPS- oder TCP-Anforderungen an verschiedene Ports auf Instances in der Umgebung weiter.
- [Application Load Balancer](#) – Ein Application-Layer Load Balancer. Leitet den Datenverkehr aus HTTP- oder HTTPS-Anforderungen auf Grundlage des Anforderungspfads an verschiedene Ports auf Instances in der Umgebung weiter.
- [Network Load Balancer](#) – Ein Network-Layer Load Balancer. Leitet Datenverkehr aus TCP-Anforderungen an verschiedene Ports auf Instances in der Umgebung weiter. Unterstützt aktive und passive Zustandsprüfungen.

Elastic Beanstalk unterstützt alle drei Load Balancer-Typen. Die folgende Tabelle zeigt, welche Typen Sie mit den beiden Nutzungsmustern verwenden können:

Load Balancer-Typ	Dediziert	Freigegeben
Classic Load Balancer	✓ Ja	✗ Nein
Application Load Balancer	✓ Ja	✓ Ja
Netzwerk-Load-Balancer	✓ Ja	✗ Nein

Standardmäßig erstellt Elastic Beanstalk einen Application Load Balancer für Ihre Umgebung, wenn Sie das Load-Balancing mit der Elastic Beanstalk-Konsole oder der EB CLI aktivieren. Der Load Balancer ist so konfiguriert, dass er den HTTP-Datenverkehr auf Port 80 überwacht und diesen Datenverkehr an die Instances auf demselben Port weiterleitet. Sie können den Typ des Load Balancer, den Ihre Umgebung verwendet, nur während der Erstellung der Umgebung wählen. Sie können Einstellungen später ändern, um das Verhalten des Load Balancer Ihrer laufenden Umgebung zu verwalten, Sie können allerdings seinen Typ nicht ändern.

Note

Ihre Umgebung muss sich in einer VPC mit Subnetzen in mindestens zwei Availability Zones befinden, damit ein Application Load Balancer erstellt wird. Alle neuen AWS-Konten umfassen Standard-VPCs, die diese Anforderung erfüllen. Befindet sich Ihre Umgebung in einer VPC mit Subnetzen in nur einer Availability Zone, wird standardmäßig ein Classic Load Balancer verwendet. Wenn keine Subnetze vorhanden sind, können Sie kein Load Balancing aktivieren.

Sie können Umgebungen mit allen Load Balancer-Typen mit der Elastic Beanstalk-Konsole, dem EB CLI [eb create \(p. 1053\)](#)-Befehl oder den Elastic Beanstalk-APIs erstellen und verwalten.

In den folgenden Themen erhalten Sie weitere Informationen zu den einzelnen von Elastic Beanstalk unterstützten Load Balancer-Typen, zu ihrer Funktionalität und zu ihrer Konfiguration und Verwaltung in einer Elastic Beanstalk-Umgebung. Sie erfahren auch, wie Sie einen Load Balancer für das [Hochladen von Zugriffsprotokollen \(p. 609\)](#) in Amazon S3 konfigurieren.

Themen

- [Konfigurieren eines Classic Load Balancers \(p. 556\)](#)
- [Konfigurieren eines Application Load Balancer \(p. 566\)](#)
- [Gemeinsamen Application Load Balancer konfigurieren \(p. 583\)](#)
- [Konfigurieren eines Network Load Balancers \(p. 599\)](#)
- [Konfigurieren von Zugriffsprotokollen \(p. 609\)](#)

Konfigurieren eines Classic Load Balancers

Wenn Sie [enable load balancing \(Load Balancing aktivieren\)](#) (p. 514), ist in der AWS Elastic Beanstalk-Umgebung ein Elastic Load Balancing-Load Balancer vorhanden, mit dem der Datenverkehr auf die Instances in der Umgebung verteilt wird. Elastic Load Balancing unterstützt verschiedene Load Balancer-Typen. Weitere Informationen dazu finden Sie im [Elastic Load Balancing-Benutzerhandbuch](#). Elastic Beanstalk kann einen Load Balancer für Sie erstellen oder Sie einen von Ihnen erstellten freigegebenen Load Balancer angeben lassen.

Dieses Thema beschreibt die Konfiguration eines [Classic Load Balancers](#), den Elastic Beanstalk erstellt und Ihrer Umgebung zuweist. Weitere Informationen zur Konfiguration aller Load Balancer-Typen, die Elastic Beanstalk unterstützt, finden Sie unter [Load Balancer Ihrer AWS Elastic Beanstalk-Umgebung \(p. 554\)](#).

Note

Sie können den Typ des Load Balancer, den Ihre Umgebung verwendet, nur während der Erstellung der Umgebung wählen. Sie können Einstellungen später ändern, um das Verhalten des Load Balancer Ihrer laufenden Umgebung zu verwalten, Sie können allerdings seinen Typ nicht ändern.

Einführung

Ein [Classic Load Balancer](#) ist der Elastic Load Balancer der vorherigen Generation. Er unterstützt das Weiterleiten des Datenverkehrs aus HTTP-, HTTPS- oder TCP-Anforderungen an verschiedene Ports auf Instances in der Umgebung.

Wenn Ihre Umgebung einen Classic Load Balancer verwendet, konfiguriert Elastic Beanstalk diesen standardmäßig auf [listen](#) für HTTP-Datenverkehr über Port 80 und leitet ihn an Instances am selben Port weiter. Zur Unterstützung sicherer Verbindungen können Sie einen Listener auf Port 443 und ein TLS-Zertifikat für den Load Balancer konfigurieren.

Der Load Balancer überprüft mittels einer [Zustandsprüfung](#), ob die Amazon EC2-Instances, auf denen die Anwendung ausgeführt wird, stabil sind. Die Zustandsprüfung sendet in festgelegten Intervallen eine Anfrage an eine angegebene URL. Wenn die URL eine Fehlermeldung zurückgibt oder nicht innerhalb eines definierten Timeouts antwortet, schlägt die Zustandsprüfung fehl.

Sofern sich die Anwendungsleistung verbessert lässt, wenn mehrere Anforderungen vom gleichen Client auf einem einzigen Server verarbeitet werden, können Sie den Load Balancer für die Nutzung von [Sticky Sessions](#) konfigurieren. Bei Verwendung von Sticky Sessions fügt der Load Balancer den HTTP-Antworten ein Cookie zur Identifizierung der Amazon EC2-Instance hinzu, von der die Anforderung verarbeitet wurde. Bei der nächsten eingehenden Anforderung vom gleichen Client nutzt der Load Balancer das Cookie, um die Anforderung an dieselbe Instance zu senden.

Wenn [zonenübergreifendes Load Balancing](#) aktiviert ist, verteilt jeder Load Balancer-Knoten für Ihren Classic Load Balancer Anfragen gleichmäßig auf die registrierten Instances in allen aktivierte Availability Zones. Wenn zonenübergreifendes Load Balancing deaktiviert ist, verteilt jeder Load Balancer-Knoten Anfragen gleichmäßig nur auf die registrierten Instances in seiner aktivierte Availability Zone.

Falls eine Instance aus dem Load Balancer entfernt wird (aufgrund ihres instabilen Zustands oder wegen einer Herunterskalierung der Umgebung), kann die Instance mithilfe von [Connection Draining](#) die Anforderungen abschließen, bevor die Verbindung zwischen der Instance und dem Load Balancer getrennt wird. Sie können den Zeitraum ändern, in dem Instances eine Antwort senden können, oder den Verbindungsausgleich vollständig deaktivieren.

Note

Connection Draining wird standardmäßig aktiviert, wenn Sie eine Umgebung mit der Elastic Beanstalk-Konsole oder der EB CLI erstellen. Bei anderen Clients können Sie die Aktivierung in den [Konfigurationsoptionen \(p. 565\)](#) vornehmen.

Mit erweiterten Load Balancer-Einstellungen können Listener für beliebige Ports eingerichtet, zusätzliche Sticky Session-Einstellungen bearbeitet und eine sichere Verbindung vom Load Balancer zu den EC2-Instances konfiguriert werden. Diese Einstellungen stehen über [Konfigurationsoptionen \(p. 565\)](#) zur Verfügung, die Sie mithilfe von Konfigurationsdateien im Quellcode oder über die Elastic Beanstalk-API direkt in der Umgebung festlegen können. Viele dieser Einstellungen sind auch in der Elastic Beanstalk-Konsole verfügbar. Außerdem können Sie den Load Balancer für den [Upload von Zugriffsprotokollen \(p. 609\)](#) in Amazon S3 konfigurieren.

Classic Load Balancer mit der Elastic Beanstalk-Konsole konfigurieren

Sie können die Elastic Beanstalk-Konsole zum Konfigurieren der Ports, des HTTPS-Zertifikats und anderer Einstellungen eines Classic Load Balancer verwenden, während die Umgebung erstellt wird oder später, wenn die Umgebung bereits ausgeführt wird.

So konfigurieren Sie einen Classic Load Balancer in der Elastic Beanstalk-Konsole während der Erstellung der Umgebung:

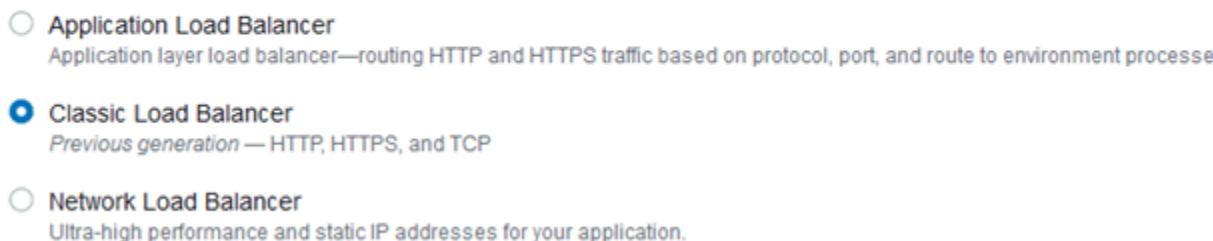
1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Environments (Umgebungen).
3. Wählen Sie [Create a new environment \(Neue Umgebung erstellen\) \(p. 437\)](#), um mit der Erstellung Ihrer Umgebung zu beginnen.
4. Wählen Sie auf der Hauptseite des Assistenten zuerst Configure more options (Weitere Optionen konfigurieren) und dann Create environment (Umgebung erstellen) aus.
5. Wählen Sie die Konfigurationsvoreinstellung High availability (Hohe Verfügbarkeit) aus.

Alternativ können Sie in der Konfigurationskategorie Capacity (Kapazität) einen Umgebungstyp mit Load balanced (Lastenverteilung) konfigurieren. Details dazu finden Sie unter [Kapazität \(p. 445\)](#).

6. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.
7. Wählen Sie die Option Classic Load Balancer aus, falls sie nicht bereits ausgewählt ist.

Elastic Beanstalk > Applications > getting-started-app

Modify load balancer



8. Nehmen Sie alle Konfigurationsänderungen am Classic Load Balancer vor, die für Ihre Umgebung erforderlich sind.
9. Wählen Sie Save (Speichern) aus und nehmen Sie dann sämtliche weiteren Konfigurationsänderungen vor, die Ihre Umgebung erfordert.
10. Wählen Sie Create environment (Umgebung erstellen) aus.

So konfigurieren Sie den Classic Load Balancer einer laufenden Umgebung in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.

Note

Ist in der Konfigurationskategorie Load balancer die Schaltfläche Edit (Bearbeiten) nicht vorhanden, verfügt Ihre Umgebung über keinen Load Balancer. Informationen zur Einrichtung finden Sie unter [Ändern des Umgebungstyps \(p. 514\)](#).

5. Nehmen Sie die für Ihre Umgebung erforderlichen Konfigurationsänderungen am Classic Load Balancer vor.
6. Wählen Sie Apply (Anwenden) aus.

Classic Load Balancer-Einstellungen

- [Listener \(p. 558\)](#)
- [Sitzungen \(p. 561\)](#)
- [Zonenübergreifendes Load Balancing \(p. 562\)](#)
- [Verbindungsausgleich \(p. 562\)](#)
- [Zustandsprüfung \(p. 563\)](#)

Listener

Verwenden Sie diese Liste, um Listener für Ihren Load Balancer anzugeben. Jeder Listener leitet den eingehenden Client-Datenverkehr eines bestimmten Ports mittels eines festgelegten Protokolls an Ihre Instances weiter. Anfänglich wird in der Liste der Standard-Listener angezeigt, der eingehenden HTTP-Datenverkehr auf Port 80 an Instance-Server Ihrer Umgebung weiterleitet, die HTTP-Port 80 überwachen.

Classic Load Balancer

You can specify listeners for your load balancer. Each listener routes incoming client traffic on a specified port using a specific protocol. By default, we've configured your load balancer with a standard web server on port 80.

Action				
Port	Protocol	Instance port	Instance protocol	SSL cert
80	HTTP	80	HTTP	--

So konfigurieren Sie einen vorhandenen Listener

1. Aktivieren Sie das Kontrollkästchen neben dem Tabelleneintrag und wählen Sie Actions (Aktionen) und dann die gewünschte Aktion aus.
2. Falls Sie Edit (Bearbeiten) ausgewählt haben, verwenden Sie das Dialogfeld Classic Load Balancer listener (Listener Ihres Classic Load Balancers), um die Einstellungen zu bearbeiten, und wählen Sie dann Save (Speichern) aus.

Beispielsweise können Sie den Standard-Listener bearbeiten und die Einstellung Protocol (Protokoll) von HTTP in TCP ändern, wenn der Load Balancer die Anforderungen unverändert weiterleiten soll. Dadurch wird verhindert, dass der Load Balancer Header umschreibt (einschließlich X-Forwarded-For). Die Technik funktioniert nicht mit Sticky Sessions.

Classic Load Balancer listener

Listener port
80

Listener protocol
The load balancer transport protocol to use for routing.
TCP

Instance port
The port on which the instance server is listening.
80

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.
TCP

Cancel **Save**

So fügen Sie einen Listener hinzu

1. Wählen Sie Add listener (Listener hinzufügen) aus.
2. Konfigurieren Sie im Dialogfeld Classic Load Balancer listener (Listener Ihres Classic Load Balancers) die gewünschten Einstellungen und wählen Sie dann Add (Hinzufügen) aus.

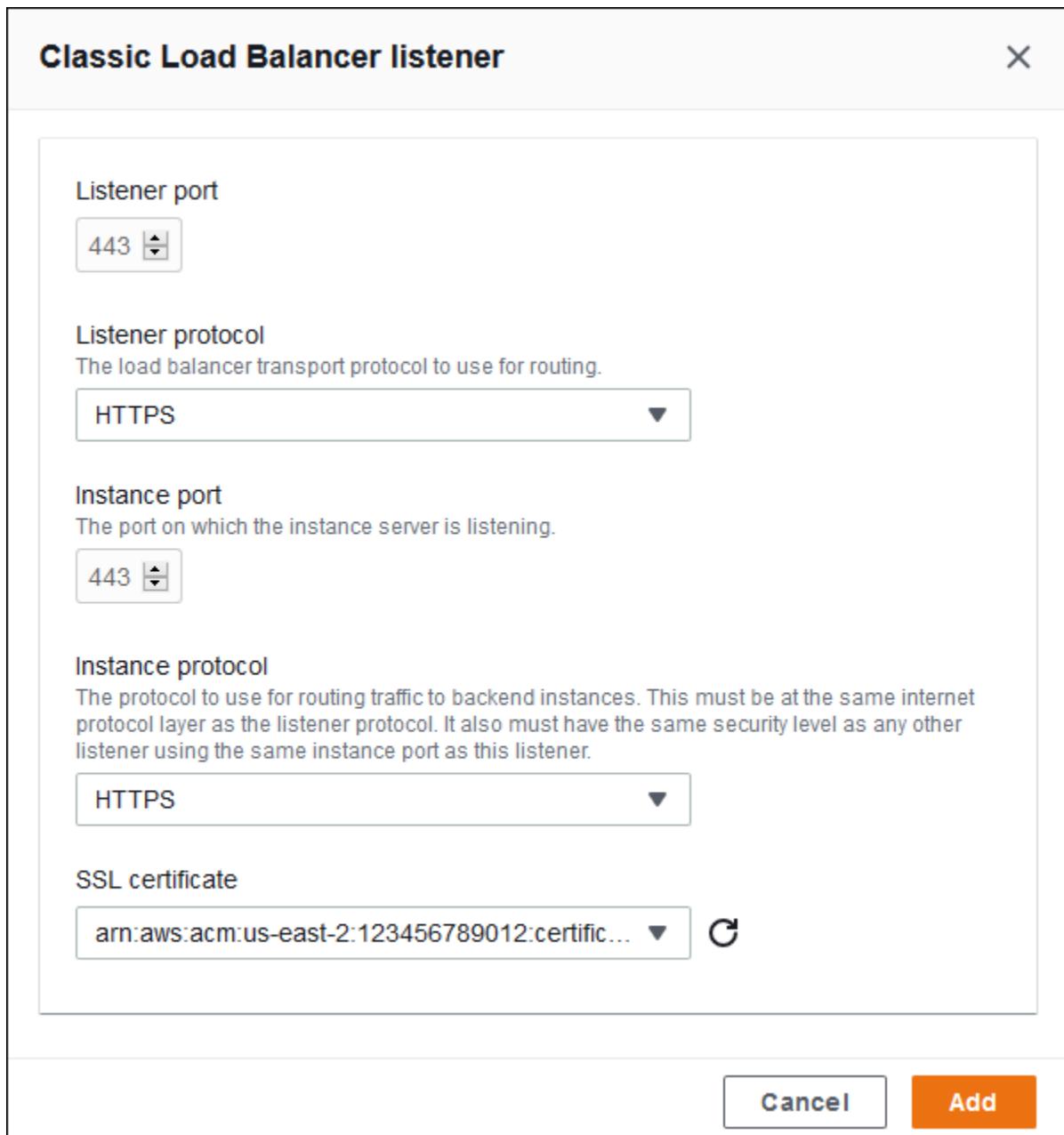
Ein häufiger Anwendungsfall ist das Hinzufügen eines sicheren Listener. Im Beispiel in der folgenden Abbildung wird ein Listener für HTTPS-Datenverkehr auf Port 443 hinzugefügt. Dieser Listener leitet den eingehenden Datenverkehr an Instance-Server der Umgebung weiter, die Port 443 auf HTTPS-Datenverkehr überwachen.

Bevor Sie einen HTTPS-Listener konfigurieren können, stellen Sie sicher, dass Sie über ein gültiges SSL-Zertifikat verfügen. Führen Sie eine der folgenden Aufgaben aus:

- Wenn AWS Certificate Manager (ACM) [in Ihrer AWS Region verfügbar ist](#), erstellen oder importieren Sie ein Zertifikat mit ACM. Weitere Informationen zum Anfordern eines ACM-Zertifikats finden Sie unter [Zertifikat anfordern](#) im AWS Certificate Manager-Benutzerhandbuch. Weitere Informationen zum Importieren von Zertifikaten von Drittanbietern in AWS Certificate Manager finden Sie unter [Importieren von Zertifikaten](#) im Leitfaden AWS Certificate Manager-Benutzerhandbuch.
- Wenn AWS Certificate Manager in Ihrer AWS-Region nicht [verfügbar ist](#), laden Sie Ihr vorhandenes Zertifikat und den vorhandenen Schlüssel in IAM hoch. Weitere Informationen zum Erstellen und Hochladen von Zertifikaten in IAM finden Sie unter [Arbeiten mit Serverzertifikaten](#) im IAM-Benutzerhandbuch.

Weitere Informationen zum Konfigurieren von HTTPS und zum Verwenden von Zertifikaten in Elastic Beanstalk finden Sie unter [Konfigurieren von HTTPS für Elastic Beanstalk-Umgebung \(p. 778\)](#).

Wählen Sie unter SSL certificate (SSL-Zertifikat) den ARN des SSL-Zertifikats aus. Beispiel:
`arn:aws:iam::123456789012:server-certificate/abc/certs/build` oder `arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`.



Informationen zum Konfigurieren von HTTPS und zum Verwenden von Zertifikaten in Elastic Beanstalk finden Sie unter [Konfigurieren von HTTPS für Elastic Beanstalk-Umgebung \(p. 778\)](#).

Sitzungen

Aktivieren oder deaktivieren Sie das Kontrollkästchen Session stickiness enabled (Gebundene Sitzung aktiviert) zum Aktivieren oder Deaktivieren von Sticky Sessions. Verwenden Sie Cookie duration (Cookie-Dauer) zum Konfigurieren der Dauer einer Sticky Session (auf bis zu **1000000** Sekunden). Wählen Sie in der Liste Load balancer ports (Load Balancer-Ports) Listener-Ports aus, für die die Standardrichtlinie (AWSEB-ELB-StickinessPolicy) gilt.

Sessions

The following settings let you control whether the load balancer routes requests for the same session to the Amazon EC2 instance 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

Zonenübergreifendes Load Balancing

Aktivieren oder deaktivieren Sie das Kontrollkästchen Load balancing across multiple Availability Zones enabled (Das Load Balancing ist für mehrere Availability Zones aktiviert) zum Aktivieren oder Deaktivieren von zonenübergreifendem Load Balancing.

Cross-zone load balancing

Load balancing across multiple Availability Zones enabled

Verbindungsmanagement

Aktivieren oder deaktivieren Sie das Kontrollkästchen Connection draining enabled (Connection Draining aktiviert) zum Aktivieren oder Deaktivieren von Connection Draining. Legen Sie das Draining timeout (Draining-Timeout) fest (auf bis zu **3600** Sekunden).

The screenshot shows a configuration panel for 'Connection draining'. It includes a checkbox labeled 'Connection draining enabled' which is unchecked. Below it is a section titled 'Draining timeout' with the sub-instruction 'Maximum time that the load balancer maintains connections to an Amazon EC2 instance before forcibly closing connections.' A numeric input field displays '20' followed by a dropdown menu set to 'seconds'.

Zustandsprüfung

Verwenden Sie die folgenden Einstellungen zum Konfigurieren von Zustandsprüfungen des Load Balancer:

- Health check path (Pfad für die Zustandsprüfung) – Der Pfad, an den der Load Balancer Zustandsprüfungsanforderungen sendet. Falls Sie den Pfad nicht festlegen, versucht der Load Balancer, über Port 80 eine TCP-Verbindung herzustellen, um die Zustandsprüfung auszuführen.
- Timeout – Die Wartezeit in Sekunden, bis eine Antwort von der Zustandsprüfung eingeht
- Interval (Intervall) – Der Zeitraum in Sekunden zwischen Zustandsprüfungen einer einzelnen Instance. Der Intervall-Wert muss größer als der Timeout-Wert sein.
- Unhealthy threshold, Healthy threshold – Die Anzahl der Zustandsprüfungen, die nicht bestanden bzw. bestanden sein müssen, bevor Elastic Load Balancing den Zustand der Instance ändert.

The screenshot shows the 'Health check' configuration section of the AWS Elastic Beanstalk Classic Load Balancer. It includes fields for 'Health check path', 'Timeout', 'Interval', 'Unhealthy threshold', and 'Healthy threshold'. Each field has a numerical input and an up/down arrow for adjustment.

Setting	Value
Health check path	(empty text input)
Timeout	5 seconds
Interval	10 seconds
Unhealthy threshold	5 requests
Healthy threshold	3 requests

Note

Die Zustandsprüfung für Elastic Load Balancing hat keinen Einfluss auf das Verhalten der Zustandsprüfung der Auto Scaling-Gruppe einer Umgebung. Instances, die die Elastic Load Balancing-Zustandsprüfung nicht bestehen, werden nicht automatisch durch die Amazon EC2 Auto Scaling ersetzt, es sei denn, Sie konfigurieren Amazon EC2 Auto Scaling manuell entsprechend. Details dazu finden Sie unter [Zustandsprüfungseinstellung für Auto Scaling \(p. 554\)](#).

Weitere Informationen zu Zustandsprüfungen und deren Auswirkungen auf den Gesamtstatus der Umgebung finden Sie unter [Grundlegende Zustandsberichte \(p. 821\)](#).

Classic Load Balancer mit der EB CLI konfigurieren

Sie werden von der EB CLI aufgefordert, einen Load Balancer-Typ auszuwählen, wenn Sie `eb create (p. 1053)` ausführen.

```
$ 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):
```

Drücken Sie die Eingabetaste, um `classic` auszuwählen.

Sie können einen Load Balancer-Typ auch mithilfe der `--elb-type`-Option angeben.

```
$ eb create test-env --elb-type classic
```

Classic Load Balancer-Konfiguration in Namespaces

In den folgenden Namespaces finden Sie Einstellungen im Zusammenhang mit Classic Load Balancern:

- [aws:elb:healthcheck \(p. 699\)](#) – Konfigurieren Sie Grenzwerte und überprüfen Sie Intervall sowie Timeout für Load Balancer-Zustandsprüfungen.
- [aws:elasticbeanstalk:application \(p. 683\)](#) – Konfigurieren Sie die Zustandsprüfungs-URL.
- [aws:elb:loadbalancer \(p. 699\)](#) – Aktivieren Sie zonenübergreifendes Load Balancing. Ordnen Sie dem Load Balancer Sicherheitsgruppen zu und überschreiben Sie die von Elastic Beanstalk erstellte Standardsicherheitsgruppe. In diesem Namespace finden Sie auch veraltete Optionen für die Konfiguration von Standard- und sicheren Listenern, die durch Optionen im Namespace `aws:elb:listener` ersetzt wurden.
- [aws:elb:listener \(p. 701\)](#) – Konfigurieren Sie den Standard-Listener auf Port 80, einen sicheren Listener auf Port 443 oder weitere Listener für beliebige Protokolle auf einem frei wählbaren Port. Sofern Sie `aws:elb:listener` als Namespace angeben, gelten die Einstellungen für den Standard-Listener auf Port 80. Wenn Sie einen Port festlegen (z. B. `aws:elb:listener:443`), wird ein Listener auf diesem Port konfiguriert.
- [aws:elb:policies \(p. 703\)](#) – Konfigurieren Sie zusätzliche Einstellungen für den Load Balancer. Richten Sie mit den Optionen in diesem Namespace Listener für beliebige Ports ein, bearbeiten Sie zusätzliche Sticky Session-Einstellungen und konfigurieren Sie eine sichere Verbindung vom Load Balancer zu den Amazon EC2-Instances.

Die EB CLI und Elastic Beanstalk-Konsole wenden die empfohlenen Werte für die vorhergehenden Optionen an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter [Empfohlene Werte \(p. 648\)](#).

Example .ebextensions/loadbalancer-terminatehttps.config

Die folgende Beispielkonfigurationsdatei wird zur Erstellung eines HTTPS-Listeners auf Port 443, für die Zuweisung eines Zertifikats, mit dem der Load Balancer die sichere Verbindung beendet, und zur Deaktivierung des Standard-Listeners auf Port 80 verwendet. Die entschlüsselten Anforderungen werden dann vom Load Balancer über HTTP:80 an die EC2-Instances in der Umgebung weitergeleitet.

```
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
```

Konfigurieren eines Application Load Balancer

Wenn Sie [enable load balancing \(Load Balancing aktivieren\)](#) (p. 514), ist in der AWS Elastic Beanstalk-Umgebung ein Elastic Load Balancing-Load Balancer vorhanden, mit dem der Datenverkehr auf die Instances in der Umgebung verteilt wird. Elastic Load Balancing unterstützt verschiedene Load Balancer-Typen. Weitere Informationen dazu finden Sie im [Elastic Load Balancing-Benutzerhandbuch](#). Elastic Beanstalk kann einen Load Balancer für Sie erstellen oder Sie einen von Ihnen erstellten freigegebenen Load Balancer angeben lassen.

Dieses Thema beschreibt die Konfiguration eines [Application Load Balancers](#), den Elastic Beanstalk erstellt und Ihrer Umgebung zuweist. Weitere Informationen finden Sie auch unter [the section called “Geteile Application Load Balancer” \(p. 583\)](#). Weitere Informationen zur Konfiguration aller Load Balancer-Typen, die Elastic Beanstalk unterstützt, finden Sie unter [the section called “Load Balancer” \(p. 554\)](#).

Note

Sie können den Typ des Load Balancer, den Ihre Umgebung verwendet, nur während der Erstellung der Umgebung wählen. Sie können Einstellungen ändern, um das Verhalten des Load Balancer Ihrer laufenden Umgebung zu verwalten, Sie können allerdings seinen Typ nicht ändern. Sie können auch nicht von einem dedizierten zu einem freigegebenen Load Balancer wechseln und umgekehrt.

Einführung

Ein Application Load Balancer untersucht Datenverkehr auf Protokollebene des Anwendungsnetzwerks, um den Pfad der Anforderung zu bestimmen, sodass er Anforderungen für verschiedene Pfade an verschiedene Ziele leiten kann.

Wenn Ihre Umgebung einen Application Load Balancer verwendet, konfiguriert Elastic Beanstalk diesen standardmäßig so, dass er die gleiche Funktion wie ein Classic Load Balancer ausführt. Der standardmäßige Listener akzeptiert HTTP-Anfragen auf Port 80 und leitet sie an die Instances in Ihrer Umgebung weiter. Sie können einen sicheren Listener auf Port 443 mit einem Zertifikat zur Verschlüsselung von HTTPS-Datenverkehr hinzufügen, Zustandsprüfungsverhalten konfigurieren und Zugriffsprotokolle vom Load Balancer an einen Amazon Simple Storage Service (Amazon S3)-Bucket übermitteln.

Note

Im Gegensatz zu einem Classic Load Balancer oder einem Network Load Balancer kann ein Application Load Balancer keine TCP- oder SSL/TLS-Listener auf Transportschicht (Schicht 4) haben. Er unterstützt nur HTTP- und HTTPS-Listener. Darüber hinaus kann er keine Backend-Authentifizierung zum Authentifizieren von HTTPS-Verbindungen zwischen dem Load Balancer und Backend-Instances verwenden.

In einer Elastic Beanstalk-Umgebung können Sie einen Application Load Balancer verwenden, um Datenverkehr für bestimmte Pfade zu einem anderen Prozess auf Ihren Webserver-Instance zu routen. Mit einem Classic Load Balancer wird der gesamte Datenverkehr zu einem Listener an einen einzelnen Port auf den Backend-Instances weitergeleitet. Mit einem Application Load Balancer können Sie mehrere Regeln auf dem Listener so konfigurieren, dass Anforderungen für bestimmte Pfade an verschiedene Backend-Prozesse weitergeleitet werden. Sie konfigurieren jeden Prozess mit dem Port, den der Prozess überwacht.

Sie können z. B. eine Anmeldung separat von der Hauptanwendung durchführen. Während die Hauptanwendung auf den Instances Ihrer Umgebung die meisten Anforderungen akzeptiert und auf Port 80 überwacht, überwacht Ihr Anmeldevorgang auf Port 5000 und akzeptiert Anforderungen an den /login-Pfad. Alle eingehenden Anforderungen von Clients gehen auf Port 80 ein. Mit einem Application Load Balancer können Sie einen einzigen Listener für eingehenden Datenverkehr auf Port 80 mit zwei Regeln konfigurieren, die den Datenverkehr je nach Pfad in der Anforderung an zwei separate Prozesse weiterleiten. Sie fügen eine benutzerdefinierte Regel hinzu, die den Datenverkehr für /login an den Anmeldeprozess weiterleitet, der Port 5000 überwacht. Die Standardregel leitet sämtlichen weiteren Datenverkehr an die Prozessüberwachung für die Hauptanwendung an Port 80 weiter.

Eine Application Load Balancer-Regel ordnet eine Anfrage einer Zielgruppe zu. In Elastic Beanstalk wird eine Zielgruppe durch einen Prozess dargestellt. Sie können einen Prozess mit Einstellungen für Protokoll, Port und Zustandsprüfung konfigurieren. Der Prozess stellt den Prozess dar, der auf Instances in Ihrer Umgebung ausgeführt wird. Der Standardprozess ist ein Listener auf Port 80 des Reverse Proxy (nginx oder Apache), der vor Ihrer Anwendung ausgeführt wird.

Note

Außerhalb von Elastic Beanstalk entspricht eine Zielgruppe einer Gruppe von Instances. Ein Listener kann Regeln und Zielgruppen für die Weiterleitung des Datenverkehrs an verschiedene Instances basierend auf dem Pfad verwenden. Innerhalb von Elastic Beanstalk sind alle Instances in Ihrer Umgebung identisch, sodass die Unterscheidung zwischen Prozessen gemacht wird, die auf verschiedenen Ports überwachen.

Ein Classic Load Balancer verwendet eine einzige Zustandsprüfungspfad für die gesamte Umgebung. Bei einem Application Load Balancer verfügt jeder Prozess über einen separaten Zustandsprüfungspfad, der durch den Load Balancer und der von Elastic Beanstalk unterstützte Zustandsprüfung überwacht wird.

Zur Verwendung Ihres Application Load Balancer muss Ihre Umgebung in einer Standard- oder benutzerdefinierten VPC sein und über eine Servicerolle mit dem Standardsatz an Berechtigungen verfügen. Wenn Sie über eine ältere Servicerolle verfügen, müssen Sie ggf. Ihre [Berechtigungen aktualisieren \(p. 908\)](#), um elasticloadbalancing:DescribeTargetHealth und elasticloadbalancing:DescribeLoadBalancers einzuschließen. Weitere Informationen über Application Load Balancer finden Sie unter [What Is an Application Load Balancer?](#).

Note

Die Zustandsprüfung bei Application Load Balancer verwendet nicht den Zustandsprüfungspfad für Elastic Beanstalk. Stattdessen wird der spezifische Pfad verwendet, der für jeden Prozess separat konfiguriert ist.

Application Load Balancer mit der Elastic Beanstalk-Konsole konfigurieren

Sie können die Elastic Beanstalk-Konsole verwenden, um die Listener, Prozesse und Regeln eines Application Load Balancers während der Erstellung der Umgebung oder später (wenn Ihre Umgebung bereits ausgeführt wird) zu konfigurieren.

So konfigurieren Sie einen Application Load Balancer in der Elastic Beanstalk-Konsole während der Erstellung der Umgebung:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Environments (Umgebungen).
3. Wählen Sie [Create a new environment \(Neue Umgebung erstellen\) \(p. 437\)](#), um mit der Erstellung Ihrer Umgebung zu beginnen.
4. Wählen Sie auf der Hauptseite des Assistenten zuerst Configure more options (Weitere Optionen konfigurieren) und dann Create environment (Umgebung erstellen) aus.

5. Wählen Sie die Konfigurationsvoreinstellung High availability (Hohe Verfügbarkeit) aus.
Alternativ können Sie in der Konfigurationskategorie Capacity (Kapazität) einen Umgebungstyp mit Load balanced (Lastenverteilung) konfigurieren. Details dazu finden Sie unter [Kapazität \(p. 445\)](#).
6. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.
7. Wählen Sie die Optionen Application Load Balancer und Dedicated (Dediziert) aus, falls sie nicht bereits ausgewählt sind.

Elastic Beanstalk > Applications > getting-started-app

Modify load balancer

Load balancer type

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.

Dedicated

Use a load balancer that Elastic Beanstalk creates exclusively for this environment.

Shared

Use a load balancer that someone in your account has created and can be shared among multiple Elastic Beanstalk environments.

8. Nehmen Sie alle Konfigurationsänderungen am Application Load Balancer vor, die für Ihre Umgebung erforderlich sind.
9. Wählen Sie Save (Speichern) aus und nehmen Sie dann sämtliche weiteren Konfigurationsänderungen vor, die Ihre Umgebung erfordert.
10. Wählen Sie Create environment (Umgebung erstellen) aus.

So konfigurieren Sie den Application Load Balancer einer laufenden Umgebung in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.

Note

Ist in der Konfigurationskategorie Load balancer die Schaltfläche Edit (Bearbeiten) nicht vorhanden, verfügt Ihre Umgebung über keinen Load Balancer. Informationen zur Einrichtung finden Sie unter [Ändern des Umgebungstyps \(p. 514\)](#).

5. Nehmen Sie die Konfigurationsänderungen am Application Load Balancer vor, die Ihre Umgebung erfordert.
6. Wählen Sie Apply (Anwenden) aus.

Application Load Balancer-Einstellungen

- [Listener \(p. 569\)](#)
- [Prozesse \(p. 570\)](#)
- [Regeln \(p. 574\)](#)
- [Zugangsprotokollerfassung \(p. 576\)](#)

Listener

Verwenden Sie diese Liste, um Listener für Ihren Load Balancer anzugeben. Jeder Listener leitet den eingehenden Client-Datenverkehr eines bestimmten Ports mittels eines festgelegten Protokolls an mindestens einen Prozess auf Ihren Instances weiter. Zunächst wird in der Liste der Standard-Listener angezeigt, der den eingehenden HTTP-Datenverkehr auf Port 80 an einen Prozess mit der Bezeichnung default (Standard) weiterleitet.

<input type="checkbox"/>	Port	Protocol	SSL certificate	Default process	Actions ▾
<input type="checkbox"/>	80	HTTP	--	default	

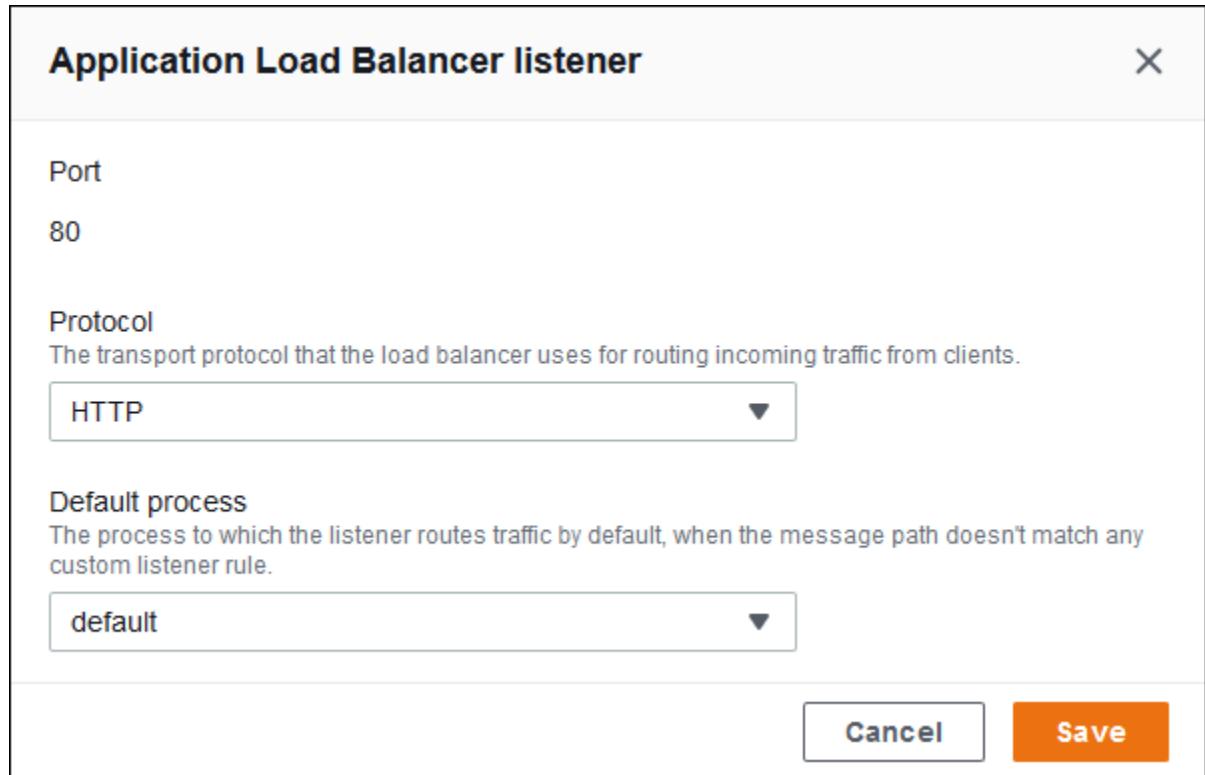
So konfigurieren Sie einen vorhandenen Listener

1. Aktivieren Sie das Kontrollkästchen neben seinem Tabelleneintrag und wählen Sie dann Actions (Aktionen), Edit (Bearbeiten) aus.
2. Verwenden Sie das Dialogfeld Application Load Balancer listener, um die Einstellungen zu bearbeiten, und wählen Sie dann Save (Speichern) aus.

So fügen Sie einen Listener hinzu

1. Wählen Sie Add listener (Listener hinzufügen) aus.
2. Konfigurieren Sie im Dialogfeld Application Load Balancer listener (Application Load Balancer-Listener) die gewünschten Einstellungen. Wählen Sie anschließend Add (Hinzufügen) aus.

Verwenden Sie die Dialogfeldeinstellungen unter Application Load Balancer listener (Application Load Balancer-Listener), um den Port und das Protokoll auszuwählen, auf dem der Listener den Datenverkehr überwacht, sowie den Prozess, an den der Datenverkehr weitergeleitet werden soll. Bei Auswahl des HTTPS-Protokolls konfigurieren Sie die SSL-Einstellungen.



Bevor Sie einen HTTPS-Listener konfigurieren können, stellen Sie sicher, dass Sie über ein gültiges SSL-Zertifikat verfügen. Führen Sie eine der folgenden Aufgaben aus:

- Wenn AWS Certificate Manager (ACM) [in Ihrer AWS Region verfügbar ist](#), erstellen oder importieren Sie ein Zertifikat mit ACM. Weitere Informationen zum Anfordern eines ACM-Zertifikats finden Sie unter [Zertifikat anfordern](#) im AWS Certificate Manager-Benutzerhandbuch. Weitere Informationen zum Importieren von Zertifikaten von Drittanbietern in AWS Certificate Manager finden Sie unter [Importieren von Zertifikaten](#) im Leitfaden AWS Certificate Manager-Benutzerhandbuch.
- Wenn AWS Certificate Manager in Ihrer AWS-Region nicht [verfügbar ist](#), laden Sie Ihr vorhandenes Zertifikat und den vorhandenen Schlüssel in IAM hoch. Weitere Informationen zum Erstellen und Hochladen von Zertifikaten in IAM finden Sie unter [Arbeiten mit Serverzertifikaten](#) im IAM-Benutzerhandbuch.

Weitere Informationen zum Konfigurieren von HTTPS und zum Verwenden von Zertifikaten in Elastic Beanstalk finden Sie unter [Konfigurieren von HTTPS für Elastic Beanstalk-Umgebung \(p. 778\)](#).

Prozesse

Verwenden Sie diese Liste, um Prozesse für Ihren Load Balancer anzugeben. Ein Prozess ist ein Ziel für Listener, an das Datenverkehr weitergeleitet wird. Jeder Listener leitet den eingehenden Client-Datenverkehr eines bestimmten Ports mittels eines festgelegten Protokolls an mindestens einen Prozess auf Ihren Instances weiter. Anfänglich zeigt die Liste den Standardprozess, der eingehenden HTTP-Datenverkehr auf Port 80 überwacht.

The screenshot shows a table titled 'Processes' with the following columns: Name, Port, Protocol, HTTP code, and Health check path. There is one row for 'default' with port 80 and protocol HTTP. An 'Actions' button is visible at the top right.

Name	Port	Protocol	HTTP code	Health check path
default	80	HTTP		/

Sie können die Einstellungen eines vorhandenen Prozesses bearbeiten oder einen neuen Prozess hinzufügen. Um mit dem Bearbeiten eines Prozesses auf der Liste zu beginnen oder Prozesse zur Liste hinzuzufügen, verwenden Sie dieselben Schritte, die für die [Liste der Listener \(p. 569\)](#) aufgelistet sind. Das Dialogfeld Environment process (Umgebungsprozess) wird geöffnet.

Prozesseinstellungen im Dialogfeld für die Umgebung des Application Load Balancers

- [Definition \(p. 571\)](#)
- [Zustandsprüfung \(p. 572\)](#)
- [Sitzungen \(p. 574\)](#)

Definition

Verwenden Sie diese Einstellungen zum Definieren des Prozesses: Name, Port und Protocol (Protokoll) für die Überwachung auf Anforderungen.

The dialog box has fields for Name (set to 'default'), Port (set to 80), and Protocol (set to 'HTTP').

Name
default

Port
80

Protocol
HTTP

Zustandsprüfung

Verwenden Sie die folgenden Einstellungen zum Konfigurieren von Zustandsprüfungen für den Prozess:

- HTTP code (HTTP-Code) – Der HTTP-Statuscode, der einen funktionsfähigen Prozess bestimmt
- Path (Pfad) – Der Anforderungspfad der Zustandsprüfung für den Prozess
- Timeout – Die Wartezeit in Sekunden, bis eine Antwort von der Zustandsprüfung eingeht
- Interval (Intervall) – Der Zeitraum in Sekunden zwischen Zustandsprüfungen einer einzelnen Instance
Der Intervall-Wert muss größer als der Timeout-Wert sein.
- Unhealthy threshold, Healthy threshold – Die Anzahl der Zustandsprüfungen, die nicht bestanden bzw.
bestanden sein müssen, bevor Elastic Load Balancing den Zustand der Instance ändert.
- Deregistration delay (Verzögerung der Registrierungsaufhebung) – Die Zeit in Sekunden, die gewartet
werden muss, bis aktive Anforderungen abgeschlossen sind, ehe eine Instance abgemeldet wird

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.

 5 seconds

Interval

Amount of time between health checks of an individual instance. The interval must be greater than the timeout.

 15 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

Deregistration delay

Amount of time to wait for active requests to complete before deregistering.

 20 seconds

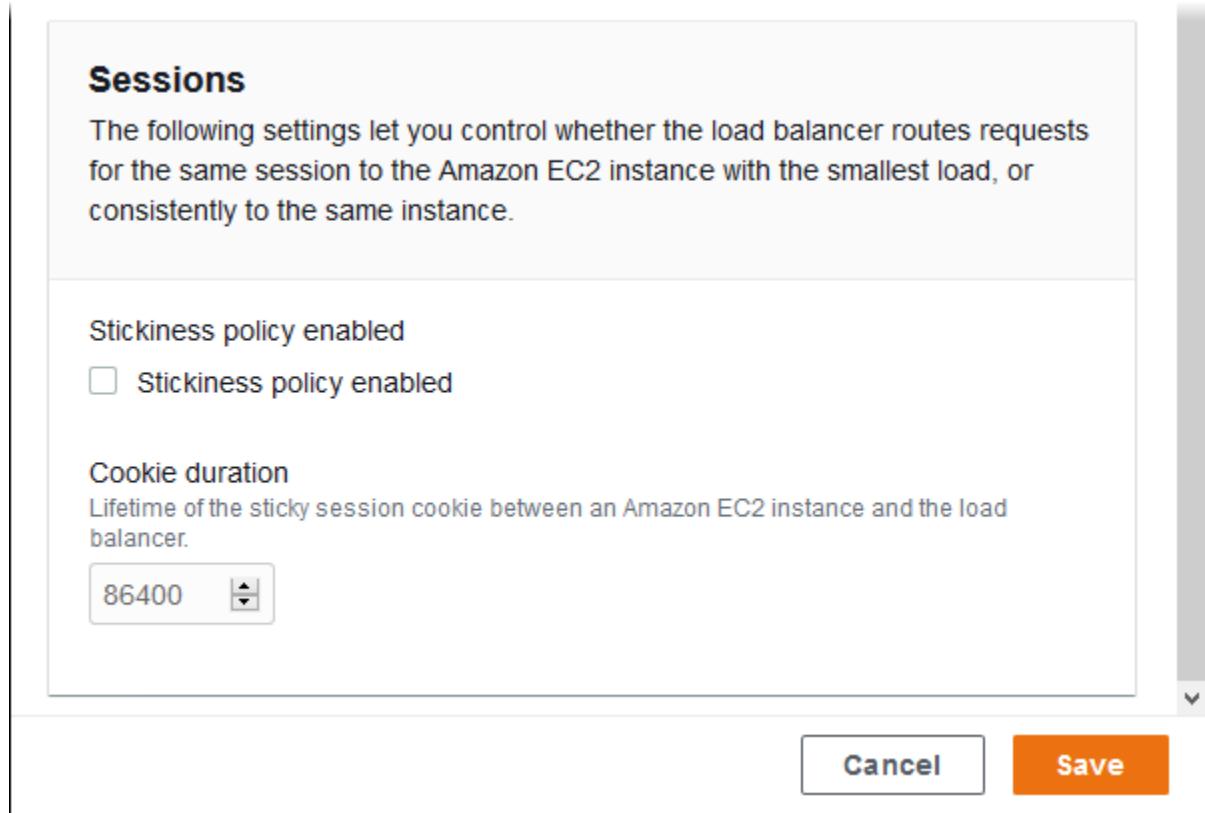
Note

Die Zustandsprüfung für Elastic Load Balancing hat keinen Einfluss auf das Verhalten der Zustandsprüfung der Auto Scaling-Gruppe einer Umgebung. Instances, die die Elastic Load Balancing-Zustandsprüfung nicht bestehen, werden nicht automatisch durch die Amazon EC2 Auto Scaling ersetzt, es sei denn, Sie konfigurieren Amazon EC2 Auto Scaling manuell entsprechend. Details dazu finden Sie unter [Zustandsprüfungseinstellung für Auto Scaling \(p. 554\)](#).

Weitere Informationen zu Zustandsprüfungen und deren Auswirkungen auf den Gesamtstatus der Umgebung finden Sie unter [Grundlegende Zustandsberichte \(p. 821\)](#).

Sitzungen

Aktivieren oder deaktivieren Sie das Kontrollkästchen Stickiness policy enabled (Sticky-Richtlinie aktiviert) zum Aktivieren oder Deaktivieren von Sticky Sessions. Verwenden Sie Cookie duration (Cookie-Dauer) zum Konfigurieren der Dauer einer Sticky Session (auf bis zu **604800** Sekunden).



Regeln

In dieser Liste können Sie benutzerdefinierte Listener-Regeln für Ihren Load Balancer angeben. Eine Regel ordnet Anforderungen, die der Listener unter einem spezifischen Pfadmuster erhält, einem Zielprozess zu. Jeder Listener kann mehrere Regeln aufweisen und Anforderungen auf verschiedenen Pfaden an verschiedene Prozesse auf Ihren Instances weiterleiten.

Regeln weisen numerische Prioritäten auf, die die Priorität ihrer Anwendung auf eingehende Anforderungen bestimmen. Für jeden Listener, den Sie hinzufügen, fügt Elastic Beanstalk eine Standardregel hinzu, die den Datenverkehr sämtlicher Listener an den Standardprozess weiterleitet. Die Standardregel weist die geringste Priorität auf. Sie wird nur angewendet, wenn für denselben Listener keine weitere Regel der

eingehenden Anforderung entspricht. Wenn Sie keine benutzerdefinierten Regeln hinzugefügt haben, ist die Liste zunächst leer. Listener-Standardregeln werden nicht angezeigt.

The screenshot shows the 'Rules' section of the AWS Elastic Beanstalk Application Load Balancer configuration. At the top, a note states: '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. The default rule has the highest 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 default process.' Below this is a table header with columns: Name, Listener port, Priority, Host headers, and Path patterns. A message below the table says: 'No additional listener rules are currently configured. Choose Add rule to add a listener rule.'

Sie können die Einstellungen einer vorhandenen Regel bearbeiten oder eine neue Regel hinzufügen. Um mit dem Bearbeiten einer Regel auf der Liste zu beginnen oder die Regel zur Liste hinzuzufügen, verwenden Sie dieselben Schritte, die für die [Liste der Listener \(p. 569\)](#) aufgelistet sind. Das Dialogfeld Listener rule (Listener-Regel) mit den folgenden Einstellungen wird geöffnet:

- Name – Der Name der Regel
- Listener port (Listener-Port) – Der Port des Listener, für den die Regel gilt
- Priority (Priorität) – Die Priorität der Regel. Je niedriger die Zahl für die Priorität, desto höher die Priorität. Prioritäten der Regeln eines Listener müssen eindeutig sein.
- Match conditions (Übereinstimmungsbedingungen) – Eine Liste von Anforderungs-URL-Bedingungen, für die die Regel gilt. Es gibt zwei Arten von Bedingungen: HostHeader (der Domänenanteil der URL) und PathPattern (der Pfadteil der URL). Sie können bis zu fünf Bedingungen hinzufügen. Jeder Bedingungswert darf bis zu 128 Zeichen enthalten, darunter auch Platzhalterzeichen.
- Process (Prozess) – Der Prozess, in dem der Load Balancer Anforderungen weiterleitet, die der Regel entsprechen

Beim Bearbeiten einer vorhandenen Regel können Sie Name und Listener port (Listener-Port) nicht ändern.

Listener rule

Name: images

Listener port: 80

Priority: 1

Match conditions:

Type: PathPattern Value: /images/* Remove

Add condition

Process: images

Cancel Add

Zugangsprotokollerfassung

Verwenden Sie diese Einstellungen, um Elastic Load Balancing so zu konfigurieren, dass Protokolle mit detaillierten Informationen über an Ihren Application Load Balancer gesendete Anfragen erfasst werden. Access Log Capture (Auf Protokollerfassung zugreifen) ist standardmäßig deaktiviert. Wenn Store logs (Protokolle speichern) aktiviert ist, speichert Elastic Load Balancing die Protokolle in dem von Ihnen konfigurierten S3-Bucket. Die Präfix-Einstellung gibt im Bucket für die Protokolle einen übergeordneten Ordner an. Elastic Load Balancing speichert die Protokolle in einem Ordner mit dem Namen `AWSLogs` unter Ihrem Präfix. Wenn Sie kein Präfix angeben, legt Elastic Load Balancing seinen Ordner auf der Stammebene des Buckets ab.

Weitere Informationen zu Zugriffsprotokollen finden Sie unter [Zugriffsprotokolle für Ihren Application Load Balancer](#).

The screenshot shows the 'Access log files' section of the AWS Elastic Beanstalk Application Load Balancer configuration. It includes a link to 'Configure Elastic Load Balancing to capture logs with detailed information about requests sent to your Load Balancer. Logs are stored in Amazon S3.' Below this, there's a 'Store logs' section with a note '(Standard Amazon S3 charges apply.)' and an 'Enabled' checkbox which is checked. Under 'S3 bucket', it says 'You must first configure bucket permissions.' with a 'Learn more' link. A dropdown menu is set to '-- Choose an Amazon S3 bucket --'. Below it, a red error message says 'Choose a bucket.'. There's also a 'Prefix' input field with a note 'Logical hierarchy in the bucket. If you don't specify a prefix, Elastic Load Balancing stores access logs at the bucket's root.' At the bottom right of the configuration panel, there's a 'Save' button.

Beispiel: Application Load Balancer mit einem sicheren Listener und zwei Prozessen

In diesem Beispiel erfordert Ihre Anwendung eine End-to-End-Verschlüsselung und einen separaten Prozess für die Verarbeitung von administrativen Anforderungen.

Um den Application Load Balancer Ihrer Umgebung so zu konfigurieren, dass er diese Anforderungen erfüllt, entfernen Sie den Standard-Listener, fügen einen HTTPS-Listener hinzu, geben an, dass der Standardprozess Port 443 auf HTTPS überwacht, und fügen einen Prozess und eine Listener-Regel für Administratordatenverkehr auf einem anderen Pfad hinzu.

So konfigurieren Sie den Load Balancer für dieses Beispiel

1. Fügen Sie einen sicheren Listener hinzu. Geben Sie in Port **443** ein. Wählen Sie in Protocol (Protokoll) die Option **HTTPS** aus. Wählen Sie in SSL certificate (SSL-Zertifikat) den ARN Ihres SSL-Zertifikats aus. Beispiel: **arn:aws:iam::123456789012:server-certificate/abc/certs/build** oder **arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678**.

Lassen Sie in Default process (Standardprozess) **default** ausgewählt.

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:certificate/12345678-12ab-34cd-56ef-12345678 ▾ C

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 ▾

Cancel Add

Sie können nun Ihre zusätzlichen Listener auf der Liste sehen.

<input type="checkbox"/>	Port	Protocol	SSL certificate
<input type="checkbox"/>	80	HTTP	--
<input type="checkbox"/>	443	Pending create	arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678

2. Deaktivieren Sie den HTTP-Listener am Standardport 80. Deaktivieren Sie für den Standard-Listener die Option Enabled (Aktiviert).

	Port	Protocol	SSL certificate
<input type="checkbox"/>	80	HTTP	--
<input type="checkbox"/>	443	Pending create	HTTPS arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678

3. Konfigurieren Sie den Standardprozess für HTTPS. Wählen Sie den Standardprozess aus und wählen Sie dann für Actions (Aktionen) die Option Edit (Bearbeiten) aus. Geben Sie in Port **443** ein. Wählen Sie in Protocol (Protokoll) die Option **HTTPS** aus.

Environment process

Name
default

Port
443

Protocol
HTTPS

4. Fügen Sie einen Administratorprozess hinzu. Geben Sie in Name (Name) **admin** ein. Geben Sie in Port **443** ein. Wählen Sie in Protocol (Protokoll) die Option **HTTPS** aus. Geben Sie in Health check (Zustandsprüfung) in Path (Pfad) **/admin** ein.

Environment process

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. Fügen Sie eine Regel für Administratordatenverkehr hinzu. Geben Sie in Name (Name) **admin** ein. Geben Sie in Listener port (Listener-Port) **443** ein. Fügen Sie in Match conditions (Übereinstimmungsbedingungen) ein PathPattern (Pfadmuster) mit dem Wert **/admin/*** hinzu. Wählen Sie in Process (Prozess) **admin** aus.

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	Remove
PathPattern	/admin/*	Remove

Add condition

Process
admin ▾

Cancel Add

The screenshot shows the 'Listener rule' configuration dialog in the AWS Elastic Beanstalk console. It includes fields for Name (admin), Listener port (443), Priority (1), Match conditions (PathPattern /admin/*), Process (admin), and buttons for Cancel and Add.

Application Load Balancer mit der EB CLI konfigurieren

Sie werden von der EB CLI aufgefordert, einen Load Balancer-Typ auszuwählen, wenn Sie `eb create` (p. 1053) ausführen.

```
$ 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):
```

Sie können auch einen Load Balancer-Typ mit der `--elb-type`-Option angeben.

```
$ eb create test-env --elb-type application
```

Application Load Balancer-Namespaces

Einstellungen im Zusammenhang mit Application Load Balancers finden Sie in den folgenden Namespaces:

- [aws:elasticbeanstalk:environment \(p. 686\)](#) – Wählen Sie den Load Balancer-Typ für die Umgebung aus. Der Wert für einen Application Load Balancer ist `application`.

Sie können diese Option nicht in Konfigurationsdateien ([.Ebextensions \(p. 722\)](#)) festlegen.

- [aws:elbv2:loadbalancer \(p. 709\)](#) – Konfigurieren Sie Zugriffsprotokolle und andere Einstellungen, die für den Application Load Balancer als Ganzes gelten.
- [aws:elbv2:listener \(p. 706\)](#) – Konfigurieren Sie Listener für den Application Load Balancer. Diese Einstellungen entsprechen den Einstellungen in `aws:elb:listener` für Classic Load Balancer.
- [aws:elbv2:listenerrule \(p. 708\)](#) – Konfigurieren Sie Regeln, die den Datenverkehr an verschiedene Prozesse weiterleiten, je nach Anfragepfad. Die Regeln gelten nur für Application Load Balancer.
- [aws:elasticbeanstalk:environment:process \(p. 687\)](#) – Konfigurieren Sie Zustandsprüfungen und geben Sie den Port und das Protokoll für die Prozesse an, die auf den Instances Ihrer Umgebung ausgeführt werden. Die Port- und Protokolleinstellungen entsprechen den Instance-Port und Instance-Protokoll-Einstellungen in `aws:elb:listener` für einen Listener auf einem Classic Load Balancer. Zustandsprüfungseinstellungen entsprechen den Einstellungen in den Namespaces `aws:elb:healthcheck` und `aws:elasticbeanstalk:application`.

Example .ebextensions/alb-access-logs.config

Die folgende Konfigurationsdatei ermöglicht Zugriffsprotokoll-Upserts für eine Umgebung mit einem Application Load Balancer.

```
option_settings:  
  aws:elbv2:loadbalancer:  
    AccessLogsS3Bucket: DOC-EXAMPLE-BUCKET  
    AccessLogsS3Enabled: 'true'  
    AccessLogsS3Prefix: beanstalk-alb
```

Example .ebextensions/alb-default-process.config

Die folgende Konfigurationsdatei ändert Zustandsprüfungs- und "Sticky"-Einstellungen auf dem Standardprozess.

```
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-secure-listener.config

Die folgende Konfigurationsdatei fügt einen sicheren Listener und einen entsprechenden Prozess auf Port 443 hinzu.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
    Protocol: HTTPS
    SSLCertificateArns: arn:aws:acm:us-east-2:123456789012:certificate/21324896-Ofa4-412b-
bf6f-f362d6eb6dd7
    aws:elasticbeanstalk:environment:process:https:
      Port: '443'
      Protocol: HTTPS
```

Example .ebextensions/alb-admin-rule.config

Die folgende Konfigurationsdatei fügt einen sicheren Listener mit einer Regel hinzu, die Datenverkehr mit einem Anfragepfad von /admin an einen Prozess namens admin, der auf Port 4443 überwacht.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
    Protocol: HTTPS
    Rules: admin
    SSLCertificateArns: arn:aws:acm:us-east-2:123456789012:certificate/21324896-Ofa4-412b-
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
```

Gemeinsamen Application Load Balancer konfigurieren

Wenn Sie [enable load balancing \(Load Balancing aktivieren\)](#) (p. 514), ist in der AWS Elastic Beanstalk-Umgebung ein Elastic Load Balancing-Load Balancer vorhanden, mit dem der Datenverkehr auf die Instances in der Umgebung verteilt wird. Elastic Load Balancing unterstützt verschiedene Load Balancer-Typen. Weitere Informationen dazu finden Sie im [Elastic Load Balancing-Benutzerhandbuch](#). Elastic Beanstalk kann einen Load Balancer für Sie erstellen oder Sie einen von Ihnen erstellten freigegebenen Load Balancer angeben lassen.

In diesem Thema wird die Konfiguration eines freigegebenen [Application Load Balancer](#) beschrieben, das Sie erstellen und mit Ihrer Umgebung verknüpfen. Weitere Informationen finden Sie auch unter [the section called “Application Load Balancer” \(p. 566\)](#). Weitere Informationen zur Konfiguration aller Load Balancer-Typen, die Elastic Beanstalk unterstützt, finden Sie unter [Load Balancer Ihrer AWS Elastic Beanstalk-Umgebung \(p. 554\)](#).

Note

Sie können den Typ des Load Balancer, den Ihre Umgebung verwendet, nur während der Erstellung der Umgebung wählen. Sie können Einstellungen ändern, um das Verhalten des Load Balancer Ihrer laufenden Umgebung zu verwalten, Sie können allerdings seinen Typ nicht ändern. Sie können auch nicht von einem dedizierten zu einem freigegebenen Load Balancer wechseln und umgekehrt.

Einführung

Ein gemeinsamer Load Balancer ist ein Load Balancer, den Sie mit dem Amazon Elastic Compute Cloud (Amazon EC2)-Service selbst erstellen und verwalten und dann in mehreren Elastic Beanstalk-Umgebungen einsetzen.

Wenn Sie eine Umgebung mit Load-Balancing und Skalierung erstellen und sich für die Verwendung eines Application Load Balancers entscheiden, erstellt Elastic Beanstalk standardmäßig einen Load Balancer für Ihre Umgebung. Was ein Application Load Balancer ist und wie er in einer Elastic Beanstalk-Umgebung funktioniert, erfahren Sie unter [Einführung \(p. 566\)](#) zur Konfiguration eines Application Load Balancers für Elastic Beanstalk.

In einigen Situationen möchten Sie möglicherweise die Kosten für mehrere dedizierte Load Balancer sparen. Dies kann bei mehreren Umgebungen hilfreich sein, z. B. wenn Ihre Anwendung eine Suite von Mikroservices statt ein monolithischer Service ist. In solchen Fällen können Sie einen freigegebenen Load Balancer verwenden.

Um einen freigegebenen Load Balancer zu verwenden, erstellen Sie ihn zuerst in Amazon EC2 und fügen Sie einen oder mehrere Listener hinzu. Während der Erstellung einer Elastic Beanstalk-Umgebung stellen Sie dann den Load Balancer bereit und wählen einen Listener-Port aus. Elastic Beanstalk ordnet den Listener dem Standardprozess in Ihrer Umgebung zu. Sie können benutzerdefinierte Listener-Regeln hinzufügen, um den Datenverkehr von bestimmten Host-Headern und -pfaden zu anderen Umgebungsprozessen weiterzuleiten.

Elastic Beanstalk fügt dem gemeinsamen Load Balancer ein Tag hinzu. Der Tag-Name lautet `elasticbeanstalk:shared-elb-environment-count`, und sein Wert ist die Anzahl der Umgebungen, für die dieser Load Balancer freigegeben ist.

Die Verwendung eines freigegebenen Load Balancer unterscheidet sich von der Verwendung eines dedizierten Load Balancer auf verschiedene Arten

Regarding	Dedizierter Application Load Balancer	Geteile Application Load Balancer
Verwaltung	Elastic Beanstalk erstellt und verwaltet den Load Balancer, Listener, Listener-Regeln und Prozesse (Zielgruppen). Elastic Beanstalk entfernt sie auch, wenn Sie Ihre Umgebung terminieren. Elastic Beanstalk kann die Erfassung des Zugriffsprotokolls des Load Balancers festlegen, wenn Sie diese Option auswählen.	Sie erstellen und verwalten den Load Balancer und die Listener außerhalb von Elastic Beanstalk. Elastic Beanstalk erstellt und verwaltet eine Standardregel und einen Standardprozess, und Sie können Regeln und Prozesse hinzufügen. Elastic Beanstalk entfernt die Listener-Regeln und -Prozesse, die während der Erstellung der Umgebung hinzugefügt wurden.
Listener-Regeln	Elastic Beanstalk erstellt eine Standardregel für jeden Listener, um den gesamten Datenverkehr an den Standardprozess des Listeners weiterzuleiten.	Elastic Beanstalk ordnet eine Standardregel nur einem Port 80-Listener zu, sofern einer vorhanden ist. Wenn Sie einen anderen Standard-Listener-Port wählen, müssen Sie ihm die Standardregel zuordnen (die Elastic Beanstalk-Konsole und die EB CLI erledigen dies für Sie).

Regarding	Dedizierter Application Load Balancer	Geteile Application Load Balancer
		<p>Um Konflikte mit Listener-Regelkonflikten in Umgebungen zu lösen, die den freigegebenen Load Balancer nutzen, fügt Elastic Beanstalk der Listener-Regel den CNAME der Umgebung als Host-Header-Bedingung hinzu.</p> <p>Elastic Beanstalk behandelt Regelprioritätseinstellungen als relativ über Umgebungen hinweg, die den freigegebenen Load Balancer nutzen, und ordnet sie bei der Erstellung absoluten Prioritäten zu.</p>
Sicherheitsgruppe	Elastic Beanstalk erstellt eine Standardsicherheitsgruppe und fügt an den Load Balancer an.	<p>Sie können eine oder mehrere Sicherheitsgruppen für den Load Balancer konfigurieren. Wenn dies nicht der Fall ist, prüft Elastic Beanstalk, ob eine vorhandene Sicherheitsgruppe, die Elastic Beanstalk verwaltet, bereits an den Load Balancer angehängt ist. Falls nicht, erstellt Elastic Beanstalk eine Sicherheitsgruppe und hängt sie an den Load Balancer an. Elastic Beanstalk löscht diese Sicherheitsgruppe, wenn die letzte Umgebung, die den gemeinsamen Load Balancer nutzt, terminiert wird.</p>
Aktualisierung	<p>Sie können Ihren Application Load Balancer nach der Erstellung der Umgebung aktualisieren. Sie können Listener, Listener-Regeln und Prozesse bearbeiten. Sie können die Zugriffsprotokollerfassung des Load Balancer konfigurieren.</p>	<p>Sie können Elastic Beanstalk nicht zur Konfiguration der Zugriffsprotokollierung in Ihrem Application Load Balancer verwenden und Sie können die Listener und Listener-Regeln nach der Erstellung der Umgebung nicht aktualisieren. Sie können nur Prozesse (Zielgruppen) aktualisieren. Verwenden Sie Amazon EC2 zum Konfigurieren der Zugriffsprotokollerfassung und zum Aktualisieren von Listenern und Listener-Regeln.</p>

Gemeinsamen Application Load Balancer mit der Elastic Beanstalk-Konsole konfigurieren

Sie können die Elastic Beanstalk-Konsole verwenden, um während der Erstellung der Umgebung einen gemeinsamen Application Load Balancer zu konfigurieren. Sie können einen der freigabefähigen Load Balancer Ihres Kontos für die Verwendung in der Umgebung auswählen, den Standard-Listener-Port auswählen und zusätzliche Prozesse und Listener-Regeln konfigurieren.

Sie können Ihre gemeinsame Application Load Balancer-Konfiguration nicht mehr in der Application Load Balancer-Konsole bearbeiten, nachdem Ihre Umgebung erstellt wurde. Verwenden Sie Amazon EC2, um Listener, Listener-Regeln, Prozesse (Zielgruppen) und Zugriffsprotokollerfassung zu konfigurieren.

So konfigurieren Sie einen Application Load Balancer in der Elastic Beanstalk-Konsole während der Erstellung der Umgebung:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Environments (Umgebungen).
3. Wählen Sie [Create a new environment \(Neue Umgebung erstellen\)](#) (p. 437), um mit der Erstellung Ihrer Umgebung zu beginnen.

4. Wählen Sie auf der Hauptseite des Assistenten zuerst Configure more options (Weitere Optionen konfigurieren) und dann Create environment (Umgebung erstellen) aus.
5. Wählen Sie die Konfigurationsvoreinstellung High availability (Hohe Verfügbarkeit) aus.
Alternativ können Sie in der Konfigurationskategorie Capacity (Kapazität) einen Umgebungstyp mit Load balanced (Lastenverteilung) konfigurieren. Details dazu finden Sie unter [Kapazität \(p. 445\)](#).
6. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.
7. Wählen Sie die Option Application Load Balancer aus, falls sie nicht bereits ausgewählt ist. Wählen Sie dann die Option Shared (Gemeinsam) aus.

Elastic Beanstalk > Applications > getting-started-app

Modify load balancer

Load balancer type

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.

Dedicated

Use a load balancer that Elastic Beanstalk creates exclusively for this environment.

Shared

Use a load balancer that someone in your account created and can be shared among multiple Elastic Beanstalk environments.

8. Nehmen Sie alle Konfigurationsänderungen am gemeinsamen Application Load Balancer vor, die für Ihre Umgebung erforderlich sind.
9. Wählen Sie Save (Speichern) aus und nehmen Sie dann sämtliche weiteren Konfigurationsänderungen vor, die Ihre Umgebung erfordert.
10. Wählen Sie Create environment (Umgebung erstellen) aus.

Einstellungen für gemeinsame Application Load Balancer

- [Geteile Application Load Balancer \(p. 586\)](#)
- [Prozesse \(p. 587\)](#)
- [Regeln \(p. 591\)](#)

Geteile Application Load Balancer

In diesem Abschnitt wählen Sie einen gemeinsamen Application Load Balancer für Ihre Umgebung aus und konfigurieren das Standard-Datenverkehrs-Routing.

Bevor Sie hier einen gemeinsamen Application Load Balancer konfigurieren können, verwenden Sie Amazon EC2, um mindestens einen gemeinsamen Application Load Balancer mit mindestens einem Listener in Ihrem Konto zu definieren. Falls Sie dies noch nicht getan haben, können Sie Manage load balancers (Load Balancer verwalten) auswählen. Elastic Beanstalk öffnet die Amazon EC2-Konsole in einer neuen Browser-Registerkarte.

Wenn Sie mit der Konfiguration freigegebener Load Balancer außerhalb von Elastic Beanstalk fertig sind, konfigurieren Sie die folgenden Einstellungen in diesem Konsolenabschnitt:

- Load Balancer ARN – Der freigegebene Load Balancer, der in dieser Umgebung verwendet werden soll. Wählen Sie aus einer Liste von Load Balancern aus oder geben Sie den Amazon-Ressourcennamen (ARN) eines Load Balancers ein.
- Default listener port (Standard-Listener-Port) – Ein Listener-Port, den der freigegebene Load Balancer überwacht. Wählen Sie aus einer Liste vorhandener Listener-Ports aus. Der Datenverkehr von diesem Listener mit dem CNAME der Umgebung im Host-Header wird an einen Standardprozess in dieser Umgebung weitergeleitet.

The screenshot shows the 'Shared Application Load Balancer' configuration page. It includes fields for 'Load balancer ARN' (containing 'arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/example-sh') and 'Default listener' (set to '80 (HTTP)'). A 'Manage load balancers' button is also visible.

Prozesse

Verwenden Sie diese Liste, um Prozesse für Ihren freigegebenen Load Balancer anzugeben. Ein Prozess ist ein Ziel für Listener, an das Datenverkehr weitergeleitet wird. Anfangs zeigt die Liste den Standardprozess an, der Datenverkehr vom Standard-Listener empfängt.

	Name	Port	Protocol	HTTP code	Health check path
<input type="checkbox"/>	default	80	HTTP		/

So konfigurieren Sie einen vorhandenen Prozess

1. Aktivieren Sie das Kontrollkästchen neben seinem Tabelleneintrag und wählen Sie dann Actions (Aktionen), Edit (Bearbeiten) aus.
2. Verwenden Sie das Dialogfeld Environment process (Umgebungsprozess), um Einstellungen zu bearbeiten, und wählen Sie dann Save (Speichern).

So fügen Sie einen Prozess hinzu

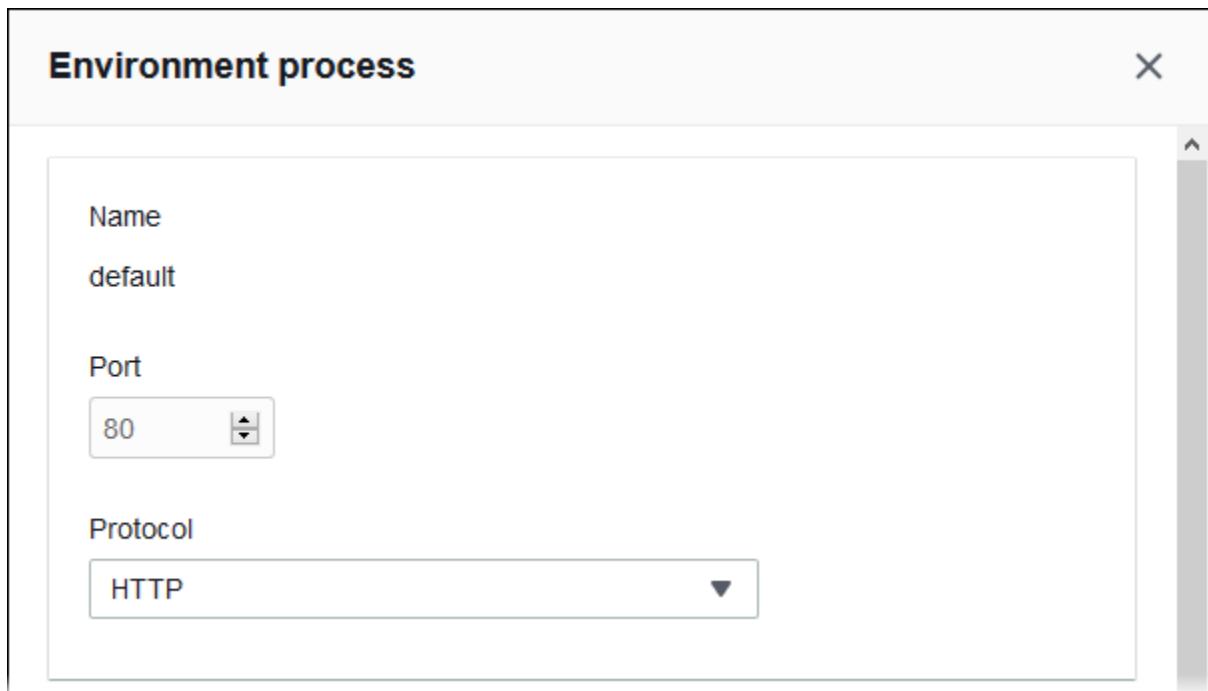
1. Wählen Sie Add process (Prozess hinzufügen).
2. Konfigurieren Sie im Dialogfeld Environment process (Umgebungsprozess) die gewünschten Einstellungen, und wählen Sie dann Add (Hinzufügen).

Prozesseinstellungen im Dialogfeld für die Umgebung des Application Load Balancers

- [Definition \(p. 571\)](#)
- [Zustandsprüfung \(p. 589\)](#)
- [Sitzungen \(p. 591\)](#)

Definition

Verwenden Sie diese Einstellungen zum Definieren des Prozesses: Name, Port und Protocol (Protokoll) für die Überwachung auf Anforderungen.



Zustandsprüfung

Verwenden Sie die folgenden Einstellungen zum Konfigurieren von Zustandsprüfungen für den Prozess:

- HTTP code (HTTP-Code) – Der HTTP-Statuscode, der einen funktionsfähigen Prozess bestimmt
- Path (Pfad) – Der Anforderungspfad der Zustandsprüfung für den Prozess
- Timeout – Die Wartezeit in Sekunden, bis eine Antwort von der Zustandsprüfung eingeht
- Interval (Intervall) – Der Zeitraum in Sekunden zwischen Zustandsprüfungen einer einzelnen Instance
Der Intervall-Wert muss größer als der Timeout-Wert sein.
- Unhealthy threshold, Healthy threshold – Die Anzahl der Zustandsprüfungen, die nicht bestanden bzw. bestanden sein müssen, bevor Elastic Load Balancing den Zustand der Instance ändert.
- Deregistration delay (Verzögerung der Registrierungsaufhebung) – Die Zeit in Sekunden, die gewartet werden muss, bis aktive Anforderungen abgeschlossen sind, ehe eine Instance abgemeldet wird

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.

 5 seconds

Interval

Amount of time between health checks of an individual instance. The interval must be greater than the timeout.

 15 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

Deregistration delay

Amount of time to wait for active requests to complete before deregistering.

 20 seconds

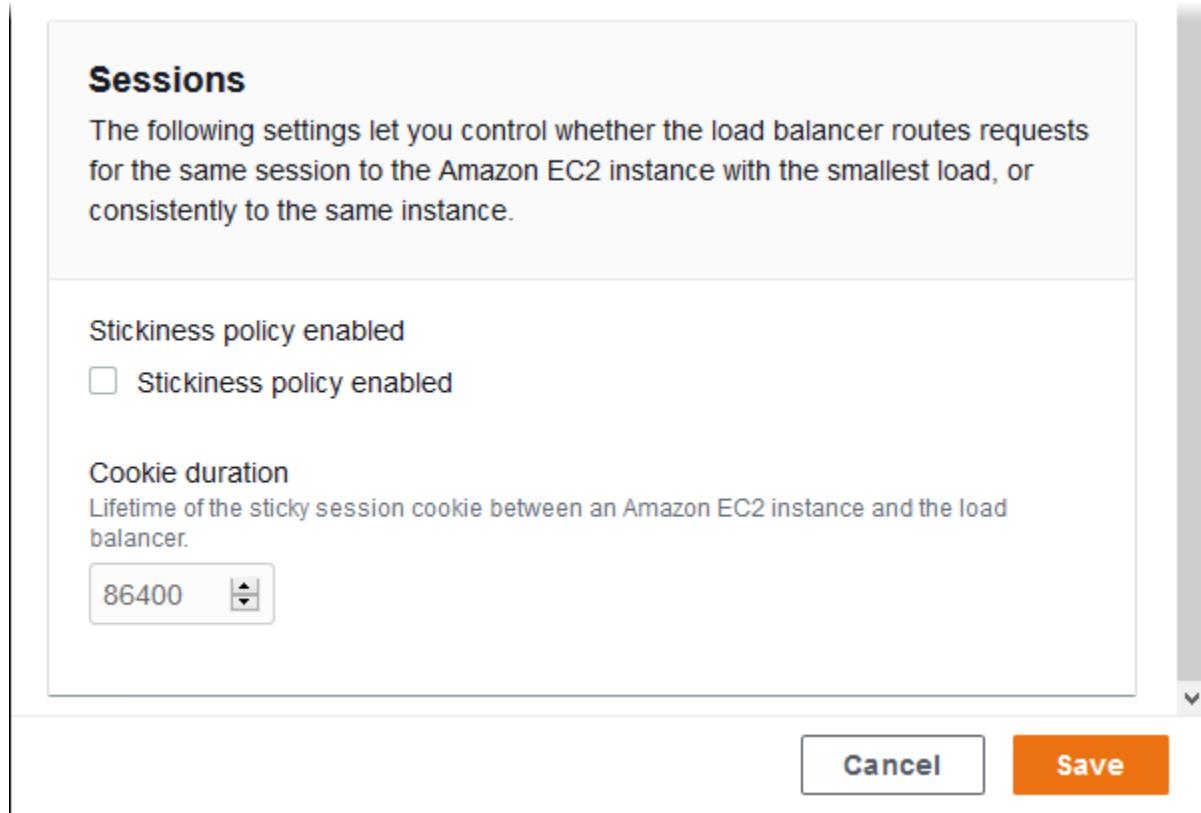
Note

Die Zustandsprüfung für Elastic Load Balancing hat keinen Einfluss auf das Verhalten der Zustandsprüfung der Auto Scaling-Gruppe einer Umgebung. Instances, die die Elastic Load Balancing-Zustandsprüfung nicht bestehen, werden nicht automatisch durch die Amazon EC2 Auto Scaling ersetzt, es sei denn, Sie konfigurieren Amazon EC2 Auto Scaling manuell entsprechend. Details dazu finden Sie unter [Zustandsprüfungseinstellung für Auto Scaling \(p. 554\)](#).

Weitere Informationen zu Zustandsprüfungen und deren Auswirkungen auf den Gesamtstatus der Umgebung finden Sie unter [Grundlegende Zustandsberichte \(p. 821\)](#).

Sitzungen

Aktivieren oder deaktivieren Sie das Kontrollkästchen Stickiness policy enabled (Sticky-Richtlinie aktiviert) zum Aktivieren oder Deaktivieren von Sticky Sessions. Verwenden Sie Cookie duration (Cookie-Dauer) zum Konfigurieren der Dauer einer Sticky Session (auf bis zu **604800** Sekunden).



Regeln

In dieser Liste können Sie benutzerdefinierte Listener-Regeln für Ihren freigegebenen Load Balancer angeben. Eine Regel ordnet Anforderungen, die der Listener unter einem spezifischen Pfadmuster erhält, einem Zielprozess zu. Jeder Listener kann mehrere Regeln aufweisen und Anforderungen auf verschiedenen Pfaden an verschiedene Prozesse auf Instances der verschiedenen Umgebungen, die den freigegebenen Listener nutzen, weiterleiten.

Regeln weisen numerische Prioritäten auf, die die Priorität ihrer Anwendung auf eingehende Anforderungen bestimmen. Elastic Beanstalk fügt eine Standardregel hinzu, die den gesamten Datenverkehr des Standard-Listeners an den Standardprozess Ihrer neuen Umgebung weiterleitet. Die Standardregel weist

die geringste Priorität auf. Sie wird nur angewendet, wenn für denselben Listener keine weitere Regel der eingehenden Anforderung entspricht. Wenn Sie keine benutzerdefinierten Regeln hinzugefügt haben, ist die Liste zunächst leer. Die Standardregel wird nicht angezeigt.

Rules

Your load balancer routes requests to environment processes based on rules. Rules are evaluated by priority in ascending numerical order. If your load balancer has existing rules configured, this environment's rules are adjusted to have lower priority than existing rules. You can manage listener rules for 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. The default rule has the highest priority among all rules of this environment. If a request doesn't match the conditions for any other rule, the default rule routes it 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 delete the environment, listener rules created outside of Elastic Beanstalk aren't automatically removed by Elastic Beanstalk.

Actions ▾

Name	Listener port	Priority	Host headers	Path patterns
No additional listener rules are currently configured.				
Choose Add rule to add a listener rule.				

Sie können die Einstellungen einer vorhandenen Regel bearbeiten oder eine neue Regel hinzufügen. Um mit dem Bearbeiten einer Regel auf der Liste zu beginnen oder die Regel zur Liste hinzuzufügen, verwenden Sie dieselben Schritte, die für die [Prozessliste \(p. 587\)](#) aufgelistet sind. Das Dialogfeld Listener rule (Listener-Regel) mit den folgenden Einstellungen wird geöffnet:

- Name – Der Name der Regel
- Listener port (Listener-Port) – Der Port des Listener, für den die Regel gilt
- Priority (Priorität) – Die Priorität der Regel. Je niedriger die Zahl für die Priorität, desto höher die Priorität. Prioritäten der Regeln eines Listener müssen eindeutig sein. Im Fall eines freigegebenen Load Balancer behandelt Elastic Beanstalk Regelprioritäten in freigebenden Umgebungen als relativ und ordnet sie während der Erstellung absoluten Prioritäten zu.
- Match conditions (Übereinstimmungsbedingungen) – Eine Liste von Anforderungs-URL-Bedingungen, für die die Regel gilt. Es gibt zwei Arten von Bedingungen: HostHeader (der Domänenanteil der URL) und PathPattern (der Pfadteil der URL). Eine Bedingung ist für die Subdomäne der Umgebung reserviert, und Sie können bis zu vier Bedingungen hinzufügen. Jeder Bedingungswert darf bis zu 128 Zeichen enthalten, darunter auch Platzhalterzeichen.
- Process (Prozess) – Der Prozess, in dem der Load Balancer Anforderungen weiterleitet, die der Regel entsprechen

Listener rule

Name: images

Listener port: 80

Priority: 1

Match conditions:

Type: PathPattern Value: /images/* Remove

Add condition

Process: images

Cancel Add

Beispiel: Verwenden eines gemeinsamen Application Load Balancers für eine sichere Mikro-Service-basierte Anwendung

In diesem Beispiel besteht Ihre Anwendung aus mehreren Micro-Services, die jeweils als Elastic Beanstalk-Umgebung implementiert sind. Darüber hinaus benötigen Sie End-to-End-Datenverkehrsverschlüsselung. Wir zeigen eine der Micro-Service-Umgebungen, die einen Hauptprozess für Benutzeranfragen und einen separaten Prozess für die Bearbeitung von administrativen Anforderungen hat.

Um diese Anforderungen zu erfüllen, verwenden Sie Amazon EC2, um einen Application Load Balancer zu erstellen, den Ihre Micro-Services gemeinsam nutzen können. Fügen Sie einen sicheren Listener auf Port 443 und das HTTPS-Protokoll hinzu. Fügen Sie dann dem Listener mehrere SSL-Zertifikate hinzu, eins pro Micro-Service-Domäne. Einzelheiten zur Erstellung des Application Load Balancers und des sicheren Listeners finden Sie unter [Application Load Balancer erstellen](#) und [HTTPS-Listener für Application Load Balancer erstellen](#) im Application Load Balancer-Benutzerhandbuch.

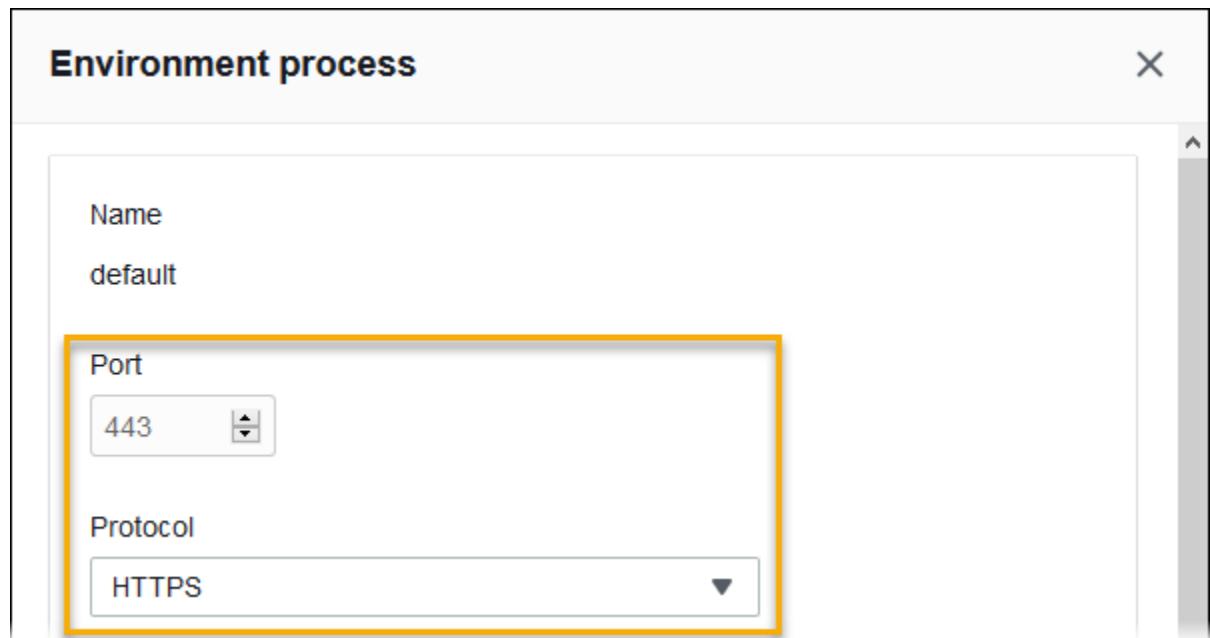
Konfigurieren Sie in Elastic Beanstalk jede Micro-Service-Umgebung für die Verwendung des gemeinsamen Application Load Balancers und setzen Sie den Standard-Listener-Port auf 443. Im Fall der bestimmten Umgebung, die wir hier demonstrieren, geben Sie an, dass der Standardprozess Port 443 auf HTTPS überwacht, und fügen Sie einen Prozess und eine Listener-Regel für den Admin-Datenverkehr auf einem anderen Pfad hinzu.

So konfigurieren Sie den freigegebenen Load Balancer für dieses Beispiel

1. Wählen Sie im Abschnitt Shared Application Load Balancer (Gemeinsamer Application Load Balancer) Ihren Load Balancer aus. Wählen Sie dann für Default listener port (Standard-Listener-Port) **443** aus. Der Listener-Port sollte bereits ausgewählt sein, wenn er der einzige Listener ist, den der Load Balancer hat.

The screenshot shows the 'Shared Application Load Balancer' configuration page. At the top, there is a button labeled 'Manage load balancers'. Below it, a field for 'Load balancer ARN' contains the value 'arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/example-sh...'. A note below the field says 'Must be an active Application Load Balancer in vpc-5732152e'. Under the 'Default listener' section, a dropdown menu shows '443 (HTTPS)' as the selected option. A note next to the dropdown says 'The default process and rule are associated with this listener.'

2. Konfigurieren Sie den Standardprozess für HTTPS. Wählen Sie den Standardprozess aus und wählen Sie dann für Actions (Aktionen) die Option Edit (Bearbeiten) aus. Geben Sie im Feld Port **443** ein. Wählen Sie in Protocol (Protokoll) die Option **HTTPS** aus.



3. Fügen Sie einen Administratorprozess hinzu. Geben Sie unter Name **admin** ein. Geben Sie im Feld Port **443** ein. Wählen Sie in Protocol (Protokoll) die Option **HTTPS** aus. Geben Sie in Health check (Zustandsprüfung) in Path (Pfad) **/admin** ein.

Environment process

Name: admin

Port: 443

Protocol: HTTPS

Health check

HTTP code: 200

Path: /admin

4. Fügen Sie eine Regel für Administratordatenverkehr hinzu. Geben Sie unter Name **admin** ein. Geben Sie in Listener port (Listener-Port) **443** ein. Fügen Sie in Match conditions (Übereinstimmungsbedingungen) ein PathPattern (Pfadmuster) mit dem Wert **/admin/*** hinzu. Wählen Sie in Process (Prozess) **admin** aus.

Listener rule

Name
admin

Listener port
443 ▾

Priority
Evaluated in ascending numerical order. Must be unique across all rules.
1 ▾

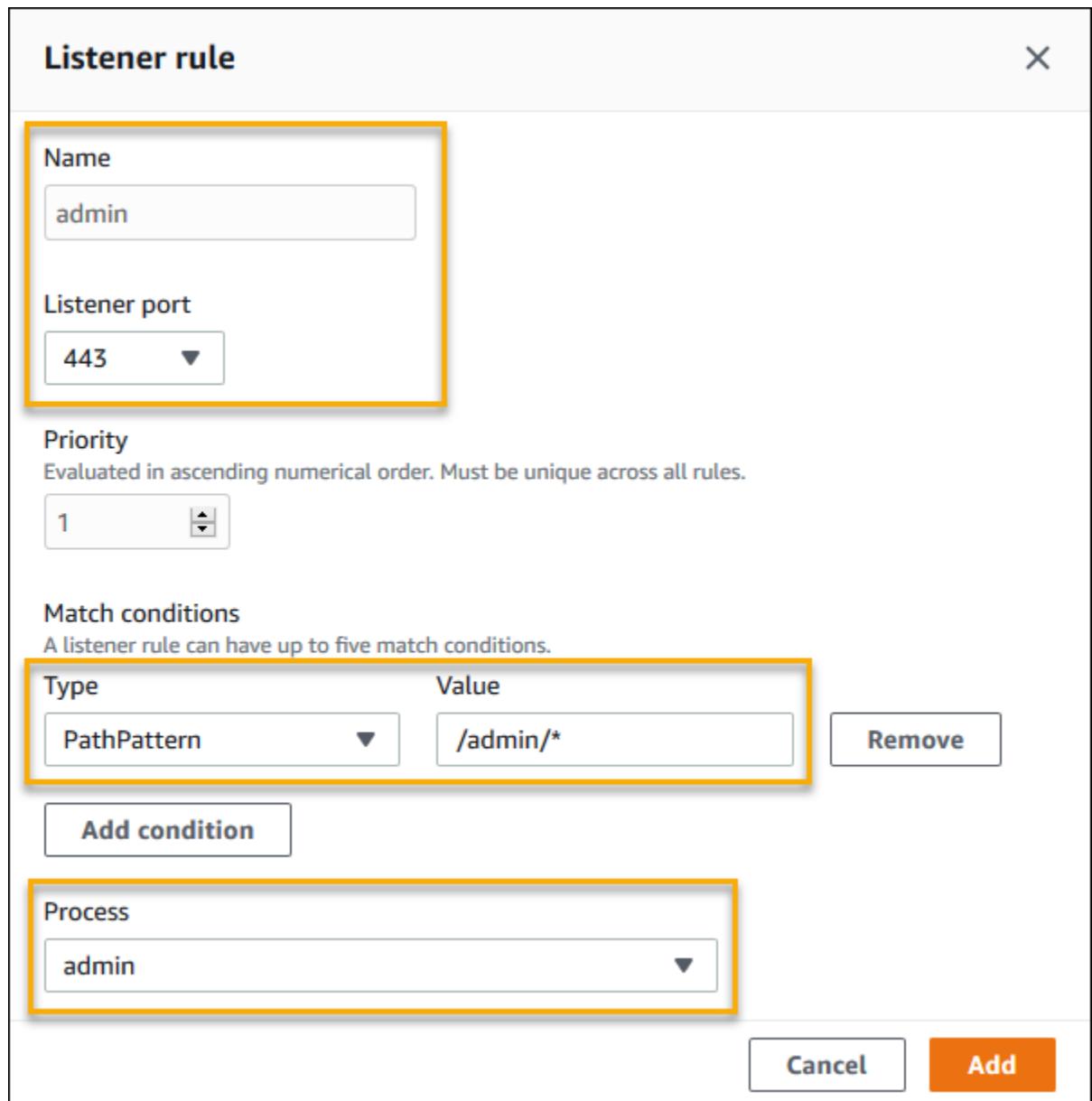
Match conditions
A listener rule can have up to five match conditions.

Type	Value	Remove
PathPattern	/admin/*	Remove

Add condition

Process
admin

Cancel Add



Gemeinsamen Application Load Balancers mit der EB CLI konfigurieren

Sie werden von der EB CLI aufgefordert, einen Load Balancer-Typ auszuwählen, wenn Sie `eb create` (p. 1053) ausführen. Wenn Sie `application` auswählen (die Standardeinstellung), und Ihr Konto über mindestens einen gemeinsamen Application Load Balancer verfügt, fragt die EB CLI Sie, ob Sie einen gemeinsamen Application Load Balancer verwenden möchten. Wenn Sie mit `y` antworten, werden Sie auch aufgefordert, den Load Balancer und den Standardport auszuwählen.

```
$ 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
```

Sie können auch einen freigegebenen Load Balancer mit Befehlsoptionen angeben.

```
$ eb create test-env --elb-type application --shared-lb MySharedALB2 --shared-lb-port 443
```

Gemeinsame Application Load Balancer-Namespaces

Einstellungen für gemeinsame Application Load Balancer finden Sie in den folgenden Namespaces:

- [aws:elasticbeanstalk:environment \(p. 686\)](#) – Wählen Sie den Load Balancer-Typ für die Umgebung aus, und teilen Sie Elastic Beanstalk mit dass Sie einen freigegebenen Load Balancer verwenden werden.

Sie können diese beiden Optionen nicht in Konfigurationsdateien ([Ebextensions \(p. 722\)](#)) festlegen.

- [aws:elbv2:loadbalancer \(p. 709\)](#) – Konfigurieren von ARN und Sicherheitsgruppen für den gemeinsamen Application Load Balancer.
- [aws:elbv2:listener \(p. 706\)](#) – Über die Auflistung von Listener-Regeln Listener des gemeinsamen Application Load Balancers den Prozessen der Umgebung zuordnen.
- [aws:elbv2:listenerrule \(p. 708\)](#) – Konfigurieren Sie Listener-Regeln, die den Datenverkehr an verschiedene Prozesse weiterleiten, je nach Anfragepfad. Die Regeln sind einzigartig für Application Load Balancer – sowohl für dedizierte als auch für gemeinsame.
- [aws:elasticbeanstalk:environment:process \(p. 687\)](#) – Konfigurieren Sie Zustandsprüfungen und geben Sie den Port und das Protokoll für die Prozesse an, die auf den Instances Ihrer Umgebung ausgeführt werden.

Example .ebextensions/application-load-balancer-shared.config

Um mit einem gemeinsamen Application Load Balancer zu beginnen, verwenden Sie die Elastic Beanstalk-Konsole, EB CLI oder API, um den Load Balancer-Typ auf `application` festzulegen und sich für die Verwendung eines gemeinsamen Load Balancers zu entscheiden. Verwenden Sie eine [Konfigurationsdatei \(p. 722\)](#), um den freigegebenen Load Balancer zu konfigurieren.

```
option_settings:
  aws:elbv2:loadbalancer:
```

```
SharedLoadBalancer: arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/  
app/MySharedALB2/e574ea4c37ad2ec8
```

Note

Sie können diese Option nur während der Erstellung der Umgebung konfigurieren.

Example .ebextensions/alb-shared-secure-listener.config

Die folgende Konfigurationsdatei wählt einen sicheren Standard-Listener auf Port 443 für den freigegebenen Load Balancer aus und legt fest, dass der Standardprozess Port 443 überwacht.

```
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

Die folgende Konfigurationsdatei baut auf dem vorherigen Beispiel auf und fügt eine Regel hinzu, die den Datenverkehr mit einem Anfragepfad von /admin an einen Prozess namens admin weiterleitet, der Port 4443 überwacht.

```
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
```

Konfigurieren eines Network Load Balancers

Wenn Sie [enable load balancing \(Load Balancing aktivieren\)](#) (p. 514), ist in der AWS Elastic Beanstalk-Umgebung ein Elastic Load Balancing-Load Balancer vorhanden, mit dem der Datenverkehr auf die Instances in der Umgebung verteilt wird. Elastic Load Balancing unterstützt verschiedene Load Balancer-Typen. Weitere Informationen dazu finden Sie im [Elastic Load Balancing-Benutzerhandbuch](#). Elastic Beanstalk kann einen Load Balancer für Sie erstellen oder Sie einen von Ihnen erstellten freigegebenen Load Balancer angeben lassen.

Dieses Thema beschreibt die Konfiguration eines [Network Load Balancers](#), den Elastic Beanstalk erstellt und Ihrer Umgebung zuweist. Weitere Informationen zur Konfiguration aller Load Balancer-Typen, die Elastic Beanstalk unterstützt, finden Sie unter [Load Balancer Ihrer AWS Elastic Beanstalk-Umgebung \(p. 554\)](#).

Note

Sie können den Typ des Load Balancer, den Ihre Umgebung verwendet, nur während der Erstellung der Umgebung wählen. Sie können Einstellungen ändern, um das Verhalten des Load Balancer Ihrer laufenden Umgebung zu verwalten, Sie können allerdings seinen Typ nicht ändern.

Einführung

Mit einem Network Load Balancer akzeptiert der standardmäßige Listener TCP-Anfragen auf Port 80 und leitet sie an die Instances in Ihrer Umgebung weiter. Sie können Zustandsprüfungsverhalten konfigurieren, den Listener-Port konfigurieren oder einen Listener zu einem anderen Port hinzufügen.

Note

Im Gegensatz zu einem Classic Load Balancer oder einem Application Load Balancer kann ein Network Load Balancer keine HTTP- oder HTTPS-Listener der Anwendungsschicht (Schicht 7) haben. Er unterstützt nur TCP-Listener auf Transportebene (Ebene 4). HTTP- und HTTPS-Datenverkehr kann über TCP an Ihre Umgebung weitergeleitet werden. Um sichere HTTPS-Verbindungen zwischen Web-Clients und Ihrer Umgebung aufzubauen, installieren Sie ein [selbstsigniertes Zertifikat \(p. 780\)](#) auf den Instances in der Umgebung und konfigurieren Sie die Instances so, dass sie auf dem entsprechenden Port (typischerweise Port 443) überwachen und HTTPS-Verbindungen beenden. Die Konfiguration variiert je nach Plattform. Anweisungen finden Sie unter [Konfigurieren Ihrer Anwendung, um HTTPS-Verbindungen auf der Instance zu beenden \(p. 785\)](#). Anschließend konfigurieren Sie Network Load Balancer so, dass ein Listener hinzugefügt wird, der einem Prozess zugeordnet wird, der auf dem entsprechenden Port überwacht.

Ein Network Load Balancer unterstützt aktive Zustandsprüfungen. Diese Prüfungen basieren auf Nachrichten an den Stamm-Pfad (/). Darüber hinaus unterstützt ein Network Load Balancer passive Zustandsprüfungen. Er erkennt automatisch fehlerhafte Backend-Instances und leitet den Datenverkehr nur an stabile Instances.

Network Load Balancer mit der Elastic Beanstalk-Konsole konfigurieren

Sie können die Elastic Beanstalk-Konsole verwenden, um die Listener und Prozesse eines Network Load Balancers während der Erstellung der Umgebung oder später (wenn Ihre Umgebung bereits ausgeführt wird) zu konfigurieren.

So konfigurieren Sie einen Network Load Balancer in der Elastic Beanstalk-Konsole während der Erstellung der Umgebung:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Environments (Umgebungen).
3. Wählen Sie [Create a new environment \(Neue Umgebung erstellen\) \(p. 437\)](#), um mit der Erstellung Ihrer Umgebung zu beginnen.
4. Wählen Sie auf der Hauptseite des Assistenten zuerst Configure more options (Weitere Optionen konfigurieren) und dann Create environment (Umgebung erstellen) aus.
5. Wählen Sie die Konfigurationsvoreinstellung High availability (Hohe Verfügbarkeit) aus.

Alternativ können Sie in der Konfigurationskategorie Capacity (Kapazität) einen Umgebungstyp mit Load balanced (Lastenverteilung) konfigurieren. Details dazu finden Sie unter [Kapazität \(p. 445\)](#).

6. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.
7. Wählen Sie die Option Network Load Balancer aus, falls sie nicht bereits ausgewählt ist.

Elastic Beanstalk > Applications > 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.

8. Nehmen Sie alle Konfigurationsänderungen am Network Load Balancer vor, die für Ihre Umgebung erforderlich sind.
9. Wählen Sie Save (Speichern) aus und nehmen Sie dann sämtliche weiteren Konfigurationsänderungen vor, die Ihre Umgebung erfordert.
10. Wählen Sie Create environment (Umgebung erstellen) aus.

So konfigurieren Sie den Network Load Balancer einer laufenden Umgebung in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.

Note

Ist in der Konfigurationskategorie Load balancer die Schaltfläche Edit (Bearbeiten) nicht vorhanden, verfügt Ihre Umgebung über keinen Load Balancer. Informationen zur Einrichtung finden Sie unter [Ändern des Umgebungstyps \(p. 514\)](#).

5. Nehmen Sie die Konfigurationsänderungen am Network Load Balancer vor, die Ihre Umgebung erfordert.
6. Wählen Sie Apply (Anwenden) aus.

Network Load Balancer-Einstellungen

- [Listener \(p. 602\)](#)
- [Prozesse \(p. 603\)](#)

Listener

Verwenden Sie diese Liste, um Listener für Ihren Load Balancer anzugeben. Jeder Listener leitet den eingehenden Client-Datenverkehr eines bestimmten Ports an einen Prozess auf Ihren Instances weiter. Anfänglich wird in der Liste der Standard-Listener angezeigt, der eingehenden Datenverkehr auf Port 80 an einen Prozess mit der Bezeichnung default (Standard) weiterleitet, der Port 80 überwacht.

The screenshot shows a table titled "Network Load Balancer" with the following data:

	Listener port	Process port	Protocol	Enabled
<input type="checkbox"/>	80	80	TCP	<input checked="" type="checkbox"/>

Actions ▾

So konfigurieren Sie einen vorhandenen Listener

1. Aktivieren Sie das Kontrollkästchen neben seinem Tabelleneintrag und wählen Sie dann Actions (Aktionen), Edit (Bearbeiten) aus.
2. Verwenden Sie das Dialogfeld Network Load Balancer listener, um die Einstellungen zu bearbeiten, und wählen Sie dann Save (Speichern) aus.

So fügen Sie einen Listener hinzu

1. Wählen Sie Add listener (Listener hinzufügen) aus.
2. Konfigurieren Sie im Dialogfeld Network Load Balancer listener die erforderlichen Einstellungen und wählen Sie dann Add (Hinzufügen) aus.

Verwenden Sie das Dialogfeld Network Load Balancer listener, um den Port zu konfigurieren, dessen Datenverkehr der Listener überwachen soll, und um den Prozess auszuwählen, an den Sie den Datenverkehr weiterleiten möchten (der durch den Port angegeben wird, den der Prozess überwacht).

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**

Prozesse

Verwenden Sie diese Liste, um Prozesse für Ihren Load Balancer anzugeben. Ein Prozess ist ein Ziel für Listener, an das Datenverkehr weitergeleitet wird. Jeder Listener leitet den eingehenden Client-Datenverkehr eines bestimmten Ports an einen Prozess auf Ihren Instances weiter. Anfänglich zeigt die Liste den Standardprozess, der eingehenden Datenverkehr auf Port 80 überwacht.

The screenshot shows a table titled 'Processes' with the following data:

Action	Process name	Process port	Interval	Healthy threshold	Unhealthy threshold
<input type="checkbox"/>	default	80	10	5	5

At the top right of the table, there is a button labeled 'Actions ▾'. Below the table, there is a large, semi-transparent gray overlay with the text 'Sie können die Einstellungen eines vorhandenen Prozesses bearbeiten oder einen neuen Prozess hinzufügen.' and 'Um mit dem Bearbeiten eines Prozesses auf der Liste zu beginnen oder Prozesse zur Liste hinzuzufügen, verwenden Sie dieselben Schritte, die für die [Liste der Listener \(p. 569\)](#) aufgelistet sind.'

Sie können die Einstellungen eines vorhandenen Prozesses bearbeiten oder einen neuen Prozess hinzufügen. Um mit dem Bearbeiten eines Prozesses auf der Liste zu beginnen oder Prozesse zur Liste hinzuzufügen, verwenden Sie dieselben Schritte, die für die [Liste der Listener \(p. 569\)](#) aufgelistet sind.

Das Dialogfeld Environment process (Umgebungsprozess) wird geöffnet.

Einstellungen des Prozess-Dialogfelds für Network Load Balancer-Umgebung

- [Definition \(p. 604\)](#)
- [Zustandsprüfung \(p. 605\)](#)

Definition

Verwenden Sie diese Einstellungen zum Definieren des Prozesses: Name und Process port (Prozess-Port) für die Überwachung auf Anforderungen.

The dialog box is titled 'Environment process'. It contains the following fields:

Name	default
Process port	80

Zustandsprüfung

Verwenden Sie die folgenden Einstellungen zum Konfigurieren von Zustandsprüfungen für den Prozess:

- Interval (Intervall) – Der Zeitraum in Sekunden zwischen Zustandsprüfungen einer einzelnen Instance
- Healthy threshold (Schwellenwert für fehlerfreien Zustand) Die Anzahl der Zustandsprüfungen, die bestanden sein müssen, bevor Elastic Load Balancing den Zustand der Instance ändert. (Für Network Load Balancer ist Unhealthy threshold (Schwellenwert für fehlerhaften Zustand) eine schreibgeschützte Einstellung, die immer gleich dem Wert des Schwellenwerts für fehlerfreien Zustand ist.)
- Deregistration delay (Verzögerung der Registrierungsaufhebung) – Die Zeit in Sekunden, die gewartet werden muss, bis aktive Anforderungen abgeschlossen sind, ehe eine Instance abgemeldet wird

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

Die Zustandsprüfung für Elastic Load Balancing hat keinen Einfluss auf das Verhalten der Zustandsprüfung der Auto Scaling-Gruppe einer Umgebung. Instances, die eine Elastic Load Balancing Zustandsprüfung nicht bestehen, werden nicht automatisch durch Amazon EC2 Auto Scaling ersetzt, es sei denn, Sie konfigurieren Amazon EC2 Auto Scaling manuell entsprechend. Details dazu finden Sie unter [Zustandsprüfungseinstellung für Auto Scaling \(p. 554\)](#).

Weitere Informationen zu Zustandsprüfungen und deren Auswirkungen auf den Gesamtstatus der Umgebung finden Sie unter [Grundlegende Zustandsberichte \(p. 821\)](#).

Beispiel: Network Load Balancer für eine Umgebung mit End-to-End-Verschlüsselung

In diesem Beispiel erfordert Ihre Anwendung eine End-to-End-Datenverkehrsverschlüsselung. Um Network Load Balancer Ihrer Umgebung so zu konfigurieren, dass er diese Anforderungen erfüllt, konfigurieren Sie den Standardprozess so, dass er Port 443 überwacht, fügen Sie einen Listener zu Port 443 hinzu, der den Datenverkehr an den Standardprozess weiterleitet, und deaktivieren Sie den Standard-Listener.

So konfigurieren Sie den Load Balancer für dieses Beispiel

1. Konfigurieren Sie den Standardprozess. Wählen Sie den Standardprozess aus und wählen Sie dann für Actions (Aktionen) die Option Edit (Bearbeiten) aus. Für Process port (Prozess-Port) geben Sie 443 ein.



2. Fügen Sie einen Listener auf Port 443 hinzu. Fügen Sie einen neuen Listener hinzu. Geben Sie für Listener port (Listener-Port) den Wert 443 ein. Stellen Sie bei Process port (Prozess-Port) sicher, dass 443 ausgewählt ist.

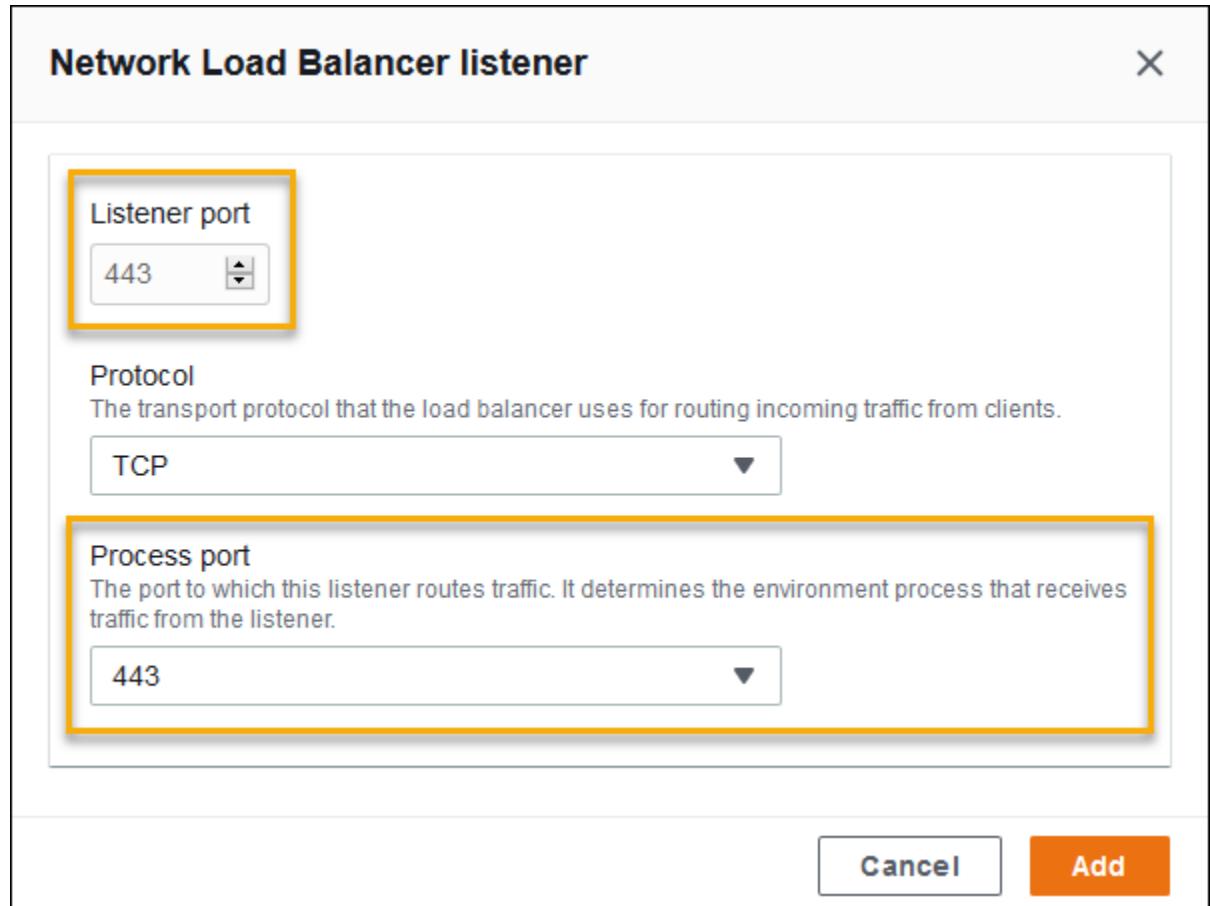
Network Load Balancer listener

Listener port
443 ▾

Protocol
TCP ▾

Process port
The port to which this listener routes traffic. It determines the environment process that receives traffic from the listener.
443 ▾

Cancel Add



Sie können nun Ihre zusätzlichen Listener auf der Liste sehen.

<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"/>

3. Deaktivieren Sie den Listener am Standardport 80. Deaktivieren Sie für den Standard-Listener die Option Enabled (Aktiviert).

<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"/>

Network Load Balancer mit der EB CLI konfigurieren

Sie werden von der EB CLI aufgefordert, einen Load Balancer-Typ auszuwählen, wenn Sie [eb create \(p. 1053\)](#) ausführen.

```
$ 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
```

Sie können auch einen Load Balancer-Typ mit der `--elb-type`-Option angeben.

```
$ eb create test-env --elb-type network
```

Network Load Balancer-Namespace

Sie finden Einstellungen im Zusammenhang mit Network Load Balancers in den folgenden Namespaces:

- [aws:elasticbeanstalk:environment \(p. 686\)](#) – Wählen Sie den Load Balancer-Typ für die Umgebung aus. Der Wert für einen Network Load Balancer ist `network`.
- [aws:elbv2:listener \(p. 706\)](#) – Konfigurieren Sie Listener für den Network Load Balancer. Diese Einstellungen entsprechen den Einstellungen in `aws:elb:listener` für Classic Load Balancer.
- [aws:elasticbeanstalk:environment:process \(p. 687\)](#) – Konfigurieren Sie Zustandsprüfungen und geben Sie den Port und das Protokoll für die Prozesse an, die auf den Instances Ihrer Umgebung ausgeführt werden. Die Port- und Protokolleinstellungen entsprechen den Instance-Port und Instance-Protokoll-Einstellungen in `aws:elb:listener` für einen Listener auf einem Classic Load Balancer. Zustandsprüfungseinstellungen entsprechen den Einstellungen in den Namespaces `aws:elb:healthcheck` und `aws:elasticbeanstalk:application`.

Example .ebextensions/network-load-balancer.config

Für die ersten Schritte mit einem Network Load Balancer verwenden Sie eine [Konfigurationsdatei \(p. 722\)](#), um den Load Balancer-Typ auf `network` festzulegen.

```
option_settings:
  aws:elasticbeanstalk:environment:
    LoadBalancerType: network
```

Note

Sie können den Load Balancer-Typ nur während der Umgebungserstellung festlegen.

Example .ebextensions/nlb-default-process.config

Die folgende Konfigurationsdatei ändert Zustandsprüfungseinstellungen auf dem Standardprozess.

```
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

Die folgende Konfigurationsdatei fügt einen Listener für sicheren Datenverkehr auf Port 443 sowie einen entsprechenden Zielprozess hinzu, der Port 443 überwacht.

```
option_settings:  
  aws:elbv2:listener:443:  
    DefaultProcess: https  
    ListenerEnabled: 'true'  
  aws:elasticbeanstalk:environment:process:https:  
    Port: '443'
```

Die Option `DefaultProcess` hat diesen Namen wegen Application Load Balancers, die Nicht-Standard-Listener auf demselben Port für Datenverkehr auf bestimmten Pfaden haben können (siehe [Application Load Balancer \(p. 566\)](#) für weitere Informationen). Bei einem Network Load Balancer gibt die Option den einzigen Zielprozess für diesen Listener an.

In diesem Beispiel haben wir den Prozess `https` genannt, da er sicheren (HTTPS) Datenverkehr überwacht. Der Listener sendet den Datenverkehr über das TCP-Protokoll an den Prozess auf dem festgelegten Port, da ein Network Load Balancer nur mit TCP funktioniert. Das ist in Ordnung, da der Netzwerksdatenverkehr für HTTP und HTTPS zusätzlich zu TCP implementiert ist.

Konfigurieren von Zugriffsprotokollen

Mithilfe der [Konfigurationsdateien \(p. 722\)](#) können Sie den Load Balancer in der Umgebung für den Upload der Zugriffsprotokolle in einen Amazon S3-Bucket konfigurieren. Weitere Anleitungen finden Sie in den folgenden Beispielkonfigurationsdateien mit GitHub:

- [loadbalancer-accesslogs-existingbucket.config](#) – Konfigurieren Sie den Load Balancer für den Upload der Zugriffsprotokolle in einen vorhandenen Amazon S3-Bucket.
- [loadbalancer-accesslogs-newbucket.config](#) – Konfigurieren Sie den Load Balancer für den Upload der Zugriffsprotokolle in einen neuen Bucket.

Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung

Elastic Beanstalk ermöglicht eine Integration mit [Amazon Relational Database Service \(Amazon RDS\)](#), um das Hinzufügen einer Datenbank-Instance zu Ihrer Elastic Beanstalk-Umgebung zu erleichtern. Sie können mit Elastic Beanstalk eine MySQL-, PostgreSQL-, Oracle- oder SQL Server-Datenbank während oder nach der Umgebungserstellung zu Ihrer Umgebung hinzufügen. Wenn Sie der Umgebung eine Datenbank-Instance hinzufügen, stellt Elastic Beanstalk die Verbindungsinformationen für Ihre Anwendung bereit, indem Umgebungseigenschaften für den Datenbank-Hostnamen, den Port, Benutzernamen und das Passwort sowie den Datenbanknamen festgelegt werden.

Eine Datenbank-Instance, die Teil Ihrer Umgebung ist, verfügt über eine Bindung an den Lebenszyklus der Umgebung. Einmal hinzugefügt, kann sie nicht aus der Umgebung entfernt werden. Wenn Sie die Umgebung beenden, wird die Datenbank-Instance auch beendet. Sie können Elastic Beanstalk zum Speichern eines Snapshots der Datenbank konfigurieren, wenn Sie Ihre Umgebung beenden, und eine Datenbank aus einem Snapshot wiederherstellen, wenn Sie eine DB-Instance zu einer Umgebung hinzufügen. Für das Speichern von Datenbank-Snapshots fallen möglicherweise Kosten an. Weitere Informationen finden Sie im Abschnitt Sicherungsspeicher unter [Amazon RDS – Preise](#).

In einer Produktionsumgebung können Sie [eine Datenbank-Instance außerhalb Ihrer Umgebung starten \(p. 973\)](#) und Ihre Anwendung so konfigurieren, dass sie außerhalb der Funktionalität von Elastic Beanstalk verbunden wird. Die Verwendung einer Datenbank-Instance, die außerhalb Ihrer Umgebung liegt, erfordert eine zusätzliche Sicherheitsgruppe und Konfiguration der Verbindungszeichenfolge. Allerdings können Sie auch aus mehreren Umgebungen eine Verbindung zu der Datenbank herstellen, Datentypen verwenden, die nicht von integrierten Datenbanken unterstützt werden, Blau/Grün-Bereitstellungen durchführen und Ihre Umgebung beenden, ohne dass sich dies auf die Datenbank-Instance auswirkt.

Abschnitte

- [Hinzufügen einer RDS-DB-Instance zur Umgebung \(p. 610\)](#)
- [Verbinden mit der Datenbank \(p. 613\)](#)
- [Konfigurieren einer integrierten RDS-DB-Instance mithilfe der Konsole \(p. 613\)](#)
- [Konfigurieren einer integrierten RDS-DB-Instance mithilfe von Konfigurationsdateien \(p. 614\)](#)

Hinzufügen einer RDS-DB-Instance zur Umgebung

Sie können der Umgebung mit der Elastic Beanstalk-Konsole eine DB-Instance hinzufügen.

So fügen Sie eine DB-Instance zu Ihrer Umgebung hinzu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).
5. Wählen Sie eine DB-Engine aus und geben Sie Benutzernamen und Passwort ein.
6. Wählen Sie Apply (Anwenden) aus.

Sie können die folgenden Optionen konfigurieren:

- Snapshot – Wählen Sie einen Datenbank-Snapshot aus. Elastic Beanstalk stellt den Snapshot wieder her und fügt ihn der Umgebung hinzu. Der Standardwert ist None (Keiner). Damit haben Sie die Möglichkeit, mithilfe der anderen Einstellungen auf dieser Seite eine neue Datenbank zu konfigurieren.
- Engine – Wählen Sie eine Datenbank-Engine aus.
- Engine version (Engine-Version) – Wählen Sie eine bestimmte Version der Datenbank-Engine aus.
- Instance class (Instance-Klasse) – Wählen Sie eine DB-Instance-Klasse aus. Weitere Informationen über die DB-Instance-Klassen finden Sie unter <https://aws.amazon.com/rds/>.
- Storage (Speicher) – Wählen Sie die Speicherkapazität für Ihre Datenbank aus. Sie können zugewiesenen Speicher zu einem späteren Zeitpunkt erhöhen, jedoch nicht verringern. Weitere Informationen zur Speicherzuweisung finden Sie unter [Funktionen](#).
- Username (Benutzername) – Geben Sie einen beliebigen Benutzernamen mit alphanumerischen Zeichen ein.
- Password (Passwort) – Geben Sie ein beliebiges Passwort aus 8–16 druckbaren ASCII-Zeichen (ohne /, \ und @) ein.

- Retention (Aufbewahrung) – Wählen Sie Create snapshot (Snapshot erstellen) aus, um einen Snapshot der Datenbank zu erstellen, wenn Sie Ihre Umgebung beenden.
- Availability (Verfügbarkeit) – Wählen Sie High (Multi-AZ) (Hoch (Multi-AZ)) aus, um ein Warm-Backup in einer zweiten Availability Zone für hohe Verfügbarkeit durchzuführen.

Note

Elastic Beanstalk erstellt einen Master-Benutzer für die Datenbank unter Verwendung des Benutzernamens und des Passworts, die Sie festgelegt haben. Weitere Informationen zum Master-Benutzer und dessen Berechtigungen finden Sie unter [Berechtigungen von Hauptbenutzerkonten](#).

Modify database

Add an Amazon RDS SQL database to your environment for development and testing. AWS Elastic Beanstalk provides information to your instances by setting environment properties for the database hostname, username, password, 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

5.6.41



Instance class

db.t2.micro



Storage

Choose a number between 5 GB and 1024 GB.

5



Username

Password

612

Retention

Create snapshot



When you terminate your environment, your database instance is also terminated. Choose Create snapshot to save a snapshot of the database.

Das Hinzufügen einer DB-Instance dauert ca. 10 Minuten. Sobald die Umgebungsaktualisierung abgeschlossen ist, stehen der Hostname der DB-Instance und andere Verbindungsinformationen über die folgenden Umgebungseigenschaften zur Verfügung:

Eigenschaftenname	Beschreibung	Eigenschaftswert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbankname, ebdb	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: DB Name (DB-Name).
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	In der Amazon RDS-Konsole nicht als Referenz verfügbar.

Verbinden mit der Datenbank

Verwenden Sie die Verbindungsinformationen für die Verbindung mit Ihrer DB von innerhalb Ihrer Anwendung über Umgebungsvariablen. Weitere Informationen zur Verwendung von Amazon RDS mit Ihren Anwendungen finden Sie in den folgenden Themen.

- Java SE – [Verbinden mit einer Datenbank \(Java SE-Plattformen\)](#) (p. 136)
- Java mit Tomcat – [Verbinden mit einer Datenbank \(Tomcat-Plattformen\)](#) (p. 137)
- Node.js – [Verbinden mit einer Datenbank](#) (p. 285)
- .NET – [Verbinden mit einer Datenbank](#) (p. 219)
- PHP – [Verbinden mit einer Datenbank mit einem PDO oder MySQLi](#) (p. 351)
- Python – [Verbinden mit einer Datenbank](#) (p. 380)
- Ruby – [Verbinden mit einer Datenbank](#) (p. 399)

Konfigurieren einer integrierten RDS-DB-Instance mithilfe der Konsole

Sie können Konfigurationseinstellungen für die DB-Instance im Abschnitt Database (Datenbank) auf der Seite Configuration (Konfiguration) der Umgebung in der [Elastic Beanstalk-Konsole](#) (p. 426) anzeigen und ändern.

So konfigurieren Sie die DB-Instance Ihrer Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Database (Datenbank) die Option Edit (Bearbeiten).

Sie können nach der Datenbankerstellung die Einstellungen Instance class (Instance-Klasse), Storage (Speicher), Password (Passwort), Retention (Aufbewahrung) und Availability (Verfügbarkeit) ändern. Wenn Sie die Instance-Klasse ändern, stellt Elastic Beanstalk die DB-Instance erneut bereit.

Warnung

Ändern Sie keine Einstellungen auf der DB-Instance außerhalb der Funktionen von Elastic Beanstalk (z. B. in der RDS-Konsole). Wenn Sie dies tun, ist Ihre Amazon RDS-DB-Konfiguration möglicherweise nicht synchron mit der Definition Ihrer Umgebung. Wenn Sie Ihre Umgebung aktualisieren oder neu starten, überschreiben die Einstellungen in der Umgebung alle Einstellungen, die Sie außerhalb von Elastic Beanstalk vornehmen.

Wenn Sie Einstellungen ändern müssen, die Elastic Beanstalk nicht direkt unterstützt, verwenden Sie die Elastic Beanstalk-Konfigurationsdateien ([p. 614](#)).

Konfigurieren einer integrierten RDS-DB-Instance mithilfe von Konfigurationsdateien

Sie können die DB-Instance Ihrer Umgebung mithilfe von [Konfigurationsdateien \(p. 722\)](#) konfigurieren. Verwenden Sie die Optionen im Namespace [aws:rds:dbinstance \(p. 711\)](#). Das folgende Beispiel ändert die Größe des zugewiesenen Datenbankspeichers auf 100 GB.

Example .ebextensions/db-instance-options.config

```
option_settings:  
  aws:rds:dbinstance:  
    DBAllocatedStorage: 100
```

Wenn Sie DB-Instance-Eigenschaften konfigurieren, die Elastic Beanstalk nicht unterstützt, können Sie dennoch eine Konfigurationsdatei verwenden und Ihre Einstellungen mithilfe des `resources`-Schlüssels angeben. Das folgende Beispiel legt Werte für die Amazon RDS-Eigenschaften `StorageType` und `Iops` fest.

Example .ebextensions/db-instance-properties.config

```
Resources:  
  AWSEBRDSDatabase:  
    Type: AWS::RDS::DBInstance  
    Properties:  
      StorageType:io1  
      Iops: 1000
```

Sicherheit Ihrer AWS Elastic Beanstalk-Umgebung

Elastic Beanstalk bietet mehrere Optionen für die Steuerung der Sicherheit Ihrer Umgebung und der darin enthaltenen Amazon EC2-Instances. In diesem Thema wird die Konfiguration dieser Optionen beschrieben

Abschnitte

- [Konfigurieren der Umgebungssicherheit \(p. 615\)](#)
- [Namespaces für die Konfiguration der Umgebungssicherheit \(p. 617\)](#)

Konfigurieren der Umgebungssicherheit

Sie können die Sicherheitskonfiguration Ihrer Elastic Beanstalk-Umgebung in der Elastic Beanstalk-Konsole ändern.

So konfigurieren Sie die Umgebungssicherheit in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Security (Sicherheit) die Option Edit (Bearbeiten).

Die folgenden Einstellungen sind verfügbar.

Einstellungen

- [Servicerolle \(p. 616\)](#)
- [EC2 key pair \(p. 616\)](#)
- [IAM-Instance-Profil \(p. 617\)](#)

Modify security

Service role

Service role

aws-elasticbeanstalk-service-role



Virtual machine permissions

EC2 key pair

-- Choose a key pair --



IAM instance profile

aws-elasticbeanstalk-ec2-role



[Cancel](#)

[Continue](#)

Servicerolle

Wählen Sie eine [Service role \(Servicerolle\) \(p. 909\)](#) aus, die Sie Ihrer Elastic Beanstalk-Umgebung zuordnen möchten. Elastic Beanstalk nimmt diese Servicerolle an, wenn es in Ihrem Namen auf andere AWS-Services zugreift. Details dazu finden Sie unter [Rollen von Elastic Beanstalk Service verwalten \(p. 909\)](#).

EC2 key pair

Mit einem Amazon EC2-Schlüsselpaar können Sie sich sicher an den für Ihre Elastic Beanstalk-Anwendung bereitgestellten Amazon Elastic Compute Cloud (Amazon EC2)-Instances anmelden. Anleitungen zum Erstellen eines Schlüsselpaares finden Sie unter [Erstellen eines Schlüsselpaares mithilfe von Amazon EC2](#) im Amazon EC2-Benutzerhandbuch für Linux Instances.

Note

Wenn Sie ein Schlüsselpaar erstellen, speichert Amazon EC2 eine Kopie Ihres öffentlichen Schlüssels. Wenn Sie es nicht mehr benötigen, um sich mit einer beliebigen Umgebungs-Instance

zu verbinden, können Sie es aus Amazon EC2 löschen. Einzelheiten finden Sie unter [Löschen des Schlüsselpaars](#) im Amazon EC2-Benutzerhandbuch für Linux Instances.

Wählen Sie ein EC2 key pair (EC2-Schlüsselpaar) aus dem Dropdown-Menü aus und weisen Sie es den Umgebungs-Instances zu. Beim Zuweisen eines Schlüsselpaars wird der öffentliche Schlüssel (zum Authentifizieren des privaten Schlüssels) auf der Instance gespeichert. Die Speicherung des lokalen Schlüssels hingegen erfolgt lokal. Der private Schlüssel wird nie in AWS gespeichert.

Weitere Informationen zum Herstellen einer Verbindung zu Amazon EC2-Instances finden Sie unter [Herstellen einer Verbindung mit Ihrer Instance](#) und [Herstellen einer Verbindung zu Linux/UNIX-Instances aus Windows mit PuTTY](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances.

IAM-Instance-Profil

Ein [Instance-Profil \(p. 22\)](#) ist eine IAM-Rolle, die für in der Elastic Beanstalk-Umgebung gestartete Instances gilt. Amazon EC2-Instances übernehmen die Instance-Profilrolle, um Anforderungen an AWS zu signieren und APIs aufzurufen, z. B. zum Hochladen von Protokollen in Amazon S3.

Wenn Sie zum ersten Mal eine Umgebung mit der Elastic Beanstalk-Konsole erstellen, werden Sie von Elastic Beanstalk zum Erstellen eines Instance-Profils mit Standardberechtigungen aufgefordert. Sie können diesem Profil weitere Berechtigungen hinzufügen, um der Instance Zugriff auf andere AWS-Services zu gewähren. Details dazu finden Sie unter [Elastic Beanstalk Instance-Profile verwalten \(p. 904\)](#).

Namespaces für die Konfiguration der Umgebungssicherheit

Elastic Beanstalk bietet [Konfigurationsoptionen \(p. 646\)](#) in den folgenden Namespaces, damit Sie die Sicherheit Ihrer Umgebung anpassen können:

- [aws:elasticbeanstalk:environment \(p. 686\)](#) – Konfigurieren der Servicerolle der Umgebung mit der Option ServiceRole.
- [aws:autoscaling:launchconfiguration \(p. 668\)](#) – Konfigurieren von Berechtigungen für die Amazon EC2-Instances der Umgebung mit den Optionen EC2KeyName und IamInstanceProfile.

Die EB CLI und Elastic Beanstalk-Konsole wenden die empfohlenen Werte für die vorhergehenden Optionen an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter [Empfohlene Werte \(p. 648\)](#).

Ressourcen in Ihren Elastic Beanstalk-Umgebungen markieren

Sie können Tags auf Ihre AWS Elastic Beanstalk-Umgebungen anwenden. Tags sind mit AWS-Ressourcen verknüpfte Schlüssel-Wert-Paare. Weitere Informationen zum Elastic Beanstalk-Ressourcen-Tagging, Anwendungsfälle, Einschränkungen für Tag-Schlüssel und -Werte sowie unterstützte Ressourcentypen finden Sie unter [Markieren von Elastic Beanstalk-Anwendungsressourcen \(p. 420\)](#).

Elastic Beanstalk wendet Umgebungs-Tags sowohl auf die Umgebungsressource selbst als auch auf andere AWS-Ressourcen an, die Elastic Beanstalk für die Umgebung erstellt. Sie können Tags zum Verwalten von Berechtigungen auf einer bestimmten Ressourcenebene in einer Umgebung verwenden.

Weitere Informationen finden Sie unter [Tagging von Amazon EC2-Ressourcen](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Standardmäßig wendet Elastic Beanstalk ein paar Tags auf Ihre Umgebung an:

- `elasticbeanstalk:environment-name` – Der Name der Umgebung.
- `elasticbeanstalk:environment-id` – Die Umgebungs-ID.
- `Name` – Ebenfalls der Name der Umgebung. `Name` wird im Amazon EC2-Dashboard verwendet, um Ressourcen zu identifizieren und zu sortieren.

Diese Standard-Tags können nicht bearbeitet werden.

Sie können Tags angeben, wenn Sie die Elastic Beanstalk-Umgebung erstellen. In einer vorhandenen Umgebung können Sie Tags hinzufügen oder entfernen sowie die Werte von vorhandenen Tags aktualisieren. Eine Umgebung kann einschließlich der Standard-Tags bis zu 50 Tags enthalten.

Hinzufügen von Tags beim Erstellen einer Umgebung

Wenn Sie eine Umgebung mit der Elastic Beanstalk-Konsole erstellen, können Sie auf der Konfigurationsseite `Modify tags` (Tags ändern) im [Assistenten zum Erstellen neuer Umgebungen](#) (p. 437) Tag-Schlüssel und -Werte angeben.

Key	Value
mytag1	value1

Add tag

49 remaining

Wenn Sie mit der EB CLI Umgebungen erstellen, verwenden Sie die `--tags`-Option mit dem Befehl `eb create` (p. 1053), um Tags hinzuzufügen.

```
~/workspace/my-app$ eb create --tags mytag1=value1,mytag2=value2
```

Mit der AWS-CLI oder anderen API-basierten Clients verwenden Sie den `--tags`-Parameter auf dem `create-environment`-Befehl:

```
$ aws elasticbeanstalk create-environment \
  --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
  --application-name my-app --environment-name my-env --cname-prefix my-app --version-
  label v1 --template-name my-saved-config
```

[Gespeicherte Konfigurationen \(p. 651\)](#) umfassen benutzerdefinierte Tags. Wenn Sie eine gespeicherte Konfiguration mit Tags während der Umgebungserstellung anwenden, werden diese Tags auf die neue Umgebung angewendet, solange Sie keine neuen Tags angeben. Wenn Sie Tags mit einer der vorstehenden Methoden zu einer Umgebung hinzufügen, werden sämtliche Tags, die in der gespeicherten Konfiguration definiert sind, verworfen.

Verwalten von Tags von bestehenden Umgebungen

Sie können Tags in einer bestehenden Elastic Beanstalk-Umgebung hinzufügen, aktualisieren und löschen. Elastic Beanstalk wendet die Änderungen auf die Ressourcen Ihrer Umgebung an.

Es ist jedoch nicht möglich, die Standard-Tags zu bearbeiten, die Elastic Beanstalk für Ihre Umgebung nutzt.

So verwalten Sie die Tags einer Umgebung in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Tags aus.

Auf der Tag-Management-Seite wird die Liste der Tags angezeigt, die in der Umgebung vorhanden sind.

The screenshot shows the 'Tags for GettingStartedApp-env' page. At the top, there's a breadcrumb navigation: Elastic Beanstalk > Environments > GettingStartedApp-env > Tags. Below the title, a note says: 'Apply up to 47 tags in addition to the default tags to the resources in your environment. You can use tags to group environments. A tag is a key-value pair. The key must be unique within the environment and is case-sensitive.' There's a table with two columns: 'Key' and 'Value'. The first row has a key 'elasticbeanstalk:environment-id' and a value 'e-cubmdjm6ga'. The second row has a key 'elasticbeanstalk:environment-name' and a value 'GettingStartedApp-env'. The third row has a key 'Name' and a value 'GettingStartedApp-env'. Below the table, there are two rows for custom tags: 'mytag1' with value 'value1' and 'mytag2' with value 'value2'. Each tag row has an 'Add tag' button and a 'Remove' button. At the bottom left, it says '45 remaining'.

4. Sie können Tags hinzufügen, aktualisieren oder löschen:

- Um ein Tag hinzuzufügen, geben Sie es in die leeren Felder am Ende der Liste ein. Um einen weiteren Tag hinzuzufügen, wählen Sie Add tag (Tag hinzufügen) aus. Elastic Beanstalk fügt ein weiteres Paar leerer Felder hinzu.
- Um den Schlüssel oder Wert eines Tags zu aktualisieren, bearbeiten Sie die entsprechenden Felder in der Zeile des Tags.
- Um ein Tag zu löschen, wählen Sie Remove (Entfernen) neben dem Wertfeld des Tags.

5. Wählen Sie Apply (Anwenden) aus.

Wenn Sie Umgebungen mit der EB CLI aktualisieren, verwenden Sie zum Hinzufügen, Aktualisieren, Löschen oder Auflisten von Tags den Befehl [eb tags \(p. 1092\)](#).

Mit dem folgenden Befehl werden beispielsweise die Tags in Ihrer Standardumgebung aufgelistet.

```
~/workspace/my-app$ eb tags --list
```

Mit dem folgenden Befehl wird das Tag mytag1 aktualisiert und das Tag mytag2 gelöscht.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2
```

Eine umfassende Liste der Optionen sowie weitere Beispiele finden Sie unter [eb tags \(p. 1092\)](#).

Wenn Sie die AWS CLI oder andere API-basierte Clients verwenden, verwenden Sie den Befehl [list-tags-for-resource](#), um die Tags einer Umgebung aufzulisten.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:environment/my-app/my-env"
```

Verwenden Sie den Befehl [update-tags-for-resource](#) zum Hinzufügen, Aktualisieren und Löschen von Tags in einer Umgebung.

```
$ 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"
```

Geben Sie sowohl hinzuzufügende als auch zu aktualisierende Tags im Parameter `--tags-to-add` des Befehls `update-tags-for-resource` an. Wenn ein Tag nicht vorhanden ist, wird es hinzugefügt, andernfalls wird es aktualisiert.

Note

Um diese beiden AWS-CLI-Befehle mit einer Elastic Beanstalk-Umgebung zu verwenden, benötigen Sie den ARN der Umgebung. Verwenden Sie den folgenden Befehl, um den ARN abzurufen.

```
$ aws elasticbeanstalk describe-environments
```

Umgebungseigenschaften und andere Softwareeinstellungen

Auf der Konfigurationsseite `Modify software` (Software ändern) können Sie die Software auf den Elastic Compute Cloud (Amazon EC2)-Instances konfigurieren, auf denen Ihre Anwendung ausgeführt wird. Sie können Umgebungseigenschaften, AWS X-Ray Debugging, das Speichern und Streamen von Instance-Protokollen sowie plattformspezifische Einstellungen konfigurieren.

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

Themen

- [Plattformspezifische Einstellungen konfigurieren \(p. 622\)](#)
- [Konfigurieren von Umgebungseigenschaften \(p. 623\)](#)
- [Softwareeinstellung Namespaces \(p. 625\)](#)
- [Zugreifen auf Umgebungseigenschaften \(p. 626\)](#)
- [Konfigurieren von AWS X-Ray Debugging \(p. 627\)](#)
- [Anzeigen Ihrer Elastic Beanstalk-Umgebungsprotokolle \(p. 630\)](#)

Plattformspezifische Einstellungen konfigurieren

Für die meisten Elastic Beanstalk-Plattformen können Sie neben den standardmäßigen Optionen für alle Umgebungen auch sprach- und Framework-spezifische Einstellungen festlegen. Diese werden im Abschnitt Platform options (Plattformoptionen) der Seite Modify software (Software ändern) angezeigt und können die folgenden Formen annehmen.

- Vordefinierte Umgebungseigenschaften – Die Ruby-Plattform verwendet Umgebungseigenschaften für Framework-Einstellungen wie `RACK_ENV` und `BUNDLE_WITHOUT`.
- Platzhalter-Umgebungseigenschaften – Die Tomcat-Plattform definiert eine Umgebungseigenschaft namens `JDBC_CONNECTION_STRING`, die keinen bestimmten Wert erhält. Solche Einstellungen waren auf älteren Plattformversionen gängiger.
- Konfigurationsoptionen – Die meisten Plattformen legen [Konfigurationsoptionen \(p. 646\)](#) in plattformspezifischen oder freigegebenen Namespaces wie `aws:elasticbeanstalk:xray` oder `aws:elasticbeanstalk:container:python` fest.

So konfigurieren Sie plattformspezifische Einstellungen in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Nehmen Sie unter Platform options (Plattformoptionen) die erforderlichen Änderungen an der Optionseinstellung vor.
6. Wählen Sie Apply (Anwenden) aus.

Informationen zu den plattformspezifischen Optionen und zum Abrufen von Umgebungseigenschaftswerten in Ihrem Code finden Sie im Thema zu Plattformen für Ihre Sprache bzw. Ihr Framework:

- Docker – [the section called “Umgebungskonfiguration” \(p. 85\)](#)
- Go – [Verwenden der Elastic Beanstalk Go-Plattform \(p. 101\)](#)
- Java SE – [Elastic Beanstalk Java SE-Plattform verwenden \(p. 128\)](#)
- Tomcat – [Elastic Beanstalk Tomcat-Plattform verwenden \(p. 117\)](#)
- .NET Core unter Linux – [Verwenden der .NET Core unter Linux-Plattform \(p. 159\)](#)
- .NET – [Verwenden der Elastic Beanstalk .NET-Plattform \(p. 193\)](#)
- Node.js – [Verwenden der Elastic Beanstalk-Node.js-Plattform \(p. 250\)](#)
- PHP – [Verwenden der Elastic Beanstalk-PHP-Plattform \(p. 288\)](#)
- Python – [Verwenden der Elastic Beanstalk-Python-Plattform \(p. 355\)](#)
- Ruby – [Verwenden der Elastic Beanstalk-Ruby-Plattform \(p. 383\)](#)

Konfigurieren von Umgebungseigenschaften

Mithilfe von Environment properties (Umgebungseigenschaften) können Sie Verschlüsselungen, Endpunkte, Debugging-Einstellungen und andere Informationen an Ihre Anwendung übergeben. Mithilfe von Umgebungseigenschaften können Sie Anwendungen zu unterschiedlichen Zwecken in mehreren Umgebungen ausführen, z. B. Entwicklung, Testing, Staging und Produktion.

Wenn Sie darüber hinaus [eine Datenbank zu Ihrer Umgebung hinzufügen \(p. 609\)](#), legt Elastic Beanstalk Umgebungseigenschaften wie `RDS_HOSTNAME` fest, die Sie im Anwendungscode auslesen können, um ein Verbindungsobjekt oder eine Verbindungszeichenfolge zu erstellen.

Umgebungsvariablen

In den meisten Fällen werden Umgebungseigenschaften als Umgebungsvariablen übergeben, das Verhalten ist jedoch abhängig von der Plattform. Die [Java SE-Plattform \(p. 128\)](#) legt beispielsweise Umgebungsvariablen fest, die Sie mit `System.getenv` abrufen. Die [Tomcat-Plattform \(p. 117\)](#) dagegen legt Java-Systemeigenschaften fest, die Sie mit `System.getProperty` abrufen. Im Allgemeinen sind Eigenschaften nicht sichtbar, wenn Sie eine Verbindung zu einer Instance herstellen und `env` ausführen.

So konfigurieren Sie Umgebungseigenschaften in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Geben Sie unter Environment properties (Umgebungseigenschaften) Schlüssel/Wert-Paare ein.

Name	Value
JDBC_CONNECTION_STRING	
XRAY_ENABLED	{ "Fn::GetOptionSetting" : {"Name" : "...}}

6. Wählen Sie Apply (Anwenden) aus.

Beschränkungen von Umgebungseigenschaften

- Schlüssel dürfen beliebige alphanumerische Zeichen sowie die folgenden Symbole enthalten: _ . : / + \ - @

Die aufgeführten Symbole gelten für Umgebungseigenschaftsschlüssel, möglicherweise aber nicht für Umgebungsvariablennamen auf der Plattform Ihrer Umgebung. Wenn Sie sicherstellen möchten, dass Ihre Umgebungsvariablen mit allen Plattformen kompatibel sind, beschränken Sie sich auf folgendes Muster: [A-Z_][A-Z0-9_]*

- Werte dürfen beliebige alphanumerische Zeichen, Leerzeichen sowie die folgenden Symbole enthalten: _ . : / = + \ - @ ' "

Note

Einfache und doppelte Anführungszeichen in Werten müssen durch das Escape-Zeichen geschützt werden.

- Schlüssel können bis zu 128 Zeichen enthalten. Werte können bis zu 256 Zeichen enthalten.
- Bei Schlüsseln und Werten wird die Groß-/Kleinschreibung berücksichtigt.

- Die Gesamtgröße aller Umgebungseigenschaften darf 4.096 Bytes nicht überschreiten, wenn diese als Zeichenfolgen im Format **Schlüssel=Wert** gespeichert werden.

Softwareeinstellung Namespaces

Mithilfe einer [Konfigurationsdatei \(p. 722\)](#) können Sie im Rahmen der Bereitstellung Konfigurationsoptionen festlegen und andere Instance-Konfigurationsaufgaben ausführen. Konfigurationsoptionen können durch den Elastic Beanstalk-Service oder die verwendete Plattform definiert und in Namespaces organisiert werden.

Sie können mit den [Konfigurationsdateien \(p. 722\)](#) von Elastic Beanstalk Umgebungseigenschaften und Konfigurationsoptionen in Ihrem Quellcode festlegen. Verwenden Sie den [aws:elasticbeanstalk:application:environmentNamespace \(p. 684\)](#), um Umgebungseigenschaften zu definieren.

Example .ebextensions/options.config

```
option_settings:  
  aws:elasticbeanstalk:application:environment:  
    API_ENDPOINT: www.example.com/api
```

Wenn Sie [benutzerdefinierte Ressourcen \(p. 745\)](#) mithilfe von Konfigurationsdateien oder AWS CloudFormation-Vorlagen erstellen, können Sie mit einer AWS CloudFormation-Funktion Informationen zu der Ressource abrufen und diese während der Bereitstellung dynamisch einer Umgebungseigenschaft zuweisen. Im folgenden Beispiel aus dem GitHub-Repository [elastic-beanstalk-samples](#) wird die [Ref-Funktion \(p. 750\)](#) verwendet, um den ARN eines Amazon SNS-Themas abzurufen, das erstellt und einer Umgebungseigenschaft namens `NOTIFICATION_TOPIC` zugewiesen wird.

Hinweise

- Wenn Sie eine AWS CloudFormation-Funktion verwenden, um eine Umgebungseigenschaft zu definieren, zeigt die Elastic Beanstalk-Konsole den Wert der Eigenschaft an, bevor die Funktion ausgewertet wird. Sie können das [get-config-Plattformskript \(p. 43\)](#) verwenden, um die Werte der Umgebungseigenschaften zu bestätigen, die für Ihre Anwendung verfügbar sind.
- Die [Multicontainer-Docker \(p. 64\)](#)-Plattform verwendet kein AWS CloudFormation zum Erstellen von Container-Ressourcen. Dies hat zur Folge, dass diese Plattform die Definition von Umgebungseigenschaften mit AWS CloudFormation-Funktionen nicht unterstützt.

Example .Ebextensions/sns-topic.config

```
Resources:  
  NotificationTopic:  
    Type: AWS::SNS::Topic  
  
option_settings:  
  aws:elasticbeanstalk:application:environment:  
    NOTIFICATION_TOPIC: `{"Ref" : "NotificationTopic"}`
```

Sie können mithilfe dieser Funktion auch Informationen aus [AWS CloudFormation-Pseudoparametern](#) bereitstellen. In diesem Beispiel wird die derzeitige Region abgerufen und einer Eigenschaft namens `AWS_REGION` zugewiesen.

Example .Ebextensions/env-regionname.config

```
option_settings:  
  aws:elasticbeanstalk:application:environment:
```

```
AWS_REGION: `{"Ref" : "AWS::Region"}'
```

Die meisten Elastic Beanstalk-Plattformen definieren zusätzliche Namespaces mit Optionen zur Konfiguration von Software, die auf der Instance ausgeführt wird, wie den Reverse-Proxy, der Anforderungen an Ihre Anwendung weiterleitet. Weitere Informationen zu den für Ihre Plattform verfügbaren Namespaces finden Sie unter:

- Go – [Go-Konfigurations-Namespace \(p. 103\)](#)
- Java SE – [Java SE-Konfigurations-Namespace \(p. 130\)](#)
- Tomcat – [Tomcat-Konfigurations-Namespaces \(p. 120\)](#)
- .NET Core unter Linux – [.NET Core unter Linux-Konfigurations-Namespace \(p. 161\)](#)
- .NET – [Der Namespace aws:elasticbeanstalk:container:dotnet:apppool \(p. 194\)](#)
- Node.js – [Node.js-Konfigurations-Namespace \(p. 253\)](#)
- PHP – [Der Namespace aws:elasticbeanstalk:container:php:phpini \(p. 291\)](#)
- Python – [Python-Konfigurations-Namespaces \(p. 358\)](#)
- Ruby – [Ruby-Konfigurations-Namespaces \(p. 386\)](#)

Elastic Beanstalk bietet zahlreiche Konfigurationsoptionen für die Anpassung Ihrer Umgebung. Neben den Konfigurationsdateien können Sie die Konfigurationsoptionen auch unter Verwendung der Konsole, gespeicherter Konfigurationen, der EB CLI oder der AWS CLI einrichten. Weitere Informationen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Zugreifen auf Umgebungseigenschaften

In den meisten Fällen können Sie in Ihrem Anwendungscode auf Umgebungseigenschaften wie UmgebungsvARIABLEN zugreifen. In der Regel werden Umgebungseigenschaften jedoch nur an die Anwendung übergeben und können nicht angezeigt werden, indem Sie sich mit einer Instance in Ihrer Umgebung verbinden und env ausführen.

- [Go \(p. 102\) – os.Getenv](#)

```
endpoint := os.Getenv("API_ENDPOINT")
```

- [Java SE \(p. 130\) – System.getenv](#)

```
String endpoint = System.getenv("API_ENDPOINT");
```

- [Tomcat \(p. 119\) – System.getProperty](#)

```
String endpoint = System.getProperty("API_ENDPOINT");
```

- [.NET Core unter Linux \(p. 160\) – Environment.GetEnvironmentVariable](#)

```
string endpoint = Environment.GetEnvironmentVariable("API_ENDPOINT");
```

- [.NET \(p. 194\) – appConfig](#)

```
NameValueCollection appConfig = ConfigurationManager.AppSettings;
string endpoint = appConfig["API_ENDPOINT"];
```

- [Node.js \(p. 252\) – process.env](#)

```
var endpoint = process.env.API_ENDPOINT
```

- [PHP \(p. 290\)](#) – `$_SERVER`

```
$endpoint = $_SERVER['API_ENDPOINT'];
```

- [Python \(p. 357\)](#) – `os.environ`

```
import os
endpoint = os.environ['API_ENDPOINT']
```

- [Ruby \(p. 385\)](#) – `ENV`

```
endpoint = ENV['API_ENDPOINT']
```

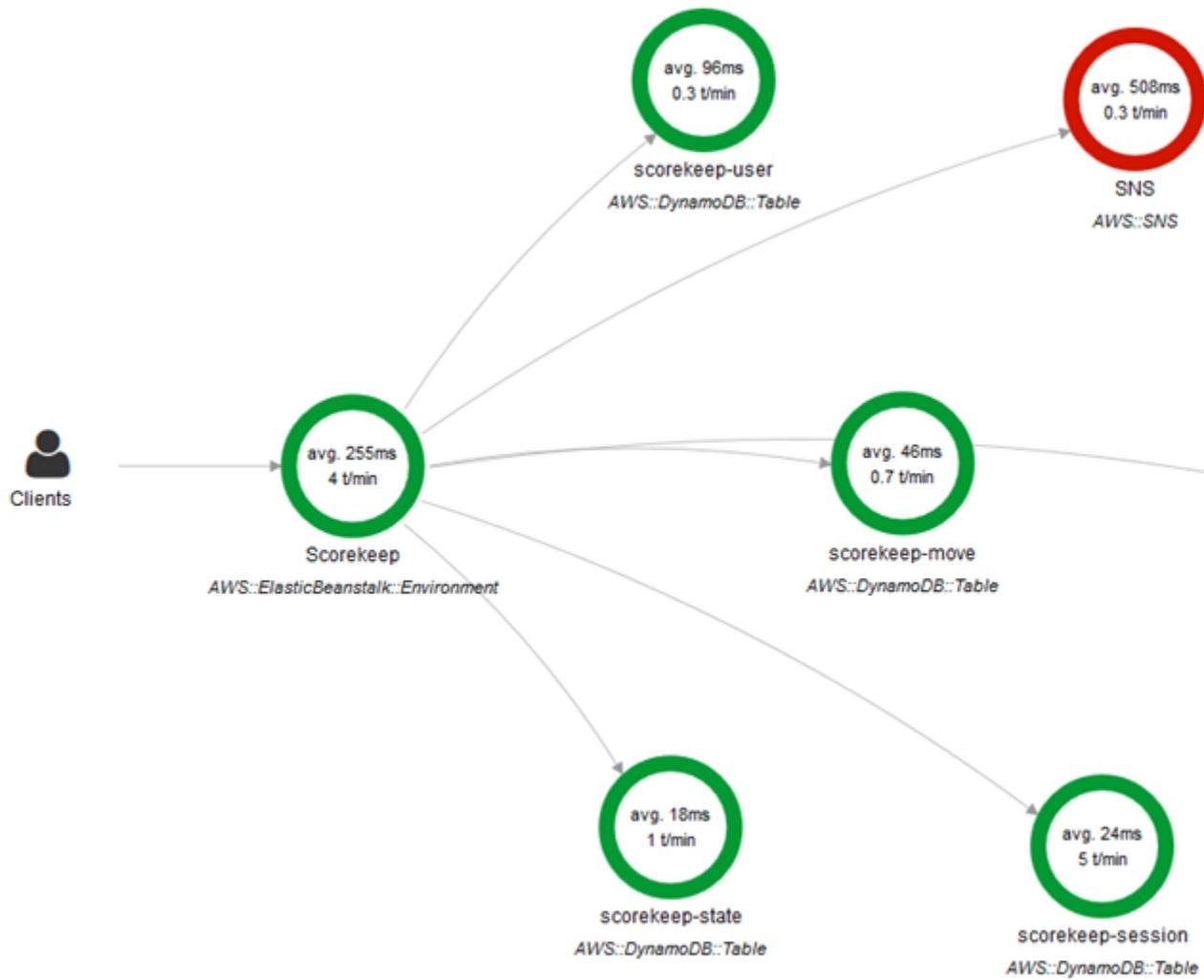
Außerhalb des Anwendungscodes, beispielsweise in Skripten, die während der Bereitstellung ausgeführt werden, können Sie auf Umgebungseigenschaften mit dem [get-config-Plattformskript \(p. 43\)](#) zugreifen. Im GitHub-Repository [elastic-beanstalk-samples](#) finden Sie Beispielkonfigurationen, in denen `get-config` verwendet wird.

Konfigurieren von AWS X-Ray Debugging

Sie können die AWS Elastic Beanstalk-Konsole oder eine Konfigurationsdatei für die Ausführung des AWS X-Ray-Daemon auf den Instances in Ihrer Umgebung verwenden. X-Ray ist ein AWS-Service, der Daten über die Anforderungen erfasst, die Ihre Anwendung verwendet, um eine Servicezuordnung zu erstellen, die Sie verwenden können, um Probleme mit Ihrer Anwendung und Möglichkeiten zur Optimierung zu identifizieren.

Note

X-Ray wird nicht in allen Regionen angeboten. Wenn Sie eine Umgebung in einer dieser Regionen erstellen, können Sie den X-Ray-Daemon auf den Instances in Ihrer Umgebung nicht ausführen. Weitere Informationen zu den AWS-Services, die in den einzelnen Regionen angeboten werden, finden Sie in der [Tabelle der Regionen](#).



X-Ray bietet ein SDK, das Sie verwenden können, um Ihren Anwendungscode zu instrumentalisieren, sowie eine Daemon-Anwendung, die Debugging-Informationen vom SDK an die X-Ray-API weiterleitet.

Unterstützte Plattformen

Sie können das X-Ray SDK mit den folgenden Elastic Beanstalk-Plattformen verwenden:

- Go – Version 2.9.1 und höher
- Java 8 – Version 2.3.0 und höher
- Java 8 mit Tomcat 8 – Version 2.4.0 und höher
- Node.js – Version 3.2.0 und höher
- Windows Server – alle Plattformversionen, die am oder nach dem 18. Dezember 2016 veröffentlicht wurden
- Python - Version 2.5.0 und höher

Auf unterstützten Plattformen können Sie eine Konfigurationsoption für die Ausführung des X-Ray-Daemon auf den Instances in Ihrer Umgebung verwenden. Sie können den Daemon in der [Elastic Beanstalk-Konsole \(p. 629\)](#) oder über eine [Konfigurationsdatei \(p. 629\)](#) aktivieren.

Der X-Ray-Daemon benötigt IAM-Berechtigungen in der verwalteten Richtlinie AWSXRayWriteOnlyAccess, um Daten in X-Ray hochzuladen. Diese Berechtigungen sind im [Elastic Beanstalk-Instance-Profil enthalten \(p. 22\)](#). Wenn Sie nicht das Standard-Instance-Profil verwenden, finden Sie Informationen unter [Erteilen der Berechtigung zum Senden von Daten an X-Ray für den Daemon im AWS X-Ray-Entwicklerhandbuch](#).

Debugging mit X-Ray erfordert die Verwendung des X-Ray-SDK. Anweisungen und Beispieldokumentationen finden Sie unter [Erste Schritte mit AWS X-Ray](#) im AWS X-Ray-Entwicklerhandbuch.

Wenn Sie eine Plattformversion verwenden, die nicht über den Daemon verfügt, können Sie sie trotzdem mit einem Skript in einer Konfigurationsdatei ausführen. Weitere Informationen finden Sie unter [Manuelles Herunterladen und Ausführen des X-Ray-Daemons \(Erweitert\)](#) im AWS X-Ray-Entwicklerhandbuch.

Abschnitte

- [Konfigurieren des Debuggings \(p. 629\)](#)
- [Der Namespace aws:elasticbeanstalk:xray \(p. 629\)](#)

Konfigurieren des Debuggings

Sie können den X-Ray-Daemon auf einer laufenden Umgebung in der Elastic Beanstalk-Konsole aktivieren.

So aktivieren Sie das Debugging in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Wählen Sie im Abschnitt AWS X-Ray die Option X-Ray daemon (X-Ray Daemon) aus.
6. Wählen Sie Apply (Anwenden) aus.

Sie können diese Option auch während der Umgebungserstellung aktivieren. Weitere Informationen finden Sie unter [Der Assistent zum Erstellen einer neuen Umgebung \(p. 437\)](#).

Der Namespace aws:elasticbeanstalk:xray

Sie können mit der XRayEnabled-Option im aws:elasticbeanstalk:xray-Namespace Debugging aktivieren.

Um das Debugging bei der Bereitstellung Ihrer Anwendung automatisch zu aktivieren, legen Sie die Option in einer [Konfigurationsdatei \(p. 722\)](#) in Ihrem Quellcode wie folgt fest.

Example .ebextensions/debugging.config

```
option_settings:  
  aws:elasticbeanstalk:xray:  
    XRayEnabled: true
```

Anzeigen Ihrer Elastic Beanstalk-Umgebungsprotokolle

AWS Elastic Beanstalk bietet zwei Möglichkeiten, regelmäßig Protokolle von den Amazon EC2-Instances anzuzeigen, auf denen Ihre Anwendung ausgeführt wird:

- Konfigurieren Sie Ihre Elastic Beanstalk-Umgebung, um gedrehte Instance-Protokolle in den Amazon S3-Bucket der Umgebung hochzuladen.
- Konfigurieren Sie die Umgebung, um Instance-Protokolle an Amazon CloudWatch Logs zu streamen.

Bei der Konfiguration eines Instance-Protokoll-Streamings auf CloudWatch Logs erstellt Elastic Beanstalk CloudWatch Logs-Protokollgruppen für Proxy- und Bereitstellungsprotokolle auf den Amazon EC2-Instances und übermittelt diese Protokolldateien in Echtzeit an CloudWatch Logs. Weitere Informationen zu Instance-Protokollen finden Sie unter [Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk-Umgebung anzeigen \(p. 869\)](#).

Zusätzlich zu den Instance-Protokollen können Sie bei Aktivierung des [erweiterten Zustands \(p. 824\)](#) für Ihre Umgebung die Umgebung für das Streamen von Zustandsinformationen an CloudWatch Logs konfigurieren. Wenn sich der Zustand der Umgebung ändert, fügt Elastic Beanstalk einen Datensatz mit dem neuen Status und einer Beschreibung der Ursache der Änderung zu einer Zustandsprotokollgruppe hinzu. Weitere Informationen zum Streaming des Umgebungszustands finden Sie unter [Streaming von Integritätsinformationen der Elastic Beanstalk-Umgebung an Amazon CloudWatch Logs \(p. 888\)](#).

Konfigurieren der Anzeige von Instance-Protokollen

Um Instance-Protokolle anzuzeigen, können Sie in der Elastic Beanstalk-Konsole die Rotation von Instance-Protokollen und das Protokoll-Streaming aktivieren.

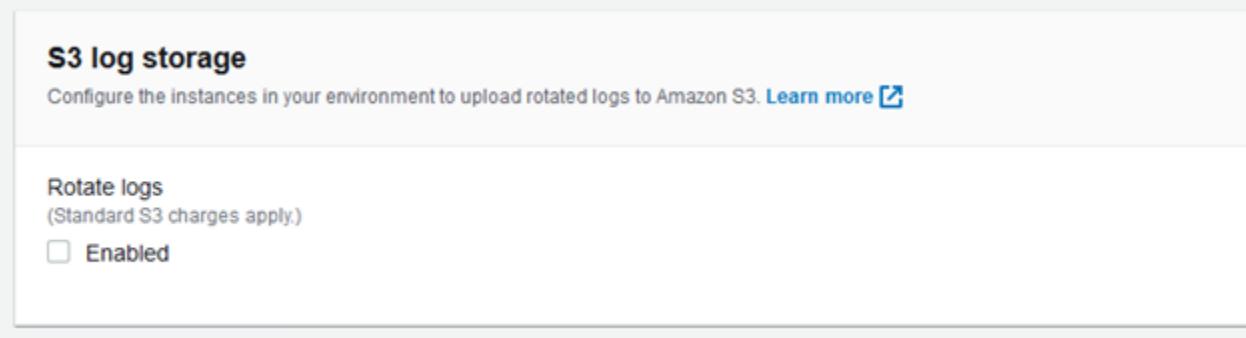
So konfigurieren Sie die Rotation von Instance-Protokollen und das Protokoll-Streaming in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Wählen Sie im Abschnitt S3 log storage (S3-Protokollspeicher) die Option Rotate logs (Protokolle rotieren), um das Hochladen rotierter Protokolle in Amazon S3 zu aktivieren.



S3 log storage
Configure the instances in your environment to upload rotated logs to Amazon S3. [Learn more](#)

Rotate logs
(Standard S3 charges apply.)

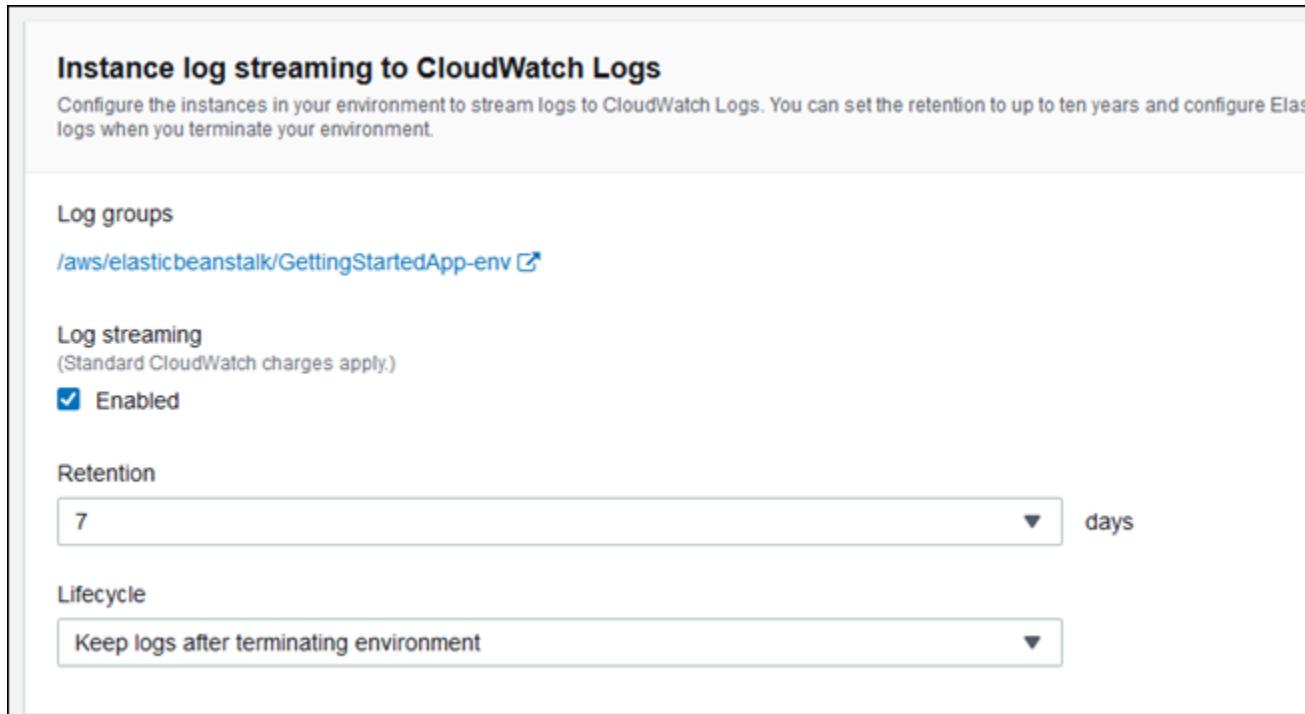
Enabled

6. Konfigurieren Sie im Bereich Instance log streaming to CloudWatch Logs (Instance-Protokoll-Streaming zu CloudWatch Logs) die folgenden Einstellungen:

- Log streaming (Protokollstreaming) – Wählen Sie diese Option aus, um das Protokollstreaming zu aktivieren.
- Retention (Aufbewahrung) – Geben Sie die Anzahl von Tagen an, für die Protokolle in CloudWatch Logs aufbewahrt werden.
- Lifecycle (Lebenszyklus) – Wählen Sie die Option Delete logs upon termination (Protokolle beim Beenden löschen) aus, um Protokolle sofort beim Beenden der Umgebung in CloudWatch Logs zu löschen, anstatt das Ablaufdatum abzuwarten.

7. Wählen Sie Apply (Anwenden) aus.

Nachdem Sie das Protokoll-Streaming aktiviert haben, können Sie zur Konfigurationskategorie oder -seite Software zurückkehren und dort den Link Log Groups (Protokollgruppen) suchen. Klicken Sie auf diesen Link, um Ihre Instance-Protokolle in der CloudWatch-Konsole anzuzeigen.



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 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

Konfigurieren der Anzeige von Umgebungszustandsprotokollen

Um Umgebungszustandsprotokolle anzuzeigen, können Sie in der Elastic Beanstalk-Konsole das Streamen der Umgebungszustandsprotokolle aktivieren.

So konfigurieren Sie das Streamen von Umgebungszustandsprotokollen mit der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

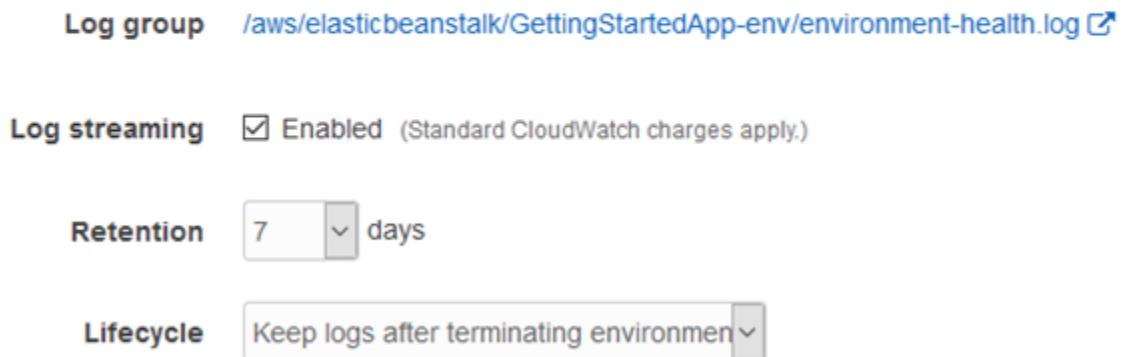
Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Monitoring (Überwachung) die Option Edit (Bearbeiten).
5. Konfigurieren Sie unter Health event streaming to CloudWatch Logs (Streaming von Zustandsereignissen nach CloudWatch Logs) die folgenden Einstellungen:
 - Log streaming (Protokollstreaming) – Wählen Sie diese Option aus, um das Streamen von Protokollen zu aktivieren.
 - Retention (Aufbewahrung) – Geben Sie die Anzahl von Tagen an, für die Protokolle in CloudWatch Logs aufbewahrt werden.
 - Lifecycle (Lebenszyklus) – Wählen Sie die Option Delete logs upon termination (Protokolle beim Beenden löschen) aus, um Protokolle sofort beim Beenden der Umgebung in CloudWatch Logs zu löschen, anstatt das Ablaufdatum abzuwarten.
6. Wählen Sie Apply (Anwenden) aus.

Nachdem Sie das Protokoll-Streaming aktiviert haben, können Sie zur Konfigurationskategorie oder -seite Monitoring (Überwachung) zurückkehren und dort den Link Log Group (Protokollgruppe) suchen. Klicken Sie auf diesen Link, um Ihre Umgebungszustandsprotokolle in der CloudWatch-Konsole anzuzeigen.

Health event streaming to CloudWatch Logs

Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention period to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.



Namespace für das Anzeigen von Protokolldateien

Die folgenden Namespaces enthalten Einstellungen für die Protokollanzeige:

- [aws:elasticbeanstalk:hostmanager](#) (p. 693) – Konfiguration für das Hochladen von rotierten Protokollen in Amazon S3.
- [aws:elasticbeanstalk:cloudwatch:logs](#) (p. 684) – Konfiguration für das Streamen von Instance-Protokollen an CloudWatch.
- [aws:elasticbeanstalk:cloudwatch:logs:health](#) (p. 684) – Konfiguration für das Streamen des Umgebungszustands an CloudWatch.

Elastic Beanstalk-Umgebungsbenachrichtigungen mit Amazon SNS

Sie können Ihre AWS Elastic Beanstalk-Umgebung so konfigurieren, dass sie Amazon Simple Notification Service (Amazon SNS) verwendet, um Sie über wichtige Ereignisse zu benachrichtigen, die Ihre Anwendung betreffen. Geben Sie eine E-Mail-Adresse während oder nach der Umgebungserstellung an, um E-Mails von AWS zu erhalten, wenn ein Fehler auftritt oder sich der Anwendungsstatus Ihrer Umgebung ändert.

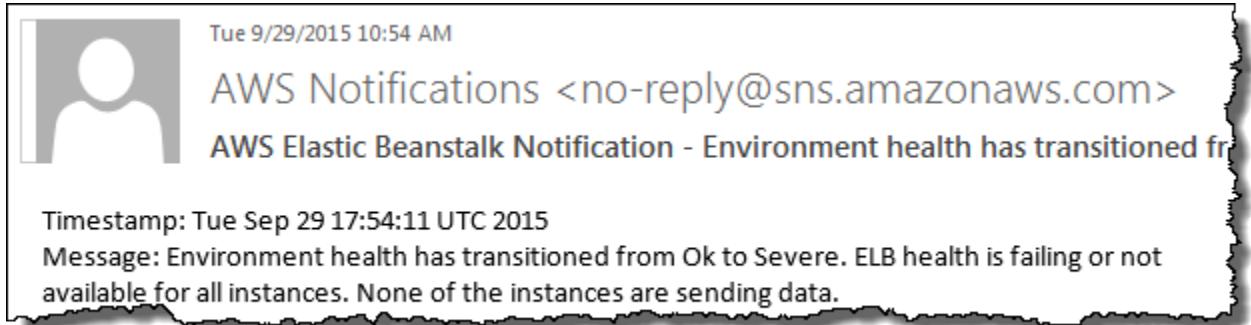
Note

Elastic Beanstalk verwendet Amazon SNS für Benachrichtigungen. Einzelheiten zu den Amazon SNS-Preisen finden Sie unter <https://aws.amazon.com/sns/pricing/>.

Wenn Sie Benachrichtigungen für Ihre Umgebung konfigurieren, erstellt Elastic Beanstalk ein Amazon SNS-Thema für Ihre Umgebung. Um Nachrichten an ein Amazon SNS-Thema zu senden, muss Elastic Beanstalk über die erforderliche Berechtigung verfügen. Details dazu finden Sie unter [Konfigurieren von Berechtigungen zum Senden von Benachrichtigungen](#) (p. 636).

Wenn ein relevantes [Ereignis](#) (p. 864) eintritt, sendet Elastic Beanstalk eine Nachricht an das Thema. Amazon SNS leitet empfangene Nachrichten an die Abonnenten des Themas weiter. Zu den besonderen

Ereignissen zählen Fehler bei der Umgebungserstellung und alle Änderungen am [Umgebungs- und Instance-Zustand](#) (p. 824). Ereignisse für Amazon EC2 Auto Scaling-Vorgänge (Hinzufügen oder Entfernen von Instances zu bzw. aus der Umgebung) und andere Informationsereignisse lösen keine Benachrichtigungen aus.



Mit der Elastic Beanstalk-Konsole können Sie während oder nach der Umgebungserstellung eine E-Mail-Adresse eingeben, um ein Amazon SNS-Thema zu erstellen und dieses zu abonnieren. Elastic Beanstalk verwaltet den Lebenszyklus des Themas, löscht dieses nach Abschluss aus der Umgebung oder wenn Sie Ihre E-Mail-Adresse aus der [Environment Management Console](#) (p. 426) entfernen.

Der Namespace `aws:elasticbeanstalk:sns:topics` bietet Optionen zum Konfigurieren eines Amazon SNS-Themas mit Konfigurationsdateien oder durch Verwendung einer CLI oder eines SDKs. Bei diesen Methoden können Sie die Art des Abonnenten sowie den Endpunkt konfigurieren, sodass der Abonnent eine Amazon SQS-Warteschlange oder HTTP-URL sein kann.

Sie können Amazon SNS-Benachrichtigungen lediglich aktivieren oder deaktivieren. Abhängig von der Größe und Zusammensetzung Ihrer Umgebung, kann die Häufigkeit der Benachrichtigungen, die an das Thema gesendet werden, groß sein. Bei Benachrichtigungen, die nur unter bestimmten Umständen gesendet werden, können Sie [Ihre Umgebung so konfigurieren, dass benutzerdefinierte Metriken veröffentlicht werden](#) (p. 849), und [Amazon CloudWatch-Alarne festlegen](#) (p. 861), die Sie benachrichtigen, wenn diese Metriken einen Unhealthy Threshold (Schwellenwert für fehlerhaften Zustand) erreichen.

Benachrichtigungen mit der Elastic Beanstalk-Konsole konfigurieren

In der Elastic Beanstalk-Konsole können Sie eine E-Mail-Adresse eingeben, um ein Amazon SNS-Thema für Ihre Umgebung zu erstellen.

So konfigurieren Sie Benachrichtigungen in der Elastic Beanstalk-Konsole:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Notifications (Benachrichtigungen) die Option Edit (Bearbeiten).
5. Geben Sie eine E-Mail-Adresse ein.

Elastic Beanstalk > Environments > GettingStartedApp-env > Configuration

Modify notifications

Email notifications

Enter an email address to receive email notifications for important events from your environment.

Email

user@example.com

Cancel Apply

6. Wählen Sie **Apply** (Anwenden) aus.

Wenn Sie eine E-Mail-Adresse für Benachrichtigungen eingeben, erstellt Elastic Beanstalk ein Amazon SNS-Thema für Ihre Umgebung und fügt ein Abonnement hinzu. Amazon SNS sendet eine E-Mail an die registrierte Adresse, um das Abonnement zu bestätigen. Zum Aktivieren des Abonnements und zum Empfangen von Benachrichtigungen müssen Sie auf den Link in der Bestätigungs-E-Mail klicken.

Konfigurieren von Benachrichtigungen mithilfe von Konfigurationsoptionen

Verwenden Sie die Optionen im [aws:elasticbeanstalk:sns:topics-Namespace](#) (p. 695), um Amazon SNS-Benachrichtigungen für Ihre Umgebung zu konfigurieren. Sie können diese Optionen über [Konfigurationsdateien](#) (p. 722), eine CLI oder ein SDK festlegen.

Notification Endpoint – E-Mail-Adresse, Amazon SQS-Warteschlange oder URL, an die Benachrichtigungen gesendet werden sollen. Mit dieser Option wird eine SQS-Warteschlange und ein Abonnement für den angegebenen Endpunkt erstellt. Wenn der Endpunkt keine E-Mail-Adresse ist, müssen Sie zudem die **Notification Protocol**-Option festlegen. SNS validiert den Wert von **Notification Endpoint** basierend auf dem Wert von **Notification Protocol**. Wenn Sie diese Option mehrfach festlegen, werden zusätzliche Abonnements für das Thema erstellt. Wenn Sie diese Option entfernen, wird das Thema gelöscht.

Notification Protocol – Das Protokoll, das zum Senden von Benachrichtigungen an den **Notification Endpoint** verwendet wird. Diese Option lautet standardmäßig `email`. Legen Sie diese Option auf `email-json` fest, um JSON-formatierte E-Mails zu senden, auf `http` oder `https`, um JSON-formatierte Benachrichtigungen an einen HTTP-Endpunkt zu posten, oder auf `sqs`, um Benachrichtigungen an eine SQS-Warteschlange zu übermitteln.

Note

AWS Lambda-Benachrichtigungen werden nicht unterstützt.

Notification Topic ARN – Lesen Sie nach dem Festlegen eines Benachrichtigungsendpunkts für Ihre Umgebung diese Einstellung, um den ARN des SNS-Themas abzurufen. Sie können diese Option auch festlegen, um ein bestehendes SNS-Thema für Benachrichtigungen zu verwenden. Ein Thema, das Sie mit dieser Option an Ihre Umgebung anfügen, wird nicht gelöscht, wenn diese Option geändert oder die Umgebung beendet wird.

Notification Topic Name – Legen Sie diese Option fest, um den Namen des Amazon SNS-Themas anzupassen, der für Umgebungsbenachrichtigungen verwendet wird. Wenn ein Thema mit demselben Namen bereits vorhanden ist, fügt Elastic Beanstalk dieses Thema an die Umgebung an.

Warning

Wenn Sie ein vorhandenes SNS-Thema mit **Notification Topic Name** an eine Umgebung anfügen, löscht Elastic Beanstalk das Thema, wenn Sie die Umgebung beenden oder diese Einstellung ändern.

Wenn Sie diese Option ändern, ändert sich auch der **Notification Topic ARN**. Wenn ein Thema bereits an die Umgebung angefügt ist, löscht Elastic Beanstalk das alte Thema und erstellt dann ein neues Thema und Abonnement.

Die EB CLI und Elastic Beanstalk-Konsole wenden die empfohlenen Werte für die vorhergehenden Optionen an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter [Empfohlene Werte \(p. 648\)](#).

Konfigurieren von Berechtigungen zum Senden von Benachrichtigungen

In diesem Abschnitt werden Sicherheitsaspekte im Hinblick auf Benachrichtigungen mit Amazon SNS beschrieben. Es gibt zwei verschiedene Fälle: unter Verwendung des standardmäßigen Amazon SNS-Themas, das Elastic Beanstalk für Ihre Umgebung erstellt, und durch Bereitstellung eines externen Amazon SNS-Themas über die Konfigurationsoptionen.

Berechtigungen für ein Standard-Thema

Wenn Sie Benachrichtigungen für Ihre Umgebung konfigurieren, erstellt Elastic Beanstalk ein Amazon SNS-Thema für Ihre Umgebung. Um Nachrichten an ein Amazon SNS-Thema zu senden, muss Elastic Beanstalk über die erforderliche Berechtigung verfügen. Wenn die Umgebung die [Service-Rolle \(p. 909\)](#), die die Elastic Beanstalk-Konsole oder die EB-CLI für die Umgebung generiert hat, oder die [Service-verknüpfte Überwachungsrolle \(p. 916\)](#) des Kontos verwendet, sind keine weiteren Schritte erforderlich. Diese verwalteten Rollen beinhalten die erforderlichen Berechtigungen.

Wenn Sie jedoch beim Erstellen Ihrer Umgebung eine benutzerdefinierte Service-Rolle bereitgestellt haben, stellen Sie sicher, dass diese benutzerdefinierte Service-Rolle die folgende Richtlinie enthält.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "sns:Publish"  
            ],  
            "Resource": [  
                "arn:aws:sns:us-east-2:123456789012:ElasticBeanstalkNotifications*"  
            ]  
        }  
    ]  
}
```

```
    ]  
}
```

Berechtigungen für ein externes Thema

In [Konfigurieren von Benachrichtigungen mithilfe von Konfigurationsoptionen \(p. 635\)](#) erklären wir, wie Sie das Amazon SNS-Thema von Elastic Beanstalk durch ein anderes Amazon SNS-Thema ersetzen können. Nachdem Sie das gemacht haben, muss Elastic Beanstalk prüfen, ob Sie über die Berechtigung zum Veröffentlichen an dieses SNS-Thema verfügen, um Ihnen zu gestatten, der Umgebung das SNS-Thema zuzuordnen. Sie sollten über dieselbe Berechtigung wie die Service-Rolle verfügen, `sns:Publish`. Um dies zu überprüfen, sendet Elastic Beanstalk als Teil Ihrer Aktion für das Erstellen oder Aktualisieren der Umgebung eine Test-Benachrichtigung an SNS. Schlägt diese Prüfung fehl, schlägt Ihr Versuch, die Umgebung zu erstellen oder zu aktualisieren, fehl, und Elastic Beanstalk zeigt eine Meldung an, die den Fehler erklärt.

Wenn Sie eine benutzerdefinierte Service-Rolle für Ihre Umgebung bereitstellen, stellen Sie sicher, dass Ihre benutzerdefinierte Service-Rolle die folgende Richtlinie enthält. Ersetzen Sie `sns_topic_name` durch den Namen des Amazon SNS-Themas, das Sie in den Konfigurationsoptionen bereitgestellt haben.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "sns:Publish"  
            ],  
            "Resource": [  
                "arn:aws:sns:us-east-2:123456789012:sns_topic_name"  
            ]  
        }  
    ]  
}
```

Amazon Virtual Private Cloud (Amazon VPC) mit Elastic Beanstalk konfigurieren

[Amazon Virtual Private Cloud](#) (Amazon VPC) ist der Netzwerk-Service, der Datenverkehr sicher zu den EC2-Instances leitet, auf denen Ihre Elastic Beanstalk-Anwendung läuft. Wenn Sie beim Start Ihrer Umgebung keine VPC konfigurieren, verwendet Elastic Beanstalk die standardmäßige VPC.

Sie können Ihre Umgebung in einer benutzerdefinierten VPC starten, um die Netzwerk- und Sicherheitseinstellungen anzupassen. Mit Elastic Beanstalk können Sie wählen, welche Subnetze Sie für Ihre Ressourcen verwenden möchten und wie IP-Adressen für die Instances und Load Balancer in Ihrer Umgebung konfiguriert werden sollen. Eine Umgebung ist für eine VPC gesperrt, wenn Sie sie erstellen. Sie können aber Subnetz- und IP-Adresseinstellungen in einer ausgeführten Umgebung ändern.

Note

Wenn Sie Ihr AWS-Konto vor dem 4. Dezember 2013 erstellt haben, verfügen Sie möglicherweise über Umgebungen, die in einigen AWS-Regionen die Amazon EC2-Classic-Netzwerkkonfiguration anstelle von Amazon VPC verwenden. Weitere Informationen zum Migrieren Ihrer Umgebungen von einer EC2-Classic- zu einer VPC-Netzwerkkonfiguration finden Sie unter [Elastic Beanstalk-Umgebungen von EC2-Classic auf eine VPC migrieren \(p. 641\)](#).

VPC-Einstellungen in der Elastic Beanstalk-Konsole konfigurieren

Wenn Sie beim Erstellen der Umgebung eine benutzerdefinierte VPC ausgewählt haben, können Sie die VPC-Einstellungen in der Elastic Beanstalk-Konsole ändern.

So konfigurieren Sie die VPC-Einstellungen Ihrer Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Network (Netzwerk) die Option Edit (Bearbeiten).

Die folgenden Einstellungen sind verfügbar.

Optionen

- [VPC \(p. 638\)](#)
- [Load Balancer-Sichtbarkeit \(p. 639\)](#)
- [Load Balancer-Subnetze \(p. 639\)](#)
- [Öffentliche IP-Adresse der Instance \(p. 639\)](#)
- [Instance-Subnetze \(p. 639\)](#)
- [Datenbanksubnetze \(p. 640\)](#)

VPC

Wählen Sie eine VPC für die Umgebung. Sie können diese Einstellung nur während des Erstellens der Umgebung ändern.

Elastic Beanstalk > Environments > GettingStartedApp-env > Configuration

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.

vpc-0f9c96ae77f3c49c1 (172.31.0.0/16) | private-public



[Create custom VPC](#)

Load Balancer-Sichtbarkeit

Wählen Sie für eine Load Balancer-Umgebung das Load Balancer-Schema. Standardmäßig ist der Load Balancer öffentlich mit einer öffentlichen IP-Adresse und einem öffentlichen Domänennamen. Wenn Ihre Anwendung nur Datenverkehr innerhalb Ihrer VPC oder über eine verbundene VPN abwickelt, deaktivieren Sie diese Option und wählen Sie private Subnetze für Ihren Load Balancer aus, um den Load Balancer als intern zu kennzeichnen und den Zugriff über das Internet zu deaktivieren.

Load Balancer-Subnetze

Wählen Sie für eine Load Balancer-Umgebung die Subnetze, die der Load Balancer verwendet, um den Datenverkehr abzuwickeln. Wählen Sie für eine öffentliche Anwendung öffentliche Subnetze. Verwenden Sie Subnetze in mehreren Availability Zones, um eine hohe Verfügbarkeit zu erzielen. Wählen Sie für eine interne Anwendung private Subnetze und deaktivieren Sie die Load Balancer-Sichtbarkeit.

The screenshot shows the 'Load balancer settings' section of the AWS Elastic Beanstalk console. It includes a note about assigning subnets to Availability Zones and a dropdown menu for visibility ('Public'). Below is a table for selecting load balancer subnets across two Availability Zones (us-east-2a and us-east-2b). Subnets are identified by their names and associated CIDR ranges.

	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-0c559eeeb1a89adb4	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

Öffentliche IP-Adresse der Instance

Wenn Sie öffentliche Subnetze für Ihre Anwendungs-Instances wählen, aktivieren Sie öffentliche IP-Adressen, damit diese über das Internet weitergeleitet werden können.

Instance-Subnetze

Wählen Sie Subnetze für Ihre Anwendungs-Instances. Wählen Sie mindestens ein Subnetz für jede Availability Zone, die Ihr Load Balancer verwendet. Wenn Sie private Subnetze für Ihre Instances

verwenden, muss Ihre VPC über einen NAT-Gateway in einem öffentlichen Subnetz verfügen, den die Instances für den Internetzugriff verwenden können.

The screenshot shows the 'Instance settings' configuration dialog for AWS Elastic Beanstalk. At the top, there is a note about choosing subnets for instances. Below this, there is a section for selecting a 'Public IP address'. A checkbox labeled 'Public IP address' is checked, with a note below it stating: 'Assign a public IP address to the Amazon EC2 instances in your environment.' The main part of the dialog is a table titled 'Instance subnets' showing the following data:

Availability Zone	Subnet	CIDR	Name
us-east-2a	subnet-04a707767b8ca8023	172.31.0.0/24	public-a
us-east-2a	subnet-0c559eeeb1a89adb4	172.31.3.0/24	private-a
us-east-2b	subnet-034a813125cd2077a	172.31.2.0/24	private-b
us-east-2b	subnet-09a24e24e7f7359fa	172.31.1.0/24	public-b

At the bottom right of the dialog are 'Cancel' and 'Create' buttons.

Datenbanksubnetze

Wenn Sie eine Amazon RDS-Datenbank ausführen, die Ihrer Elastic Beanstalk-Umgebung angehängt ist, wählen Sie Subnetze für Ihre Datenbank-Instances aus. Hohe Verfügbarkeit erzielen Sie, indem Sie die Datenbank zu einer Multi-AZ-Datenbank machen und ein Subnetz für jede Availability Zone wählen. Um sicherzustellen, dass Ihre Anwendung eine Verbindung zu Ihrer Datenbank herstellen kann, führen Sie beide in denselben Subnetzen aus.

Der aws:ec2:vpc-Namespace

Sie können die Konfigurationsoptionen im [aws:ec2:vpc](#) (p. 682)-Namespace für die Konfiguration der Netzwerkeinstellungen Ihrer Umgebung verwenden.

Die folgende [Konfigurationsdatei](#) (p. 722) verwendet Optionen in diesem Namespace, um die VPC und Subnetze der Umgebung für eine öffentlich-private Konfiguration festzulegen. Zum Festlegen der VPC-ID in einer Konfigurationsdatei muss die Datei im Anwendungsquellen-Bundle während der Erstellung der Umgebung enthalten sein. Unter [Festlegen von Konfigurationsoptionen während der Umgebungserstellung](#) (p. 653) finden Sie weitere Methoden zum Konfigurieren dieser Einstellungen während des Erstellens der Umgebung.

Example .ebextensions/vpc.config – Public-private

```
option_settings:  
  aws:ec2:vpc:  
    VPCId: vpc-087a68c03b9c50c84  
    AssociatePublicIpAddress: 'false'  
    ELBScheme: public  
    ELBSubnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2  
    Subnets: subnet-026c6117b178a9c45,subnet-0839e902f656e8bd1
```

Dieses Beispiel zeigt eine öffentlich-öffentliche Konfiguration, bei der Load Balancer- und EC2-Instances in denselben öffentlichen Subnetzen ausgeführt werden.

Example .ebextensions/vpc.config – Public-public

```
option_settings:  
  aws:ec2:vpc:  
    VPCId: vpc-087a68c03b9c50c84  
    AssociatePublicIpAddress: 'true'  
    ELBScheme: public  
    ELBSubnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2  
    Subnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
```

Elastic Beanstalk-Umgebungen von EC2-Classic auf eine VPC migrieren

Dieses Thema beschreibt die Optionen für die Migration Ihrer Elastic Beanstalk-Umgebungen von einer EC2-Classic-Netzwerkplattform zu einem [Amazon Virtual Private Cloud](#) (Amazon VPC)-Netzwerk.

Wenn Sie Ihr AWS-Konto vor dem 4. Dezember 2013 erstellt haben, verfügen Sie möglicherweise über Umgebungen, die in einigen AWS-Regionen die EC2-Classic-Netzwerkkonfiguration verwenden.

Note

Sie können die Konfigurationseinstellungen für das Netzwerk Ihrer Umgebung in der Kategorie Network configuration (Netzwerkkonfiguration) auf der Seite [Configuration overview \(Konfigurationsübersicht\) \(p. 527\)](#) in der [Elastic Beanstalk-Konsole](#) anzeigen.

Warum Sie migrieren sollten

In den meisten Fällen sollten Sie Ihr AWS-Konto auf der EC2-Classic-Plattform zu einem neuen AWS-Konto migrieren, das VPCs unterstützt. Alle AWS-Konten, die am oder nach dem 4. Dezember 2013 erstellt wurden, verfügen in jeder AWS-Region über eine Standard-VPC. Sie werden vom Zugriff auf die neueste AWS-Funktionalität und die neuesten unterstützten Funktionen profitieren.

Dazu müssen Sie ein neues AWS-Konto erstellen und Ihre AWS EC2-Classic-Umgebungen in Ihrem neuen AWS-Konto neu erstellen. Sie haben keinen zusätzlichen Konfigurationsaufwand für Ihre Umgebungen, um die Standard-VPC zu verwenden. Wenn die Standard-VPC Ihren Anforderungen nicht entspricht, können Sie manuell eine benutzerdefinierte VPC erstellen und sie Ihren Umgebungen zuordnen.

Wenn das vorhandene AWS-Konto über Ressourcen verfügt, die Sie nicht zu einem neuen AWS-Konto migrieren können, können Sie Ihrem aktuellen Konto auch eine VPC hinzufügen und Ihre Umgebungen für ihre Verwendung konfigurieren.

Migration einer Umgebung aus EC2-Classic zu einem neuen AWS-Konto (empfohlen)

Wenn Sie noch kein AWS-Konto besitzen, das am oder nach dem 4. Dezember 2013 erstellt wurde, müssen Sie zunächst ein neues Konto erstellen. Sie werden Ihre Umgebungen zu diesem neuen Konto migrieren.

1. Ihr neues AWS-Konto stellt eine Standard-VPC für seine Umgebungen bereit. Wenn Sie keine benutzerdefinierte VPC erstellen müssen, fahren Sie mit Schritt 2 fort.

Sie können eine benutzerdefinierte VPC auf eine der folgenden Arten erstellen:

- Über den Amazon VPC-Konsolenassistent können Sie mithilfe einer der verfügbaren Konfigurationsoptionen schnell eine VPC einrichten. Weitere Informationen finden Sie unter [Konfigurationen über den Amazon VPC-Konsolenassistent](#).
- Wenn Sie spezifischere Anforderungen an Ihre VPC haben, z. B. eine bestimmte Anzahl von Subnetzen, können Sie auch die Amazon VPC-Konsole verwenden. Weitere Informationen finden Sie unter [VPCs und Subnetze](#).
- Wenn Sie es vorziehen, AWS CloudFormation-Vorlagen mit Ihren Elastic Beanstalk-Umgebungen zu verwenden, bietet das [elastic-beanstalk-samples](#)-Repository auf der GitHub-Website AWS CloudFormation-Vorlagen, die Sie zur Erstellung einer VPC verwenden können. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon VPC \(p. 988\)](#).

Note

Alternativ können Sie eine benutzerdefinierte VPC während der Neuerstellung der Umgebung in Ihrem neuen AWS-Konto erstellen, indem Sie den [Assistenten zum Erstellen neuer Umgebungen \(p. 437\)](#) verwenden. Der Assistent ermöglicht Ihnen die Erstellung einer benutzerdefinierten VPC und leitet Sie zur Amazon VPC-Konsole, wenn Sie dies auswählen.

2. Erstellen Sie im neuen AWS-Konto eine neue Umgebung mit der Konfiguration der Umgebung in dem AWS-Konto, aus dem Sie die Migration ausführen. Sie können dies auf eine der folgenden Arten ausführen.

Note

Wenn Ihre neue Umgebung nach der Migration weiter denselben CNAME verwenden muss, müssen Sie zunächst die ursprüngliche Umgebung auf der EC2-Classic-Plattform beenden, um den CNAME zur Verwendung freizugeben. Dies führt zu einer Ausfallzeit für diese Umgebung und birgt das Risiko, dass ein anderer Kunde in der Zeit zwischen der Beendigung Ihrer EC2-Classic-Umgebung und der Erstellung der neuen Umgebung Ihren CNAME auswählt. Weitere Informationen finden Sie unter [Elastic Beanstalk-Umgebung terminieren \(p. 459\)](#).

Für Umgebungen mit einem eigenen proprietären Domänennamen gibt es dieses Problem für den CNAME nicht. Sie können einfach Ihr Domain Name System (DNS) aktualisieren, sodass Anfragen an den neuen CNAME weitergeleitet werden.

- Verwenden Sie den [Assistenten für die Erstellung einer neuen Umgebung \(p. 437\)](#) in der [Elastic Beanstalk-Konsole](#). Der Assistent enthält eine Option zum Erstellen einer benutzerdefinierten VPC. Wenn Sie keine benutzerdefinierte VPC erstellen möchten, wird eine Standard-VPC zugewiesen.
- Verwenden Sie die Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI), um Ihre Umgebung in Ihrem neuen AWS-Konto neu zu erstellen. Eines der [Beispiele \(p. 1061\)](#) in der Befehlsbeschreibung eb create zeigt die Erstellung einer Umgebung in einer benutzerdefinierten VPC. Wenn Sie keine VPC-ID angeben, verwendet die Umgebung die Standard-VPC.

Dieser Ansatz ermöglicht Ihnen die Verwendung einer gespeicherten Konfigurationsdatei für beide AWS-Konten, sodass Sie nicht alle Konfigurationsdaten manuell eingeben müssen. Sie müssen die Konfigurationseinstellungen für die EC2-Classic-Umgebung, die Sie migrieren, mit dem Befehl [eb config save \(p. 1050\)](#) speichern. Kopieren Sie die gespeicherte Konfigurationsdatei in ein neues Verzeichnis für die Umgebung im neuen Konto.

Note

Sie müssen einige Daten in der gespeicherten Konfigurationsdatei bearbeiten, bevor Sie diese im neuen Konto verwenden können. Sie müssen Daten zu Ihrem alten Konto mit den richtigen Daten für Ihr neues Konto aktualisieren. Sie müssen z. B. den Amazon-Ressourcennamen (ARN) der AWS Identity and Access Management (IAM)-Rolle durch den ARN der IAM-Rolle für das neue Konto ersetzen.

Der Befehl [eb create \(p. 1053\)](#) mit der Option `cfg` erstellt die neue Umgebung mit der angegebenen gespeicherten Konfigurationsdatei. Weitere Informationen finden Sie unter [Verwenden gespeicherter Elastic Beanstalk-Konfigurationen \(p. 764\)](#).

Migration einer Umgebung aus EC2-Classic innerhalb desselben AWS-Kontos

Ihr vorhandenes AWS-Konto enthält möglicherweise Ressourcen, die Sie nicht zu einem neuen AWS-Konto migrieren können. Sie müssen Ihre Umgebungen neu erstellen und für jede von Ihnen erstellte Umgebung eine VPC manuell konfigurieren.

Migration Ihrer Umgebungen zu einer benutzerdefinierten VPC

Voraussetzungen

Bevor Sie beginnen, müssen Sie über eine VPC verfügen. Sie können eine nicht standardmäßige (benutzerdefinierte) VPC auf eine der folgenden Arten erstellen:

- Über den Amazon VPC-Konsolenassistent können Sie mithilfe einer der verfügbaren Konfigurationsoptionen schnell eine VPC einrichten. Weitere Informationen finden Sie unter [Konfigurationen über den Amazon VPC-Konsolenassistent](#).
- Wenn Sie spezifischere Anforderungen an Ihre VPC haben, z. B. eine bestimmte Anzahl von Subnetzen, können Sie auch die Amazon VPC-Konsole verwenden. Weitere Informationen finden Sie unter [VPCs und Subnetze](#).
- Wenn Sie es vorziehen, AWS CloudFormation-Vorlagen mit Ihren Elastic Beanstalk-Umgebungen zu verwenden, bietet das [elastic-beanstalk-samples](#)-Repository auf der GitHub-Website AWS CloudFormation-Vorlagen, die Sie zur Erstellung einer VPC verwenden können. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon VPC \(p. 988\)](#).

In den folgenden Schritten verwenden Sie die generierte VPC-ID und die generierten Subnetz-IDs, wenn Sie die VPC in der neuen Umgebung konfigurieren.

1. Erstellen Sie eine neue Umgebung mit der gleichen Konfiguration wie Ihre vorhandene Umgebung. Sie können dies auf eine der folgenden Arten ausführen.

Note

Mit der Funktion „Saved Configurations (Gespeicherte Konfigurationen)“ können Sie Ihre Umgebungen im neuen Konto neu erstellen. Weitere Informationen finden Sie unter [Verwenden gespeicherter Elastic Beanstalk-Konfigurationen \(p. 764\)](#).

- Wenden Sie während der Erstellung der neuen Umgebung, in der die VPC verwendet werden wird, über die [Elastic Beanstalk-Konsole](#) eine gespeicherte Konfiguration aus Ihrer EC2-

Classic-Umgebung an. Weitere Informationen finden Sie unter [Verwenden gespeicherter Elastic Beanstalk-Konfigurationen \(p. 764\)](#).

- Führen Sie über die Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI) den Befehl `eb create (p. 1053)` aus, um Ihre Umgebung neu zu erstellen. Geben Sie die Parameter aus der ursprünglichen Umgebung und den VPC-Bezeichner an. Eines der [Beispiele \(p. 1061\)](#) in der Befehlsbeschreibung eb create zeigt die Erstellung einer Umgebung in einer benutzerdefinierten VPC.
 - Das AWS Command Line Interface (AWS CLI) Erstellen Sie Ihre Umgebung neu, indem Sie den Befehl `elasticbeanstalk create-environment` verwenden. Geben Sie die Parameter der ursprünglichen Umgebung mit dem VPC-Bezeichner an. Detaillierte Anweisungen finden Sie unter [Elastic Beanstalk-Umgebungen mit der AWS CLI erstellen \(p. 460\)](#).
2. Tauschen Sie die CNAMEs der alten Umgebung und der entsprechenden neuen Umgebung aus. Auf diese Weise kann die neu erstellte Umgebung mit der vertrauten Adresse referenziert werden. Sie können die EB CLI oder die AWS CLI verwenden.
- Tauschen Sie über die EB CLI die CNAMEs der Umgebungen mithilfe des Befehls `eb swap` aus. Weitere Informationen finden Sie unter [Verwenden des Elastic Beanstalk Command Line Interface \(EB CLI\) \(p. 1008\)](#).
 - Tauschen Sie über die AWS CLI die CNAMEs der Umgebungen mithilfe des Befehls `elasticbeanstalk swap-environment-cnames` aus. Weitere Informationen finden Sie in der [AWS CLI-Befehlsreferenz](#).

Der Domänenname Ihrer Elastic Beanstalk-Umgebung

Standardmäßig steht Ihre Umgebung Benutzern in einer Subdomäne von `elasticbeanstalk.com` zur Verfügung. Wenn Sie [eine Umgebung erstellen \(p. 435\)](#), können Sie einen Hostnamen für Ihre Anwendung auswählen. Die Subdomäne und die Domäne werden automatisch in `region.elasticbeanstalk.com` eingetragen.

Um Benutzer zu Ihrer Umgebung weiterzuleiten, registriert Elastic Beanstalk einen CNAME-Eintrag, der auf den Load Balancer Ihrer Umgebung verweist. Sie können die URL der Anwendung Ihrer Umgebung mit dem aktuellen Wert des CNAME auf der [Umgebungsübersichtsseite \(p. 427\)](#) der Elastic Beanstalk-Konsole anzeigen.

Elastic Beanstalk > Environments > GettingStartedApp-env

GettingStartedApp-env

GettingStartedApp-env.bx7dx222kw.us-east-2.elasticbeanstalk.com

Health

Ok

Running version

Sample Application-3

Upload and deploy

Wählen Sie die URL auf der Übersichtsseite oder wählen Sie im Navigationsbereich die Option Go to environment (Zur Umgebung wechseln), um zur Webseite Ihrer Anwendung zu navigieren.

Sie können den CNAME-Wert der Umgebung ändern, indem Sie diesen mit dem CNAME einer anderen Umgebung austauschen. Detaillierte Anweisungen finden Sie unter [Blau/Grün-Bereitstellungen mit AWS Elastic Beanstalk \(p. 481\)](#).

Wenn Sie einen Domänennamen besitzen, können Sie Amazon Route 53 verwenden, um diesen in die Umgebung aufzulösen. Sie können einen Domänennamen mit Amazon Route 53 erwerben oder den Domänennamen verwenden, den Sie von einem anderen Anbieter gekauft haben.

Informationen zum Erwerb eines Domänenamens mit Route 53 finden Sie unter [Registrieren einer neuen Domäne](#) im Amazon Route 53-Entwicklerhandbuch.

Weitere Informationen zur Verwendung einer benutzerdefinierten Domäne finden Sie unter [Weiterleiten des Datenverkehrs an eine AWS Elastic Beanstalk-Umgebung](#) im Amazon Route 53-Entwicklerhandbuch.

Important

Wenn Sie eine Umgebung beenden, müssen Sie auch alle CNAME-Zuordnungen löschen, die Sie erstellt haben, damit andere Kunden einen verfügbaren Hostnamen wiederverwenden können.

Konfigurieren von Elastic Beanstalk-Umgebungen (Erweitert)

Beim Erstellen einer AWS Elastic Beanstalk-Umgebung werden alle AWS-Ressourcen, die für die Anwendungsausführung und -unterstützung erforderlich sind, von Elastic Beanstalk bereitgestellt und konfiguriert. Neben den Metadaten und dem Aktualisierungsverhalten der Umgebung können Sie auch diese Ressourcen anpassen, indem Sie Werte für [Konfigurationsoptionen \(p. 646\)](#) angeben. Beispielsweise können Sie eine Amazon SQS-Warteschlange und einen Alarm für die Warteschlange oder einen Amazon ElastiCache-Cluster hinzufügen.

Die meisten Konfigurationsoptionen haben Standardwerte, die automatisch von Elastic Beanstalk übernommen werden. Sie können diese Standardwerte mit Konfigurationsdateien, gespeicherten Konfigurationen, Befehlszeilenoptionen oder mit dem direkten Aufruf der Elastic Beanstalk-API überschreiben. Die EB CLI und die Elastic Beanstalk-Konsole übernehmen ebenfalls empfohlene Werte für einige Optionen.

Indem Sie Ihrem Quell-Bundle eine Konfigurationsdatei beifügen, können Sie die Umgebung einfach gleichzeitig mit der Bereitstellung der Anwendungsversion anpassen. Wenn die Software auf der Instance angepasst werden soll, ist der Einsatz einer Konfigurationsdatei sinnvoller, als ein benutzerdefiniertes AMI zu erstellen, da Sie keinen AMI-Satz vorhalten müssen.

Möglicherweise muss die Software, von der Ihre Anwendung abhängig ist, im Rahmen der Anwendungsbereitstellung angepasst und konfiguriert werden. Bei diesen Dateien kann es sich entweder um von der Anwendung benötigte Abhängigkeiten (z. B. zusätzliche Pakete aus dem "yum"-Repository) oder um Konfigurationsdateien handeln (wie z. B. ein Ersatz der Datei "httpd.conf" zum Überschreiben bestimmter Standardeinstellungen von AWS Elastic Beanstalk).

Themen

- [Konfigurationsoptionen \(p. 646\)](#)
- [Erweiterte Umgebungsanpassung mit Konfigurationsdateien \(.ebextensions\) \(p. 722\)](#)
- [Verwenden gespeicherter Elastic Beanstalk-Konfigurationen \(p. 764\)](#)
- [Umgebungsmanifest \(env.yaml\) \(p. 771\)](#)
- [Verwenden eines benutzerdefinierten Amazon Machine Image \(AMI\) \(p. 773\)](#)
- [Bereitstellen von statischen Dateien \(p. 776\)](#)
- [Konfigurieren von HTTPS für Elastic Beanstalk-Umgebung \(p. 778\)](#)

Konfigurationsoptionen

Elastic Beanstalk definiert eine große Anzahl von Konfigurationsoptionen, mithilfe derer Sie das Verhalten Ihrer Umgebung und die darin enthaltenen Ressourcen konfigurieren können. Die Konfigurationsoptionen sind in Namespaces wie `aws:autoscaling:asg` unterteilt, die Optionen für die Auto Scaling-Gruppe einer Umgebung definieren.

Die Elastic Beanstalk-Konsole und die EB CLI legen beim Erstellen einer Umgebung Konfigurationsoptionen fest, einschließlich von Ihnen explizit festgelegter Optionen und [empfohlener Werte \(p. 648\)](#), die vom Client definiert werden. Sie können Konfigurationsoptionen auch in gespeicherten Konfigurationen und Konfigurationsdateien festlegen. Wenn dieselbe Option an mehreren Orten festgelegt wird, wird der zu verwendende Wert durch die [Rangfolge \(p. 647\)](#) bestimmt.

Konfigurationsoptionseinstellungen können im Textformat definiert und vor der Umgebungserstellung gespeichert, während der Umgebungserstellung unter Verwendung eines beliebigen unterstützten Clients übernommen und nach der erfolgten Umgebungserstellung hinzugefügt, geändert oder entfernt werden. Eine detaillierte Aufstellung aller verfügbaren Methoden für die Verwendung von Konfigurationsoptionen in jeder der drei Phasen finden Sie in den folgenden Themen:

- [Festlegen von Konfigurationsoptionen vor der Umgebungserstellung \(p. 649\)](#)
- [Festlegen von Konfigurationsoptionen während der Umgebungserstellung \(p. 653\)](#)
- [Festlegen von Konfigurationsoptionen nach der Umgebungserstellung \(p. 658\)](#)

Eine vollständige Liste der Namespaces und Optionen, einschließlich der Standard- und unterstützten Werte, finden Sie unter [Allgemeine Optionen für alle Umgebungen \(p. 666\)](#) und unter [Plattformspezifische Optionen \(p. 712\)](#).

Precedence

Während der Umgebungserstellung werden die Konfigurationsoptionen mit folgender Rangfolge (von der höchsten bis zur niedrigsten) aus mehreren Quellen übernommen:

- Der Umgebung direkt zugewiesene Einstellungen – Einstellungen, die bei der Umgebungserstellung oder -aktualisierung mit der Elastic Beanstalk-API von einem Client (einschließlich Elastic Beanstalk-Konsole, EB CLI, AWS-CLI und SDKs) angegeben werden. Elastic Beanstalk-Konsole und EB CLI weisen außerdem einige Optionen für dieses Level [empfohlene Werte \(p. 648\)](#) zu, sofern diese nicht überschrieben werden.
 - Gespeicherte Konfigurationen – Optionseinstellungen, die nicht direkt in die Umgebung übernommen, sondern aus einer gespeicherten Konfiguration geladen werden (sofern vorhanden).
 - Konfigurationsdateien (.ebextensions) – Optionseinstellungen, die weder direkt in die Umgebung übernommen noch in einer gespeicherten Konfiguration angegeben, sondern aus Konfigurationsdateien des Ordners .ebextensions im Stamm des Quell-Bundles der Anwendung geladen werden.
- Konfigurationsdateien werden in alphabetischer Reihenfolge ausgeführt. So wird beispielsweise .ebextensions/01run.config vor .ebextensions/02do.config ausgeführt.
- Standardwerte – Wenn für eine Konfigurationsoption ein Standardwert vorhanden ist, wird dieser nur übernommen, sofern diese Option auf keiner der oben genannten Ebenen festgelegt wurde.

Falls dieselbe Konfigurationsoption an mehreren Orten definiert ist, gilt die Einstellung mit der höchsten Priorität. Eine Einstellung wird als Teil der Umgebungskonfiguration gespeichert, sofern sie aus einer gespeicherten Konfiguration geladen oder direkt in der Umgebung bereitgestellt wird. Diese Einstellungen können [mit der AWS CLI \(p. 666\)](#) oder [mit der EB CLI \(p. 663\)](#) entfernt werden.

Einstellungen in Konfigurationsdateien werden nicht direkt in die Umgebung übernommen und können nicht ohne Änderung der Konfigurationsdateien und Bereitstellung einer neuen Anwendungsversion entfernt werden. Wenn Sie eine Einstellung, die mit einer der anderen Methoden übernommen wurde, entfernen, wird dieselbe Einstellung aus den Konfigurationsdateien im Quell-Bundle geladen.

Beispiel: Sie legen im Rahmen der Umgebungserstellung mit der Elastic Beanstalk-Konsole, einer Befehlszeilenoption oder einer gespeicherten Konfiguration die Mindestanzahl der Instances in der Umgebung auf 5 fest. Im Quell-Bundle der Anwendung ist ebenfalls eine Konfigurationsdatei enthalten, in der die Mindestanzahl der Instances mit 2 angegeben wird.

Wenn Sie die Umgebung erstellen, setzt Elastic Beanstalk die Option `MinSize` im Namespace `aws:autoscaling:asg` auf 5. Wenn Sie anschließend die Option aus der Umgebungskonfiguration entfernen, wird der Wert aus der Konfigurationsdatei geladen, sodass die Mindestanzahl der Instances auf 2 festgelegt wird. Falls Sie dann die Konfigurationsdatei aus dem Quell-Bundle entfernen und eine neue Bereitstellung ausführen, verwendet Elastic Beanstalk den Standardwert 1.

Empfohlene Werte

Die Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI) und die Elastic Beanstalk-Konsole stellen für einige Konfigurationsoptionen empfohlene Werte bereit. Diese Werte können von den Standardwerten abweichen und werden im Rahmen der Umgebungserstellung auf API-Ebene festgelegt. Mithilfe der empfohlenen Werte kann Elastic Beanstalk die Standardumgebungskonfiguration optimieren, ohne abwärts inkompatible Änderungen an der API vorzunehmen.

So wird beispielsweise die Konfigurationsoption für den EC2-Instance-Typ sowohl von der EB CLI als auch von der Elastic Beanstalk-Konsole festgelegt (`InstanceType` im Namespace `aws:autoscaling:launchconfiguration`). Jeder Client bietet eine andere Möglichkeit zur Überschreibung der Standardeinstellung. In der Konsole können Sie über ein Dropdown-Menü auf der Seite Configuration Details (Konfigurationsdetails) im Assistenten Create New Environment (Neue Umgebung erstellen) einen anderen Instance-Typ auswählen. In der EB CLI verwenden Sie den Parameter `--instance_type` für [eb create \(p. 1053\)](#).

Da die empfohlenen Werte auf API-Ebene festgelegt werden, überschreiben diese die Werte, die Sie für dieselben Optionen in Konfigurationsdateien oder gespeicherten Konfigurationen definiert haben. Folgende Optionen werden festgelegt:

Elastic Beanstalk-Konsole

- Namespace: `aws:autoscaling:launchconfiguration`
Optionsnamen: `IamInstanceProfile`, `EC2KeyName`, `InstanceType`
- Namespace: `aws:autoscaling:updatepolicy:rollingupdate`
Optionsnamen: `RollingUpdateType` und `RollingUpdateEnabled`
- Namespace: `aws:elasticbeanstalk:application`
Optionsname: Application Healthcheck URL
- Namespace: `aws:elasticbeanstalk:command`
Optionsnamen: `DeploymentPolicy`, `BatchSize` und `BatchSizeType`
- Namespace: `aws:elasticbeanstalk:environment`
Optionsname: ServiceRole
- Namespace: `aws:elasticbeanstalk:healthreporting:system`
Optionsnamen: `SystemType` und `HealthCheckSuccessThreshold`
- Namespace: `aws:elasticbeanstalk:sns:topics`
Optionsname: Notification Endpoint
- Namespace: `aws:elasticbeanstalk:sqsd`
Optionsname: `HttpConnections`
- Namespace: `aws:elb:loadbalancer`
Optionsname: CrossZone
- Namespace: `aws:elb:policies`
Optionsnamen: `ConnectionDrainingTimeout` und `ConnectionDrainingEnabled`

EB CLI

- Namespace: `aws:autoscaling:launchconfiguration`

- Optionsnamen: `IamInstanceProfile`, `InstanceType`
- Namespace: `aws:autoscaling:updatepolicy:rollingupdate`
 - Optionsnamen: `RollingUpdateType` und `RollingUpdateEnabled`
 - Namespace: `aws:elasticbeanstalk:command`
 - Optionsnamen: `BatchSize` und `BatchSizeType`
 - Namespace: `aws:elasticbeanstalk:environment`
 - Optionsname: `ServiceRole`
 - Namespace: `aws:elasticbeanstalk:healthreporting:system`
 - Optionsname: `SystemType`
 - Namespace: `aws:elb:loadbalancer`
 - Optionsname: `CrossZone`
 - Namespace: `aws:elb:policies`
 - Optionsname: `ConnectionDrainingEnabled`

Festlegen von Konfigurationsoptionen vor der Umgebungserstellung

AWS Elastic Beanstalk unterstützt eine große Anzahl von [Konfigurationsoptionen \(p. 646\)](#), mit denen Sie die Einstellungen, die für die Ressourcen in der Umgebung gelten, ändern können. Viele dieser Optionen haben Standardwerte, die Sie überschreiben können, um Ihre Umgebung anzupassen. Andere Optionen lassen sich für die Aktivierung zusätzlicher Funktionen konfigurieren.

Elastic Beanstalk unterstützt zwei Methoden, um die Einstellungen der Konfigurationsoptionen zu speichern. Konfigurationsdateien im YAML- oder JSON-Format können im Quellcode der Anwendung in ein Verzeichnis mit dem Namen `.ebextensions` eingebunden und als Teil des Quell-Bundles der Anwendung bereitgestellt werden. Die Konfigurationsdateien lassen sich lokal erstellen und verwalten.

Gespeicherte Konfigurationen sind Vorlagen, die Sie aus einer ausgeführten Umgebung oder einer JSON-Optionsdatei erstellen und in Elastic Beanstalk speichern. Vorhandene gespeicherte Konfigurationen können auch erweitert werden, um eine neue Konfiguration zu erstellen.

Note

Die in Konfigurationsdateien und gespeicherten Konfigurationen definierten Einstellungen haben eine geringere Priorität als Einstellungen, die während oder nach der Umgebungserstellung konfiguriert wurden, einschließlich der empfohlenen Werte, die von der Elastic Beanstalk-Konsole und der [EB CLI \(p. 1008\)](#) zugewiesen werden. Details dazu finden Sie unter [Precedence \(p. 647\)](#).

Zudem können Optionen auch in Form eines JSON-Dokuments angegeben und im Rahmen einer Umgebungserstellung oder -aktualisierung mit der EB CLI oder der AWS-CLI direkt in Elastic Beanstalk bereitgestellt werden. Werden Optionen auf diese Weise direkt in Elastic Beanstalk bereitgestellt, setzt dies alle anderen Methoden außer Kraft.

Eine vollständige Liste der verfügbaren Optionen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Methoden

- [Konfigurationsdateien \(.ebextensions\) \(p. 650\)](#)
- [Gespeicherte Konfigurationen \(p. 651\)](#)

- [JSON-Dokument \(p. 652\)](#)
- [EB CLI-Konfiguration \(p. 653\)](#)

Konfigurationsdateien (`.ebextensions`)

Konfigurieren Sie mit `.ebextensions` die Optionen, die für eine ordnungsgemäße Funktionsweise der Anwendung erforderlich sind, und stellen Sie Standardwerte für andere Optionen bereit, die mit einer höheren [Priorität \(p. 647\)](#) überschrieben werden können. Die in `.ebextensions` spezifizierten Optionen weisen die niedrigste Priorität auf und werden von Einstellungen mit jeder anderen Priorität außer Kraft gesetzt.

Zur Verwendung von Konfigurationsdateien erstellen Sie einen Ordner mit dem Namen `.ebextensions` auf der obersten Ebene des Projektquellcodes. Fügen Sie eine Datei mit der Erweiterung `.config` hinzu und geben Sie die Optionen wie folgt an:

```
option_settings:
  - namespace: namespace
    option_name: option name
    value: option value
  - namespace: namespace
    option_name: option name
    value: option value
```

Beispielsweise wird mit der folgenden Konfigurationsdatei die Zustandsprüfungs-URL für die Anwendung auf `/health` festgelegt:

`healthcheckurl.config`

```
option_settings:
  - namespace: aws:elasticbeanstalk:application
    option_name: Application Healthcheck URL
    value: /health
```

Im JSON-Format:

```
{
  "option_settings" :
  [
    {
      "namespace" : "aws:elasticbeanstalk:application",
      "option_name" : "Application Healthcheck URL",
      "value" : "/health"
    }
  ]
}
```

Damit wird der Elastic Load Balancing Load Balancer in der Elastic Beanstalk-Umgebung so konfiguriert, dass eine HTTP-Anforderung über den Pfad `/health` an die einzelnen EC2-Instances gesendet wird, um deren Zustand zu bestimmen.

Note

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Binden Sie das Verzeichnis `.ebextensions` in das [Quell-Bundle der Anwendung \(p. 412\)](#) ein und stellen Sie es in einer neuen oder vorhandenen Elastic Beanstalk-Umgebung bereit.

Konfigurationsdateien unterstützen neben `option_settings` noch weitere Abschnitte für die Anpassung der Software und Dateien, die auf den Servern in der Umgebung ausgeführt werden. Weitere Informationen finden Sie unter [.Ebextensions \(p. 722\)](#).

Gespeicherte Konfigurationen

Erstellen Sie eine gespeicherte Konfiguration, um Einstellungen zu speichern, die Sie während oder nach einer Umgebungserstellung mit der Elastic Beanstalk-Konsole, der EB CLI oder der AWS-CLI einer vorhandenen Umgebung zugewiesenen haben. Gespeicherte Konfigurationen gehören zu einer Anwendung und können auf neue oder bestehende Umgebungen für die Anwendung angewendet werden.

Clients

- [Elastic Beanstalk-Konsole \(p. 651\)](#)
- [EB CLI \(p. 651\)](#)
- [AWS-CLI \(p. 652\)](#)

Elastic Beanstalk-Konsole

So erstellen Sie eine gespeicherte Konfiguration (Elastic Beanstalk-Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) und dann Save configuration (Konfiguration speichern) aus.
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Aktion abzuschließen.

Gespeicherte Konfigurationen werden im S3-Bucket von Elastic Beanstalk in einem Ordner gespeichert, der nach der Anwendung benannt ist. Beispielsweise befinden sich Konfigurationen für die Anwendung `my-app` in der Region us-west-2 für die Kontonummer 123456789012 unter `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app`.

EB CLI

Die [EB CLI \(p. 1008\)](#) bietet unter [eb config \(p. 1050\)](#) auch Unterbefehle für die Interaktion mit gespeicherten Konfigurationen:

So erstellen Sie eine gespeicherte Konfiguration (EB CLI)

1. Speichern Sie die zugeordnete aktuelle Konfiguration der Umgebung:

```
~/project$ eb config save --cfg my-app-v1
```

In der EB CLI wird die Konfiguration unter `~/project/.elasticbeanstalk/saved_configs/my-app-v1.cfg.yml` gespeichert.

2. Nehmen Sie bei Bedarf lokale Änderungen an der gespeicherten Konfiguration vor.
3. Laden Sie die gespeicherte Konfiguration in S3 hoch:

```
~/project$ eb config put my-app-v1
```

AWS-CLI

Erstellen Sie mit `aws elasticbeanstalk create-configuration-template` eine gespeicherte Konfiguration aus einer ausgeführten Umgebung.

So erstellen Sie eine gespeicherte Konfiguration (AWS CLI)

1. Ermitteln Sie die Umgebungs-ID der Elastic Beanstalk-Umgebung mit `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. Speichern Sie die aktuelle Konfiguration der Umgebung mit `create-configuration-template`:

```
$ aws elasticbeanstalk create-configuration-template --environment-id e-vcghmm2zwk --application-name my-app --template-name v1
```

Elastic Beanstalk speichert die Konfiguration in Ihrem Elastic Beanstalk-Bucket in Amazon S3.

JSON-Dokument

Sofern Sie Umgebungen mit der AWS CLI erstellen und aktualisieren, können Sie auch Konfigurationsoptionen im JSON-Format bereitstellen. Eine Bibliothek mit Konfigurationsdateien im JSON-Format ist hilfreich, wenn Sie Umgebungen mit der AWS CLI erstellen und verwalten.

Beispielsweise wird mit folgendem JSON-Dokument die Zustandsprüfungs-URL für die Anwendung auf /health festgelegt:

```
~/ebconfigs/healthcheckurl.json
```

```
[  
  {  
    "Namespace": "aws:elasticbeanstalk:application",  
    "OptionName": "Application Healthcheck URL",  
    "Value": "/health"  
  }  
]
```

EB CLI-Konfiguration

Die EB CLI unterstützt nicht nur gespeicherte Konfigurationen sowie die direkte Umgebungskonfiguration mit eb config-Befehlen, sondern verfügt auch über eine Konfigurationsdatei. Mithilfe der darin enthaltenen Option default_ec2_keyname können Sie ein Amazon EC2-Schlüsselpaar für den SSH-Zugriff auf Instances in der Umgebung angeben. In der EB CLI wird diese Option verwendet, um die Konfigurationsoption EC2KeyName im Namespace aws:autoscaling:launchconfiguration festzulegen.

~/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
```

Festlegen von Konfigurationsoptionen während der Umgebungserstellung

Wenn Sie eine AWS Elastic Beanstalk-Umgebung mithilfe der Elastic Beanstalk-Konsole, EB CLI, AWS-CLI, eines SDK oder der Elastic Beanstalk-API erstellen, können Sie Werte für Konfigurationsoptionen bereitstellen, um Ihre Umgebung und die darin gestarteten AWS-Ressourcen anzupassen.

Konfigurationsänderungen, die nicht einmalig sind, können Sie in [Konfigurationsdateien speichern \(p. 649\)](#), und zwar lokal, im Quell-Bundle oder in Amazon S3.

In diesem Thema werden die Verfahren aller Methoden zum Festlegen von Konfigurationsoptionen während der Umgebungserstellung dargelegt.

Clients

- [In der Elastic Beanstalk-Konsole \(p. 654\)](#)
- [Verwenden der EB CLI \(p. 655\)](#)
- [Verwenden der AWS CLI \(p. 657\)](#)

In der Elastic Beanstalk-Konsole

Wenn Sie eine Elastic Beanstalk-Umgebung in der Elastic Beanstalk-Konsole erstellen, können Sie Konfigurationsoptionen mit Konfigurationsdateien, gespeicherten Konfigurationen und Formularen im Assistenten Create New Environment (Neue Umgebung erstellen) bereitstellen.

Methoden

- [Verwenden von Konfigurationsdateien \(.ebextensions\) \(p. 654\)](#)
- [Verwenden einer gespeicherten Konfiguration \(p. 654\)](#)
- [Verwenden des Assistenten für die neue Umgebung \(p. 655\)](#)

Verwenden von Konfigurationsdateien (.ebextensions)

Binden Sie .config-Dateien in den Ordner .ebextensions im [Quell-Bundle der Anwendung \(p. 412\)](#) ein.

Weitere Informationen zu Konfigurationsdateien finden Sie unter [.Ebextensions \(p. 722\)](#).

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   '-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Laden Sie das Quell-Bundle wie üblich während der [Umgebungserstellung \(p. 435\)](#) in Elastic Beanstalk hoch.

Die Elastic Beanstalk-Konsole weist einige Konfigurationsoptionen [empfohlene Werte \(p. 648\)](#) zu und stellt für andere Optionen Formularfelder bereit. Über die Elastic Beanstalk-Konsole konfigurierte Optionen werden direkt in der Umgebung bereitgestellt und überschreiben die Einstellungen in den Konfigurationsdateien.

Verwenden einer gespeicherten Konfiguration

Beim Erstellen einer neuen Umgebung mit der Elastic Beanstalk-Konsole besteht einer der ersten Schritte darin, eine Konfiguration auszuwählen. Bei der Konfiguration kann es sich um eine [predefined configuration \(vordefinierte Konfiguration\) \(p. 32\)](#) (in der Regel die neueste Version einer Plattform wie PHP oder Tomcat) oder um eine saved configuration (gespeicherte Konfiguration) handeln.

So wenden Sie eine gespeicherte Konfiguration während der Umgebungserstellung an (Elastic Beanstalk-Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.

Note

Wenn Sie viele Anwendungen haben, verwenden Sie die Suchleiste, um die Anwendungsliste zu filtern.

3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Saved configurations (Gespeicherte Konfigurationen) aus.
4. Wählen Sie die gespeicherte Konfiguration aus, die Sie anwenden möchten, und wählen Sie dann Launch environment (Umgebung starten).

5. Führen Sie die Schritte im Assistenten aus, um die Umgebung zu erstellen.

Gespeicherte Konfigurationen sind anwendungsspezifisch. Weitere Informationen zum Erstellen von gespeicherten Konfigurationen finden Sie unter [Gespeicherte Konfigurationen \(p. 651\)](#).

Verwenden des Assistenten für die neue Umgebung

Die meisten standardmäßigen Konfigurationsoptionen finden Sie auf der Seite [Configure more options \(Weitere Optionen konfigurieren\)](#) des [Create New Environment-Assistenten \(p. 437\)](#). Wenn Sie eine Amazon RDS-Datenbank erstellen oder eine VPC für die Umgebung konfigurieren, stehen für diese Ressourcen zusätzliche Konfigurationsoptionen zur Verfügung.

So legen Sie Konfigurationsoptionen während der Umgebungserstellung fest (Elastic Beanstalk-Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Applications (Anwendungen).
3. Wählen Sie eine Anwendung aus oder [erstellen \(p. 403\)](#) Sie eine.
4. Wählen Sie Actions (Aktionen) und dann Create Environment (Umgebung erstellen).
5. Fahren Sie mithilfe des Assistenten fort und wählen Sie [Configure more options \(Weitere Optionen konfigurieren\)](#) aus.
6. Wählen Sie in einer oder mehreren Konfigurationskategorien die configuration presets (Konfigurationsvoreinstellungen) und dann Edit (Bearbeiten), um eine Gruppe verwandter Konfigurationsoptionen zu ändern.
7. Wählen Sie nach der Optionsauswahl Create environment (Umgebung erstellen) aus.

Alle Optionen, die Sie im Assistenten für die neue Umgebung festgelegt haben, werden direkt in die Umgebung übernommen und überschreiben alle Optionseinstellungen aus den von Ihnen angewendeten gespeicherten Konfigurationen oder Konfigurationsdateien (.ebextensions). Nach der Umgebungserstellung können Sie die Einstellungen mit der [EB CLI \(p. 661\)](#) oder der [AWS CLI \(p. 664\)](#) entfernen, damit die Einstellungen aus den gespeicherten Konfigurationen oder Konfigurationsdateien wirksam werden.

Weitere Informationen über den neuen Umgebungsassistenten finden Sie unter [Der Assistent zum Erstellen einer neuen Umgebung \(p. 437\)](#).

Verwenden der EB CLI

Methoden

- [Verwenden von Konfigurationsdateien \(.ebextensions\) \(p. 655\)](#)
- [Verwenden von gespeicherten Konfigurationen \(p. 656\)](#)
- [Verwenden von Befehlszeilenoptionen \(p. 656\)](#)

Verwenden von Konfigurationsdateien (.ebextensions)

Binden Sie .config-Dateien unter .ebextensions in den Projektordner ein, um sie mit dem Anwendungscode bereitzustellen.

Weitere Informationen zu Konfigurationsdateien finden Sie unter [.Ebextensions \(p. 722\)](#).

```
~/workspace/my-app/  
|-- .ebextensions
```

```
|   |-- environmentvariables.config
|   '-- healthcheckurl.config
|-- .elasticbeanstalk
|   '-- config.yml
|-- index.php
`-- styles.css
```

Erstellen Sie die Umgebung und stellen Sie den entsprechenden Quellcode mit eb create bereit.

```
~/workspace/my-app$ eb create my-env
```

Verwenden von gespeicherten Konfigurationen

Um eine gespeicherte Konfiguration bei der Umgebungserstellung mit [eb create \(p. 1053\)](#) anzuwenden, verwenden Sie die Option --cfg.

```
~/workspace/my-app$ eb create --cfg savedconfig
```

Sie können die gespeicherte Konfiguration in Ihrem Projektordner oder in Ihrem Elastic Beanstalk-Speicherort auf Amazon S3 speichern. In vorherigem Beispiel sucht die EB CLI zunächst nach einer gespeicherten Konfigurationsdatei mit dem Namen savedconfig.cfg.yml im Ordner .elasticbeanstalk/saved_configs/. Geben Sie keine Dateinamenerweiterungen (.cfg.yml) an, wenn Sie eine gespeicherte Konfiguration mit --cfg übernehmen.

```
~/workspace/my-app/
|-- .ebextensions
|   '-- healthcheckurl.config
|-- .elasticbeanstalk
|   '-- saved_configs
|       '-- savedconfig.cfg.yml
|   '-- config.yml
|-- index.php
`-- styles.css
```

Wenn die EB CLI die Konfiguration nicht lokal findet, sucht sie im Elastic Beanstalk-Speicherort in Amazon S3. Weitere Informationen zum Erstellen, Bearbeiten und Hochladen gespeicherter Konfigurationen finden Sie unter [Gespeicherte Konfigurationen \(p. 651\)](#).

Verwenden von Befehlszeilenoptionen

Der eb create-Befehl der EB CLI bietet mehrere [Optionen \(p. 1054\)](#), mit denen sich während der Umgebungserstellung Konfigurationsoptionen festlegen lassen. Mithilfe dieser Optionen können Sie eine RDS-Datenbank zur Umgebung hinzufügen, eine VPC konfigurieren oder [empfohlene Werte \(p. 648\)](#) überschreiben.

Beispielsweise verwendet die EB CLI standardmäßig den Instance-Typ t2.micro. Mit der Option --instance_type können Sie einen anderen Instance-Typ auswählen.

```
$ eb create my-env --instance_type t2.medium
```

Verwenden Sie die --database-Optionen, um eine Amazon RDS-DB-Instance zu erstellen und mit der Umgebung zu verknüpfen.

```
$ eb create --database.engine postgres --database.username dbuser
```

Falls Sie den Umgebungsnamen, das Datenbankpasswort oder andere Parameter nicht angeben, die zur Umgebungserstellung benötigt werden, werden Sie von der EB CLI zur Eingabe aufgefordert.

Eine vollständige Liste der verfügbaren Optionen sowie Anwendungsbeispiele finden Sie unter [eb create \(p. 1053\)](#).

Verwenden der AWS CLI

Wenn Sie den Befehl `create-environment` verwenden, um eine Elastic Beanstalk-Umgebung mit der AWS-CLI zu erstellen, werden von der AWS-CLI keine [empfohlenen Werte \(p. 648\)](#) übernommen. Alle Konfigurationsoptionen werden in den Konfigurationsdateien im von Ihnen vorgegebenen Quell-Bundle definiert.

Methoden

- [Verwenden von Konfigurationsdateien \(.ebextensions\) \(p. 657\)](#)
- [Verwenden einer gespeicherten Konfiguration \(p. 658\)](#)
- [Verwenden von Befehlszeilenoptionen \(p. 658\)](#)

Verwenden von Konfigurationsdateien (.ebextensions)

Um Konfigurationsdateien für eine mit der AWS-CLI erstellten Umgebung zu nutzen, binden Sie die Dateien in das Quell-Bundle der Anwendung ein, das Sie in Amazon S3 hochladen.

Weitere Informationen zu Konfigurationsdateien finden Sie unter [.Ebextensions \(p. 722\)](#).

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

So laden Sie das Quell-Bundle der Anwendung hoch und erstellen eine Umgebung mit der AWS CLI

1. Wenn Sie noch keinen Elastic Beanstalk-Bucket in Amazon S3 haben, erstellen Sie einen mit `create-storage-location`.

```
$ aws elasticbeanstalk create-storage-location
{
    "S3Bucket": "elasticbeanstalk-us-west-2-123456789012"
}
```

2. Laden Sie das Quell-Bundle der Anwendung in Amazon S3 hoch.

```
$ aws s3 cp sourcebundle.zip s3://elasticbeanstalk-us-west-2-123456789012/my-app/
sourcebundle.zip
```

3. Erstellen Sie die Anwendungsversion.

```
$ 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. Erstellen Sie die Umgebung.

```
$ 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"
```

Verwenden einer gespeicherten Konfiguration

Um eine gespeicherte Konfiguration während der Erstellung für eine Umgebung zu verwenden, nutzen Sie den Parameter `--template-name`.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name my-env --template-name savedconfig --version-label v1
```

Wenn Sie eine gespeicherte Konfiguration angeben, müssen Sie keinen Lösungs-Stack-Namen spezifizieren. In den gespeicherten Konfigurationen ist bereits ein Lösungs-Stack definiert, sodass Elastic Beanstalk eine Fehlermeldung ausgibt, wenn Sie beide Optionen angeben.

Verwenden von Befehlszeilenoptionen

Mit dem Parameter `--option-settings` können Sie Konfigurationsoptionen im JSON-Format angeben.

```
$ 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"  
  }  
]
```

Laden Sie JSON mit dem Präfix `file://` aus einer Datei.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name my-env --version-label v1 --template-name savedconfig --option-settings file:///healthcheckurl.json
```

Elastic Beanstalk übernimmt Optionseinstellungen, die Sie mit der Option `--option-settings` spezifiziert haben, direkt in die Umgebung. Falls dieselben Optionen in einer gespeicherten Konfiguration oder Konfigurationsdatei angegeben sind, werden die Werte von `--option-settings` überschrieben.

Festlegen von Konfigurationsoptionen nach der Umgebungserstellung

Sie können die Optionseinstellungen in einer laufenden Umgebung ändern, indem Sie gespeicherte Konfigurationen anwenden, ein neues Quell-Bundle mit Konfigurationsdateien (`.ebextensions`) hochladen oder ein JSON-Dokument nutzen. Die EB CLI und die Elastic Beanstalk-Konsole enthalten auch Client-spezifische Funktionen für das Einrichten und Aktualisieren von Konfigurationsoptionen.

Wenn Sie eine Konfigurationsoption festlegen oder ändern, können Sie ein vollständiges Umgebungs-Update auslösen, je nach Schweregrad der Änderung. Beispiel: Änderungen an Optionen in `aws:autoscaling:launchconfiguration` (p. 668), z. B. `InstanceType`, verlangen, dass die Amazon EC2-Instances in Ihrer Umgebung neu bereitgestellt werden müssen. Dadurch wird ein [fortlaufendes Update](#) (p. 484) ausgelöst. Andere Konfigurationsänderungen können ohne Unterbrechung oder erneute Bereitstellung angewendet werden.

Sie können Optionseinstellungen aus einer Umgebung mit EB CLI- oder AWS CLI-Befehlen entfernen. Wenn Sie eine Option entfernen, die direkt auf einer Umgebung auf einer API-Ebene festgelegt wurde, können Einstellungen in Konfigurationsdateien, die anderenfalls durch Einstellungen verborgen sind, die direkt auf eine Umgebung angewendet werden, eingeblendet und wirksam werden.

Einstellungen in gespeicherten Konfigurationen und Konfigurationsdateien können überschrieben werden, indem Sie diese Option mit einer der anderen Konfigurationsmethoden direkt in der Umgebung festlegen. Diese können jedoch nur vollständig entfernt werden, indem Sie eine aktualisierte gespeicherte Konfiguration oder eine Konfigurationsdatei anwenden. Wenn eine Option nicht in einer gespeicherten Konfiguration, Konfigurationsdatei oder direkt auf eine Umgebung festgelegt ist, wird der Standardwert angewendet, sofern vorhanden. Details dazu finden Sie unter [Precedence \(p. 647\)](#).

Clients

- [Die Elastic Beanstalk-Konsole. \(p. 659\)](#)
- [Die EB CLI \(p. 661\)](#)
- [Die AWS CLI \(p. 664\)](#)

Die Elastic Beanstalk-Konsole.

Sie können die Einstellungen der Konfigurationsoptionen in der Elastic Beanstalk-Konsole durch die Bereitstellung eines Anwendungsquell-Bundles mit Konfigurationsdateien aktivieren, eine gespeicherte Konfiguration zuweisen oder die Umgebung direkt mit der Seite Configuration (Konfiguration) in der Environment Management Console ändern.

Methoden

- [Verwenden von Konfigurationsdateien \(.ebextensions\) \(p. 659\)](#)
- [Verwenden einer gespeicherten Konfiguration \(p. 660\)](#)
- [Verwenden der Elastic Beanstalk-Konsole \(p. 660\)](#)

Verwenden von Konfigurationsdateien (.ebextensions)

Aktualisieren Sie Konfigurationsdateien in Ihrem Quellverzeichnis, erstellen Sie ein neues Quell-Bundle und stellen Sie die neue Version in Ihrer Elastic Beanstalk-Umgebung bereit, um die Änderungen zu übernehmen.

Weitere Informationen zu Konfigurationsdateien finden Sie unter [.Ebextensions \(p. 722\)](#).

So stellen Sie ein Quell-Bundle bereit

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie auf der Umgebungsübersichtsseite die Option Upload and deploy (Hochladen und bereitstellen).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um das Quellpaket hochzuladen.
5. Wählen Sie Deploy (Bereitstellen) aus.
6. Wenn die Bereitstellung abgeschlossen ist, wählen Sie die Website-URL, um Ihre Website in einer neuen Registerkarte zu öffnen.

Änderungen an Konfigurationsdateien überschreiben Optionseinstellungen in gespeicherten Konfigurationen oder Einstellungen, die direkt auf die Umgebung auf API-Ebene anwendet werden, nicht. Details dazu finden Sie unter [Precedence \(p. 647\)](#).

Verwenden einer gespeicherten Konfiguration

Wenden Sie eine gespeicherte Konfiguration auf eine laufende Umgebung an, um Optionseinstellungen anzuwenden, die durch sie definiert werden.

So wenden Sie eine gespeicherte Konfiguration auf eine laufende Umgebung an (Elastic Beanstalk-Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.

Note

Wenn Sie viele Anwendungen haben, verwenden Sie die Suchleiste, um die Anwendungsliste zu filtern.

3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Saved configurations (Gespeicherte Konfigurationen) aus.
4. Wählen Sie die gespeicherte Konfiguration aus, die Sie anwenden möchten, und wählen Sie dann Load (Laden).
5. Wählen Sie eine Umgebung und anschließend Load (Laden) aus.

Einstellungen in einer gespeicherten Konfiguration überschreiben Einstellungen in Konfigurationsdateien und werden von Einstellungen überschrieben, die mit der Environment Management Console konfiguriert werden.

Weitere Informationen zum Erstellen von gespeicherten Konfigurationen finden Sie unter [Gespeicherte Konfigurationen \(p. 651\)](#).

Verwenden der Elastic Beanstalk-Konsole

Die Elastic Beanstalk-Konsole stellt viele Konfigurationsoptionen für jede Umgebung auf der Seite Configuration (Konfiguration) bereit.

So ändern Sie die Konfigurationsoptionen in einer laufenden Umgebung (Elastic Beanstalk-Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Suchen Sie die Konfigurationsseite, die Sie bearbeiten möchten:
 - Wenn Sie die Option sehen, an der Sie interessiert sind, oder Sie wissen, in welcher Konfigurationskategorie sie sich befindet, wählen Sie in der Konfigurationskategorie für sie die Option Edit (Bearbeiten).
 - Um eine Option zu suchen, aktivieren Sie Table View (Tabellenansicht) und geben Sie dann Suchbegriffe in das Suchfeld ein. Während der Eingabe wird die Liste auf die Optionen verkürzt, die mit Ihren Suchbegriffen übereinstimmen.

Wenn Sie die Option sehen, nach der Sie suchen, wählen Sie in der Konfigurationskategorie, in der sie enthalten ist, die Option Edit (Bearbeiten).

The screenshot shows the 'Configuration overview' section of the AWS Elastic Beanstalk console. At the top right are 'Cancel' and 'Review changes' buttons. A search bar contains the text 'log'. Below is a table with two columns: 'Category' and 'Options'. The 'Software' category has three options: 'Lifecycle: Keep logs after terminating environment', 'Rotate logs: disabled', and 'Log streaming: disabled'. The 'Monitoring' category has one option: 'Log group: logStream:group=/aws/elasticbeanstalk/Node-env/environment-health target=_blank /aws/elasticbeanstalk/Node-env/environment-health.log'.

Category	Options
Software	Lifecycle: Keep logs after terminating environment Rotate logs: disabled Log streaming: disabled
Monitoring	Log group: logStream:group=/aws/elasticbeanstalk/Node-env/environment-health target=_blank /aws/elasticbeanstalk/Node-env/environment-health.log

5. Ändern Sie die Einstellungen und wählen Sie anschließend Save (Speichern) aus.
6. Wiederholen Sie bei Bedarf die vorherigen beiden Schritte in weiteren Konfigurationskategorien.
7. Wählen Sie Apply (Anwenden) aus.

Änderungen an den Konfigurationsoptionen in der Environment Management Console werden direkt auf die Umgebung angewendet. Diese Änderungen überschreiben die Einstellungen für dieselben Optionen in Konfigurationsdateien oder gespeicherten Konfigurationen. Details dazu finden Sie unter [Precedence \(p. 647\)](#).

Weitere Informationen zum Ändern von Konfigurationsoptionen für eine ausgeführte Umgebung mit der Elastic Beanstalk-Konsole finden Sie in den entsprechenden Themen unter [Konfigurieren von Elastic Beanstalk-Umgebungen \(p. 526\)](#).

Die EB CLI

Sie können die Einstellungen der Konfigurationsoptionen mit der EB CLI durch die Bereitstellung eines Quellcodes mit Konfigurationsdateien aktivieren, Einstellungen aus einer gespeicherten Konfiguration anwenden oder die Umgebungskonfiguration direkt mit dem eb config-Befehl ändern.

Methoden

- [Verwenden von Konfigurationsdateien \(.ebextensions\) \(p. 661\)](#)
- [Verwenden einer gespeicherten Konfiguration \(p. 662\)](#)
- [Die Verwendung von eb config \(p. 662\)](#)
- [Die Verwendung von eb setenv \(p. 663\)](#)

Verwenden von Konfigurationsdateien (.ebextensions)

Binden Sie .config-Dateien unter .ebextensions in den Projektordner ein, um sie mit dem Anwendungscode bereitzustellen.

Weitere Informationen zu Konfigurationsdateien finden Sie unter [.Ebextensions \(p. 722\)](#).

```
~/workspace/my-app/
|-- .ebextensions
|   |-- environmentvariables.config
|   '-- healthcheckurl.config
|-- .elasticbeanstalk
|   '-- config.yml
|-- index.php
`-- styles.css
```

Stellen Sie Ihren Quellcode mit eb deploy bereit.

```
~/workspace/my-app$ eb deploy
```

Verwenden einer gespeicherten Konfiguration

Um eine gespeicherte Konfiguration auf eine laufende Umgebung anzuwenden, nutzen Sie den Befehl eb config. Verwenden Sie die --cfg-Option mit dem Namen der gespeicherten Konfiguration, um die Einstellungen auf Ihre Umgebung anzuwenden.

```
$ eb config --cfg v1
```

In diesem Beispiel ist v1 der Name einer [zuvor erstellten und gespeicherten Konfigurationsdatei \(p. 651\)](#).

Einstellungen, die mit diesem Befehl auf eine Umgebung angewendet wurden, überschreiben Einstellungen, die während der Umgebungserstellung angewendet wurden, sowie Einstellungen, die in den Konfigurationsdateien in Ihrem Quell-Bundle der Anwendung definiert sind.

Die Verwendung von eb config

Mit dem EB CLI-eb config-Befehl können Sie Optionseinstellungen direkt auf einer Umgebung festlegen und entfernen, indem Sie einen Text-Editor verwenden.

Wenn Sie eb config ausführen, zeigt die EB CLI Einstellungen in Ihrer Umgebung aus allen Quellen, einschließlich Konfigurationsdateien, gespeicherte Konfigurationen, empfohlene Werte, Optionen, die direkt auf der Umgebung festgelegt sind, und API-Standardeinstellungen.

Note

eb config zeigt keine Umgebungseigenschaften. Zum Festlegen von Umgebungseigenschaften, die Sie in Ihrer Anwendung lesen können, verwenden Sie [eb setenv \(p. 663\)](#).

Das folgende Beispiel zeigt Einstellungen, die im aws:autoscaling:launchconfiguration-Namespace angewendet wurden. Diese Einstellungen umfassen Folgendes:

- Zwei empfohlene Werte für `IamInstanceProfile` und `InstanceType`, die von der EB CLI während der Umgebungserstellung angewendet wurden.
- Die Option `EC2KeyName`, die während der Erstellung basierend auf der Repository-Konfiguration direkt auf der Umgebung festgelegt wurde.
- API-Standardwerte für andere Optionen.

```
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  
...
```

Sie legen Sie Konfigurationsoptionen mit eb config fest oder ändern Sie

1. Führen Sie eb config aus, um die Umgebungskonfiguration anzuzeigen.

```
~/workspace/my-app/$ eb config
```

2. Ändern der Einstellungswerte mit dem Standard-Text-Editor.

```
aws:autoscaling:launchconfiguration:  
  BlockDeviceMappings: null  
  EC2KeyName: my-key  
  IamInstanceProfile: aws-elasticbeanstalk-ec2-role  
  ImageId: ami-1f316660  
  InstanceType: t2.medium
```

3. Speichern Sie die vorübergehende Konfigurationsdatei und beenden Sie sie.
4. Die EB CLI aktualisiert Ihre Umgebungskonfiguration.

Das Einrichten der Konfigurationsoptionen mit eb config überschreibt Einstellungen aus allen anderen Quellen.

Sie können auch Optionen aus Ihrer Umgebung mit eb config entfernen.

So entfernen Sie Konfigurationsoptionen (EB CLI)

1. Führen Sie eb config aus, um die Umgebungskonfiguration anzuzeigen.

```
~/workspace/my-app/$ eb config
```

2. Ersetzen Sie gezeigte Werte mit der Zeichenfolge null. Sie können auch die gesamte Zeile mit der Option, die Sie entfernen möchten, löschen.

```
aws:autoscaling:launchconfiguration:  
  BlockDeviceMappings: null  
  EC2KeyName: my-key  
  IamInstanceProfile: aws-elasticbeanstalk-ec2-role  
  ImageId: ami-1f316660  
  InstanceType: null
```

3. Speichern Sie die vorübergehende Konfigurationsdatei und beenden Sie sie.
4. Die EB CLI aktualisiert Ihre Umgebungskonfiguration.

Durch das Entfernen von Optionen aus Ihrer Umgebung mit eb config sind Einstellungen für dieselben Optionen möglich, die aus den Konfigurationsdateien im Quell-Bundle Ihrer Anwendung ermittelt werden. Details dazu finden Sie unter [Precedence \(p. 647\)](#).

Die Verwendung von eb setenv

Zum Festlegen von Umgebungseigenschaften mit der EB CLI verwenden Sie 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.
```

Mit diesem Befehl werden Umgebungseigenschaften im [aws:elasticbeanstalk:application:environment-Namespace \(p. 684\)](#) festgelegt. Nach einer kurzen Aktualisierung stehen die mit eb setenv definierten Umgebungseigenschaften für die Anwendung zur Verfügung.

Zeigen Sie die Umgebungseigenschaften für Ihre Umgebung mit eb printenv an.

```
~/workspace/my-app/$ eb printenv
Environment Variables:
ENVVAR = TEST
```

Die AWS CLI

Sie können die Einstellungen der Konfigurationsoptionen mit der AWS CLI durch die Bereitstellung eines Quell-Bundles mit Konfigurationsdateien aktivieren, eine remote gespeicherte Konfiguration anwenden oder die Umgebungskonfiguration direkt mit dem `aws elasticbeanstalk update-environment`-Befehl ändern.

Methoden

- [Verwenden von Konfigurationsdateien \(.ebextensions\) \(p. 664\)](#)
- [Verwenden einer gespeicherten Konfiguration \(p. 665\)](#)
- [Verwenden von Befehlszeilenoptionen \(p. 665\)](#)

Verwenden von Konfigurationsdateien (.ebextensions)

Um Konfigurationsdateien auf eine ausgeführte Umgebung mit der AWS-CLI bereitzustellen, binden Sie die Dateien in das Quell-Bundle der Anwendung ein, das Sie in Amazon S3 hochladen.

Weitere Informationen zu Konfigurationsdateien finden Sie unter [.Ebextensions \(p. 722\)](#).

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

So laden Sie das Quell-Bundle der Anwendung hoch und wenden es auf eine ausgeführte Umgebung an (AWS CLI)

1. Wenn Sie noch keinen Elastic Beanstalk-Bucket in Amazon S3 haben, erstellen Sie einen mit `create-storage-location`:

```
$ aws elasticbeanstalk create-storage-location
{
    "S3Bucket": "elasticbeanstalk-us-west-2-123456789012"
}
```

2. Laden Sie das Quell-Bundle der Anwendung in Amazon S3 hoch.

```
$ aws s3 cp sourcebundlev2.zip s3://elasticbeanstalk-us-west-2-123456789012/my-app/sourcebundlev2.zip
```

3. Erstellen Sie die Anwendungsversion.

```
$ 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. Aktualisieren Sie die Umgebung.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --version-label v2
```

Verwenden einer gespeicherten Konfiguration

Sie können eine gespeicherte Konfiguration auf eine laufende Umgebung mit der `--template-name`-Option auf dem `aws elasticbeanstalk update-environment`-Befehl anwenden.

Die gespeicherte Konfiguration muss sich in Ihrem Elastic Beanstalk-Bucket in einem Pfad mit dem Namen Ihrer Anwendung unter `resources/templates` befinden. Beispiel: Die Vorlage `v1` für die Anwendung `my-app` in der Region USA West (Oregon) (`us-west-2`) für Konto `123456789012` befindet sich unter `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app/v1`.

So wenden Sie eine gespeicherte Konfiguration auf eine laufende Umgebung an (AWS CLI)

- Geben Sie die gespeicherte Konfiguration in einem `update-environment`-Aufruf mit der Option `--template-name` an.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --template-name v1
```

Elastic Beanstalk platziert Konfigurationen in diesem Speicherort, wenn sie mit `aws elasticbeanstalk create-configuration-template` erstellt werden. Sie können auch gespeicherte Konfigurationen lokal ändern und sie selbst an diesem Speicherort ablegen.

Verwenden von Befehlszeilenoptionen

So ändern Sie Konfigurationsoptionen mit einem JSON-Dokument (AWS CLI)

1. Definieren Sie die Optionseinstellungen im JSON-Format in einer lokalen Datei.
2. Führen Sie `update-environment` mit der `--option-settings`-Option aus.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --option-settings file://-/ebconfigs/as-zero.json
```

In diesem Beispiel definiert `as-zero.json` Optionen, die die Umgebung mit einem Minimum und Maximum an Null-Instances konfigurieren. Dies stoppt die Instances in der Umgebung, ohne die Umgebung zu beenden.

`~/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

Das Einrichten der Konfigurationsoptionen mit `update-environment` überschreibt Einstellungen aus allen anderen Quellen.

Sie können auch Optionen aus Ihrer Umgebung mit `update-environment` entfernen.

So entfernen Sie Konfigurationsoptionen (AWS CLI)

- Führen Sie den Befehl `update-environment` mit der Option `--settings-to-remove` aus.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --options-to-remove
Namespace=aws:autoscaling:launchconfiguration,OptionName=InstanceType
```

Durch das Entfernen von Optionen aus Ihrer Umgebung mit `update-environment` sind Einstellungen für dieselben Optionen möglich, die aus den Konfigurationsdateien im Quell-Bundle Ihrer Anwendung ermittelt werden. Wenn eine Option nicht mit einer der folgenden Methoden konfiguriert ist, wird der API-Standardwert angewendet, falls vorhanden. Details dazu finden Sie unter [Precedence \(p. 647\)](#).

Allgemeine Optionen für alle Umgebungen

Namespaces

- [aws:autoscaling:asg \(p. 667\)](#)
- [aws:autoscaling:launchconfiguration \(p. 668\)](#)
- [aws:autoscaling:scheduledaction \(p. 674\)](#)
- [aws:autoscaling:trigger \(p. 675\)](#)
- [aws:autoscaling:updatepolicy:rollingupdate \(p. 677\)](#)
- [aws:ec2:instances \(p. 680\)](#)
- [aws:ec2:vpc \(p. 682\)](#)
- [aws:elasticbeanstalk:application \(p. 683\)](#)
- [aws:elasticbeanstalk:application:environment \(p. 684\)](#)
- [aws:elasticbeanstalk:cloudwatch:logs \(p. 684\)](#)
- [aws:elasticbeanstalk:cloudwatch:logs:health \(p. 684\)](#)
- [aws:elasticbeanstalk:command \(p. 685\)](#)
- [aws:elasticbeanstalk:environment \(p. 686\)](#)
- [aws:elasticbeanstalk:environment:process:default \(p. 687\)](#)
- [aws:elasticbeanstalk:environment:process:process_name \(p. 689\)](#)
- [aws:elasticbeanstalk:environment:proxy:staticfiles \(p. 691\)](#)
- [aws:elasticbeanstalk:healthreporting:system \(p. 692\)](#)

- [aws:elasticbeanstalk:hostmanager \(p. 693\)](#)
- [aws:elasticbeanstalk:managedactions \(p. 693\)](#)
- [aws:elasticbeanstalk:managedactions:platformupdate \(p. 694\)](#)
- [aws:elasticbeanstalk:monitoring \(p. 695\)](#)
- [aws:elasticbeanstalk:sns:topics \(p. 695\)](#)
- [aws:elasticbeanstalk:sqsd \(p. 696\)](#)
- [aws:elasticbeanstalk:trafficsplitting \(p. 698\)](#)
- [aws:elasticbeanstalk:xray \(p. 699\)](#)
- [aws:elb:healthcheck \(p. 699\)](#)
- [aws:elb:loadbalancer \(p. 699\)](#)
- [aws:elb:listener \(p. 701\)](#)
- [aws:elb:listener:listener_port \(p. 702\)](#)
- [aws:elb:policies \(p. 703\)](#)
- [aws:elb:policies:policy_name \(p. 704\)](#)
- [aws:elbv2:listener:default \(p. 705\)](#)
- [aws:elbv2:listener:listener_port \(p. 706\)](#)
- [aws:elbv2:listenerrule:rule_name \(p. 708\)](#)
- [aws:elbv2:loadbalancer \(p. 709\)](#)
- [aws:rds:dbinstance \(p. 711\)](#)

aws:autoscaling:asg

Konfigurieren Sie die Auto Scaling-Gruppe der Umgebung.

Namespace: **aws:autoscaling:asg**

Name	Beschreibung	Standard	Zulässige Werte
Availability Zones	Availability Zones (AZs) sind eindeutige Standorte in einer Region. Sie sollen vor Fehlern in anderen Availability Zonen schützen und eine kostengünstige Netzwerkverbindung mit geringer Latenz zu anderen Availability Zones in der gleichen Region bereitstellen. Wählen Sie die Anzahl der AZs für die Instances aus.	Any	Any Any 1 Any 2 Any 3
Cooldown	Ruhephasen verhindern, dass Amazon EC2 Auto Scaling zusätzliche Skalierungen initiiert, bevor die Auswirkungen von vorherigen Skalierungen erkennbar werden. Eine Ruhephase ist die Zeit in Sekunden, die nach einer Skalierung verstreichen muss, bevor eine weitere Skalierung eingeleitet werden kann.	360	0 auf 10000
Custom Availability Zones	Definieren Sie Availability Zones für die Instances.	Keine	us-east-1a us-east-1b us-east-1c

Name	Beschreibung	Standard	Zulässige Werte
			<code>us-east-1d</code> <code>us-east-1e</code> <code>eu-central-1</code>
MinSize	Mindestanzahl von Instances, die Sie in Ihrer Auto Scaling-Gruppe haben möchten.	1	1 auf 10000
MaxSize	Maximale Anzahl von Instances, die Sie in Ihrer Auto Scaling-Gruppe haben möchten.	4	1 auf 10000

aws:autoscaling:launchconfiguration

Konfigurieren Sie die Amazon Elastic Compute Cloud (Amazon EC2)-Instances Ihrer Umgebung.

Die Instances Ihrer Umgebung werden entweder mit einer Amazon EC2-Startvorlage oder einer Startkonfigurationsressource für Auto Scaling-Gruppen erstellt. Diese Optionen funktionieren mit beiden Ressourcentypen.

Weitere Informationen finden Sie unter [the section called “Amazon EC2-Instances” \(p. 531\)](#).

Namespace: **aws:autoscaling:launchconfiguration**

Name	Beschreibung	Standard	Zulässige Werte
DisableIMDSv1	Setzen Sie dies auf <code>true</code> , um Instance Metadata Service Version 1 (IMDSv1) zu deaktivieren. Standardmäßig aktivieren die Instances Ihrer Umgebung sowohl IMDSv1 als auch IMDSv2. Weitere Informationen finden Sie unter Konfiguration des Instance-Metadatenservice .	false	<code>true</code> <code>false</code>
EC2KeyName	Mit einem Schlüsselpaar können Sie sich sicher an der EC2-Instance anmelden. Note Wenn Sie die Elastic Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert (p. 648) .	Keine	
IamInstanceProfile	Ein Instance-Profil ermöglicht es AWS Identity and Access Management (IAM)-Benutzern und AWS-Services, auf temporäre Sicherheitsnachweise zuzugreifen, um AWS-API-Aufrufe	Keine	Instance-Profilname oder ARN

Name	Beschreibung	Standard	Zulässige Werte
	<p>durchzuführen. Geben Sie den Namen des Instance-Profils oder dessen ARN an.</p> <p>Beispiele:</p> <ul style="list-style-type: none"> • aws-elasticbeanstalk-ec2-role • arn:aws:iam::123456789012:instance-profile/aws-elasticbeanstalk-ec2-role <p>Note</p> <p>Wenn Sie die Elastic Beanstalk-Konsole oder EB CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole und EB CLI überschreiben diese Option mit einem empfohlenen Wert (p. 648).</p>		
ImageId	<p>Sie können das Standard-AMI (Amazon Machine Image) mit Ihrer eigenen benutzerdefinierten AMI-ID überschreiben.</p> <p>Beispiel: ami-1f316660</p>	Keine	

Name	Beschreibung	Standard	Zulässige Werte
InstanceType	<p>Der Instance-Typ, mit dem Ihre Anwendung in einer Elastic Beanstalk-Umgebung ausgeführt wird.</p> <p>Important</p> <p>Die Option <code>InstanceType</code> ist veraltet. Sie wird durch die neuere und leistungsfähigere Option <code>InstanceTypes</code> im aws:ec2:instances (p. 680)-Namespace ersetzt. Mit der neuen Option können Sie eine Liste einer oder mehrerer Instance-Typen für Ihre Umgebung angeben. Der erste Wert auf dieser List entspricht dem Wert der Option <code>InstanceType</code>, die in dem hier beschriebenen <code>aws:autoscaling:launchconfiguration</code>-Namespace enthalten ist. Als Methode zur Angabe von Instance-Typen wird die neue Option empfohlen. Wenn die neue Option angegeben wird, hat sie vor der alten Vorrang. Weitere Informationen finden Sie unter the section called “Der Namespace aws:ec2:instances” (p. 546).</p> <p>Die verfügbaren Instance-Typen hängen von der Plattform, dem Lösungs-Stack (Konfiguration) und der Region ab. Eine Liste der verfügbaren Instance-Typen für das Lösungs-Stack Ihrer Wahl erhalten Sie mit der Aktion <code>DescribeConfigurationOptions</code> in der API bzw. dem Befehl <code>describe-configuration-options</code> in der AWS CLI.</p> <p>Beispielsweise werden mit dem folgenden Befehl die verfügbaren Instance-Typen für die Version 1.4.3 des PHP 5.6-Stacks in der aktuellen Region aufgeführt:</p> <pre>\$ aws elasticbeanstalk describe-configuration-options --options Namespace=aws:autoscaling:launchconfiguration,OptionName=InstanceType --solution-stack-name "64bit Amazon Linux 2015.03 v1.4.3 running PHP 5.6"</pre>	Unabhängig von der Region	

Name	Beschreibung	Standard	Zulässige Werte
	<p>Note</p> <p>Wenn Sie die Elastic Beanstalk-Konsole oder EB CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole und EB CLI überschreiben diese Option mit einem empfohlenen Wert (p. 648).</p>		
MonitoringInterval	Intervall, in dem Amazon CloudWatch-Metriken zurückgegeben werden sollen.	5 minute	1 minute 5 minute
SecurityGroups	<p>Führt die Amazon EC2-Sicherheitsgruppen auf, die den EC2-Instances in der Auto Scaling-Gruppe zugeordnet werden sollen, um Firewall-Regeln für die Instances zu definieren.</p> <p>Sie können eine einzelne Zeichenfolge mit durch Komma getrennten Werten bereitstellen, in der die Namen von vorhandenen Amazon EC2-Sicherheitsgruppen oder Verweise auf AWS::EC2::SecurityGroup-Ressourcen, die in der Vorlage erstellt wurden, enthalten sind. Bei Namen von Sicherheitsgruppen wird zwischen Groß- und Kleinschreibung unterschieden.</p> <p>Wenn Sie Amazon Virtual Private Cloud (Amazon VPC) mit Elastic Beanstalk verwenden, damit Ihre Instances in einer Virtual Private Cloud (VPC) gestartet werden, geben Sie anstelle von Sicherheitsgruppennamen Sicherheitsgruppen-IDs an.</p>	elasticbeanstalk-default	

Name	Beschreibung	Standard	Zulässige Werte
SSHSourceRestriction	<p>Wird zum Sperren des SSH-Zugriffs in einer Umgebung verwendet. Beispielsweise können Sie den SSH-Zugriff auf die EC2-Instances sperren, sodass nur ein Bastion-Host auf die Instances im privaten Subnetz zugreifen kann.</p> <p>Die Zeichenfolge hat folgendes Format:</p> <p><i>protocol, fromPort, toPort, source_restriction</i></p> <p>Protokoll</p> <p>Das Protokoll für die Regel zum eingehenden Datenverkehr.</p> <p>fromPort</p> <p>Die Nummer des Start-Ports.</p> <p>toPort</p> <p>Die Nummer des End-Ports.</p> <p>source_restriction</p> <p>Der CIDR-Bereich oder der Name einer Sicherheitsgruppe, von der Datenverkehr zulässig sein soll. Wenn Sie eine Sicherheitsgruppe von einem anderen Konto angeben möchten (nur EC2 Classic, muss in derselben Region sein), binden Sie die Konto-ID vor dem Sicherheitsgruppennamen ein (z. B. <i>other_account_id/ security_group_name</i>). Wenn Sie Amazon Virtual Private Cloud (Amazon VPC) mit Elastic Beanstalk verwenden, damit Ihre Instances in einer Virtual Private Cloud (VPC) gestartet werden, geben Sie eine Sicherheitsgruppen-ID anstelle eines Sicherheitsgruppennamens an.</p> <p>Beispiel: <code>tcp, 22, 22, 54.240.196.185/32</code></p> <p>Beispiel: <code>tcp, 22, 22, my-security-group</code></p> <p>Beispiel (EC2 Classic): <code>tcp, 22, 22, 123456789012/their-security-group</code></p>	Keine	

Name	Beschreibung	Standard	Zulässige Werte
	Beispiel (VPC): <code>tcp, 22, 22, sg-903004f8</code>		
BlockDeviceMappings	<p>Fügen Sie zusätzliche Amazon EBS-Volumes oder Instance-Speicher-Volumes auf allen Instances in der Auto Scaling-Gruppe hinzu.</p> <p>Bei der Zuordnung von Instance-Speicher-Volumes wird nur der Gerätename einem Volume-Namen zugewiesen. Bei der Zuordnung von Amazon EBS-Volumes hingegen, können Sie die folgenden Felder (durch einen Doppelpunkt getrennt) angeben:</p> <ul style="list-style-type: none"> • Snapshot-ID • Größe in GB • Bei Beenden löschen (<code>true</code> oder <code>false</code>) • Speichertyp (<code>gp2</code>, <code>standard</code>, <code>st1</code>, <code>sc1</code> oder <code>io1</code>) • IOPS (nur bei <code>io1</code>-Volumes). <p>In folgendem Beispiel werden drei Amazon EBS-Volumes angefügt, und zwar ein leerer gp2-Volume mit 100 GB und einem Snapshot, ein leerer io1-Volume mit 20 GB und 2 000 bereitgestellten IOPS und das Instance-Speicher-Volume <code>ephemeral0</code>. Sofern der Instance-Typ dies unterstützt, können mehrere Instance-Speicher-Volumes hinzugefügt werden.</p> <pre>/dev/sdj=:100:true:gp2,/dev/sdh=snap-51eef269,/dev/sdi=:20:true:io1:2000,/dev/sdb=ephemeral0</pre>	Keine	
RootVolumeType	Volume-Typ (Magnetic-, Allzweck-SSD- oder bereitgestellte IOPS-SSD-Volumes) des Amazon EBS-Stamm-Volumes, das an die EC2-Instances der Umgebung angefügt ist.	Je nach Plattform unterschiedlich.	<p>standard für Magnetspeicher</p> <p>gp2 für Allzweck-SSD</p> <p>io1 für bereitgestellte IOPS-SSD</p>

Name	Beschreibung	Standard	Zulässige Werte
RootVolumeSize	<p>Speicherkapazität des Amazon EBS-Stamm-Volumes in GB.</p> <p>Dies ist erforderlich, wenn Sie für RootVolumeType bereitgestellte IOPS-SSD angeben.</p> <p>Beispiel, "64".</p>	<p>Je nach Plattform für Magnetspeicher und Allzweck-SSD unterschiedlich. Kein Standardwert für bereitgestellte IOPS-SSD.</p> <p>8 bis 1024 GB für Magnetspeicher.</p>	10 bis 16384 GB für Allzweck-SSD und bereitgestellte IOPS-SSD.
RootVolumeIOPS	<p>Gewünschte IOPS (input/output operations per second, E/A-Operationen pro Sekunde) für ein bereitgestelltes IOPS-SSD-Stamm-Volume.</p> <p>Das maximale Verhältnis von IOPS zur Volume-Größe liegt bei 30 zu 1. Beispielsweise muss ein Volume mit 3 000 IOPS mindestens eine Größe von 100 GB haben.</p>	Keine	100 auf 20000

aws:autoscaling:scheduledaction

Konfigurieren Sie die [geplanten Aktionen \(p. 550\)](#) der Auto Scaling-Gruppe der Umgebung. Geben Sie für jede Aktion neben dem Optionsnamen, dem Namespace und dem Wert für jede Einstellung einen `resource_name` an. Beispiele finden Sie unter [Der Namespace aws:autoscaling:scheduledaction \(p. 552\)](#).

Namespace: **aws:autoscaling:scheduledaction**

Name	Beschreibung	Standard	Zulässige Werte
StartTime	<p>Wählen Sie für einmalige Aktionen das Datum und die Uhrzeit aus, zu der die Aktion ausgeführt werden soll. Legen Sie für wiederkehrende Aktionen fest, wann die Aktion aktiviert wird.</p>	Keine	<p>Ein eindeutiger ISO-8601-Zeitstempel für alle geplanten Skalierungen</p>
EndTime	<p>Ein in der Zukunft liegender Zeitpunkt (mit Datum und Uhrzeit, in der Zeitzone UTC/GMT), zu dem die Skalierungsaktion nicht mehr wiederholt werden soll. Wenn Sie für EndTime keinen Wert festlegen, wird die Aktion entsprechend dem Ausdruck <code>Recurrence</code> wiederholt.</p> <p>Beispiel: 2015-04-28T04:07:2Z</p> <p>Wenn eine geplante Aktion endet, wird Amazon EC2 Auto Scaling nicht automatisch auf die vorherigen Einstellungen zurückgesetzt. Konfigurieren Sie daher gegebenenfalls eine zweite geplante Aktion, um zu den ursprünglichen Einstellungen zurückzukehren.</p>	Keine	<p>Ein eindeutiger ISO-8601-Zeitstempel für alle geplanten Skalierungen</p>

Name	Beschreibung	Standard	Zulässige Werte
MaxSize	Der Maximalwert für Instances, der beim Ausführen der Aktion angewendet werden soll	Keine	0 auf 10000
MinSize	Der Minimalwert für Instances, der beim Ausführen der Aktion angewendet werden soll	Keine	0 auf 10000
DesiredCapacity	Legen Sie die gewünschte Anfangskapazität für die Auto Scaling-Gruppe ein. Nachdem die geplante Aktion angewendet wurde, wird die gewünschte Kapazität basierend auf den Einstellungen angepasst.	Keine	0 auf 10000
Recurrence	Gibt an, wie häufig eine geplante Aktion ausgeführt werden soll. Falls Sie keine Wiederholung angeben, wird die Skalierung nur einmal (gemäß StartTime) ausgeführt.	Keine	Ein Cron-Ausdruck
Suspend	Setzen Sie die Option auf true fest, um eine wiederkehrende geplante Aktion vorübergehend zu deaktivieren.	false	true false

aws:autoscaling:trigger

Konfigurieren Sie Skalierungsauslöser für die Auto Scaling-Gruppe der Umgebung.

Note

Drei Optionen in diesem Namespace bestimmen, wie lange eine Metrik des Auslösers das definierte Limit überschreiten kann, bevor der Auslöser aktiviert wird. Diese Optionen stehen in folgendem Zusammenhang:

`BreachDuration = Period * EvaluationPeriods`

Die Standardwerte für diese Optionen (5, 5 bzw. 1) erfüllen diese Gleichung. Wenn Sie inkonsistente Werte angeben, ändert Elastic Beanstalk möglicherweise einen der Werte so, dass die Gleichung weiterhin erfüllt ist.

Namespace: `aws:autoscaling:trigger`

Name	Beschreibung	Standard	Zulässige Werte
BreachDuration	Gibt den Zeitraum an, für den der Grenzwert einer Metrik überschritten werden darf (wie unter <code>UpperThreshold</code> und <code>LowerThreshold</code> definiert), bevor der Auslöser ausgelöst wird.	5	1 auf 600
LowerBreachScaleIncrements	Gibt an, wie viele Amazon EC2-Instances im Rahmen einer Skalierung entfernt werden sollen.	-1	

Name	Beschreibung	Standard	Zulässige Werte
LowerThreshold	Falls die Messung diesen Wert der Verletzungsdauer unterschreitet, wird ein Auslöser ausgelöst.	2000000	0 auf 20000000
MeasureName	Für den Auto Scaling-Auslöser verwendete Metrik.	NetworkOut	CPUUtilization NetworkIn NetworkOut DiskWriteOps DiskReadBytes DiskReadOps DiskWriteBytes Latency RequestCount HealthyHostCount UnhealthyHostCount TargetResponseTime
Period	Gibt an, wie häufig die Metriken für den Auslöser von Amazon CloudWatch gemessen werden. Der Wert ist die Anzahl der Minuten zwischen zwei aufeinanderfolgenden Zeiträumen.	5	1 auf 600
EvaluationPeriods	Die Anzahl der aufeinanderfolgenden Auswertungszeiträumen, die verwendet, um festzustellen, ob eine Verletzung auftritt.	1	1 auf 600
Statistic	Gibt die Statistik an, die der Auslöser verwenden soll, z. B. Average.	Average	Minimum Maximum Sum Average

Name	Beschreibung	Standard	Zulässige Werte
Unit	Maßeinheit für die Auslösermessung, z. B. Bytes.	Bytes	Seconds Percent Bytes Bits Count Bytes/Second Bits/Second Count/Second None
UpperBreachScaleIncrements	Gibt an, wie viele Amazon EC2-Instances im Rahmen einer Skalierung hinzugefügt werden sollen.	1	
UpperThreshold	Falls die Messung diesen Wert der Verletzungsdauer überschreitet, wird ein Auslöser ausgelöst.	6000000	0 auf 20000000

aws:autoscaling:updatepolicy:rollingupdate

Konfigurieren Sie fortlaufende Aktualisierungen für die Auto Scaling-Gruppe der Umgebung.

Namespace: **aws:autoscaling:updatepolicy:rollingupdate**

Name	Beschreibung	Standard	Zulässige Werte
MaxBatchSize	Anzahl der Instances, die in jedem fortlaufenden Aktualisierungsstapel enthalten sind.	Ein Drittel der Mindestgröße der Auto Scaling-Gruppe, gerundet auf die nächsthöhere Ganzzahl.	1 auf 10000
MinInstancesInService	Mindestanzahl der Instances, die in der Auto Scaling-Gruppe aktiv sein müssen, wenn andere Instances beendet werden.	Die Mindestgröße der Auto Scaling-Gruppe oder eine Instance weniger als die maximale Größe der Auto Scaling-Gruppe – je nachdem, welcher Wert der niedrigere ist.	0 auf 9999
RollingUpdateEnabled	Wenn für diesen Parameter true festgelegt ist,	false	true false

Name	Beschreibung	Standard	Zulässige Werte
	<p>sind fortlaufende Aktualisierungen für eine Umgebung aktiviert. Fortlaufende Aktualisierungen sind sinnvoll, wenn Sie häufig kleinere Aktualisierungen für die Elastic Beanstalk-Softwareanwendung ohne Ausfallzeit der Anwendung ausführen möchten.</p> <p>Wird dieser Wert auf "true" gesetzt, aktiviert das System automatisch die Optionen <code>MaxBatchSize</code>, <code>MinInstancesInService</code> und <code>PauseTime</code>. Wenn Sie für eine dieser Optionen einen Wert angeben, wird auch die Option <code>RollingUpdateEnabled</code> automatisch auf den Wert <code>true</code> gesetzt. Mit dem Wert <code>false</code> lassen sich fortlaufende Aktualisierungen deaktivieren.</p> <p>Note</p> <p>Wenn Sie die Elastic Beanstalk-Konsole oder EB CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole und EB CLI überschreiben diese Option mit einem</p>		

Name	Beschreibung	Standard	Zulässige Werte
	empfohlenen Wert (p. 648).		
RollingUpdateType	<p>Bei zeitbasierten fortlaufenden Aktualisierungen kommt zwischen den Stapeln ein "PauseTime"-Wert zum Tragen. Bei zustandsbasierten fortlaufenden Aktualisierungen wird gewartet, bis die neuen Instances die Zustandsprüfungen bestanden haben, bevor mit dem nächsten Stapel begonnen wird.</p> <p>Bei unveränderlichen Aktualisierungen (p. 488) wird eine komplette Instance-Gruppe in einer neuen Auto Scaling-Gruppe gestartet.</p> <p>Note</p> <p>Wenn Sie die Elastic Beanstalk-Konsole oder EB CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole und EB CLI überschreiben diese Option mit einem empfohlenen Wert (p. 648).</p>	Time	Time Health Immutable

Name	Beschreibung	Standard	Zulässige Werte
PauseTime	Gibt den Zeitraum an, den der Elastic Beanstalk-Service nach erfolgreicher Aktualisierung eines Instance-Stapels wartet, bevor mit dem nächsten Stapel begonnen wird.	Wird automatisch auf Basis von Instance-Typ und Container berechnet.	PT0S* (0 Sekunden) bis PT1H (1 Stunde)
Timeout	Gibt den maximalen Zeitraum an, der gewartet wird, bis alle Instances in einem Stapel die Zustandsprüfungen bestehen, bevor die Aktualisierung abgebrochen wird.	PT30M (30 Minuten)	PT5M* (5 Minuten) bis PT1H (1 Stunde) * ISO 8601-Format für Zeitangaben: PT#H#M#S, wobei jedes #-Zeichen Stunden, Minuten und/oder Sekunden angibt.

aws:ec2:instances

Konfigurieren Sie die Instances Ihrer Umgebung, einschließlich Spot-Optionen. Dieser Namespace ergänzt [aws:autoscaling:launchconfiguration \(p. 668\)](#) und [aws:autoscaling:asg \(p. 667\)](#).

Weitere Informationen finden Sie unter [the section called “Auto Scaling-Gruppe” \(p. 540\)](#).

Namespace: **aws:ec2:instances**

Name	Beschreibung	Standard	Zulässige Werte
EnableSpot	Aktivieren Sie Spot-Instance-Anforderungen für Ihre Umgebung. Wenn <code>false</code> , werden einige Optionen in diesem Namespace nicht wirksam.	<code>false</code>	<code>true</code> <code>false</code>
InstanceTypes	Eine durch Kommas getrennte Liste von Instance-Typen, die in Ihrer Umgebung verwendet werden sollen. Beispiel: <code>t2.micro, t3.micro</code> Wenn Spot-Instances deaktiviert sind (<code>EnableSpot</code> ist <code>false</code>), wird nur der erste Instance-Typ in der Liste verwendet. Der erste Instance-Typ in der Liste in dieser Option entspricht dem Wert der <code>InstanceType</code> -Option im aws:autoscaling:launchconfiguration (p. 668) Namespace. Letztere ist veraltet, und wir empfehlen nicht, sie zu verwenden. Wenn Sie beide angeben, wird der erste Instance-Typ in der Liste in der <code>InstanceTypes</code> -Option verwendet und <code>InstanceType</code> wird ignoriert.	Eine Liste mit zwei Instance-Typen. Ist je nach Konto, Region und Plattform verschieden.	1 bis 10 EC2-Instance-Typen (mindestens zwei empfohlen)

Name	Beschreibung	Standardwert	Zulässige Werte
	<p>Note</p> <p>Einige ältere AWS-Konten stellen möglicherweise Standard-Instance-Typen für Elastic Beanstalk bereit, die keine Spot-Instances unterstützen (z. B. t1.micro). Wenn Sie Spot-Instance-Anforderungen aktivieren und eine Fehlermeldung zu einem Instance-Typ erhalten, der keine Spot-Flotten unterstützt, müssen Sie Instance-Typen konfigurieren, die Spot-Flotten unterstützen. Verwenden Sie zur Auswahl von Spot-Instance-Typen Spot Instance Advisor.</p> <p>Wenn Sie die Umgebungskonfiguration aktualisieren und einen oder mehrere Instance-Typen aus der <code>InstanceTypes</code>-Option entfernen, beendet Elastic Beanstalk alle Amazon EC2-Instances, die auf einem der entfernten Instance-Typen ausgeführt werden. Die Auto Scaling-Gruppe Ihrer Umgebung startet dann nach Bedarf neue Instances, um die gewünschte Kapazität zu vervollständigen, unter Verwendung der aktuell angegebenen Instance-Typen.</p>		
<code>SpotFleetOnDemandBaseMinSize</code>	<p>Die Mindestanzahl von On-Demand-Instances, die von der Auto Scaling-Gruppe bereitgestellt werden, bevor Spot-Instances bei der Skalierung Ihrer Umgebung in Betracht gezogen werden.</p> <p>Diese Option ist nur relevant, wenn <code>EnableSpot</code> true ist.</p>	0	Option 0 bis <code>MaxSize</code> Option im aws:autoscaling:asg (p. 667)-Namespace
<code>SpotFleetOnDemandAboveBasePercentage</code>	<p>Der Prozentanteil der On-Demand-Instances als Teil der zusätzlichen Kapazität, die von der Auto Scaling-Gruppe über die <code>SpotOnDemandBaseInstances</code> hinaus bereitgestellt wird.</p> <p>Diese Option ist nur relevant, wenn <code>EnableSpot</code> true ist.</p>	0 für eine Umgebung mit einer einzelnen Instance 70 für eine Umgebung mit Lastausgleich	0 auf 100

Name	Beschreibung	Standard	Zulässige Werte
SpotMaxPrice	<p>Der Höchstpreis pro Einheitsstunde in USD, den Sie für eine Spot-Instance zu zahlen bereit sind</p> <p>Diese Option ist nur relevant, wenn <code>EnableSpot</code> true ist.</p>	On-Demand-Preis, pro Instance-Typ. Der Wert der Option ist in diesem Fall null.	0.001 auf 20.0 null

aws:ec2:vpc

Konfigurieren Sie Ihre Umgebung, um Ressourcen in einer benutzerdefinierten [Amazon Virtual Private Cloud](#) (Amazon VPC) zu starten. Falls Sie keine Einstellungen in diesem Namespace konfigurieren, startet Elastic Beanstalk die Ressourcen in der Standard-VPC.

Namespace: **aws:ec2:vpc**

Name	Beschreibung	Standard	Zulässige Werte
VPCId	Die ID der Amazon VPC.	Keine	
Subnets	Gibt die Subnetz-IDs für die Auto Scaling-Gruppe an. Bei mehreren Subnetzen geben Sie den Wert in einer durch Komma getrennten Zeichenfolge der Subnetz-IDs an (z. B. "subnet-11111111, subnet-22222222").	Keine	
ELBSubnets	Gibt die Subnetz-IDs für den Elastic Load Balancer an. Bei mehreren Subnetzen geben Sie den Wert in einer durch Komma getrennten Zeichenfolge der Subnetz-IDs an (z. B. "subnet-11111111, subnet-22222222").	Keine	
ELBScheme	Geben Sie <code>internal</code> an, ob ein interner Load Balancer in der Amazon VPC erstellt werden soll, damit nur innerhalb der Amazon VPC auf die Elastic Beanstalk-Anwendung zugegriffen werden kann. Wenn Sie einen anderen Wert als <code>public</code> oder <code>internal</code> angeben, wird dieser von Elastic Beanstalk ignoriert.	public	public internal
DBSubnets	Enthält die IDs der Datenbanksubnetze. Dieser Parameter wird nur verwendet, wenn eine Amazon RDS-DB-Instance als Teil der Anwendung hinzugefügt werden soll. Bei mehreren Subnetzen geben Sie den Wert in einer durch Komma getrennten Zeichenfolge der Subnetz-IDs an (z. B. "subnet-11111111, subnet-22222222").	Keine	
AssociatePublicIpAddress	Gibt an, ob die Instances mit öffentlichen IP-Adressen in der Amazon VPC gestartet werden sollen. Instances mit öffentlichen IP-Adressen erfordern kein NAT-Gerät, um mit dem Internet verbunden zu sein.	Keine	true false

Name	Beschreibung	Standard	Zulässige Werte
	<p>dem Internet zu kommunizieren. Sie müssen den Wert auf <code>true</code> setzen, wenn der Load Balancer und die Instances in ein einziges öffentliches Subnetz einbezogen werden sollen.</p> <p>Diese Option hat keine Auswirkungen auf eine Single-Instance-Umgebung, die stets eine einzelne Amazon EC2-Instance mit einer Elastic IP-Adresse aufweist. Die Option ist relevant für skalierbare Umgebungen mit Lastausgleich.</p>		

aws:elasticbeanstalk:application

Konfigurieren Sie einen Pfad für die Zustandsprüfung der Anwendung. Weitere Informationen finden Sie unter [Grundlegende Zustandsberichte \(p. 821\)](#).

Namespace: **aws:elasticbeanstalk:application**

Name	Beschreibung	Standard	Zulässige Werte
URL für die Zustandsprüfung der Anwendung	<p>Gibt den Pfad an, an den Zustandsprüfungsanforderungen gesendet werden. Falls kein Pfad definiert ist, versucht der Load Balancer über Port 80 eine TCP-Verbindung herzustellen, um die Zustandsprüfung auszuführen. Geben Sie als erstes Zeichen im Pfad <code>/</code> an, damit eine HTTP GET-Anforderung an diesen Pfad gesendet wird. Sie können vor dem Pfad auch ein Protokoll (HTTP, HTTPS, TCP oder SSL) und einen Port zur Überprüfung der HTTPS-Verbindung einbinden oder einen anderen als den Standard-Port verwenden.</p> <p>Note</p> <p>Wenn Sie die Elastic Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert (p. 648).</p>	Keine	<code>/</code> (HTTP GET zum Stammpfad) <code>/health</code> <code>HTTPS:443/</code> <code>HTTPS:443/health</code> usw.

Die EB CLI und Elastic Beanstalk-Konsole wenden die empfohlenen Werte für die vorhergehenden Optionen an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter [Empfohlene Werte \(p. 648\)](#).

aws:elasticbeanstalk:application:environment

Konfigurieren Sie Umgebungseigenschaften für die Anwendung.

Namespace: **aws:elasticbeanstalk:application:environment**

Name	Beschreibung	Standard	Zulässige Werte
Name einer beliebigen Umgebungsvariable	Dient der Übergabe von Schlüssel-Wert-Paaren.	Keine	Wert einer beliebigen Umgebungsvariable.

Weitere Informationen finden Sie unter [Umgebungseigenschaften und andere Softwareeinstellungen \(p. 621\)](#).

aws:elasticbeanstalk:cloudwatch:logs

Konfigurieren Sie das Instance-Protokoll-Streaming für Ihre Anwendung.

Namespace: **aws:elasticbeanstalk:cloudwatch:logs**

Name	Beschreibung	Standard	Zulässige Werte
StreamLogs	Gibt an, ob in CloudWatch Logs Gruppen für Proxy- und Bereitstellungsprotokolle erstellt und Protokolle von jeder Instance der Umgebung gestreamt werden sollen.	false	true false
DeleteOnTermination	Gibt an, ob die Protokollgruppen gelöscht werden sollen, wenn die Umgebung beendet wird. Lautet der Wert false, werden die Protokolle für die unter RetentionInDays angegebene Dauer gespeichert.	false	true false
RetentionInDays	Gibt die Anzahl der Tage an, die Protokollereignisse gespeichert werden, bevor sie ablaufen.	7	1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653

aws:elasticbeanstalk:cloudwatch:logs:health

Konfigurieren Sie das Umgebungszustand-Protokoll-Streaming für Ihre Anwendung.

Namespace: **aws:elasticbeanstalk:cloudwatch:logs:health**

Name	Beschreibung	Standard	Zulässige Werte
HealthStreaming	Einhaben Umgebungen mit erweiterten Zustandsberichten, ob eine Gruppe in CloudWatch Logs für den Umgebungszustand erstellt werden soll, und ob Elastic Beanstalk-Umgebungszustandsdaten	false	true false

Name	Beschreibung	Standard	Zulässige Werte
	archiviert werden sollen. Weitere Informationen zur Aktivierung des erweiterten Zustands finden Sie unter aws:elasticbeanstalk:healthreporting:system (p. 692) .		
DeleteOnTermination	Gibt an, ob die Protokollgruppe gelöscht werden soll, wenn die Umgebung beendet wird. Bei <code>false</code> werden die Zustandsdaten <code>RetentionInDays</code> Tage gespeichert.	<code>false</code>	<code>true</code> <code>false</code>
RetentionInDays	Die Anzahl der Tage, wie lange archivierte Zustandsdaten aufbewahrt werden, bevor sie ablaufen.	7	1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653

aws:elasticbeanstalk:command

Konfigurieren Sie die Bereitstellungsrichtlinie für den Anwendungscode. Weitere Informationen finden Sie unter [the section called “Optionen für die Bereitstellung” \(p. 474\)](#).

Namespace: **aws:elasticbeanstalk:command**

Name	Beschreibung	Standard	Zulässige Werte
DeploymentPolicy	<p>Wählen Sie eine Bereitstellungsrichtlinie (p. 474) für die Bereitstellung von Anwendungsversionen aus.</p> <p>Note</p> <p>Wenn Sie die Elastic Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert (p. 648).</p>	AllAtOnce	AllAtOnce Rolling RollingWithAddition Immutable TrafficSplitting
Timeout	<p>Gibt an, wie viele Sekunden gewartet wird, bis eine Instance Ausführungsbefehle abgeschlossen hat.</p> <p>Elastic Beanstalk fügt dem Timeout-Wert intern 240 Sekunden (vier Minuten) hinzu. Beispiel: Der effektive Timeout liegt standardmäßig bei 840 Sekunden (600 + 240) oder 14 Minuten.</p>	600	1 auf 3600

Name	Beschreibung	Standard	Zulässige Werte
BatchSizeType	<p>Gibt den im Parameter BatchSize spezifizierten Zahlentyp an.</p> <p>Note</p> <p>Wenn Sie die Elastic Beanstalk-Konsole oder EB CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole und EB CLI überschreiben diese Option mit einem empfohlenen Wert (p. 648).</p>	Percentage	Percentage Fixed
BatchSize	<p>Prozentsatz oder feste Anzahl der Amazon EC2-Instances in der Auto Scaling-Gruppe, für die gleichzeitig Bereitstellungen ausgeführt werden. Die zulässigen Werte variieren je nach BatchSizeType-Einstellung.</p> <p>Note</p> <p>Wenn Sie die Elastic Beanstalk-Konsole oder EB CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole und EB CLI überschreiben diese Option mit einem empfohlenen Wert (p. 648).</p>	100	1 auf 100 (Percentage). 1 bis aws:autoscaling:asg::Max (Fixed)
IgnoreHealthCheck	Damit lässt sich festlegen, dass eine Bereitstellung nicht aufgrund fehlgeschlagener Zustandsprüfungen abgebrochen wird.	false	true false

aws:elasticbeanstalk:environment

Konfigurieren Sie die Architektur und die Servicerolle für die Umgebung.

Namespace: **aws:elasticbeanstalk:environment**

Name	Beschreibung	Standard	Zulässige Werte
EnvironmentType	Legen Sie SingleInstance fest, damit eine EC2-Instance ohne Load Balancer gestartet wird.	LoadBalanced	SingleInstance LoadBalanced

Name	Beschreibung	Standard	Zulässige Werte
ServiceRole	<p>Der Name einer IAM-Rolle, die Elastic Beanstalk zur Verwaltung der Umgebungsressourcen verwendet. Geben Sie den Namen einer Rolle (optional mit einem benutzerdefinierten Pfad als Präfix) oder ihren ARN an.</p> <p>Beispiele:</p> <ul style="list-style-type: none"> aws-elasticbeanstalk-service-role <i>custom-path/custom-role</i> arn:aws:iam::123456789012:role/aws-elasticbeanstalk-service-role <p>Note</p> <p>Wenn Sie die Elastic Beanstalk-Konsole oder EB CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole und EB CLI überschreiben diese Option mit einem empfohlenen Wert (p. 648).</p>	Keine	IAM-Rollennamen, Pfad/Name oder ARN
LoadBalancerType	Der Typ des Load Balancers für Ihre Umgebung. Details dazu finden Sie unter the section called “Load Balancer” (p. 554) .	classic	classic application network
LoadBalancerIsShared	<p>Legt fest, ob der Load Balancer der Umgebung dediziert oder freigegeben ist. Diese Option kann nur für einen Application Load Balancer festgelegt werden. Sie kann nach der Erstellung der Umgebung nicht geändert werden.</p> <p>Wenn <code>false</code>, verfügt die Umgebung über einen eigenen dedizierten Load Balancer, der von Elastic Beanstalk erstellt und verwaltet wird. Wenn <code>true</code>, verwendet die Umgebung einen freigegebenen, von Ihnen erstellten Load Balancer, der in der Option <code>SharedLoadBalancer</code> des Namespace aws:elbv2:loadbalancer (p. 709) angegeben ist.</p>	false	true false

aws:elasticbeanstalk:environment:process:default

Konfigurieren Sie den Standardprozess für die Umgebung.

Namespace: **aws:elasticbeanstalk:environment:process:default**

Name	Beschreibung	Standard	Zulässige Werte
DeregistrationDelay	Gibt an, wie lange (in Sekunden) auf den Abschluss aktiver Anforderungen gewartet wird, bevor die Abmeldung erfolgt.	20	0 auf 3600
HealthCheckInterval	Gibt an, wie häufig (in Sekunden) Elastic Load Balancing Zustandsprüfungen für die Amazon EC2 Instances der Anwendung ausführt.	Bei Classic oder Application Load Balancer: 15 Bei Network Load Balancer: 30	Bei Classic oder Application Load Balancer: 5 bis 300 Bei Network Load Balancer: 10, 30
HealthCheckPath	Gibt den Pfad an, an den HTTP-Anforderungen für Zustandsprüfungen gesendet werden.	/	Ein routingfähiger Pfad.
HealthCheckTimeout	Gibt an, wie lange (in Sekunden) während einer Zustandsprüfung auf eine Antwort gewartet wird. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	5	1 auf 60
HealthyThresholdCount	Aufeinanderfolgende erfolgreiche Anforderungen vor Elastic Load Balancing ändert den Instance-Zustand.	Bei Classic oder Application Load Balancer: 3 Bei Network Load Balancer: 5	2 auf 10
MatcherHTTPCode	Eine durch Komma getrennte Liste von HTTP-Codes (s), die anzeigt, dass eine Instance fehlerfrei ist. Diese Option ist nur für Umgebungen mit einem Application Load Balancer bestimmt.	200	Bei Application Load Balancer: 200 bis 499 Bei Network Load Balancer: 200 bis 399
Port	Gibt den vom Prozess überwachten Port an.	80	1 auf 65535

Name	Beschreibung	Standard	Zulässige Werte
Protocol	<p>Gibt das vom Prozess verwendete Protokoll an.</p> <p>Bei einem Application Load Balancer können Sie diese Option nur auf <code>HTTP</code> oder <code>HTTPS</code> festlegen.</p> <p>Bei einem Network Load Balancer können Sie diese Option nur auf <code>TCP</code> festlegen.</p>	<p>Bei Classic oder Application Load Balancer: <code>HTTP</code></p> <p>Bei Network Load Balancer: <code>TCP</code></p>	<code>TCP</code> <code>HTTP</code> <code>HTTPS</code>
StickinessEnabled	<p>Legen Sie den Wert auf "true" fest, um Sticky Sessions zu aktivieren.</p> <p>Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.</p>	'false'	<code>'false'</code> <code>'true'</code>
StickinessLBCookieDuration	<p>Gibt die Lebensdauer des Sticky Session-Cookies in Sekunden an.</p> <p>Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.</p>	86400 (ein Tag)	1 auf 604800
StickinessType	<p>Legen Sie den Wert auf <code>lb_cookie</code> fest, um Cookies für Sticky Sessions zu nutzen.</p> <p>Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.</p>	<code>lb_cookie</code>	<code>lb_cookie</code>
UnhealthyThresholdCount	Aufeinanderfolgende erfolgreiche Anforderungen vor Elastic Load Balancing ändert den Instance-Zustand.	5	2 auf 10

aws:elasticbeanstalk:environment:process:process_name

Konfigurieren Sie zusätzliche Prozesse für die Umgebung.

Namespace: **aws:elasticbeanstalk:environment:process:process_name**

Name	Beschreibung	Standard	Zulässige Werte
DeregistrationDelay	Gibt an, wie lange (in Sekunden) auf den Abschluss aktiver Anforderungen gewartet wird, bevor die Abmeldung erfolgt.	20	0 auf 3600
HealthCheckInterval	Gibt an, wie häufig (in Sekunden) Elastic Load Balancing Zustandsprüfungen für die Amazon EC2 Instances der Anwendung ausführt.	Bei Classic oder Application Load Balancer: 15 Bei Network Load Balancer: 30	Bei Classic oder Application Load Balancer: 5 bis 300 Bei Network Load Balancer: 10, 30
HealthCheckPath	Gibt den Pfad an, an den HTTP-Anforderungen für Zustandsprüfungen gesendet werden.	/	Ein routingfähiger Pfad.
HealthCheckTimeout	Gibt an, wie lange (in Sekunden) während einer Zustandsprüfung auf eine Antwort gewartet wird. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	5	1 auf 60
HealthyThresholdCount	Aufeinanderfolgende erfolgreiche Anforderungen vor Elastic Load Balancing ändert den Instance-Zustand.	Bei Classic oder Application Load Balancer: 3 Bei Network Load Balancer: 5	2 auf 10
MatcherHTTPCode	Eine durch Komma getrennte Liste von HTTP-Codes (s), die anzeigt, dass eine Instance fehlerfrei ist. Diese Option ist nur für Umgebungen mit einem Application Load Balancer bestimmt.	200	Bei Application Load Balancer: 200 bis 499 Bei Network Load Balancer: 200 bis 399
Port	Gibt den vom Prozess überwachten Port an.	80	1 auf 65535

Name	Beschreibung	Standard	Zulässige Werte
Protocol	<p>Gibt das vom Prozess verwendete Protokoll an.</p> <p>Bei einem Application Load Balancer können Sie diese Option nur auf <code>HTTP</code> oder <code>HTTPS</code> festlegen.</p> <p>Bei einem Network Load Balancer können Sie diese Option nur auf <code>TCP</code> festlegen.</p>	<p>Bei Classic oder Application Load Balancer: <code>HTTP</code></p> <p>Bei Network Load Balancer: <code>TCP</code></p>	<code>TCP</code> <code>HTTP</code> <code>HTTPS</code>
StickinessEnabled	<p>Legen Sie den Wert auf "true" fest, um Sticky Sessions zu aktivieren.</p> <p>Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.</p>	'false'	<code>'false'</code> <code>'true'</code>
StickinessLBCookieDuration	<p>Gibt die Lebensdauer des Sticky Session-Cookies in Sekunden an.</p> <p>Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.</p>	86400 (ein Tag)	1 auf 604800
StickinessType	<p>Legen Sie den Wert auf <code>lb_cookie</code> fest, um Cookies für Sticky Sessions zu nutzen.</p> <p>Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.</p>	<code>lb_cookie</code>	<code>lb_cookie</code>
UnhealthyThresholdCount	Aufeinanderfolgende erfolgreiche Anforderungen vor Elastic Load Balancing ändert den Instance-Zustand.	5	2 auf 10

aws:elasticbeanstalk:environment:proxy:staticfiles

Sie können den Proxy-Server mit dem folgenden Namespace konfigurieren, um statische Dateien bereitzustellen. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält,

stellt er die Datei direkt bereit, statt die Anfrage an Ihre Anwendung weiterzuleiten. Dies reduziert die Anzahl der Anfragen, die Ihre Anwendung verarbeiten muss.

Bilden Sie einen von dem Proxy-Server bedienten Pfad auf einen Ordner in Ihrem Quellcode ab, der statische Komponenten enthält. Jede Option, die Sie in diesem Namespace definieren, weist einen anderen Pfad zu.

Note

Dieser Namespace gilt für Plattformvarianten basierend auf Amazon Linux 2. Wenn Ihre Umgebung eine Plattformversion verwendet, die auf Amazon Linux AMI (vor Amazon Linux 2) basiert, finden Sie unter [the section called “Plattformspezifische Optionen” \(p. 712\)](#) Informationen zu plattformspezifischen Namespaces für statische Dateien.

Namespace: **aws:elasticbeanstalk:environment:proxy:staticfiles**

Name	Wert
Der Pfad, in dem der Proxy-Server die Dateien bereitstellt Starten Sie den Wert mit /. Beispiel: Geben Sie /images an, um Dateien unter subdomain.eleasticbeanstalk.com/images bereitzustellen.	Der Name des Verzeichnisses, das die Dateien enthält Beispiel: Geben Sie staticimages an, um Dateien aus einem Verzeichnis mit dem Namen staticimages auf oberster Ebene Ihres Quell-Bundles bereitzustellen.

aws:elasticbeanstalk:healthreporting:system

Konfigurieren Sie erweiterte Zustandsberichte für die Umgebung.

Namespace: **aws:elasticbeanstalk:healthreporting:system**

Name	Beschreibung	Standard	Zulässige Werte
SystemType	System für Zustandsberichte (grundlegend (p. 821) oder erweitert (p. 824)). Erweiterte Zustandsberichte erfordern eine Servicerolle (p. 21) und eine Plattformversion (p. 32) mit Version 2 oder höher. Note Wenn Sie die Elastic Beanstalk-Konsole oder EB CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole und EB CLI überschreiben diese Option mit einem empfohlenen Wert (p. 648) .	basic	basic enhanced
ConfigDocument	Ein JSON-Dokument, das die Instance- und Umgebungsmetriken beschreibt, die in CloudWatch veröffentlicht werden sollen.	Keine	
EnhancedHealthAuthEnabled	Aktivieren Sie die Autorisierung für die interne API, die von Elastic Beanstalk für die Übermittlung	false	true false

Name	Beschreibung	Standard	Zulässige Werte
	<p>erweiterter Integritätsinformationen aus Ihren Umgebungs-Instances an den Elastic Beanstalk-Service verwendet wird.</p> <p>Details dazu finden Sie unter the section called "Rollen in erweiterten Zustandsberichten" (p. 830).</p> <p>Note</p> <p>Diese Option kann nur auf erweiterte Integritätsberichte angewendet werden (d. h. bei Festlegung von <code>SystemType</code> auf <code>enhanced</code>).</p>		
HealthCheckSuccessThreshold	<p>Der Grenzwert herabgesetzt werden, sodass Instances die Zustandsprüfungen bestehen.</p> <p>Note</p> <p>Wenn Sie die Elastic Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert (p. 648).</p>	Ok	Ok Warning Degraded Severe

aws:elasticbeanstalk:hostmanager

Konfigurieren Sie die EC2-Instances der Umgebung für den Upload von rotierten Protokollen in Amazon S3.

Namespace: **aws:elasticbeanstalk:hostmanager**

Name	Beschreibung	Standard	proxy:staticfiles Gültige Werte
LogPublicationControl	Kopieren Sie die Protokolldateien für die Amazon EC2-Instances der Anwendung in den Amazon S3-Bucket dieser Anwendung.	false	true false

aws:elasticbeanstalk:managedactions

Konfigurieren Sie verwaltete Plattformaktualisierungen für die Umgebung.

Namespace: **aws:elasticbeanstalk:managedactions**

Name	Beschreibung	Standard	Zulässige Werte
ManagedActionsEnabled	Aktivieren Sie verwaltete Plattformaktualisierungen (p. 501) .	true	true false

Name	Beschreibung	Standard	Zulässige Werte
	Wenn Sie diesen Parameter auf <code>true</code> setzen, müssen Sie auch einen Wert für <code>PreferredStartTime</code> und <code>UpdateLevel</code> (p. 694) angeben.		
PreferredStartTime	Konfigurieren Sie ein Wartungsfenster für verwaltete Aktionen in UTC (koordinierter Weltzeit). Beispiel, "Tue:09:00".	Keine	Tag und Uhrzeit im Format Tag:Stunde:Minute .
ServiceRoleForManagedUpdates	Der Name einer IAM-Rolle, die Elastic Beanstalk verwendet, um verwaltete Plattform-Updates für Ihre Umgebung durchzuführen. Sie können entweder dieselbe Rolle verwenden, die Sie für die <code>ServiceRole</code> -Option des <code>aws:elasticbeanstalk:environment-</code> Namespaces angegeben haben, oder die serviceverknüpfte Rolle für verwaltete Updates (p. 921) Ihres Kontos. Wenn im zweiten Fall das Konto noch nicht über eine serviceverknüpfte Rolle für verwaltete Updates verfügt, wird sie von Elastic Beanstalk erstellt.	Keine	Identisch mit <code>ServiceRole</code> oder <code>AWSServiceRoleForElasticBeanstalk</code>

aws:elasticbeanstalk:managedactions:platformupdate

Konfigurieren Sie verwaltete Plattformaktualisierungen für die Umgebung.

Namespace: **aws:elasticbeanstalk:managedactions:platformupdate**

Name	Beschreibung	Standard	Zulässige Werte
UpdateLevel	Die höchste Aktualisierungsebene, die mit verwalteten Plattformaktualisierungen möglich ist. Plattformen sind nach dem Schema Hauptversion.Nebenversion.Patch-Version aufgegliedert. Beispielsweise gibt die Plattformversion 2.0.8 die Hauptversion 2, die Nebenversion 0 und die Patch-Version 8 an.	Keine	patch nur für Patch-Versionsaktualisierungen. minor für sowohl Neben- als auch Patch-Versionsaktualisierungen.

Name	Beschreibung	Standard	Zulässige Werte
InstanceRefreshEnabled	Damit aktivieren Sie den wöchentlichen Instance-Ersatz. Der Parameter <code>ManagedActionsEnabled</code> muss auf den Wert <code>true</code> festgelegt sein.	false	true false

aws:elasticbeanstalk:monitoring

Konfigurieren Sie die Umgebung so, dass EC2-Instances mit nicht bestandenen Zustandsprüfungen beendet werden.

Namespace: **aws:elasticbeanstalk:monitoring**

Name	Beschreibung	Standard	Zulässige Werte
Automatically Terminate Unhealthy Instances	<p>Wenn eine Instance die Zustandsprüfungen nicht besteht, wird sie beendet.</p> <p>Note</p> <p>Diese Option wurde nur in Legacy-Umgebungen (p. 502) unterstützt. Sie bestimmt den Zustand einer Instance basierend auf ihrer Erreichbarkeit und auf anderen instance-basierten Metriken. Elastic Beanstalk bietet keine Möglichkeit zum automatischen Beenden von Instances anhand des Anwendungszustands.</p>	true	true false

aws:elasticbeanstalk:sns:topics

Konfigurieren Sie Benachrichtigungen für die Umgebung.

Namespace: **aws:elasticbeanstalk:sns:topics**

Name	Beschreibung	Standard	Zulässige Werte
Notification Endpoint	Gibt den Endpunkt an, über den Sie Benachrichtigungen zu wichtigen Ereignissen für die Anwendung erhalten möchten.	Keine	

Name	Beschreibung	Standard	Zulässige Werte
	<p>Note</p> <p>Wenn Sie die Elastic Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert (p. 648).</p>		
Notification Protocol	Gibt das Protokoll an, über das Benachrichtigungen an Ihren Endpunkt gesendet werden.	email	http https email email-json sqS
Notification Topic ARN	Gibt den Amazon-Ressourcennamen (ARN) für das abonnierte Thema an.	Keine	
Notification Topic Name	Gibt den Namen des abonnierten Themas an.	Keine	

aws:elasticbeanstalk:sqsd

Konfigurieren Sie die Amazon SQS-Warteschlange für eine Worker-Umgebung.

Namespace: **aws:elasticbeanstalk:sqsd**

Name	Beschreibung	Standard	Zulässige Werte
WorkerQueueURL	<p>Die URL der Warteschlange, aus der Nachrichten vom Daemon der Worker-Umgebungsebene gelesen werden.</p> <p>Note</p> <p>Wenn Sie keinen Wert angeben, handelt es sich bei der Warteschlange, die Elastic Beanstalk automatisch erstellt, um eine Standard-Amazon SQS-Warteschlange. Wenn Sie einen Wert angeben, können Sie die URL einer Standard- oder einer FIFO-Amazon SQS-Warteschlange</p>	Wird automatisch generiert.	Wenn Sie keinen Wert angeben, erstellt Elastic Beanstalk automatisch eine Warteschlange.

Name	Beschreibung	Standard	Zulässige Werte
	angeben. Beachten Sie, dass bei Angabe einer FIFO-Warteschlange regelmäßige Aufgaben (p. 519) nicht unterstützt werden.		
HttpPath	Gibt den relativen Pfad zur Anwendung an, an die HTTP POST-Nachrichten gesendet werden.	/	
MimeType	Gibt den MIME-Typ der Nachricht an, die mit der HTTP POST-Anforderung gesendet wird.	application/json	application/json application/x-www-form-urlencoded application/xml text/plain Benutzerdefinierter MIME-Typ.
HttpConnections	Gibt die maximale Anzahl gleichzeitiger Verbindungen zu den Anwendungen auf einer Amazon EC2-Instance an. Note Wenn Sie die Elastic Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert (p. 648) .	50	1 auf 100
ConnectTimeout	Gibt die Zeitspanne (in Sekunden) an, die auf erfolgreiche Verbindungserstellungen zu einer Anwendung gewartet wird.	5	1 auf 60
InactivityTimeout	Gibt die Zeitspanne (in Sekunden) an, die bei einer bestehenden Verbindung zu einer Anwendung auf eine Antwort gewartet wird. Die Nachricht wird solange erneut verarbeitet, bis der Daemon die Antwort "200 OK" von der Anwendung der Worker-Umgebungsebene erhält oder der <code>RetentionPeriod</code> -Wert abgelaufen ist.	299	1 auf 36000

Name	Beschreibung	Standard	Zulässige Werte
VisibilityTimeout	Gibt die Zeitspanne (in Sekunden) an, die eine eingehende Nachricht aus der Amazon SQS-Warteschlange zur Verarbeitung gesperrt ist. Nach Ablauf des konfigurierten Zeitraums wird die Nachricht in der Warteschlange wieder sichtbar und kann von einem anderen Daemon gelesen werden.	300	0 auf 43200
ErrorVisibilityTimeout	Gibt die Zeitspanne (in Sekunden) an, die verstreichen muss, bevor Elastic Beanstalk eine Nachricht nach einem fehlgeschlagenen Verarbeitungsversuch mit einem expliziten Fehler an die Amazon SQS-Warteschlange zurückgibt.	2 Sekunden	0 bis 43200 Sekunden
RetentionPeriod	Gibt die Zeitspanne (in Sekunden) an, die eine Nachricht gültig ist und aktiv verarbeitet wird.	345600	60 auf 1209600
MaxRetries	Gibt die maximale Anzahl der Wiederholversuche an, mit denen Elastic Beanstalk versuchen kann, die Nachricht zur Verarbeitung an die entsprechende Webanwendung zu senden, bevor die Nachricht in die Warteschlange für unzustellbare Nachrichten verschoben wird.	10	1 auf 100

aws:elasticbeanstalk:trafficsplitting

Konfigurieren Sie Bereitstellungen für die Datenverkehrsaufteilung für Ihre Umgebung.

Dieser Namespace wird angewendet, wenn Sie die `DeploymentPolicy`-Option des Namespace `aws:elasticbeanstalk:command` (p. 685) auf `TrafficSplitting` festlegen. Weitere Informationen zu Bereitstellungsrichtlinien finden Sie unter [the section called “Optionen für die Bereitstellung” \(p. 474\)](#).

Namespace: `aws:elasticbeanstalk:trafficsplitting`

Name	Beschreibung	Standard	Zulässige Werte
NewVersionPercent	Der anfängliche Prozentsatz des eingehenden Clientdatenverkehrs, den Elastic Beanstalk auf Umgebungs-Instances verlagert, auf denen die von Ihnen bereitgestellte neue Anwendungsversion ausgeführt wird.	10	1 auf 100
EvaluationTime	Der Zeitraum (in Minuten), den Elastic Beanstalk nach einer anfänglich fehlerfreien Bereitstellung wartet, bevor der gesamte eingehende Clientdatenverkehr auf die	5	3 auf 600

Name	Beschreibung	Standard	Zulässige Werte
	von Ihnen bereitgestellte neue Anwendungsversion verschoben wird.		

aws:elasticbeanstalk:xray

Führen Sie den AWS X-Ray-Daemon aus, um Trace-Informationen aus Ihrer [X-Ray-integrierten \(p. 627\)](#) Anwendung weiterzuleiten.

Namespace: **aws:elasticbeanstalk:xray**

Name	Beschreibung	Standard	Zulässige Werte
XRayEnabled	Auf true setzen, um den X-Ray-Daemon auf den Instances in Ihrer Umgebung auszuführen.	false	true false

aws:elb:healthcheck

Konfigurieren Sie Zustandsprüfungen für den Classic Load Balancer.

Namespace: **aws:elb:healthcheck**

Name	Beschreibung	Standard	Zulässige Werte
HealthyThreshold	Aufeinanderfolgende erfolgreiche Anforderungen vor Elastic Load Balancing ändert den Instance-Zustand.	3	2 auf 10
Interval	Gibt an, wie häufig Elastic Load Balancing Zustandsprüfungen für die Amazon EC2-Instances der Anwendung ausführt.	10	5 auf 300
Timeout	Gibt die Anzahl der Sekunden an, die Elastic Load Balancing auf eine Antwort wartet, bevor die Instance als nicht mehr reagierend eingestuft wird.	5	2 auf 60
UnhealthyThreshold	Aufeinanderfolgende erfolglose Anforderungen vor Elastic Load Balancing ändert den Instance-Zustand.	5	2 auf 10
(veraltet) Target	Gibt das Ziel auf der Backend-Instance an, an das die Zustandsprüfungen gesendet werden sollen. Verwenden Sie stattdessen Application Healthcheck URL im Namespace aws:elasticbeanstalk:application (p. 683) .	TCP:80	Ziel im Format PROTOKOLL:PORT/PFAD

aws:elb:loadbalancer

Konfigurieren Sie den Classic Load Balancer der Umgebung.

Mehrere Optionen in diesem Namespace gelten als veraltet und werden durch die Listener-spezifischen Optionen im Namespace [aws:elb:listener \(p. 701\)](#) ersetzt. Mit den veralteten Optionen können nur zwei Listener (ein sicherer und ein nicht sicherer) auf Standard-Ports konfiguriert werden.

Namespace: **aws:elb:loadbalancer**

Name	Beschreibung	Standard	Zulässige Werte
CrossZone	<p>Konfigurieren Sie den Load Balancer so, dass der Datenverkehr gleichmäßig auf alle Instances in allen Availability Zones (und nicht nur in den einzelnen Zonen) weitergeleitet wird.</p> <p>Note</p> <p>Wenn Sie die Elastic Beanstalk-Konsole oder EB CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole und EB CLI überschreiben diese Option mit einem empfohlenen Wert (p. 648).</p>	false	true false
SecurityGroups	Ordnen Sie dem Load Balancer eine oder mehrere der von Ihnen erstellten Sicherheitsgruppen zu.	Keine	Eine oder mehrere Sicherheitsgruppen-IDs.
ManagedSecurityGroup	<p>Ordnen Sie dem Load Balancer der Umgebung eine vorhandene Sicherheitsgruppe zu, anstatt eine neue zu erstellen. Wenn Sie diese Einstellung nutzen möchten, müssen Sie die Einstellung <code>SecurityGroups</code> in diesem Namespace aktualisieren, damit die ID Ihrer Sicherheitsgruppe einbezogen und die ID der automatisch generierten Sicherheitsgruppe (sofern vorhanden) gelöscht wird.</p> <p>Um den Datenverkehr vom Load Balancer zu den EC2-Instances zu ermöglichen, fügt Elastic Beanstalk eine Regel, die eingehenden Datenverkehr von der verwalteten Sicherheitsgruppe zulässt, zur Sicherheitsgruppe der Instance hinzu.</p>	Keine	Eine Sicherheitsgruppen-ID.
(veraltet) LoadBalancerHTTPPort	Gibt den zu überwachenden Port für den nicht sicheren Listener an.	80	OFF 80
(veraltet) LoadBalancerPortProtocol	Gibt das zu verwendende Protokoll für den nicht sicheren Listener an.	HTTP	HTTP TCP
(veraltet) LoadBalancerHTTPSPort	Gibt den zu überwachenden Port für den sicheren Listener an.	OFF	OFF 443 8443
(veraltet) LoadBalancerSSLPortProtocol	Gibt das zu verwendende Protokoll für den sicheren Listener an.	HTTPS	HTTPS SSL

Name	Beschreibung	Standard	Zulässige Werte
(veraltet) SSLCertificateId	Gibt den ARN eines SSL-Zertifikats an, das an den sicheren Listener gebunden werden soll.	Keine	

aws:elb:listener

Konfigurieren Sie den Standard-Listener (Port 80) für einen Classic Load Balancer.

Namespace: **aws:elb:listener**

Name	Beschreibung	Standard	Zulässige Werte
ListenerProtocol	Gibt das vom Listener verwendete Protokoll an.	HTTP	HTTP TCP
InstancePort	Port, der von diesem Listener für die Kommunikation mit den EC2-Instances verwendet wird.	80	1 auf 65535
InstanceProtocol	<p>Protokoll, das von diesem Listener für die Kommunikation mit den EC2-Instances verwendet wird.</p> <p>Die Internetprotokoll-Ebene muss mit <code>ListenerProtocol</code> übereinstimmen. Außerdem muss die Sicherheitsebene mit allen anderen Listenern identisch sein, die denselben <code>InstancePort</code> nutzen wie dieser Listener.</p> <p>Wenn <code>ListenerProtocol</code> beispielsweise auf <code>HTTPS</code> (Anwendungsebene, über eine sichere Verbindung) festgelegt ist, können Sie <code>InstanceProtocol</code> auf <code>HTTP</code> (ebenfalls Anwendungsebene, aber unsichere Verbindung) einstellen. Falls Sie zudem <code>InstancePort</code> auf 80 festgelegt haben, müssen Sie <code>InstanceProtocol</code> in allen anderen Listenern, bei denen <code>HTTP</code> auf <code>InstancePort</code> festgelegt ist, auf 80 einstellen.</p>	HTTP, wenn <code>ListenerProtocol</code> den Wert <code>HTTP</code> hat. TCP, wenn <code>ListenerProtocol</code> den Wert <code>TCP</code> hat.	HTTP oder HTTPS , wenn <code>ListenerProtocol</code> den Wert <code>HTTP</code> oder <code>HTTPS</code> hat. TCP oder SSL, wenn <code>ListenerProtocol</code> den Wert <code>TCP</code> oder <code>SSL</code> hat.
PolicyNames	Eine durch Komma getrennte Liste mit Richtliniennamen, die für den Port dieses Listeners gelten. Es wird empfohlen, stattdessen die Option "LoadBalancerPorts" im Namespace aws:elb:policies (p. 703) zu verwenden.	Keine	
ListenerEnabled	Gibt an, ob dieser Listener aktiviert ist. Wenn Sie den Wert <code>false</code> festlegen, wird der Listener nicht in den Load Balancer einbezogen.	true	true false

aws:elb:listener:listener_port

Konfigurieren Sie zusätzliche Listener für einen Classic Load Balancer.

Namespace: **aws:elb:listener:*listener_port***

Name	Beschreibung	Standard	Zulässige Werte
ListenerProtocol	Gibt das vom Listener verwendete Protokoll an.	HTTP	HTTP HTTPS TCP SSL
InstancePort	Port, der von diesem Listener für die Kommunikation mit den EC2-Instances verwendet wird.	Entspricht <i>listener_port</i> .	1 auf 65535
InstanceProtocol	<p>Protokoll, das von diesem Listener für die Kommunikation mit den EC2-Instances verwendet wird.</p> <p>Die Internetprotokoll-Ebene muss mit <i>ListenerProtocol</i> übereinstimmen. Außerdem muss die Sicherheitsebene mit allen anderen Listenern identisch sein, die denselben <i>InstancePort</i> nutzen wie dieser Listener.</p> <p>Wenn <i>ListenerProtocol</i> beispielsweise auf HTTPS (Anwendungsebene, über eine sichere Verbindung) festgelegt ist, können Sie <i>InstanceProtocol</i> auf HTTP (ebenfalls Anwendungsebene, aber unsichere Verbindung) einstellen. Falls Sie zudem <i>InstancePort</i> auf 80 festgelegt haben, müssen Sie <i>InstanceProtocol</i> in allen anderen Listenern, bei denen HTTP auf <i>InstancePort</i> festgelegt ist, auf 80 einstellen.</p>	HTTP, wenn <i>ListenerProtocol</i> den Wert HTTP oder HTTPS hat. TCP, wenn <i>ListenerProtocol</i> den Wert TCP oder SSL hat.	HTTP oder HTTPS, wenn <i>ListenerProtocol</i> den Wert HTTP oder HTTPS hat. SSL, wenn <i>ListenerProtocol</i> den Wert TCP oder SSL hat.
PolicyNames	Eine durch Komma getrennte Liste mit Richtliniennamen, die für den Port dieses Listeners gelten. Es wird empfohlen, stattdessen die Option "LoadBalancerPorts" im Namespace aws:elb:policies (p. 703) zu verwenden.	Keine	
SSLCertificateId	ARN eines SSL-Zertifikats, das an den Listener gebunden werden soll.	Keine	
ListenerEnabled	Gibt an, ob dieser Listener aktiviert ist. Wenn Sie den Wert false festlegen, wird der Listener nicht in den Load Balancer einbezogen.	true, wenn eine andere Option festgelegt ist; ansonsten false.	true false

aws:elb:policies

Ändern Sie bei einem Classic Load Balancer die Sticky Sessions-Standardrichtlinien sowie die globalen Load Balancer-Richtlinien.

Namespace: **aws:elb:policies**

Name	Beschreibung	Standard	Zulässige Werte
ConnectionDrainingEnabled	Gibt an, ob der Load Balancer bestehende Verbindungen zu instabil gewordenen oder abgemeldeten Instances halten soll, um laufende Anforderungen abzuschließen. Note Wenn Sie die Elastic Beanstalk-Konsole oder EB CLI verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole und EB CLI überschreiben diese Option mit einem empfohlenen Wert (p. 648) .	false	true false
ConnectionDrainingTimeout	Gibt an, wie viele Sekunden der Load Balancer bei einem Verbindungsausgleich bestehende Verbindungen zu einer Instance maximal halten soll, bevor das Beenden der Verbindungen erzwungen wird. Note Wenn Sie die Elastic Beanstalk-Konsole verwenden, um eine Umgebung zu erstellen, können Sie diese Option nicht in einer Konfigurationsdatei (p. 722) festlegen. Die Konsole überschreibt diese Option mit einem empfohlenen Wert (p. 648) .	20	1 auf 3600
ConnectionSettingIdleTimeout	Gibt die Anzahl der Sekunden an, die der Load Balancer wartet, dass Daten über die Verbindung gesendet oder empfangen werden. Wenn nach Verstreichen dieses Zeitraums keine Daten versendet oder empfangen wurden, schließt der Load Balancer die Verbindung.	60	1 auf 3600
LoadBalancerPorts	Eine durch Komma getrennte Liste der Listener-Ports, für die die Standardrichtlinie (AWSEB-ELB-StickinessPolicy) gilt.	Keine	Mit :all können Sie alle Listener-Ports angeben.
Stickiness Cookie Expiration	Gibt die Zeitspanne (in Sekunden) an, die ein Cookie gültig ist. Von diesem Parameter wird die Standardrichtlinie (AWSEB-ELB-StickinessPolicy) verwendet.	0	0 auf 1000000

Name	Beschreibung	Standard	Zulässige Werte
Stickiness Policy	Damit wird eine Benutzersitzung an eine bestimmte Server-Instance gebunden, sodass alle vom Benutzer während der Sitzung gesendeten Anforderungen an die gleiche Server-Instance übermittelt werden. Von diesem Parameter wird die Standardrichtlinie (AWSEB-ELB-StickinessPolicy) verwendet.	false	true false

aws:elb:policies:policy_name

Erstellen Sie zusätzliche Load Balancer-Richtlinien für einen Classic Load Balancer.

Namespace: **aws:elb:policies:*policy_name***

Name	Beschreibung	Standard	Zulässige Werte
CookieName	Der Name des (von der Anwendung generierten) Cookies, mit dem die Sitzungsgültigkeitsdauer einer AppCookieStickinessPolicyType-Richtlinie gesteuert wird. Diese Richtlinie kann nur mit HTTP/HTTPS-Listenern verknüpft werden.	Keine	
InstancePorts	Eine durch Komma getrennte Liste der Instance-Ports, für die diese Richtlinie gilt.	Keine	Eine Liste mit Ports oder :all
LoadBalancerPorts	Eine durch Komma getrennte Liste der Listener-Ports, für die diese Richtlinie gilt.	Keine	Eine Liste mit Ports oder :all
ProxyProtocol	Gibt bei einer ProxyProtocolPolicyType-Richtlinie an, ob die IP-Adresse und der Port der ursprünglichen Anforderung in TCP-Nachrichten eingebunden werden sollen. Diese Richtlinie kann nur mit TCP/SSL-Listenern verknüpft werden.	Keine	true false
PublicKey	Der Inhalt eines öffentlichen Schlüssels, der von einer PublicKeyPolicyType-Richtlinie zur Authentifizierung der Backend-Server oder der Server verwendet wird. Diese Richtlinie kann nicht direkt für Backend-Server oder Listener eingesetzt werden, sondern muss Teil einer BackendServerAuthenticationPolicyType-Richtlinie sein.	Keine	
PublicKeyPolicyNames	Eine durch Komma getrennte Liste der Richtliniennamen (aus den PublicKeyPolicyType-Richtlinien) für eine BackendServerAuthenticationPolicyType-Richtlinie, mit der die Authentifizierung am Backend-Server oder an den Servern gesteuert wird. Diese Richtlinie kann nur	Keine	

Name	Beschreibung	Standard	Zulässige Werte
	mit Backend-Servern verknüpft werden, die HTTPS/SSL verwenden.		
SSLProtocols	Eine durch Komma getrennte Liste der SSL-Protokolle, die für eine <code>SSLNegotiationPolicyType</code> -Richtlinie aktiviert werden sollen, in der die vom Load Balancer akzeptierten Verschlüsselungsverfahren und Protokolle definiert werden. Diese Richtlinie kann nur mit HTTPS/SSL-Listenern verknüpft werden.	Keine	
SSLReferencePolicy	Der Name einer vordefinierten Sicherheitsrichtlinie, die bewährte AWS-Sicherheitsmethoden befolgt und für eine <code>SSLNegotiationPolicyType</code> -Richtlinie aktiviert werden soll, in der die vom Load Balancer akzeptierten Verschlüsselungsverfahren und Protokolle definiert werden. Diese Richtlinie kann nur mit HTTPS/SSL-Listenern verknüpft werden.	Keine	
Stickiness Cookie Expiration	Gibt die Zeitspanne (in Sekunden) an, die ein Cookie gültig ist.	0	0 auf 1000000
Stickiness Policy	Damit wird eine Benutzersitzung an eine bestimmte Server-Instance gebunden, sodass alle vom Benutzer während der Sitzung gesendeten Anforderungen an die gleiche Server-Instance übermittelt werden.	false	true false

aws:elbv2:listener:default

Konfigurieren Sie den Standard-Listener (Port 80) auf einem Application Load Balancer oder einem Network Load Balancer.

Dieser Namespace gilt nicht für eine Umgebung, die einen freigegebenen Load Balancer verwendet. Freigegebene Load Balancer besitzen keinen Standard-Listener.

Namespace: **aws:elbv2:listener:default**

Name	Beschreibung	Standard	Zulässige Werte
DefaultProcess	Name des Prozesses (p. 687) , an den der Datenverkehr weitergeleitet wird, falls keine Regel zutrifft.	default	Ein Prozessname.
ListenerEnabled	Legen Sie den Wert auf <code>false</code> fest, um den Listener zu deaktivieren. Mit dieser Option können Sie den Standard-Listener auf Port 80 deaktivieren.	true	true false

Name	Beschreibung	Standard	Zulässige Werte
Protocol	Protokoll des zu verarbeitenden Datenverkehrs.	Bei Application Load Balancer: <code>HTTP</code> Bei Network Load Balancer: <code>TCP</code>	Bei Application Load Balancer: <code>HTTP</code> , <code>HTTPS</code> Bei Network Load Balancer: <code>TCP</code>
Rules	Liste der Regeln (p. 708) , die für diesen Listener gelten sollen. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	Keine	Eine durch Komma getrennte Liste der Regelnamen.
SSLCertificateArns	ARN des SSL-Zertifikats, das an den Listener gebunden werden soll. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	Keine	Der ARN eines in IAM oder ACM gespeicherten Zertifikats.
SSLPolicy	Geben Sie eine Sicherheitsrichtlinie an, die für den Listener gelten soll. Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.	Keine (ELB-Standard)	Der Name einer Load Balancer-Sicherheitsrichtlinie.

aws:elbv2:listener:listener_port

Konfigurieren Sie zusätzliche Listener auf einem Application Load Balancer oder einem Network Load Balancer.

Note

Für einen gemeinsam genutzten Application Load Balancer können Sie nur die `Rule` Option angeben. Die übrigen Optionen können nicht auf freigegebene Load Balancer angewendet werden.

Namespace: **aws:elbv2:listener:`listener_port`**

Name	Beschreibung	Standard	Zulässige Werte
DefaultProcess	Name des Prozesses (p. 687) , an den der Datenverkehr weitergeleitet wird, falls keine Regel zutrifft.	<code>default</code>	Ein Prozessname.

Name	Beschreibung	Standard	Zulässige Werte
ListenerEnabled	Legen Sie den Wert auf <code>false</code> fest, um den Listener zu deaktivieren. Mit dieser Option können Sie den Standard-Listener auf Port 80 deaktivieren.	<code>true</code>	<code>true</code> <code>false</code>
Protocol	Protokoll des zu verarbeitenden Datenverkehrs.	Bei Application Load Balancer: <code>HTTP</code> Bei Network Load Balancer: <code>TCP</code>	Bei Application Load Balancer: <code>HTTP</code> , <code>HTTPS</code> Bei Network Load Balancer: <code>TCP</code>
Rules	<p>Liste der Regeln (p. 708), die für diesen Listener gelten sollen.</p> <p>Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.</p> <p>Wenn Ihre Umgebung einen freigegebenen Application Load Balancer verwendet und Sie diese Option für keinen Listener angeben, ordnet Elastic Beanstalk die Regel <code>default</code> automatisch einem Port 80-Listener zu.</p>	Keine	Eine durch Komma getrennte Liste der Regelnamen.
SSLCertificateArns	<p>ARN des SSL-Zertifikats, das an den Listener gebunden werden soll.</p> <p>Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.</p>	Keine	Der ARN eines in IAM oder ACM gespeicherten Zertifikats.
SSLPolicy	<p>Geben Sie eine Sicherheitsrichtlinie an, die für den Listener gelten soll.</p> <p>Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.</p>	Keine (ELB-Standard)	Der Name einer Load Balancer-Sicherheitsrichtlinie.

aws:elbv2:listenerrule:rule_name

Definieren Sie Listener-Regeln für einen Application Load Balancer. Wenn eine Anforderung dem Host-Namen oder Pfaden in einer Regel entspricht, leitet der Load Balancer sie an den angegebenen Prozess weiter. Wenn Sie eine Regel verwenden möchten, fügen Sie sie dem Listener mit der Option `Rules` im [aws:elbv2:listener:listener_port \(p. 706\)](#)-Namespace zu.

Note

Dieser Namespace ist nicht auf Umgebungen mit einem Network Load Balancer anwendbar.

Namespace: aws:elbv2:listenerrule:rule_name

Name	Beschreibung	Standard	Zulässige Werte
HostHeaders	<p>Liste mit den zuzuordnenden Host-Namen. Beispiel, my.example.com.</p>	Dedizierter Load Balancer: Keiner Freigegebener Load Balancer: der CNAME der Umgebung	Jeder Name darf maximal 128 Zeichen (A bis Z, a bis z, 0 bis 9, –) und bis zu drei Platzhalterzeichen enthalten. (* entspricht keinem oder mehr Zeichen; ? entspricht genau einem Zeichen.) Sie können mehrere durch Komma getrennte Namen hinzufügen. Application Load Balancer unterstützt bis zu fünf kombinierte HostHeader- und PathPattern-Regeln. Weitere Informationen finden Sie unter Hostbedingungen im Benutzerhandbuch für Application Load Balancers.
PathPatterns	<p>Pfadmuster, mit denen Übereinstimmung erzielt werden soll. Beispiel, /img/*.</p> <p>Diese Option ist nur auf Umgebungen mit einem Application Load Balancer anwendbar.</p>	Keine	Jedes Muster darf maximal 128 Zeichen (A bis Z, a bis z, 0 bis 9, –) und bis zu drei Platzhalterzeichen enthalten. (* entspricht keinem oder mehr Zeichen; ? entspricht genau einem Zeichen.) Sie können mehrere durch Komma getrennte Pfadmuster hinzufügen. Application Load Balancer unterstützt bis zu fünf kombinierte HostHeader- und PathPattern-Regeln.

Name	Beschreibung	Standard	Zulässige Werte
			Weitere Informationen finden Sie unter Pfadbedingungen im Benutzerhandbuch für Application Load Balancers.
Priority	<p>Gibt die Priorität dieser Regel an, wenn mehrere Regeln zutreffen. Die niedrigere Zahl hat Vorrang. Zwei Regeln dürfen nicht dieselbe Priorität haben.</p> <p>Im Fall eines freigegebenen Load Balancer behandelt Elastic Beanstalk Regelprioritäten in freigebenden Umgebungen als relativ und ordnet sie während der Erstellung absoluten Prioritäten zu.</p>	1	1 auf 1000
Process	Name des Prozesses (p. 687), an den der Datenverkehr weitergeleitet wird, wenn diese Regel auf die Anforderung zutrifft.	default	Ein Prozessname.

aws:elbv2:loadbalancer

Konfigurieren Sie einen Application Load Balancer.

Im Fall eines freigegebenen Load Balancer sind nur die Optionen `SecurityGroups` und `SharedLoadBalancer` gültig.

Note

Dieser Namespace ist nicht auf Umgebungen mit einem Network Load Balancer anwendbar.

Namespace: `aws:elbv2:loadbalancer`

Name	Beschreibung	Standard	Zulässige Werte
AccessLogsS3Bucket	Amazon S3-Bucket, in dem Zugriffsprotokolle gespeichert werden sollen. Der Bucket muss in derselben Region sein wie die Umgebung und Schreibzugriff für den Load Balancer gewähren.	Keine	Ein Bucket-Name.
AccessLogsS3Enabled	Damit aktivieren Sie die Speicherung von Zugriffsprotokollen.	false	true false
AccessLogsS3Prefix	Dieses Präfix wird den Zugriffsprotokollnamen vorangestellt. Standardmäßig werden die Protokolle von Load Balancer in das Verzeichnis "AWSLogs" im angegebenen Bucket hochgeladen. Geben Sie ein Präfix an, um das Verzeichnis "AWSLogs" in ein anderes Verzeichnis zu verschieben.	Keine	

Name	Beschreibung	Standard	Zulässige Werte
IdleTimeout	Gibt an, wie lange auf den Abschluss einer Anforderung gewartet wird, bevor die Verbindungen zu Client und Instance beendet werden.	Keine	1 auf 3600
ManagedSecurityGroup	<p>Ordnen Sie dem Load Balancer der Umgebung eine vorhandene Sicherheitsgruppe zu, anstatt eine neue zu erstellen. Wenn Sie diese Einstellung nutzen möchten, müssen Sie die Einstellung <code>SecurityGroups</code> in diesem Namespace aktualisieren, damit die ID Ihrer Sicherheitsgruppe einbezogen und die ID der automatisch generierten Sicherheitsgruppe (sofern vorhanden) gelöscht wird.</p> <p>Um den Datenverkehr vom Load Balancer zu den EC2-Instances zu ermöglichen, fügt Elastic Beanstalk eine Regel, die eingehenden Datenverkehr von der verwalteten Sicherheitsgruppe zulässt, zur Sicherheitsgruppe der Instance hinzu.</p>	Die Sicherheitsgruppe, die Elastic Beanstalk für den Load Balancer erstellt.	Eine Sicherheitsgruppen-ID.
SecurityGroups	<p>Liste der Sicherheitsgruppen, die dem Load Balancer zugeordnet werden sollen.</p> <p>Wenn Sie diesen Wert für einen freigegebenen Load Balancer nicht angeben, prüft Elastic Beanstalk, ob bereits eine vorhandene Sicherheitsgruppe, die von ihm verwaltet wird, dem Load Balancer angefügt ist. Ist dies nicht der Fall, erstellt Elastic Beanstalk eine Sicherheitsgruppe und fügt diese an den Load Balancer an. Elastic Beanstalk löscht diese Sicherheitsgruppe, wenn die letzte Umgebung, die den Load Balancer teilt, beendet wird.</p> <p>Die Load Balancer-Sicherheitsgruppen werden verwendet, um die Regel für den eingehenden Datenverkehr für die Amazon EC2-Instance-Sicherheitsgruppe einzurichten.</p>	Die Sicherheitsgruppe, die Elastic Beanstalk für den Load Balancer erstellt.	Durch Komma getrennte Liste der Sicherheitsgruppen-IDs.

Name	Beschreibung	Standard	Zulässige Werte
SharedLoadBalancer	<p>Der ARN eines freigegebenen Load Balancer. Diese Option ist nur für einen Application Load Balancer relevant. Sie muss festgelegt werden, wenn die Option <code>LoadBalancerIsShared</code> des Namespace aws:elasticbeanstalk:environment (p. 686) auf <code>true</code> festgelegt ist. Sie können den ARN des freigegebenen Load Balancer nach Erstellung der Umgebung nicht ändern.</p> <p>Kriterien für einen gültigen Wert:</p> <ul style="list-style-type: none"> • Es muss sich um einen gültigen, aktiven Load Balancer in der AWS-Region der Umgebung handeln. • Es muss sich in derselben Amazon Virtual Private Cloud (Amazon VPC) wie die Umgebung befinden. • Es darf sich nicht um einen Load Balancer handeln, der von Elastic Beanstalk als dedizierter Load Balancer für eine andere Umgebung erstellt wurde. Sie können diese dedizierten Load Balancer anhand des Präfixes <code>awseb-</code> identifizieren. <p>Beispiel:</p> <pre>arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/FrontEndLB/0dbf78d8ad96abbc</pre>	Keine	ARN eines gültigen Load Balancer, der die hier beschriebenen Kriterien erfüllt.

aws:rds:dbinstance

Konfigurieren Sie eine angeschlossene Amazon RDS DB-Instance.

Namespace: **aws:rds:dbinstance**

Name	Beschreibung	Standard	Zulässige Werte
DBAllocatedStorage	Die in Gigabyte (GB) angegebene Größe des zugewiesenen Datenbankspeichers.	MySQL: 5 Oracle: 10 sqlserver-se: 200 sqlserver-ex: 30 sqlserver-web: 30	MySQL: 5-1024 Oracle: 10-1024 sqlserver: nicht änderbar

Name	Beschreibung	Standard	Zulässige Werte
DBDeletionPolicy	Damit wird festgelegt, ob von der DB-Instance ein Snapshot erstellt oder diese gelöscht werden soll, wenn die Umgebung beendet wird. Warning Das Löschen einer DB-Instance führt zu einem permanenten Datenverlust.	Delete	Delete Snapshot
DBEngine	Der Name der Datenbank-Engine, die für diese Instance verwendet werden soll.	mysql	mysql oracle-se1 sqlserver-ex sqlserver-web sqlserver-se postgres
DBEngineVersion	Die Versionsnummer des Datenbank-Engines.	5.5	
DBInstanceClass	Der Typ der Datenbank-Instance.	db.t2.micro (db.m1.large für eine Umgebung, die nicht in einer Amazon VPC ausgeführt wird)	Wechseln Sie zu DB-Instance-Klasse im Amazon Relational Database Service-Benutzerhandbuch.
DBPassword	Der Name des Masterbenutzerkennworts dieser DB-Instance.	Keine	
DBSnapshotIdentifier	Kennung für den DB-Snapshot, der zur Wiederherstellung verwendet werden soll.	Keine	
DBUser	Der Name des Masterbenutzers dieser DB-Instance.	eroot	
MultiAZDatabase	Gibt an, ob für diese DB-Instance eine Multi-AZ-Bereitstellung erfolgen soll. Weitere Informationen zu Multi-AZ-Bereitstellungen mit Amazon Relational Database Service (RDS), finden Sie unter Regionen und Availability Zones (Verfügbarkeitszonen) im Amazon Relational Database Service-Benutzerhandbuch.	false	true false

Plattformspezifische Optionen

Einige Elastic Beanstalk-Plattformen definieren Options-Namespaces, die spezifisch für die Plattform sind. Diese Namespaces und ihre Optionen sind unterhalb jeder Plattform aufgeführt.

Note

Ehemals wurden in Leiterplattenversionen, die auf Amazon Linux AMI (vor Amazon Linux 2) basierten, die folgenden zwei Features und ihre jeweiligen Namespaces als plattformspezifische Features betrachtet und hier pro Plattform aufgelistet:

- Proxy-Konfiguration für statische Dateien – [aws:elasticbeanstalk:environment:proxy:staticfiles \(p. 691\)](#)
- Unterstützung von AWS X-Ray – [aws:elasticbeanstalk:xray \(p. 699\)](#)

In Amazon Linux 2-Plattformversionen implementiert Elastic Beanstalk diese Funktionen auf konsistente Weise auf allen unterstützenden Plattformen. Der zugehörige Namespace wird nun auf der Seite [the section called “Allgemeine Optionen” \(p. 666\)](#) aufgelistet. Sie werden auf dieser Seite nur noch für Plattformen erwähnt, deren Namespaces anders benannt sind.

Plattformen

- [Docker-Plattformoptionen \(p. 713\)](#)
- [Go-Plattformoptionen \(p. 714\)](#)
- [Java SE-Plattform – Optionen \(p. 714\)](#)
- [Optionen für Java mit Tomcat-Plattform \(p. 715\)](#)
- [.NET Core auf Linux-Plattformoptionen \(p. 716\)](#)
- [.NET-Plattformoptionen \(p. 716\)](#)
- [Node.js-Plattformoptionen \(p. 716\)](#)
- [PHP-Plattformoptionen \(p. 718\)](#)
- [Python-Plattformoptionen \(p. 719\)](#)
- [Ruby-Plattformoptionen \(p. 721\)](#)

Docker-Plattformoptionen

Die folgenden für Docker spezifischen Konfigurationsoptionen gelten für Docker- und Preconfigured Docker-Plattformen.

Note

Diese Konfigurationsoptionen gelten nicht für

- die Docker-Plattform (Amazon Linux 2) mit Docker Compose
- die Multicontainer Docker-Plattform (Amazon Linux AMI)

Namespace: **aws:elasticbeanstalk:environment:proxy**

Name	Beschreibung	Standard	Zulässige Werte
ProxyServer	Gibt den Webserver an, der als Proxy verwendet werden soll.	nginx	nginx none – Amazon Linux AM und Docker nur mit DC

Go-Plattformoptionen

Amazon Linux AMI (vor Amazon Linux 2)-Plattformoptionen

Namespace: `aws:elasticbeanstalk:container:golang:staticfiles`

Sie können den Proxy-Server mit dem folgenden Namespace konfigurieren, um statische Dateien bereitzustellen. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, statt die Anfrage an Ihre Anwendung weiterzuleiten. Dies reduziert die Anzahl der Anfragen, die Ihre Anwendung verarbeiten muss.

Bilden Sie einen von dem Proxy-Server bedienten Pfad auf einen Ordner in Ihrem Quellcode ab, der statische Komponenten enthält. Jede Option, die Sie in diesem Namespace definieren, weist einen anderen Pfad zu.

Name	Wert
Der Pfad, in dem der Proxy-Server die Dateien bereitzustellt Beispiel: Verwenden Sie <code>/images</code> , um Dateien unter <code>subdomain.elasticbeanstalk.com/images</code> bereitzustellen.	Der Name des Verzeichnisses, das die Dateien enthält Beispiel: Verwenden Sie <code>staticimages</code> , um Dateien aus einem Verzeichnis namens <code>staticimages</code> auf oberster Ebene Ihres Quell-Bundles bereitzustellen.

Java SE-Plattform – Optionen

Amazon Linux AMI (vor Amazon Linux 2)-Plattformoptionen

Namespace: `aws:elasticbeanstalk:container:java:staticfiles`

Sie können den Proxy-Server mit dem folgenden Namespace konfigurieren, um statische Dateien bereitzustellen. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, statt die Anfrage an Ihre Anwendung weiterzuleiten. Dies reduziert die Anzahl der Anfragen, die Ihre Anwendung verarbeiten muss.

Bilden Sie einen von dem Proxy-Server bedienten Pfad auf einen Ordner in Ihrem Quellcode ab, der statische Komponenten enthält. Jede Option, die Sie in diesem Namespace definieren, weist einen anderen Pfad zu.

Name	Wert
Der Pfad, in dem der Proxy-Server die Dateien bereitzustellt Beispiel: Verwenden Sie <code>/images</code> , um Dateien unter <code>subdomain.elasticbeanstalk.com/images</code> bereitzustellen.	Der Name des Verzeichnisses, das die Dateien enthält Beispiel: Verwenden Sie <code>staticimages</code> , um Dateien aus einem Verzeichnis namens <code>staticimages</code> auf oberster Ebene Ihres Quell-Bundles bereitzustellen.

Optionen für Java mit Tomcat-Plattform

Namespace: **aws:elasticbeanstalk:application:environment**

Name	Beschreibung	Standard	Zulässige Werte
JDBC_CONNECTION_STRING	Verbindungszeichenfolge für eine externe Datenbank.	–	–

Weitere Informationen finden Sie unter [Umgebungseigenschaften und andere Softwareeinstellungen \(p. 621\)](#).

Namespace: **aws:elasticbeanstalk:container:tomcat:jvmoptions**

Name	Beschreibung	Standard	Zulässige Werte
JVM Options	Übergeben von Befehlszeilen-Optionen an die JVM beim Starten.	–	–
Xmx	Maximale JVM-Heap-Größen.	256m	–
XX:MaxPermSize	Bereich des JVM-Heap, der zum Speichern von Klassendefinitionen und zugehörige Metadaten verwendet wird. Note Diese Option gilt nur für Java-Versionen vor Java 8 und wird auf Elastic Beanstalk Tomcat-Plattformen basierend auf Amazon Linux 2 nicht unterstützt.	64m	–
Xms	Ursprüngliche JVM-Heap-Größen.	256m	–
<i>optionName</i>	Geben Sie neben den von der Tomcat-Plattform definierten JVM-Optionen beliebige weitere JVM-Optionen an.	–	–

Namespace: **aws:elasticbeanstalk:environment:proxy**

Name	Beschreibung	Standard	Zulässige Werte
GzipCompression	Festlegen auf <code>false</code> , um die Antwortkomprimierung zu deaktivieren. Nur für Amazon Linux AMI-Plattformversionen gültig (vor Amazon Linux 2).	true	true false
ProxyServer	Legen Sie die Proxys fest, der für die Instances Ihrer Umgebung verwendet werden soll. Wenn Sie für diese Option apache festlegen, verwendet Elastic Beanstalk Apache 2.4 . Legen Sie sie auf <code>apache/2.2</code> fest, wenn Ihre Anwendung aufgrund einer inkompatiblen	nginx (Amazon Linux 2) apache (Amazon	apache apache/2.2 – nur Amazon Linux AMI nginx

Name	Beschreibung	Standard	Zulässige Werte
	<p>Proxy-Konfigurationseinstellung für die Migration von Apache 2.2 nicht bereit ist. Dieser Wert ist nur für Amazon Linux AMI-Plattformversionen gültig (vor Amazon Linux 2).</p> <p>Legen Sie sie auf <code>nginx</code> fest, um <code>nginx</code> zu verwenden. Dies ist die Standardeinstellung ab Amazon Linux 2-Plattformversionen.</p> <p>Weitere Informationen finden Sie unter Konfigurieren des Proxy-Servers Ihrer Tomcat-Umgebung (p. 124).</p>	Linux AMI)	

.NET Core auf Linux-Plattformoptionen

Namespace: **aws:elasticbeanstalk:environment:proxy**

Name	Beschreibung	Standard	Zulässige Werte
ProxyServer	Gibt den Webserver an, der als Proxy verwendet werden soll.	nginx	nginx none

.NET-Plattformoptionen

Namespace: **aws:elasticbeanstalk:container:dotnet:apppool**

Name	Beschreibung	Standard	Zulässige Werte
Target Runtime	Wählen Sie die Version von .NET-Framework für Ihre Anwendung.	4.0	2.0 4.0
Enable 32-bit Applications	Legen Sie <code>True</code> fest, um 32-Bit-Anwendungen auszuführen.	False	True False

Node.js-Plattformoptionen

Namespace: **aws:elasticbeanstalk:environment:proxy**

Name	Beschreibung	Standard	Zulässige Werte
ProxyServer	Legen Sie die Proxys fest, der für die Instances Ihrer Umgebung verwendet werden soll.	nginx	apache nginx

Amazon Linux AMI (vor Amazon Linux 2)-Plattformoptionen

Namespace: `aws:elasticbeanstalk:container:nodejs`

Name	Beschreibung	Standard	Zulässige Werte
NodeCommand	Befehl zum Starten der Node.js-Anwendung. Wenn eine leere Zeichenfolge angegeben wird, wird <code>app.js</code> , dann <code>server.js</code> , dann <code>npm start</code> (in dieser Reihenfolge) verwendet.	""	-
NodeVersion	<p>Version von Node.js. Beispiel, <code>4.4.6</code></p> <p>Unterstützte Node.js-Versionen variieren zwischen den Node.js-Plattformversionen. Eine Liste der aktuell unterstützten Versionen finden Sie unter Node.js im Dokument AWS Elastic Beanstalk-Plattformen.</p> <p>Note</p> <p>Wenn die von Ihnen verwendete Version von Node.js von der Plattform nicht mehr unterstützt wird, müssen Sie die Versionseinstellung vor der Ausführung einer Plattformaktualisierung (p. 491) ändern oder löschen. Dieser Fall kann eintreten, wenn eine Sicherheitslücke für eine oder mehrere Versionen von Node.js identifiziert wird. In diesem Fall tritt bei dem Versuch, ein Upgrade auf eine neue Plattformversion, die die konfigurierte NodeVersion (p. 716) nicht unterstützt, ein Fehler auf. Um keine neue Umgebung erstellen zu müssen, ändern Sie die Konfigurationsoption <code>NodeVersion</code> in eine Node.js-Version, die sowohl von der alten als auch von der neuen Plattformversion unterstützt wird, oder entfernen Sie die Optionseinstellung (p. 658) und führen Sie dann die Plattformaktualisierung aus.</p>	variert	variert
GzipCompression	Gibt an, ob die gzip-Komprimierung aktiviert ist. Wenn <code>ProxyServer</code> auf <code>none</code> festgelegt ist, ist die GZIP-Komprimierung deaktiviert.	false	true false
ProxyServer	Gibt an, welcher Webserver verwendet werden soll, um Proxy-Verbindungen zu Node.js zu verwenden. Wenn <code>ProxyServer</code> auf <code>none</code> festgelegt ist, werden statische Dateizuweisungen nicht wirksam und die gzip-Komprimierung ist deaktiviert.	nginx	apache nginx none

Namespace: `aws:elasticbeanstalk:container:nodejs:staticfiles`

Sie können den Proxy-Server mit dem folgenden Namespace konfigurieren, um statische Dateien bereitzustellen. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, statt die Anfrage an Ihre Anwendung weiterzuleiten. Dies reduziert die Anzahl der Anfragen, die Ihre Anwendung verarbeiten muss.

Bilden Sie einen von dem Proxy-Server bedienten Pfad auf einen Ordner in Ihrem Quellcode ab, der statische Komponenten enthält. Jede Option, die Sie in diesem Namespace definieren, weist einen anderen Pfad zu.

Note

Statische Dateieinstellungen werden nicht angewendet, wenn `aws:elasticbeanstalk:container:nodejs::ProxyFiles` auf `none` festgelegt ist.

Name	Wert
Der Pfad, in dem der Proxy-Server die Dateien bereitzustellt Beispiel: Verwenden Sie <code>/images</code> , um Dateien unter <code>subdomain.elasticbeanstalk.com/images</code> bereitzustellen.	Der Name des Verzeichnisses, das die Dateien enthält Beispiel: Verwenden Sie <code>staticimages</code> , um Dateien aus einem Verzeichnis namens <code>staticimages</code> auf oberster Ebene Ihres Quell-Bundles bereitzustellen.

PHP-Plattformoptionen

Namespace: `aws:elasticbeanstalk:container:php:phpini`

Name	Beschreibung	Standard	Zulässige Werte
document_root	Geben Sie das untergeordnete Verzeichnis Ihres Projekts an, das als der öffentlich zugängliche Webstamm behandelt wird.	/	Es wird eine leere Zeichenfolge als / behandelt oder geben Sie eine Zeichenfolge an, die wie folgt beginnt: /
memory_limit	Menge des Speichers, der der PHP-Umgebung zugeordnet ist.	256M	–
zlib.output_compression	Gibt an, ob PHP die Komprimierung für die Ausgabe nutzen soll.	Off	On Off true false
allow_url_fopen	Gibt an, wenn die Dateifunktionen von PHP Daten aus Remote-Standorten abrufen können, wie Websites oder FTP-Server.	On	On Off true false

Name	Beschreibung	Standard	Zulässige Werte
display_errors	Gibt an, ob Fehlermeldungen Teil der Ausgabe sein sollen.	Off	On Off
max_execution_time	Gibt die maximale Zeitspanne (in Sekunden) an, die ein Skript ausgeführt werden kann, bevor es von der Umgebung beendet wird.	60	0 auf 9223372036854775807 (PHP_INT_MAX)
composer_options	Legt fest, welche benutzerdefinierten Optionen verwendet werden sollen, wenn Sie Abhängigkeiten mit Composer über "composer.phar install" installieren. Weitere Informationen sowie die verfügbaren Optionen finden Sie unter http://getcomposer.org/doc/03-cli.md#install .	-	-

Namespace: **aws:elasticbeanstalk:environment:proxy**

Name	Beschreibung	Standard	Zulässige Werte
ProxyServer	Legen Sie die Proxys fest, der für die Instances Ihrer Umgebung verwendet werden soll.	nginx	apache nginx

Note

Weitere Informationen zur PHP-Plattform finden Sie unter [Verwenden der Elastic Beanstalk-PHP-Plattform \(p. 288\)](#).

Python-Plattformoptionen

Namespace: **aws:elasticbeanstalk:application:environment**

Name	Beschreibung	Standard	Zulässige Werte
DJANGO_SETTINGS_MODULE	Gibt an, welche Einstellungsdatei verwendet werden soll.	-	-

Weitere Informationen finden Sie unter [Umgebungseigenschaften und andere Softwareeinstellungen \(p. 621\)](#).

Namespace: **aws:elasticbeanstalk:container:python**

Name	Beschreibung	Standard	Zulässige Werte
WSGIPath	Die Datei, die die WSGI-Anwendung enthält. Diese Datei muss über eine aufrufbare application verfügen.	Auf Amazon Linux 2 Python-Plattformversionen: application	-

Name	Beschreibung	Standard	Zulässige Werte
	Auf Amazon Linux AMI Python-Plattformversionen application.py		
NumProcesses	Die Anzahl der Daemon-Prozesse, die für die Prozessgruppe gestartet werden sollen, wenn WSGI-Anwendungen ausgeführt werden.	1	–
NumThreads	Die Anzahl der Threads, die erstellt werden sollen, um Anfragen in jedem Daemon-Prozess innerhalb der Prozessgruppe zu verarbeiten, wenn WSGI-Anwendungen ausgeführt werden.	15	–

Namespace: `aws:elasticbeanstalk:environment:proxy`

Name	Beschreibung	Standard	Zulässige Werte
ProxyServer	Legen Sie die Proxys fest, der für die Instances Ihrer Umgebung verwendet werden soll.	nginx	apache nginx

Amazon Linux AMI (vor Amazon Linux 2)-Plattformoptionen

Namespace: `aws:elasticbeanstalk:container:python:staticfiles`

Sie können den Proxy-Server mit dem folgenden Namespace konfigurieren, um statische Dateien bereitzustellen. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, statt die Anfrage an Ihre Anwendung weiterzuleiten. Dies reduziert die Anzahl der Anfragen, die Ihre Anwendung verarbeiten muss.

Bilden Sie einen von dem Proxy-Server bedienten Pfad auf einen Ordner in Ihrem Quellcode ab, der statische Komponenten enthält. Jede Option, die Sie in diesem Namespace definieren, weist einen anderen Pfad zu.

Standardmäßig stellt der Proxy-Server in einer Python-Umgebung alle Dateien im Ordner `static` unter dem Pfad `/static` bereit.

Namespace: `aws:elasticbeanstalk:container:python:staticfiles`

Name	Wert
Der Pfad, in dem der Proxy-Server die Dateien bereitstellt	Der Name des Verzeichnisses, das die Dateien enthält
Beispiel: Verwenden Sie <code>/images</code> , um Dateien unter <code>subdomain.elasticbeanstalk.com/images</code> bereitzustellen.	Beispiel: Verwenden Sie <code>staticimages</code> , um Dateien aus einem Verzeichnis namens <code>staticimages</code> auf oberster Ebene Ihres Quell-Bundles bereitzustellen.

Ruby-Plattformoptionen

Namespace: **aws:elasticbeanstalk:application:environment**

Name	Beschreibung	Standard	Zulässige Werte
RAILS_SKIP_MIGRATIONS	Gibt an, ob `rake db:migrate` im Namen der Anwendungen der Benutzer ausgeführt werden oder ob dies übersprungen werden soll. Dies gilt nur für Rails-3-Anwendungen.	false	true false
RAILS_SKIP_ASSET_COMPILATION	Gibt an, ob der Container `rake assets:precompile` im Namen der Anwendungen der Benutzer ausgeführt werden oder ob dies übersprungen werden soll. Dies gilt auch für Rails-3-Anwendungen.	false	true false
BUNDLE_WITHOUT	Eine durch Doppelpunkte (:) getrennte Liste mit Gruppen, die bei der Installation von Abhängigkeiten aus einer "Gemfile"-Datei ignoriert werden sollen.	test:development	
RACK_ENV	Gibt an, in welcher Umgebungsphase eine Anwendung ausgeführt werden kann. Beispiele für gängige Umgebungen sind Entwicklung, Produktion, Test.	production	-

Weitere Informationen finden Sie unter [Umgebungseigenschaften und andere Softwareeinstellungen \(p. 621\)](#).

Benutzerdefinierte Optionen

Verwenden Sie den `aws:elasticbeanstalk:customoption`-Namespace, um Optionen und Werte zu definieren, die in Resources-Blocks in anderen Konfigurationsdateien gelesen werden können. Verwenden Sie benutzerdefinierte Optionen zur Erfassung von benutzerspezifischen Einstellungen in einer einzelnen Konfigurationsdatei.

Vielleicht haben Sie eine komplexe Konfigurationsdatei, die eine Ressource definiert, die vom Benutzer konfiguriert werden kann, der die Umgebung startet. Wenn Sie mit `Fn::GetOptionSetting` den Wert für eine benutzerdefinierte Option abrufen, können Sie die Definition dieser Option in eine andere Konfigurationsdatei legen, wo sie einfacher entdeckt und vom Benutzer geändert werden kann.

Da es sich um Konfigurationsoptionen handelt, können benutzerdefinierte Optionen außerdem auf API-Ebene festgelegt werden, um Werte in einer Konfigurationsdatei zu überschreiben. Informationen finden Sie unter [Precedence \(p. 647\)](#).

Benutzerdefinierte Optionen sind genau wie alle anderen Optionen definiert:

```
option_settings:
  aws:elasticbeanstalk:customoption:
    option name: option value
```

Die folgende Konfigurationsdatei erstellt beispielsweise eine Option mit dem Namen `ELBAlarmEmail` und legt den Wert auf `someone@example.com` fest:

```
option_settings:  
  aws:elasticbeanstalk:customoption:  
    ELBAlarmEmail: someone@example.com
```

In anderen Fällen definiert eine Konfigurationsdatei ein SNS-Thema, das die Option mit Fn::GetOptionSetting liest, um den Wert des Endpoint-Attributs zu füllen:

```
Resources:  
  MySNSTopic:  
    Type: AWS::SNS::Topic  
    Properties:  
      Subscription:  
        - Endpoint:  
          Fn::GetOptionSetting:  
            OptionName: ELBAlarmEmail  
            DefaultValue: nobody@example.com  
          Protocol: email
```

Weitere Beispieldausschnitte mit Fn::GetOptionSetting finden Sie unter [Hinzufügen und Anpassen von Elastic Beanstalk-Umgebungsressourcen \(p. 745\)](#).

Erweiterte Umgebungsanpassung mit Konfigurationsdateien (.ebextensions)

Sie können AWS Elastic Beanstalk-Konfigurationsdateien (.ebextensions) in den Quellcode Ihrer Webanwendung einfügen, um die Umgebung zu konfigurieren und die darin enthaltenen AWS-Ressourcen anzupassen. Bei Konfigurationsdateien handelt es sich um Dokumente im YAML- oder JSON-Format mit der Dateierweiterung .config, die Sie in einem Ordner mit dem Namen .ebextensions speichern und im Quell-Bundle der Anwendung bereitstellen.

Example .ebextensions/network-load-balancer.config

In diesem Beispiel wird eine einfache Konfigurationsänderung vorgenommen. Es wird eine Konfigurationsoption geändert, um den Typ des Load Balancers Ihrer Umgebung auf „Network Load Balancer“ festzulegen.

```
option_settings:  
  aws:elasticbeanstalk:environment:  
    LoadBalancerType: network
```

Wir empfehlen die Verwendung von YAML für Konfigurationsdateien, da das Format besser lesbar ist als JSON. YAML unterstützt Kommentare, mehrzeilige Befehle, verschiedene Alternativen für Anführungszeichen und vieles mehr. Sie können jedoch alle Konfigurationsänderungen in Elastic Beanstalk-Konfigurationsdateien gleichermaßen mit YAML oder JSON vornehmen.

Tipp

Für die Entwicklung und die Testphase neuer Konfigurationsdateien starten Sie eine bereinigte Umgebung, auf der die Standardanwendung ausgeführt wird, und stellen diese dort bereit. Schlecht formatierte Konfigurationsdateien führen dazu, dass eine neue Umgebung nicht gestartet werden kann und nicht wiederherstellbar ist.

Im Abschnitt option_settings der Konfigurationsdatei werden die Werte für [Konfigurationsoptionen \(p. 646\)](#) definiert. Mithilfe von Konfigurationsoptionen können Sie die

Elastic Beanstalk-Umgebung, die darin enthaltenen AWS-Ressourcen sowie die Software für die Anwendungsausführung konfigurieren. Konfigurationsdateien sind nur eine von mehreren Möglichkeiten zum Festlegen von Konfigurationsoptionen.

Im Abschnitt [können Sie die Ressourcen in der Anwendungsumgebung weiter anpassen und zusätzliche AWS-Ressourcen konfigurieren. Dies geht über die Funktionalität der Konfigurationsoptionen hinaus. Resources \(p. 745\)](#) Sie können alle von AWS CloudFormation unterstützten Ressourcen, die Elastic Beanstalk zur Umgebungserstellung einsetzt, hinzufügen und konfigurieren.

In den anderen Abschnitten einer Konfigurationsdatei (`packages`, `sources`, `files`, `users`, `groups`, `commands`, `container_commands` und `services`) können Sie die EC2-Instances konfigurieren, die in der Umgebung gestartet werden. Jedes Mal, wenn ein Server in der Umgebung gestartet wird, führt Elastic Beanstalk die in diesen Abschnitten definierten Vorgänge aus, um das Betriebs- und das Speichersystem für die Anwendung vorzubereiten.

Beispiele für häufig verwendete `.ebextensions` finden Sie im [Elastic Beanstalk-Konfigurationsdateien-Repository](#).

Voraussetzungen

- Location (Standort) – Platzieren Sie Ihre gesamten Konfigurationsdateien in einem einzelnen Verzeichnis namens `.ebextensions` im Stammverzeichnis Ihres Quell-Bundles. Verzeichnisse, die mit einem Punkt beginnen, können von Dateibrowsern ausgeblendet werden. Stellen Sie daher sicher, dass das Verzeichnis beim Erstellen des Quell-Bundles angelegt wird. Anweisungen finden Sie unter [Erstellen des Quell-Bundles einer Anwendung. \(p. 412\)](#).
- Naming (Namensgebung) – Konfigurationsdateien müssen die Dateiendung `.config` haben.
- Formatting (Format) – Konfigurationsdateien müssen den YAML- oder JSON-Spezifikationen entsprechen.

Bei Verwendung von YAML immer Leerzeichen verwenden, um Schlüssel in den verschiedenen Verschachtelungsebenen einzurücken. Weitere Informationen zu YAML finden Sie unter [YAML Ain't Markup Language \(YAML™\) Version 1.1](#).

- Uniqueness (Eindeutigkeit) – Verwenden Sie jeden Schlüssel in jeder Konfigurationsdatei nur einmal.

Warnung

Wenn Sie einen Schlüssel (z. B. `option_settings`) innerhalb einer Konfigurationsdatei mehrfach verwenden, wird einer der Abschnitte gelöscht. Kombinieren Sie doppelte Abschnitte in einem Abschnitt oder platzieren Sie diese in separaten Konfigurationsdateien.

Der Prozess für die Bereitstellung kann je nach dem Client, der für die Umgebungsverwaltung verwendet wird, unterschiedlich sein. Ausführliche Informationen finden Sie in den folgenden Abschnitten:

- [Elastic Beanstalk-Konsole \(p. 654\)](#)
- [EB CLI \(p. 655\)](#)
- [AWS-CLI \(p. 657\)](#)

Themen

- [Optionseinstellungen \(p. 724\)](#)
- [Anpassen von Software auf Linux-Servern \(p. 725\)](#)
- [Anpassen von Software auf Windows-Servern \(p. 738\)](#)
- [Hinzufügen und Anpassen von Elastic Beanstalk-Umgebungsressourcen \(p. 745\)](#)

Optionseinstellungen

Mit dem `option_settings`-Schlüssel können Sie die Elastic Beanstalk-Konfiguration ändern und Variablen definieren, die von der Anwendung über Umgebungsvariablen abgerufen werden. Einige Namespaces bieten die Möglichkeit, die Anzahl der Parameter zu erhöhen und Parameternamen anzugeben. Eine Liste der Namespaces und Konfigurationsoptionen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Optionseinstellungen können während der Umgebungserstellung oder einer -aktualisierung auch direkt in eine Umgebung übernommen werden. Werden Einstellungen direkt für die Umgebung bereitgestellt, überschreibt dies die Einstellungen für dieselben Optionen in Konfigurationsdateien. Wenn Sie Einstellungen aus der Konfiguration einer Umgebung entfernen, werden die Einstellungen in Konfigurationsdateien wirksam. Details dazu finden Sie unter [Precedence \(p. 647\)](#).

Syntax

Die Standardsyntax für Optionseinstellungen besteht aus einem Array von Objekten, von denen jedes die Schlüssel `namespace`, `option_name` und `value` aufweist.

```
option_settings:  
  - namespace: namespace  
    option_name: option name  
    value: option value  
  - namespace: namespace  
    option_name: option name  
    value: option value
```

Der `namespace`-Schlüssel ist optional. Falls Sie keinen Namespace angeben, wird der Standardwert `aws:elasticbeanstalk:application:environment` verwendet:

```
option_settings:  
  - option_name: option name  
    value: option value  
  - option_name: option name  
    value: option value
```

Elastic Beanstalk unterstützt auch eine Syntax-Kurznotation für Optionseinstellungen, mit der Sie Optionen als Schlüssel-Wert-Paare unter dem Namespace angeben können:

```
option_settings:  
  namespace:  
    option_name: option value  
    option_name: option value
```

Beispiele

Im folgenden Beispiel wird eine spezifische Option für die Tomcat-Plattform im Namespace `aws:elasticbeanstalk:container:tomcat:jvmoptions` und in der Umgebungseigenschaft `MYPARAMETER` eingerichtet.

In YAML-Standardformat:

Example `.ebextensions/options.config`

```
option_settings:  
  - namespace: aws:elasticbeanstalk:container:tomcat:jvmoptions  
    option_name: Xmx
```

```
  value: 256m
- option_name: MYPARAMETER
  value: parametervalue
```

In der Kurznotation:

Example .ebextensions/options.config

```
option_settings:
  aws:elasticbeanstalk:container:tomcat:jvmoptions:
    Xmx: 256m
  aws:elasticbeanstalk:application:environment:
    MYPARAMETER: parametervalue
```

Im JSON-Format:

Example .ebextensions/options.config

```
{
  "option_settings": [
    {
      "namespace": "aws:elasticbeanstalk:container:tomcat:jvmoptions",
      "option_name": "Xmx",
      "value": "256m"
    },
    {
      "option_name": "MYPARAMETER",
      "value": "parametervalue"
    }
  ]
}
```

Anpassen von Software auf Linux-Servern

Möglicherweise muss die Software, von der Ihre Anwendung abhängig ist, angepasst und konfiguriert werden. Sie können Befehle hinzufügen, die während der Instance-Bereitstellung ausgeführt werden sollen, Linux-Benutzer und -Gruppen definieren und Dateien auf Ihre Umgebungs-Instances herunterladen oder dort direkt erstellen. Bei diesen Dateien kann es sich entweder um von der Anwendung benötigte Abhängigkeiten (z. B. zusätzliche Pakete aus dem yum-Repository) oder um Konfigurationsdateien handeln (wie z. B. ein Ersatz für eine Proxykonfigurationsdatei zum Überschreiben bestimmter Standardeinstellungen von Elastic Beanstalk).

In diesem Abschnitt wird beschrieben, welche Art von Informationen in einer Konfigurationsdatei aufgenommen werden können, die Sie verwenden können, um die Software auf Ihren EC2-Instances mit Linux zu konfigurieren. Allgemeine Informationen zum Anpassen und Konfigurieren der Elastic Beanstalk-Umgebungen finden Sie unter [Konfigurieren von Elastic Beanstalk-Umgebungen \(p. 526\)](#). Weitere Informationen zum Anpassen von Software auf Ihren EC2-Instances mit Windows finden Sie unter [Anpassen von Software auf Windows-Servern \(p. 738\)](#).

Hinweise

- Wir empfehlen dringend, auf Amazon Linux 2-Plattformen Buildfile zu verwenden, anstatt Dateien und Befehle in .ebextensions-Konfigurationsdateien bereitzustellen. Procfile und Plattform-Hooks möglichst zum Konfigurieren und Ausführen von benutzerdefiniertem Code auf Ihren Umgebungs-Instances während der Instance-Bereitstellung. Details zu diesen Mechanismen finden Sie unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#).
- Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und

stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Konfigurationsdateien unterstützen die folgenden Schlüssel, die sich auf die Linux-Server auswirken, auf denen die Anwendung ausgeführt wird.

Schlüssel

- [Pakete \(p. 726\)](#)
- [Gruppen \(p. 727\)](#)
- [Benutzer \(p. 728\)](#)
- [Quellen \(p. 729\)](#)
- [Datei- \(p. 729\)](#)
- [Befehle \(p. 731\)](#)
- [Services \(p. 733\)](#)
- [Container-Befehle \(p. 734\)](#)
- [Beispiel: Verwenden von benutzerdefinierten Amazon CloudWatch-Metriken \(p. 735\)](#)

Schlüssel werden in der hier aufgeführten Reihenfolge verarbeitet.

Beobachten Sie die [Ereignisse \(p. 864\)](#) in der Umgebung, während Sie Konfigurationsdateien entwickeln und testen. Elastic Beanstalk ignoriert eine Konfigurationsdatei, die Validierungsfehler enthält, z. B. einen ungültigen Schlüssel, und verarbeitet keinen der in der betreffenden Datei enthaltenen Schlüssel. Wenn dies geschieht, fügt Elastic Beanstalk eine Warnung in das Ereignisprotokoll ein.

Pakete

Mit dem Schlüssel `packages` können Sie vorgefertigte Anwendungen und Komponenten herunterladen und installieren.

Syntax

```
packages:  
  name of package manager:  
    package name: version  
    ...  
  name of package manager:  
    package name: version  
    ...  
  ...
```

Sie können unter jedem Paketmanager-Schlüssel mehrere Pakete angeben.

Unterstützte Paketformate

Elastic Beanstalk unterstützt derzeit die folgenden Paketmanager: yum, rubygems, python und rpm. Pakete werden in der folgenden Reihenfolge verarbeitet: rpm, yum und dann rubygems und python. Es gibt keine Reihenfolge zwischen rubygems und python. Innerhalb jedes Paketmanagers ist die Reihenfolge der Paketinstallation nicht garantiert. Verwenden Sie einen Paketmanager, der vom Betriebssystem unterstützt wird.

Note

Elastic Beanstalk unterstützt zwei zugrundeliegende Paketmanager für Python, pip und easy_install. In der Syntax der Konfigurationsdatei müssen Sie den Namen des Paketmanagers

als `python` angeben. Wenn Sie eine Konfigurationsdatei verwenden, um einen Python-Paketmanager anzugeben, verwendet Elastic Beanstalk Python 2.7. Wenn Ihre Anwendung sich auf eine andere Version von Python stützt, können Sie die zu installierenden Pakete in einer `requirements.txt`-Datei angeben. Weitere Informationen finden Sie unter [Angeben von Abhängigkeiten mithilfe einer Anforderungsdatei \(p. 360\)](#).

Angeben von Versionen

Innerhalb jedes Paketmanagers ist jedes Paket als ein Paketname und eine Liste der Versionen angegeben. Die Version kann eine Zeichenfolge, eine Liste von Versionen oder eine leere Zeichenfolge oder Liste sein. Eine leere Zeichenfolge oder Liste gibt an, dass Sie die neueste Version verwenden möchten. Bei RPM-Manager wird die Version als ein Pfad zu einer Datei auf einem Datenträger oder eine URL angegeben. Relative Pfade werden nicht unterstützt.

Wenn Sie eine Version eines Pakets angeben, versucht Elastic Beanstalk, diese Version zu installieren, auch wenn auf der Instance bereits eine neuere Version des Pakets installiert ist. Wenn eine neuere Version bereits installiert ist, schlägt die Bereitstellung fehl. Einige Paketmanager unterstützen mehrere Versionen, andere wiederum nicht. In der Dokumentation zu Ihrem Paketmanager finden Sie weitere Informationen. Wenn Sie keine Version angeben und bereits eine Version des Pakets installiert ist, installiert Elastic Beanstalk keine neue Version. Es wird angenommen, dass Sie die bestehende Version behalten und verwenden möchten.

Beispielausschnitt

Der folgende Codeausschnitt gibt eine Versions-URL für RPM an, fordert die neueste Version von yum und Version 0.10.2 von chef von rubygems an.

```
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'
```

Gruppen

Sie können den `groups`-Schlüssel verwenden, um Linux-/UNIX-Gruppen zu erstellen und Gruppen-IDs zuzuweisen. Um eine Gruppe zu erstellen, fügen Sie ein neues Schlüssel-Wert-Paar hinzu, das einer optionalen Gruppen-ID einen neuen Gruppennamen zuordnet. Der Gruppenschlüssel kann einen oder mehrere Gruppennamen enthalten. In der folgenden Tabelle werden die verfügbaren Schlüssel aufgeführt.

Syntax

```
groups:
  name of group: {}
  name of group:
    gid: "group id"
```

Optionen

`gid`

Eine Gruppen-ID-Nummer.

Wenn eine Gruppen-ID angegeben ist und die Gruppe nach Name bereits vorhanden ist, schlägt die Gruppenerstellung fehl. Wenn eine andere Gruppe die angegebene Gruppen-ID aufweist, lehnt das Betriebssystem die Gruppenerstellung möglicherweise ab.

Beispielausschnitt

Der folgende Codeausschnitt gibt eine Gruppe mit dem Namen groupOne an, ohne eine Gruppen-ID zuzuweisen, und eine Gruppe mit dem Namen groupTwo, die den Gruppen-ID-Wert 45 angegeben hat.

```
groups:  
  groupOne: {}  
  groupTwo:  
    gid: "45"
```

Benutzer

Sie können den `users`-Schlüssel verwenden, um auf der EC2-Instance Linux-/UNIX-Benutzer zu erstellen.

Syntax

```
users:  
  name of user:  
    groups:  
      - name of group  
    uid: "id of the user"  
    homeDir: "user's home directory"
```

Optionen

`uid`

Eine Benutzer-ID. Der Erstellungsprozess ist nicht erfolgreich, wenn der Benutzername mit einer anderen Benutzer-ID vorhanden ist. Wenn die Benutzer-ID bereits einem vorhandenen Benutzer zugewiesen ist, lehnt das Betriebssystem die Erstellungsanforderung möglicherweise ab.

`groups`

Eine Liste von Gruppennamen. Der Benutzer wird jeder Gruppe in der Liste hinzugefügt.

`homeDir`

Das Stammverzeichnis des Benutzers.

Benutzer werden als nicht interaktive Systembenutzer mit der Shell `/sbin/nologin` erstellt. Dies ist beabsichtigt und kann nicht geändert werden.

Beispielausschnitt

```
users:  
  myuser:  
    groups:  
      - group1  
      - group2  
    uid: "50"  
    homeDir: "/tmp"
```

Quellen

Mit dem Schlüssel `sources` können Sie eine Archivdatei über eine öffentliche URL herunterladen und diese auf der EC2-Instance in einem Zielverzeichnis entpacken.

Syntax

```
sources:  
  target directory: location of archive file
```

Unterstützte Formate

Unterstützte Formate sind tar, tar+gzip, tar+bz2 und zip. Sie können auf externe Speicherort wie Amazon Simple Storage Service (Amazon S3) verweisen (z. B. <https://mybucket.s3.amazonaws.com/myobject>), sofern die URL öffentlich zugänglich ist.

Beispielausschnitt

Im folgenden Beispiel wird eine öffentliche ZIP-Datei von einem Amazon S3-Bucket heruntergeladen und in `/etc/myapp` entpackt:

```
sources:  
  /etc/myapp: https://mybucket.s3.amazonaws.com/myobject
```

Note

Mehrere Extraktionen sollten nicht den gleichen Zielpfad wiederverwenden. Das Extrahieren einer anderen Quelle in denselben Zielpfad ersetzt den Inhalt, statt an den Inhalt angefügt zu werden.

Datei-

Sie können den `files`-Schlüssel verwenden, um Dateien auf der EC2-Instance zu erstellen. Der Inhalt kann entweder in die Konfigurationsdatei eingebettet sein oder von einer URL abgerufen werden. Die Dateien werden in lexikalischer Reihenfolge auf den Datenträger geschrieben.

Wenn private Dateien mit dem Schlüssel `files` von Amazon S3 heruntergeladen werden sollen, stellen Sie ein Instance-Profil für die Autorisierung bereit.

Wenn der von Ihnen angegebene Dateipfad bereits auf der Instance vorhanden ist, wird die vorhandene Datei beibehalten, wobei die Erweiterung `.bak` an ihren Namen angehängt wird.

Syntax

```
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:
```

Optionen

content

Zeichenfolgeninhalt zum Hinzufügen zur Datei. Geben Sie `content` oder `source` an, jedoch nicht beides.

source

URL einer Datei zum Herunterladen. Geben Sie `content` oder `source` an, jedoch nicht beides.

encoding

Das Codierungsformat der angegebene Zeichenfolge mit der `content`-Option.

Zulässige Werte: `plain` | `base64`

group

Linux-Gruppe, der die Datei gehört.

owner

Linux-Benutzer, dem die Datei gehört.

mode

Ein sechsstelliger Oktalwert, der für den Modus für diese Datei steht. Wird für Windows-Systeme nicht unterstützt. Verwenden Sie die ersten drei Ziffern für Symlinks und die letzten drei Ziffern für das Einrichten von Berechtigungen. Um einen Symlink zu erstellen, geben Sie `120xxx` an, wobei `xxx` die Berechtigungen der Zielfile definiert. Um Berechtigungen für eine Datei anzugeben, verwenden Sie die letzten drei Ziffern, z. B. `000644`.

authentication

Der Name der zu verwendenden [AWS CloudFormation-Authentifizierungsmethode](#). Mit dem Ressourcenschlüssel können Sie Authentifizierungsmethoden zu den Metadaten der Auto Scaling-Gruppe hinzufügen. Ein Beispiel finden Sie unten.

Beispielausschnitt

```
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
```

Beispiel mit einem Symlink. Dies erzeugt einen Link `/tmp/myfile2.txt`, der auf die Datei `/tmp/myfile1.txt` verweist.

```
files:
```

```
"/tmp/myfile2.txt" :  
  mode: "120400"  
  content: "/tmp/myfile1.txt"
```

In dem folgenden Beispiel wird der Ressourcenschlüssel zum Hinzufügen einer Authentifizierungsmethode mit dem Namen S3Auth und zum Herunterladen einer privaten Datei aus einem Amazon S3-Bucket verwendet:

```
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"
```

Befehle

Sie können den `commands`-Schlüssel verwenden, um Befehle für die EC2-Instance auszuführen. Die Befehle werden ausgeführt, bevor die Anwendung und der Webserver eingerichtet sind und die Anwendungsversionsdatei extrahiert wird.

Die angegebenen Befehle werden als Root-Benutzer ausgeführt und in alphabetischer Reihenfolge dem Namen nach verarbeitet. Befehle werden standardmäßig im Stammverzeichnis ausgeführt. Um die Befehle aus einem anderen Verzeichnis auszuführen, verwenden Sie die `cwd`-Option.

Um Probleme mit Befehlen zu beheben, schlagen Sie deren Ausgabe in den [Instance-Protokollen \(p. 869\)](#) nach.

Syntax

```
commands:  
  command name:  
    command: command to run  
    cwd: working directory  
    env:  
      variable name: variable value  
    test: conditions for command  
    ignoreErrors: true
```

Optionen

command

Ein Array ([Blocksequenzsammlung](#) in YAML-Syntax) oder eine Zeichenfolge, das bzw. die den auszuführenden Befehl angibt. Einige wichtige Hinweise:

- Wenn Sie eine Zeichenfolge verwenden, müssen Sie die Zeichenfolge nicht in Anführungszeichen einschließen. Wenn Sie Anführungszeichen verwenden, muss literal verwendeten Anführungszeichen desselben Typs ein Escape-Zeichen vorangestellt werden.
- Wenn Sie ein Array verwenden, müssen Sie Leerzeichen kein Escape-Zeichen voranstellen und Befehlsparameter nicht in Anführungszeichen angeben. Jedes Arrayelement ist ein einzelnes Befehlsargument. Verwenden Sie ein Array nicht, um mehrere Befehle anzugeben.

Die folgenden Beispiele sind gleichwertig:

```
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.
```

Wenn Sie mehrere Befehle angeben, verwenden Sie ein [literales Blockskalar](#), wie im folgenden Beispiel gezeigt.

```
commands:  
  command block:  
    command: |  
      git commit -m "This is a comment."  
      git push
```

env

(Optional) Legt Umgebungsvariablen für den Befehl fest. Diese Eigenschaft überschreibt die vorhandene Umgebung, anstatt sie anzuhängen.

cwd

(Optional) Das Arbeitsverzeichnis. Sofern nicht definiert, werden Befehle aus dem Stammverzeichnis (/) ausgeführt.

test

(Optional) Ein Befehl, der den Wert true (Beendigungscode 0) zurückgeben muss, damit Elastic Beanstalk den im Schlüssel command enthaltenen Befehl, wie ein Shell-Skript, verarbeitet.

ignoreErrors

(Optional) Ein boolescher Wert, der bestimmt, ob andere Befehle ausgeführt werden sollen, wenn der im Schlüssel command enthaltene Befehl fehlschlägt (gibt einen Wert ungleich null zurück). Legen Sie diesen Wert auf true fest, damit weiterhin Befehle ausgeführt werden, selbst wenn der Befehl fehlschlägt. Legen Sie den Wert auf false fest, um anzugeben, dass keine Befehle mehr ausgeführt werden sollen, wenn der Befehl fehlschlägt. Der Standardwert ist false.

Beispielausschnitt

Der folgende Beispielausschnitt führt ein Python-Skript aus.

```
commands:
```

```
python_install:  
  command: myscript.py  
  cwd: /home/ec2-user  
  env:  
    myvarname: myvarvalue  
  test: "[ -x /usr/bin/python ]"
```

Services

Mit dem Schlüssel `services` definieren Sie, welche Services beim Start der Instance gestartet oder gestoppt werden sollen. Zudem lassen sich mit dem Schlüssel `services` auch Abhängigkeiten von Quellen, Paketen und Dateien angeben, sodass Elastic Beanstalk im Falle eines Neustarts aufgrund von installierten Dateien den Service neu startet.

Syntax

```
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"
```

Optionen

ensureRunning

Legen Sie den Wert auf `true` fest, um sicherzustellen, dass der Service ausgeführt wird, nachdem Elastic Beanstalk abgeschlossen wurde.

Legen Sie den Wert auf `false` fest, um sicherzustellen, dass der Service nicht ausgeführt wird, nachdem Elastic Beanstalk abgeschlossen wurde.

Lassen Sie diesen Schlüssel aus, um keine Änderungen am Servicestatus vorzunehmen.

enabled

Legen Sie den Wert auf `true` fest, um sicherzustellen, dass der Service beim Systemstart automatisch gestartet wird.

Legen Sie den Wert auf `false` fest, um sicherzustellen, dass der Service beim Systemstart nicht automatisch gestartet wird.

Lassen Sie diesen Schlüssel aus, um keine Änderungen an dieser Eigenschaft vorzunehmen.

files

Eine Liste von Dateien. Wenn Elastic Beanstalk eine Datei direkt über den Dateienblock ändert, wird der Service neu gestartet.

sources

Eine Liste von Verzeichnissen. Wenn Elastic Beanstalk ein Archiv in eines dieser Verzeichnisse erweitert, wird der Service neu gestartet.

packages

Eine Zuordnung des Paket-Managers zu einer Liste von Paketnamen. Wenn Elastic Beanstalk eines dieser Pakete installiert oder aktualisiert, wird der Service neu gestartet.

commands

Eine Liste von Befehlsnamen. Wenn Elastic Beanstalk den angegebenen Befehl ausführt, wird der Service neu gestartet.

Beispielausschnitt

Es folgt ein Beispiel für einen Ausschnitt:

```
services:  
  sysvinit:  
    myservice:  
      enabled: true  
      ensureRunning: true
```

Container-Befehle

Mit dem Schlüssel `container_commands` können Sie Befehle ausführen, die sich auf den Anwendungsquellcode auswirken. Container-Befehle werden ausgeführt, nachdem die Anwendung und der Webserver eingerichtet sind und die Anwendungsversionsdatei extrahiert wurde, aber bevor die Anwendungsversion bereitgestellt wird. Nicht-Container-Befehle und andere Anpassungen werden vor der Extraktion des Anwendungsquellcodes ausgeführt.

Die angegebenen Befehle werden als Root-Benutzer ausgeführt und in alphabetischer Reihenfolge dem Namen nach verarbeitet. Container-Befehle werden aus dem Staging-Verzeichnis ausgeführt, aus dem der Quellcode vor der Bereitstellung auf dem Anwendungsserver extrahiert wird. Alle Änderungen, die Sie mithilfe eines Container-Befehls am Quellcode im Staging-Verzeichnis vornehmen, werden bei der Bereitstellung der Quelle am endgültigen Speicherort einbezogen.

Um Probleme mit Container-Befehlen zu beheben, schlagen Sie deren Ausgabe in den [Instance-Protokollen \(p. 869\)](#) nach.

Verwenden Sie den Befehl `leader_only`, wenn der Befehl nur auf einer einzelnen Instance ausgeführt werden soll, oder konfigurieren Sie `test` so, dass der Befehl nur ausgeführt wird, sofern der Testbefehl den Wert `true` ergibt. Die Ausführung von Container-Befehlen des Leader-only-Typs erfolgt nur während der Umgebungserstellung und -bereitstellung, wohingegen andere Befehle und Serveranpassungen bei jeder Instance-Bereitstellung oder -Aktualisierung ausgeführt werden. Container-Befehle des Leader-only-Typs werden bei Startkonfigurationsänderungen nicht ausgeführt, so z. B. eine Änderung der AMI-ID oder des Instance-Typs.

Syntax

```
container_commands:  
  name of container_command:  
    command: "command to run"  
    leader_only: true  
  name of container_command:  
    command: "command to run"
```

Optionen

command

Eine auszuführende Zeichenfolge bzw. ein Array von Zeichenfolgen.

env

(Optional) Legen Sie Umgebungsvariablen vor der Befehlsausführung fest, um alle vorhandenen Werte zu überschreiben.

cwd

(Optional) Das Arbeitsverzeichnis. Standardmäßig ist dies das Staging-Verzeichnis der nicht komprimierten Anwendung.

leader_only

(Optional) Führen Sie den Befehl nur auf einer einzelnen, von Elastic Beanstalk ausgewählten Instance aus. Container-Befehle des Leader-only-Typs werden vor anderen Container-Befehlen ausgeführt. Ein Befehl kann vom Typ Leader-only oder test sein, aber nicht beides (leader_only hat Vorrang).

test

(Optional) Führen Sie einen Testbefehl aus, der true zurückgeben muss, damit der Container-Befehl ausgeführt wird. Ein Befehl kann vom Typ Leader-only oder test sein, aber nicht beides (leader_only hat Vorrang).

ignoreErrors

(Optional) Bereitstellungen schlagen nicht fehl, sofern der Container-Befehl einen anderen Wert als 0 (Erfolg) zurückgibt. Legen Sie den Wert auf true fest, um dies zu aktivieren.

Beispielausschnitt

Es folgt ein Beispiel für einen Ausschnitt.

```
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"
```

Beispiel: Verwenden von benutzerdefinierten Amazon CloudWatch-Metriken

Amazon Cloudwatch ist ein Web-Service, mit dem Sie verschiedene Metriken überwachen, verwalten und veröffentlichen und Alarmaktionen anhand der Daten von Metriken konfigurieren können. Sie können benutzerdefinierte Metriken definieren, die Elastic Beanstalk per Push an Amazon CloudWatch überträgt. Sobald die benutzerdefinierten Metriken in Amazon CloudWatch enthalten sind, können Sie diese in der Amazon CloudWatch-Konsole anzeigen.

Anhand der Amazon CloudWatch-Überwachungsskripts für Linux wird veranschaulicht, wie benutzerdefinierte Amazon CloudWatch-Metriken erstellt und genutzt werden. Die Skripts umfassen ein voll funktionsfähiges Beispiel, das Speicher-, Swap- und Speicherplatz-Auslastungsmetriken für eine Amazon Elastic Compute Cloud (Amazon EC2) Linux-Instance meldet. Weitere Informationen zu den Amazon CloudWatch-Überwachungsskripts finden Sie unter [Amazon CloudWatch-Überwachungsskripts für Linux](#) im Amazon CloudWatch-Entwicklerhandbuch.

Note

Die [erweiterten Integritätsberichte \(p. 824\)](#) von Elastic Beanstalk bieten native Unterstützung für die Veröffentlichung zahlreicher Instance- und Umgebungsmetriken in CloudWatch. Details dazu finden Sie unter [Veröffentlichen von benutzerdefinierten Amazon CloudWatch-Metriken für eine Umgebung \(p. 849\)](#).

Themen

- [.Ebextensions-Konfigurationsdatei \(p. 736\)](#)
- [Berechtigungen \(p. 737\)](#)
- [Anzeigen von Metriken in der CloudWatch-Konsole \(p. 737\)](#)

.Ebextensions-Konfigurationsdatei

In diesem Beispiel werden Befehle und Optionseinstellungen in einer .ebextensions-Konfigurationsdatei verwendet, um die von Amazon CloudWatch bereitgestellten Überwachungsskripts herunterzuladen, zu installieren und auszuführen.

Damit Sie dieses Beispiel einsetzen können, speichern Sie es in einer Datei mit dem Namen cloudwatch.config in einem Verzeichnis mit dem Namen .ebextensions im Top-Level-Verzeichnis des Projekts. Dann stellen Sie die Anwendung über die Elastic Beanstalk-Konsole (binden Sie das Verzeichnis .ebextensions in das [Quell-Bundle \(p. 412\)](#) ein) oder die [EB CLI \(p. 1008\)](#) bereit.

Weitere Informationen zu Konfigurationsdateien finden Sie unter [Erweiterte Umgebungsanpassung mit Konfigurationsdateien \(.ebextensions\) \(p. 722\)](#).

.ebextensions/cloudwatch.config

```
packages:
  yum:
    perl-DateTime: []
    perl-Sys-Syslog: []
    perl-LWP-Protocol-https: []
    perl-Switch: []
    perl-URI: []
    perl-Bundle-LWP: []

sources:
  /opt/cloudwatch: https://aws-cloudwatch.s3.amazonaws.com/downloads/
CloudWatchMonitoringScripts-1.2.1.zip

container_commands:
  01-setupcron:
    command: |
      echo '* */5 * * * root perl /opt/cloudwatch/aws-scripts-mon/mon-put-instance-data.pl
`{"Fn::GetOptionSetting" : { "OptionName" : "CloudWatchMetrics", "DefaultValue" : "--mem-
util --disk-space-util --disk-path="/" }}` >> /var/log/cwpump.log 2>&1' > /etc/cron.d/cwpump
  02-changeperm:
    command: chmod 644 /etc/cron.d/cwpump
  03-changeperm:
    command: chmod u+x /opt/cloudwatch/aws-scripts-mon/mon-put-instance-data.pl

option_settings:
  "aws:autoscaling:launchconfiguration" :
    IamInstanceProfile : "aws-elasticbeanstalk-ec2-role"
  "aws:elasticbeanstalk:customoption" :
    CloudWatchMetrics : "--mem-util --mem-used --mem-avail --disk-space-util --disk-space-
used --disk-space-avail --disk-path=/ --auto-scaling"
```

Nachdem Sie die korrekte Funktionsweise der Konfigurationsdatei sichergestellt haben, können Sie die Befehlsumleitung von einer Protokolldatei (>> /var/log/cwpump.log 2>&1) in /dev/null (> /dev/null) ändern, um Speicherplatz zu sparen.

Berechtigungen

Um benutzerdefinierte Amazon CloudWatch-Metriken veröffentlichen zu können, benötigen die Instances der Umgebung die Berechtigung zur Verwendung von CloudWatch. Sie erteilen den Umgebungs-Instances die Berechtigungen, indem Sie diese zum [Instance-Profil \(p. 22\)](#) der Umgebung hinzufügen. Die Berechtigungen für das Instance-Profil können Sie vor oder nach der Anwendungsbereitstellung hinzufügen.

So erteilen Sie Berechtigungen zum Veröffentlichen von CloudWatch-Metriken

1. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich Roles aus.
3. Wählen Sie die Instance-Profilrolle der Umgebung aus. Wenn Sie eine Umgebung mit der Elastic Beanstalk-Konsole oder der [EB CLI \(p. 1008\)](#) erstellen, ist das standardmäßig aws-elasticbeanstalk-ec2-role.
4. Wählen Sie die Registerkarte Permissions aus.
5. Klicken Sie unter Inline Policies im Bereich Permissions auf Create Role Policy.
6. Wählen Sie Custom Policy und dann Select aus.
7. Füllen Sie die folgenden Felder aus und klicken Sie auf Apply Policy:

Richtlinienname

Der Name der Richtlinie.

Policy Document

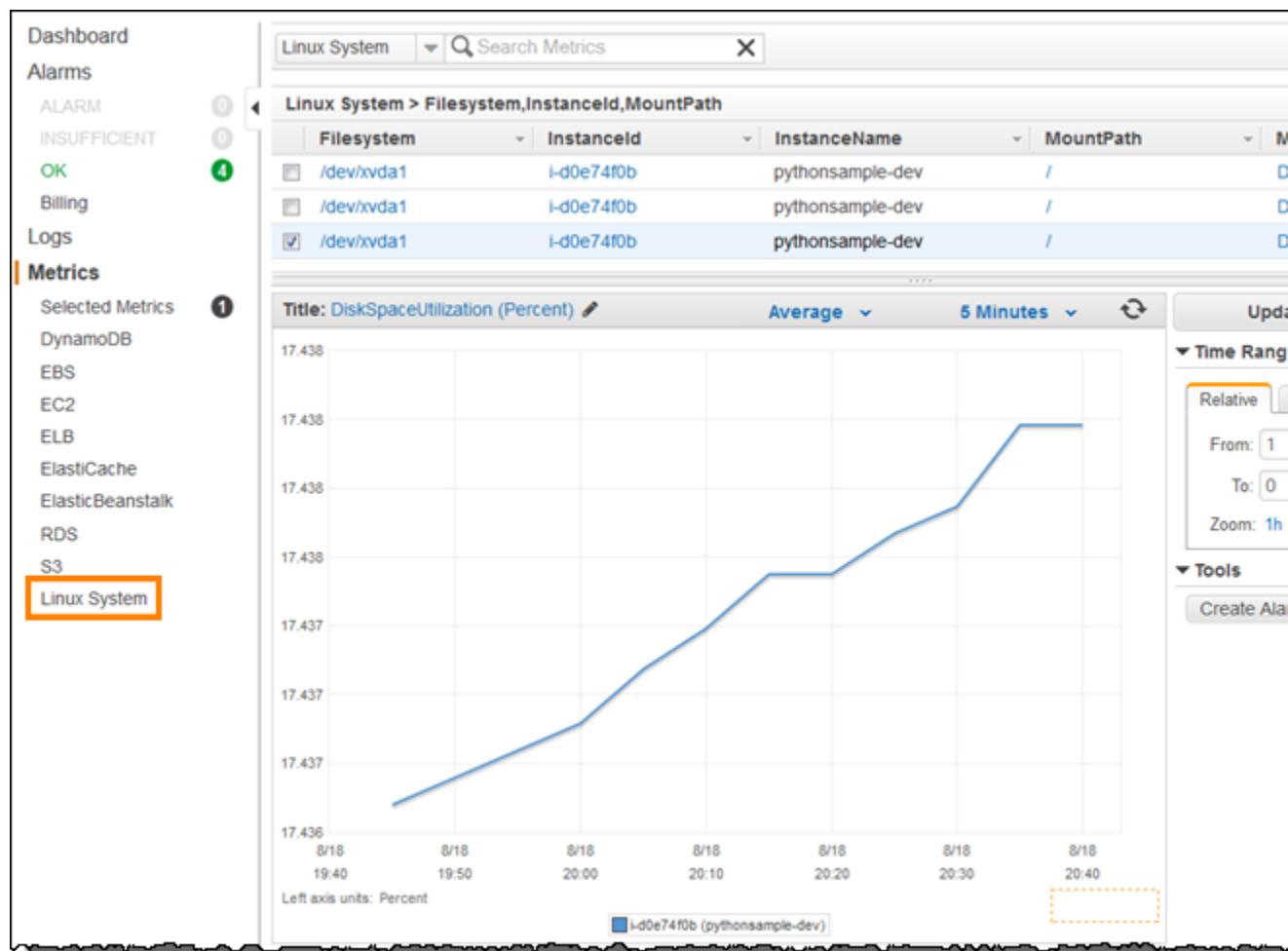
Kopieren Sie den folgenden Text und fügen Sie ihn in das Richtliniendokument ein:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "cloudwatch:PutMetricData",  
                "ec2:DescribeTags"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "*"  
            ]  
        }  
    ]  
}
```

Weitere Informationen zum Verwalten von Richtlinien finden Sie unter [Arbeiten mit Richtlinien](#) im IAM-Benutzerhandbuch.

Anzeigen von Metriken in der CloudWatch-Konsole

Nachdem Sie die CloudWatch-Konfigurationsdatei in der Umgebung bereitgestellt haben, können Sie die Metriken in der [Amazon CloudWatch-Konsole](#) anzeigen. Benutzerdefinierte Metriken weisen das Präfix Linux System auf.



Anpassen von Software auf Windows-Servern

Möglicherweise muss die Software, von der Ihre Anwendung abhängig ist, angepasst und konfiguriert werden. Bei diesen Dateien kann es sich um von der Anwendung benötigte Abhängigkeiten handeln, z. B. zusätzliche Pakete oder Services, die ausgeführt werden müssen. Allgemeine Informationen zum Anpassen und Konfigurieren der Elastic Beanstalk-Umgebungen finden Sie unter [Konfigurieren von Elastic Beanstalk-Umgebungen \(p. 526\)](#).

Note

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Konfigurationsdateien unterstützen die folgenden Schlüssel, die sich auf die Windows-Server auswirken, auf denen die Anwendung ausgeführt wird.

Schlüssel

- [Pakete \(p. 739\)](#)
- [Quellen \(p. 740\)](#)
- [Datei- \(p. 740\)](#)

- [Befehle \(p. 741\)](#)
- [Services \(p. 743\)](#)
- [Container-Befehle \(p. 744\)](#)

Schlüssel werden in der hier aufgeführten Reihenfolge verarbeitet.

Note

Ältere Versionen (ohne Versionsangabe) von .NET-Plattformversionen können die Konfigurationsdateien nicht in der richtigen Reihenfolge verarbeiten. Weitere Informationen finden Sie unter [Migration über die Hauptversionen der Elastic Beanstalk-Windows Server-Plattform hinweg \(p. 195\)](#).

Beobachten Sie die [Ereignisse \(p. 864\)](#) in der Umgebung, während Sie Konfigurationsdateien entwickeln und testen. Elastic Beanstalk ignoriert eine Konfigurationsdatei, die Validierungsfehler enthält, z. B. einen ungültigen Schlüssel, und verarbeitet keinen der in der betreffenden Datei enthaltenen Schlüssel. Wenn dies geschieht, fügt Elastic Beanstalk eine Warnung in das Ereignisprotokoll ein.

Pakete

Mit dem Schlüssel `packages` können Sie vorgefertigte Anwendungen und Komponenten herunterladen und installieren.

In Windows-Umgebungen unterstützt Elastic Beanstalk das Herunterladen und Installieren von MSI-Paketen. (Linux-Umgebungen unterstützen zusätzliche Paketmanager. Weitere Informationen finden Sie unter [Pakete \(p. 726\)](#) auf der Seite Anpassen von Software auf Linux-Servern.)

Sie können auf einen beliebigen externen Speicherort verweisen, z. B. ein Amazon Simple Storage Service (Amazon S3)-Objekt, sofern die URL öffentlich zugänglich ist.

Wenn Sie mehrere `msi:` Pakete angeben, wird deren Installationsreihenfolge nicht garantiert.

Syntax

Geben Sie einen Namen Ihrer Wahl als Paketnamen an, und eine URL zu einem MSI-Dateispeicherort als den Wert. Sie können mehrere Pakete unter dem `msi:-Schlüssel` angeben.

```
packages:  
  msi:  
    package name: package url  
    ...
```

Beispiele

Im folgenden Beispiel wird eine URL zum Herunterladen von mysql von <https://dev.mysql.com/> angegeben.

```
packages:  
  msi:  
    mysql: https://dev.mysql.com/get/Downloads/Connector-Net/mysql-connector-net-8.0.11.msi
```

Im folgenden Beispiel wird ein Amazon S3-Objekt als MSI-Dateispeicherort angegeben.

```
packages:  
  msi:  
    mymsi: https://mybucket.s3.amazonaws.com/myobject.msi
```

Quellen

Mit dem Schlüssel `sources` können Sie eine Archivdatei über eine öffentliche URL herunterladen und diese auf der EC2-Instance in einem Zielverzeichnis entpacken.

Syntax

```
sources:  
  target directory: location of archive file
```

Unterstützte Formate

In Windows-Umgebungen unterstützt Elastic Beanstalk das ZIP-Format. (Linux-Umgebungen unterstützen zusätzliche Formate. Weitere Informationen finden Sie unter [Quellen \(p. 729\)](#) auf der Seite Anpassen von Software auf Linux-Servern.)

Sie können auf einen beliebigen externen Speicherort verweisen, z. B. ein Amazon Simple Storage Service (Amazon S3)-Objekt, sofern die URL öffentlich zugänglich ist.

Beispiel

Im folgenden Beispiel wird eine öffentliche ZIP-Datei von einem Amazon S3-Bucket heruntergeladen und in `c:/myproject/myapp` entpackt.

```
sources:  
  "c:/myproject/myapp": https://mybucket.s3.amazonaws.com/myobject.zip
```

Datei-

Mit dem Schlüssel `files` können Sie Dateien auf der EC2-Instance erstellen. Die Inhalte können entweder in der Konfigurationsdatei enthalten sein oder über eine URL abgerufen werden. Die Dateien werden in lexikalischer Reihenfolge auf den Datenträger geschrieben. Wenn private Dateien von Amazon S3 heruntergeladen werden sollen, stellen Sie ein Instance-Profil für die Autorisierung bereit.

Syntax

```
files:  
  "target file location on disk":  
    source: URL  
    authentication: authentication name:  
  
  "target file location on disk":  
    content: |  
      this is my content  
    encoding: encoding format
```

Optionen

content

(Optional) Eine Zeichenfolge.

source

(Optional) Die URL, über die diese Datei geladen wird. Diese Option kann nicht mit dem Inhaltsschlüssel angegeben werden.

encoding

(Optional) Das Codierungsformat. Diese Option wird nur für einen bereitgestellten Inhaltsschlüsselwert verwendet. Der Standardwert ist `plain`.

Zulässige Werte: `plain | base64`

authentication

(Optional) Der Name der zu verwendenden [AWS CloudFormation-Authentifizierungsmethode](#). Mit dem Ressourcenschlüssel können Sie Authentifizierungsmethoden zu den Metadaten der Auto Scaling-Gruppe hinzufügen.

Beispiele

Das folgende Beispiel zeigt die beiden Möglichkeiten, Dateiinhalt bereitzustellen: von einer URL oder inline in der Konfigurationsdatei.

```
files:  
  "c:\\targetdirectory\\targetfile.txt":  
    source: http://foo.bar/myfile  
  
  "c:/targetdirectory/targetfile.txt":  
    content: |  
      # this is my file  
      # with content
```

Note

Falls im Dateipfad ein Backslash (\) verwendet wird, muss diesem – wie im vorigen Beispiel – ein weiterer Backslash (das Escape-Zeichen) vorangestellt werden.

In dem folgenden Beispiel wird der Ressourcenschlüssel zum Hinzufügen einer Authentifizierungsmethode mit dem Namen S3Auth und zum Herunterladen einer privaten Datei aus einem Amazon S3-Bucket verwendet:

```
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"
```

Befehle

Mit dem Schlüssel `commands` können Sie Befehle auf der EC2-Instance ausführen. Die Befehle werden in alphabetischer Reihenfolge nach Name verarbeitet und ausgeführt, bevor die Anwendung und der Webserver eingerichtet sind und die Anwendungsversionsdatei extrahiert wird.

Die angegebenen Befehle werden als Administrator-Benutzer ausgeführt.

Um Probleme mit Befehlen zu beheben, schlagen Sie deren Ausgabe in den [Instance-Protokollen \(p. 869\)](#) nach.

Syntax

```
commands:  
  command name:  
    command: command to run
```

Optionen

command

Entweder ein Array oder eine Zeichenfolge, das bzw. die den auszuführenden Befehl angibt.
Wenn Sie ein Array verwenden, müssen Sie Leerzeichen kein Escape-Zeichen voranstellen und Befehlsparameter nicht in Anführungszeichen angeben.

cwd

(Optional) Das Arbeitsverzeichnis. Standardmäßig sucht Elastic Beanstalk nach dem Verzeichnispfad des Projekts. Falls dieser nicht gefunden wird, gilt c:\Windows\System32 als Standardwert.

env

(Optional) Legt Umgebungsvariablen für den Befehl fest. Diese Eigenschaft überschreibt die vorhandene Umgebung, anstatt sie anzuhängen.

ignoreErrors

(Optional) Ein boolescher Wert, der bestimmt, ob andere Befehle ausgeführt werden sollen, wenn der im Schlüssel command enthaltene Befehl fehlschlägt (gibt einen Wert ungleich Null zurück). Legen Sie diesen Wert auf true fest, damit weiterhin Befehle ausgeführt werden, selbst wenn der Befehl fehlschlägt. Legen Sie den Wert auf false fest, um anzugeben, dass keine Befehle mehr ausgeführt werden sollen, wenn der Befehl fehlschlägt. Der Standardwert ist false.

test

(Optional) Ein Befehl, der den Wert true (Beendigungscode 0) zurückgeben muss, damit Elastic Beanstalk den im Schlüssel command enthaltenen Befehl verarbeitet.

waitForCompletion

(Optional) Wartezeit in Sekunden, nachdem ein Befehl abgeschlossen ist, bevor der nächste Befehl ausgeführt wird. Falls für das System nach der Ausführung des Befehls ein Neustart erforderlich ist, erfolgt dieser, nachdem die angegebene Sekundenanzahl verstrichen ist. Wenn das System aufgrund eines Befehls neu gestartet wird, führt Elastic Beanstalk die Wiederherstellung bis zum Punkt nach diesem Befehl in der Konfigurationsdatei aus. Die Standard-Wert ist 60 Sekunden. Sie können auch forever angeben, aber das System muss vor der Ausführung eines weiteren Befehls neu gestartet werden.

Beispiel

Im folgenden Beispiel wird die Ausgabe des Befehls set in der angegebenen Datei gespeichert. Wenn ein weiterer Befehl vorhanden ist, führt Elastic Beanstalk diesen unverzüglich nach Abschluss dieses Befehls aus. Falls dieser Befehl einen Neustart erforderlich macht, startet Elastic Beanstalk die Instance direkt nach Abschluss des Befehls neu.

```
commands:
```

```
test:  
  command: set > c:\\myapp\\set.txt  
  waitAfterCompletion: 0
```

Services

Mit dem Schlüssel `services` definieren Sie, welche Services beim Start der Instance gestartet oder gestoppt werden sollen. Zudem lassen sich mit dem Schlüssel `services` auch Abhängigkeiten von Quellen, Paketen und Dateien angeben, sodass Elastic Beanstalk im Falle eines Neustarts aufgrund von installierten Dateien den Service neu startet.

Syntax

```
services:  
  windows:  
    name of service:  
      files:  
        - "file name"  
      sources:  
        - "directory"  
      packages:  
        name of package manager:  
          "package name[: version]"  
      commands:  
        - "name of command"
```

Optionen

ensureRunning

(Optional) Legen Sie den Wert auf `true` fest, um sicherzustellen, dass der Service ausgeführt wird, nachdem Elastic Beanstalk abgeschlossen wurde.

Legen Sie den Wert auf `false` fest, um sicherzustellen, dass der Service nicht ausgeführt wird, nachdem Elastic Beanstalk abgeschlossen wurde.

Lassen Sie diesen Schlüssel aus, um keine Änderungen am Servicestatus vorzunehmen.

enabled

(Optional) Legen Sie den Wert auf `true` fest, um sicherzustellen, dass der Service beim Systemstart automatisch gestartet wird.

Legen Sie den Wert auf `false` fest, um sicherzustellen, dass der Service beim Systemstart nicht automatisch gestartet wird.

Lassen Sie diesen Schlüssel aus, um keine Änderungen an dieser Eigenschaft vorzunehmen.

files

Eine Liste von Dateien. Wenn Elastic Beanstalk eine Datei direkt über den Dateienblock ändert, wird der Service neu gestartet.

sources

Eine Liste von Verzeichnissen. Wenn Elastic Beanstalk ein Archiv in eines dieser Verzeichnisse erweitert, wird der Service neu gestartet.

packages

Eine Zuordnung des Paket-Managers zu einer Liste von Paketnamen. Wenn Elastic Beanstalk eines dieser Pakete installiert oder aktualisiert, wird der Service neu gestartet.

commands

Eine Liste von Befehlsnamen. Wenn Elastic Beanstalk den angegebenen Befehl ausführt, wird der Service neu gestartet.

Beispiel

```
services:  
  windows:  
    myservice:  
      enabled: true  
      ensureRunning: true
```

Container-Befehle

Mit dem Schlüssel `container_commands` können Sie Befehle ausführen, die sich auf den Anwendungsquellcode auswirken. Container-Befehle werden ausgeführt, nachdem die Anwendung und der Webserver eingerichtet sind und die Anwendungsversionsdatei extrahiert wurde, aber bevor die Anwendungsversion bereitgestellt wird. Nicht-Container-Befehle und andere Anpassungen werden vor der Extraktion des Anwendungsquellcodes ausgeführt.

Container-Befehle werden aus dem Staging-Verzeichnis ausgeführt, aus dem der Quellcode vor der Bereitstellung auf dem Anwendungsserver extrahiert wird. Alle Änderungen, die Sie mithilfe eines Container-Befehls am Quellcode im Staging-Verzeichnis vornehmen, werden bei der Bereitstellung der Quelle am endgültigen Speicherort einbezogen.

Um Probleme mit Container-Befehlen zu beheben, schlagen Sie deren Ausgabe in den [Instance-Protokollen \(p. 869\)](#) nach.

Verwenden Sie den Befehl `leader_only`, wenn der Befehl nur auf einer einzelnen Instance ausgeführt werden soll, oder konfigurieren Sie `test` so, dass der Befehl nur ausgeführt wird, sofern der Testbefehl den Wert `true` ergibt. Die Ausführung von Container-Befehlen des Leader-only-Typs erfolgt nur während der Umgebungserstellung und -bereitstellung, wohingegen andere Befehle und Serveranpassungen bei jeder Instance-Bereitstellung oder -Aktualisierung ausgeführt werden. Container-Befehle des Leader-only-Typs werden bei Startkonfigurationsänderungen nicht ausgeführt, so z. B. eine Änderung der AMI-ID oder des Instance-Typs.

Syntax

```
container_commands:  
  name of container_command:  
    command: command to run
```

Optionen

command

Eine auszuführende Zeichenfolge bzw. ein Array von Zeichenfolgen.

env

(Optional) Legen Sie Umgebungsvariablen vor der Befehlsausführung fest, um alle vorhandenen Werte zu überschreiben.

cwd

(Optional) Das Arbeitsverzeichnis. Standardmäßig ist dies das Staging-Verzeichnis der nicht komprimierten Anwendung.

leader_only

(Optional) Führen Sie den Befehl nur auf einer einzelnen, von Elastic Beanstalk ausgewählten Instance aus. Container-Befehle des Leader-only-Typs werden vor anderen Container-Befehlen ausgeführt. Ein Befehl kann vom Typ Leader-only oder test sein, aber nicht beides (leader_only hat Vorrang).

test

(Optional) Führen Sie einen Testbefehl aus, der true zurückgeben muss, damit der Container-Befehl ausgeführt wird. Ein Befehl kann vom Typ Leader-only oder test sein, aber nicht beides (leader_only hat Vorrang).

ignoreErrors

(Optional) Bereitstellungen schlagen nicht fehl, sofern der Container-Befehl einen anderen Wert als 0 (Erfolg) zurückgibt. Legen Sie den Wert auf true fest, um dies zu aktivieren.

waitAfterCompletion

(Optional) Wartezeit in Sekunden, nachdem ein Befehl abgeschlossen ist, bevor der nächste Befehl ausgeführt wird. Falls für das System nach der Ausführung des Befehls ein Neustart erforderlich ist, erfolgt dieser, nachdem die angegebene Sekundenanzahl verstrichen ist. Wenn das System aufgrund eines Befehls neu gestartet wird, führt Elastic Beanstalk die Wiederherstellung bis zum Punkt nach diesem Befehl in der Konfigurationsdatei aus. Die Standard-Wert ist **60** Sekunden. Sie können auch **forever** angeben, aber das System muss vor der Ausführung eines weiteren Befehls neu gestartet werden.

Beispiel

Im folgenden Beispiel wird die Ausgabe des Befehls set in der angegebenen Datei gespeichert. Elastic Beanstalk führt den Befehl auf einer Instance aus und startet diese direkt nach Abschluss des Befehls neu.

```
container_commands:  
  foo:  
    command: set > c:\\myapp\\set.txt  
    leader_only: true  
    waitAfterCompletion: 0
```

Hinzufügen und Anpassen von Elastic Beanstalk-Umgebungsressourcen

Sie können Ihre Umgebungsressourcen, die Teil Ihrer Elastic Beanstalk-Umgebung sind, auch anpassen. Beispielsweise können Sie eine Amazon SQS-Warteschlange und einen Alarm für die Warteschlange oder einen Amazon ElastiCache-Cluster hinzufügen. Indem Sie Ihrem Quell-Bundle eine Konfigurationsdatei beifügen, können Sie die Umgebung einfach gleichzeitig mit der Bereitstellung der Anwendungsversion anpassen.

Sie können den Resources-Schlüssel in einer [Konfigurationsdatei \(p. 722\)](#) zum Erstellen und Anpassen von AWS-Ressourcen in Ihrer Umgebung verwenden. Ressourcen, die in den Konfigurationsdateien definiert sind, werden zur AWS CloudFormation-Vorlage hinzugefügt, die zum Starten Ihrer Umgebung verwendet wird. Alle AWS CloudFormation-Ressourcentypen werden unterstützt.

Beispielsweise fügt die folgende Konfigurationsdatei einen Auto Scaling-Lebenszyklus-Hook zur Standard-Auto Scaling-Gruppe hinzu, die von Elastic Beanstalk erstellt wurde:

```
~/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": [
            "sns:SendMessage",
            "sns: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" ] }

```

In diesem Beispiel werden drei Ressourcen definiert, `hookrole`, `hooktopic` und `lifecyclehook`. Die ersten beiden Ressourcen sind eine IAM-Rolle, die Amazon EC2 Auto Scaling die Berechtigung zum Veröffentlichen von Nachrichten an Amazon SNS erteilt, sowie ein SNS-Thema, das Nachrichten aus der Auto Scaling-Gruppe an eine E-Mail-Adresse weiterleitet. Elastic Beanstalk erstellt diese Ressourcen mit den angegebenen Eigenschaften und Typen.

Die letzte Ressource, `lifecyclehook`, ist der Lebenszyklus-Hook selbst:

```

lifecyclehook:
  Type: AWS::AutoScaling::LifecycleHook
  Properties:
    AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
    LifecycleTransition: autoscaling:EC2_INSTANCE_TERMINATING
    NotificationTargetARN: { "Ref" : "hooktopic" }
    RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn" ] }

```

Die Lebenszyklus-Hook-Definition verwendet zwei [Funktionen \(p. 750\)](#), um die Werte für die Hook-Eigenschaften zu füllen. { "Ref" : "AWSEBAutoScalingGroup" } ruft den Namen der Auto Scaling-Gruppe ab, die von Elastic Beanstalk für die Umgebung erstellt wurde. AWSEBAutoScalingGroup ist einer der Standard-Ressourcennamen ([p. 747](#)), die von Elastic Beanstalk bereitgestellt wurden.

Für `AWS::IAM::Role` gibt `Ref` nur den Namen der Rolle, nicht die ARN, zurück. Wenn Sie den ARN für den `RoleARN`-Parameter abrufen möchten, verwenden Sie stattdessen eine andere intrinsische Funktion, `Fn::GetAtt`, mit der jedes Attribut aus einer Ressource abgerufen werden kann. `RoleARN: { "Fn::GetAtt" : ["hookrole", "Arn"] }` ruft das `Arn`-Attribut aus der `hookrole`-Ressource ab.

`{ "Ref" : "hooktopic" }` ruft den ARN des Amazon SNS-Themas ab, der zuvor in der Konfigurationsdatei erstellt wurde. Der von `Ref` zurückgegebene Wert variiert je nach Ressourcentyp und kann im AWS CloudFormation-Benutzerhandbuch unter dem [Thema für den AWS::SNS::Topic-Ressourcentyp](#) gefunden werden.

Ändern der Ressourcen, die Elastic Beanstalk für Ihre Umgebung erstellt

Die Ressourcen, die Elastic Beanstalk für Ihre Umgebung erstellt, haben Namen. Mithilfe dieser Namen können Sie über eine [Funktion \(p. 750\)](#) Informationen zu den Ressourcen abrufen oder die Eigenschaften der Ressourcen ändern, um deren Verhalten anzupassen.

Webserver-Umgebungen verfügen über die folgenden Ressourcen.

Webserver-Umgebungen

- `AWSEBAutoScalingGroup` ([AWS::AutoScaling::AutoScalingGroup](#)) – Die Auto Scaling-Gruppe, die Ihrer Umgebung zugeordnet ist.
- Eine der folgenden beiden Ressourcen.
 - `AWSEBAutoScalingLaunchConfiguration` ([AWS::AutoScaling::LaunchConfiguration](#)) – Die Startkonfiguration, die der Auto Scaling-Gruppe Ihrer Umgebung zugeordnet ist.
 - `AWSEBEC2LaunchTemplate` ([AWS::EC2::LaunchTemplate](#)) – Die von der Auto Scaling-Gruppe Ihrer Umgebung verwendete Amazon EC2-Startvorlage.

Note

Wenn in Ihrer Umgebung Funktionen verwendet werden, die Amazon EC2-Startvorlagen erfordern und Ihrer Benutzerrichtlinie die erforderlichen Berechtigungen fehlen, schlägt das Erstellen oder Aktualisieren der Umgebung möglicherweise fehl. Verwenden Sie die [verwaltete Benutzerrichtlinie \(p. 926\)](#) `AWSElasticBeanstalkFullAccess` oder fügen Sie der [benutzerdefinierten Richtlinie \(p. 929\)](#) die erforderlichen Berechtigungen hinzu.

- `AWSEBEnvironmentName` ([AWS::ElasticBeanstalk::Environment](#)) – Ihre Umgebung.
- `AWSEBSecurityGroup` ([AWS::EC2::SecurityGroup](#)) – Die Sicherheitsgruppe, die Ihrer Auto Scaling-Gruppe zugeordnet ist.
- `AWSEBRDSDatabase` ([AWS::RDS::DBInstance](#)) – Die Amazon RDS-DB-Instance, die Ihrer Umgebung zugeordnet ist (sofern anwendbar).

In einer Umgebung mit Load Balancer können Sie auf zusätzliche Ressourcen für den Load Balancer zugreifen. Classic Load Balancer verfügen über eine Ressource für den Load Balancer und über eine weitere Ressource für die ihm zugeordnete Sicherheitsgruppe. Anwendungs- und Netzwerk-Load Balancer verfügen über zusätzliche Ressourcen für den Standard-Listener, die Listener-Regel und die Zielgruppe des Load Balancers.

Umgebungen mit Load Balancing

- `AWSEBLoadBalancer` ([AWS::ElasticLoadBalancing::LoadBalancer](#)) – Der Classic Load Balancer Ihrer Umgebung.
- `AWSEBV2LoadBalancer` ([AWS::ElasticLoadBalancingV2::LoadBalancer](#)) – Der Application oder Network Load Balancer Ihrer Umgebung.

- **AWSEBLoadBalancerSecurityGroup** ([AWS::EC2::SecurityGroup](#)) – Nur in einer benutzerdefinierten **Amazon Virtual Private Cloud** (Amazon VPC) ist dies der Name der Sicherheitsgruppe, die Elastic Beanstalk für den Load Balancer erstellt. In einer Standard-VPC oder EC2 Classic weist Elastic Load Balancing dem Load Balancer eine Standard-Sicherheitsgruppe zu.
- **AWSEBV2LoadBalancerListener** ([AWS::ElasticLoadBalancingV2::Listener](#)) – Ein Listener, der dem Load Balancer erlaubt, auf Verbindungsanfragen zu prüfen und sie an eine oder mehrere Zielgruppen weiterzuleiten.
- **AWSEBV2LoadBalancerListenerRule** ([AWS::ElasticLoadBalancingV2::ListenerRule](#)) – Definiert, bei welchen Anfragen ein Elastic Load Balancing-Listener Aktionen ausführt und welche Aktion ausgeführt wird.
- **AWSEBV2LoadBalancerTargetGroup** ([AWS::ElasticLoadBalancingV2::TargetGroup](#)) – Eine Elastic Load Balancing-Zielgruppe, die Anfragen an ein oder mehrere registrierte Ziele wie Amazon EC2-Instances weiterleitet.

Worker-Umgebungen verfügen über Ressourcen für die SQS-Warteschlange, die eingehende Anfragen puffert, sowie eine Amazon DynamoDB-Tabelle, die von den Instances zur Leader-Wahl verwendet wird.

Worker-Umgebungen

- **AWSEBWorkerQueue** ([AWS::SQS::Queue](#)) – Die Amazon SQS-Warteschlange, aus der der Daemon Anfragen zieht, die verarbeitet werden müssen.
- **AWSEBWorkerDeadLetterQueue** ([AWS::SQS::Queue](#)) – Die Amazon SQS-Warteschlange, in der Nachrichten gespeichert werden, die nicht zugestellt werden können oder anderweitig nicht erfolgreich vom Daemon verarbeitet wurden.
- **AWSEBWorkerCronLeaderRegistry** ([AWS::DynamoDB::Table](#)) – Die Amazon DynamoDB-Tabelle, bei der es sich um die interne Registrierung handelt, die vom Daemon für regelmäßige Aufgaben verwendet wird.

Weitere Schlüssel für AWS CloudFormation-Vorlagen

Wir haben bereits Konfigurationsdateischlüssel aus AWS CloudFormation wie `Resources`, `files` und `packages` vorgestellt. Elastic Beanstalk fügt den Inhalt von Konfigurationsdateien in die AWS CloudFormation-Vorlage für Ihre Umgebung ein. So können Sie andere AWS CloudFormation-Abschnitte verwenden, um erweiterte Aufgaben in Ihren Konfigurationsdateien auszuführen.

Schlüssel

- [Parameter \(p. 748\)](#)
- [Outputs \(p. 749\)](#)
- [Mappings \(p. 749\)](#)

Parameter

Parameter sind eine Alternative zu den eigenen [benutzerdefinierten Optionen \(p. 721\)](#) von Elastic Beanstalk, die Sie zum Definieren von Werten verwenden können, die Sie an anderer Stelle in Ihren Konfigurationsdateien verwenden. Ebenso wie benutzerdefinierte Optionen können Sie Parameter verwenden, um vom Benutzer konfigurierbare Werte zentral zu erfassen. Anders als bei benutzerdefinierten Optionen können Sie die Elastic Beanstalk-API nicht verwenden, um Parameterwerte festzulegen. Die Anzahl der Parameter, die Sie in einer Vorlage definieren können, ist durch AWS CloudFormation begrenzt.

Ein Grund für die Verwendung von Parametern kann sein, dass Sie Ihre Konfigurationsdateien als AWS CloudFormation-Vorlagen verwenden möchten. Wenn Sie anstelle von benutzerdefinierten Optionen Parameter verwenden, können Sie die Konfigurationsdatei verwenden, um dieselbe Ressource in AWS

CloudFormation als eigenen Stack zu erstellen. Beispiel: Sie haben eine Konfigurationsdatei, die ein Amazon EFS-Dateisystem zu Testzwecken zur Umgebung hinzufügt. Dann verwenden Sie dieselbe Datei, um ein unabhängiges Dateisystem zu erstellen, das nicht in den Produktionslebenszyklus Ihrer Umgebung eingebunden ist.

Das folgende Beispiel zeigt die Verwendung von Parametern, um vom Benutzer konfigurierbare Werte zu Beginn einer Konfigurationsdatei zu erfassen.

Example [Loadbalancer-accesslogs-existingbucket.config](#) – Parameter

```
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

Verwenden Sie einen `Outputs`-Block, um Informationen zu den erstellten Ressourcen nach AWS CloudFormation zu exportieren. Verwenden Sie dann die Funktion `Fn::ImportValue`, um den Wert in eine AWS CloudFormation-Vorlage außerhalb von Elastic Beanstalk zu ziehen.

Im folgenden Beispiel wird ein Amazon SNS-Thema erstellt und dessen ARN mit dem Namen `NotificationTopicArn` nach AWS CloudFormation exportiert.

Example [sns-topic.config](#)

```
Resources:  
  NotificationTopic:  
    Type: AWS::SNS::Topic  
  
Outputs:  
  NotificationTopicArn:  
    Description: Notification topic ARN  
    Value: { "Ref" : "NotificationTopic" }  
    Export:  
      Name: NotificationTopicArn
```

Verwenden Sie in Konfigurationsdateien für andere Umgebungen oder AWS CloudFormation-Vorlagen außerhalb von Elastic Beanstalk die Funktion `Fn::ImportValue`, um den exportierten ARN abzurufen. In diesem Beispiel wird der exportierte Wert einer Umgebungseigenschaft namens `TOPIC_ARN` zugewiesen.

Example [env.config](#)

```
option_settings:  
  aws:elasticbeanstalk:application:environment:  
    TOPIC_ARN: `'{ "Fn::ImportValue" : "NotificationTopicArn" }`'
```

Mappings

Sie können eine Zuweisung verwenden, um Schlüssel-Wert-Paare nach Namespace sortiert zu speichern. Mit Zuweisungen können Sie Werte, die Sie in Ihren Konfigurationsdateien verwenden, strukturieren oder

einen Parameterwert abhängig von einem anderen Wert ändern. mit der folgenden Konfiguration wird beispielsweise der Wert eines Konto-ID-Parameters basierend auf der aktuellen Region festgelegt.

Example [Loadbalancer-accesslogs-newbucket.config](#) – Zuordnungen

```
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
```

Funktionen

Sie können Funktionen in den Konfigurationsdateien verwenden, um Werte der Ressourceneigenschaften mit Informationen aus anderen Ressourcen oder aus den Einstellungen der Konfigurationsoptionen von Elastic Beanstalk zu füllen. Elastic Beanstalk unterstützt AWS CloudFormation-Funktionen (`Ref`, `Fn::GetAtt`, `Fn::Join`) und eine Elastic Beanstalk-spezifische Funktion, `Fn::GetOptionSetting`.

Funktionen

- [Punkt \(p. 750\)](#)
- [Fn::GetAtt \(p. 751\)](#)
- [Fn::Join \(p. 751\)](#)
- [Fn::GetOptionSetting \(p. 751\)](#)

Punkt

Mit `Ref` wird die standardmäßige Zeichenfolgendarstellung einer AWS-Ressource abgerufen. Der von `Ref` zurückgegebene Wert hängt vom Ressourcentyp – und gelegentlich auch von weiteren Faktoren – ab. Beispielsweise gibt eine Sicherheitsgruppe ([AWS::EC2::SecurityGroup](#)) entweder den Namen oder die ID der Sicherheitsgruppe zurück, je nachdem, ob sie sich in einer Standard-Amazon Virtual Private Cloud (Amazon VPC), in EC2 Classic oder einer benutzerdefinierten VPC befindet.

```
{ "Ref" : "resource name" }
```

Note

Weitere Informationen zu den einzelnen Ressourcentypen einschließlich der Rückgabewerte von `Ref` finden Sie unter [AWS-Ressourcentypen – Referenz](#) im AWS CloudFormation-Benutzerhandbuch.

Aus dem Beispiel [Auto Scaling-Lebenszyklus-Hook \(p. 745\)](#):

```
Resources:  
  lifecyclehook:  
    Type: AWS::AutoScaling::LifecycleHook  
    Properties:  
      AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
```

Mit `Ref` lässt sich auch der AWS CloudFormation-Parameterwert abrufen, der entweder an anderer Stelle in derselben Datei oder in einer anderen Konfigurationsdatei definiert ist.

Fn::GetAtt

Mit `Fn::GetAtt` wird der Wert eines AWS-Ressourcenattributs abgerufen.

```
{ "Fn::GetAtt" : [ "resource name", "attribute name" ] }
```

Aus dem Beispiel [Auto Scaling-Lebenszyklus-Hook \(p. 745\)](#):

```
Resources:  
  lifecyclehook:  
    Type: AWS::AutoScaling::LifecycleHook  
    Properties:  
      RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn" ] }
```

Weitere Informationen finden Sie unter [Fn::GetAtt](#).

Fn::Join

Mit `Fn::Join` werden Zeichenfolgen mit einem Trennzeichen zusammengefasst. Die Zeichenfolgen können hartcodiert sein oder die Ausgabe von `Fn::GetAtt` oder `Ref` verwenden.

```
{ "Fn::Join" : [ "delimiter", [ "string1", "string2" ] ] }
```

Weitere Informationen finden Sie unter [Fn::Join](#).

Fn::GetOptionSetting

Mit `Fn::GetOptionSetting` wird der Einstellungswert einer [Konfigurationsoption \(p. 646\)](#) abgerufen, die für die Umgebung gilt.

```
"Fn::GetOptionSetting":  
  Namespace: "namespace"  
  OptionName: "option name"  
  DefaultValue: "default value"
```

Aus dem Beispiel [storing private keys \(p. 814\)](#):

```
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"
```

Beispiele für benutzerdefinierte Ressourcen

Nachstehend finden Sie eine Liste mit Beispielen für Konfigurationsdateien, die Sie verwenden können, um Ihre Elastic Beanstalk-Umgebungen anzupassen:

- [DynamoDB, CloudWatch, and SNS](#)
- [Elastic Load Balancing and CloudWatch](#)
- [ElastiCache](#)
- [RDS and CloudWatch](#)
- [SQS, SNS, and CloudWatch](#)

Unterthemen dieser Seite enthalten einige erweiterte Beispiele für das Hinzufügen und Konfigurieren benutzerdefinierter Ressourcen in einer Elastic Beanstalk-Umgebung.

Beispiele

- [Beispiel: ElastiCache \(p. 752\)](#)
- [Beispiel: SQS, CloudWatch und SNS \(p. 758\)](#)
- [Beispiel: DynamoDB, CloudWatch und SNS \(p. 760\)](#)

Beispiel: ElastiCache

Die folgenden Beispiele fügen einen Amazon ElastiCache-Cluster zu EC2-Classic- und EC2-VPC-Plattformen (Standard- und benutzerdefinierte [Amazon Virtual Private Cloud](#) (Amazon VPC)) hinzu. Weitere Informationen zu diesen Plattformen und darüber, wie Sie bestimmen, welche EC2 für Ihre Region und Ihr AWS-Konto unterstützt, finden Sie unter <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-supported-platforms.html>. Sehen Sie dann im Abschnitt in diesem Thema nach, der für Ihre Plattform gilt.

- [EC2-Classic-Plattformen \(p. 752\)](#)
- [EC2-VPC \(Standard\) \(p. 754\)](#)
- [EC2-VPC \(Benutzerdefiniert\) \(p. 756\)](#)

EC2-Classic-Plattformen

In diesem Beispiel wird ein Amazon ElastiCache-Cluster zu einer Umgebung mit Instances hinzugefügt, die in der EC2-Classic-Plattform gestartet werden. Alle Eigenschaften, die in diesem Beispiel aufgeführt sind, sind die mindestens erforderlichen Eigenschaften, die für jeden Ressourcentyp eingerichtet werden müssen. Sie können das Beispiel unter [ElastiCache-Beispiel](#) herunterladen.

Note

Mit diesem Beispiel werden AWS-Ressourcen erstellt, für die möglicherweise Kosten anfallen. Für weitere Informationen zu AWS-Preisen siehe <http://aws.amazon.com/pricing/>. Einige Services sind im Umfang des kostenlosen Nutzungskontingents für AWS enthalten. Als Neukunde können Sie diese Services kostenlos testen. Weitere Informationen finden Sie unter <http://aws.amazon.com/free/>.

In diesem Beispiel gehen Sie wie folgt vor:

1. Erstellen Sie ein [.ebextensions \(p. 722\)](#)-Verzeichnis im Verzeichnis der obersten Ebene Ihres Quell-Bundles.
2. Erstellen Sie zwei Konfigurationsdateien mit der Erweiterung `.config` und speichern Sie sie in Ihrem Verzeichnis `.ebextensions`. Eine Konfigurationsdatei definiert die Ressourcen und die andere die Optionen.
3. Stellen Sie Ihre Anwendung in Elastic Beanstalk bereit.

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Erstellen Sie eine Konfigurationsdatei (z. B. `elasticache.config`), in der die Ressourcen definiert werden. In diesem Beispiel erstellen wir den ElastiCache-Cluster, indem Sie den Namen der ElastiCache-Cluster-Ressource (`MyElasticCache`) angeben, den Typ deklarieren und dann die Eigenschaften für den Cluster konfigurieren. Im Beispiel wird auf den Namen der ElastiCache-Sicherheitsgruppenressource verwiesen, die in dieser Konfigurationsdatei erstellt und definiert wird. Als Nächstes erstellen Sie eine ElastiCache-Sicherheitsgruppe. Wir definieren den Namen für diese Ressource, deklarieren ihren Typ und fügen eine Beschreibung für die Sicherheitsgruppe hinzu. Schließlich legen wir Ingress-Regeln für die ElastiCache-Gruppe fest, um den Zugriff nur von Instances innerhalb der ElastiCache-Sicherheitsgruppe (`MyCacheSecurityGroup`) und der Elastic Beanstalk-Sicherheitsgruppe (`AWSEBSecurityGroup`) zu erlauben. Der Parametername `AWSEBSecurityGroup` ist ein fester Ressourcennamen von Elastic Beanstalk. Sie müssen `AWSEBSecurityGroup` zur Ingress-Regel Ihrer ElastiCache-Sicherheitsgruppe hinzufügen, damit Ihre Elastic Beanstalk-Anwendung eine Verbindung zu den Instances in Ihrem ElastiCache-Cluster herstellen kann.

```
#This sample requires you to create a separate configuration file that defines the custom option settings for CacheCluster properties.

Resources:
  MyElasticCache:
    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
```

Weitere Informationen zu den Ressourcen, die in dieser Beispielkonfigurationsdatei verwendet wurden, finden Sie unter folgenden Referenzen:

- AWS::ElastiCache::CacheCluster
- AWS::ElastiCache::SecurityGroup
- AWS::ElastiCache::SecurityGroupIngress

Erstellen Sie eine separate Konfigurationsdatei mit dem Namen `options.config` und definieren Sie benutzerdefinierte Optionseinstellungen.

```
option_settings:  
  "aws:elasticbeanstalk:customoption":  
    CacheNodeType : cache.m1.small  
    NumCacheNodes : 1  
    Engine : memcached
```

Mit diesen Zeilen wird Elastic Beanstalk angewiesen, die Werte für die Eigenschaften CacheNodeType, NumCacheNodes und Engine aus den Werten CacheNodeType, NumCacheNodes und Engine einer Konfigurationsdatei (in diesem Beispiel „`options.config`“) abzurufen. Diese enthält im Bereich „`option_settings`“ den Abschnitt `aws:elasticbeanstalk:customoption` mit einem Name-Wert-Paar, in dem sich der zu verwendende Wert befindet. Im obigen Beispiel werden folglich die Werte "cache.m1.small", "1" und "memcached" verwendet. Weitere Informationen zu `Fn::GetOptionSetting` finden Sie unter [Funktionen \(p. 750\)](#).

EC2-VPC (Standard)

In diesem Beispiel wird ein Amazon ElastiCache-Cluster zu einer Umgebung mit Instances hinzugefügt, die in der EC2-VPC-Plattform gestartet werden. Dabei gelten vor allem die Informationen in diesem Abschnitt gilt für ein Szenario, in dem EC2 Instances in der Standard-VPC startet. Alle Eigenschaften in diesem Beispiel sind die mindestens erforderlichen Eigenschaften, die für jeden Ressourcentyp eingerichtet werden müssen. Weitere Informationen zu Standard-VPCs finden Sie unter [Ihre Standard-VPC und -Subnetze](#).

Note

Mit diesem Beispiel werden AWS-Ressourcen erstellt, für die möglicherweise Kosten anfallen. Für weitere Informationen zu AWS-Preisen siehe <http://aws.amazon.com/pricing/>. Einige Services sind im Umfang des kostenlosen Nutzungskontingents für AWS enthalten. Als Neukunde können Sie diese Services kostenlos testen. Weitere Informationen finden Sie unter <http://aws.amazon.com/free/>.

In diesem Beispiel gehen Sie wie folgt vor:

1. Erstellen Sie ein [.ebextensions \(p. 722\)](#)-Verzeichnis im Verzeichnis der obersten Ebene Ihres Quell-Bundles.
2. Erstellen Sie zwei Konfigurationsdateien mit der Erweiterung `.config` und speichern Sie sie in Ihrem Verzeichnis `.ebextensions`. Eine Konfigurationsdatei definiert die Ressourcen und die andere die Optionen.
3. Stellen Sie Ihre Anwendung in Elastic Beanstalk bereit.

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Benennen Sie nun die Ressourcenkonfigurationsdatei `elasticache.config`. Zum Erstellen eines ElastiCache-Clusters geben wir in diesem Beispiel den Namen der ElastiCache-Cluster-Ressource (`MyElasticCache`) an, deklarieren den Typ und konfigurieren dann die Eigenschaften für den Cluster. Im Beispiel wird auf die ID der Sicherheitsgruppenressource verwiesen, die wir in dieser Konfigurationsdatei erstellt und definiert haben.

Als Nächstes erstellen Sie eine EC2-Sicherheitsgruppe. Wir definieren den Namen für diese Ressource, deklarieren ihren Typ, fügen eine Beschreibung hinzu und legen Ingress-Regeln für die Sicherheitsgruppe fest, um Zugriff nur von Instances innerhalb der Elastic Beanstalk-Sicherheitsgruppe (`AWSEBSecurityGroup`) zu erlauben. (Der Parametername `AWSEBSecurityGroup` ist ein fester Ressourcennamen von Elastic Beanstalk. Sie müssen `AWSEBSecurityGroup` zur Ingress-Regel Ihrer ElastiCache-Sicherheitsgruppe hinzufügen, damit Ihre Elastic Beanstalk-Anwendung eine Verbindung zu den Instances in Ihrem ElastiCache-Cluster herstellen kann.)

Die Ingress-Regeln für die EC2-Sicherheitsgruppe definiert auch das IP-Protokoll und die Port-Nummern, auf denen Cache-Knoten Verbindung akzeptieren können. Für Redis ist die Standard-Portnummer 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"
  MyElasticCache:
    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 : "MyElasticCache"
```

Weitere Informationen zu den Ressourcen, die in dieser Beispielkonfigurationsdatei verwendet wurden, finden Sie unter folgenden Referenzen:

- [AWS::ElastiCache::CacheCluster](#)
- [AWS::EC2::SecurityGroup](#)

Geben Sie als Nächstes der Optionen-Konfigurationsdatei den Namen `options.config` und definieren Sie benutzerdefinierte Optionseinstellungen.

```
option_settings:  
  "aws:elasticbeanstalk:customoption":  
    CacheNodeType : cache.t2.micro  
    NumCacheNodes : 1  
    Engine : redis  
    CachePort : 6379
```

Diese Zeilen informieren Elastic Beanstalk, die Werte für die Eigenschaften `CacheNodeType`, `NumCacheNodes`, `Engine` und `CachePort` aus den Werten `CacheNodeType`, `NumCacheNodes`, `Engine` und `CachePort` in einer Konfigurationsdatei (in diesem Beispiel `options.config`) abzurufen. Diese Datei enthält einen `aws:elasticbeanstalk:customoption`-Abschnitt (unter `option_settings`) mit Namen-Wert-Paaren mit den tatsächlichen zu verwendenden Werten. Im vorherigen Beispiel würden `cache.t2.micro`, `1`, `redis` und `6379` für diese Werte verwendet. Weitere Informationen zu `Fn::GetOptionSetting` finden Sie unter [Funktionen \(p. 750\)](#).

EC2-VPC (Benutzerdefiniert)

Wenn Sie eine benutzerdefinierte VPC auf der EC2-VPC-Plattform erstellen und sie als die VPC angeben, in die EC2 Instances startet, unterscheidet sich der Vorgang des Hinzufügens eines Amazon ElastiCache-Clusters zu Ihrer Umgebung von dem einer Standard-VPC. Der wesentliche Unterschied besteht darin, dass Sie eine Subnetzgruppe für den ElastiCache-Cluster erstellen müssen. Alle Eigenschaften in diesem Beispiel sind die mindestens erforderlichen Eigenschaften, die für jeden Ressourcentyp eingerichtet werden müssen.

Note

Mit diesem Beispiel werden AWS-Ressourcen erstellt, für die möglicherweise Kosten anfallen. Für weitere Informationen zu AWS-Preisen siehe <http://aws.amazon.com/pricing/>. Einige Services sind im Umfang des kostenlosen Nutzungskontingents für AWS enthalten. Als Neukunde können Sie diese Services kostenlos testen. Weitere Informationen finden Sie unter <http://aws.amazon.com/free/>.

In diesem Beispiel gehen Sie wie folgt vor:

1. Erstellen Sie ein [.ebextensions \(p. 722\)](#)-Verzeichnis im Verzeichnis der obersten Ebene Ihres Quell-Bundles.
2. Erstellen Sie zwei Konfigurationsdateien mit der Erweiterung `.config` und speichern Sie sie in Ihrem Verzeichnis `.ebextensions`. Eine Konfigurationsdatei definiert die Ressourcen und die andere die Optionen.
3. Stellen Sie Ihre Anwendung in Elastic Beanstalk bereit.

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Benennen Sie nun die Ressourcenkonfigurationsdatei `elasticache.config`. Zum Erstellen eines ElastiCache-Clusters geben wir in diesem Beispiel den Namen der ElastiCache-Cluster-Ressource (`MyElasticCache`) an, deklarieren den Typ und konfigurieren dann die Eigenschaften für den Cluster. Die Eigenschaften im Beispiel verweisen auf den Namen der Subnetzgruppe für den ElastiCache-Cluster und die ID der Sicherheitsgruppenressource, die wir in dieser Konfigurationsdatei erstellen und definieren.

Als Nächstes erstellen Sie eine EC2-Sicherheitsgruppe. Wir definieren den Namen für diese Ressource, deklarieren ihren Typ, fügen eine Beschreibung und die VPC-ID hinzu und legen Ingress-Regeln für die Sicherheitsgruppe fest, um Zugriff nur von Instances innerhalb der Elastic Beanstalk-Sicherheitsgruppe (`AWSEBSecurityGroup`) zu erlauben. (Der Parametername `AWSEBSecurityGroup` ist ein fester Ressourcennamen von Elastic Beanstalk. Sie müssen `AWSEBSecurityGroup` zur Ingress-Regel Ihrer

ElastiCache-Sicherheitsgruppe hinzufügen, damit Ihre Elastic Beanstalk-Anwendung eine Verbindung zu den Instances in Ihrem ElastiCache-Cluster herstellen kann.)

Die Ingress-Regeln für die EC2-Sicherheitsgruppe definiert auch das IP-Protokoll und die Port-Nummern, auf denen Cache-Knoten Verbindung akzeptieren können. Für Redis ist die Standard-Portnummer 6379. Schließlich erstellt dieses Beispiel eine Subnetzgruppe für den ElastiCache-Cluster. Wir definieren den Namen für diese Ressource, deklarieren ihren Typ und fügen eine Beschreibung und ID des Subnetzes in der Subnetzgruppe hinzu.

Note

Wir empfehlen, dass Sie private Subnetze für den ElastiCache-Cluster verwenden.

Weitere Informationen zu einer VPC mit einem privaten Subnetz finden Sie unter 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:
  MyElasticCache:
    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"
```

Weitere Informationen zu den Ressourcen, die in dieser Beispielkonfigurationsdatei verwendet wurden, finden Sie unter folgenden Referenzen:

- [AWS::ElastiCache::CacheCluster](#)
- [AWS::EC2::SecurityGroup](#)
- [AWS::ElastiCache::SubnetGroup](#)

Geben Sie als Nächstes der Optionen-Konfigurationsdatei den Namen `options.config` und definieren Sie benutzerdefinierte Optionseinstellungen.

Note

Im folgenden Beispiel ersetzen Sie die Beispiel-`CacheSubnets`- und `-VpcId`-Werte durch Ihre eigenen Subnetze und die 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
```

Diese Zeilen informieren Elastic Beanstalk, die Werte für die Eigenschaften `CacheNodeType`, `NumCacheNodes`, `Engine`, `CachePort`, `CacheSubnets` und `VpcId` aus den Werten `CacheNodeType`, `NumCacheNodes`, `Engine`, `CachePort`, `CacheSubnets` und `VpcId` in einer Konfigurationsdatei (in diesem Beispiel `options.config`) abzurufen. Diese Datei enthält einen `aws:elasticbeanstalk:customoption`-Abschnitt (unter `option_settings`) mit Namen-Wert-Paaren mit den Beispielwerten. Im vorherigen Beispiel würden `cache.t2.micro`, `1`, `redis`, `6379`, `subnet-1a1a1a1a`, `subnet-2b2b2b2b`, `subnet-3c3c3c3c` und `vpc-4d4d4d4d` für diese Werte verwendet. Mehr über `Fn::GetOptionSetting` erfahren Sie unter [Funktionen \(p. 750\)](#).

Beispiel: SQS, CloudWatch und SNS

In diesem Beispiel wird der Umgebung eine Amazon SQS-Warteschlange sowie ein Alarm für die Warteschlange hinzugefügt. In diesem Beispiel werden die mindestens erforderlichen Eigenschaften verwendet, die für jede Ressource eingerichtet werden müssen. Sie können das Beispiel unter [SQS, SNS und CloudWatch](#) herunterladen.

Note

Mit diesem Beispiel werden AWS-Ressourcen erstellt, für die möglicherweise Kosten anfallen. Für weitere Informationen zu AWS-Preisen siehe <http://aws.amazon.com/pricing/>. Einige Services sind im Umfang des kostenlosen Nutzungskontingents für AWS enthalten. Als Neukunde können Sie diese Services kostenlos testen. Weitere Informationen finden Sie unter <http://aws.amazon.com/free/>.

In diesem Beispiel gehen Sie wie folgt vor:

1. Erstellen Sie ein [.ebextensions \(p. 722\)](#)-Verzeichnis im Verzeichnis der obersten Ebene Ihres Quell-Bundles.

2. Erstellen Sie zwei Konfigurationsdateien mit der Erweiterung `.config` und speichern Sie sie in Ihrem Verzeichnis `.ebextensions`. Eine Konfigurationsdatei definiert die Ressourcen und die andere die Optionen.
3. Stellen Sie Ihre Anwendung in Elastic Beanstalk bereit.

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Erstellen Sie eine Konfigurationsdatei (z. B. `sqs.config`), in der die Ressourcen definiert werden. In diesem Beispiel wird eine SQS-Warteschlange erstellt und die `VisibilityTimeout`-Eigenschaft wird in der `MySQSQueue`-Ressource definiert. Anschließend wird ein SNS-Topic erstellt und angegeben, dass die E-Mail im Falle eines ausgelösten Alarms an `someone@example.com` gesendet wird. Abschließend wird ein CloudWatch-Alarm erstellt, falls die Warteschlange mehr als zehn Nachrichten umfasst. In der Eigenschaft `Dimensions` werden der Name der Dimension und der Wert, der die Maßeinheit der Dimension darstellt, angegeben. Hier wird `Fn::GetAtt` für die Rückgabe des Werts `QueueName` von `MySQSQueue` verwendet.

```
#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" ] }
```

Weitere Informationen zu den Ressourcen, die in dieser Beispielkonfigurationsdatei verwendet wurden, finden Sie unter folgenden Referenzen:

- [AWS::SQS::Queue](#)
- [AWS::SNS::Topic](#)
- [AWS::CloudWatch::Alarm](#)

Erstellen Sie eine separate Konfigurationsdatei mit dem Namen `options.config` und definieren Sie benutzerdefinierte Optionseinstellungen.

```
option_settings:  
  "aws:elasticbeanstalk:customoption":  
    VisibilityTimeout : 30  
    AlarmEmail : "nobody@example.com"
```

Mit diesen Zeilen wird Elastic Beanstalk angewiesen, die Werte für die Eigenschaften `VisibilityTimeout` und `Subscription Endpoint` aus den Werten `VisibilityTimeout` und `Subscription Endpoint` einer Konfigurationsdatei (in diesem Beispiel „`options.config`“) abzurufen. Diese enthält im Bereich „`option_settings`“ den Abschnitt `aws:elasticbeanstalk:customoption` mit einem Name-Wert-Paar, in dem sich der zu verwendende Wert befindet. Im obigen Beispiel werden folglich die Werte „30“ und „`nobody@amazon.com`“ verwendet. Mehr über `Fn::GetOptionSetting` erfahren Sie unter [the section called “Funktionen” \(p. 750\)](#).

Beispiel: DynamoDB, CloudWatch und SNS

Diese Konfigurationsdatei richtet die DynamoDB-Tabelle als Session Handler für eine PHP-basierte Anwendung mit dem AWS SDK for PHP 2 ein. Wenn Sie dieses Beispiel verwenden möchten, müssen Sie ein IAM-Instance-Profil haben, das zu den Instances in Ihrer Umgebung hinzugefügt und für den Zugriff auf die DynamoDB-Tabelle verwendet wird.

Sie können das in diesem Schritt verwendete Beispiel unter [DynamoDB Session Support-Beispiel](#) herunterladen. Das Beispiel enthält die folgenden Dateien:

- Die Beispieldatenbank, `index.php`
- Eine Konfigurationsdatei, `dynamodb.config`, um eine DynamoDB-Tabelle und andere AWS-Ressourcen zu erstellen und zu konfigurieren sowie Software auf den EC2-Instances zu installieren, die die Anwendung in einer Elastic Beanstalk-Umgebung hosten
- Die Konfigurationsdatei `options.config`, die die Standardwerte in `dynamodb.config` mit spezifischen Einstellungen für diese bestimmte Installation überschreibt

index.php

```
<?php  
  
// Include the SDK using the Composer autoloader  
require '../vendor/autoload.php';  
  
use Aws\DynamoDb\AwsClient;  
  
// 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 <a href="https://aws.amazon.com/php/">PHP Developer Center</a>. </p>

<form id="SimpleForm" name="SimpleForm" method="post" action="index.php">
<?php
echo 'Request serviced from instance ' . $instanceId . '<br/>';
echo '<br/>';

if (isset($_POST['continue'])) {
    session_start();
    $_SESSION['visits'] = $_SESSION['visits'] + 1;
    echo 'Welcome back ' . $_SESSION['username'] . '<br/>';
    echo 'This is visit number ' . $_SESSION['visits'] . '<br/>';
    session_write_close();
    echo '<br/>';
    echo '<input type="Submit" value="Refresh" name="continue" id="continue"/>';
    echo '<input type="Submit" value="Delete Session" name="killsession" id="killsession"/>';
} elseif (isset($_POST['killsession'])) {
    session_start();
    echo 'Goodbye ' . $_SESSION['username'] . '<br/>';
    session_destroy();
    echo 'Username: <input type="text" name="username" id="username" size="30"/><br/>';
    echo '<br/>';
    echo '<input type="Submit" value="New Session" name="newsession" id="newsession"/>';
} elseif (isset($_POST['newsession'])) {
    session_start();
    $_SESSION['username'] = $_POST['username'];
    $_SESSION['visits'] = 1;
    echo 'Welcome to a new session ' . $_SESSION['username'] . '<br/>';
    session_write_close();
    echo '<br/>';
    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.<br/>';
    echo '<br/>';
    echo 'Username: <input type="text" name="username" id="username" size="30"/><br/>';
    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.*"

```

In der Beispielkonfigurationsdatei erstellen wir zunächst die DynamoDB-Tabelle und konfigurieren die Primärschlüssel-Struktur für die Tabelle und die Kapazitätseinheiten, um ausreichend Ressourcen zuzuweisen, um den angeforderten Durchsatz zu bieten. Als Nächstes erstellen wir CloudWatch-Alarne für WriteCapacity und ReadCapacity. Wir erstellen ein SNS-Thema, das eine E-Mail an "nobody@amazon.com" sendet, wenn die Alarmschwellenwerte überschritten werden.

Nachdem wir unsere AWS-Ressourcen für unsere Umgebung erstellt und konfiguriert haben, müssen wir die EC2-Instances anpassen. Wir verwenden den `files`-Schlüssel, um die Details der DynamoDB-Tabelle an die EC2-Instances in unserer Umgebung zu übergeben und ein "require" in der `composer.json`-Datei für das AWS SDK for PHP 2 hinzuzufügen. Schließlich führen wir Container-Befehle aus, um

den Composer und die erforderlichen Abhängigkeiten zu installieren und das Installationsprogramm zu entfernen.

.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
```

Ersetzen Sie den SessionAlarmEmail-Wert durch die E-Mail, an die Alarmbenachrichtigungen gesendet werden sollen. Die options.config-Datei enthält die Werte, die für einige der in dynamodb.config definierten Variablen verwendet werden. Beispielsweise enthält dynamodb.config die folgenden Zeilen:

```
Subscription:
  - Endpoint:
    Fn::GetOptionSetting:
      OptionName: SessionAlarmEmail
      DefaultValue: "nobody@amazon.com"
```

Mit diesen Zeilen wird Elastic Beanstalk angewiesen, den Wert für die Eigenschaft Endpoint aus dem Wert SessionAlarmEmail in einer Konfigurationsdatei (in unserer Beispieldatenbank options.config) abzurufen. Diese enthält im Bereich „option_settings“ den Abschnitt aws:elasticbeanstalk:customoption mit einem Name-Wert-Paar, in dem sich der zu verwendende Wert befindet. Im obigen Beispiel bedeutet dies, dass SessionAlarmEmail der Wert nobody@amazon.com zugewiesen würde.

Weitere Informationen zu den CloudFormation-Ressourcen, die in diesem Beispiel verwendet wurden, finden Sie unter folgenden Referenzen:

- [AWS::DynamoDB::Table](#)
- [AWS::CloudWatch::Alarm](#)
- [AWS::SNS::Topic](#)

Verwenden gespeicherter Elastic Beanstalk-Konfigurationen

Sie können die Umgebungskonfiguration in Amazon Simple Storage Service (Amazon S3) als Objekt speichern und dieses im Rahmen der Umgebungserstellung für andere Umgebungen verwenden oder in einer ausgeführten Umgebung einsetzen. Gespeicherte Konfigurationen sind Vorlagen im YAML-Format, mit denen die [Plattformversion \(p. 32\)](#), [Ebene \(p. 14\)](#), Einstellungen für [Konfigurationsoptionen \(p. 646\)](#) und Tags für eine Umgebung definiert werden.

Sie können Tags auf eine gespeicherte Konfiguration anwenden, wenn Sie sie erstellen, und die Tags der vorhandenen gespeicherten Konfigurationen bearbeiten. Details dazu finden Sie unter [Markieren von gespeicherten Konfigurationen \(p. 769\)](#).

Note

Die Tags einer gespeicherten Konfiguration sind nicht mit den Tags verbunden, die mit dem Tags-Schlüssel in einer gespeicherten Konfiguration angegeben sind. Letztere werden auf eine Umgebung angewendet, wenn Sie die gespeicherte Konfiguration auf die Umgebung anwenden.

Erstellen Sie eine gespeicherte Konfiguration des aktuellen Umgebungsstatus mit der Elastic Beanstalk-Managementkonsole.

So speichern Sie die Umgebungskonfiguration

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) und dann Save configuration (Konfiguration speichern) aus.
4. Verwenden Sie das Formular auf dem Bildschirm, um die gespeicherte Konfiguration zu benennen. Geben Sie optional eine kurze Beschreibung ein und fügen Sie Tag-Schlüssel und -Werte hinzu.
5. Wählen Sie Save (Speichern) aus.
6. Wählen Sie Save (Speichern) aus.

Save Configuration

Save this environment's current configuration.

Environment:

GettingStartedApp-env

Configuration name:

base

Description:

Base configuration

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

mytag1

Value

value1

Remove tag

Add tag

49 remaining

Cancel

Save

Die gespeicherte Konfiguration enthält alle Einstellungen, die Sie über die Konsole oder einen anderen Client, der die Elastic Beanstalk-API nutzt, für die Umgebung vorgenommen haben. Sie können die gespeicherte Konfiguration zu einem späteren Zeitpunkt nutzen, um den ursprünglichen Zustand der Umgebung wiederherzustellen, oder Sie können sie während der [Umgebungserstellung \(p. 437\)](#) in eine neue Umgebung übernehmen.

Wie in folgendem Beispiel veranschaulicht, können Sie eine Konfiguration mit dem EB CLI-Befehl [the section called “eb config” \(p. 1050\)](#) herunterladen. **NAME** steht für den Namen der gespeicherten Konfiguration.

```
eb config get NAME
```

So wenden Sie eine gespeicherte Konfiguration während der Umgebungserstellung an (Elastic Beanstalk-Konsole)

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.

Note

Wenn Sie viele Anwendungen haben, verwenden Sie die Suchleiste, um die Anwendungsliste zu filtern.

3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Saved configurations (Gespeicherte Konfigurationen) aus.
4. Wählen Sie die gespeicherte Konfiguration aus, die Sie anwenden möchten, und wählen Sie dann Launch environment (Umgebung starten).
5. Führen Sie die Schritte im Assistenten aus, um die Umgebung zu erstellen.

Gespeicherte Konfigurationen enthalten keine Einstellungen, die mithilfe von [Konfigurationsdateien \(p. 722\)](#) im Anwendungsquellcode übernommen wurden. Falls dieselbe Einstellung sowohl in einer Konfigurationsdatei als auch in einer gespeicherten Konfiguration definiert ist, hat die Einstellung aus der gespeicherten Konfiguration Vorrang. Ebenso überschreiben in der Elastic Beanstalk-Konsole angegebene Optionen die Optionen in gespeicherten Konfigurationen. Weitere Informationen finden Sie unter [Precedence \(p. 647\)](#).

Gespeicherte Konfigurationen werden im S3-Bucket von Elastic Beanstalk in einem Ordner gespeichert, der nach der Anwendung benannt ist. Beispielsweise befinden sich Konfigurationen für die Anwendung `my-app` in der Region us-west-2 für die Kontonummer 123456789012 unter `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app/`.

Zeigen Sie die Inhalte einer gespeicherten Konfiguration in einem Texteditor an. In der folgenden Beispielkonfiguration wird die Konfiguration einer Webserverumgebung, die mit der Elastic Beanstalk-Managementkonsole gestartet wird, dargestellt.

```
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
```

Sie können die Inhalte einer gespeicherten Konfiguration ändern und am gleichen Speicherort in Amazon S3 speichern. Alle ordnungsgemäß formatierten gespeicherten Konfigurationen können – sofern sie am korrekten Speicherort gespeichert sind – mit der Elastic Beanstalk-Managementkonsole in eine Umgebung übernommen werden.

Folgende Schlüssel werden unterstützt.

- AWSConfigurationTemplateVersion (erforderlich) – Version der Konfigurationsvorlage (1.1.0.0).

```
AWSConfigurationTemplateVersion: 1.1.0.0
```

- Plattform – Der Amazon-Ressourcenname (ARN) der Plattformversion für die Umgebung. Sie können die Plattform nach ARN oder Lösungs-Stack-Namen angeben.

```
Platform:  
  PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit  
  Amazon Linux/2.5.0
```

- SolutionStack – Vollständiger Name des für die Umgebungserstellung verwendeten [Lösungs-Stacks \(p. 32\)](#).

```
SolutionStack: 64bit Amazon Linux 2017.03 v2.5.0 running Java 8
```

- OptionSettings – [Konfigurationsoptions-Einstellungen \(p. 646\)](#) für die Umgebung. Beispielsweise wird mit folgendem Eintrag der Instance-Typ auf "t2.micro" festgelegt.

```
OptionSettings:  
  aws:autoscaling:launchconfiguration:  
    InstanceType: t2.micro
```

- Tags – Bis zu 47 Tags können für in der Umgebung erstellte Ressourcen verwendet werden.

```
Tags:  
  Cost Center: WebApp Dev
```

- EnvironmentTier – Typ der zu erstellenden Umgebung. Bei einer Webserverumgebung kann dieser Bereich ausgelassen werden (Webserver ist der Standardwert). Bei einer Worker-Umgebung verwenden Sie Folgendes.

```
EnvironmentTier:  
  Name: Worker  
  Type: SQS/HTTP
```

Weitere Informationen zu alternativen Methoden zum Erstellen und Verwenden von gespeicherten Konfigurationen finden Sie in den folgenden Themen:

- Festlegen von Konfigurationsoptionen vor der Umgebungserstellung (p. 649)
- Festlegen von Konfigurationsoptionen während der Umgebungserstellung (p. 653)
- Festlegen von Konfigurationsoptionen nach der Umgebungserstellung (p. 658)

Markieren von gespeicherten Konfigurationen

Sie können Ihre gespeicherten AWS Elastic Beanstalk-Konfigurationen mit Tags markieren. Tags sind mit AWS-Ressourcen verknüpfte Schlüssel-Wert-Paare. Weitere Informationen zum Elastic Beanstalk-Ressourcen-Tagging, zu Anwendungsfällen, Einschränkungen für Tag-Schlüssel und -Werte sowie zu unterstützten Ressourcentypen finden Sie unter [Markieren von Elastic Beanstalk-Anwendungsressourcen \(p. 420\)](#).//

Sie können Tags angeben, wenn Sie eine gespeicherte Konfiguration erstellen. In einer vorhandenen gespeicherten Konfiguration können Sie Tags hinzufügen oder entfernen sowie die Werte von vorhandenen Tags aktualisieren. Sie können jeder gespeicherten Konfiguration bis zu 50 Tags.

Hinzufügen von Tags beim Erstellen von gespeicherten Konfigurationen

Wenn Sie die Elastic Beanstalk-Konsole zum [Speichern einer Konfiguration \(p. 764\)](#) verwenden, können Sie Tag-Schlüssel und -Werte auf der Seite Save Configuration (Gespeicherte Konfiguration) angeben.

Wenn Sie eine Konfiguration über die EB CLI speichern, verwenden Sie die --tags-Option mit dem Befehl [eb config \(p. 1050\)](#), um Tags hinzuzufügen.

```
~/workspace/my-app$ eb config --tags mytag1=value1,mytag2=value2
```

Fügen Sie Tags mit der AWS CLI oder anderen API-basierten Clients hinzu, indem Sie den --tags-Parameter im [create-configuration-template](#)-Befehl verwenden.

```
$ 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
```

Verwalten von Tags einer vorhandenen gespeicherten Konfiguration

Sie können Tags in einer vorhandenen gespeicherten Elastic Beanstalk-Konfiguration hinzufügen, aktualisieren und löschen.

So verwalten Sie die Tags einer gespeicherten Konfiguration mit der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.

2. Wählen Sie im Navigationsbereich Anwendungen und wählen Sie dann den Namen Ihrer Anwendung in der Liste aus.

Note

Wenn Sie viele Anwendungen haben, verwenden Sie die Suchleiste, um die Anwendungsliste zu filtern.

3. Suchen Sie im Navigationsbereich den Namen Ihrer Anwendung und wählen Sie Saved configurations (Gespeicherte Konfigurationen) aus.
4. Wählen Sie die gespeicherte Konfiguration, die Sie verwalten möchten.
5. Klicken Sie auf Actions (Aktionen) und dann auf Manage Tags (Tags verwalten).
6. Verwenden Sie das Formular auf dem Bildschirm, um Tags hinzuzufügen, zu aktualisieren oder zu löschen.
7. Wählen Sie Apply (Anwenden) aus.

Wenn Sie die gespeicherte Konfiguration über die EB CLI aktualisieren, verwenden Sie den Befehl [eb tags \(p. 1092\)](#) zum Hinzufügen, Aktualisieren, Löschen oder Auflisten von Tags.

Beispiel: Der folgende Befehl listet die Tags einer gespeicherten Konfiguration auf.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-template"
```

Mit dem folgenden Befehl wird das Tag `mytag1` aktualisiert und das Tag `mytag2` gelöscht.

```
~/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"
```

Eine umfassende Liste der Optionen sowie weitere Beispiele finden Sie unter [eb tags \(p. 1092\)](#).

Wenn Sie die AWS CLI oder andere API-basierte Clients verwenden, verwenden Sie den [list-tags-for-resource](#)-Befehl, um die Tags einer gespeicherten Konfiguration aufzulisten.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-template"
```

Verwenden Sie den Befehl [update-tags-for-resource](#) zum Hinzufügen, Aktualisieren und Löschen von Tags in einer gespeicherten Konfiguration.

```
$ 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"
```

Geben Sie sowohl hinzuzufügende als auch zu aktualisierende Tags im Parameter `--tags-to-add` des Befehls `update-tags-for-resource` an. Wenn ein Tag nicht vorhanden ist, wird es hinzugefügt, andernfalls wird es aktualisiert.

Note

Um einige der EB CLI- und AWS CLI-Befehle mit einer gespeicherten Elastic Beanstalk-Konfiguration zu verwenden, benötigen Sie den ARN der gespeicherten Konfiguration. Um

den ARN zu erstellen, verwenden Sie zunächst den folgenden Befehl, um den Namen der gespeicherten Konfiguration abzurufen.

```
$ aws elasticbeanstalk describe-applications --application-names my-app
```

Suchen Sie den ConfigurationTemplates-Schlüssel in der Ausgabe des Befehls. Dieses Element zeigt den Namen der gespeicherten Konfiguration. Verwenden Sie diesen Namen, wobei *my-template* in den auf dieser Seite erwähnten Befehlen angegeben ist.

Umgebungsmanifest (env.yaml)

Sie können ein Umgebungsmanifest im YAML-Format in den Stamm des Quell-Bundles der Anwendung einbinden und so den Umgebungsnamen, das Lösungs-Stack und [Umgebungslinks \(p. 524\)](#) konfigurieren, die beim Erstellen der Umgebung verwendet werden sollen.

Dieses Dateiformat unterstützt Umgebungsgruppen. Für die Verwendung von Gruppen geben Sie den Umgebungsnamen mit einem Pluszeichen (+) am Ende im Manifest an. Beim Erstellen oder Aktualisieren der Umgebung geben Sie den Gruppennamen mit --group-name (AWS CLI) oder mit --env-group-suffix (EB CLI) an. Weitere Informationen zu Gruppen finden Sie unter [Gruppen von Elastic Beanstalk Umgebungen erstellen oder aktualisieren \(p. 469\)](#).

Im folgenden Beispielmanifest wird eine Webserverumgebung mit einem Link zu einer Basiskomponente in einer Worker-Umgebung definiert. Im Manifest werden Gruppen verwendet, damit mehrere Umgebungen mit dem gleichen Quell-Bundle erstellt werden können:

```
-/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+"
```

Folgende Schlüssel werden unterstützt.

- AWSConfigurationTemplateVersion (erforderlich) – Version der Konfigurationsvorlage (1.1.0.0).

```
AWSConfigurationTemplateVersion: 1.1.0.0
```

- Plattform – Der Amazon-Ressourcenname (ARN) der Plattformversion für die Umgebung. Sie können die Plattform nach ARN oder Lösungs-Stack-Namen angeben.

```
Platform:  
  PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit  
    Amazon Linux/2.5.0
```

- SolutionStack – Vollständiger Name des für die Umgebungserstellung verwendeten [Lösungs-Stacks \(p. 32\)](#).

```
SolutionStack: 64bit Amazon Linux 2017.03 v2.5.0 running Java 8
```

- OptionSettings – [Konfigurationsoptions-Einstellungen \(p. 646\)](#) für die Umgebung. Beispielsweise wird mit folgendem Eintrag der Instance-Typ auf "t2.micro" festgelegt.

```
OptionSettings:  
  aws:autoscaling:launchconfiguration:  
    InstanceType: t2.micro
```

- Tags – Bis zu 47 Tags können für in der Umgebung erstellte Ressourcen verwendet werden.

```
Tags:  
  Cost Center: WebApp Dev
```

- EnvironmentTier – Typ der zu erstellenden Umgebung. Bei einer Webserverumgebung kann dieser Bereich ausgelassen werden (Webserver ist der Standardwert). Bei einer Worker-Umgebung verwenden Sie Folgendes.

```
EnvironmentTier:  
  Name: Worker  
  Type: SQS/HTTP
```

- CName – CNAME der Umgebung. Binden Sie das Plus-Zeichen (+) am Ende des Namens ein, um Gruppen zu aktivieren.

```
CName: front-A08G28LG+
```

- EnvironmentName – Name der zu erstellenden Umgebung. Binden Sie das Plus-Zeichen (+) am Ende des Namens ein, um Gruppen zu aktivieren.

```
EnvironmentName: front+
```

Wenn Gruppen aktiviert sind, müssen Sie bei der Umgebungserstellung einen Gruppennamen angeben. Elastic Beanstalk hängt den Gruppennamen mit einem Bindestrich an die Umgebung an. Beispielsweise erstellt Elastic Beanstalk aus dem Umgebungsnamen `front+` und dem Gruppennamen `dev` eine Umgebung mit dem Namen `front-dev`.

- EnvironmentLinks – Zuordnung von Variablennamen und Umgebungsnamen von Abhängigkeiten. Das folgende Beispiel macht aus der `worker+-`-Umgebung eine Abhängigkeit und weist Elastic Beanstalk an, die Link-Informationen in einer Variable namens `WORKERQUEUE` zu speichern.

```
EnvironmentLinks:
```

```
"WORKERQUEUE" : "worker+"
```

Der Wert der Link-Variable variiert abhängig vom Typ der verknüpften Umgebung. In einer Webserverumgebung besteht der Link aus dem CNAME der Umgebung. Bei einer Worker-Umgebung ist der Link der Amazon Simple Queue Service (Amazon SQS)-Warteschlangenname der Umgebung.

Mit den Schlüsseln CName, EnvironmentName und EnvironmentLinks lassen sich [Umgebungsgruppen \(p. 469\)](#) und [Links zu anderen Umgebungen \(p. 524\)](#) erstellen. Diese Funktionen werden derzeit bei der Verwendung der EB CLI, der AWS CLI oder einem SDK unterstützt.

Verwenden eines benutzerdefinierten Amazon Machine Image (AMI)

Wenn Sie eine AWS Elastic Beanstalk-Umgebung erstellen, können Sie ein Amazon Machine Image (AMI) angeben, das anstelle des in Ihrer Plattformversion enthaltenen Standard-Elastic Beanstalk-AMI verwendet werden soll. Ein benutzerdefiniertes AMI kann Bereitstellungszeiten verbessern, wenn Instances in Ihrer Umgebung gestartet werden, wenn Sie viel Software installieren müssen, die nicht in den Standard-AMIs enthalten ist.

Die Verwendung von [Konfigurationsdateien \(p. 722\)](#) eignet sich hervorragend für die schnelle und konsistente Konfiguration und Anpassung Ihrer Umgebung. Das Anwenden von Konfigurationen kann jedoch während der Umgebungserstellung und Updates viel Zeit in Anspruch nehmen. Wenn Sie viele Server-Konfigurationen in Konfigurationsdateien vornehmen, können Sie diese Zeit verringern, indem Sie eine benutzerdefinierte AMI erstellen, die bereits über die benötigte Software und Konfiguration verfügt.

Mit einem benutzerdefinierten AMI können Sie auch Änderungen an niedrigen Komponenten vornehmen, wie z. B. dem Linux-Kernel, die schwierig zu implementieren sind oder lange für die Anwendung auf Konfigurationsdateien brauchen. Zum Erstellen eines benutzerdefinierten AMI starten Sie ein Elastic Beanstalk-AMI in Amazon EC2, passen die Software und Konfiguration an Ihre Anforderungen an, halten die Instance an und speichern ein AMI daraus.

Erstellen eines benutzerdefinierten AMI

So identifizieren Sie das Basis-AMI von Elastic Beanstalk

1. Führen Sie in einem Befehlsfenster einen ähnlichen Befehl wie den folgenden aus. Geben Sie die AWS-Region an, in der Sie Ihr benutzerdefiniertes AMI verwenden möchten, und ersetzen Sie den Plattform-ARN und die Versionsnummer durch die Elastic Beanstalk-Plattform, auf der Ihre Anwendung basiert.

```
$ 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"
    }
]
```

- Notieren Sie sich den `ImageId`-Wert, der im Ergebnis wie `ami-020ae06fdda6a0f66` aussieht.

Der Wert ist das vorhandene Elastic Beanstalk-AMI für die Plattformversion, EC2-Instance-Architektur und AWS-Region, die für Ihre Anwendung relevant sind. Wenn Sie AMIs für mehrere Plattformen, Architekturen oder AWS-Regionen erstellen müssen, wiederholen Sie diesen Vorgang, um das korrekte Basis-AMI für eine beliebige Kombination zu identifizieren.

Hinweise

- Erstellen Sie kein AMI aus einer Instance, die in einer Elastic Beanstalk-Umgebung gestartet wurde. Elastic Beanstalk nimmt Änderungen an Instances während der Bereitstellung vor, die zu Problemen im gespeicherten AMI führen können. Durch das Speichern eines Image von einer Instance in einer Elastic Beanstalk-Umgebung wird auch die Version Ihrer Anwendung, die für die Instance bereitgestellt wurde, zu einem festen Teil des Image.
- Wir empfehlen Ihnen, immer die neueste Plattformversion zu verwenden. Wenn Sie auf eine neue Plattformversion aktualisieren, empfehlen wir Ihnen außerdem, Ihr benutzerdefiniertes AMI auf das AMI der neuen Plattformversion festzulegen. Dies minimiert Bereitstellungsfehler aufgrund inkompatibler Paket- oder Bibliotheksversionen.

Es ist auch möglich, ein benutzerdefiniertes AMI von einem Community-AMI zu erstellen, dass nicht von Elastic Beanstalk veröffentlicht wurde. Sie können das neueste [Amazon Linux](#)-AMI als Ausgangspunkt verwenden. Wenn Sie eine Umgebung mit einem Linux AMI starten, das nicht von Elastic Beanstalk verwaltet wird, versucht Elastic Beanstalk die Platformsoftware (Sprache, Framework, Proxy-Server usw.) und zusätzliche Komponenten zur Unterstützung von Funktionen, wie [Erweiterte Zustandsberichte \(p. 824\)](#) zu installieren.

Note

AMIs, die nicht von Elastic Beanstalk verwaltet werden, werden für Windows Server-basierte Elastic Beanstalk-Plattformen nicht unterstützt.

Obwohl Elastic Beanstalk ein AMI verwenden kann, das nicht von Elastic Beanstalk verwaltet wird, kann die erhöhte Bereitstellungszeit, die sich daraus ergibt, dass Elastic Beanstalk fehlende Komponenten installiert, die Vorteile der Erstellung eines benutzerdefinierten AMI im Vorhinein verringern oder aufheben. Andere Linux-Distributionen könnten mit einiger Fehlerbehebung funktionieren, werden aber nicht offiziell unterstützt. Wenn Ihre Anwendung eine spezifische Linux-Distribution erfordert, ist es alternativ auch möglich, ein Docker-Image zu erstellen und es auf der [Docker-Plattform \(p. 48\)](#) oder [Multicontainer Docker-Plattform \(p. 64\)](#) von Elastic Beanstalk auszuführen.

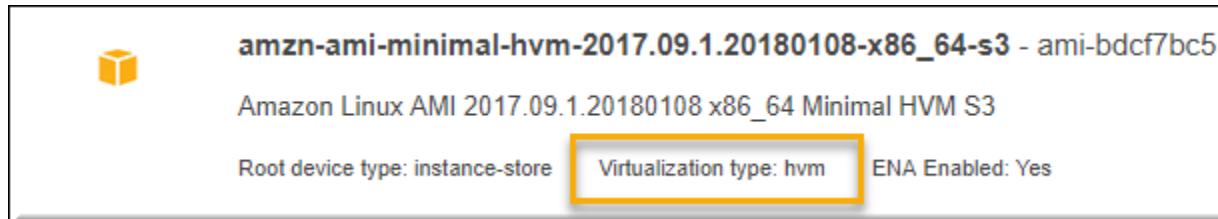
So erstellen Sie ein benutzerdefiniertes AMI

- Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
- Wählen Sie Launch Instance aus.
- Wählen Sie Community AMIs (Community-AMIs) aus.
- Wenn Sie ein Basis-AMI von Elastic Beanstalk oder Amazon Linux identifiziert haben, das Sie anpassen möchten, um ein benutzerdefiniertes AMI zu erhalten, geben Sie die AMI-ID in das Suchfeld ein und drücken anschließend auf Enter (Eingabetaste).

Sie können auch die Liste für eine andere Community-AMI durchsuchen, die Ihren Anforderungen entspricht.

Note

Wir empfehlen, dass Sie eine AMI mit HVM-Virtualisierung verwenden. Diese AMIs zeigen Virtualization type: hvm (Virtualisierungstyp: hvm) in ihrer Beschreibung an.



Weitere Informationen zu Instance-Virtualisierungstypen finden Sie unter [Linux AMI-Virtualisierungstypen](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances.

5. Wählen Sie Select (Auswählen) zum Auswählen des AMI aus.
 6. Wählen Sie einen Instance-Typen und danach Next: Configure Instance Details (Weiter: Instance-Details konfigurieren) aus.
 7. (Linux-Plattformen) Erweitern Sie den Abschnitt Advanced Details (Fortgeschrittene Details) und fügen Sie den folgenden Text in das Feld User Data (Benutzerdaten) ein.

```
#cloud-config
  repo_releasever: repository version number
  repo_upgrade: none
```

Die Repository-Versionsnummer ist das Jahr und der Monat der Version im AMI-Namen. Beispiel: AMIs basierend auf der März 2015-Version von Amazon Linux haben die Repository-Versionsnummer 2015 . 03. Bei einem Elastic Beanstalk-Image entspricht dies dem Datum, das im Namen des Lösungs-Stacks für Ihre [Plattformversion](#) (p. 32) auf der Basis von Amazon Linux AMI (Vorgängerversion von Amazon Linux 2) angegeben ist.

Note

Die Einstellung `repo_releasever` konfiguriert die Lock-on-Launch-Funktion für ein Amazon Linux AMI. Dies bewirkt, dass das AMI beim Start eine feste, spezifische Repository-Version verwendet. Diese Funktion wird unter Amazon Linux 2 nicht unterstützt. Geben Sie sie nicht an, wenn Ihre Umgebung einen aktuellen Amazon Linux 2-Plattformzweig verwendet. Die Einstellung ist erforderlich, wenn Sie ein benutzerdefiniertes AMI mit Elastic Beanstalk ausschließlich auf Amazon Linux AMI-Plattformzweigen verwenden (vor Amazon Linux 2). Die Einstellung `repo_upgrade` deaktiviert die automatische Installation von Sicherheitsupdates. Dies ist für die Verwendung eines benutzerdefinierten AMI mit Elastic Beanstalk erforderlich.

8. Führen Sie die Schritte im Assistenten aus, um die [EC2-Instance zu starten](#). Wenn Sie dazu aufgefordert werden, wählen Sie ein Schlüsselpaar, auf das Sie Zugriff haben, um die Verbindung zur Instance für die nächsten Schritte herzustellen.
 9. [Stellen Sie eine Verbindung mit der Instance](#) über SSH oder RDP her.
 10. Führen Sie alle gewünschten Anpassungen durch.
 11. (Windows-Plattformen) Führen Sie EC2Config service Sysprep aus. Weitere Informationen zu EC2Config finden Sie unter [Konfigurieren einer Windows-Instance mithilfe des EC2Config-Service](#). Stellen Sie sicher, dass Sysprep konfiguriert ist, um ein zufälliges Kennwort zu generieren, das von der AWS Management Console abgerufen werden kann.
 12. Stoppen Sie in der Amazon EC2-Konsole die EC2-Instance. Anschließend wählen Sie im Menü Instance Actions (Instance-Aktionen) die Option Create Image (EBS AMI) (Image erstellen (EBS AMI)) aus.
 13. Um zusätzliche AWS-Gebühren zu vermeiden, [beenden Sie die EC2-Instance](#).

So verwenden Sie Ihr benutzerdefiniertes AMI in einer Elastic Beanstalk-Umgebung

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Capacity (Kapazität) die Option Edit (Bearbeiten).
5. Geben Sie bei AMI-ID Ihre benutzerdefinierte AMI-ID ein.
6. Wählen Sie Apply (Anwenden) aus.

Wenn Sie eine neue Umgebung mit dem benutzerdefinierten AMI erstellen, sollten Sie dieselbe Plattformversion verwenden, die Sie als Grundlage für das Erstellen des AMI verwendet haben. Wenn Sie später eine [Plattformaktualisierung \(p. 491\)](#) für eine Umgebung mit einem benutzerdefinierten AMI anwenden, versucht Elastic Beanstalk die Bibliotheks- und Konfigurationsaktualisierungen während des Bootstrapping-Prozesses anzuwenden.

Bereinigen eines benutzerdefinierten AMI

Wenn Sie mit einem benutzerdefinierten AMI fertig sind und dieses nicht mehr benötigen, um Elastic Beanstalk-Umgebungen zu starten, sollten Sie in Betracht ziehen, es zu bereinigen, um die Speicherkosten zu minimieren. Die Bereinigung eines benutzerdefinierten AMI beinhaltet dessen Abmeldung von Amazon EC2 und das Löschen anderer zugehöriger Ressourcen. Weitere Informationen finden Sie unter [Abmelden Ihres Linux AMI](#) oder [Abmelden Ihres Windows AMI](#).

Bereitstellen von statischen Dateien

Um die Leistung zu verbessern, können Sie den Proxy-Server so konfigurieren, dass er statische Dateien (z. B. HTML oder Bilder) aus Verzeichnissen innerhalb Ihrer Webanwendung bereitstellt. Wenn der Proxy-Server eine Anfrage nach einer Datei unter dem angegebenen Pfad erhält, stellt er die Datei direkt bereit, statt die Anfrage an Ihre Anwendung weiterzuleiten.

Elastic Beanstalk unterstützt die Konfiguration des Proxys zur Bereitstellung statischer Dateien für die meisten auf Amazon Linux 2 basierenden Branches. Die einzige Ausnahme ist Docker.

Note

Elastic Beanstalk konfiguriert auf den Python- und Ruby-Plattformen standardmäßig einige statische Dateiordner. Weitere Informationen finden Sie in den Abschnitten zur Konfiguration statischer Dateien für [Python \(p. 357\)](#) und [Ruby \(p. 385\)](#). Sie können zusätzliche Ordner konfigurieren, wie auf dieser Seite erläutert.

Konfigurieren statischer Dateien über die Konsole

So konfigurieren Sie den Proxy-Server für die Bereitstellung statischer Dateien

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.

2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Geben Sie im Abschnitt Static files (Statische Dateien) einen Pfad für die Bereitstellung statischer Dateien und das Verzeichnis der statischen Dateien ein, die zur leeren Zeile am Ende der Liste bereitgestellt werden sollen.

Note

Wenn Ihnen der Abschnitt Static Files (Statische Dateien) nicht angezeigt wird, müssen Sie mittels einer [Konfigurationsdatei \(p. 722\)](#) mindestens eine Zuweisung hinzufügen. Weitere Informationen finden Sie unter [the section called “Konfigurieren statischer Dateien mithilfe von Konfigurationsoptionen” \(p. 777\)](#) auf dieser Seite.

Der Pfad muss mit einem Schrägstrich beginnen (/). Geben Sie im Stamm des Quellcodes Ihrer Anwendung einen Verzeichnisnamen ein. Dieser darf nicht mit einem Schrägstrich beginnen.

Wenn Sie eine Zuordnung hinzufügen, erscheint eine zusätzliche Zeile (falls Sie eine weitere hinzufügen möchten). Um eine Zuweisung zu entfernen, klicken Sie auf das Symbol für Remove (Entfernen).

Static files

Configure the proxy server to serve static files to reduce the request load on your application. [Learn more](#)

Path (Example: /assets)	Directory (Example: /static/assets)
/html	statichtml X
/images	staticimages X

6. Wählen Sie Apply (Anwenden) aus.

Konfigurieren statischer Dateien mithilfe von Konfigurationsoptionen

Sie können eine [Konfigurationsdatei \(p. 722\)](#) verwenden, um statische Dateipfade und Verzeichnisspeicherorte mithilfe von Konfigurationsoptionen zu konfigurieren. Sie können dem Quellpaket

Ihrer Anwendung eine Konfigurationsdatei hinzufügen und sie während der Erstellung der Umgebung oder einer späteren Bereitstellung bereitstellen.

Wenn Ihre Umgebung einen Plattformvariante basierend auf Amazon Linux 2 verwendet, verwenden Sie den Namespace [aws:elasticbeanstalk:environment:proxy:staticfiles \(p. 691\)](#).

In der folgenden Beispielkonfigurationsdatei wird der Proxyserver angewiesen, Dateien im Ordner statichtml auf dem Pfad /html und Dateien im Ordner staticimages auf dem Pfad /images bereitzustellen.

Example .ebextensions/static-files.config

```
option_settings:  
  aws:elasticbeanstalk:environment:proxy:staticfiles:  
    /html: statichtml  
    /images: staticimages
```

Wenn Ihre Elastic Beanstalk-Umgebung eine Amazon Linux-AMI-Plattformversion (vor Amazon Linux 2) verwendet, lesen Sie die folgenden zusätzlichen Informationen:

Plattformspezifische Namespaces für Amazon Linux-AMI

Auf Amazon Linux AMI-Plattformvarianten variieren Namespaces für statische Dateien je nach Plattform. Details finden Sie auf einer der folgenden Seiten:

- [Go-Konfigurations-Namespace \(p. 103\)](#)
- [Java SE-Konfigurations-Namespace \(p. 130\)](#)
- [Tomcat-Konfigurations-Namespaces \(p. 120\)](#)
- [Node.js-Konfigurations-Namespace \(p. 253\)](#)
- [Python-Konfigurations-Namespaces \(p. 358\)](#)

Konfigurieren von HTTPS für Elastic Beanstalk-Umgebung

Wenn Sie einen [benutzerdefinierten Domänennamen \(p. 644\)](#) für Ihre Elastic Beanstalk-Umgebung gekauft und konfiguriert haben, können Sie HTTPS verwenden, damit die Benutzer eine sichere Verbindung mit Ihrer Website herstellen können. Wenn Sie nicht über einen Domänennamen verfügen, können Sie HTTPS trotzdem mit einem selbstsignierten Zertifikat für Entwicklungs- und Testzwecke verwenden. HTTPS ist ein Muss für jede Anwendung, die Benutzerdaten oder Anmeldeinformationen überträgt.

Die einfachste Möglichkeit für die Verwendung von HTTPS mit einer Elastic Beanstalk-Umgebung ist, [ein Serverzertifikat zum Load Balancer Ihrer Umgebung zuzuweisen \(p. 783\)](#). Wenn Sie Ihren Load Balancer so konfigurieren, dass er HTTPS beendet, ist die Verbindung zwischen dem Client und dem Load Balancer sicher. Backend-Verbindungen zwischen dem Load Balancer und EC2-Instances verwenden HTTP, sodass keine zusätzliche Konfiguration der Instances erforderlich ist.

Note

Mit [AWS Certificate Manager \(ACM\)](#) können Sie kostenlos ein vertrauenswürdiges Zertifikat für Ihre Domänennamen erstellen. ACM-Zertifikate können nur mit AWS Load Balancern verwendet werden, und Amazon CloudFront-Verteilungen und ACM stehen [nur in bestimmten Regionen zur Verfügung](#).

Informationen zur Verwendung eines ACM-Zertifikats mit Elastic Beanstalk finden Sie unter [Load Balancer Ihrer Elastic Beanstalk-Umgebung zum Terminieren von HTTPS konfigurieren \(p. 783\)](#).

Wenn Sie Ihre Anwendung in einer Umgebung mit einer einzelnen Instance ausführen oder die Verbindung bis zu den EC2-Instances hinter dem Load Balancer sichern müssen, können Sie [den Proxy-Server, der auf der Instance ausgeführt wird, so konfigurieren, dass HTTPS beendet wird \(p. 785\)](#). Das Konfigurieren Ihrer Instances für das Beenden von HTTPS-Verbindungen erfordert die Nutzung von [Konfigurationsdateien \(p. 722\)](#) zum Ändern der Software, die auf den Instances ausgeführt wird, und die Änderung von Sicherheitsgruppen, um sichere Verbindungen zuzulassen.

Für End-to-End-HTTPS in einer Umgebung mit Lastausgleich können Sie [Instance- und Load Balancer-Beendigung kombinieren \(p. 810\)](#), um beide Verbindungen zu verschlüsseln. Wenn Sie den Load Balancer so konfigurieren, dass er Datenverkehr mithilfe von HTTPS weiterleitet, wird standardmäßig jedem Zertifikat vertraut, das ihm von den Backend-Instances vorgelegt wird. Zur Maximierung der Sicherheit können Sie Richtlinien an den Load Balancer anfügen, die verhindern, dass eine Verbindung mit Instances hergestellt wird, die kein öffentliches Zertifikat darstellen, dem er vertraut.

Note

Sie können den Load Balancer auch so konfigurieren, dass er [HTTPS-Datenverkehr ohne Entschlüsselung weiterleitet \(p. 813\)](#). Der Nachteil dieser Methode besteht darin, dass der Load Balancer die Anfragen nicht sehen und so das Routing nicht optimieren oder Antwortmetriken melden kann.

Wenn ACM in Ihrer Region nicht verfügbar ist, können Sie ein vertrauenswürdiges Zertifikat von einem Drittanbieter kaufen. Ein Drittanbieter-Zertifikat kann verwendet werden, um HTTPS-Datenverkehr auf Ihrem Load Balancer, auf den Backend-Instances oder beiden zu entschlüsseln.

Für die Entwicklung und das Testen können Sie selbst [ein Zertifikat erstellen und signieren \(p. 780\)](#). Verwenden Sie dazu Open-Source-Tools. Selbstsignierte Zertifikate können kostenlos und einfach erstellt werden, sie können aber nicht für die Front-End-Entschlüsselung auf öffentlichen Websites verwendet werden. Wenn Sie versuchen, ein selbstsigniertes Zertifikat für eine HTTPS-Verbindung mit einem Client zu verwenden, zeigt der Browser des Benutzers eine Fehlermeldung an, die darauf hinweist, dass Ihre Website nicht sicher ist. Sie können jedoch ein selbstsigniertes Zertifikat verwenden, um Backend-Verbindungen ohne Probleme zu sichern.

Wir empfehlen ACM als bevorzugtes Tool, wenn Sie Ihr Server-Zertifikate programmgesteuert oder unter Verwendung der AWS-CLI bereitstellen und verwalten. Wenn ACM nicht [in Ihrer AWS-Region verfügbar ist](#), können Sie [ein Drittanbieter- oder selbstsigniertes Zertifikat und einen privaten Schlüssel \(p. 782\)](#) mit Hilfe der AWS-CLI in AWS Identity and Access Management (IAM) hochladen. Zertifikate, die in IAM gespeichert sind, können nur mit Load Balancern und CloudFront-Verteilungen verwendet werden.

Note

Die Beispielanwendung [Does it have Snakes?](#) auf GitHub umfasst Konfigurationsdateien und Anweisungen für jede einzelne Methode der Konfiguration von HTTPS mit einer Tomcat-Webanwendung. Weitere Informationen dazu finden Sie in der [Readme-Datei](#) und den [HTTPS-Anweisungen](#).

Themen

- [Erstellen und Signieren eines X.509-Zertifikats \(p. 780\)](#)
- [Hochladen eines Zertifikats in IAM \(p. 782\)](#)
- [Load Balancer Ihrer Elastic Beanstalk-Umgebung zum Terminieren von HTTPS konfigurieren \(p. 783\)](#)
- [Konfigurieren Ihrer Anwendung, um HTTPS-Verbindungen auf der Instance zu beenden \(p. 785\)](#)
- [Ende-zu-Ende-Verschlüsselung in einer Umgebung mit Elastic Beanstalk-Load-Balancing konfigurieren \(p. 810\)](#)
- [Konfigurieren des Load Balancers der Umgebung für TCP-Pass-Through \(p. 813\)](#)
- [Sicheres Speichern von privaten Schlüsseln in Amazon S3 \(p. 814\)](#)

- Konfigurieren der Umleitung von HTTP zu HTTPS (p. 815)

Erstellen und Signieren eines X.509-Zertifikats

Sie können ein X.509-Zertifikat für Ihre Anwendung mit OpenSSL erstellen. OpenSSL ist eine Standard-Open-Source-Bibliothek, die eine Vielzahl von kryptografischen Funktionen unterstützt, einschließlich der Erstellung und Signierung von x509-Zertifikaten. Weitere Informationen zu OpenSSL finden Sie unter www.openssl.org.

Note

Sie müssen nur ein Zertifikat lokal erstellen, wenn Sie [HTTPS in einer Einzel-Instance-Umgebung verwenden \(p. 785\)](#) oder [auf dem Backend neu verschlüsseln \(p. 810\)](#) möchten mit einem selbstsignierten Zertifikat. Wenn Sie einen Domänennamen besitzen, können Sie ein Zertifikat in AWS erstellen und es kostenlos mit einer lastverteilten Umgebung über AWS Certificate Manager (ACM) verwenden. Weitere Informationen finden Sie unter [Request a Certificate](#) im AWS Certificate Manager-Benutzerhandbuch.

Führen Sie `openssl version` in der Befehlszeile aus, um zu prüfen, ob Sie bereits OpenSSL installiert haben. Wenn nicht, können Sie den Quellcode mithilfe der Anweisungen im [öffentlichen GitHub-Repository](#) erstellen und installieren oder Ihren bevorzugten Package Manager verwenden. OpenSSL ist auch auf dem Linux-Image von Elastic Beanstalk installiert. Eine schnelle Alternative, eine Verbindung zu einer EC2-Instance in einer laufenden Umgebung herzustellen, ist also die Verwendung des [EB CLI \(p. 1008\)](#)-eb ssh-Befehls:

```
~/eb$ eb ssh
[ec2-user@ip-255-55-55-255 ~]$ openssl version
OpenSSL 1.0.1k-fips 8 Jan 2015
```

Sie müssen einen privaten RSA-Schlüssel erstellen, um Ihre Zertifikatsignieranforderung zu erstellen. Zum Erstellen Ihres privaten Schlüssels verwenden Sie den Befehl `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

Der Name der Datei, in der Sie den privaten Schlüssel speichern möchten. Normalerweise zeigt der Befehl `openssl genrsa` den Inhalt des privaten Schlüssels auf dem Bildschirm an; dieser Befehl übergibt die Ausgabe jedoch in eine Datei. Wählen Sie einen beliebigen Dateinamen und speichern Sie die Datei an einem sicheren Ort, sodass Sie sie später abrufen können. Wenn Sie den privaten Schlüssel verlieren, können Sie das Zertifikat nicht mehr verwenden.

Eine CSR ist eine Datei, die Sie an eine Zertifizierungsstelle senden, um ein digitales Serverzertifikat zu erhalten. Zum Erstellen einer CSR verwenden Sie den Befehl `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.

Geben Sie die erforderlichen Informationen ein und drücken Sie Enter (Eingabetaste). Die folgende Tabelle beschreibt und enthält Beispiele für jedes Feld.

Name	Beschreibung	Beispiel
Ländername	Die zweistellige ISO-Abkürzung für Ihr Land	US = USA
Bundesstaat oder Provinz	Der Name des Bundesstaats oder der Provinz, in dem bzw. der sich Ihre Organisation befindet. Dieser Name darf nicht abgekürzt werden.	Washington
Ortsname	Der Name der Stadt, in der sich Ihre Organisation befindet.	Seattle
Name der Organisation	Der vollständige, offizielle Name Ihrer Organisation. Kürzen Sie den Namen Ihrer Organisation nicht ab.	Beispielunternehmen
Organisationseinheit	Optional, für weitere Informationen zu Ihrer Organisation.	Marketing
Common Name	Der vollqualifizierte Domänenname für Ihre Website. Dieser muss mit dem Domänenamen übereinstimmen, den Benutzer sehen, wenn sie Ihre Website besuchen. Andernfalls werden Zertifikatfehler angezeigt.	www.example.com
E-Mail-Adresse	Die E-Mail-Adresse des Websiteadministrators.	someone@example.com

Sie können die Signierungsanforderung zur Signierung an einen Dritten weitergeben oder sie für Entwicklung und Tests selbst signieren. Selbstsignierte Zertifikate können auch für Backend-HTTPS zwischen einem Load Balancer und EC2-Instances verwendet werden.

Zum Signieren des Zertifikats verwenden Sie den Befehl openssl x509. Im folgenden Beispiel werden der private Schlüssel aus dem vorherigen Schritt ([privatekey.pem](#)) und die Signierungsanforderung ([csr.pem](#)) zum Erstellen eines öffentlichen Zertifikats mit dem Namen [public.crt](#) verwendet, das [365](#) Tage gültig ist.

```
$ 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
```

Behalten Sie den privaten Schlüssel und das öffentliche Zertifikat für eine spätere Nutzung. Sie können die Signierungsanforderung verwerfen. [Speichern Sie den privaten Schlüssel immer an einem sicheren Ort \(p. 814\)](#) und vermeiden Sie es, ihn zu Ihrem Quellcode hinzuzufügen.

Zum Verwenden des Zertifikats mit der Windows Server-Plattform müssen Sie es in ein PFX-Format konvertieren. Verwenden Sie den folgenden Befehl zum Erstellen eines PFX-Zertifikats aus dem privaten Schlüssel und den öffentlichen Zertifikatsdateien:

```
$ openssl pkcs12 -export -out example.com.pfx -inkey privatekey.pem -in public.crt
Enter Export Password: password
Verifying - Enter Export Password: password
```

Wenn Sie ein Zertifikat haben, können Sie es für die Verwendung mit einem Load Balancer [in IAM hochladen \(p. 782\)](#) oder [die Instances in Ihrer Umgebung konfigurieren, um HTTPS zu beenden \(p. 785\)](#).

Hochladen eines Zertifikats in IAM

Um Ihr Zertifikat mit dem Load Balancer Ihrer Elastic Beanstalk-Umgebung zu verwenden, laden Sie das Zertifikat und den privaten Schlüssel in AWS Identity and Access Management (IAM) hoch. Sie können ein im IAM gespeichertes Zertifikat mit Elastic Load Balancing Load Balancers und Amazon CloudFront-Distributionen verwenden.

Note

Für die Bereitstellung und Verwaltung von Serverzertifikaten empfehlen wir, AWS Certificate Manager (AWS Certificate Manager) zu verwenden. Weitere Informationen zum Anfordern eines ACM-Zertifikats finden Sie unter [Zertifikat anfordern](#) im AWS Certificate Manager-Benutzerhandbuch. Weitere Informationen zum Importieren von Zertifikaten von Drittanbietern in AWS Certificate Manager finden Sie unter [Importieren von Zertifikaten](#) im Leitfaden AWS Certificate Manager-Benutzerhandbuch. Verwenden Sie nur dann IAM zum Hochladen eines Zertifikats, wenn AWS Certificate Manager [in Ihrer AWS-Region nicht zur Verfügung steht](#).

Sie können Ihr Zertifikat mit der [AWS Command Line Interface \(p. 1007\)](#) (AWS CLI) hochladen. Der folgende Befehl lädt ein selbstsigniertes Zertifikat mit dem Namen `https-cert.crt` mit dem privaten Schlüssel `private-key.pem` hoch:

```
$ 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": "AS5YBEIONO2Q7CAIHNGC",
        "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"
    }
}
```

Das `file://`-Präfix weist die AWS-Befehlszeilenschnittstelle an, den Inhalt einer Datei in das aktuelle Verzeichnis zu laden. `elastic-beanstalk-x509` gibt den Namen zum Aufrufen des Zertifikats in IAM an.

Wenn Sie ein Zertifikat von einer Zertifizierungsstelle erworben und eine Zertifikatkettendatei erhalten haben, laden Sie diese auch mit der `--certificate-chain`-Option hoch:

```
$ 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
```

Notieren Sie den Amazon-Ressourcennamen (ARN) für Ihr Zertifikat. Sie verwenden diesen, wenn Sie die Konfigurationseinstellungen Ihres Load Balancer für die Verwendung von HTTPS aktualisieren.

Note

Ein Zertifikat in IAM hochgeladenes Zertifikat bleibt auch dann gespeichert, wenn es in keinem Load Balancer der Umgebung mehr verwendet wird. Es enthält sensible Daten. Wenn Sie das Zertifikat für keine Umgebung mehr benötigen, stellen Sie sicher, dass Sie es löschen. Weitere Informationen zum Löschen eines Zertifikats von IAM finden Sie unter https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_server-certs.html#delete-server-certificate.

Weitere Informationen über Serverzertifikate in IAM finden Sie unter [Arbeiten mit Serverzertifikaten im IAM-Benutzerhandbuch](#).

Load Balancer Ihrer Elastic Beanstalk-Umgebung zum Terminieren von HTTPS konfigurieren

Wenn Sie Ihre AWS Elastic Beanstalk-Umgebung für die Verwendung von HTTPS aktualisieren müssen, konfigurieren Sie einen HTTPS-Listener für den Load Balancer in Ihrer Umgebung. Zwei Arten von Load Balancer unterstützen einen HTTPS-Listener: Classic Load Balancer und Application Load Balancer.

Sie können die Elastic Beanstalk-Konsole oder eine Konfigurationsdatei verwenden, um einen sicheren Listener zu konfigurieren und das Zertifikat zuzuweisen.

Note

Umgebungen mit einer Instance enthalten keinen Load Balancer und unterstützen die HTTPS-Terminierung am Load Balancer nicht.

Sicheren Listener mit der Elastic Beanstalk-Konsole konfigurieren

So weisen Sie ein Zertifikat zum Load Balancer Ihrer Umgebung hinzu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Load Balancer die Option Edit (Bearbeiten) aus.

Note

Ist in der Konfigurationskategorie Load balancer die Schaltfläche Edit (Bearbeiten) nicht vorhanden, verfügt Ihre Umgebung über keinen [Load Balancer \(p. 514\)](#).

5. Auf der Seite Modify load balancer (Load Balancer ändern) variiert die Vorgehensweise je nach dem Typ des Load Balancer, der Ihrer Umgebung zugeordnet ist.
 - Classic Load Balancer
 - a. Wählen Sie Add listener (Listener hinzufügen) aus.
 - b. Konfigurieren Sie im Dialogfeld Classic Load Balancer listener die folgenden Einstellungen:
 - Geben Sie für Listener port (Listener-Port) den Port für eingehenden Datenverkehr ein, in der Regel 443.
 - Wählen Sie für Listener protocol (Listener-Protokoll) die Option HTTPS aus.
 - Geben Sie für Instance port (Instance-Port) den Wert 80 ein.
 - Wählen Sie für Instance protocol (Instance-Protokoll) die Option HTTP aus.
 - Wählen Sie für SSL certificate (SSL-Zertifikat) Ihr Zertifikat aus.
 - c. Wählen Sie Add aus.
 - Application Load Balancer
 - a. Wählen Sie Add listener (Listener hinzufügen) aus.

- b. Konfigurieren Sie im Dialogfeld Application Load Balancer listener die folgenden Einstellungen:
 - Geben Sie für Port den Port für eingehenden Datenverkehr ein, in der Regel 443.
 - Wählen Sie für Protocol (Protokoll) die Option HTTPS aus.
 - Wählen Sie für SSL certificate (SSL-Zertifikat) Ihr Zertifikat aus.
- c. Wählen Sie Add aus.

Note

Wenn das Dropdown-Menü für Classic Load Balancer und Application Load Balancer keine Zertifikate anzeigt, sollten Sie ein Zertifikat für Ihren [benutzerdefinierten Domänennamen \(p. 644\)](#) in [AWS Certificate Manager \(ACM\)](#) erstellen oder hochladen (bevorzugt). Alternativ können Sie mit der AWS-CLI ein Zertifikat in IAM hochladen.

- Netzwerk-Load-Balancer
 - a. Wählen Sie Add listener (Listener hinzufügen) aus.
 - b. Geben Sie im Dialogfeld Network Load Balancer-Listener für Port den Port für eingehenden Datenverkehr ein, normalerweise 443.
 - c. Wählen Sie Add aus.
- 6. Wählen Sie Apply (Anwenden) aus.

Konfigurieren eines sicheren Listeners mit einer Konfigurationsdatei

Sie können einen sicheren Listener auf Ihrem Load Balancer mit einer der folgenden [Konfigurationsdateien \(p. 722\)](#) konfigurieren.

Example .ebextensions/securelistener-clb.config

Verwenden Sie dieses Beispiel, wenn Ihre Umgebung über einen Classic Load Balancer verfügt. Im Beispiel werden Optionen im `aws:elb:listener`-Namespace verwendet, um einen HTTPS-Listener auf Port 443 mit dem angegebenen Zertifikat zu konfigurieren und den verschlüsselten Datenverkehr an die Instances in Ihrer Umgebung auf Port 80 weiterzuleiten.

```
option_settings:  
  aws:elb:listener:443:  
    SSLCertificateId: arn:aws:acm:us-east-2:1234567890123:certificate/  
#####  
  ListenerProtocol: HTTPS  
  InstancePort: 80
```

Ersetzen Sie den markierten Text mit dem ARN Ihres Zertifikats. Dies kann ein Zertifikat sein, das Sie entweder in AWS Certificate Manager (ACM) (bevorzugt) erstellt oder hochgeladen haben, oder das sie mit der AWS-CLI in IAM hochgeladen haben.

Weitere Informationen zu Classic Load Balancer-Konfigurationsoptionen finden Sie unter [Classic Load Balancer-Konfiguration in Namespaces \(p. 565\)](#).

Example .ebextensions/securelistener-alb.config

Verwenden Sie dieses Beispiel, wenn Ihre Umgebung über einen Application Load Balancer verfügt. Im Beispiel werden Optionen im `aws:elbv2:listener`-Namespace zum Konfigurieren eines HTTPS-

Listener auf Port 443 mit dem angegebenen Zertifikat verwendet. Der Listener leitet den Datenverkehr an den Standardprozess weiter.

```
option_settings:  
  aws:elbv2:listener:443:  
    ListenerEnabled: 'true'  
    Protocol: HTTPS  
    SSLCertificateArns: arn:aws:acm:us-east-2:1234567890123:certificate/  
#####
```

Example .ebextensions/securelistener-nlb.config

Verwenden Sie dieses Beispiel, wenn Ihre Umgebung über einen Network Load Balancer verfügt. Im Beispiel werden Optionen im aws:elbv2:listener-Namespace zum Konfigurieren eines Listener auf Port 443 verwendet. Der Listener leitet den Datenverkehr an den Standardprozess weiter.

```
option_settings:  
  aws:elbv2:listener:443:  
    ListenerEnabled: 'true'
```

Konfigurieren einer Sicherheitsgruppe

Wenn Sie Ihren Load Balancer so konfigurieren, dass er Datenverkehr an einen anderen Instance-Port als Port 80 weiterleitet, müssen Sie der Sicherheitsgruppe eine entsprechende Regel hinzufügen, die eingehenden Datenverkehr über den Instance-Port von Ihrem Load Balancer zulässt. Wenn Sie Ihre Umgebung in einer benutzerdefinierten VPC erstellen, fügt Elastic Beanstalk diese Regel für Sie hinzu.

Sie fügen diese Regel durch das Hinzufügen eines Resources-Schlüssels zu einer [Konfigurationsdatei \(p. 722\)](#) im .ebextensions-Verzeichnis für Ihre Anwendung hinzu.

Die folgende Beispiel-Konfigurationsdatei fügt eine Ingress-Regel zur AWSEBSecurityGroup-Sicherheitsgruppe hinzu. Dadurch ist Datenverkehr auf Port 1000 von der Sicherheitsgruppe des Load Balancer zulässig.

Example .ebextensions/sg-ingressfromlb.config

```
Resources:  
  sslSecurityGroupIngress:  
    Type: AWS::EC2::SecurityGroupIngress  
    Properties:  
      GroupId: {"Fn::GetAtt": ["AWSEBSecurityGroup", "GroupId"]}  
      IpProtocol: tcp  
      ToPort: 1000  
      FromPort: 1000  
      SourceSecurityGroupName: {"Fn::GetAtt": ["AWSEBLoadBalancer",  
      "SourceSecurityGroup.GroupName"]}
```

Konfigurieren Ihrer Anwendung, um HTTPS-Verbindungen auf der Instance zu beenden

Sie können [Konfigurationsdateien \(p. 722\)](#) zum Konfigurieren des Proxy-Servers verwenden, der Datenverkehr an Ihre Anwendung weiterleitet, um HTTPS-Verbindungen zu beenden. Dies ist nützlich, wenn Sie HTTPS mit einer Umgebung mit einer einzelnen Instance verwenden möchten, oder wenn Sie den Load Balancer so konfigurieren möchten, dass Datenverkehr durchgeht, ohne entschlüsselt werden.

Für die Aktivierung von HTTPS müssen Sie eingehenden Datenverkehr an die EC2-Instance, auf der die Elastic Beanstalk-Anwendung ausgeführt wird, auf Port 443 zulassen. Dazu verwenden Sie den

Resources-Schlüssel in der Konfigurationsdatei, um eine Regel für Port 443 zu den Ingress-Regeln für die AWSEBSecurityGroup-Sicherheitsgruppe hinzuzufügen.

Der folgende Codeausschnitt fügt eine entsprechende Ingress-Regel zur AWSEBSecurityGroup-Sicherheitsgruppe hinzu, die Port 443 für allen Datenverkehr für eine Umgebung mit einer einzelnen Instance öffnet:

.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
```

In einer Umgebung mit Load-Balancing in einer [Amazon Virtual Private Cloud](#)-Standardumgebung (Amazon VPC) können Sie diese Richtlinie so ändern, dass nur Datenverkehr vom Load Balancer akzeptiert wird. Ein Beispiel finden Sie unter [Ende-zu-Ende-Verschlüsselung in einer Umgebung mit Elastic Beanstalk-Load-Balancing konfigurieren \(p. 810\)](#).

Plattformen

- [Beenden von HTTPS auf EC2-Instances, auf denen Docker ausgeführt wird \(p. 786\)](#)
- [Beenden von HTTPS auf EC2-Instances, auf denen Go ausgeführt wird \(p. 788\)](#)
- [Beenden von HTTPS auf EC2-Instances, auf denen Java SE ausgeführt wird \(p. 791\)](#)
- [Beenden von HTTPS auf EC2-Instances, auf denen Node.js ausgeführt wird \(p. 793\)](#)
- [Beenden von HTTPS auf EC2-Instances, auf denen PHP ausgeführt wird \(p. 795\)](#)
- [Beenden von HTTPS auf EC2-Instances, auf denen Python ausgeführt wird \(p. 797\)](#)
- [Beenden von HTTPS auf EC2-Instances, auf denen Ruby ausgeführt wird \(p. 800\)](#)
- [Beenden von HTTPS auf EC2-Instances, auf denen Tomcat ausgeführt wird \(p. 804\)](#)
- [HTTPS auf Amazon EC2-Instances mit .NET Core unter Linux terminieren \(p. 807\)](#)
- [Beenden von HTTPS auf Amazon EC2-Instances, auf denen .NET ausgeführt wird \(p. 809\)](#)

Beenden von HTTPS auf EC2-Instances, auf denen Docker ausgeführt wird

Verwenden Sie bei Docker-Containern eine [Konfigurationsdatei \(p. 722\)](#) für die HTTPS-Aktivierung.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Daten für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Quell-Bundle im Verzeichnis `.ebextensions`. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel `files` generiert folgende Dateien auf der Instance:
`/etc/nginx/conf.d/https.conf`

Konfiguriert den nginx-Server. Diese Datei wird mit dem Start des nginx-Service geladen.

`/etc/pki/tls/certs/server.crt`

Damit wird die Zertifikatdatei auf der Instance erstellt. Ersetzen Sie den *Inhalt der Zertifikatdatei* durch den Inhalt Ihres Zertifikats.

Note

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
-----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`

Damit wird die private Schlüsseldatei auf der Instance erstellt. Ersetzen Sie den *Inhalt des privaten Schlüssels* durch den Inhalt des privaten Schlüssels, mit dem Sie die Zertifikatanforderung oder das selbstsignierte Zertifikat erstellt haben.

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

Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter [Sicheres Speichern von privaten Schlüsseln in Amazon S3 \(p. 814\)](#).

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Mit der folgenden Konfigurationsdatei wird die ID der Sicherheitsgruppe mit einer AWS CloudFormation-Funktion (p. 750) abgerufen und eine Regel hinzugefügt.

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
```

Bei einer Umgebung mit Load Balancer konfigurieren Sie den Load Balancer entweder so, dass sicherer Datenverkehr unangetastet durchgelassen wird (p. 813), oder für Verschlüsseln und erneutes Entschlüsseln (p. 810) mit End-to-End-Verschlüsselung.

Beenden von HTTPS auf EC2-Instances, auf denen Go ausgeführt wird

Verwenden Sie bei Go-Containertypen eine [Konfigurationsdatei \(p. 722\)](#) für die HTTPS-Aktivierung sowie eine nginx-Konfigurationsdatei, um den nginx-Server für die Verwendung von HTTPS zu konfigurieren.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Platzhalter für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Quell-Bundle im Verzeichnis .ebextensions. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel Resources aktiviert Port 443 auf der Sicherheitsgruppe, die von der Umgebungs-Instance verwendet wird.
- Der Schlüssel files generiert folgende Dateien auf der Instance:

/etc/pki/tls/certs/server.crt

Damit wird die Zertifikatdatei auf der Instance erstellt. Ersetzen Sie den *Inhalt der Zertifikatdatei* durch den Inhalt Ihres Zertifikats.

Note

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
-----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

Damit wird die private Schlüsseldatei auf der Instance erstellt. Ersetzen Sie den *Inhalt des privaten Schlüssels* durch den Inhalt des privaten Schlüssels, mit dem Sie die Zertifikatanforderung oder das selbstsignierte Zertifikat erstellt haben.

- Mit dem Schlüssel `container_commands` wird der nginx-Server nach Abschluss der Konfiguration neu gestartet, damit der Server die nginx-Konfigurationsdatei lädt.

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

Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter [Sicheres Speichern von privaten Schlüsseln in Amazon S3 \(p. 814\)](#).

Binden Sie folgendes Snippet in eine .conf-Datei ein und platzieren Sie diese im Verzeichnis .ebextensions/nginx/conf.d/ des Quell-Bundles (z. B. .ebextensions/nginx/conf.d/https.conf). Ersetzen Sie [app_port](#) durch die Portnummer, die von der Anwendung überwacht wird. In diesem Beispiel wird der nginx-Server zur Überwachung von Port 443 mit SSL konfiguriert. Weitere Informationen zu diesen Konfigurationsdateien für die Go-Plattform finden Sie unter [Konfigurieren des Reverse-Proxy-Servers \(p. 105\)](#).

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;
    }
}
```

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Mit der folgenden Konfigurationsdatei wird die ID der Sicherheitsgruppe mit einer AWS CloudFormation-Funktion (p. 750) abgerufen und eine Regel hinzugefügt.

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
```

Bei einer Umgebung mit Load Balancer konfigurieren Sie den Load Balancer entweder so, dass sicherer Datenverkehr unangetastet durchgelassen wird (p. 813), oder für Verschlüsseln und erneutes Entschlüsseln (p. 810) mit End-to-End-Verschlüsselung.

Beenden von HTTPS auf EC2-Instances, auf denen Java SE ausgeführt wird

Verwenden Sie bei Java SE-Containertypen eine [.ebextensions-Konfigurationsdatei \(p. 722\)](#) für die HTTPS-Aktivierung sowie eine nginx-Konfigurationsdatei, um den nginx-Server für die Verwendung von HTTPS zu konfigurieren.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Platzhalter für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Verzeichnis `.ebextensions`. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel `files` generiert folgende Dateien auf der Instance:
`/etc/pki/tls/certs/server.crt`

Damit wird die Zertifikatdatei auf der Instance erstellt. Ersetzen Sie den *Inhalt der Zertifikatdatei* durch den Inhalt Ihres Zertifikats.

Note

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
-----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`

Damit wird die private Schlüsseldatei auf der Instance erstellt. Ersetzen Sie den *Inhalt des privaten Schlüssels* durch den Inhalt des privaten Schlüssels, mit dem Sie die Zertifikatanforderung oder das selbstsignierte Zertifikat erstellt haben.

- Mit dem Schlüssel `container_commands` wird der nginx-Server nach Abschluss der Konfiguration neu gestartet, damit der Server die nginx-Konfigurationsdatei lädt.

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

Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter [Sicheres Speichern von privaten Schlüsseln in Amazon S3 \(p. 814\)](#).

Binden Sie folgendes Snippet in eine .conf-Datei ein und platzieren Sie diese im Verzeichnis .ebextensions/nginx/conf.d/ des Quell-Bundles (z. B. .ebextensions/nginx/conf.d/https.conf). Ersetzen Sie **app_port** durch die Portnummer, die von der Anwendung überwacht wird. In diesem Beispiel wird der nginx-Server zur Überwachung von Port 443 mit SSL konfiguriert. Weitere Informationen zu diesen Konfigurationsdateien für die Java SE-Plattform finden Sie unter [Konfigurieren des Reverse-Proxy-Servers \(p. 132\)](#).

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;  
    }  
}
```

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Mit der folgenden Konfigurationsdatei wird die ID der Sicherheitsgruppe mit einer AWS CloudFormation-Funktion (p. 750) abgerufen und eine Regel hinzugefügt.

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
```

Bei einer Umgebung mit Load Balancer konfigurieren Sie den Load Balancer entweder so, dass sicherer Datenverkehr unangetastet durchgelassen wird (p. 813), oder für Verschlüsseln und erneutes Entschlüsseln (p. 810) mit End-to-End-Verschlüsselung.

Beenden von HTTPS auf EC2-Instances, auf denen Node.js ausgeführt wird

In der folgenden Beispielkonfigurationsdatei wird die nginx-Standardkonfiguration erweitert (p. 257), sodass Port 443 überwacht und SSL/TLS-Verbindungen mit öffentlichem Zertifikat und privatem Schlüssel beendet werden.

Wenn Sie Ihre Umgebung für erweiterte Zustandsberichte (p. 824) konfiguriert haben, müssen Sie nginx für die Generierung von Zugriffsprotokollen konfigurieren. Hierzu muss die Auskommentierung des Zeilenblocks unter dem Kommentar mit dem Text `# For enhanced health...` durch Entfernen der führenden #-Zeichen erfolgen.

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 $4;
        #}
        #access_log /var/log/nginx/healthd/application.log.$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;
        }
      }
    
```

```
        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-----
```

Der Schlüssel files generiert folgende Dateien auf der Instance:

/etc/nginx/conf.d/https.conf

Konfiguriert den nginx-Server. Diese Datei wird mit dem Start des nginx-Service geladen.

/etc/pki/tls/certs/server.crt

Damit wird die Zertifikatdatei auf der Instance erstellt. Ersetzen Sie den *Inhalt der Zertifikatdatei* durch den Inhalt Ihres Zertifikats.

Note

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in server.crt auf.

```
-----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

Damit wird die private Schlüsseldatei auf der Instance erstellt. Ersetzen Sie den *Inhalt des privaten Schlüssels* durch den Inhalt des privaten Schlüssels, mit dem Sie die Zertifikatanforderung oder das selbstsignierte Zertifikat erstellt haben.

Note

Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben,

speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter [Sicheres Speichern von privaten Schlüsseln in Amazon S3 \(p. 814\)](#).

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Mit der folgenden Konfigurationsdatei wird die ID der Sicherheitsgruppe mit einer AWS CloudFormation-Funktion (p. 750) abgerufen und eine Regel hinzugefügt.

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
```

Bei einer Umgebung mit Load Balancer konfigurieren Sie den Load Balancer entweder so, dass sicherer Datenverkehr unangetastet durchgelassen wird (p. 813), oder für Verschlüsseln und erneutes Entschlüsseln (p. 810) mit End-to-End-Verschlüsselung.

Beenden von HTTPS auf EC2-Instances, auf denen PHP ausgeführt wird

Verwenden Sie bei PHP-Containertypen eine [Konfigurationsdatei \(p. 722\)](#), um den Apache HTTP-Server für die Verwendung von HTTPS zu aktivieren.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Daten für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Quell-Bundle im Verzeichnis .ebextensions.

Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel packages nutzt "yum" für die Installation von mod24_ssl.
- Der Schlüssel files generiert folgende Dateien auf der Instance:
`/etc/httpd/conf.d/ssl.conf`

Konfiguriert den Apache-Server. Diese Datei wird mit dem Start des Apache-Service geladen.

`/etc/pki/tls/certs/server.crt`

Damit wird die Zertifikatdatei auf der Instance erstellt. Ersetzen Sie den *Inhalt der Zertifikatdatei* durch den Inhalt Ihres Zertifikats.

Note

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
-----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

Damit wird die private Schlüsseldatei auf der Instance erstellt. Ersetzen Sie den *Inhalt des privaten Schlüssels* durch den Inhalt des privaten Schlüssels, mit dem Sie die Zertifikatanforderung oder das selbstsignierte Zertifikat erstellt haben.

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

Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter [Sicheres Speichern von privaten Schlüsseln in Amazon S3 \(p. 814\)](#).

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Mit der folgenden Konfigurationsdatei wird die ID der Sicherheitsgruppe mit einer AWS CloudFormation-Funktion (p. 750) abgerufen und eine Regel hinzugefügt.

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
```

Bei einer Umgebung mit Load Balancer konfigurieren Sie den Load Balancer entweder so, dass sicherer Datenverkehr [unangetastet durchgelassen wird \(p. 813\)](#), oder [für Verschlüsseln und erneutes Entschlüsseln \(p. 810\)](#) mit End-to-End-Verschlüsselung.

Beenden von HTTPS auf EC2-Instances, auf denen Python ausgeführt wird

Für Python-Containertypen mit Apache HTTP-Server mit dem Web Server Gateway Interface (WSGI) verwenden Sie eine [Konfigurationsdatei \(p. 722\)](#), um den Apache HTTP-Server für die Verwendung von HTTPS zu aktivieren.

Fügen Sie der [Konfigurationsdatei \(p. 722\)](#) das folgende Snippet hinzu. Folgen Sie den Anweisungen, um die Daten für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im .ebextensions-Verzeichnis des Quell-Bundles. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel `packages` nutzt "yum" für die Installation von `mod24_ssl`.
- Der Schlüssel `files` generiert folgende Dateien auf der Instance:
`/etc/httpd/conf.d/ssl.conf`

Konfiguriert den Apache-Server. Wenn Ihre Anwendung nicht den Namen `application.py` hat, ersetzen Sie den markierten Text im Wert für `WSGIScriptAlias` durch den lokalen Pfad zu Ihrer

Anwendung. Beispiel: Eine Django-Anwendung kann sich zum Beispiel unter `django/wsgi.py` befinden. Der Speicherort sollte mit dem Wert der `WSGIPath`-Option, die Sie für Ihre Umgebung festlegen, übereinstimmen.

Abhängig von den Anforderungen Ihrer Anwendung müssen Sie auch andere Verzeichnisse zum `python-path`-Parameter hinzufügen.

`/etc/pki/tls/certs/server.crt`

Damit wird die Zertifikatdatei auf der Instance erstellt. Ersetzen Sie den *Inhalt der Zertifikatdatei* durch den Inhalt Ihres Zertifikats.

Note

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
-----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`

Damit wird die private Schlüsseldatei auf der Instance erstellt. Ersetzen Sie den *Inhalt des privaten Schlüssels* durch den Inhalt des privaten Schlüssels, mit dem Sie die Zertifikatanforderung oder das selbstsignierte Zertifikat erstellt haben.

- Mit dem Schlüssel `container_commands` wird der `httpd`-Service nach der abgeschlossenen Konfiguration beendet, sodass der Service die neue `https.conf`-Datei und das entsprechende Zertifikat verwendet.

Note

Das Beispiel funktioniert nur in Umgebungen, die die [Python \(p. 355\)](#)-Plattform verwenden.

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

Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter [Sicheres Speichern von privaten Schlüsseln in Amazon S3 \(p. 814\)](#).

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Mit der folgenden Konfigurationsdatei wird die ID

der Sicherheitsgruppe mit einer AWS CloudFormation-Funktion (p. 750) abgerufen und eine Regel hinzugefügt.

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
```

Bei einer Umgebung mit Load Balancer konfigurieren Sie den Load Balancer entweder so, dass sicherer Datenverkehr unangetastet durchgelassen wird (p. 813), oder für Verschlüsseln und erneutes Entschlüsseln (p. 810) mit End-to-End-Verschlüsselung.

Beenden von HTTPS auf EC2-Instances, auf denen Ruby ausgeführt wird

Bei Ruby-Containertypen hängt die Art und Weise der HTTPS-Aktivierung vom eingesetzten Anwendungsserver ab.

Themen

- Konfigurieren von HTTPS für Ruby mit Puma (p. 800)
- Konfigurieren von HTTPS für Ruby mit Passenger (p. 802)

Konfigurieren von HTTPS für Ruby mit Puma

Verwenden Sie bei Ruby-Containertypen, die einen Puma-Anwendungsserver nutzen, eine Konfigurationsdatei (p. 722) für die HTTPS-Aktivierung.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Daten für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Quell-Bundle im Verzeichnis .ebextensions. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel files generiert folgende Dateien auf der Instance:

/etc/nginx/conf.d/https.conf

Konfiguriert den nginx-Server. Diese Datei wird mit dem Start des nginx-Service geladen.

/etc/pki/tls/certs/server.crt

Damit wird die Zertifikatdatei auf der Instance erstellt. Ersetzen Sie den *Inhalt der Zertifikatdatei* durch den Inhalt Ihres Zertifikats.

Note

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in server.crt auf.

```
-----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

Damit wird die private Schlüsseldatei auf der Instance erstellt. Ersetzen Sie den *Inhalt des privaten Schlüssels* durch den Inhalt des privaten Schlüssels, mit dem Sie die Zertifikatanforderung oder das selbstsignierte Zertifikat erstellt haben.

- Mit dem Schlüssel `container_commands` wird der nginx-Server nach Abschluss der Konfiguration neu gestartet, damit der Server die neue `https.conf`-Datei verwendet.

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

Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter [Sicheres Speichern von privaten Schlüsseln in Amazon S3 \(p. 814\)](#).

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Mit der folgenden Konfigurationsdatei wird die ID der Sicherheitsgruppe mit einer AWS CloudFormation-Funktion (p. 750) abgerufen und eine Regel hinzugefügt.

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
```

Bei einer Umgebung mit Load Balancer konfigurieren Sie den Load Balancer entweder so, dass sicherer Datenverkehr unangetastet durchgelassen wird (p. 813), oder [für Verschlüsseln und erneutes Entschlüsseln \(p. 810\)](#) mit End-to-End-Verschlüsselung.

Konfigurieren von HTTPS für Ruby mit Passenger

Verwenden Sie bei Ruby-Containertypen, die einen Passenger-Anwendungsserver nutzen, sowohl eine Konfigurationsdatei als auch eine JSON-Datei für die HTTPS-Aktivierung.

So konfigurieren Sie HTTPS für Ruby mit Passenger

1. Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Daten für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Quell-Bundle im Verzeichnis `.ebextensions`. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:
 - Der Schlüssel `files` generiert folgende Dateien auf der Instance:

/etc/pki/tls/certs/server.crt

Damit wird die Zertifikatdatei auf der Instance erstellt. Ersetzen Sie den *Inhalt der Zertifikatdatei* durch den Inhalt Ihres Zertifikats.

Note

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in server.crt auf.

```
-----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

Damit wird die private Schlüsseldatei auf der Instance erstellt. Ersetzen Sie den *Inhalt des privaten Schlüssels* durch den Inhalt des privaten Schlüssels, mit dem Sie die Zertifikatanforderung oder das selbstsignierte Zertifikat erstellt haben.

Example .Ebextensions-Snippet zum Konfigurieren von HTTPS für Ruby mit 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

Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter [Sicheres Speichern von privaten Schlüsseln in Amazon S3 \(p. 814\)](#).

2. Erstellen Sie eine Textdatei und fügen Sie die folgenden JSON-Daten ein. Speichern Sie die Datei im Stammverzeichnis des Quell-Bundles mit dem Namen `passenger-standalone.json`. Mit dieser JSON-Datei wird Passenger für die Verwendung von HTTPS konfiguriert.

Important

Die JSON-Datei darf keine Markierung der Bytereihenfolge enthalten. Sollte das der Fall sein, kann die Passenger-JSON-Bibliothek die Datei nicht korrekt lesen und der Passenger-Service wird nicht gestartet.

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"  
}
```

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Mit der folgenden Konfigurationsdatei wird die ID der Sicherheitsgruppe mit einer AWS CloudFormation-[Funktion \(p. 750\)](#) abgerufen und eine Regel hinzugefügt.

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
```

Bei einer Umgebung mit Load Balancer konfigurieren Sie den Load Balancer entweder so, dass sicherer Datenverkehr [unaufgetastet durchgelassen wird \(p. 813\)](#), oder [für Verschlüsseln und erneutes Entschlüsseln \(p. 810\)](#) mit End-to-End-Verschlüsselung.

Beenden von HTTPS auf EC2-Instances, auf denen Tomcat ausgeführt wird

Verwenden Sie bei Tomcat-Containertypen eine [Konfigurationsdatei \(p. 722\)](#), um den Apache HTTP-Server für die Verwendung von HTTPS zu aktivieren, wenn dieser als Reverse-Proxy-Server für Tomcat eingesetzt wird.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Daten für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Quell-Bundle im Verzeichnis `.ebextensions`. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel `files` generiert folgende Dateien auf der Instance:
`/etc/pki/tls/certs/server.crt`

Damit wird die Zertifikatdatei auf der Instance erstellt. Ersetzen Sie den `Inhalt der Zertifikatdatei` durch den Inhalt Ihres Zertifikats.

Note

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und

stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

/etc/pki/tls/certs/server.key

Damit wird die private Schlüsseldatei auf der Instance erstellt. Ersetzen Sie den **Inhalt des privaten Schlüssels** durch den Inhalt des privaten Schlüssels, mit dem Sie die Zertifikatanforderung oder das selbstsignierte Zertifikat erstellt haben.

/opt/elasticbeanstalk/hooks/appdeploy/post/99_start_httpd.sh

Erstellt nach der Bereitstellung ein Hook-Skript für den Neustart des httpd-Service.

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
```

Sie müssen auch den Proxy-Server Ihrer Umgebung für die Überwachung von Port 443 konfigurieren. Mit der folgenden Apache 2.4-Konfiguration wird ein Listener auf Port 443 hinzugefügt: Weitere Informationen hierzu finden Sie unter [Konfigurieren des Proxy-Servers Ihrer Tomcat-Umgebung \(p. 124\)](#).

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>
```

Möglicherweise erhalten Sie von Ihrem Zertifikatanbieter auch Zwischenzertifikate, die Sie für eine bessere Kompatibilität mit mobilen Clients installieren können. Konfigurieren Sie Apache mit einem Zwischenzertifizierungsstellen-Bundle, indem Sie Folgendes zur SSL-Konfigurationsdatei hinzufügen (siehe [Erweitern und Überschreiben der Apache-Standardkonfiguration \(p. 126\)](#) für die Platzierung):

- Geben Sie in den `ssl.conf`-Dateiinhalten die Chain-Datei an:

```
SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"
SSLCertificateChainFile "/etc/pki/tls/certs/gd_bundle.crt"
SSLCipherSuite ECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
```

- Fügen Sie dem `SSL files` einen neuen Eintrag mit den Inhalten der Zwischenzertifikate hinzu:

```
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

Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter [Sicheres Speichern von privaten Schlüsseln in Amazon S3 \(p. 814\)](#).

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Mit der folgenden Konfigurationsdatei wird die ID der Sicherheitsgruppe mit einer AWS CloudFormation-Funktion (p. 750) abgerufen und eine Regel hinzugefügt.

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
```

Bei einer Umgebung mit Load Balancer konfigurieren Sie den Load Balancer entweder so, dass sicherer Datenverkehr unangetastet durchgelassen wird (p. 813), oder für Verschlüsseln und erneutes Entschlüsseln (p. 810) mit End-to-End-Verschlüsselung.

HTTPS auf Amazon EC2-Instances mit .NET Core unter Linux terminieren

Sie verwenden für .NET Core on Linux-Containertypen eine [.ebextensions-Konfigurationsdatei \(p. 722\)](#), um HTTPS zu aktivieren, und eine nginx-Konfigurationsdatei, um den nginx-Server für die Verwendung von HTTPS zu konfigurieren.

Fügen Sie der Konfigurationsdatei das folgende Snippet hinzu, folgen Sie den Anweisungen, um die Platzhalter für Zertifikat und privaten Schlüssel zu ersetzen, und speichern Sie dies im Verzeichnis `.ebextensions`. Von der Konfigurationsdatei werden folgende Schritte ausgeführt:

- Der Schlüssel `files` generiert folgende Dateien auf der Instance:

`/etc/pki/tls/certs/server.crt`

Damit wird die Zertifikatdatei auf der Instance erstellt. Ersetzen Sie den *Inhalt der Zertifikatdatei* durch den Inhalt Ihres Zertifikats.

Note

Für YAML sind konsistente Einrückungen erforderlich. Wählen Sie die entsprechende Einrückungsebene aus, wenn Sie Inhalte in einer Beispielkonfigurationsdatei ersetzen, und stellen Sie sicher, dass Ihr Texteditor Leerzeichen statt Tabulatorzeichen zum Einrücken verwendet.

Wenn Zwischenzertifikate vorhanden sind, nehmen Sie sie nach Ihrem Site-Zertifikat in `server.crt` auf.

```
-----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`

Damit wird die private Schlüsseldatei auf der Instance erstellt. Ersetzen Sie den *Inhalt des privaten Schlüssels* durch den Inhalt des privaten Schlüssels, mit dem Sie die Zertifikatanforderung oder das selbstsignierte Zertifikat erstellt haben.

- Mit dem Schlüssel `container_commands` wird der nginx-Server nach Abschluss der Konfiguration neu gestartet, damit der Server die nginx-Konfigurationsdatei lädt.

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

Führen Sie möglichst keinen Commit für eine Konfigurationsdatei, die Ihren privaten Schlüssel enthält, an die Quellüberwachung durch. Nachdem Sie die Konfiguration getestet haben, speichern Sie Ihren privaten Schlüssel in Amazon S3 und ändern Sie die Konfiguration, um sie während der Bereitstellung herunterzuladen. Detaillierte Anweisungen finden Sie unter [Sicheres Speichern von privaten Schlüsseln in Amazon S3 \(p. 814\)](#).

Fügen Sie Folgendes in eine Datei mit der Erweiterung .conf im Verzeichnis .ebextensions/nginx/conf.d/ Ihres Quellpaket ein (z. B. .ebextensions/nginx/conf.d/https.conf). Ersetzen Sie [app_port](#) durch die Portnummer, die von der Anwendung überwacht wird. In diesem Beispiel wird der nginx-Server zur Überwachung von Port 443 mit SSL konfiguriert. Weitere Informationen zu diesen Konfigurationsdateien auf der .NET Core on Linux-Plattform finden Sie unter [the section called “Proxy-Server” \(p. 163\)](#).

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;  
    }  
}
```

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Mit der folgenden Konfigurationsdatei wird die ID der Sicherheitsgruppe mit einer AWS CloudFormation-Funktion (p. 750) abgerufen und eine Regel hinzugefügt.

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
```

Bei einer Umgebung mit Load Balancer konfigurieren Sie den Load Balancer entweder so, dass sicherer Datenverkehr [unangetastet durchgelassen wird](#) (p. 813), oder [für Verschlüsseln und erneutes Entschlüsseln](#) (p. 810) mit End-to-End-Verschlüsselung.

Beenden von HTTPS auf Amazon EC2-Instances, auf denen .NET ausgeführt wird

Die folgende [Konfigurationsdatei](#) (p. 722) erstellt ein Windows PowerShell-Skript und führt es aus. Dieses übernimmt die folgenden Aufgaben:

- Prüft auf ein vorhandenes HTTPS-Zertifikat für eine Bindung mit Port 443
- Ruft das [PFX-Zertifikat](#) (p. 780) und Passwort aus einem Amazon S3-Bucket ab

Note

Fügen Sie eine `AmazonS3ReadOnlyAccess`-Richtlinie zu `aws-elasticbeanstalk-ec2-role` hinzu, um auf das SSL-Zertifikat und Passwortdateien im Amazon S3-Bucket zugreifen zu können.

- Installiert das Zertifikat
- Bindet das Zertifikat an Port 443

Note

Um den HTTP-Endpunkt (Port 80) zu entfernen, fügen Sie den `Remove-WebBinding`-Befehl unter dem Abschnitt Remove the HTTP binding des Beispiels hinzu.

Example .ebextensions/https-instance-dotnet.config

```
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
      $pwdkey = "password.txt"      ## S3 object key for a text file containing the
      certificate's password
      ##

      # Set variables
      $certfile = "C:\\cert.pfx"
      $pwdfile = "C:\\certs\\pwdcontent"
      Read-S3Object -BucketName $bucket -Key $pwdkey -File $pwdfile
      $pwd = Get-Content $pwdfile -Raw

      # 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
```

In einer einzelnen Instance-Umgebung müssen Sie außerdem die Sicherheitsgruppe der Instance ändern, damit Datenverkehr über Port 443 zugelassen wird. Mit der folgenden Konfigurationsdatei wird die ID der Sicherheitsgruppe mit einer AWS CloudFormation-Funktion (p. 750) abgerufen und eine Regel hinzugefügt.

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
```

Bei einer Umgebung mit Load Balancer konfigurieren Sie den Load Balancer entweder so, dass sicherer Datenverkehr unangetastet durchgelassen wird (p. 813), oder für Verschlüsseln und erneutes Entschlüsseln (p. 810) mit End-to-End-Verschlüsselung.

Ende-zu-Ende-Verschlüsselung in einer Umgebung mit Elastic Beanstalk-Load-Balancing konfigurieren

Das Beenden sicherer Verbindungen am Load Balancer und das Verwenden von HTTP am Backend kann ausreichend für Ihre Anwendung sein. Der Netzwerkverkehr zwischen AWS-Ressourcen kann nicht von Instances überwacht werden, die nicht Teil der Verbindung sind, selbst wenn sie im selben Konto ausgeführt werden.

Wenn Sie jedoch eine Anwendung entwickeln, die strenge externe Vorschriften einhalten muss, müssen Sie ggf. alle Netzwerkverbindungen sichern. Sie können die Elastic Beanstalk-Konsole oder Konfigurationsdateien (p. 722) verwenden, damit der Load Balancer der Elastic Beanstalk-Umgebung

eine sichere Verbindung zu Backend-Instances herstellt, um diese Anforderungen zu erfüllen. Die folgenden Schritte konzentrieren sich auf Konfigurationsdateien.

Zuerst [fügen Sie einen sicheren Listener zu Ihrem Load Balancer hinzu](#) (p. 783), falls Sie dies noch nicht getan haben.

Sie müssen auch die Instances in Ihrer Umgebung so konfigurieren, dass sie auf dem sicheren Port überwachen und HTTPS-Verbindungen beenden. Die Konfiguration variiert je nach Plattform. Anweisungen finden Sie unter [Konfigurieren Ihrer Anwendung, um HTTPS-Verbindungen auf der Instance zu beenden](#) (p. 785). Sie können ohne Probleme ein [selbstsigniertes Zertifikat](#) (p. 780) für die EC2-Instances verwenden.

Anschließend konfigurieren Sie den Listener für die Weiterleitung von Datenverkehr mithilfe von HTTPS auf den sicheren Port, der von Ihrer Anwendung verwendet wird. Verwenden Sie eine der folgenden Konfigurationsdateien, je nach Typ des Load Balancer, den Ihre Umgebung verwendet.

.ebextensions/https-reencrypt-clb.config

Verwenden Sie diese Konfigurationsdatei mit einem Classic Load Balancer. Neben der Konfiguration des Load Balancer ändert die Konfigurationsdatei auch die standardmäßige Zustandsprüfung, sodass Port 443 und HTTPS verwendet werden, um sicherzustellen, dass der Load Balancer in der Lage ist, eine sichere Verbindung herzustellen.

```
option_settings:
  aws:elb:listener:443:
    InstancePort: 443
    InstanceProtocol: HTTPS
  aws:elasticbeanstalk:application:
    Application Healthcheck URL: HTTPS:443/
```

.ebextensions/https-reencrypt-alb.config

Verwenden Sie diese Konfigurationsdatei mit einem 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

Verwenden Sie diese Konfigurationsdatei mit einem Network Load Balancer.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
```

Die Option `DefaultProcess` hat diesen Namen wegen Application Load Balancers, die Nicht-Standard-Listener auf demselben Port für Datenverkehr auf bestimmten Pfaden haben können (siehe [Application Load Balancer](#) (p. 566) für weitere Informationen). Bei einem Network Load Balancer gibt die Option den einzigen Zielprozess für diesen Listener an.

In diesem Beispiel haben wir den Prozess `https` genannt, da er sicheren (HTTPS) Datenverkehr überwacht. Der Listener sendet den Datenverkehr über das TCP-Protokoll an den Prozess auf dem

festgelegten Port, da ein Network Load Balancer nur mit TCP funktioniert. Das ist in Ordnung, da der Netzwerdatenverkehr für HTTP und HTTPS zusätzlich zu TCP implementiert ist.

Note

Die EB CLI und Elastic Beanstalk-Konsole wenden die empfohlenen Werte für die vorhergehenden Optionen an. Diese Einstellungen müssen entfernt werden, wenn Sie Konfigurationsdateien zum Konfigurieren dieser Einstellungen verwenden möchten. Details dazu finden Sie unter [Empfohlene Werte \(p. 648\)](#).

In der nächsten Aufgabe müssen Sie die Sicherheitsgruppe des Load Balancer so ändern, dass Datenverkehr zulässig ist. Abhängig von der [Amazon Virtual Private Cloud](#) (Amazon VPC) (die Standard-VPC oder eine benutzerdefinierte VPC), in der Sie Ihre Umgebung starten, variiert die Sicherheitsgruppe des Load Balancers. In einer Standard-VPC bietet Elastic Load Balancing eine Standard-Sicherheitsgruppe, die alle Load Balancer verwenden können. In einer von Ihnen erstellten Amazon VPC erstellt Elastic Beanstalk eine Sicherheitsgruppe, die der Load Balancer verwenden kann.

Zur Unterstützung beider Szenarien können Sie eine Sicherheitsgruppe erstellen und Elastic Beanstalk anweisen, diese zu verwenden. Die folgende Konfigurationsdatei erstellt eine Sicherheitsgruppe und fügt Sie an den Load Balancer an.

.ebextensions/https-lbsecuritygroup.config

```
option_settings:
    # Use the custom security group for the load balancer
    aws:elb:loadbalancer:
        SecurityGroups: `'{ "Ref" : "loadbalancerssg" }`'
        ManagedSecurityGroup: `'{ "Ref" : "loadbalancerssg" }`'

Resources:
    loadbalancerssg:
        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
```

Ersetzen Sie den markierten Text mit der Standard- oder benutzerdefinierten VPC-ID. Das vorherige Beispiel enthält ein- und ausgehenden Datenverkehr über Port 80, um HTTP-Verbindungen zuzulassen. Sie können diese Eigenschaften entfernen, wenn Sie nur sichere Verbindungen zulassen möchten.

Fügen Sie nun Eingangs- und Ausgangsregeln hinzu, mit denen die Kommunikation über Port 443 zwischen der Sicherheitsgruppe des Load Balancer und der Sicherheitsgruppe der Instances möglich ist.

.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" : ["loadbalancerssg", "GroupId"]}
# Add 443-outbound to load balancer security group (loadbalancerssg)
httpsToBackendInstances:
Type: AWS::EC2::SecurityGroupEgress
Properties:
 GroupId: {"Fn::GetAtt" : ["loadbalancerssg", "GroupId"]}
  IpProtocol: tcp
  ToPort: 443
  FromPort: 443
  DestinationSecurityGroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
```

Wenn Sie dies separat von der Erstellung der Sicherheitsgruppe durchführen, können Sie die Quell- und Zielsicherheitsgruppen beschränken, ohne einen Zirkelbezug zu erstellen.

Wenn alle vorherigen Aufgaben durchgeführt wurden, stellt der Load Balancer eine sichere Verbindung zu Ihren Backend-Instances über HTTPS her. Der Load Balancer berücksichtigt nicht, ob das Zertifikat der Instance selbstsigniert oder von einer vertrauenswürdigen Zertifizierungsstelle ausgestellt wurde, und akzeptiert alle Zertifikate, die im vorgelegt werden.

Sie können dieses Verhalten ändern, indem Sie Richtlinien zum Load Balancer hinzufügen, die besagen, dass nur einem bestimmten Zertifikat vertraut werden darf. Mit der folgenden Konfigurationsdatei werden zwei Richtlinien erstellt. Eine Richtlinie gibt ein öffentliches Zertifikat an und die andere weist den Load Balancer an, nur diesem Zertifikat für Verbindungen zum Instance-Port 443 zu vertrauen

.ebextensions/https-backendauth.config

Ersetzen Sie den markierten Text mit den Inhalten des öffentlichen Zertifikats Ihrer EC2-Instanz.

Konfigurieren des Load Balancers der Umgebung für TCP-Pass-Through

Wenn der Load Balancer der AWS Elastic Beanstalk-Umgebung den HTTPS-Datenverkehr nicht entschlüsseln soll, können Sie den sicheren Listener so konfigurieren, dass Anforderungen unverändert an Backend-Instances weitergeleitet werden.

Zuerst konfigurieren Sie die EC2-Instances der Umgebung, um HTTPS-Verbindungen zu beenden (p. 785). Testen Sie die Konfiguration in einer Umgebung mit einer einzelnen Instance und stellen Sie die ordnungsgemäße Funktionsweise sicher, bevor Sie den Load Balancer hinzufügen.

Fügen Sie eine [Konfigurationsdatei \(p. 722\)](#) zum Projekt hinzu, um einen Listener auf Port 443 zu konfigurieren, der TCP-Pakete unverändert an Port 443 auf den Backend-Instances weitergibt:

.ebextensions/https-lb-passthrough.config

```
option_settings:  
  aws:elb:listener:443:  
    ListenerProtocol: TCP  
    InstancePort: 443  
    InstanceProtocol: TCP
```

In einer Standard-[Amazon Virtual Private Cloud](#) (Amazon VPC) muss der Sicherheitsgruppe der Instances zudem eine Regel hinzugefügt werden, damit eingehender Verkehr vom Load Balancer über Port 443 zulässig ist:

.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"] }
```

In einer benutzerdefinierten VPC aktualisiert Elastic Beanstalk die Konfiguration der Sicherheitsgruppe für Sie.

Sicheres Speichern von privaten Schlüsseln in Amazon S3

Der private Schlüssel, den Sie zum Signieren des öffentlichen Zertifikats verwenden, ist geheim und sollte nicht an den Quellcode übergeben werden. Sie können vermeiden, dass private Schlüssel in Konfigurationsdateien gespeichert werden, indem Sie sie in Amazon S3 hochladen und Elastic Beanstalk so konfigurieren, dass die Datei während der Anwendungsbereitstellung aus Amazon S3 heruntergeladen wird.

Das folgende Beispiel veranschaulicht, wie Sie in den Abschnitten [Resources \(p. 745\)](#) und [files \(p. 729\)](#) einer [Konfigurationsdatei \(p. 722\)](#) eine Datei mit einem privaten Schlüssel aus einem Amazon S3-Bucket herunterladen.

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
```

Ersetzen Sie den Bucket-Namen und die URL im Beispiel durch Ihre eigenen Angaben. Mit dem ersten Eintrag der Datei wird eine Authentifizierungsmethode mit dem Namen S3Auth zu den Metadaten der Auto Scaling-Gruppe für diese Umgebung hinzugefügt. Falls Sie ein benutzerdefiniertes [Instance-Profil \(p. 22\)](#) für die Umgebung konfiguriert haben, wird dieses verwendet. Andernfalls wird der Standardwert von aws-elasticbeanstalk-ec2-role übernommen. Das Instance-Standardprofil besitzt die Berechtigung zum Lesen der Speicher-Buckets von Elastic Beanstalk. Bei Verwendung eines anderen Buckets [fügen Sie Berechtigungen zum Instance-Profil hinzu \(p. 908\)](#).

Vom zweiten Eintrag wird die Authentifizierungsmethode S3Auth verwendet, um den privaten Schlüssel über die angegebene URL herunterzuladen und unter /etc/pki/tls/certs/server.key zu speichern. Der Proxy-Server kann dann den privaten Schlüssel an diesem Speicherort auslesen und [HTTPS-Verbindungen zur Instance beenden \(p. 785\)](#).

Das Instance-Profil, das Sie den EC2-Instances der Umgebung zugeordnet haben, muss über die Berechtigung zum Lesen des Schlüsselobjekts aus dem angegebenen Bucket verfügen. [Stellen Sie sicher, dass das Instance-Profil über die Berechtigung \(p. 907\)](#) zum Lesen des Objekts in IAM verfügt und dass die Berechtigungen für den Bucket und das Objekt das Instance-Profil nicht ausschließen.

So zeigen Sie die Berechtigungen eines Buckets an

1. Öffnen Sie die [Amazon S3-Managementkonsole](#).
2. Wählen Sie einen Bucket aus.
3. Wählen Sie Properties (Eigenschaften) und dann Permissions (Berechtigungen) aus.
4. Vergewissern Sie sich, dass Ihr Konto für diesen Bucket berechtigt ist und Leseberechtigung hat.
5. Wenn eine Bucket-Richtlinie angefügt ist, wählen Sie Bucket policy (Bucket-Richtlinie) aus, um die dem Bucket zugewiesenen Berechtigungen anzuzeigen.

Konfigurieren der Umleitung von HTTP zu HTTPS

In [Konfigurieren von HTTPS für Elastic Beanstalk-Umgebung \(p. 778\)](#) und in den entsprechenden Unterthemen behandeln wir die Konfiguration Ihrer Elastic Beanstalk-Umgebung für die Verwendung von HTTPS zur Sicherstellung der Verschlüsselung des Datenverkehrs in Ihrer Anwendung. In diesem Thema wird beschrieben, wie Sie den von Endbenutzern weiterhin initiierten HTTP-Datenverkehr in Ihrer Anwendung ganz einfach verarbeiten können. Konfigurieren Sie hierzu die Umleitung von HTTP zu HTTPS (manchmal auch bezeichnet als Erzwingen von HTTPS).

Um die Umleitung zu konfigurieren, müssen Sie zunächst Ihre Umgebung für die Verarbeitung von HTTPS-Datenverkehr konfigurieren. Dann leiten Sie HTTP-Datenverkehr zu HTTPS um. Diese beiden Schritte werden in den folgenden Unterabschnitten erläutert.

Umgebung für die Verarbeitung von HTTPS-Datenverkehr konfigurieren

Führen Sie je nach Load-Balancing-Konfiguration in Ihrer Umgebung einen der folgenden Schritte aus:

- Umgebung mit Lastausgleich – [Konfiguriert den Load Balancer zum Beenden von HTTPS \(p. 783\)](#).

- Umgebung mit einer einzelnen Instance – [Konfiguriert Ihre Anwendung zum Beenden von HTTPS-Verbindungen auf der Instance \(p. 785\)](#). Diese Konfiguration ist abhängig von Ihrer Umgebungsplattform.

HTTP-Datenverkehr an HTTPS umleiten

Sie können entweder die Webserver in den Instances Ihrer Umgebung oder den Application Load Balancer der Umgebung so konfigurieren, dass der HTTP-Datenverkehr an HTTPS umgeleitet wird. Führen Sie eine der folgenden Aufgaben aus:

- Konfigurieren Sie Instance-Webserver – Diese Methode funktioniert in jeder Webserverumgebung. Konfigurieren Sie Webserver auf Ihren Amazon Elastic Compute Cloud-Instance (Amazon EC2) so, dass sie auf HTTP-Verkehr mit einem HTTP-Umleitungs-Antwortstatus antworten. Diese Konfiguration ist abhängig von Ihrer Umgebungsplattform. Suchen Sie den Ordner für Ihre Plattform in der [https-redirect](#)-Sammlung auf GitHub und verwenden Sie die Beispiel-Konfigurationsdatei in diesem Ordner.

Wenn Ihre Umgebung [Elastic Load Balancing-Zustandsprüfungen \(p. 822\)](#) verwendet, erwartet der Load Balancer, dass die HTTP-Zustandsprüfungs-Nachrichten von einer funktionierenden Instance mit einer 200 OK-HTTP-Antwort beantwortet werden. Aus diesem Grund sollte Ihr Webserver diese Nachrichten nicht an HTTPS umleiten. In den Beispiel-Konfigurationsdateien in [https-redirect](#) wird diese Anforderung korrekt erfüllt.

- Load Balancer konfigurieren – Diese Methode funktioniert, wenn Sie eine Umgebung mit Load Balancing haben, die einen [Application Load Balancer \(p. 566\)](#) verwendet. Application Load Balancer kann Umleitungsantworten senden, wenn HTTP-Datenverkehr hereinkommt. In diesem Fall müssen Sie die Umleitung für die Instances Ihrer Umgebung nicht konfigurieren. Wir haben zwei Beispielkonfigurationsdateien auf GitHub, die zeigen, wie Application Load Balancer für die Umleitung konfiguriert werden kann. Die Konfigurationsdatei [alb-http-to-https-redirection-full.config](#) erstellt einen HTTPS-Listener an Port 443 und ändert den Standard-Port 80-Listener, um eingehenden HTTP-Datenverkehr an HTTPS umzuleiten. Die Konfigurationsdatei [alb-http-to-https-redirection.config](#) erwartet, dass der 443-Listener definiert ist (Sie können die Standard-Namespace der Elastic Beanstalk-Konfiguration oder die Elastic Beanstalk-Konsole verwenden). Anschließend sorgt sie dafür, dass der Port 80-Listener für die Umleitung geändert wird.

Überwachen einer Umgebung

Bei Ausführung einer Produktionswebsite ist es wichtig zu wissen, dass die Anwendung verfügbar ist und auf Anforderungen reagiert. Für die Überwachung der Reaktionsfähigkeit der Anwendung können Sie die Elastic Beanstalk-Funktionen nutzen, mit denen Sie Anwendungsstatistiken überwachen und Warnungen erstellen können, die bei einer Überschreitung der Schwellenwerte ausgelöst werden.

Themen

- [Überwachen des Umgebungszustands in der AWS Managementkonsole \(p. 817\)](#)
- [Grundlegende Zustandsberichte \(p. 821\)](#)
- [Erweiterte Zustandsberichte und Überwachung \(p. 824\)](#)
- [Verwalten von Alarmen \(p. 861\)](#)
- [Ereignis-Stream einer Elastic Beanstalk-Umgebung anzeigen \(p. 864\)](#)
- [Auflisten von Server-Instances/Verbinden mit Server-Instances \(p. 866\)](#)
- [Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen \(p. 869\)](#)

Überwachen des Umgebungszustands in der AWS Managementkonsole

Sie können über die Elastic Beanstalk-Konsole auf operative Informationen zu Ihrer Anwendung zugreifen. Umgebungsstatus und Anwendungszustand werden in der Konsole übersichtlich dargestellt. Auf der Seite Environments (Umgebungen) der Konsole und auf der Seite jeder Anwendung werden die Umgebungen in der Liste farbcodiert, um den Status anzuzeigen.

So überwachen Sie eine Umgebung in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Monitoring (Überwachung) aus.

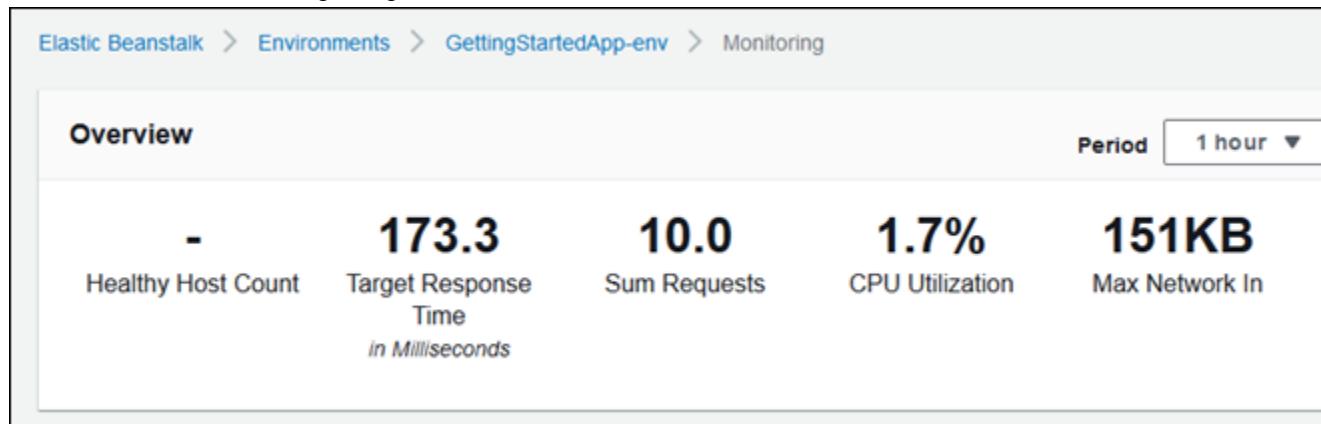
Auf der Seite "Monitoring" werden allgemeine Statistikwerte zur Umgebung angegeben, z. B. CPU-Auslastung und durchschnittliche Latenz. Neben diesen allgemeinen Statistikdaten können Sie Überwachungsdiagramme mit der Ressourcennutzung im Verlauf der Zeit aufrufen. Klicken Sie auf ein Diagramm, um detailliertere Informationen zu sehen.

Note

Standardmäßig sind nur CloudWatch-Metriken aktiviert. Sie geben Daten in Abständen von fünf Minuten zurück. Sie können noch genauere CloudWatch-Metriken in 1-Minuten-Intervallen aktivieren, indem Sie die Konfigurationseinstellungen der Umgebung anpassen.

Übersicht

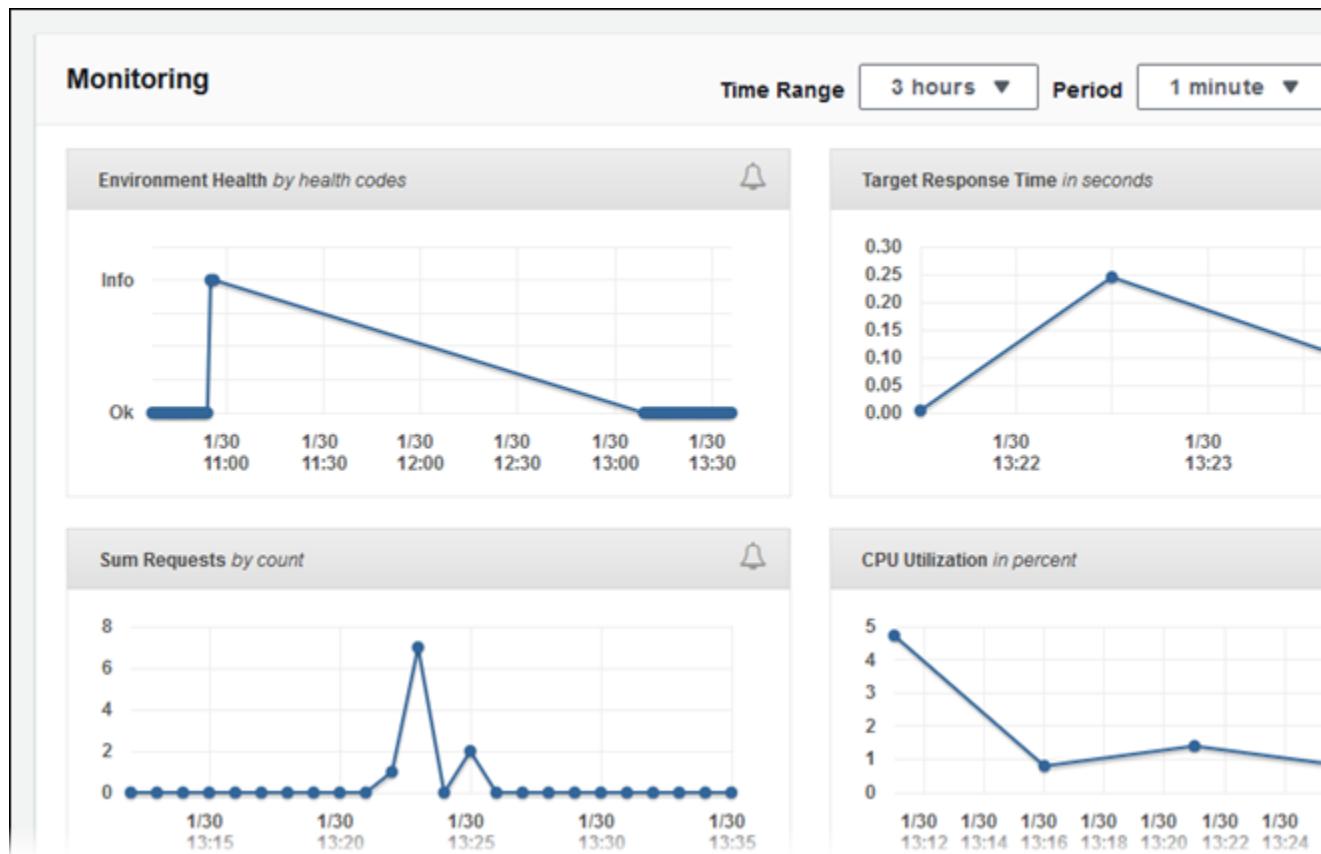
Eine Übersicht über den Umgebungszustand finden Sie oben auf dem Bildschirm.



Im Übersichtsbereich wird eine anpassbare Zusammenfassung der Aktivitäten in Ihrer Umgebung über einen bestimmten Zeitraum angezeigt. Wählen Sie das Drop-down-Menü Period (Zeitraum) und wählen Sie eine Zeitspanne aus, um Informationen für einen Zeitraum zwischen einer Minute und einem Tag anzuzeigen.

Überwachungsdiagramme

Unter der Übersicht werden Diagramme mit Daten zum allgemeinen Umgebungszustand für einen bestimmten Zeitraum angezeigt. Wählen Sie das Drop-down-Menü Period (Zeitraum) und wählen Sie eine Zeitspanne aus, um die Zeit zwischen den beiden Plot-Punkten auf einen Zeitraum zwischen einer Minute und einem Tag festzulegen. Wählen Sie das Drop-down-Menü Time Range (Zeitbereich) und wählen Sie eine Zeitspanne aus, um die Zeitachse des Diagramms auf einen Zeitraum zwischen drei Stunden und zwei Wochen festzulegen.



Anpassen der Überwachungskonsole

Wählen Sie Edit (Bearbeiten) neben einem der Überwachungsfenster, um die angezeigten Informationen anzupassen.

The screenshot shows the AWS Elastic Beanstalk Overview page. At the top, there's a summary section with four cards: 'Healthy Host Count' (173.3), 'Target Response Time in Milliseconds' (10.0), 'Sum Requests' (0.8%), and 'CPU Utilization' (0.8%). Below this is a large empty space labeled 'Add Overview'. In this space, there's a form to add a new metric. It includes fields for 'Resource' (set to 'AWSEBV2LoadBalancer'), 'CloudWatch metric' (with a 'Refresh' button), 'Statistic' (with a dropdown menu), and 'Description' (an empty input field). A note at the bottom says 'Added "Avg ActiveConnectionCount." Click Save to preserve your config'.

Klicken Sie zum Entfernen vorhandener Elemente auf in der oberen rechten Ecke.

So fügen Sie eine Übersicht oder ein Diagramm hinzu

1. Wählen Sie Edit (Bearbeiten) im Abschnitt Overview (Übersicht) oder Monitoring (Überwachung).
2. Wählen Sie eine Resource aus. Die unterstützten Ressourcen sind die Auto Scaling-Gruppe Ihrer Umgebung, der Elastic Load Balancing-Load Balancer und die Umgebung selbst.
3. Wählen Sie für die Ressource einen CloudWatch metric-Wert aus. Eine vollständige Liste der unterstützten Metriken finden Sie unter [Veröffentlichen von benutzerdefinierten Amazon CloudWatch-Metriken für eine Umgebung \(p. 849\)](#).
4. Wählen Sie einen Wert für Statistic aus. Die Standardstatistik ist der Mittelwert der ausgewählten CloudWatch-Metrik für den Zeitraum (Übersicht) oder zwischen den Plot-Punkten (Diagramm).
5. Geben Sie eine Description ein. Die Beschreibung ist die Bezeichnung für das Element, das in der Überwachungskonsole angezeigt wird.
6. Wählen Sie Add aus.
7. Wiederholen Sie die vorherigen Schritte, um weitere Elemente hinzuzufügen, oder klicken Sie auf Save (Speichern), um diese Bearbeitung des Abschnitts abzuschließen.

Weitere Informationen zu den Metriken und Maßen der einzelnen Ressourcen finden Sie unter [Referenz der Amazon CloudWatch-Metriken, Namespaces und Maße](#) im Amazon CloudWatch-Benutzerhandbuch.

Elastic Load Balancing und [Amazon EC2](#)-Metriken sind für alle Umgebungen aktiviert.

Mit [erweiterten Zustandsberichten \(p. 824\)](#) wird die Metrik "EnvironmentHealth" aktiviert. Ein entsprechendes Diagramm wird automatisch zur Überwachungskonsole hinzugefügt. Sofern Sie weitere Metriken in der Umgebungskonfiguration aktivieren, stehen auch diese in der Überwachungskonsole zur Verfügung. Zudem wird mit erweiterten Zustandsberichten auch eine [Zustandsprüfungsseite \(p. 836\)](#) zur Management Console hinzugefügt.

Note

Wenn Sie zusätzliche CloudWatch-Metriken für die Umgebung aktivieren, dauert es einige Minuten, bis Berichte für diese erstellt und sie in der Metrikliste angezeigt werden, die Sie zum Hinzufügen von Diagrammen und Übersichtsstatistiken verwenden.

Eine Liste der verfügbaren erweiterten Zustandsmetriken finden Sie unter [Veröffentlichen von benutzerdefinierten Amazon CloudWatch-Metriken für eine Umgebung \(p. 849\)](#).

Grundlegende Zustandsberichte

AWS Elastic Beanstalk nutzt Informationen aus verschiedenen Quellen, um zu bestimmen, ob Ihre Umgebung verfügbar ist und Anfragen aus dem Internet verarbeitet. Der Zustand einer Umgebung wird durch eine von vier Farben dargestellt und auf der [Umgebungsübersichtsseite \(p. 426\)](#) der Elastic Beanstalk-Konsole angezeigt. Er ist auch über die [DescribeEnvironments](#)-API und durch Aufruf von eb status mit der [EB CLI \(p. 1008\)](#) verfügbar.

Vor Version 2-Linux-Plattformversionen gab es nur das grundlegende System für Zustandsberichte. Das grundlegende Zustandsberichtssystem bietet Informationen über den Zustand der Instances in einer Elastic Beanstalk-Umgebung basierend auf Zustandsprüfungen, die von Elastic Load Balancing für lastverteilte Umgebungen oder von Amazon Elastic Compute Cloud für Umgebungen mit einer Instance durchgeführt werden.

Zusätzlich zum Prüfen des Zustands Ihrer EC2-Instances überwacht Elastic Beanstalk außerdem die anderen Ressourcen in Ihrer Umgebung und meldet fehlende oder falsch konfigurierte Ressourcen, die dazu führen können, dass Ihre Umgebung nicht mehr für Benutzer verfügbar ist.

Metriken, die von den Ressourcen in Ihrer Umgebung gesammelt werden, werden in Fünf-Minuten-Intervallen in Amazon CloudWatch veröffentlicht. Dazu zählen Betriebssystemmetriken aus EC2, Anforderungs metriken aus Elastic Load Balancing. Sie können Diagramme basierend auf diesen CloudWatch-Metriken auf der Seite [Monitoring \(p. 817\)](#) der Umgebungskonsole anzeigen. Für grundlegende Zustandsberichte werden diese Metriken nicht verwendet, um den Zustand einer Umgebung zu bestimmen.

Themen

- [Zustandsfarben \(p. 822\)](#)
- [Elastic Load Balancing-Zustandsprüfungen \(p. 822\)](#)
- [Zustandsprüfungen für Umgebungen mit einer einzelnen Instance oder Worker-Ebene \(p. 823\)](#)
- [Zusätzliche Prüfungen \(p. 823\)](#)
- [Amazon CloudWatch-Metriken \(p. 823\)](#)

Zustandsfarben

Elastic Beanstalk meldet den Zustand einer Webserverumgebung je nachdem, wie die Anwendung, die darin ausgeführt wird, auf die Zustandsprüfung reagiert. Elastic Beanstalk nutzt eine von vier Farben zum Beschreiben des Status, wie in der folgenden Tabelle gezeigt:

Farbe	Beschreibung
Grau	Ihre Umgebung wird aktualisiert.
Grün	Ihre Umgebung ist die letzte Zustandsprüfung bestanden. Mindestens eine Instance in Ihrer Umgebung ist verfügbar und verarbeitet Anfragen.
Gelb	Ihre Umgebung hat eine oder mehrere Zustandsprüfungen nicht bestanden. Einige Anfragen in Ihrer Umgebung schlagen fehl.
Rot	Ihre Umgebung hat drei oder mehr Zustandsprüfungen nicht bestanden oder eine Umgebungsressource ist nicht mehr verfügbar. Anfragen schlagen durchgängig fehl.

Diese Beschreibungen gelten nur für Umgebungen, die grundlegende Zustandsberichte verwenden. Unter [Farben und Status in Zustandsangaben \(p. 841\)](#) finden Sie Details im Zusammenhang mit der erweiterten Zustandsprüfung.

Elastic Load Balancing-Zustandsprüfungen

In einer Umgebung mit Lastenausgleich sendet Elastic Load Balancing alle 10 Sekunden eine Anfrage an jede Instance in einer Umgebung, um zu bestätigen, dass diese stabil sind. Standardmäßig ist der Load Balancer so konfiguriert, dass er eine TCP-Verbindung auf Port 80 öffnet. Wenn die Instance die Verbindung anerkennt, wird sie als stabil eingestuft.

Sie können diese Einstellung überschreiben, indem Sie eine vorhandene Ressource in Ihrer Anwendung angeben. Wenn Sie einen Pfad angeben, wie `/health`, ist die Zustandsprüfungs-URL auf `HTTP:80/health` festgelegt. Die Zustandsprüfungs-URL sollte auf einen Pfad gesetzt sein, der immer von Ihrer Anwendung bedient wird. Wenn eine statische Webseite festgelegt ist, die vom Webserver vor Ihrer Anwendung bereitgestellt oder zwischengespeichert wird, zeigen Zustandsprüfungen keine Probleme mit dem Anwendungsserver oder Webcontainer. Anweisungen zum Ändern der Zustandsprüfungs-URL finden Sie unter [Zustandsprüfung \(p. 563\)](#).

Wenn eine Zustandsprüfungs-URL konfiguriert ist, erwartet Elastic Load Balancing eine GET-Anfrage, auf die die Antwort `200 OK` zurückgegeben werden soll. Die Anwendung besteht die Zustandsprüfung nicht, wenn sie nicht innerhalb von fünf Sekunden reagiert oder einen anderen HTTP-Statuscode zurückgibt. Nach 5 aufeinanderfolgenden Fehlern bei der Zustandsprüfung nimmt Elastic Load Balancing die Instanz außer Betrieb.

Weitere Informationen zu den Zustandsprüfungen von Elastic Load Balancing finden Sie unter [Zustandsprüfungen](#) im Elastic Load Balancing-Benutzerhandbuch.

Note

Wenn Sie eine URL für die Zustandsprüfung konfigurieren, ändert dies das Verhalten einer Auto Scaling-Gruppe in der Umgebung nicht. Eine fehlerhafte Instance wird aus dem Load Balancer gelöscht, aber nicht automatisch durch Amazon EC2 Auto Scaling ersetzt, es sei denn, Amazon EC2 Auto Scaling wird für die Verwendung der Elastic Load Balancing-Zustandsprüfung als Basis für das Ersetzen von Instances konfiguriert. Informationen zum Konfigurieren von Amazon EC2

Auto Scaling zum Ersetzen von Instances, die eine Zustandsprüfung durch Elastic Load Balancing nicht bestehen, finden Sie unter [Zustandsprüfungseinstellung für Auto Scaling \(p. 554\)](#)

Zustandsprüfungen für Umgebungen mit einer einzelnen Instance oder Worker-Ebene

In einer Umgebung mit einer einzelnen Instance oder Worker-Ebene bestimmt Elastic Beanstalk den Zustand einer Instance durch Überwachung des Amazon EC2-Instance-Status. Einstellungen der Elastic Load Balancing-Zustandsprüfung, einschließlich HTTP-Zustandsprüfungs-URLs, können in diesen Umgebungstypen nicht verwendet werden.

Weitere Informationen zu Statusprüfungen von Amazon EC2-Instances finden Sie auf der Seite über das [Überwachen von Instances mittels Statusprüfungen](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Zusätzliche Prüfungen

Zusätzlich zu den Elastic Load Balancing -Zustandsprüfungen überwacht Elastic Beanstalk-Ressourcen in Ihrer Umgebung und ändert den Status in Rot, wenn sie nicht bereitgestellt werden, nicht korrekt konfiguriert sind oder nicht mehr verfügbar sind. Diese Prüfungen bestätigen, dass:

- Die Auto Scaling-Gruppe der Umgebung verfügbar ist und über mindestens eine Instance verfügt.
- Die Sicherheitsgruppe der Umgebung ist verfügbar und so konfiguriert, dass eingehender Datenverkehr auf Port 80 zulässig ist.
- Der Umgebungs-CNAME ist vorhanden und weist auf den rechten Load Balancer.
- In einer Worker-Umgebung wird die Amazon Simple Queue Service (Amazon SQS)-Warteschlange mindestens einmal alle drei Minuten abgefragt.

Amazon CloudWatch-Metriken

Bei grundlegenden Zustandsberichten veröffentlicht der Elastic Beanstalk-Service keine Metriken in Amazon CloudWatch. Die CloudWatch-Metriken, die zum Erstellen von Diagrammen auf der Seite [Monitoring \(p. 817\)](#) der Umgebungskonsole verwendet werden, werden von den Ressourcen in Ihrer Umgebung veröffentlicht.

Beispiel: EC2 veröffentlicht die folgenden Metriken für die Instances in der Auto Scaling-Gruppe Ihrer Umgebung:

CPUUtilization

Prozentsatz der Recheneinheiten, die zurzeit verwendet werden.

`DiskReadBytes, DiskReadOps, DiskWriteBytes, DiskWriteOps`

Anzahl der gelesenen und geschriebenen Bytes und die Anzahl der Lese- und Schreibvorgänge.

`NetworkIn, NetworkOut`

Anzahl der gesendeten und empfangenen Bytes.

Elastic Load Balancing veröffentlicht die folgenden Metriken für den Load Balancer Ihrer Umgebung:

`BackendConnectionErrors`

Anzahl der fehlgeschlagenen Verbindung zwischen dem Load Balancer und den Umgebung-Instances.

HTTPCode_Backend_2XX, HTTPCode_Backend_4XX

Anzahl der erfolgreichen (2XX) und Client-Fehler-Antwortcodes (4XX), die von Instances in Ihrer Umgebung generiert wurden.

Latency

Anzahl der Sekunden zwischen dem Zeitpunkt, an dem der Load Balancer eine Anfrage an eine Instance weiterleitet, und dem Zeitpunkt, zu dem die Antwort empfangen wird.

RequestCount

Anzahl der abgeschlossenen Anfragen.

Diese Listen sind nicht vollständig. Eine vollständige Liste der Metriken, die für diese Ressourcen gemeldet werden können, finden Sie in den folgenden Themen im Amazon CloudWatch-Entwicklerhandbuch:

Metriken

Namespace	Topic
AWS::ElasticLoadBalancing::LoadBalancer	Elastic Load Balancing-Metriken und -Ressourcen
AWS::AutoScaling::AutoScalingGroup	Amazon Elastic Compute Cloud-Metriken und -Ressourcen
AWS::SQS::Queue	Amazon SQS-Metriken und -Ressourcen
AWS::RDS::DBInstance	Amazon RDS-Dimensionen und -Metriken

Worker-Umgebung – Zustandsmetrik

Nur für Worker-Umgebungen veröffentlicht der SQS-Daemon eine benutzerdefinierte Metrik für den Umgebungszustand in CloudWatch, wobei ein Wert von 1 grün bedeutet. Sie können die CloudWatch-Zustandsmetrikdaten in Ihrem Konto mithilfe des `ElasticBeanstalk/SQSD`-Namespace überprüfen. Die Metrikdimension ist `EnvironmentName` und der Metrikname lautet `Health`. Alle Instances veröffentlichen ihre Metriken auf dem gleichen Namespace.

Damit der Daemon Metriken veröffentlichen kann, muss das Instance-Profil der Umgebung die Berechtigung zum Aufrufen von `cloudwatch:PutMetricData` haben. Diese Berechtigung ist im Standard-Instance-Profil enthalten. Weitere Informationen finden Sie unter [Elastic Beanstalk Instance-Profile verwalten \(p. 904\)](#).

Erweiterte Zustandsberichte und Überwachung

Erweiterte Integritätsberichte ist eine Funktion, die Sie auf Ihrer Umgebung aktivieren können, damit AWS Elastic Beanstalk weitere Informationen über Ressourcen in Ihrer Umgebung erfassen kann. Elastic Beanstalk analysiert die gesammelten Informationen, um ein besseres Bild der Gesamtumgebungsintegrität zu erhalten und um bei der Identifizierung von Problemen zu helfen, die dazu führen können, dass Ihre Anwendung nicht mehr verfügbar ist.

Zusätzlich zu Änderungen bei der Funktionsweise von Zustandsfarben, fügt der erweiterte Zustandsbericht einen Status-Deskriptor hinzu, der auf den Schweregrad von Problemen hinweist, wenn eine Umgebung gelb oder rot ist. Wenn weitere Informationen über den aktuellen Status verfügbar sind, können Sie die Schaltfläche `Causes` wählen, um detaillierte Zustandsinformationen auf der Seite [Zustand \(p. 835\)](#) anzuzeigen.

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.bx7dx222kv.us-east-2.elasticbeanstalk.com](#)

Health Tomcat 8
64b

Running version
Sample Application
[Upload and deploy](#)

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.

Um detaillierte Integritätsinformationen zu den Amazon EC2-Instances bereitzustellen, die in Ihrer Umgebung ausgeführt werden, enthält Elastic Beanstalk für jede Plattformversion, die erweiterte Integritätsberichte unterstützt, einen [Integritäts-Agenten \(p. 827\)](#) im Amazon Machine Image (AMI). Der Integritäts-Agent überwacht Webserverprotokolle und Systemmetriken und leitet sie an den Elastic Beanstalk-Service weiter. Elastic Beanstalk analysiert diese Metriken und Daten von Elastic Load Balancing und Amazon EC2 Auto Scaling, um ein Gesamtbild der Integrität einer Umgebung zu erhalten.

Zusätzlich zum Sammeln und Präsentieren von Informationen über die Ressourcen Ihrer Umgebung überwacht Elastic Beanstalk die Ressourcen in Ihrer Umgebung auf verschiedene Fehlerbedingungen und bietet Benachrichtigungen, um Fehler zu vermeiden und Konfigurationsprobleme zu lösen. [Faktoren, die den Zustand Ihrer Umgebung beeinflussen \(p. 827\)](#), umfassen die Ergebnisse jeder einzelnen Anforderung, die von Ihrer Anwendung bereitgestellt wird, Metriken des Betriebssystems Ihrer Instance und den Status der letzten Bereitstellung.

Sie können den Integritätsstatus in Echtzeit auf der [Umgebungsübersichtsseite \(p. 835\)](#) in der Elastic Beanstalk-Konsole oder mit dem Befehl [eb health \(p. 1033\)](#) in der [Elastic Beanstalk-Befehlszeilenschnittstelle \(p. 1008\)](#) (EB CLI) anzeigen. Zum Erfassen und Verfolgen der Umgebungs- und Instance-Integrität im Laufe der Zeit können Sie Ihre Umgebung so konfigurieren, dass die von

Elastic Beanstalk erfassten Informationen für erweiterte Integritätsberichte in Amazon CloudWatch als benutzerdefinierte Metriken veröffentlicht werden. CloudWatch-[Gebühren](#) für benutzerdefinierte Metriken gelten für alle Metriken mit Ausnahme von `EnvironmentHealth`, was kostenlos ist.

Erweiterte Zustandsberichte erfordern Version 2 oder eine neuere [Plattformversion \(p. 32\)](#). Zum Überwachen von Ressourcen und Veröffentlichungen von Metriken benötigt Ihre Umgebung sowohl ein [Instance-Profil als auch eine Service \(p. 824\)](#)-Rolle. haben. Die Multicontainer-Docker-Plattform umfasst standardmäßig keinen Webserver, kann jedoch mit erweiterten Zustandsberichten verwendet werden, wenn Sie Ihren Webserver so konfigurieren, dass [Protokolle im richtigen Format bereitgestellt werden \(p. 856\)](#).

Hinweise zur Windows-Plattform

- Diese Funktion steht bei [Windows Server-Plattformversionen](#) vor Version 2 (v2) nicht zur Verfügung.
- Wenn Sie erweiterte Zustandsberichte für eine Windows Server-Umgebung aktivieren, ändern Sie nicht die [IIS-Protokollierungskonfiguration](#). Damit die erweiterte Statusüberwachung einwandfrei funktioniert, muss die IIS-Protokollierung mit dem W3C-Format und den Protokollereigniszielden ETW event only (Nur ETW-Ereignis) oder Both log file and ETW event (Sowohl Protokolldatei als auch ETW-Ereignis) konfiguriert werden.

Außerdem dürfen Sie den Windows-Service des [Elastic Beanstalk-Integritäts-Agenten \(p. 827\)](#) auf keiner der Instances Ihrer Umgebung deaktivieren oder anhalten. Damit erweiterte Zustandsinformationen für eine Instance gesammelt und als Bericht zusammengefasst werden, muss dieser Service aktiviert sein und ausgeführt werden.

Für den verbesserten Zustand muss die Umgebung über ein Instance-Profil verfügen. Das Instance-Profil sollte über Rollen verfügen, die Ihren Umgebungs-Instances Berechtigungen zum Sammeln und Berichten von erweiterten Integritätsinformationen bereitstellen. Wenn Sie erstmals eine Umgebung mit einer v2-Plattformversion in der Elastic Beanstalk-Konsole erstellen, fordert Elastic Beanstalk Sie auf, die erforderlichen Rollen zu erstellen, und aktiviert standardmäßig die erweiterten Integritätsberichte. Lesen Sie Details darüber, wie die erweiterten Integritätsberichte funktionieren, oder besuchen Sie [Aktivieren der erweiterten Elastic Beanstalk-Integritätsberichte \(p. 832\)](#), um die Funktion sofort zu verwenden.

Amazon Linux 2-Plattformen benötigen Instance-Profile, sodass sie immer eine erweiterte Integrität unterstützen können. Wenn Sie eine Umgebung mit einer Amazon Linux 2-Plattform erstellen, aktiviert Elastic Beanstalk immer die erweiterte Integrität. Dies gilt unabhängig davon, wie Sie die Umgebung erstellen – mit der Elastic Beanstalk-Konsole, der EB CLI, der AWS CLI oder der API.

Themen

- [Der Elastic Beanstalk-Integritäts-Agent \(p. 827\)](#)
- [Faktoren bei der Bestimmung des Instance- und Umgebungszustands \(p. 827\)](#)
- [Anpassung der Regel für die Zustandsprüfung \(p. 829\)](#)
- [Rollen in erweiterten Zustandsberichten \(p. 830\)](#)
- [Erweiterte Zustandsautorisierung \(p. 830\)](#)
- [Ereignisse in erweiterten Zustandsberichten \(p. 831\)](#)
- [Verhalten der erweiterten Zustandsberichte bei Aktualisierungen, Bereitstellungen und Skalierung \(p. 831\)](#)
- [Aktivieren der erweiterten Elastic Beanstalk-Integritätsberichte \(p. 832\)](#)
- [Erweiterte Integritätsüberwachung mithilfe der Environment Management Console \(p. 835\)](#)
- [Farben und Status in Zustandsangaben \(p. 841\)](#)
- [Instance-Metriken \(p. 843\)](#)
- [Konfigurieren von Regeln für den erweiterten Zustand einer Umgebung \(p. 845\)](#)

- [Veröffentlichen von benutzerdefinierten Amazon CloudWatch-Metriken für eine Umgebung \(p. 849\)](#)
- [Verwenden der erweiterten Integritätsberichte mit der Elastic Beanstalk-API \(p. 855\)](#)
- [Format der Protokolle der erweiterten Zustandsberichte \(p. 856\)](#)
- [Benachrichtigungen und Fehlerbehebung \(p. 859\)](#)

Der Elastic Beanstalk-Integritäts-Agent

Der Elastic Beanstalk-Integritäts-Agent ist ein Daemon-Prozess (oder in Windows-Umgebungen ein Service), der auf den einzelnen Amazon EC2-Instances in Ihrer Umgebung ausgeführt wird, das Betriebssystem und Integritätsmetriken auf Anwendungsebene überwacht und Probleme an Elastic Beanstalk meldet. Der Zustandsagent ist in allen Plattformversionen ab Version 2.0 einer jeden Plattform enthalten.

Der Integritäts-Agent meldet ähnliche Metriken wie die, die von Amazon EC2 Auto Scaling und Elastic Load Balancing [in CloudWatch veröffentlicht werden \(p. 823\)](#), als Teil der [grundlegenden Integritätsberichte \(p. 821\)](#), einschließlich CPU-Auslastung, HTTP-Codes und Latenz. Der Integritäts-Agent meldet jedoch direkt an Elastic Beanstalk – mit höherer Granularität und Häufigkeit als die grundlegenden Integritätsberichte.

Für grundlegende Zustandsberichte werden diese Metriken alle fünf Minuten veröffentlicht und können mit Diagrammen in der Environment Management Console überwacht werden. Bei den erweiterten Integritätsberichten meldet der Elastic Beanstalk-Integritäts-Agent die Metriken alle 10 Sekunden an Elastic Beanstalk. Elastic Beanstalk verwendet die Metriken des Integritäts-Agenten, um den Integritätsstatus jeder Instance in der Umgebung zu bestimmen, und, in Kombination mit anderen [Faktoren \(p. 827\)](#), um die Gesamtintegrität der Umgebung zu bestimmen.

Die Gesamtintegrität der Umgebung kann auf der Umgebungsübersichtsseite der Elastic Beanstalk-Konsole in Echtzeit angezeigt werden und wird von Elastic Beanstalk alle 60 Sekunden in CloudWatch veröffentlicht. Sie können detaillierte Metriken, die vom Zustandsagenten in Echtzeit gemeldet werden, mit dem [eb health \(p. 1033\)](#)-Befehl in der [EB CLI \(p. 1008\)](#) anzeigen.

Gegen eine zusätzliche Gebühr können Sie einzelne Metriken auf Instance- und Umgebungsebene alle 60 Sekunden in CloudWatch veröffentlichen. In CloudWatch veröffentlichte Metriken können dann verwendet werden, um [Überwachungsdiagramme \(p. 819\)](#) in der [Environment Management Console \(p. 426\)](#) zu erstellen.

Für erweiterte Integritätsberichte fällt nur dann eine Gebühr an, wenn Sie erweiterte Integritätsmetriken in CloudWatch veröffentlichen. Wenn Sie erweiterte Zustandsberichte verwenden, werden die grundlegenden Zustandsmetriken noch kostenlos veröffentlicht, auch wenn Sie keine erweiterten Zustandsmetriken veröffentlichen.

Unter [Instance-Metriken \(p. 843\)](#) finden Sie Details zu Metriken, die vom Integritäts-Agenten veröffentlicht werden. Weitere Informationen zum Veröffentlichen erweiterter Integritätsmetriken in CloudWatch finden Sie unter [Veröffentlichen von benutzerdefinierten Amazon CloudWatch-Metriken für eine Umgebung \(p. 849\)](#).

Faktoren bei der Bestimmung des Instance- und Umgebungszustands

Zusätzlich zu den grundlegenden Systemprüfungen der Integritätsberichte, einschließlich [Elastic Load Balancing-Zustandsprüfungen \(p. 822\)](#) und [Ressourcenüberwachung \(p. 823\)](#), sammeln die erweiterten Integritätsberichte von Elastic Beanstalk zusätzliche Daten zur Integrität der Instances in Ihrer Umgebung. Dazu zählen Betriebssystemmetriken, Serverprotokolle und der Zustand laufender Umgebungsoperationen, wie Bereitstellungen und Updates. Das Integritätsberichtesystem von Elastic Beanstalk kombiniert

Informationen aus allen verfügbaren Quellen und analysiert sie, um die Gesamtintegrität der Umgebung zu bestimmen.

Operationen und Befehle

Wenn Sie eine Operation in Ihrer Umgebung durchführen, z. B. die Bereitstellung einer neuen Version einer Anwendung, nimmt Elastic Beanstalk mehrere Änderungen vor, die sich auf den Integritätsstatus der Umgebung auswirken.

Wenn Sie beispielsweise eine neue Version einer Anwendung in einer Umgebung bereitstellen, in der mehreren Instances ausgeführt werden, werden beim Überwachen des Zustands der Umgebung [mithilfe der EB CLI \(p. 1033\)](#) möglicherweise Meldung ähnlich der folgenden angezeigt:

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
i-ba65c144	Pending	Performing application deployment (running for 31 seconds)
i-f6a2d525	Ok	Performing initialization (running for 12 seconds)
seconds		Application deployment completed 23 seconds ago and took 26
i-e8a2d53b	Pending	94 % of CPU is in use. 52 % in I/O wait
i-e81cca40	Ok	Performing application deployment (running for 33 seconds)

In diesem Beispiel lautet der Gesamtstatus der Umgebung `Ok` und die Ursache dieses Status ist, dass der Befehl drei von fünf Instances ausführt. Drei der Instances in der Umgebung haben den Status `Pending`, was bedeutet, dass ein Vorgang ausgeführt wird.

Wenn ein Vorgang abgeschlossen ist, meldet Elastic Beanstalk zusätzliche Informationen über die Operation. Zum Beispiel zeigt Elastic Beanstalk die folgenden Informationen zu einer Instance, die bereits mit der neuen Version der Anwendung aktualisiert wurde:

i-f6a2d525	Ok	Application deployment completed 23 seconds ago and took 26
------------	----	---

Instance-Zustandsinformationen umfassen auch Details zur aktuellen Bereitstellung für jede Instance in Ihrer Umgebung. Jede Instance meldet eine Bereitstellungs-ID und einen Status. Die Bereitstellungs-ID ist eine ganze Zahl, die sich jedes Mal erhöht, wenn Sie eine neue Version Ihrer Anwendung bereitstellen oder Einstellungen für die Konfigurationsoptionen für die Instance, wie Umgebungsvariablen, ändern. Sie können die Bereitstellungsinformationen zur Identifizierung von Instances verwenden, auf denen die falsche Version Ihrer Anwendung ausgeführt wird, nachdem eine [fortlaufende Bereitstellung \(p. 474\)](#) fehlgeschlagen ist.

In der Spalte mit der Ursache schließt Elastic Beanstalk Informationsmeldungen zu erfolgreichen Operationen und anderen fehlerfreien Zuständen in mehreren Integritätsprüfungen ein, diese werden aber nicht unbegrenzt beibehalten. Ursachen für fehlerhafte Umgebungsstatus verbleiben, bis die Umgebung in einen fehlerfreien Status zurückkehrt.

Befehls-Timeout

Elastic Beanstalk wendet einen Befehls-Timeout ab dem Zeitpunkt an, an dem eine Operation beginnt, damit eine Instance in einen fehlerfreien Zustand übergehen kann. Dieser Befehls-Timeout wird in der Update- und Bereitstellungskonfiguration Ihrer Umgebung festgelegt (im [aws: elasticbeanstalk: Befehl \(p. 685\)](#)-Namespace) und standardmäßig auf zehn Minuten gesetzt.

Während fortlaufenden Updates wendet Elastic Beanstalk einen separaten Timeout auf jeden Stapel in der Operation an. Dieser Timeout wird als Teil der fortlaufenden Update-Konfiguration der Umgebung (im [aws:](#)

[autoscaling: updatepolicy: rollingupdate \(p. 677\)](#)-Namespace) festgelegt. Wenn alle Instances im Stapel innerhalb des Befehls-Timeout fehlerfrei sind, fährt die Operation mit dem nächsten Stapel fort. Wenn dies nicht der Fall ist, schlägt die Operation fehl.

Note

Wenn Ihre Anwendung Integritätsprüfungen nicht mit dem Status OK besteht, sie auf einer anderen Stufe jedoch stabil ist, können Sie die `HealthCheckSuccessThreshold`-Option im [aws:elasticbeanstalk:command namespace \(p. 685\)](#)-Namespace so festlegen, dass die Stufe geändert wird, auf der Elastic Beanstalk eine Instance als stabil betrachtet.

Damit eine Webserverumgebung als fehlerfrei angesehen wird, muss jede Instance in der Umgebung oder im Stapel zwölf aufeinanderfolgende Zustandsprüfungen im Verlauf von zwei Minuten bestehen. Bei einer Umgebung mit Worker-Ebene muss jede Instance 18 Zustandsprüfungen bestehen. Bevor ein Befehls-Timeout auftritt, senkt Elastic Beanstalk den Integritätsstatus einer Umgebung nicht, wenn die Integritätsprüfungen fehlschlagen. Wenn die Instances in der Umgebung innerhalb des Befehls-Timeouts fehlerfrei werden, ist die Operation erfolgreich.

HTTP-Anforderungen

Wenn keine Operation in einer Umgebung ausgeführt wird, ist die primäre Quelle für Informationen über den Instance- und Umgebungszustand die Webserverprotokolle für jede Instance. Um die Integrität einer Instance und die Gesamtintegrität der Umgebung zu prüfen, berücksichtigt Elastic Beanstalk die Anzahl der Anforderungen, das Ergebnis jeder Anforderung und die Geschwindigkeit, mit der jede Anforderung gelöst wurde.

Auf Linux-basierten Plattformen liest und analysiert Elastic Beanstalk Web-Server-Protokolle zum Abrufen von Informationen über HTTP-Anfragen. Auf der Windows Server-Plattform erhält Elastic Beanstalk diese Informationen [direkt von dem IIS-Webserver \(p. 845\)](#).

Ihre Umgebung verfügt möglicherweise nicht über einen aktiven Webserver. So beinhaltet die Multicontainer Docker-Plattform beispielsweise keinen Webserver. Andere Plattformen enthalten einen Webserver, der von Ihrer Anwendung möglicherweise deaktiviert wird. In diesen Fällen ist für Ihre Umgebung eine zusätzliche Konfiguration erforderlich, um dem [Elastic Beanstalk-Integritäts-Agenten \(p. 827\)](#) Protokolle in dem Format zur Verfügung zu stellen, das zur Weiterleitung von Integritätsinformationen an den Elastic Beanstalk-Service benötigt wird. Für Details siehe [Format der Protokolle der erweiterten Zustandsberichte \(p. 856\)](#).

Betriebssystemmetriken

Elastic Beanstalk überwacht Betriebssystemmetriken, die vom Integritäts-Agenten gemeldet wurden, um Instances zu identifizieren, die konsequent geringe Systemressourcen haben.

Unter [Instance-Metriken \(p. 843\)](#) finden Sie Details zu Metriken, die vom Integritäts-Agenten veröffentlicht werden.

Anpassung der Regel für die Zustandsprüfung

Erweiterte Integritätsberichte von Elastic Beanstalk basieren auf mehreren Regeln, um die Integrität Ihrer Umgebung zu bestimmen. Einige dieser Regeln sind möglicherweise nicht für Ihre jeweilige Anwendung geeignet. Ein häufiger Fall ist eine Anwendung, die standardmäßig viele HTTP-4xx-Fehler zurückgibt. Elastic Beanstalk kommt unter Verwendung einer seiner Standardregeln zu dem Schluss, dass etwas schief läuft, und ändert den Integritätsstatus Ihrer Umgebung je nach Fehlerrate von „OK“ auf „Warning (Warnung)“, „Degraded (Schwach)“ oder „Severe (Stark)“. Um diesen Fall ordnungsgemäß zu verarbeiten, gestattet Ihnen Elastic Beanstalk, diese Regel zu konfigurieren und HTTP-4xx-Fehler von Anwendungen zu ignorieren. Details dazu finden Sie unter [Konfigurieren von Regeln für den erweiterten Zustand einer Umgebung \(p. 845\)](#).

Rollen in erweiterten Zustandsberichten

Erweiterte Integritätsberichte erfordern zwei Rollen – eine Servicerolle für Elastic Beanstalk und ein Instance-Profil für die Umgebung. Die Servicerolle ermöglicht Elastic Beanstalk die Interaktion mit anderen AWS-Services in Ihrem Auftrag, um Informationen über die Ressourcen in Ihrer Umgebung zu sammeln. Das Instance-Profil ermöglicht es den Instances in Ihrer Umgebung, Protokolle in Amazon S3 zu schreiben und erweiterte Integritätsinformationen an den Elastic Beanstalk-Service zu übermitteln.

Wenn Sie eine Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-Konsole oder der EB CLI erstellen, erstellt Elastic Beanstalk eine Standard-Servicerolle und fügt die erforderlichen verwalteten Richtlinien an ein Standard-Instance-Profil für Ihre Umgebung an.

Wenn Sie die API, ein SDK oder die AWS-CLI zum Erstellen von Umgebungen verwenden, müssen Sie diese Rollen im Vorfeld erstellen und während der Umgebungserstellung angeben, um erweiterte Zustandsberichte zu verwenden. Anweisungen zum Erstellen geeigneter Rollen für Ihre Umgebungen finden Sie unter [Servicerollen, Instance-Profile und Benutzerrichtlinien \(p. 21\)](#).

Es wird empfohlen, verwaltete Richtlinien für Ihr Instance-Profil und Ihre Servicerolle zu verwenden. Verwaltete Richtlinien sind AWS Identity and Access Management (IAM)-Richtlinien, die von Elastic Beanstalk verwaltet werden. Durch die Verwendung verwalteter Richtlinien wird sichergestellt, dass Ihre Umgebung über alle erforderlichen Berechtigungen verfügt, um ordnungsgemäß zu funktionieren.

Für das Instance-Profil können Sie die verwalteten Richtlinien `AWSElasticBeanstalkWebTier` oder `AWSElasticBeanstalkWorkerTier` verwenden, je nachdem, ob es sich um eine Umgebung auf [Webserverebene \(p. 15\)](#) oder auf [Worker-Ebene \(p. 16\)](#) handelt. Ausführliche Informationen zu diesen beiden Richtlinien für verwaltete Instance-Profile finden Sie unter [the section called “Instance-Profile” \(p. 904\)](#).

Erweiterte Zustandsautorisierung

Die Richtlinien des verwalteten Elastic Beanstalk-Instance-Profils enthalten die Berechtigung für die Aktion `elasticbeanstalk:PutInstanceStatistics`. Diese Aktion ist nicht Teil der Elastic Beanstalk-API. Sie ist Teil einer anderen API, die Umgebungs-Instances intern verwenden, um erweiterte Integritätsinformationen an den Elastic Beanstalk-Service zu übermitteln. Sie rufen diese API nicht direkt auf.

Standardmäßig ist die Autorisierung für die Aktion `elasticbeanstalk:PutInstanceStatistics` nicht aktiviert. Daher ist es nicht erforderlich, sie für Ihre Anwendungs- und Umgebungsressourcen in Ihrem Instance-Profil zu erlauben, wie es die verwaltete Richtlinie tut. Um die Sicherheit Ihrer Umgebung zu erhöhen und das Spoofing von Zustandsdaten in Ihrem Namen zu verhindern, empfehlen wir, die Option `EnhancedHealthAuthEnabled` im Namespace [the section called “aws:elasticbeanstalk:healthreporting:system” \(p. 692\)](#) auf `true` festzulegen. Sie können diese Option mithilfe einer [Optionseinstellung \(p. 724\)](#) in einer [Konfigurationsdatei \(p. 722\)](#) konfigurieren.

Wenn Sie die Autorisierung für `elasticbeanstalk:PutInstanceStatistics` mithilfe der Option `EnhancedHealthAuthEnabled` aktivieren und verwaltete Richtlinien für Ihr Instance-Profil verwenden, müssen Sie nichts weiter tun, unsere verwalteten Richtlinien decken die erforderliche Autorisierung ab. Wenn Sie anstelle einer verwalteten Richtlinie ein benutzerdefiniertes Instance-Profil verwenden, wird für Ihre Umgebung möglicherweise der Status No Data (Keine Daten) angezeigt. Dies geschieht, weil die Instances nicht für die Aktion autorisiert sind, die erweiterte Zustandsdaten an den Service übermittelt. Um die Aktion zu autorisieren, fügen Sie die folgende Anweisung in Ihr Instance-Profil ein.

```
{  
    "Sid": "ElasticBeanstalkHealthAccess",  
    "Action": [  
        "elasticbeanstalk:PutInstanceStatistics"  
    ],
```

```
"Effect": "Allow",
"Resource": [
    "arn:aws:elasticbeanstalk::application/*",
    "arn:aws:elasticbeanstalk::environment/*"
]
}
```

Ereignisse in erweiterten Zustandsberichten

Das System für erweiterte Zustandsberichte generiert Ereignisse, wenn eine Umgebung zwischen Zuständen wechselt. Das folgende Beispiel zeigt ausgegebene Ereignisse durch eine Umgebungsübertragung zwischen den Zuständen Info, OK und Severe (Schwerwiegend).

Recent events		
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-0a4a27bbbff9994ba5] 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. 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.

Wenn in einen schlechteren Zustand gewechselt wird, enthält das Ereignis der erweiterten Zustandsberichte eine Meldung mit der Ursache des Wechsels.

Nicht alle Änderungen des Status auf Instance-Ebene bewirken, dass Elastic Beanstalk ein Ereignis ausgibt. Um Fehlalarme zu verhindern, generiert Elastic Beanstalk nur dann ein Integritätsbezogenes Ereignis, wenn ein Problem über mehrere Prüfungen hinweg bestehen bleibt.

Echtzeit-Integritätsinformationen auf Umgebungsebene, einschließlich Zustand, Farbe und Ursache, sind auf der [Umgebungsübersichtsseite \(p. 427\)](#) der Elastic Beanstalk-Konsole und in der [EB CLI \(p. 1008\)](#) verfügbar. Indem Sie die EB CLI an Ihre Umgebung anfügen und den [eb health \(p. 1033\)](#)-Befehl ausführen, können Sie auch Echtzeitstatus aus jeder der Instances in Ihrer Umgebung anzeigen.

Verhalten der erweiterten Zustandsberichte bei Aktualisierungen, Bereitstellungen und Skalierung

Das Aktivieren von erweiterten Zustandsberichten kann beeinflussen, wie sich Ihre Umgebung während Konfigurations-Updates und -bereitstellungen verhält. Elastic Beanstalk schließt eine Reihe von Aktualisierungen erst dann ab, wenn alle Instances die Integritätsprüfungen durchgängig bestehen. Da die erweiterten Zustandsberichte einen höheren Standard für den Zustand anwenden und mehr Faktoren überwachen, bestehen Instances, die die grundlegende [ELB-Zustandsprüfung \(p. 822\)](#) der Zustandsberichte bestehen, nicht notwendigerweise die Prüfung mit erweiterten Zustandsberichten.

Weitere Informationen dazu, wie Zustandsprüfungen den Update-Prozess beeinflussen, finden Sie in den Themen [fortlaufende Konfigurations-Updates \(p. 484\)](#) und [fortlaufende Bereitstellungen \(p. 474\)](#).

Erweiterte Integritätsberichte können auch die Notwendigkeit, eine ordnungsgemäße [Integrationsprüfungs-URL \(p. 563\)](#) für Elastic Load Balancing festzulegen, hervorheben. Wenn Ihre Umgebung nach oben skaliert wird, um den Bedarf zu erfüllen, beginnen neue Instances mit der Annahme von Anforderungen, sobald sie ausreichend ELB-Zustandsprüfungen bestehen. Wenn eine Zustandsprüfungs-URL nicht konfiguriert ist, dauert es nur 20 Sekunden, bis eine neue Instance eine TCP-Anwendung akzeptieren kann.

Wenn Ihre Anwendung den Start nicht abgeschlossen hat, bis der Load Balancer sie für ausreichend stabil für den Empfang von Datenverkehr erklärt hat, sehen Sie zahlreiche fehlerhafte Anforderungen und Ihre Umgebung besteht die Zustandsprüfungen nicht mehr. Eine Zustandsprüfungs-URL, die auf einen Pfad abzielt, der von Ihrer Anwendung verarbeitet wird, kann dieses Problem verhindern. ELB-Zustandsprüfungen schlagen so lange fehl, bis eine GET-Anforderung an die Zustandsprüfungs-URL als Statuscode 200 zurückgibt.

Aktivieren der erweiterten Elastic Beanstalk-Integritätsberichte

Neue Umgebungen, die mit den aktuellen [Plattformversionen \(p. 32\)](#) erstellt wurden, verfügen über den [Integrations-Agenten \(p. 827\)](#) von AWS Elastic Beanstalk, der erweiterte Integritätsberichte unterstützt. Wenn Sie Ihre Umgebung in der Elastic Beanstalk-Konsole oder mit der EB CLI erstellen, sind erweiterte Integritätsberichte standardmäßig aktiviert. Sie können die Einstellungen der erweiterten Zustandsberichte auch im Quellcode Ihrer Anwendung mithilfe von [Konfigurationsdateien \(p. 722\)](#) festlegen.

Erweiterte Zustandsberichte erfordern ein [Instance-Profil \(p. 22\)](#) und eine [Servicerolle \(p. 21\)](#) mit den Standardberechtigungen. Wenn Sie eine Umgebung in der Elastic Beanstalk-Konsole erstellen, legt Elastic Beanstalk die erforderlichen Rollen automatisch fest. Anweisungen zum Erstellen Ihrer ersten Umgebung finden Sie unter [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#).

Themen

- [Aktivieren der erweiterten Integritätsberichte mit der Elastic Beanstalk-Konsole \(p. 832\)](#)
- [Aktivieren der erweiterten Zustandsberichte mit der EB-CLI \(p. 834\)](#)
- [Aktivieren der erweiterten Zustandsberichte mit einer Konfigurationsdatei \(p. 835\)](#)

Aktivieren der erweiterten Integritätsberichte mit der Elastic Beanstalk-Konsole

So aktivieren Sie erweiterte Integritätsberichte in einer laufenden Umgebung mit der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Monitoring (Überwachung) die Option Edit (Bearbeiten).
5. Klicken Sie unter Health reporting (Zustandsberichte) für die Option System auf Enhanced (Erweitert).

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. A custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Metrics](#).

System

Enhanced

Basic

CloudWatch Custom Metrics - Instance

[Choose metrics](#)

CloudWatch Custom Metrics - Environment

[Choose metrics](#)

[Cancel](#)

Note

Die Optionen für erweiterte Zustandsberichte werden nicht angezeigt, wenn Sie eine [nicht unterstützte Plattform oder Version \(p. 824\)](#) verwenden.

6. Wählen Sie **Apply (Anwenden)** aus.

Die Elastic Beanstalk-Konsole erstellt standardmäßig erweiterte Integritätsberichte, wenn Sie eine neue Umgebung mit einer Version 2 (v2)-Plattform erstellen. Sie können erweiterte Zustandsberichte deaktivieren, indem Sie die Option für Zustandsberichte während der Umgebungserstellung ändern.

So deaktivieren Sie erweiterte Integritätsberichte, wenn Sie eine Umgebung mit der Elastic Beanstalk-Konsole erstellen

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. [Erstellen Sie eine Anwendung \(p. 403\)](#) oder wählen Sie eine bestehende.
3. [Erstellen Sie eine Umgebung \(p. 435\)](#). Wählen Sie auf der Seite **Create a new environment (Neue Umgebung erstellen)** zuerst **Configure more options (Weitere Optionen konfigurieren)** aus und klicken Sie dann auf **Create environment (Umgebung erstellen)**.
4. Wählen Sie in der Konfigurationskategorie **Monitoring (Überwachung)** die Option **Edit (Bearbeiten)**.
5. Klicken Sie unter **Health reporting (Zustandsberichte)** für die Option **System** auf **Basic (Grundlegend)**.

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. A custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Metrics](#).

System
 Enhanced
 Basic

CloudWatch Custom Metrics - Instance

CloudWatch Custom Metrics - Environment

Health monitoring rule customization
Configure the HTTP application and load balancer status codes included in determining your environment's health. [Learn more](#)

6. Wählen Sie Save (Speichern) aus.

Aktivieren der erweiterten Zustandsberichte mit der EB-CLI

Wenn Sie eine neue Umgebung mit dem eb create-Befehl erstellen, aktiviert die EB CLI erweiterte Zustandsberichte standardmäßig und wendet das Standard-Instance-Profil und die Standard-Servicerolle an.

Sie können mit der --service-role-Option eine andere Servicerolle nach Namen festlegen.

Wenn Sie eine Umgebung haben, auf der grundlegende Zustandsberichte auf einer Version 2 (v2)-Plattformversion ausgeführt werden, und Sie zu erweiterten Zustandsberichten wechseln möchten, führen Sie die folgenden Schritte aus.

So aktivieren Sie die erweiterten Zustandsberichte auf einer laufenden Umgebung mit der [EB CLI \(p. 1008\)](#)

1. Verwenden Sie den eb config-Befehl, um die Konfigurationsdatei im Standard-Texteditor zu öffnen.

```
~/project$ eb config
```

2. Suchen Sie den aws:elasticbeanstalk:environment-Namespace im Bereich mit den Einstellungen. Stellen Sie sicher, dass der Wert von ServiceRole nicht null ist und mit dem Namen Ihrer [Servicerolle \(p. 21\)](#) übereinstimmt.

```
aws:elasticbeanstalk:environment:  
  EnvironmentType: LoadBalanced  
  ServiceRole: aws-elasticbeanstalk-service-role
```

3. Ändern Sie unter dem `aws:elasticbeanstalk:healthreporting:system:`-Namespace den Wert von `SystemType` zu **enhanced**.

```
aws:elasticbeanstalk:healthreporting:system:  
  SystemType: enhanced
```

4. Speichern Sie die Konfigurationsdatei und schließen Sie den Text-Editor.
5. Die EB CLI startet ein Umgebungs-Update, um Ihre Konfigurationsänderungen anzuwenden. Warten Sie, bis der Vorgang abgeschlossen ist, oder drücken Sie `Ctrl+C`, um den Vorgang sicher zu beenden.

```
~/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.
```

Aktivieren der erweiterten Zustandsberichte mit einer Konfigurationsdatei

Sie können erweiterte Zustandsberichte aktivieren, indem Sie eine [Konfigurationsdatei \(p. 722\)](#) zu Ihrem Quell-Bundle hinzufügen. Das folgende Beispiel zeigt eine Konfigurationsdatei, mit der erweiterte Zustandsberichte aktiviert und der Standardservice und das Standard-Instance-Profil zur Umgebung hinzugefügt werden:

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
```

Wenn Sie Ihr eigenes Instance-Profil oder Ihre eigene Servicerolle erstellt haben, ersetzen Sie den hervorgehobenen Text mit den Namen dieser Rollen.

Erweiterte Integritätsüberwachung mithilfe der Environment Management Console

Wenn Sie erweiterte Integritätsberichte in AWS Elastic Beanstalk aktiviert haben, können Sie die Umgebungsintegrität in der [Environment Management Console \(p. 426\)](#) überwachen.

Themen

- [Umgebungsübersicht \(p. 835\)](#)
- [Seite „Umgebungsintegrität“ \(p. 836\)](#)
- [Überwachungsseite \(p. 841\)](#)

Umgebungsübersicht

In der [Umgebungsübersicht \(p. 427\)](#) werden der [Integritätsstatus \(p. 841\)](#) der Umgebung angezeigt und Ereignisse aufgelistet, die Informationen zu aktuellen Änderungen in Bezug auf den Integritätsstatus bereitstellen.

So zeigen Sie die Umgebungsübersicht an

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

Ausführliche Informationen zur aktuellen Integrität der Umgebung finden Sie auf der Seite Health (Integrität) durch Auswahl von Causes (Ursachen). Alternativ können Sie im Navigationsbereich die Option Health (Integrität) auswählen.

Elastic Beanstalk > Environments > GettingStartedApp-env

GettingStartedApp-env
GettingStartedApp-env.bx7dx222kw.us-east-2.elasticbeanstalk.com

Health
Severe
Causes

Running version
Sample Application-2
Upload and deploy

Plat...
Tomcat 8.5 with J...
64bit Amazon...
Ch...

Recent events

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 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.

Seite „Umgebungsintegrität“

Die Seite Health (Integrität) zeigt die Integrität, Metriken und Ursachen für die Umgebung und für jede Amazon EC2-Instance in der Umgebung an.

Note

Elastic Beanstalk zeigt die Seite Health (Integrität) nur dann an, wenn Sie die [erweiterte Integritätsüberwachung für die Umgebung aktiviert haben \(p. 832\)](#).

Die folgende Abbildung zeigt die Seite Health (Integrität) in einer Linux-Umgebung.

Enhanced health overview											
Instances: 2 Total, 2 Ok											
Learn more about enhanced health .											
Instance ID	Status	Running	Deployment ID	Requests/sec	2xx Responses	3xx Responses	4xx Responses	5xx Responses	P99 Latency	P90 Latency	P50 Latency
Overall	Ok	N/A	N/A	0.4	100%	0.0%	0.0%	0.0%	0.002	0.00	0.00
i-0227807c4c4a1334	Ok	2 hours	3	0.2	2	0	0	0	0.002	0.002	0.002
i-03280193ba1ba4171	Ok	19 days	3	0.2	2	0	0	0	0.001	0.001	0.001

Die folgende Abbildung zeigt die Seite Health (Integrität) für eine Windows-Umgebung. Beachten Sie, dass sich CPU-Metriken von denen in einer Linux-Umgebung unterscheiden.

Enhanced health overview											
Instances: 1 Total, 1 Ok											
Learn more about enhanced health .											
Instance ID	Status	Running	Deployment ID	Requests/sec	2xx Responses	3xx Responses	4xx Responses	5xx Responses	P99 Latency	P90 Latency	P50 Latency
Overall	Ok	N/A	N/A	0.2	100%	0.0%	0.0%	0.0%	0.015	0.015	0.015
i-04b33b4c983018af	Ok	20 days	1	0.2	2	0	0	0	0.015	0.015	0.015

Oben auf der Seite können Sie die Gesamtzahl der Umgebungs-Instances sowie die Anzahl der Instances pro Status anzeigen. Wenn Sie nur Instances mit einem bestimmten Status anzeigen möchten, wählen Sie Filter By (Filtern nach) und dann einen [Status \(p. 841\)](#) aus.

Filter by ▲	
	Instance actions
	Remove Filter
Ok	
Pending	
Info	CPU utilization
Unknown	Idle%
No data	N/A
Warning	99.8
Warning or worse	
Degraded	
Degraded or worse	99.8
Severe	

Um eine fehlerhafte Instance neu zu starten oder zu beenden, klicken Sie auf Instance Actions (Instanz-Aktionen) und anschließend auf Reboot (Neu starten) oder Terminate (Beenden).

Instance actions ▲
Reboot
Terminate
Details

Elastic Beanstalk aktualisiert die Seite Health (Integrität) alle 10 Sekunden. Es werden Informationen zur Integrität von Umgebung und Instance angezeigt.

Für jede Amazon EC2-Instance in der Umgebung werden auf der Seite die ID und der [Status \(p. 841\)](#) der Instance angezeigt, die seit dem Start der Instance verstrichene Zeit, die ID der zuletzt ausgeführten Bereitstellung auf der Instance, die Antworten und Latenz der Anforderungen, die von der Instance erfüllt wurden, sowie Informationen zu CPU-Auslastung und -Nutzung. In der Zeile Overall (Gesamt) werden Informationen zur durchschnittlichen Antwortzeit und Latenz für die gesamte Umgebung angezeigt.

Die Seite zeigt zahlreiche Details in einer sehr breiten Tabelle an. Um Spalten auszublenden, wählen Sie  (Preferences (Einstellungen)) aus. Markieren oder löschen Sie Spaltennamen und klicken Sie anschließend auf Confirm (Bestätigen).

Preferences

Options

Visible columns

Instance ID

Status

Running

Deployment ID

Requests/sec

2xx Responses

3xx Responses

Wählen Sie die Instance ID (Instance-ID) einer Instance aus, um weitere Informationen zur Instance anzuzeigen, einschließlich Availability Zone und Instance-Typ.

Instance ID	Status	Running	Deployment ID	Request
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

Wählen Sie die Deployment ID (Bereitstellungs-ID) einer Instance aus, um Informationen zur letzten Bereitstellung (p. 471) auf der Instance anzuzeigen.

The screenshot shows a table of instance status and deployment history. An arrow points from the 'Deployment ID' column of the third row to a detailed view below. The detailed view shows 'Deployment ID 3' with a 'Version: Sample Application-3' and a green 'Deployed' button indicating it was deployed '1 day ago'.

Instance ID	Status	Running	Deployment ID	Request
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

Zu den Bereitstellungsinformationen gehören:

- Deployment ID (Bereitstellungs-ID) – Die eindeutige ID für die [Bereitstellung \(p. 471\)](#). Bereitstellungs-IDs beginnen bei 1 und nehmen jedes Mal um eine Zahl zu, wenn Sie eine neue Anwendungsversion bereitstellen oder Konfigurationseinstellungen ändern, die Auswirkungen auf die Software oder das Betriebssystem auf den Instances in Ihrer Umgebung haben.
- Version – Die Versionsbezeichnung des Anwendungsquellcodes, der in der Bereitstellung verwendet wird.
- Status – Der Status der Bereitstellung, der `In Progress`, `Deployed` oder `Failed` lauten kann.
- Time (Zeit) – Für angefangene Bereitstellungen ist dies die Zeit, zu der die Bereitstellung gestartet wurde. Für abgeschlossene Bereitstellungen ist dies die Zeit, zu der die Bereitstellung beendet wurde.

Wenn Sie die [X-Ray-Integration für Ihre Umgebung aktivieren \(p. 627\)](#) und Ihre Anwendung mit dem AWS X-Ray SDK instrumentieren, fügt die Seite Health (Integrität) Links zur AWS X-Ray-Konsole in der Übersichtszeile hinzu.

The screenshot shows a table of response statistics. The first row is highlighted with a yellow box. The columns are: Requests/sec, 2xx Responses, 3xx Responses, 4xx Responses, 5xx Responses, P99 Latency, P90 Latency, P75 Latency, and P50 Latency. The data for the first row is: 100%, 0.0%, 0.0%, 0.0%, 0.002, 0.002, 0.002, 0.002, and 0.002.

Requests/sec	2xx Responses	3xx Responses	4xx Responses	5xx Responses	P99 Latency	P90 Latency	P75 Latency	P50 Latency
100%	0.0%	0.0%	0.0%	0.002	0.002	0.002	0.002	0.002
1	0	0	0	0.002	0.002	0.002	0.002	0.002
1	0	0	0	0.001	0.001	0.001	0.001	0.001

Wählen Sie einen Link, um Ablaufverfolgungen im Zusammenhang mit der markierten Statistik in der AWS X-Ray-Konsole anzuzeigen.

Überwachungsseite

Die Seite Monitoring (Überwachung) zeigt zusammenfassende Statistiken und Diagramme für die benutzerdefinierten Amazon CloudWatch-Metriken an, die vom System für erweiterte Integritätsberichte generiert werden. Unter [Überwachen des Umgebungszustands in der AWS Managementkonsole \(p. 817\)](#) finden Sie Anweisungen zum Hinzufügen von Diagrammen und Statistiken zu dieser Seite.

Farben und Status in Zustandsangaben

Erweiterte Zustandsberichte zeigen den Zustand von Instances und der Umgebung insgesamt mit vier Farben an, ähnlich wie bei [grundlegenden Zustandsberichten \(p. 821\)](#). Erweiterte Zustandsberichte bieten außerdem sieben Zustandsstatus, wobei es sich einzelne Wortdeskriptoren handelt, die eine bessere Angabe des Zustands Ihrer Umgebung bieten.

Instance-Status und Umgebungsstatus

Jedes Mal, wenn Elastic Beanstalk eine Integritätsprüfung für Ihre Umgebung ausführt, wird mit den erweiterten Integritätsberichten die Integrität jeder Instance in Ihrer Umgebung durch die Analyse der [verfüglichen Daten \(p. 827\)](#) überprüft. Wenn eine der untergeordneten Prüfungen fehlschlägt, setzt Elastic Beanstalk die Integrität der Instance herunter.

Elastic Beanstalk zeigt die Integritätsinformationen für die gesamte Umgebung (Farbe, Status und Ursache) in der [Environment Management Console \(p. 426\)](#) an. Diese Informationen sind auch in der EB CLI verfügbar. Status- und Ursachenmeldungen für einzelne Instances werden alle 10 Sekunden aktualisiert und sind in der [EB CLI \(p. 1008\)](#) verfügbar, wenn Sie den Status mit `eb health` (p. 1033) anzeigen.

Elastic Beanstalk nutzt Änderungen bei der Instance-Integrität, um die Umgebungsintegrität zu evaluieren, ändert jedoch nicht sofort den Umgebungsintegritätsstatus. Wenn eine Instance die Integritätsprüfungen mindestens drei Mal in einem 1-Minuten-Zeitraum nicht besteht, kann Elastic Beanstalk die Integrität der Umgebung herabsetzen. Abhängig von der Anzahl der Instances in der Umgebung und dem identifizierten Problem kann eine fehlerhafte Instance dazu führen, dass Elastic Beanstalk eine Informationsmeldung anzeigt oder den Integritätsstatus der Umgebung von Grün (OK) in Gelb (Warning (Warnung)) oder Rot (Degraded (Schwach)) oder Severe (Stark)) ändert.

OK (Grün)

Dieser Status wird angezeigt, wenn:

- Eine Instance besteht die Zustandsprüfungen und der Zustandsagent meldet keine Probleme.
- Die meisten Instances in der Umgebung bestehen die Zustandsprüfungen und der Zustandsagent meldet keine wesentlichen Probleme.
- Eine Instance besteht Zustandsprüfungen und führt Anfragen normal aus.

Beispiel: Ihre Umgebung wurde kürzlich bereitgestellt und führt Anfragen normal aus. 5 % der Anfragen geben Serienfehler 400 zurück. Die Bereitstellung wurde auf jeder Instance normal abgeschlossen.

Nachricht (Instance): Anwendungsbereitstellung vor 23 Sekunden abgeschlossen (Dauer: 26 Sekunden).

Warning (Gelb)

Dieser Status wird angezeigt, wenn:

- Der Zustandsagent meldet eine geringe Anzahl an Anfragefehlern oder andere Probleme für eine Instance oder Umgebung.
- Ein Vorgang ist auf einer Instance in Bearbeitung und dauert sehr lange.

Beispiel: Eine Instance in der Umgebung hat den Status Severe (Stark).

Nachricht (Umgebung): Eingeschränkte Services bei 1 von 5 Instances.

Degraded (Rot)

Dieser Status wird angezeigt, wenn der Zustandsagent eine große Anzahl an Anfragefehlern oder andere Probleme für eine Instance oder Umgebung meldet.

Beispiel: Umgebung wird auf bis zu 5 Instance aufwärts skaliert.

Nachricht (Umgebung): 4 aktive Instances liegen unter der Mindestgröße für Auto Scaling-Gruppen von 5.

Severe (Rot)

Dieser Status wird angezeigt, wenn der Zustandsagent eine sehr große Anzahl an Anfragefehlern oder andere Probleme für eine Instance oder Umgebung meldet.

Beispiel: Elastic Beanstalk kann den Load Balancer nicht zum Abrufen der Instance-Integrität kontaktieren.

Nachricht (Umgebung): ELB-Zustandsprüfung schlägt fehl oder ist nicht für alle Instances verfügbar. Keine der Instances sendet Daten. Die Rolle „arn:aws:iam::123456789012:role/aws-elasticbeanstalk-service-role“ kann nicht angenommen werden. Überprüfen Sie, ob die Rolle vorhanden ist und richtig konfiguriert wurde.

Nachricht (Instances): Instance-ELB-Zustandsprüfung war 37 Minuten nicht verfügbar. Keine Daten. Zuletzt vor 37 Minuten gesehen.

Info (Grün)

Dieser Status wird angezeigt, wenn:

- Eine Operation wird auf einer Instance ausgeführt.
- Eine Operation wird auf mehreren Instances in einer Umgebung ausgeführt.

Beispiel: Eine neue Anwendungsversion wird für laufende Instances bereitgestellt.

Nachricht (Umgebung): Befehl wird auf 3 von 5 Instances ausgeführt.

Nachricht (Instance): Durchführen der Anwendungsbereitstellung (drei Sekunden lang ausgeführt).

Pending (Grau)

Dieser Status wird angezeigt, wenn auf einer Instance innerhalb des [Befehls-Timeout \(p. 828\)](#) eine Operation ausgeführt wird.

Beispiel: Sie haben kürzlich die Umgebung erstellt und Instances werden per Bootstrapping bereitgestellt.

Nachricht: Durchführen der Initialisierung (12 Sekunden lang ausgeführt).

Unknown (Grau)

Dieser Status wird angezeigt, wenn Elastic Beanstalk und der Integritäts-Agent eine unzureichende Anzahl an Daten für eine Instance melden.

Beispiel: Keine Daten werden empfangen.

Suspended (Grau)

Dieser Status wird angezeigt, wenn Elastic Beanstalk die Überwachung der Integrität der Umgebung gestoppt hat. Die Umgebung funktioniert möglicherweise nicht korrekt. Einige gravierende Integritätsbedingungen bewirken bei längerer Dauer, dass Elastic Beanstalk die Umgebung in den Status Suspended (Gesperrt) versetzt.

Beispiel: Elastic Beanstalk kann nicht auf die [Servicerolle \(p. 909\)](#) der Umgebung zugreifen.

Beispiel: Die von Elastic Beanstalk für die Umgebung erstellte [Auto Scaling-Gruppe \(p. 540\)](#) wurde gelöscht.

Nachricht: Der Umgebungszustand hat von OK zu Severe (Stark) gewechselt. Es sind keine Instances vorhanden. Die gewünschte Kapazität für die Auto Scaling-Gruppe ist auf 1 gesetzt.

Instance-Metriken

Instance-Metriken bieten Informationen über den Zustand der Instances in Ihrer Umgebung. Der [Elastic Beanstalk-Integritäts-Agent \(p. 827\)](#) wird auf jeder Instance ausgeführt. Er erfasst Metriken zu Instances und leitet diese an Elastic Beanstalk weiter. Dort erfolgt die Analyse der Metriken, um die Integrität der Instances in Ihrer Umgebung zu bestimmen.

Der Elastic Beanstalk-Integritäts-Agent für die Instance erfasst Metriken zu Instances von Webservern und dem Betriebssystem. Um Informationen vom Webserver auf Linux-basierten Plattformen zu erhalten, liest und analysiert Elastic Beanstalk Web-Server-Protokolle. Auf der Windows Server-Plattform erhält Elastic Beanstalk diese Informationen direkt von dem IIS-Webserver. Webserver bieten Informationen zu eingehenden HTTP-Anforderungen, z. B. wie viele Anforderungen eingegangen sind, wie viele davon zu Fehlern führten und wie lange die Verarbeitung gedauert hat. Das Betriebssystem bietet Snapshot-Informationen zum Status der Instance-Ressourcen, wie z. B. CPU-Auslastung und Verteilung der aufgewendeten Zeit je Prozesstyp.

Der Integritäts-Agent erfasst Webserver- und Betriebssystemmetriken und leitet diese in Intervallen von 10 Sekunden an Elastic Beanstalk weiter. Elastic Beanstalk analysiert die Daten und verwendet die Ergebnisse, um den Integritätsstatus für die einzelnen Instances und die Umgebung zu aktualisieren.

Themen

- [Webserver-Metriken \(p. 843\)](#)
- [Betriebssystemmetriken \(p. 844\)](#)
- [Erfassung von Webserver-Metriken in IIS auf Windows Server \(p. 845\)](#)

Webserver-Metriken

Auf Linux-basierten Plattformen liest der Elastic Beanstalk-Integritäts-Agent die Webservermetriken aus den Protokollen aus, die vom Webcontainer oder Server, der die Anforderungen für alle Instances der Umgebung verarbeitet, generiert werden. Elastic Beanstalk-Plattformen sind so konfiguriert, dass zwei Protokolle generiert werden, nämlich ein visuell lesbares und ein maschinenlesbares Format. Der Integritäts-Agent leitet die maschinenlesbaren Protokolle alle 10 Sekunden an Elastic Beanstalk weiter.

Weitere Informationen zum Protokollformat von Elastic Beanstalk finden Sie unter [Format der Protokolle der erweiterten Zustandsberichte \(p. 856\)](#).

Auf der Windows Server-Plattform fügt Elastic Beanstalk ein Modul zur Anfrage-Pipeline des IIS-Webservers hinzu und erfasst Metriken über HTTP-Anfragezeiten und Antwortcodes. Das Modul sendet diese Metriken über einen leistungsfähigen IPC-Kanal (Inter-Process Communication) an den On-Instance-

Zustandsagenten. Weitere Informationen zur Implementierung finden Sie in [Erfassung von Webserver-Metriken in IIS auf Windows Server \(p. 845\)](#).

Gemeldete Webservermetriken

RequestCount

Anzahl der vom Webserver pro Sekunde verarbeiteten Anforderungen in den letzten zehn Sekunden. Dargestellt als Durchschnittswert r/sec (Anfragen pro Sekunde) in der EB CLI und der Seite „Umgebungsintegrität“ (p. 836).

Status2xx, Status3xx, Status4xx, Status5xx

Anzahl der Anforderungen je Statuscode in den letzten zehn Sekunden. Beispielsweise geben erfolgreiche Anforderungen den Code "200 OK" und Weiterleitungen den Code "301" zurück. Falls die eingegebene URL mit keiner Ressource in der Anwendung übereinstimmt, wird der Code "404" zurückgegeben.

In der EB CLI und der Seite „Umgebungsintegrität“ (p. 836) werden diese Metriken sowohl als reine Anzahl der Anforderungen für Instances als auch als Prozentsatz der generellen Anforderungen für die Umgebungen dargestellt.

p99.9, p99, p95, p90, p85, p75, p50, p10

Durchschnittliche Latenzzeit der am langsamsten verarbeiteten x Prozent an Anforderungen in den letzten zehn Sekunden (wobei x die Differenz zwischen der Ziffer und 100 angibt). Beispielsweise gibt p99.1.403 an, dass für die am langsamsten verarbeiteten 1 Prozent der Anforderungen in den letzten zehn Sekunden die durchschnittliche Latenz 1,403 Sekunden betrug.

Betriebssystemmetriken

Der Elastic Beanstalk-Integritäts-Agent meldet die folgenden Betriebssystemmetriken. Elastic Beanstalk verwendet diese Metriken, um Instances zu ermitteln, bei denen die Auslastung durchgängig hoch ist. Die Metriken unterscheiden sich je nach Betriebssystem.

Gemeldete Betriebssystemmetriken – Linux

Running

Der seit dem Instance-Start verstrichene Zeitraum.

Load 1, Load 5

Durchschnittliche Auslastung in den letzten Ein-Minuten- und Fünf-Minuten-Zeiträumen. Dieser Dezimalwert gibt die durchschnittliche Anzahl der in diesem Zeitraum ausgeführten Prozesse an. Falls die angezeigte Zahl höher ist als die Anzahl verfügbarer vCPUs (Threads), gibt der Rest die durchschnittliche Anzahl der wartenden Prozesse an.

Wenn beispielsweise der Instance-Typ über vier vCPUs verfügt und die Auslastung bei 4,5 liegt, haben durchschnittlich 0,5 Prozesse in diesem Zeitraum gewartet – das heißt, ein Prozess hat 50 Prozent der Zeit gewartet.

User %, Nice %, System %, Idle %, I/O Wait %

Prozentangabe des Zeitraums, den die CPU in den letzten zehn Sekunden im jeweiligen Status gewesen ist.

Gemeldete Betriebssystemmetriken – Windows

Running

Der seit dem Instance-Start verstrichene Zeitraum.

% User Time, % Privileged Time, % Idle Time

Prozentangabe des Zeitraums, den die CPU in den letzten zehn Sekunden im jeweiligen Status gewesen ist.

Erfassung von Webserver-Metriken in IIS auf Windows Server

Auf der Windows Server-Plattform fügt Elastic Beanstalk ein Modul zur Anfrage-Pipeline des IIS-Webservers hinzu und erfasst Metriken über HTTP-Anfragezeiten und Antwortcodes. Das Modul sendet diese Metriken über einen leistungsfähigen IPC-Kanal (Inter-Process Communication) an den On-Instance-Zustandsagenten. Der Integritäts-Agent aggregiert diese Metriken, kombiniert sie mit Metriken des Betriebssystems und sendet sie an den Elastic Beanstalk-Service.

Implementierungsinformationen

Zum Erfassen von Metriken von IIS implementiert Elastic Beanstalk ein verwaltetes [IHttpModule](#) und abonniert die Ereignisse [BeginRequest](#) und [EndRequest](#). Dies ermöglicht es dem Modul, HTTP-Anfragelatenz und Antwortcodes für alle vom IIS bearbeiteten Webanfragen zu melden. Um das Modul der IIS-Anfrage-Pipeline hinzuzufügen, registriert Elastic Beanstalk das Modul im [`<modules>`](#)-Abschnitt der IIS-Konfigurationsdatei `%windir%\System32\inetsrv\config\applicationHost.config`.

Das Elastic Beanstalk-Modul in IIS sendet die erfassten Webanfragemetriken an den Integritäts-Agenten für die Instance. Dabei handelt es sich um einen Windows-Service mit dem Namen `HealthD`. Um diese Daten zu senden, verwendet das Modul [NetNamedPipeBinding](#), die eine sichere und zuverlässige Bindung bietet, die für die Kommunikation auf der Maschine optimiert ist.

Konfigurieren von Regeln für den erweiterten Zustand einer Umgebung

Erweiterte Integritätsberichte von AWS Elastic Beanstalk basieren auf mehreren Regeln, um die Integrität Ihrer Umgebung zu bestimmen. Einige dieser Regeln sind möglicherweise nicht für Ihre jeweilige Anwendung geeignet. Im Folgenden finden Sie einige häufig verwendete Beispiele:

- Sie verwenden clientseitige Testwerkzeuge. In diesem Fall werden häufige HTTP-Clientfehler (4xx) erwartet.
- Sie verwenden [AWS WAF](#) in Verbindung mit dem Application Load Balancer Ihrer Umgebung, um unerwünschten eingehenden Datenverkehr zu blockieren. In diesem Fall gibt der Application Load Balancer für jede abgelehnte eingehende Nachricht „HTTP 403“ zurück.

Standardmäßig berücksichtigt Elastic Beanstalk alle HTTP 4xx-Fehler von Anwendungen, wenn die Integrität der Umgebung ermittelt wird. Abhängig von der Fehlerquote ändert sich der Zustandsstatus der Umgebung von OK in Warning (Warnung), Degraded (Beeinträchtigt) oder Severe (Schwerwiegend). Damit Fälle, wie die erwähnten Beispiele, korrekt gehandhabt werden, können Sie mit Elastic Beanstalk einige Regeln für die erweiterte Integrität konfigurieren. Sie können wählen, ob HTTP 4xx-Fehler von Anwendungen auf Instances der Umgebung oder ob HTTP 4xx-Fehler, die vom Load Balancer der Umgebung zurückgegeben werden, ignoriert werden sollen. In diesem Thema wird beschrieben, wie diese Konfigurationsänderungen vorgenommen werden.

Note

Derzeit sind dies die einzigen verfügbaren Anpassungen von Regeln zum erweiterten Zustand. Die erweiterte Integrität kann nicht so konfiguriert werden, dass andere HTTP-Fehler zusätzlich zu 4xx ignoriert werden.

Konfigurieren von Regeln für die erweiterte Integrität mit der Elastic Beanstalk-Konsole

Sie können die Elastic Beanstalk-Konsole verwenden, um Regeln für die erweiterte Integrität in einer Umgebung zu konfigurieren.

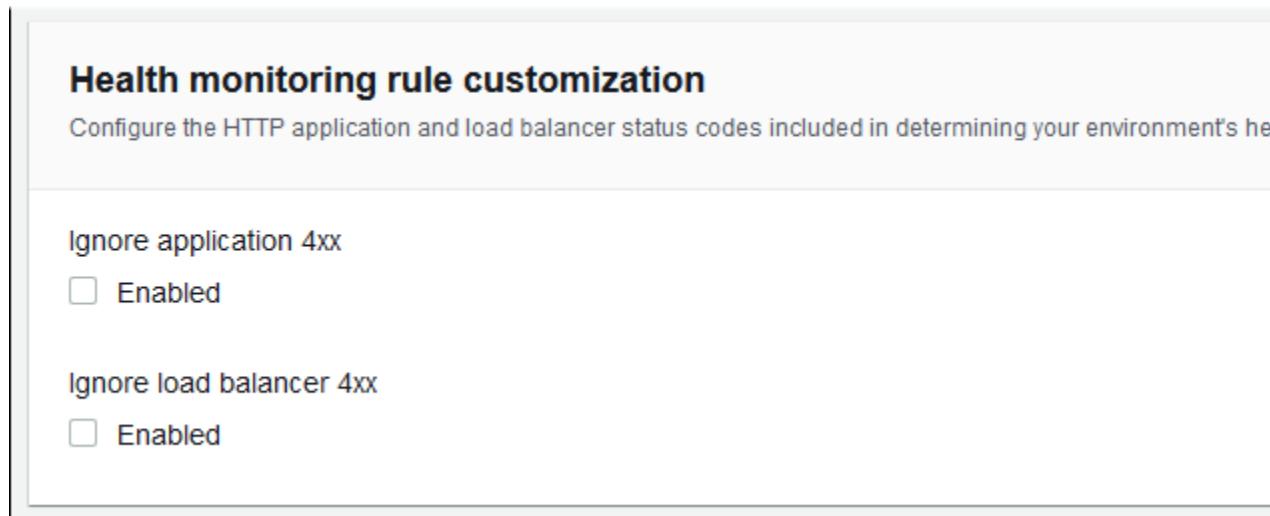
So konfigurieren Sie die HTTP 4xx-Statuscodeprüfung mithilfe der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Monitoring (Überwachung) die Option Edit (Bearbeiten).
5. Aktivieren oder deaktivieren Sie unter Anpassung der Integritätsüberwachungsregel die gewünschten Ignore (Ignorieren)-Optionen.



6. Wählen Sie Apply (Anwenden) aus.

Konfigurieren von erweiterten Zustandsregeln unter Verwendung der EB CLI

Sie können mit der EB CLI Regeln für die erweiterte Integrität konfigurieren, indem Sie die Konfiguration Ihrer Umgebung lokal speichern, einen Eintrag hinzufügen, der Regeln für die erweiterte Integrität konfiguriert, und die Konfiguration dann in Elastic Beanstalk hochladen. Sie können die gespeicherte Konfiguration zu einer Umgebung während oder nach der Erstellung hinzufügen.

So konfigurieren Sie die HTTP 4xx-Statuscodeprüfung mit der EB CLI und gespeicherten Konfigurationen

1. Initialisieren Sie Ihren Projektordner mit [eb init \(p. 1017\)](#).

2. Erstellen Sie eine Umgebung, indem Sie den [eb create \(p. 1020\)](#)-Befehl ausführen.
3. Speichern Sie eine Konfigurationsvorlage lokal, indem Sie den eb config save-Befehl ausführen. Im folgenden Beispiel wird die --cfg-Option verwendet, um den Namen der Konfiguration anzugeben.

```
$ eb config save --cfg 01-base-state
Configuration saved at: ~/project/.elasticbeanstalk/saved_configs/01-base-state.cfg.yml
```

4. Öffnen Sie die gespeicherte Konfigurationsdatei in einem Texteditor.
5. Fügen Sie unter OptionSettings > aws:elasticbeanstalk:healthreporting:system: einen ConfigDocument-Schlüssel hinzu, um alle Regeln für den erweiterten Zustand aufzulisten, die konfiguriert werden können. Im Folgenden wird mit ConfigDocument die Überprüfung der HTTP 4xx-Statuscodes von Anwendungen deaktiviert, während die Überprüfung des HTTP 4xx-Codes des Load Balancer aktiviert bleibt.

```
OptionSettings:
  ...
  aws:elasticbeanstalk:healthreporting:system:
    ConfigDocument:
      Rules:
        Environment:
          Application:
            ApplicationRequests4xx:
              Enabled: false
        ELB:
          ELBRequests4xx:
            Enabled: true
      Version: 1
      SystemType: enhanced
  ...
```

Note

Sie können Rules und CloudWatchMetrics in derselben ConfigDocument-Optionseinstellung kombinieren. CloudWatchMetrics sind in [Veröffentlichen von benutzerdefinierten Amazon CloudWatch-Metriken für eine Umgebung \(p. 849\)](#) beschrieben.

Wenn Sie zuvor CloudWatchMetrics aktiviert haben, hat die Konfigurationsdatei, die Sie mit dem Befehl eb config save abrufen, bereits einen ConfigDocument-Schlüssel mit einem CloudWatchMetrics-Abschnitt. Löschen Sie ihn nicht – fügen Sie einen Rules-Abschnitt in denselben ConfigDocument-Optionswert ein.

6. Speichern Sie die Konfigurationsdatei und schließen Sie den Text-Editor. Bei diesem Beispiel wird die aktualisierte Konfigurationsdatei mit einem Namen (02-cloudwatch-enabled.cfg.yml) gespeichert, der sich von dem der heruntergeladenen Konfigurationsdatei unterscheidet. Dadurch wird eine separat gespeicherte Konfiguration erstellt, wenn die Datei hochgeladen wird. Sie können denselben Namen wie die heruntergeladene Datei verwenden, um die vorhandene Konfiguration zu überschreiben, ohne dass Sie eine neue erstellen müssen.
7. Verwenden Sie zum Hochladen der aktualisierten Konfigurationsdatei in Elastic Beanstalk den Befehl eb config put.

```
$ eb config put 02-cloudwatch-enabled
```

Schließen Sie die Dateinamenerweiterung bei Verwendung der Befehle eb config get und put mit gespeicherten Konfigurationen nicht mit ein.

8. Wenden Sie die gespeicherte Konfiguration auf Ihre laufende Umgebung an.

```
$ eb config --cfg 02-cloudwatch-enabled
```

Die `--cfg`-Option gibt eine benannte Konfigurationsdatei an, die auf die Umgebung angewendet wird. Sie können die Konfigurationsdatei lokal oder in Elastic Beanstalk speichern. Wenn eine Konfigurationsdatei mit dem angegebenen Namen in beiden Speicherorten vorhanden ist, verwendet die EB CLI die lokale Datei.

Konfigurieren von erweiterten Zustandsregeln unter Verwendung eines Konfigurationsdokuments

Das Konfigurationsdokument (config) für Regeln für erweiterte Zustände ist ein JSON-Dokument, das die zu konfigurierenden Regeln auflistet.

Das folgende Beispiel zeigt ein Konfigurationsdokument, das die Überprüfung von HTTP 4xx-Statuscodes von Anwendungen deaktiviert und die Überprüfung von HTTP 4xx-Statuscodes für Load Balancer aktiviert.

```
{
  "Rules": {
    "Environment": {
      "Application": {
        "ApplicationRequests4xx": {
          "Enabled": false
        }
      },
      "ELB": {
        "ELBRequests4xx": {
          "Enabled": true
        }
      }
    },
    "Version": 1
  }
}
```

Für die AWS CLI müssen Sie das Dokument als ein Wert für den `Value`-Schlüssel in ein Optionseinstellungsargument übergeben, das selbst ein JSON-Objekt ist. In diesem Fall müssen Anführungszeichen im eingebetteten Dokument durch Escape-Zeichen geschützt werden. Mit dem folgenden Befehl wird überprüft, ob die Konfigurationseinstellungen gültig sind.

```
$ 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 }"
  }
]'
```

Für eine `.ebextensions`-Konfigurationsdatei in YAML können Sie das JSON-Dokument unverändert bereitstellen.

```
option_settings:
  - namespace: aws:elasticbeanstalk:healthreporting:system
    option_name: ConfigDocument
    value: {
      "Rules": {
        "Environment": {
```

```
    "Application": {
        "ApplicationRequests4xx": {
            "Enabled": false
        }
    },
    "ELB": {
        "ELBRequests4xx": {
            "Enabled": true
        }
    }
},
"Version": 1
}
```

Veröffentlichen von benutzerdefinierten Amazon CloudWatch-Metriken für eine Umgebung

Sie können die von den erweiterten AWS Elastic Beanstalk-Integritätsberichten gesammelten Daten in Amazon CloudWatch als benutzerdefinierte Metriken veröffentlichen. Durch das Veröffentlichen von Metriken in CloudWatch können Sie Änderungen in der Anwendungsleistung über den Zeitverlauf überwachen und potenzielle Probleme identifizieren, indem Sie verfolgen, wie die Ressourcennutzung und Anfragelatenz mit Last skaliert wird.

Durch das Veröffentlichen von Metriken in CloudWatch sind sie auch für die Verwendung mit [Überwachungsdiagrammen \(p. 818\)](#) und [Alarmen \(p. 861\)](#) verfügbar. Eine kostenlose Metrik, EnvironmentHealth, ist automatisch aktiviert, wenn Sie erweiterte Zustandsberichte verwenden. Für benutzerdefinierte Metriken außer EnvironmentHealth fallen Standard-[CloudWatch-Gebühren](#) an.

Zum Veröffentlichen von benutzerdefinierten CloudWatch-Metriken für eine Umgebung müssen Sie zunächst die erweiterten Integritätsberichte zur Umgebung aktivieren. Anweisungen finden Sie unter [Aktivieren der erweiterten Elastic Beanstalk-Integritätsberichte \(p. 832\)](#).

Themen

- [Metriken der erweiterten Zustandsberichte \(p. 849\)](#)
- [Konfigurieren von CloudWatch-Metriken mit der Elastic Beanstalk-Konsole \(p. 851\)](#)
- [Konfigurieren von benutzerdefinierten CloudWatch-Metriken mit der EB CLI \(p. 851\)](#)
- [Bereitstellen von benutzerdefinierten Metrikkonfigurations-Dokumenten \(p. 853\)](#)

Metriken der erweiterten Zustandsberichte

Wenn Sie erweiterte Integritätsberichte in Ihrer Umgebung aktivieren, veröffentlicht das entsprechende System automatisch eine [benutzerdefinierte CloudWatch-Metrik](#), EnvironmentHealth. Um zusätzliche Metriken in CloudWatch zu veröffentlichen, konfigurieren Sie die Umgebung unter Verwendung der [Elastic Beanstalk-Konsole \(p. 851\)](#), der [EB CLI \(p. 851\)](#) oder von [.ebextensions \(p. 646\)](#) mit den betreffenden Metriken.

Sie können die folgenden erweiterten Integritätsmetriken aus Ihrer Umgebung an CloudWatch veröffentlichen.

Verfügbare Metriken – alle Plattformen

EnvironmentHealth

Nur Umgebung. Dies ist die einzige CloudWatch-Metrik, die das System für erweiterte Integritätsberichte veröffentlicht, es sei denn, Sie konfigurieren zusätzliche Metriken. Der

Umgebungszustand wird durch einen von sieben [Status \(p. 841\)](#) dargestellt. In der CloudWatch-Konsole sind diese Status folgenden Werten zugeordnet:

- 0 – OK
- 1 – Info
- 5 – Unknown
- 10 – No data
- 15 – Warning
- 20 – Degraded
- 25 – Severe

`InstancesSevere, InstancesDegraded, InstancesWarning, InstancesInfo, InstancesOk, InstancesPending, InstancesUnknown, InstancesNoData`

Nur Umgebung. Diese Metriken geben die Anzahl der Instances in der Umgebung mit dem jeweiligen Zustand an. `InstancesNoData` gibt die Anzahl der Instances an, für die keine Daten empfangen wurden.

`ApplicationRequestsTotal, ApplicationRequests5xx, ApplicationRequests4xx, ApplicationRequests3xx, ApplicationRequests2xx`

Instance und Umgebung. Gibt die Gesamtanzahl der Anforderungen an, die von der Instance oder Umgebung abgeschlossen wurden, und die Anzahl der Anfragen, die mit jeder Statuscodekategorie abgeschlossen wurde.

`ApplicationLatencyP10, ApplicationLatencyP50, ApplicationLatencyP75, ApplicationLatencyP85, ApplicationLatencyP90, ApplicationLatencyP95, ApplicationLatencyP99, ApplicationLatencyP99.9`

Instance und Umgebung. Gibt die durchschnittliche Zeit in Sekunden an, die es dauert, bis die schnellsten x Prozent der Anfragen abgeschlossen wurden.

`InstanceHealth`

Nur Instance. Gibt den aktuellen Zustand der Instance an. Der Instance-Zustand wird durch einen von sieben [Status \(p. 841\)](#) dargestellt. In der CloudWatch-Konsole sind diese Status folgenden Werten zugeordnet:

- 0 – OK
- 1 – Info
- 5 – Unknown
- 10 – No data
- 15 – Warning
- 20 – Degraded
- 25 – Severe

Verfügbarer Metriken – Linux

`CPUIRQ, CPUIdle, CPUUser, CPUSystem, CPUSoftirq, CPUWait, CPUNice`

Nur Instance. Gibt eine Prozentangabe des Zeitraums an, den die CPU in der letzten Minute im jeweiligen Status gewesen ist.

`LoadAverage1min`

Nur Instance. Die durchschnittliche CPU-Auslastung der Instance innerhalb der letzten Minute.

RootFilesystemUtil

Nur Instance. Gibt den prozentualen Anteil des verwendeten Speicherplatzes an.

Verfügaren Metriken – Windows

CPUIdle, CPUUser, CPUPrileged

Nur Instance. Gibt eine Prozentangabe des Zeitraums an, den die CPU in der letzten Minute im jeweiligen Status gewesen ist.

Konfigurieren von CloudWatch-Metriken mit der Elastic Beanstalk-Konsole

Sie können die Elastic Beanstalk-Konsole verwenden, um die Umgebung so zu konfigurieren, dass Metriken der erweiterten Integritätsberichte in CloudWatch veröffentlicht und für die Verwendung mit Überwachungsdiagrammen und Alarmen verfügbar sind.

So konfigurieren Sie benutzerdefinierte CloudWatch-Metriken in der Elastic Beanstalk-Konsole

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Monitoring (Überwachung) die Option Edit (Bearbeiten).
5. Wählen Sie unter Health reporting (Integritätsberichte) die Instance- und Umgebungsmetriken aus, die Sie in CloudWatch veröffentlichen möchten. Zur Auswahl mehrerer Metriken drücken Sie die Strg-Taste während der Auswahl.
6. Wählen Sie Apply (Anwenden) aus.

Durch das Aktivieren von benutzerdefinierten CloudWatch-Metriken werden sie zur Metrikliste auf der [Seite Monitoring \(Überwachung\) \(p. 817\)](#) hinzugefügt.

Konfigurieren von benutzerdefinierten CloudWatch-Metriken mit der EB CLI

Sie können mit der EB CLI benutzerdefinierte Metriken konfigurieren, indem Sie die Konfiguration Ihrer Umgebung lokal speichern, einen Eintrag hinzufügen, der die Metriken für die Veröffentlichung definiert, und die Konfiguration dann in Elastic Beanstalk hochladen. Sie können die gespeicherte Konfiguration zu einer Umgebung während oder nach der Erstellung hinzufügen.

So konfigurieren Sie benutzerdefinierte CloudWatch-Metriken mit der EB CLI und gespeicherten Konfigurationen

1. Initialisieren Sie Ihren Projektordner mit [eb init \(p. 1017\)](#).
2. Erstellen Sie eine Umgebung, indem Sie den [eb create \(p. 1020\)](#)-Befehl ausführen.

3. Speichern Sie eine Konfigurationsvorlage lokal, indem Sie den eb config save-Befehl ausführen. Im folgenden Beispiel wird die --cfg-Option verwendet, um den Namen der Konfiguration anzugeben.

```
$ eb config save --cfg 01-base-state
Configuration saved at: ~/project/.elasticbeanstalk/saved_configs/01-base-state.cfg.yml
```

4. Öffnen Sie die gespeicherte Konfigurationsdatei in einem Texteditor.
5. Fügen Sie unter OptionSettings > aws:elasticbeanstalk:healthreporting:system: einen ConfigDocument-Schlüssel hinzu, um die einzelnen gewünschten CloudWatch-Metriken zu aktivieren. Wenn beispielsweise der folgende ConfigDocument ApplicationRequests5xx- und ApplicationRequests4xx -Metriken auf Umgebungsebene und ApplicationRequestsTotal-Metriken auf Instance-Ebene veröffentlicht.

```
OptionSettings:
  ...
  aws:elasticbeanstalk:healthreporting:system:
    ConfigDocument:
      CloudWatchMetrics:
        Environment:
          ApplicationRequests5xx: 60
          ApplicationRequests4xx: 60
        Instance:
          ApplicationRequestsTotal: 60
        Version: 1
      SystemType: enhanced
  ...
```

Im Beispiel gibt 60 die Anzahl der Sekunden zwischen Messungen an. Derzeit ist dies der einzige unterstützte Wert.

Note

Sie können CloudWatchMetrics und Rules in derselben ConfigDocument-Optionseinstellung kombinieren. Rules sind in [Konfigurieren von Regeln für den erweiterten Zustand einer Umgebung \(p. 845\)](#) beschrieben.

Wenn Sie zuvor Rules verwendet haben, um Regeln für den erweiterten Zustand zu konfigurieren, hat die Konfigurationsdatei, die Sie mit dem Befehl eb config save abrufen, bereits einen ConfigDocument-Schlüssel mit einem Rules-Abschnitt. Löschen Sie ihn nicht – fügen Sie einen CloudWatchMetrics-Abschnitt in denselben ConfigDocument-Optionswert ein.

6. Speichern Sie die Konfigurationsdatei und schließen Sie den Text-Editor. Bei diesem Beispiel wird die aktualisierte Konfigurationsdatei mit einem Namen (02-cloudwatch-enabled.cfg.yml) gespeichert, der sich von dem der heruntergeladenen Konfigurationsdatei unterscheidet. Dadurch wird eine separate gespeicherte Konfiguration erstellt, wenn die Datei hochgeladen wird. Sie können denselben Namen wie die heruntergeladene Datei verwenden, um die vorhandene Konfiguration zu überschreiben, ohne dass Sie eine neue erstellen müssen.
7. Verwenden Sie zum Hochladen der aktualisierten Konfigurationsdatei in Elastic Beanstalk den Befehl eb config put.

```
$ eb config put 02-cloudwatch-enabled
```

Schließen Sie die Dateierweiterung bei Verwendung der Befehle eb config get und put mit gespeicherten Konfigurationen nicht mit ein.

8. Wenden Sie die gespeicherte Konfiguration auf Ihre laufende Umgebung an.

```
$ eb config --cfg 02-cloudwatch-enabled
```

Die `--cfg`-Option gibt eine benannte Konfigurationsdatei an, die auf die Umgebung angewendet wird. Sie können die Konfigurationsdatei lokal oder in Elastic Beanstalk speichern. Wenn eine Konfigurationsdatei mit dem angegebenen Namen in beiden Speicherorten vorhanden ist, verwendet die EB CLI die lokale Datei.

Bereitstellen von benutzerdefinierten Metrikkonfigurations-Dokumenten

Das Konfigurationsdokument (config) für benutzerdefinierte Amazon CloudWatch-Metriken ist ein JSON-Dokument, das die Metriken zur Veröffentlichung auf den Umgebungs- und Instance-Ebenen auflistet. Das folgende Beispiel zeigt ein Konfigurationsdokument, das alle verfügbaren benutzerdefinierten Metriken aktiviert.

```
{  
    "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,  
            "CPUIidle": 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,  
            "CPUMemoryUtil": 60  
        }  
    }  
}
```

```

        "ApplicationRequests4xx": 60,
        "InstanceHealth": 60,
        "CPUSoftirq": 60
    }
},
"Version": 1
}

```

Für die AWS CLI müssen Sie das Dokument als ein Wert für den `Value`-Schlüssel in ein Optionseinstellungsargument übergeben, das selbst ein JSON-Objekt ist. In diesem Fall müssen Anführungszeichen im eingebetteten Dokument durch Escape-Zeichen geschützt werden.

```
$ 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, \"ApplicationLatencyP85\": 60, \"CPUUser\": 60, \"ApplicationRequests2xx\": 60, \"CPUIdle\": 60, \"ApplicationLatencyP50\": 60, \"ApplicationRequestsTotal\": 60, \"RootFilesystemUtil\": 60, \"LoadAverage1min\": 60, \"CPUIRQ\": 60, \"CPUNice\": 60, \"CPUWait\": 60, \"ApplicationLatencyP10\": 60, \"LoadAverage5min\": 60, \"ApplicationRequests5xx\": 60, \"ApplicationLatencyP75\": 60, \"CPUSystem\": 60, \"ApplicationRequests3xx\": 60, \"ApplicationRequests4xx\": 60, \"InstanceHealth\": 60, \"CPUSoftirq\": 60}}, \"Version\": 1}"
}
]'
```

Für eine `.ebextensions`-Konfigurationsdatei in YAML können Sie das JSON-Dokument unverändert bereitstellen.

```

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,
        "CPUIidle": 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
}
```

Verwenden der erweiterten Integritätsberichte mit der Elastic Beanstalk-API

Da die erweiterten Integritätsberichte von AWS Elastic Beanstalk sowohl Lösungs-Stack- als auch Rollenanforderungen aufweisen, müssen Skripts und Codezeilen, die vor der Veröffentlichung der erweiterten Integritätsberichte verwendet wurden, erst aktualisiert werden, bevor Sie die erweiterten Integritätsberichte einsetzen können. Um die Abwärtskompatibilität zu gewährleisten, ist bei einer Umgebungserstellung mithilfe der Elastic Beanstalk-API die Funktion der erweiterten Integritätsberichte standardmäßig deaktiviert.

Konfigurieren Sie die erweiterten Integritätsberichte, indem Sie die Servicerolle, das Instance-Profil und die Amazon CloudWatch-Konfigurationsoptionen für die Umgebung festlegen. Dafür gibt es drei Möglichkeiten: Sie legen die Konfigurationsoptionen im Ordner `.ebextensions` fest, Sie nutzen gespeicherte Konfigurationen oder Sie nehmen die Konfiguration direkt im Parameter `create-environment` des Aufrufs `option-settings` vor.

Folgende Schritte sind erforderlich, wenn Sie die API, SDKs oder die AWS CLI zum Erstellen einer Umgebung verwenden, die erweiterte Zustandsberichte unterstützt:

- Erstellen Sie eine Servicerolle und ein Instance-Profil mit den entsprechenden [Berechtigungen \(p. 21\)](#).
- Erstellen Sie eine neue Umgebung mit einer neuen [Plattformversion \(p. 32\)](#).
- Legen Sie die [Konfigurationsoptionen \(p. 646\)](#) für Zustandssystemtyp, Instance-Profil und Servicerolle fest.

Verwenden Sie die folgenden Konfigurationsoptionen in den Namespaces `aws:elasticbeanstalk:healthreporting:system`, `aws:autoscaling:launchconfiguration` und `aws:elasticbeanstalk:environment`, um die erweiterten Zustandsberichte in der Umgebung zu konfigurieren.

Konfigurationsoptionen für erweiterte Zustandsberichte

SystemType

Namespace: `aws:elasticbeanstalk:healthreporting:system`

Legen Sie den Wert auf **enhanced** fest, um die Funktion der erweiterten Zustandsberichte zu nutzen.

IamInstanceProfile

Namespace: `aws:autoscaling:launchconfiguration`

Geben Sie den Namen eines Instance-Profs an, das für die Verwendung mit Elastic Beanstalk konfiguriert ist.

ServiceRole

Namespace: `aws:elasticbeanstalk:environment`

Geben Sie den Namen einer Servicerolle an, die für die Verwendung mit Elastic Beanstalk konfiguriert ist.

ConfigDocument (optional)

Namespace: `aws:elasticbeanstalk:healthreporting:system`

Ein JSON-Dokument, das die Instance- und Umgebungsmetriken definiert, die in CloudWatch veröffentlicht werden sollen. Zum Beispiel:

```
{  
    "CloudWatchMetrics":  
    {  
        "Environment":  
        {  
            "ApplicationLatencyP99.9": 60,  
            "InstancesSevere": 60  
        }  
        "Instance":  
        {  
            "ApplicationLatencyP85": 60,  
            "CPUUser": 60  
        }  
    }  
    "Version": 1  
}
```

Note

Je nach der Bereitstellung für Elastic Beanstalk erfordern Config-Dokumente möglicherweise eine spezielle Formatierung (z. B. Escape-Anführungszeichen). Beispiele finden Sie unter [Bereitstellen von benutzerdefinierten Metrikkonfigurations-Dokumenten \(p. 853\)](#).

Format der Protokolle der erweiterten Zustandsberichte

AWS Elastic Beanstalk-Plattformen verwenden ein benutzerdefiniertes Webserver-Protokollformat, um Informationen über HTTP-Anforderungen effizient an das System für erweiterte Integritätsberichte

weiterzuleiten. Das System analysiert die Protokolle, identifiziert Probleme und legt den Instance- und Umgebungszustand entsprechend fest. Wenn Sie den Webserver-Proxy in Ihrer Umgebung deaktivieren und Anfragen direkt über den Webcontainer verarbeiten, können Sie erweiterte Integritätsberichte trotzdem vollumfänglich nutzen, indem Sie den Server für die Ausgabe von Protokollen in den Speicherort und im Format konfigurieren, den bzw. das der [Elastic Beanstalk-Integritäts-Agent \(p. 827\)](#) verwendet.

Note

Die Informationen auf dieser Seite gilt nur für Linux-basierte Plattformen. Auf der Windows Server-Plattform erhält Elastic Beanstalk diese Informationen über HTTP-Anfragen direkt von dem IIS-Webserver. Details dazu finden Sie unter [Erfassung von Webserver-Metriken in IIS auf Windows Server \(p. 845\)](#).

Konfiguration von Webserver-Protokollen

Elastic Beanstalk-Plattformen sind so konfiguriert, dass sie zwei Protokolle mit Informationen über HTTP-Anfragen ausgeben. Die erste ist im Verbose-Format und bietet detaillierte Informationen über die Anfrage, einschließlich der Benutzer-Agent-Informationen des Anforderers und eines lesbaren Zeitstempels.

/var/log/nginx/access.log

Im folgenden Beispiel wird ein nginx-Proxy auf einer Ruby-Webserverumgebung ausgeführt, das Format ist jedoch vergleichbar für 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"
```

Das zweite Protokoll ist im Terse-Format. Es enthält Informationen, die nur für erweiterte Zustandsberichte relevant sind. Dieses Protokoll wird in einen Unterordner mit dem Namen `healthd` ausgegeben und ständig rotiert. Alte Protokolle werden sofort nach der Rotation gelöscht.

/var/log/nginx/healthd/application.log.2015-07-23-00

Das folgende Beispiel zeigt ein Protokoll in einem für Computer lesbaren Format.

```
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
```

Das Protokollformat für erweiterte Zustandsberichte enthält die folgenden Informationen:

- Die Uhrzeit der Anfrage in Unix-Zeit
- Den Pfad der Anfrage

- Den HTTP-Statuscode für das Ergebnis
- Die Anfragezeit
- Die Upstream-Zeit
- Den X-Forwarded-For-HTTP-Header

Für nginx-Proxys werden Zeiten in Floating-Point-Sekunden mit drei Dezimalstellen gedruckt. Für Apache werden ganze Mikrosekunden verwendet.

Note

Wenn Sie eine Warnung ähnlich der Folgenden in einer Protokolldatei sehen, in der Datum und Uhrzeit DATE-TIME ist, und Sie einen benutzerdefinierten Proxy verwenden, z. B. wie in einer Multicontainer-Docker-Umgebung, müssen Sie eine .ebextension zur Konfiguration Ihrer Umgebung verwenden, sodass die healthd Ihre Protokolldateien lesen kann:

```
W, [DATE-TIME #1922] WARN -- : log file "/var/log/nginx/healthd/application.log.DATE-TIME" does not exist
```

Sie können mit .ebextension im [Beispiel für Multicontainer-Docker](#) beginnen.

/etc/nginx/conf.d/webapp_healthd.conf

Das folgende Beispiel zeigt die Protokollkonfiguration für nginx mit dem healthd-Protokollformat hervorgehoben.

```
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 above
        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

Im folgenden Beispiel wird die Protokollkonfiguration für Apache gezeigt.

```
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
```

Generieren von Protokollen für erweiterte Zustandsberichte

Um Protokolle für den Zustandsagenten bereitzustellen, müssen Sie die folgenden Schritte ausführen:

- Protokolle im korrekten Format ausgeben, wie im vorherigen Abschnitt gezeigt
- Protokolle in `/var/log/nginx/healthd/` ausgeben
- Protokolle mit folgendem Format benennen: `application.log.$year-$month-$day-$hour`
- Protokolle einmal pro Stunde rotieren
- Protokolle nicht kürzen

Benachrichtigungen und Fehlerbehebung

Diese Seite zeigt Beispiel-Ursachenmeldungen für häufige Probleme und Links zu weiteren Informationen. Ursachenmeldungen werden auf der [Umgebungsübersichtsseite \(p. 817\)](#) der Elastic Beanstalk-Konsole angezeigt und in [Ereignisse \(p. 864\)](#) aufgezeichnet, wenn Integritätsprobleme über mehrere Prüfungen bestehen.

Bereitstellungen

Elastic Beanstalk überwacht Ihre Umgebung nach Bereitstellungen auf Konsistenz. Wenn eine fortlaufende Bereitstellung fehlschlägt, kann die Version Ihrer Anwendung auf den Instances in Ihrer Umgebung variieren. Dies kann der Fall sein, wenn eine Bereitstellung auf einem oder mehreren Stapeln erfolgreich ist, aber vor allen abgeschlossenen Stapeln fehlschlägt.

Falsche Anwendungsversion auf zwei von fünf Instances gefunden. Erwartete Version "v1" (Bereitstellung 1).

Falsche Anwendungsversion auf Umgebungs-Instances. Erwartete Version "v1" (Bereitstellung 1).

Die erwartete Anwendungsversion wird auf einigen oder allen Instances in einer Umgebung nicht ausgeführt.

Falsche Anwendungsversion "v2" (Bereitstellung 2). Erwartete Version "v1" (Bereitstellung 1).

Die Anwendung, die auf einer Instance bereitgestellt wurde, unterscheidet sich von der erwarteten Version. Wenn eine Bereitstellung fehlschlägt, wird die erwartete Version auf die Version aus der letzten erfolgreichen Bereitstellung zurückgesetzt. Im vorangegangenen Beispiel war die erste Bereitstellung

(Version "v1") erfolgreich, aber die zweite Bereitstellung (Version "v2") ist fehlgeschlagen. Alle Instances mit "v2" werden als fehlerhaft angesehen.

Um dieses Problem zu lösen, starten Sie eine andere Bereitstellung. Sie können [eine frühere Version erneut bereitstellen \(p. 471\)](#), von der Sie wissen, dass sie funktioniert, oder Ihre Umgebung so konfigurieren, dass [Zustandsprüfungen \(p. 475\)](#) während der Bereitstellung ignoriert werden und die neue Version erneut bereitgestellt wird, um einen Abschluss der Bereitstellung zu erzwingen.

Sie können auch die Instances identifizieren und beenden, die mit der falschen Anwendungsversion ausgeführt werden. Elastic Beanstalk startet Instances mit der richtigen Version, um alle Instances zu ersetzen, die Sie beenden. Verwenden Sie den [EB CLI-Zustandsbefehl \(p. 1033\)](#), um Instances zu identifizieren, die mit der falschen Anwendungsversion ausgeführt werden.

Anwendungsserver

15 % der Abfragen geben den Fehler HTTP 4xx zurück

20 % der Abfragen an den ELB geben den Fehler HTTP 4xx zurück.

Ein hoher Prozentsatz der HTTP-Anfragen an eine Instance oder eine Umgebung schlagen mit 4xx-Fehlern fehl.

Ein Serienstatuscode 400 gibt an, dass der Benutzer eine fehlerhafte Anfrage getätigt hat, beispielsweise die Anfrage einer Seite, die nicht vorhanden ist (404 File Not Found) oder auf die der Benutzer keinen Zugriff hat (403 Forbidden). Eine geringe Anzahl von 404s ist nicht ungewöhnlich, aber eine große Anzahl kann bedeuten, dass es interne oder externe Links zu nicht verfügbaren Seiten gibt. Diese Probleme können behoben werden, indem fehlerhafte interne Links repariert und Umleitungen für fehlerhafte externe Links hinzugefügt werden.

5 % der Anfragen schlagen mit HTTP-5xx fehl

3 % der Abfragen an den ELB schlagen mit HTTP 5xx fehl.

Ein hoher Prozentsatz der HTTP-Anfragen an eine Instance oder eine Umgebung schlagen mit Serienstatuscodes 500 fehl.

Ein Seriestatuscode 500 gibt an, dass beim Anwendungsserver ein interner Fehler aufgetreten ist. Diese Fehler zeigen an, dass es einen Fehler in Ihrem Anwendungscode gibt, der identifiziert und schnell behoben werden sollte.

95 % der CPU wird verwendet

Auf einer Instance meldet der Zustandsagent einen extrem hohen Prozentsatz an CPU-Auslastung und setzt den Instance-Zustand auf Warning (Warnung) oder Degraded (Schwach).

Skalieren Sie Ihre Umgebung, um Last von Instances zu nehmen.

Worker-Instances

20 Nachrichten in der Warteschlange (vor 25 Sekunden)

Anfragen werden schneller zur Warteschlange Ihrer Worker-Umgebung hinzugefügt, als sie verarbeitet werden können. Skalieren Sie Ihre Umgebung zur Erhöhung der Kapazität.

5 Nachrichten in Warteschlange für unzustellbare Nachrichten (vor 15 Sekunden)

Worker-Anfragen schlagen wiederholt fehl und werden zur [the section called "Warteschlangen für unzustellbare Nachrichten" \(p. 519\)](#) hinzugefügt. Prüfen Sie die Anfragen in der Warteschlange für unzustellbare Nachrichten, um zu sehen, warum sie nicht erfüllt werden.

Sonstige Ressourcen

4 aktive Instances liegen unter der Mindestgröße für Auto Scaling-Gruppen von 5

Die Anzahl der Instances in Ihrer Umgebung ist geringer als die minimale Konfiguration für die Auto Scaling-Gruppe.

Benachrichtigungen der Auto Scaling-Gruppe (Gruppenname) wurden gelöscht oder geändert

Die Benachrichtigungen, die für Ihre Auto Scaling-Gruppe konfiguriert sind, wurden außerhalb von Elastic Beanstalk geändert.

Verwalten von Alarmen

Sie können Alarne für Metriken erstellen, die Sie mit der Elastic Beanstalk-Konsole überwachen. Alarne unterstützen Sie bei der Überwachung von Änderungen in der AWS Elastic Beanstalk-Umgebung, sodass Sie Probleme ganz einfach identifizieren und noch vor dem Auftreten bekämpfen können. Sie können beispielsweise einen Alarm festlegen, der Sie benachrichtigt, wenn die CPU-Auslastung in einer Umgebung einem bestimmten Schwellenwert überschreitet, wodurch sichergestellt wird, dass Sie benachrichtigt werden, bevor ein potentielles Problem auftritt. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon CloudWatch \(p. 879\)](#).

Note

Elastic Beanstalk nutzt CloudWatch für die Überwachung und Alarne, d. h. CloudWatch-Kosten für verwendete Alarne werden auf Ihr AWS-Konto aufgeschlagen.

Weitere Informationen zur Überwachung spezifischer Metriken finden Sie unter [Grundlegende Zustandsberichte \(p. 821\)](#).

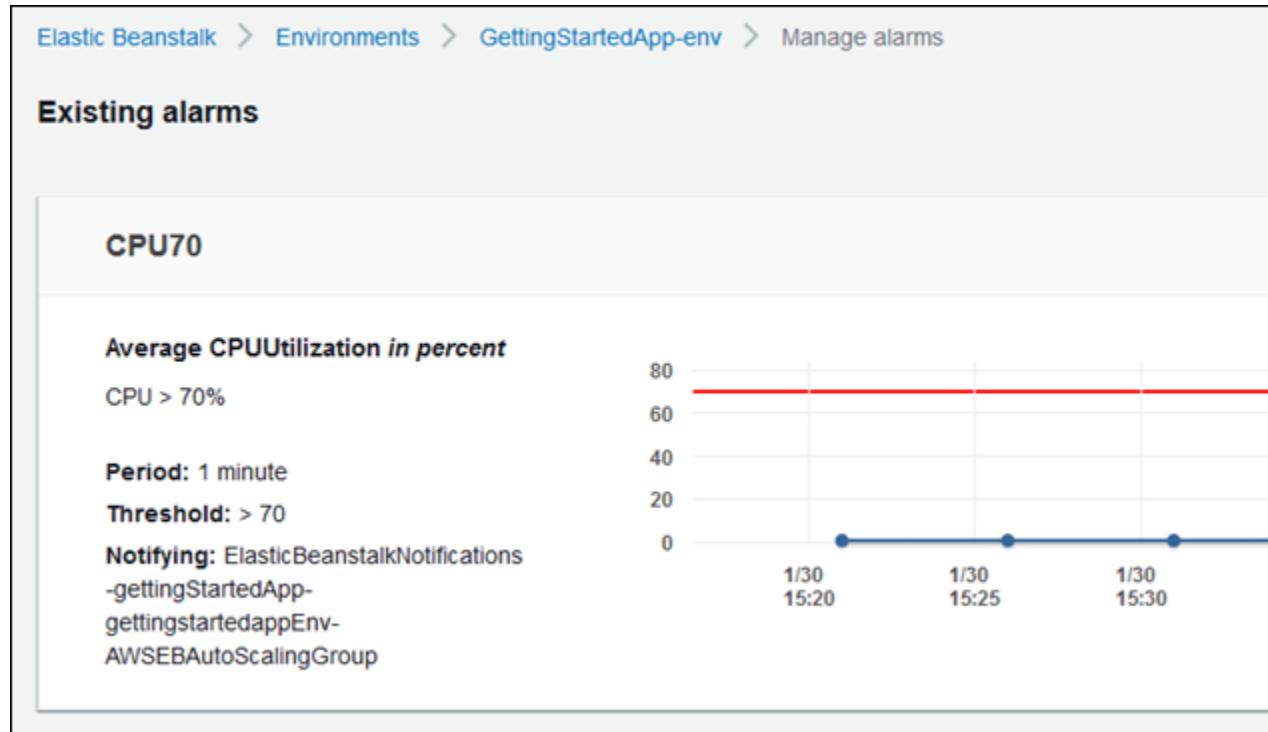
So prüfen Sie den Status Ihrer Alarne

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie im Navigationsbereich auf Alarne.



Die Seite zeigt eine Liste vorhandener Alarme an. Wenn sich Alarne im Alarmzustand befinden, werden sie mit (Warnung) gekennzeichnet.

4. Um Alarne zu filtern, wählen Sie das Drop-down-Menü und dann einen Filter aus.
5. Um einen Alarm zu bearbeiten oder zu löschen, wählen Sie (Bearbeiten) bzw. (Löschen) aus.

So erstellen Sie einen Alarm

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Monitoring (Überwachung) aus.
4. Suchen Sie die Metrik, für die Sie einen Alarm erstellen möchten, und wählen Sie dann (Alarm). Die Seite Add alarm (Alarm hinzufügen) wird angezeigt.

Elastic Beanstalk > Environments > GettingStartedApp-env > Add alarm

Add Alarm

Average CPUUtilization in percent

Name:

Name should be less than 238 characters in length and can only contain numbers and letters

Description:

Optional.

Period:

Threshold: Average CPUUtilization

Change state after:

Notify: Refresh

Topic name:

E-mail address:

Notify when state changes to:

OK
 Alarm
 Insufficient data

Other Alarms For This Metric

CPU70

Time	Average CPUUtilization
1/30 15:35	0.85
1/30 15:40	0.85
1/30 15:45	0.82

5. Geben Sie Informationen zum Alarm ein:

- Name: Ein Name für diesen Alarm.
- Description (optional): Eine kurze Beschreibung dieses Alarms.
- Period: Das Zeitintervall zwischen den Auslesungen.
- Threshold: Beschreibt das Verhalten und den Wert, den die Metrik überschreiten muss, um einen Alarm auszulösen.

- Change state after: Der Zeitraum nach dem Überschreiten eines Schwellenwerts, der eine Änderung des Status des Alarms auslöst.
 - Notify: Das Amazon SNS-Thema, das benachrichtigt wird, wenn ein Alarm einen anderen Zustand annimmt.
 - Benachrichtigung, wenn sich der Zustand in folgende Status ändert:
 - OK: Die Metrik liegt innerhalb des festgelegten Schwellenwerts.
 - Alarm: Die Metrik hat den festgelegten Schwellenwert überschritten.
 - Insufficient data: Der Alarm wurde soeben gestartet; die Metrik ist nicht verfügbar oder es sind nicht genügend Daten verfügbar, damit die Metrik den Alarmstatus bestimmen kann.
6. Wählen Sie Add aus. Der Umgebungsstatus wechselt zu Grau, während die Umgebung aktualisiert wird. Sie können den erstellten Alarm anzeigen, indem Sie im Navigationsbereich Alarms (Alarne) wählen.

Ereignis-Stream einer Elastic Beanstalk-Umgebung anzeigen

Verwenden Sie die AWS Management Console für den Zugriff auf Ereignisse und Benachrichtigungen im Zusammenhang mit Ihrer Anwendung.

So zeigen Sie Ereignisse an

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich die Option Events.

The screenshot shows the AWS Elastic Beanstalk Events page. At the top, there is a breadcrumb navigation: Elastic Beanstalk > Environments > GettingStartedApp-env > Events. Below this, a blue box contains a link to switch to the previous console. The main section is titled "Events" and has a "Severity" dropdown set to "TRACE". A navigation bar at the bottom shows pages 1 through 7. The event table has columns for Time, Type, and Details. The events listed are:

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 Warning. Environment update completed 85 seconds ago and took 1 minute 1 second.
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.

Die Seite „Event“ (Ereignisse) zeigt eine Liste aller Ereignisse an, die für die Umgebung aufgezeichnet wurden. Sie können durch die Liste blättern, indem Sie < (vorherige), > (nächste) oder Seitenzahlen wählen. Sie können nach der Art der angezeigten Ereignisse filtern, indem Sie die Drop-down-Liste Severity (Schweregrad) verwenden.

Die [EB CLI \(p. 1008\)](#) und die [AWS CLI](#) bieten Befehle zum Abrufen von Ereignissen. Wenn Sie Ihre Umgebung mit der EB CLI verwalten, verwenden Sie [eb events \(p. 1064\)](#), um eine Liste der Ereignisse zu drucken. Dieser Befehl verfügt zudem über eine --follow-Option, die weiterhin neue Ereignisse angezeigt, bis Sie Strg+C drücken, um die Ausgabe zu beenden.

Um Ereignisse mit der AWS CLI abzurufen, verwenden Sie den `describe-events`-Befehl und geben die Umgebung mit Namen oder ID an:

```
$ aws elasticbeanstalk describe-events --environment-id e-gbjzqccra3
{
    "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"
    },
...
```

Weitere Informationen zu Befehlszeilen-Tools finden Sie unter [Tools \(p. 1008\)](#).

Auflisten von Server-Instances/Verbinden mit Server-Instances

Sie können in der Elastic Beanstalk-Konsole eine Liste der Amazon EC2-Instances anzeigen, auf denen Ihre AWS Elastic Beanstalk-Anwendungsumgebung ausgeführt wird. Eine Verbindung zu den Instances stellen Sie über einen beliebigen SSH-Client her. Die Verbindung zu Instances, auf denen Windows ausgeführt wird, können Sie mit Remote Desktop herstellen.

Einige Hinweise zu spezifischen Entwicklungsumgebungen:

- Weitere Informationen zum Auflisten von Server-Instances und zum Herstellen einer Verbindung zu diesen mit dem AWS Toolkit for Eclipse finden Sie unter [Auflisten von Server-Instances/Verbinden mit Server-Instances \(p. 154\)](#).
- Weitere Informationen zum Auflisten von Server-Instances und zum Herstellen einer Verbindung zu diesen mit dem AWS Toolkit for Visual Studio finden Sie unter [Auflisten von Server-Instances/Verbinden mit Server-Instances \(p. 242\)](#).

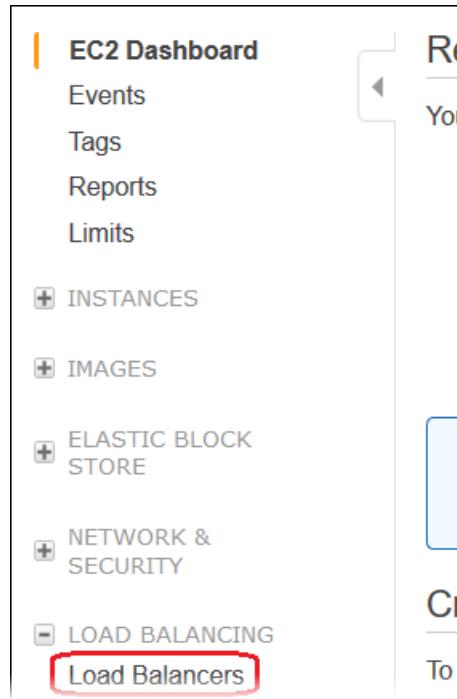
Important

Bevor Sie auf Ihre von Elastic Beanstalk bereitgestellten Amazon EC2-Instances zugreifen können, müssen Sie ein Amazon EC2-Schlüsselpaar erstellen und Ihre von Elastic Beanstalk bereitgestellten Amazon EC2-Instances für die Verwendung des Amazon EC2-Schlüsselpaares konfigurieren. Die Amazon EC2-Schlüsselpaare richten Sie mit der [AWS-Managementkonsole](#) ein. Anleitungen zum Erstellen eines Schlüsselpaares für Amazon EC2 finden Sie im Amazon EC2-Handbuch "Erste Schritte". Weitere Informationen zum Konfigurieren der Amazon EC2-Instances zur Verwendung eines Amazon EC2-Schlüsselpaares finden Sie unter [EC2 key pair \(p. 616\)](#).

Bei Elastic Beanstalk sind Remote-Verbindungen zu EC2-Instances in einem Windows-Container standardmäßig deaktiviert (außer bei Windows-Legacy-Containern). (Elastic Beanstalk konfiguriert EC2-Instances in Windows-Legacy-Containern zur Nutzung von Port 3389 für RDP-Verbindungen.) Für EC2-Instances, auf denen Windows ausgeführt wird, können Sie Remote-Verbindungen aktivieren. Dazu fügen Sie der Sicherheitsgruppe eine Regel hinzu, die eingehenden Datenverkehr für die Instances autorisiert. Es wird ausdrücklich empfohlen, diese Regel bei Beendigung der Remote-Verbindung zu entfernen. Sie können die Regel bei der nächsten Remote-Anmeldung wieder hinzufügen. Weitere Informationen finden Sie unter [Hinzufügen einer Regel für eingehenden RDP-Verkehr zu einer Windows-Instance](#) und [Verbinden mit der Windows Instanz](#) im Amazon Elastic Compute Cloud-Benutzerhandbuch für Microsoft Windows.///

So zeigen Sie Amazon EC2-Instances für eine Umgebung an und stellen eine Verbindung zu diesen her

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich der Konsole Load Balancers aus.

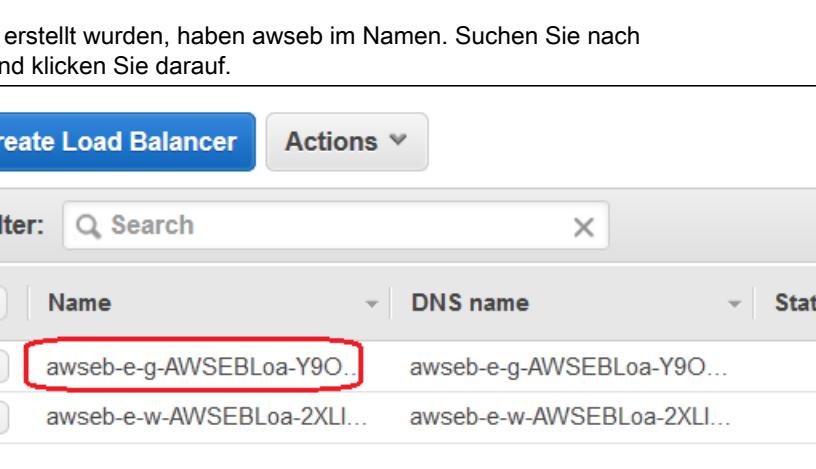


The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'LOAD BALANCING' section, the 'Load Balancers' link is highlighted with a red box. The main content area is titled 'Resources' and displays statistics for Amazon EC2 resources in the US East (N. Virginia) region. It shows 2 Running Instances, 0 Dedicated Hosts, 2 Volumes, 0 Key Pairs, and 0 Placement Groups. A promotional message for Amazon Lightsail is visible at the bottom.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an instance. You can do this by clicking the 'Create Instance' button in the top right corner of the EC2 dashboard or by selecting the 'Instances' link in the sidebar and then clicking the 'Create New Instance' button.

3. Load Balancer, die von Elastic Beanstalk erstellt wurden, haben awseb im Namen. Suchen Sie nach dem Load Balancer für Ihre Umgebung und klicken Sie darauf.



The screenshot shows the AWS EC2 Instances page. The 'Load Balancers' section is displayed, with a search bar labeled 'Filter: Search'. A table lists three load balancers. The first one, 'awseb-e-g-AWSEBLoa-Y9O...', has its name highlighted with a red box. The other two entries are 'awseb-e-w-AWSEBLoa-2XLI...' and 'awseb-e-w-AWSEBLoa-2XLI...'. The table includes columns for Name, DNS name, and Status.

4. Wählen Sie im unteren Abschnitt der Konsole die Registerkarte Instances aus.

The screenshot shows the AWS Elastic Beanstalk Instances tab for a load balancer named "awseb-e-g-AWSEBLoa-Y9O SHFJJQWI". The "Instances" tab is selected, highlighted with a red box. Below it, a message says "Connection Draining: Disabled (Edit)". A button labeled "Edit Instances" is visible. The main table lists one instance:

Instance ID	Name	Availability Zone
i-97d76d0e	Default-Environment	us-east-1c

Die Liste der Instances, die vom Load Balancer für die Elastic Beanstalk-Umgebung verwendet werden, wird angezeigt. Notieren Sie sich die ID der Instance, zu der eine Verbindung hergestellt werden soll.

5. Wählen Sie im Navigationsbereich der Amazon EC2-Konsole Instances aus und suchen Sie Ihre Instance-ID in der Liste.

The screenshot shows the AWS EC2 Instances page. The left sidebar has "INSTANCES" selected, with "Instances" highlighted with a red box. The main pane displays a list of instances, with the instance "i-97d76d0e" selected and its ID highlighted with a red box.

6. Klicken Sie mit der rechten Maustaste auf die Instance-ID der Amazon EC2-Instance, die vom Load Balancer dieser Umgebung verwendet wird, und wählen Sie im Kontextmenü Connect (Verbinden) aus.
7. Notieren Sie sich die öffentliche DNS-Adresse der Instance von der Registerkarte Description.
8. Stellen Sie eine Verbindung zu einer Instance her, auf der Linux ausgeführt wird, indem Sie einen SSH-Client Ihrer Wahl verwenden und `ssh -i .ec2/mykeypair.pem ec2-user@<public-DNS-of-the-instance>` eingeben.

Weitere Informationen zum Herstellen einer Verbindung mit einer Amazon EC2 Linux-Instance finden Sie unter [Erste Schritte mit Amazon EC2 Linux-Instances](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Wenn Ihre Elastic Beanstalk-Umgebung [.NET auf der Windows Server-Plattform verwendet \(p. 193\)](#), lesen Sie [Erste Schritte mit Amazon EC2 Windows-Instances](#) im Amazon EC2-Benutzerhandbuch für Windows-Instances.

Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen

Die Amazon EC2-Instances der Elastic Beanstalk-Umgebung generieren Protokolle, die Sie zur Behebung von Fehlern mit der Anwendung oder mit Konfigurationsdateien aufrufen können. Protokolle, die vom Webserver, Anwendungsserver, Elastic Beanstalk-Plattform-Scripts und AWS CloudFormation erstellt werden, werden lokal auf einzelnen Instances gespeichert. Sie können diese über die [Environment Management Console \(p. 426\)](#) oder die EB CLI ganz einfach abrufen. Sie können Ihre Umgebung auch so konfigurieren, dass Protokolle in Echtzeit an Amazon CloudWatch Logs gestreamt werden.

Als Protokollfragmente werden die letzten 100 Zeilen der am häufigsten verwendeten Protokolldateien bezeichnet, z. B. Elastic Beanstalk-Betriebsprotokolle sowie Protokolle vom Webserver oder Anwendungsserver. Wenn Sie Protokollfragmente in der Environment Management Console oder mit eb logs anfordern, verkettet eine Instance der Umgebung die letzten Protokolleinträge zu einer einzigen Textdatei und lädt diese in Amazon S3 hoch.

Bei Bundle-Protokollen handelt es sich um vollständige Protokolle für ein breiteres Spektrum an Protokolldateien, darunter Protokolle von yum und cron und verschiedene Protokolle von AWS CloudFormation. Bei der Anforderung von Bundle-Protokollen komprimiert eine Instance der Umgebung die vollständigen Protokolldateien in ein ZIP-Archiv und lädt dieses in Amazon S3 hoch.

Note

Elastic Beanstalk Windows Server-Plattformen unterstützen keine Bundle-Protokolle.

Damit rotierte Protokolle in Amazon S3 hochgeladen werden können, müssen die Instances der Umgebung über ein [Instance-Profil \(p. 22\)](#) mit Schreibberechtigung für den Elastic Beanstalk Amazon S3-Bucket verfügen. Diese Berechtigungen sind im Instance-Standardprofil enthalten, zu dessen Erstellung Sie von Elastic Beanstalk aufgefordert werden, wenn Sie zum ersten Mal eine Umgebung in der Elastic Beanstalk-Konsole starten.

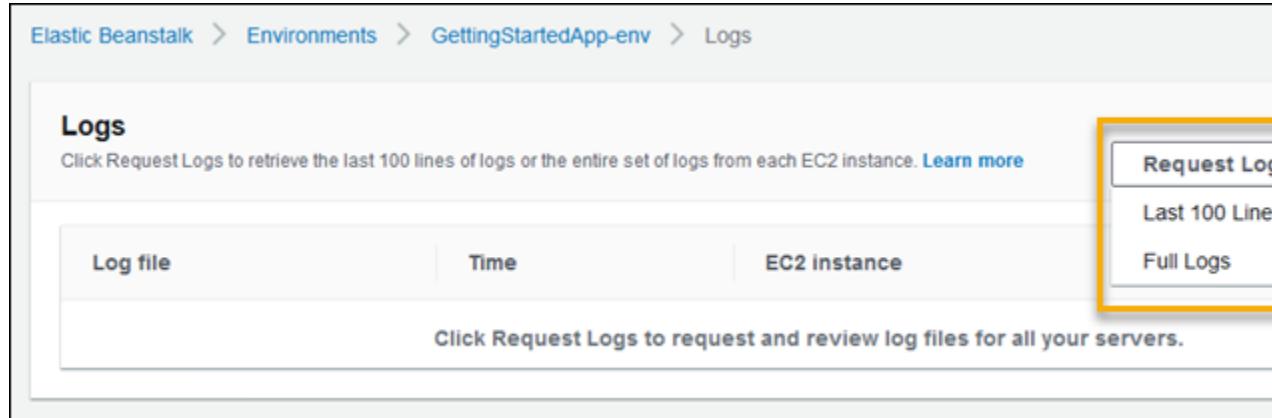
So rufen Sie Instance-Protokolle ab:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Logs aus.
4. Wählen Sie Request Logs (Protokolle anfordern) und wählen Sie dann den Typ der Protokolle, die abgerufen werden sollen. Um Protokollfragmente abzurufen, wählen Sie Last 100 Lines aus. Für den Abruf von Bundle-Protokollen wählen Sie Full Logs aus.



The screenshot shows the AWS Elastic Beanstalk interface for viewing logs. At the top, the navigation path is: Elastic Beanstalk > Environments > GettingStartedApp-env > Logs. The main title is "Logs". Below it, a sub-instruction says: "Click Request Logs to retrieve the last 100 lines of logs or the entire set of logs from each EC2 instance. [Learn more](#)". A table with three columns—Log file, Time, and EC2 instance—is shown. At the bottom, a button reads: "Click Request Logs to request and review log files for all your servers." On the far right, there's a vertical bar with three options: "Request Log" (which is highlighted with a yellow box), "Last 100 Line", and "Full Logs".

5. Wenn Elastic Beanstalk mit dem Abrufen Ihrer Protokolle fertig ist, wählen Sie Download (Herunterladen) aus.

Elastic Beanstalk speichert Tail- und Bundle-Protokolle in einem Amazon S3-Bucket und generiert eine vordefinierte Amazon S3-URL, über die Sie auf Ihre Protokolle zugreifen können. Elastic Beanstalk löscht die Dateien von Amazon S3 nach 15 Minuten.

Warning

Jeder im Besitz der vorsignierten Amazon S3-URL kann vor dem Löschen auf die Dateien zugreifen. Geben Sie nur vertrauenswürdigen Parteien Zugriff auf die URL.

Note

Ihre Benutzerrichtlinie muss die Berechtigung `s3:DeleteObject` haben. Elastic Beanstalk verwendet Ihre Benutzerberechtigungen, um die Protokolle von Amazon S3 zu löschen.

Damit Protokolle erhalten bleiben, können Sie Ihre Umgebung so konfigurieren, dass Protokolle nach dem Rotieren automatisch in Amazon S3 veröffentlicht werden. Befolgen Sie zum Rotieren von Protokollen in Amazon S3 die Anleitung unter [Konfigurieren der Anzeige von Instance-Protokollen \(p. 630\)](#). Instances in der Umgebung versuchen dann, Protokolle hochzuladen, die einmal pro Stunde rotiert wurden.

Wenn die Anwendung Protokolle an einem Speicherort generiert, der nicht zur Standardkonfiguration der Umgebungsplattform gehört, können Sie die Standardkonfiguration mithilfe von Konfigurationsdateien erweitern ([.ebextensions \(p. 722\)](#)). Die Protokolldateien der Anwendung können zu Protokollfragmenten, zu Bundle-Protokollen oder zur Protokollrotation hinzugefügt werden.

Konfigurieren Sie Ihre Umgebung für das [Streamen von Protokollen an Amazon CloudWatch Logs \(p. 874\)](#), wenn Sie ein Echtzeit-Streaming und eine langfristige Speicherung wünschen.

Abschnitte

- [Speicherort der Protokolle auf Amazon EC2-Instances \(p. 870\)](#)
- [Speicherort der Protokolle in Amazon S3 \(p. 871\)](#)
- [Protokollrotations-Einstellungen auf Linux \(p. 872\)](#)
- [Erweitern der Standardkonfiguration für Protokollaufgaben \(p. 872\)](#)
- [Protokolldateien an Amazon CloudWatch Logs streamen \(p. 874\)](#)

Speicherort der Protokolle auf Amazon EC2-Instances

Protokolle werden an Standardspeicherorten auf den Amazon EC2-Instances Ihrer Umgebung gespeichert. Elastic Beanstalk erzeugt die folgenden Protokolle.

Linux

- `/var/log/eb-activity.log`
- `/var/log/eb-commandprocessor.log`

Windows Server

- `C:\Program Files\Amazon\ElasticBeanstalk\logs\`
- `C:\cfn\logs\cfn-init.log`

Diese Protokolle enthalten Meldungen über Bereitstellungsaktivitäten, einschließlich solcher über Konfigurationsdateien ([.ebextensions \(p. 722\)](#)).

Jede Anwendung und jeder Webserver speichert Protokolle in einem eigenen Ordner:

- 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/tomcat8/`

Speicherort der Protokolle in Amazon S3

Wenn Sie Protokollfragmente oder Bundle-Protokolle aus Ihrer Umgebung anfordern oder wenn Instances rotierte Protokolle hochgeladen haben, werden diese in Ihrem Elastic Beanstalk-Bucket in Amazon S3 gespeichert. Elastic Beanstalk erstellt einen Bucket namens `elasticbeanstalk-region-account-id` für jede AWS-Region, in der Sie Umgebungen erstellen. In diesem Bucket werden Protokolle unter dem Pfad `resources/environments/logs/logtype/environment-id/instance-id` gespeichert.

Beispielsweise werden Protokolle der Instance `i-0a1fd158` der Elastic Beanstalk-Umgebung `e-mpcwnwheky` in der AWS-Region `us-west-2` des Kontos `123456789012` an den folgenden Speicherorten gespeichert:

- Tail Logs –

```
s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/  
tail/e-mpcwnwheky/i-0a1fd158
```

- Bundle Logs –

```
s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/  
bundle/e-mpcwnwheky/i-0a1fd158
```

- Rotated Logs –

```
s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/  
publish/e-mpcwnwheky/i-0a1fd158
```

Note

Die Umgebungs-ID finden Sie in der Environment Management Console.

Elastic Beanstalk löscht Tail- und Bundle-Protokolle von Amazon S3 automatisch 15 Minuten nach ihrer Erstellung. Rotierte Protokolle werden solange aufbewahrt, bis Sie diese löschen oder in S3 Glacier verschieben.

Protokollrotations-Einstellungen auf Linux

Auf Linux-Plattformen verwendet Elastic Beanstalk `logrotate`, um Protokolle periodisch zu rotieren. Nach der lokalen Rotation des Protokolls erfasst es die Protokollrotationsaufgabe und lädt es in Amazon S3 hoch (sofern dies konfiguriert ist). Lokal rotierte Protokolle werden standardmäßig nicht in Protokollfragmente oder Bundle-Protokolle aufgenommen.

Elastic Beanstalk-Konfigurationsdateien für `logrotate` finden Sie unter `/etc/logrotate.elasticbeanstalk.hourly/`. Diese Rotationseinstellungen sind plattformspezifisch und können sich in späteren Versionen der Plattform ändern. Führen Sie `man logrotate` aus, um weitere Informationen zu den verfügbaren Einstellungen sowie Beispielkonfigurationen zu erhalten.

Die Konfigurationsdateien werden von Cron-Aufträgen in `/etc/cron.hourly/` aufgerufen. Führen Sie zum Erhalten weiterer Informationen zu `cron man cron` aus.

Erweitern der Standardkonfiguration für Protokollaufgaben

In Elastic Beanstalk werden Dateien in den Unterordnern von `/opt/elasticbeanstalk/tasks` (Linux) oder `C:\Program Files\Amazon\ElasticBeanstalk\config` (Windows Server) auf der Amazon EC2-Instance verwendet, um Aufgaben für Protokollfragmente, Bundle-Protokolle und Protokollrotation zu konfigurieren.

Unter Linux:

- Tail Logs –

`/opt/elasticbeanstalk/tasks/taillogs.d/`

- Bundle Logs –

`/opt/elasticbeanstalk/tasks/bundlelogs.d/`

- Rotated Logs –

`/opt/elasticbeanstalk/tasks/publishlogs.d/`

Auf Windows Server:

- Tail Logs –

`c:\Program Files\Amazon\ElasticBeanstalk\config\taillogs.d\`

- Rotated Logs –

`c:\Program Files\Amazon\ElasticBeanstalk\config\publogs.d\`

Beispielsweise werden mit der Datei `eb-activity.conf` auf Linux zwei Protokolldateien zur Protokollfragmentaufgabe hinzugefügt:

`/opt/elasticbeanstalk/tasks/taillogs.d/eb-activity.conf`

`/var/log/eb-commandprocessor.log`

```
/var/log/eb-activity.log
```

Sie können Umgebungskonfigurationsdateien ([.ebextensions \(p. 722\)](#)) verwenden, um Ihre eigenen .conf-Dateien diesen Ordnern hinzuzufügen. Eine .conf-Datei listet für Ihre Anwendung spezifische Protokolldateien auf, die Elastic Beanstalk zu den Protokollaufgaben hinzufügt.

Verwenden Sie den Abschnitt [files \(p. 729\)](#), um Konfigurationsdateien zu den Aufgaben hinzuzufügen, die geändert werden sollen. Der folgende Konfigurationstext fügt beispielsweise eine Protokollkonfigurationsdatei zu den einzelnen Instances Ihrer Umgebung hinzu. Die Protokollkonfigurationsdatei cloud-init.conf fügt /var/log/cloud-init.log zu Protokollfragmenten hinzu.

```
files:
  "/opt/elasticbeanstalk/tasks/taillogs.d/cloud-init.conf" :
    mode: "000755"
    owner: root
    group: root
    content: |
      /var/log/cloud-init.log
```

Fügen Sie diesen Text zu einer Datei mit der Dateinamenerweiterung .config zu Ihrem Quell-Bundle in einen Ordner namens .ebextensions hinzu.

```
~/workspace/my-app
|-- .ebextensions
|   '-- tail-logs.config
|-- index.php
`-- styles.css
```

Auf Linux-Plattformen können Sie bei der Konfiguration von Protokollaufgaben auch Platzhalterzeichen einsetzen. Mithilfe dieser Konfigurationsdatei werden alle Dateien mit der Erweiterung .log, die sich im Ordner log des Anwendungsstamms befinden, zu Bundle-Protokollen hinzugefügt.

```
files:
  "/opt/elasticbeanstalk/tasks/bundlereads.d/applogs.conf" :
    mode: "000755"
    owner: root
    group: root
    content: |
      /var/app/current/log/*.log
```

Konfiguration von Protokollaufgaben unterstützen keine Platzhalterzeichen auf Windows-Plattformen.

Note

Wenn Sie sich mit den Verfahren zur Anpassung von Protokollen vertraut machen möchten, können Sie mithilfe der [EB-CLI \(p. 1008\)](#) eine Beispielanwendung bereitstellen. Dazu erstellt die EB-CLI ein lokales Anwendungsverzeichnis, in dem .ebextensions-Unterverzeichnis mit einer Beispielkonfiguration enthalten ist. Sie können die Protokolldateien der Beispielanwendung auch dazu verwenden, die in diesem Thema beschriebene Protokollabruffunktion zu untersuchen. Weitere Informationen zum Erstellen einer Beispielanwendung mit der EB-CLI finden Sie unter [Grundlagen der EB-CLI \(p. 1020\)](#).

Weitere Informationen zur Verwendung von Konfigurationsdateien finden Sie unter [Erweiterte Umgebungsanpassung mit Konfigurationsdateien \(.ebextensions\) \(p. 722\)](#).

So wie Sie Protokollfragmente und Bundle-Protokolle erweitern können, so können Sie auch Protokollrotation mit einer Konfigurationsdatei erweitern. Jedes Mal, wenn Elastic Beanstalk seine eigenen

Protokolle rotiert und auf Amazon S3 hochlädt, rotiert es auch Ihre zusätzlichen Protokolle und lädt sie hoch. Die Protokollrotation-Erweiterung verhält sich abhängig vom Betriebssystem der Plattform anders. In den folgenden Abschnitten werden diese beiden Fälle beschrieben.

Erweitern der Protokollrotation auf Linux

Wie in [Protokollrotations-Einstellungen auf Linux \(p. 872\)](#) erläutert, verwendet Elastic Beanstalk logrotate zum Rotieren von Protokollen auf Linux-Plattformen. Wenn Sie die Protokolldateien Ihrer Anwendung für Protokollrotation konfigurieren, muss die Anwendung keine Kopien von Protokolldateien erstellen. Elastic Beanstalk konfiguriert logrotate so, dass bei jeder Rotation eine Kopie der Protokolldateien Ihrer Anwendung erstellt wird. Aus diesem Grund muss die Anwendung die Protokolldateien entsperrt halten, wenn sie nicht aktiv in sie schreibt.

Ausdehnen der Protokollrotation auf Windows Server

Auf Windows Server muss die Anwendung die Protokolldateien regelmäßig rotieren, wenn Sie Ihre Anwendung für Protokollrotation konfigurieren. Elastic Beanstalk sucht nach Dateien mit Namen, die mit dem von Ihnen konfigurierten Muster beginnen, und markiert sie zum Hochladen in Amazon S3. Darüber hinaus werden Punkte im Dateinamen ignoriert und Elastic Beanstalk betrachtet den Namen bis zum Punkt als Basis-Protokolldateinamen.

Elastic Beanstalk lädt alle Versionen einer Basis-Protokolldatei hoch, mit Ausnahme der neuesten, da es diese als aktive Protokolldatei der Anwendung betrachtet, die möglicherweise gesperrt sein kann. Ihre Anwendung kann daher die aktive Protokolldatei zwischen Rotationen gesperrt halten.

Beispiel: Ihre Anwendung schreibt in eine Protokolldatei mit dem Namen `my_log.log` und Sie geben diesen Namen in der `.conf`-Datei an. Die Anwendung rotiert die Datei in regelmäßigen Abständen. Während des Elastic Beanstalk-Rotationszyklus findet die Anwendung die folgenden Dateien im Ordner der Protokolldatei: `my_log.log`, `my_log.0800.log` und `my_log.0830.log`. Elastic Beanstalk betrachtet sie alle als Versionen des gleichen Basisnamens `my_log`. Die Datei `my_log.log` hat die neueste Änderungszeit, daher lädt Elastic Beanstalk nur die beiden anderen Dateien, `my_log.0800.log` und `my_log.0830.log`, hoch.

Protokolldateien an Amazon CloudWatch Logs streamen

Sie können Ihre Umgebung in der Elastic Beanstalk-Konsole oder mit [Konfigurationsoptionen \(p. 646\)](#) so konfigurieren, dass Protokolle an Amazon CloudWatch Logs gestreamt werden. Mit CloudWatch Logs streamt jede Instance der Umgebung die Protokolle an Protokollgruppen, die je nach Konfiguration für Wochen oder Jahre aufbewahrt werden (auch bei beendeter Umgebung).

Die gestreamten Protokollsätze sind je nach Umgebung unterschiedlich, enthalten aber immer das Protokoll `eb-activity.log` sowie die Zugriffsprotokolle des nginx- oder Apache-Proxy-Servers, der vor der Anwendung ausgeführt wird.

Sie können das Protokoll-Streamen in der Elastic Beanstalk-Konsole entweder beim [Erstellen der Umgebung \(p. 443\)](#) oder für eine vorhandene Umgebung ([p. 630](#)) konfigurieren. Im folgenden Beispiel werden Protokolle auch dann für sieben Tage gespeichert, wenn die Umgebung beendet wird.

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 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

Mit der folgenden [Konfigurationsdatei](#) (p. 722) wird das Protokoll-Streaming aktiviert und die Protokolle bleiben 180 Tage erhalten, auch bei beendeter Umgebung.

Example .ebextensions/log-streaming.config

```
option_settings:  
  aws:elasticbeanstalk:cloudwatch:logs:  
    StreamLogs: true  
    DeleteOnTerminate: false  
    RetentionInDays: 180
```

Elastic Beanstalk mit anderen AWS-Services verwenden

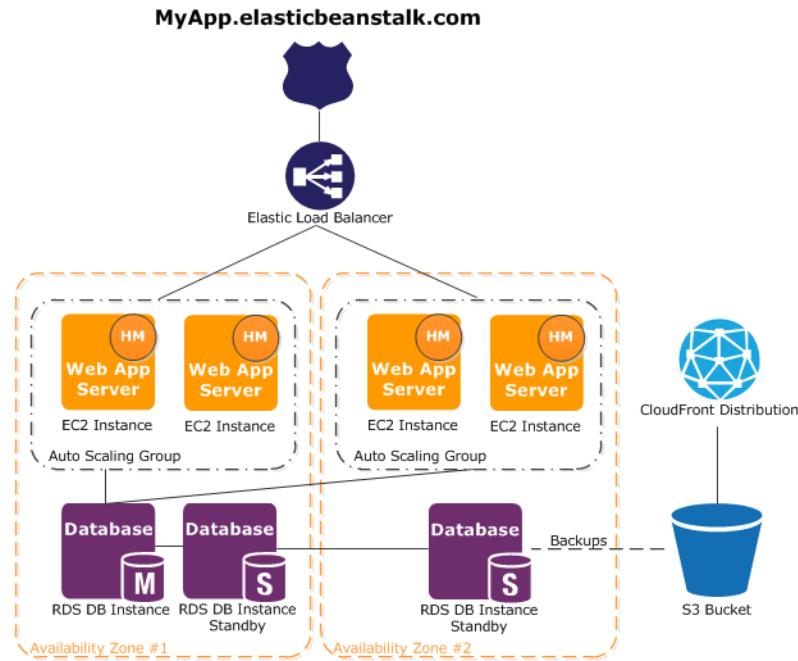
Zum Implementieren der Anwendungsumgebungen verwaltet Elastic Beanstalk Ressourcen anderer AWS-Services oder nutzt deren Funktionalität. Darüber hinaus lässt sich Elastic Beanstalk mit AWS-Services integrieren, die nicht direkt als Teil der Umgebungen genutzt werden. Die Themen in diesem Abschnitt beschreiben viele Verwendungsmöglichkeiten dieser zusätzlichen Services mit Elastic Beanstalk-Anwendungen.

Themen

- [Übersicht über die Architektur \(p. 876\)](#)
- [Verwenden von Elastic Beanstalk mit Amazon CloudFront \(p. 877\)](#)
- [Protokollieren von Elastic Beanstalk-API-Aufrufen mit AWS CloudTrail \(p. 878\)](#)
- [Verwenden von Elastic Beanstalk mit Amazon CloudWatch \(p. 879\)](#)
- [Verwenden von Elastic Beanstalk mit Amazon CloudWatch Logs \(p. 880\)](#)
- [Verwenden von Elastic Beanstalk mit Amazon EventBridge \(p. 891\)](#)
- [Suchen und Verfolgen von Elastic Beanstalk-Ressourcen mit AWS Config \(p. 895\)](#)
- [Elastic Beanstalk mit Amazon DynamoDB verwenden \(p. 900\)](#)
- [Verwenden von Elastic Beanstalk mit Amazon ElastiCache \(p. 901\)](#)
- [Elastic Beanstalk mit Amazon Elastic File System verwenden \(p. 901\)](#)
- [Elastic Beanstalk mit AWS Identity and Access Management verwenden \(p. 903\)](#)
- [Verwenden von Elastic Beanstalk mit Amazon RDS \(p. 973\)](#)
- [Verwenden von Elastic Beanstalk mit Amazon S3 \(p. 985\)](#)
- [Verwenden von Elastic Beanstalk mit Amazon VPC \(p. 988\)](#)

Übersicht über die Architektur

Das folgende Diagramm veranschaulicht eine Beispielarchitektur von Elastic Beanstalk über mehrere Availability Zones hinweg, die mit anderen AWS-Produkten wie Amazon CloudFront, Amazon Simple Storage Service (Amazon S3) und Amazon Relational Database Service (Amazon RDS) zusammenarbeitet.



Um eine Fehlertoleranz einzuplanen, sollten Sie über N+1 Amazon EC2-Instances verfügen und Ihre Instances über mehrere Availability Zones verteilen. Für den unwahrscheinlichen Fall, dass eine Availability Zone ausfällt, haben Sie immer noch Ihre anderen Amazon EC2-Instances, die in einer weiteren Availability Zone ausgeführt werden. Sie können Amazon EC2 Auto Scaling so anpassen, dass eine Mindestanzahl von Instances sowie mehrere Availability Zones zulässig sind. Anweisungen dazu finden Sie unter [Auto Scaling-Gruppe für Ihre Elastic Beanstalk-Umgebung \(p. 540\)](#). Weitere Informationen zum Erstellen von fehlertoleranten Anwendungen finden Sie unter [Erstellung fehlertoleranter Anwendungen in AWS](#).

In den folgenden Abschnitten wird die Integration mit Amazon CloudFront, Amazon CloudWatch, Amazon DynamoDB Amazon ElastiCache, Amazon RDS, Amazon Route 53, Amazon Simple Storage Service, Amazon VPC und IAM ausführlicher besprochen.

Verwenden von Elastic Beanstalk mit Amazon CloudFront

Amazon CloudFront ist ein Web-Service, der die Verteilung Ihrer statischen und dynamischen Webinhalte, z. B. HTML-, CSS-, PHP-, Bild- und Mediendateien, an Endbenutzer beschleunigt. Mit CloudFront können Sie Ihre Inhalte über ein globales Netzwerk aus Edge-Standorten bereitstellen. Wenn ein Endbenutzer Inhalte anfordert, die Sie mit CloudFront bereitstellen, wird der Benutzer zum Edge-Standort mit der kürzesten Latenz geleitet, damit die Inhalte mit der bestmöglichen Leistung übertragen werden. Wenn sich die Inhalte bereits an diesem Edge-Standort befinden, stellt sie CloudFront sofort bereit. Wenn sich die Inhalte derzeit nicht an diesem Edge-Standort befinden, ruft CloudFront sie aus einem Amazon S3-Bucket oder von einem HTTP-Server (z. B. einem Webserver) ab, den Sie als Quelle der definitiven Version Ihrer Inhalte angegeben haben.

Nachdem Sie Ihre Elastic Beanstalk-Anwendung erstellt und bereitgestellt haben, können Sie sich bei CloudFront registrieren und CloudFront zur Verteilung Ihres Inhalts verwenden. Erfahren Sie mehr über CloudFront im [Amazon CloudFront-Entwicklerhandbuch](#).

Protokollieren von Elastic Beanstalk-API-Aufrufen mit AWS CloudTrail

Elastic Beanstalk ist in AWS CloudTrail integriert, einen Service, der die Aktionen eines Benutzers, einer Rolle oder eines AWS-Services in Elastic Beanstalk protokolliert. CloudTrail erfasst alle API-Aufrufe für Elastic Beanstalk als Ereignisse, einschließlich der Aufrufe von der Elastic Beanstalk-Konsole, von der EB CLI und von Ihrem Code an die Elastic Beanstalk-APIs. Wenn Sie einen Trail erstellen, aktivieren Sie die kontinuierliche Bereitstellung von CloudTrail-Ereignissen an einen Amazon S3-Bucket, einschließlich Ereignissen für Elastic Beanstalk. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail-Konsole trotzdem in Event history (Ereignisverlauf) anzeigen. Mit den von CloudTrail erfassten Informationen können Sie die an Elastic Beanstalk gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen.

Weitere Informationen zu CloudTrail finden Sie im [AWS CloudTrail-Benutzerhandbuch](#).

Elastic Beanstalk-Informationen in CloudTrail

CloudTrail wird beim Erstellen Ihres AWS-Kontos für Sie aktiviert. Die in Elastic Beanstalk auftretenden Aktivitäten werden als CloudTrail-Ereignis zusammen mit anderen AWS-Serviceereignissen in Event history (Ereignisverlauf) aufgezeichnet. Sie können die neusten Ereignisse in Ihrem AWS-Konto anzeigen, suchen und es dorthin herunterladen. Weitere Informationen finden Sie unter [Anzeigen von Ereignissen mit dem CloudTrail-Ereignisverlauf](#).

Erstellen Sie für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, darunter Ereignisse für Elastic Beanstalk, einen Trail. Ein Pfad ermöglicht es CloudTrail, Protokolldateien in einem Amazon S3-Bucket bereitzustellen. Wenn Sie einen Pfad in der Konsole anlegen, gilt dieser standardmäßig für alle Regionen. Der Pfad protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien in dem Amazon S3-Bucket bereit, den Sie angeben. Darüber hinaus können Sie andere AWS-Services konfigurieren, um die in den CloudTrail-Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie unter:

- [Übersicht zum Erstellen eines Pfads](#)
- [Siehe Von CloudTrail unterstützte Services und Integrationen](#).
- [Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail-Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail-Protokolldateien von mehreren Konten](#).

Alle Elastic Beanstalk-Aktionen werden von CloudTrail protokolliert und sind in der [AWS Elastic Beanstalk-API-Referenz](#) dokumentiert. Zum Beispiel generieren Aufrufe der Aktionen `DescribeApplications`, `UpdateEnvironment` und `ListTagsForResource` Einträge in den CloudTrail-Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Root- oder IAM-Benutzeranmeldeinformationen ausgeführt wurde.
- Ob die Anfrage mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen föderierten Benutzer ausgeführt wurde.
- Ob die Anforderung von einem anderen AWS-Service getätigkt wurde.

Weitere Informationen finden Sie unter dem [CloudTrail userIdentity-Element](#).

Grundlegendes zu den Elastic Beanstalk-Einträgen in der Protokolldatei

Ein Trail ist eine Konfiguration, durch die Ereignisse als Protokolldateien an den von Ihnen angegebenen Amazon S3 -Bucket übermittelt werden. CloudTrail-Protokolldateien können einen oder mehrere Einträge enthalten. Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie über die Anfrageparameter. CloudTrail-Protokolleinträge sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher nicht in einer bestimmten Reihenfolge.

Das folgende Beispiel zeigt einen CloudTrail-Protokolleintrag, der die Aktion `UpdateEnvironment` mit Aufruf durch den IAM-Benutzer `intern` für die `sample-env`-Umgebung in der `sample-app`-Anwendung demonstriert.

```
{  
    "Records": [  
        {  
            "eventVersion": "1.05",  
            "userIdentity": {  
                "type": "IAMUser",  
                "principalId": "AIXDAYQEXAMPLEUMLYNGL",  
                "arn": "arn:aws:iam::123456789012:user/intern",  
                "accountId": "123456789012",  
                "accessKeyId": "ASXIAQEXAMPLEQULKNXV",  
                "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"  
        }]  
}
```

Verwenden von Elastic Beanstalk mit Amazon CloudWatch

Mit Amazon CloudWatch überwachen, verwalten und veröffentlichen Sie verschiedene Metriken und konfigurieren Alarmaktionen auf Basis dieser Metrikdaten. Die Amazon CloudWatch-Überwachung

ermöglicht es Ihnen, System- und Anwendungsmetriken zu sammeln, zu analysieren und aufzurufen, um betriebliche und geschäftliche Entscheidungen schnell und sicher treffen zu können.

Amazon CloudWatch sammelt automatisch Metriken über Ihre Amazon Web Services-Ressourcen (AWS) – zum Beispiel die Leistung Ihrer Amazon EC2-Instances. Sie können Ihre eigenen Metriken auch direkt in Amazon CloudWatch veröffentlichen. Amazon CloudWatch-Alarme unterstützen Sie bei der Implementierung von Entscheidungen. Sie ermöglichen Ihnen das Senden von Nachrichten oder das automatische Vornehmen von Änderungen an den Ressourcen, die Sie überwachen, gemäß der von Ihnen festgelegten Regeln. Sie können beispielsweise Alarne erstellen, die in Ihrem Namen Aktionen des Amazon EC2 Auto Scaling-Services und des Amazon Simple Notification Service (Amazon SNS) starten.

Elastic Beanstalk nutzt Amazon CloudWatch automatisch zur Kontrolle Ihres Anwendungs- und Umgebungsstatus. Sie können zur Amazon CloudWatch-Konsole navigieren, um Ihr Dashboard zu sehen und einen Überblick über alle Ihre Ressourcen und Ihre Alarne zu erhalten. Sie können auch mehrere Metriken anzeigen oder benutzerdefinierte Metriken hinzufügen.

Weitere Informationen zu Amazon CloudWatch finden Sie im [Amazon CloudWatch Entwicklerhandbuch](#). Ein Beispiel für die Verwendung von Amazon CloudWatch mit Elastic Beanstalk finden Sie unter [the section called “Beispiel: Verwenden von benutzerdefinierten Amazon CloudWatch-Metriken” \(p. 735\)](#).

Verwenden von Elastic Beanstalk mit Amazon CloudWatch Logs

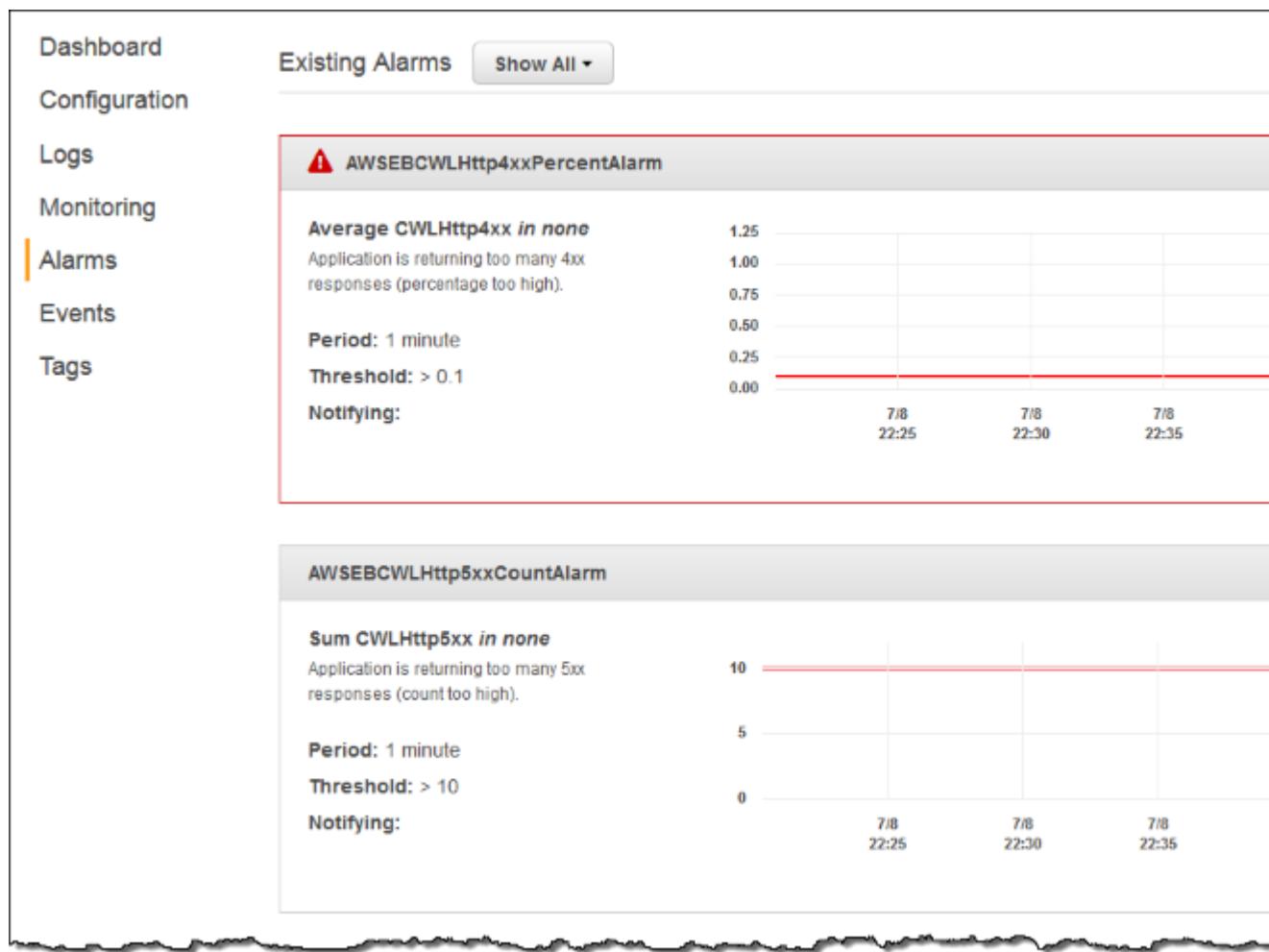
Mit CloudWatch Logs können Sie Ihre Elastic Beanstalk-Anwendungs-, System- und benutzerdefinierten Protokolldateien von Amazon EC2-Instances Ihrer Umgebungen überwachen und archivieren. Sie können auch Alarne konfigurieren, durch die Sie einfacher auf bestimmte Protokollstreamereignisse reagieren können, die Ihre Metrikfilter extrahieren. Der auf jeder Amazon EC2-Instance in Ihrer Umgebung installierte CloudWatch Logs-Agent veröffentlicht Metrik-Datenpunkte im CloudWatch-Service für jede Protokollgruppe, die Sie konfigurieren. Jede Protokollgruppe wendet ihre eigenen Filtermuster an, um zu bestimmen, welche Protokollstreamereignisse als Datenpunkte an CloudWatch gesendet werden. Protokollstreams, die zur selben Protokollgruppe gehören, besitzen dieselben Einstellungen für die Aufbewahrung, Überwachung und Zugriffskontrolle. Sie können Elastic Beanstalk so konfigurieren, dass Protokolle automatisch an den CloudWatch-Service gestreamt werden, wie in [Streamen von Instance-Protokollen an CloudWatch Logs \(p. 886\)](#) beschrieben. Weitere Informationen zu CloudWatch Logs, einschließlich Terminologie und Konzepten, finden Sie im [Amazon CloudWatch Logs-Benutzerhandbuch](#).

Zusätzlich zu den Instance-Protokollen können Sie bei Aktivierung der [erweiterten Integrität \(p. 824\)](#) für Ihre Umgebung die Umgebung für das Streamen von Integritätsinformationen an CloudWatch Logs konfigurieren. Siehe [Streaming von Integritätsinformationen der Elastic Beanstalk-Umgebung an Amazon CloudWatch Logs \(p. 888\)](#).

Die folgende Abbildung zeigt die Seite Monitoring (Überwachung) und Graphen für eine Umgebung, die mit der CloudWatch Logs-Integration konfiguriert ist. Die Beispielmetriken in dieser Umgebung haben die Bezeichnungen CWLHttp4xx und CWLHttp5xx. Einer der Graphen zeigt, dass die CWLHttp4xx-Metrik einen Alarm gemäß den Bedingungen ausgelöst hat, die in den Konfigurationsdateien angegeben sind.



Die folgende Abbildung zeigt Diagramme auf der Seite Alarms (Alarne) und Graphen für die Beispielalarme mit der Bezeichnung AWSEBCWLHttp4xxPercentAlarm und AWSEBCWLHttp5xxCountAlarm, die den Metriken CWLHttp4xx bzw. CWLHttp5xx entsprechen.



Themen

- [Voraussetzungen für das Instance-Protokoll-Streaming an CloudWatch Logs \(p. 882\)](#)
- [Wie Elastic Beanstalk CloudWatch Logs einrichtet \(p. 883\)](#)
- [Streamen von Instance-Protokollen an CloudWatch Logs \(p. 886\)](#)
- [Fehlerbehebung bei der CloudWatch Logs-Integration \(p. 888\)](#)
- [Streaming von Integritätsinformationen der Elastic Beanstalk-Umgebung an Amazon CloudWatch Logs \(p. 888\)](#)

Voraussetzungen für das Instance-Protokoll-Streaming an CloudWatch Logs

Um das Streaming von Protokollen von Amazon EC2-Instances Ihrer Umgebung nach CloudWatch Logs zu aktivieren, müssen Sie die folgenden Bedingungen erfüllen.

- Platform (Plattform) – Da diese Funktion nur in Plattformversionen zur Verfügung steht, die mit oder nach [diesem Release](#) veröffentlicht wurden, müssen Sie Ihre Umgebung aktualisieren, falls Sie eine frühere Plattformversion verwenden.

- Wenn Sie noch nicht über die von Elastic Beanstalk verwaltete Richtlinie AWSElasticBeanstalkWebTier oder AWSElasticBeanstalkWorkerTier in Ihrem [Elastic Beanstalk-Instance-Profil \(p. 22\)](#) verfügen, müssen Sie Folgendes zu Ihrem Profil hinzufügen, um diese Funktion zu aktivieren.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "logs:PutLogEvents",  
                "logs>CreateLogStream"  
            ],  
            "Resource": [  
                "*"  
            ]  
        }  
    ]  
}
```

Wie Elastic Beanstalk CloudWatch Logs einrichtet

Elastic Beanstalk installiert einen CloudWatch-Protokollagenten mit den Standard-Konfigurationseinstellungen auf jeder Instance, die er erstellt. Weitere Informationen finden Sie in der [Referenz für den CloudWatch Logs-Agenten](#).

Wenn Sie das Streamen von Instance-Protokollen an CloudWatch Logs aktivieren, sendet Elastic Beanstalk Protokolldateien von den Instances Ihrer Umgebung an CloudWatch Logs. Unterschiedliche Plattformen streamen unterschiedliche Protokolle. In der folgenden Tabelle finden Sie die Protokolle nach Plattform sortiert.

Plattform	Protokolle
Docker	<ul style="list-style-type: none">/var/log/eb-engine.log/var/log/eb-hooks.log/var/log/docker/var/log/docker-events.log/var/log/eb-docker/containers/eb-current-app/stdouterr.log/var/log/nginx/access.log/var/log/nginx/error.log
Go	<ul style="list-style-type: none">/var/log/eb-engine.log
Corretto	<ul style="list-style-type: none">/var/log/eb-hooks.log/var/log/web.stdout.log
.NET Core auf Linux	<ul style="list-style-type: none">/var/log/nginx/access.log/var/log/nginx/error.log
Node.js	<ul style="list-style-type: none">/var/log/eb-engine.log
Python	<ul style="list-style-type: none">/var/log/eb-hooks.log/var/log/web.stdout.log/var/log/httpd/access_log/var/log/httpd/error_log/var/log/nginx/access.log

Plattform	Protokolle
	<ul style="list-style-type: none"> • /var/log/nginx/error.log
Tomcat	<ul style="list-style-type: none"> • /var/log/eb-engine.log
PHP	<ul style="list-style-type: none"> • /var/log/eb-hooks.log • /var/log/httpd/access_log • /var/log/httpd/error_log • /var/log/nginx/access.log • /var/log/nginx/error.log
.NET mit Windows Server	<ul style="list-style-type: none"> • C:\inetpub\logs\LogFiles\W3SVC1\u_ex*.log • C:\Program Files\Amazon\ElasticBeanstalk\logs\AWSDeployment.log • C:\Program Files\Amazon\ElasticBeanstalk\logs\Hooks.log
Ruby	<ul style="list-style-type: none"> • /var/log/eb-engine.log • /var/log/eb-hooks.log • /var/log/puma/puma.log • /var/log/web.stdout.log • /var/log/nginx/access.log • /var/log/nginx/error.log

Protokolldateien auf Amazon Linux AMI-Plattformen

In der folgenden Tabelle werden die Protokolldateien nach Plattform aufgelistet, die aus Instances in Plattformverzweigungen auf der Basis von Amazon Linux AMI (vor Amazon Linux 2) gestreamt werden.

Plattform	Protokolle
Docker	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/nginx/error.log • /var/log/docker-events.log • /var/log/docker • /var/log/nginx/access.log • /var/log/eb-docker/containers/eb-current-app/stdouterr.log
Multicontainer-Docker (generisch)	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/ecs/ecs-init.log • /var/log/eb-ecs-mgr.log • /var/log/ecs/ecs-agent.log • /var/log/docker-events.log
Glassfish (Docker vorkonfiguriert)	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/nginx/error.log • /var/log/docker-events.log • /var/log/docker • /var/log/nginx/access.log
Go (Docker vorkonfiguriert)	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/nginx/error.log

Plattform	Protokolle
	<ul style="list-style-type: none"> • /var/log/docker-events.log • /var/log/docker • /var/log/nginx/access.log
Python (Docker vorkonfiguriert)	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/nginx/error.log • /var/log/docker-events.log • /var/log/docker • /var/log/nginx/access.log
Go	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/nginx/error.log • /var/log/nginx/access.log
Java SE	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/nginx/access.log • /var/log/nginx/error.log • /var/log/web-1.error.log • /var/log/web-1.log
Tomcat	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/httpd/error_log • /var/log/httpd/access_log • /var/log/nginx/error_log • /var/log/nginx/access_log
Node.js	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/nodejs/nodejs.log • /var/log/nginx/error.log • /var/log/nginx/access.log • /var/log/httpd/error.log • /var/log/httpd/access.log
PHP	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/httpd/error_log • /var/log/httpd/access_log
Python	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/httpd/error_log • /var/log/httpd/access_log • /opt/python/log/supervisord.log
Ruby (Puma)	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/log/nginx/error.log • /var/log/puma/puma.log • /var/log/nginx/access.log

Plattform	Protokolle
Ruby (Passenger)	<ul style="list-style-type: none"> • /var/log/eb-activity.log • /var/app/support/logs/passenger.log • /var/app/support/logs/access.log • /var/app/support/logs/error.log

Elastic Beanstalk konfiguriert Protokollgruppen in CloudWatch Logs für die verschiedenen Protokolldateien, die es streamt. Zum Abrufen bestimmter Protokolldateien aus CloudWatch Logs müssen Sie den Namen der entsprechenden Protokollgruppe kennen. Das Namensschema der Protokollgruppe hängt vom Betriebssystem der Plattform ab.

Für Linux-Plattformen stellen Sie der Protokolldatei auf der Instance das Präfix `/aws/elasticbeanstalk/environment_name` voraus, um den Protokollgruppennamen zu erhalten. Um beispielsweise die Datei `/var/log/nginx/error.log` abzurufen, geben Sie die Protokollgruppe `/aws/elasticbeanstalk/environment_name/var/log/nginx/error.log` an.

Für Windows-Plattformen finden Sie in der folgenden Tabelle die Protokollgruppen für alle Protokolldateien.

Protokolldatei auf der Instance	Protokollgruppe
C:\Program Files\Amazon\ElasticBeanstalk\logs\AWSDeployment.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 (das gesamte Verzeichnis)	/aws/elasticbeanstalk/<environment-name>/IIS-Log

Streamen von Instance-Protokollen an CloudWatch Logs

Sie können das Streaming für Instance-Protokolle an CloudWatch Logs mithilfe der Elastic Beanstalk-Konsole, der EB CLI oder der Konfigurationsoptionen aktivieren.

Bevor Sie dies aktivieren, müssen Sie IAM-Berechtigungen zur Verwendung mit dem CloudWatch Logs-Agenten festlegen. Sie können die folgende benutzerdefinierte Richtlinien an das [Instance-Profil \(p. 22\)](#) anhängen, das Sie Ihrer Umgebung zuweisen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

```
}
```

Streamen von Instance-Protokollen mit der Elastic Beanstalk-Konsole

So streamen Sie Instance-Protokolle an CloudWatch Log

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Unter Instance log streaming to CloudWatch Logs (Streaming von Instance-Protokollen an CloudWatch-Protokolle):
 - Aktivieren Sie Log streaming (Protokoll-Streaming).
 - Setzen Sie Retention (Aufbewahrungszeitdauer) auf die Anzahl an Tagen, wie lange die Protokolle aufbewahrt werden sollen.
 - Wählen Sie die Einstellung Lifecycle (Lebenszyklus), die bestimmt, ob die Protokolle gespeichert werden, nachdem die Umgebung beendet wurde.
6. Wählen Sie Apply (Anwenden) aus.

Nachdem Sie das Protokoll-Streaming aktiviert haben, können Sie zur Konfigurationskategorie oder -seite Software zurückkehren und dort den Link Log Groups (Protokollgruppen) suchen. Klicken Sie auf diesen Link, um Ihre Protokolle in der CloudWatch-Konsole anzuzeigen.

Streamen von Instance-Protokollen mit der EB-CLI

Um das Streaming von Instance-Protokollen an CloudWatch Logs mit der EB CLI zu aktivieren, verwenden Sie den Befehl [eb logs \(p. 1074\)](#).

```
$ eb logs --cloudwatch-logs enable
```

Sie können auch eb logs verwenden, um Protokolle von CloudWatch Logs abzurufen. Sie können alle Instance-Protokolle der Umgebung abrufen, oder die vielen Optionen des Befehls verwenden, um Untergruppen abzurufender Protokolle anzugeben. Der folgende Befehl beispielsweise ruft den vollständigen Satz an Instance-Protokollen für Ihre Umgebung auf und speichert diese in ein Verzeichnis unter .elasticbeanstalk/logs.

```
$ eb logs --all
```

Mit der Option --log-group können Sie Instance-Protokolle einer bestimmten Protokollgruppe abrufen, die einer bestimmten Protokolldatei einer Instance entspricht. Dazu müssen Sie den Namen der Protokollgruppe kennen, die der abzurufenden Protokolldatei entspricht. Sie finden diese Informationen in [Wie Elastic Beanstalk CloudWatch Logs einrichtet \(p. 883\)](#).

Streamen von Instance-Protokollen mit Konfigurationsdateien

Wenn Sie eine Umgebung erstellen oder aktualisieren, können Sie eine Konfigurationsdatei zum Einrichten und Konfigurieren des Streamings von Instance-Protokollen an CloudWatch Logs verwenden. Die folgende Beispiel-Konfigurationsdatei ermöglicht ein standardmäßiges Streaming der Instance-Protokolle. Elastic Beanstalk streamt den Standardsatz an Protokolldateien für die Plattform Ihrer Umgebung. Um das Beispiel zu verwenden, kopieren Sie den Text in eine Datei mit der Erweiterung `.config` im Verzeichnis `.ebextensions` auf der obersten Ebene Ihres Anwendungsquellpaketes.

```
option_settings:  
  - namespace: aws:elasticbeanstalk:cloudwatch:logs  
    option_name: StreamLogs  
    value: true
```

Benutzerdefiniertes Streamen von Protokolldateien

Die Elastic Beanstalk-Integration mit CloudWatch Logs unterstützt das Streaming von benutzerdefinierten Protokolldateien, die von Ihrer Anwendung generiert werden, nicht direkt. Um benutzerdefinierte Protokolle zu streamen, verwenden Sie eine Konfigurationsdatei, um den CloudWatch Logs-Agent direkt zu installieren und die Dateien, die gestreamt werden sollen, zu konfigurieren. Eine Beispielkonfigurationsdatei finden Sie unter [logs-streamtocloudwatch-linux.config](#).

Note

Das Beispiel funktioniert nicht auf der Windows-Plattform.

Weitere Informationen zur Konfiguration von CloudWatch Logs finden Sie in der [Referenz für den CloudWatch Logs-Agenten](#) im Amazon CloudWatch Logs-Benutzerhandbuch.

Fehlerbehebung bei der CloudWatch Logs-Integration

Wenn Sie einige der Instance-Protokollen der Umgebung nicht finden, die Sie in CloudWatch Logs erwarten, können Sie die folgenden häufig vorkommenden Probleme untersuchen:

- Ihre IAM-Rolle verfügt nicht über die erforderlichen IAM-Berechtigungen.
- Sie haben Ihre Umgebung in einer AWS-Region gestartet, die CloudWatch Logs nicht unterstützt.
- Eine Ihrer benutzerdefinierten Protokolldateien existiert nicht unter dem von Ihnen angegebenen Pfad.

Streaming von Integritätsinformationen der Elastic Beanstalk-Umgebung an Amazon CloudWatch Logs

Wenn Sie [erweiterte Integritätsberichte \(p. 824\)](#) für Ihre Umgebung aktivieren, können Sie die Umgebung so konfigurieren, dass sie Integritätsinformationen an CloudWatch Logs streamt. Dieses Streaming ist unabhängig vom Streaming von Amazon EC2-Instance-Protokollen. Dieses Thema beschreibt Informationen zum Streaming des Umgebungszustands. Weitere Informationen zum Streaming von Instance-Protokollen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon CloudWatch Logs \(p. 880\)](#).

Wenn Sie das Streaming der Umgebungsintegrität konfigurieren, erstellt Elastic Beanstalk eine CloudWatch Logs-Protokollgruppe für die Umgebungsintegrität. Der Name der Protokollgruppe ist `/aws/elasticbeanstalk/environment-name/environment-health.log`. Innerhalb dieser Protokollgruppe erstellt Elastic Beanstalk Protokollstreams mit dem Namen `YYYY-MM-DD#<hash-suffix>` (möglicherweise mehr als einen Protokollstream pro Datum).

Wenn sich der Integritätsstatus der Umgebung ändert, fügt Elastic Beanstalk einen Datensatz in den Integritätsprotokoll-Stream ein. Der Datensatz stellt den Zustandsübergang dar – den neuen Status und

eine Beschreibung der Ursache für die Änderung. Beispielsweise könnte sich der Zustand einer Umgebung auf "Severe" (Schwerwiegend) ändern, weil der Load Balancer ausgefallen ist. Eine Beschreibung der erweiterten Zustandsstatus finden Sie unter [Farben und Status in Zustandsangaben \(p. 841\)](#).

Voraussetzungen für das Streaming der Umgebungsintegrität an CloudWatch Logs

Um das Streaming der Umgebungsintegrität nach CloudWatch Logs zu aktivieren, müssen Sie die folgenden Bedingungen erfüllen:

- Platform (Plattform) – Sie müssen eine Plattformversion verwenden, die erweiterte Integritätsberichte unterstützt.
- Permissions (Berechtigungen) – Sie müssen bestimmte Berechtigungen im Zusammenhang mit der Protokollierung an Elastic Beanstalk erteilen, sodass es in Ihrem Namen agieren kann, um die Integritätsinformationen für Ihre Umgebung zu streamen. Wenn Ihre Umgebung keine Servicerolle verwendet, die Elastic Beanstalk dafür erstellt hat (`aws-elasticbeanstalk-service-role`), und auch nicht die serviceverknüpfte Rolle Ihres Kontos (`AWSServiceRoleForElasticBeanstalk`), stellen Sie sicher, dass Sie Ihrer benutzerdefinierten Servicerolle die folgenden Berechtigungen hinzufügen.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "logs:DescribeLogStreams",  
        "logs>CreateLogStream",  
        "logs:PutLogEvents"  
    ],  
    "Resource": "arn:aws:logs:*:log-group:/aws/elasticbeanstalk/*:log-stream:*"  
}
```

Streaming von Umgebungsintegritätsprotokollen an CloudWatch Logs

Sie können das Streaming der Umgebungsintegrität an CloudWatch Logs mithilfe der Elastic Beanstalk-Konsole, der EB CLI oder der Konfigurationsoptionen aktivieren.

Streamen von Umgebungsintegritätsprotokollen mit der Elastic Beanstalk-Konsole

So streamen Sie Umgebungsintegritätsprotokolle an CloudWatch Logs

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Konfigurationskategorie Monitoring (Überwachung) die Option Edit (Bearbeiten).
5. Stellen Sie sicher, dass unter Health reporting (Zustandsberichte) das System auf Enhanced (Erweitert) gesetzt ist.
6. Unter Health event streaming to CloudWatch Logs (Streaming von Zustandsereignissen an CloudWatch-Protokolle):

- Aktivieren Sie Log streaming (Protokoll-Streaming).
 - Setzen Sie Retention (Aufbewahrungsduer) auf die Anzahl an Tagen, wie lange die Protokolle aufbewahrt werden sollen.
 - Wählen Sie die Einstellung Lifecycle (Lebenszyklus), die bestimmt, ob die Protokolle gespeichert werden, nachdem die Umgebung beendet wurde.
7. Wählen Sie Apply (Anwenden) aus.

Nachdem Sie das Protokoll-Streaming aktiviert haben, können Sie zur Konfigurationskategorie oder -seite Monitoring (Überwachung) zurückkehren und dort den Link Log Group (Protokollgruppe) suchen. Klicken Sie auf diesen Link, um Ihre Umgebungsintegritätsprotokolle in der CloudWatch-Konsole anzuzeigen.

Streamen von Umgebungszustandsprotokollen mit der EB-CLI

Um das Streaming von Umgebungsintegritätsprotokollen an CloudWatch Logs mit der EB CLI zu aktivieren, verwenden Sie den Befehl [eb logs \(p. 1074\)](#).

```
$ eb logs --cloudwatch-logs enable --cloudwatch-log-source environment-health
```

Sie können auch eb logs verwenden, um Protokolle von CloudWatch Logs abzurufen. Der folgende Befehl beispielsweise ruft alle Zustandsprotokolle für Ihre Umgebung auf und speichert diese in ein Verzeichnis unter .elasticbeanstalk/logs.

```
$ eb logs --all --cloudwatch-log-source environment-health
```

Streamen von Umgebungszustandsprotokollen mit Konfigurationsdateien

Wenn Sie eine Umgebung erstellen oder aktualisieren, können Sie eine Konfigurationsdatei zum Einrichten und Konfigurieren des Streamings von Umgebungsintegritätsprotokollen an CloudWatch Logs verwenden. Um das nachfolgende Beispiel zu verwenden, kopieren Sie den Text in eine Datei mit der Dateinamenerweiterung .config im Verzeichnis .ebextensions auf der obersten Ebene Ihres Anwendungs-Quell-Bundles. Das Beispiel konfiguriert Elastic Beanstalk so, dass das Streaming der Umgebungsintegritätsprotokolle aktiviert ist, dass die Protokolle nach dem Beenden der Umgebung beibehalten und für 30 Tage gespeichert werden.

Example Konfigurationsdatei für das Streaming des Zustands

```
#####
## 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
```

Standardwerte und gültige Werte für die Option finden Sie unter [aws:elasticbeanstalk:cloudwatch:logs:health \(p. 684\)](#).

Verwenden von Elastic Beanstalk mit Amazon EventBridge

Amazon EventBridge überwacht Statusänderungsereignisse in Elastic Beanstalk-Ressourcen. Ereignisse aus Elastic Beanstalk werden nahezu in Echtzeit an EventBridge bereitgestellt. Sie können einfache Regeln schreiben, die angeben, welche Ereignisse für Sie interessant sind, einschließlich automatisierter Aktionen, die durchgeführt werden sollen, wenn ein Ereignis mit einer Regel übereinstimmt. Die automatischen Aktionen, die ausgelöst werden können, umfassen die folgenden Aktionen:

- Aufrufen einer AWS Lambda-Funktion
- Benachrichtigen eines Amazon SNS-Themas

Einige Beispiele für Elastic Beanstalk-Statusänderungsereignisse, die EventBridge überwacht, sind:

- Statusänderungen für Umgebungsoperationen wie das Erstellen, Aktualisieren oder Beenden einer Umgebung. Das Ereignis gibt an, ob die Statusänderung begonnen hat, erfolgreich war oder fehlgeschlagen ist.
- Statusänderungen für andere Ressourcen. Neben Umgebungen umfassen andere Ressourcen, die überwacht werden, Load Balancer, Auto Scaling-Gruppen und Instances.
- Integritätsübergang für Umgebungen. Das Ereignis gibt an, wo die Umgebungsintegrität von einem Integritätsstatus zum anderen übergegangen ist.
- Statusänderung für verwaltete Updates. Das Ereignis gibt an, ob die Statusänderung begonnen hat, erfolgreich war oder fehlgeschlagen ist.

Um bestimmte Elastic Beanstalk-Ereignisse zu erfassen, an denen Sie interessiert sind, definieren Sie ereignisspezifische Muster, mit denen EventBridge die Ereignisse erkennen kann. Ereignismuster können allgemein oder detaillierter sein, einschließlich spezifischer Elastic Beanstalk-Ereignisfelder und der übereinstimmenden Zeichenfolgenwerte. Der folgende Abschnitt enthält verschiedene, spezifische Beispiele für Elastic Beanstalk-Ereignismuster.

Eine Liste der Felder, die in Elastic Beanstalk-Ereignissen enthalten sind, und deren möglichen Zeichenfolgenwerten finden Sie unter [Elastic Beanstalk-Ereignisfeldzuordnung \(p. 894\)](#). Informationen darüber, wie EventBridge-Regeln mit Ereignismustern funktionieren, finden Sie unter [Ereignisse und Ereignismuster in EventBridge](#).

Überwachen des Ressourcenstatus mit EventBridge

Mit EventBridge können Sie Regeln erstellen, die zu ergreifende Aktionen definieren, wenn Elastic Beanstalk Benachrichtigungen für Statusänderungen sendet, die sich auf eine oder mehrere seiner Ressourcen beziehen. Sie können beispielsweise eine Regel erstellen, die Ihnen eine E-Mail-Nachricht sendet, wenn sich der Status einer Umgebung ändert.

So erstellen Sie eine Regel für Änderungen des Ressourcenstatus

1. Melden Sie sich bei AWS mit einem Konto an, das über Berechtigungen zur Verwendung von EventBridge und Elastic Beanstalk verfügt.
2. Öffnen Sie die Amazon EventBridge-Konsole unter <https://console.aws.amazon.com/events/>.
3. Wählen Sie Create rule (Regel erstellen) aus.
4. Geben Sie für die Regel einen Name (Namen) und optional eine Beschreibung ein.
5. Wählen Sie unter Define pattern (Muster definieren) die Option Event pattern (Ereignismuster) aus.

6. Wählen Sie unter Event matching pattern (Event-Matching-Muster) die Option Custom pattern (Benutzerdefiniertes Muster) aus.
7. Fügen Sie im Feld Event pattern (Ereignismuster) das folgende Muster hinzu und wählen Sie dann Save (Speichern) aus.

```
{  
    "source": [  
        "aws.elasticbeanstalk"  
    ],  
    "detail-type": [  
        "Elastic Beanstalk resource status change"  
    ]  
}
```

8. Wählen Sie im Abschnitt Select event bus (Ereignisbus auswählen) den zu verwendenden Ereignisbus aus. Wenn Sie noch keinen benutzerdefinierten Ereignisbus erstellt haben, wählen Sie den AWS default event bus (AWS Standard-Ereignisbus) aus.

Vergewissern Sie sich, dass Enable the rule on the selected event bus (Aktivieren Sie die Regel auf dem ausgewählten Ereignisbus) eingeschaltet ist.

9. Wählen Sie unter Select targets (Ziele auswählen) die durchzuführende Zielaktion aus, wenn ein Ereignis zur Änderung des Ressourcenstatus von Elastic Beanstalk empfangen wird.

Sie können beispielsweise ein Amazon Simple Notification Service (SNS)-Thema verwenden, um eine E-Mail oder SMS zu senden, wenn ein Ereignis eintritt. Dazu müssen Sie mit der Amazon SNS Konsole ein Amazon SNS-Thema erstellen. Weitere Informationen finden Sie unter [Verwenden von Amazon SNS für Benutzerbenachrichtigungen](#).

Important

Einige Zielaktionen erfordern möglicherweise die Nutzung anderer Services und verursachen zusätzliche Gebühren, wie z. B. der Amazon SNS- oder Lambda-Service. Für weitere Informationen zu AWS-Preisen siehe <http://aws.amazon.com/pricing/>. Einige Services sind im Umfang des kostenlosen Nutzungskontingents für AWS enthalten. Als Neukunde können Sie diese Services kostenlos testen. Weitere Informationen finden Sie unter <http://aws.amazon.com/free/>.

10. Wählen Sie optional Add target (Ziel hinzufügen) aus, um eine zusätzliche Zielaktion für die Ereignisregel anzugeben.
11. Wählen Sie Create (Erstellen) aus.

Sie können dieselben Schritte auch verwenden, um eine Aktion für andere Ereignisse mithilfe eines der folgenden Muster auszulösen:

- Statusänderung anderer Ressourcen für Load Balancer, Auto Scaling-Gruppen und Instances

```
{  
    "source": [  
        "aws.elasticbeanstalk"  
    ],  
    "detail-type": [  
        "Other resource status change"  
    ]  
}
```

- Statusänderung der Integrität für Umgebungen

```
{  
    "source": [  
        "aws.elasticbeanstalk"  
    ],  
    "detail-type": [  
        "Environment status change"  
    ]  
}
```

```
        "aws.elasticbeanstalk"
    ],
    "detail-type": [
        "Health status change"
    ]
}
```

- Statusänderung verwalteter Updates für Umgebungen

```
{
    "source": [
        "aws.elasticbeanstalk"
    ],
    "detail-type": [
        "Managed update status change"
    ]
}
```

- So erfassen Sie alle Ereignisse aus Elastic Beanstalk (schließen Sie den Abschnitt detail-type aus)

```
{
    "source": [
        "aws.elasticbeanstalk"
    ]
}
```

Beispiel für Elastic Beanstalk-Ereignisse

Es folgt ein Beispiel eines Elastic Beanstalk-Ereignisses für eine Statusänderung der Ressourcen:

```
{
    "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"
    }
}
```

Es folgt ein Beispiel eines Elastic Beanstalk-Ereignisses für eine Änderung des Integritätsstatus:

```
{
    "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"
}
}

```

Elastic Beanstalk-Ereignisfeldzuordnung

In der folgenden Tabelle werden Elastic Beanstalk-Ereignisfelder und ihre möglichen Zeichenfolgenwerte dem EventBridge-Feld `detail-type` zugeordnet. Weitere Informationen darüber, wie EventBridge mit Ereignismustern für einen Service funktioniert, finden Sie unter [Ereignisse und Ereignismuster in EventBridge](#).

EventBridge-Feld <code>detail-type</code>	Elastic Beanstalk-Feld Status	Elastic Beanstalk-Feld Severity (Schweregrad)	Elastic Beanstalk-Feld Message (Nachricht)
Statusänderung von Elastic Beanstalk-Ressourcen	Umgebungsstartung gestartet	INFO	createEnvironment wird gestartet.
	Umgebungsstartung erfolgreich	INFO	createEnvironment wurde erfolgreich abgeschlossen.
	Umgebungsstartung erfolgreich	INFO	Gestartete Umgebung: <Environment Name>. Es gab jedoch Probleme während des Starts. Einzelheiten finden Sie im Ereignisprotokoll.
	Umgebungsstartung fehlgeschlagen	ERROR	Die Umgebung konnte nicht gestartet werden.
	Umgebungs-Update gestartet	INFO	Das Umgebungs-Update wird gestartet.
	Umgebungs-Update erfolgreich	INFO	Das Umgebungs-Update wurde erfolgreich abgeschlossen.
	Umgebungs-Update fehlgeschlagen	ERROR	Die Konfiguration konnte nicht bereitgestellt werden.
	Umgebungsbedeutung gestartet	INFO	terminateEnvironment wird gestartet.
	Umgebungsbedeutung erfolgreich	INFO	terminateEnvironment wurde erfolgreich abgeschlossen.

EventBridge-Feld detail-type	Elastic Beanstalk-Feld Status	Elastic Beanstalk-Feld Severity (Schweregrad)	Elastic Beanstalk-Feld Message (Nachricht)
	Umgebungsbeendigung fehlgeschlagen	INFO	Der Schritt für die Umgebungsbeendigung ist fehlgeschlagen, da mindestens einer der Workflows zur Umgebungsbeendigung fehlgeschlagen ist.
Statusänderung anderer Ressourcen	Auto Scaling-Gruppe erstellt	INFO	createEnvironment wird gestartet.
	Auto Scaling-Gruppe gelöscht	INFO	createEnvironment wird gestartet.
	Instance hinzugefügt	INFO	Die Instance [i-123456789a12b1234] wurde Ihrer Umgebung hinzugefügt.
	Instance entfernt	INFO	Die Instance [i-123456789a12b1234] wurde aus Ihrer Umgebung entfernt.
	Load Balancer erstellt	INFO	Erstellter Load Balancer namens: <LB Name>
	Load Balancer gelöscht	INFO	Gelöschter Load Balancer namens: <LB Name>
Statusänderung der Integrität	Umgebungsintegrität geändert	INFO/WARN	Die Umgebungsintegrität hat zu <healthStatus> gewechselt.
	Umgebungsintegrität geändert	INFO/WARN	Die Umgebungsintegrität hat von <healthStatus> zu <healthStatus> gewechselt.
Statusänderung verwalteter Updates	Verwaltetes Update gestartet	INFO	Das verwaltete Plattform-Update wird durchgeführt.
	Verwaltetes Update fehlgeschlagen	INFO	Das verwaltete Update ist fehlgeschlagen, Wiederholung in %s Minuten.

Suchen und Verfolgen von Elastic Beanstalk-Ressourcen mit AWS Config

AWS Config bietet eine detaillierte Ansicht der Konfiguration der AWS-Ressourcen in Ihrem AWS-Konto. Sie können Ressourcenbeziehungen betrachten, einen Verlauf der Konfigurationsänderungen anzeigen und feststellen, wie sich Beziehungen und Konfigurationen im Lauf der Zeit ändern. Sie können AWS Config verwenden, um Regeln zu definieren, die Ressourcenkonfigurationen auf Datenkonformität prüfen.

Mehrere Elastic Beanstalk-Ressourcentypen sind in AWS Config integriert:

- Anwendungen
- Anwendungsversionen
- Umgebungen

Der folgende Abschnitt zeigt, wie AWS Config für die Aufzeichnung von Ressourcen dieser Typen konfiguriert wird.

Weitere Informationen über AWS Config finden Sie im [AWS Config-Entwicklerhandbuch](#). Preisinformationen finden Sie auf der [AWS Config-Preisinformationsseite](#).

Einrichten von AWS Config

Zur erstmaligen Einrichtung von AWS Config lesen die folgenden Themen im [AWS Config-Entwicklerhandbuch](#).

- Einrichten von AWS Config mit der Konsole
- Einrichten von AWS Config mit der AWS CLI

Konfigurieren von AWS Config zur Aufzeichnung von Elastic Beanstalk-Ressourcen

Standardmäßig zeichnet AWS Config die Konfigurationsänderungen für alle unterstützten Typen von regionalen Ressourcen auf, die in der Region erkannt werden, in der die Umgebung ausgeführt wird. Sie können AWS Config so anpassen, dass Änderungen nur für bestimmte Ressourcentypen oder nur Änderungen an globalen Ressourcen aufgezeichnet werden. Sie können beispielsweise AWS Config für die Aufzeichnung von Änderungen an Elastic Beanstalk-Ressourcen sowie einer Teilmenge anderer AWS-Ressourcen konfigurieren, die Elastic Beanstalk für Sie startet.

In der folgenden Abbildung wird die Seite Settings (Einstellungen) für AWS Config mit Elastic Beanstalk-Ressourcentypen gezeigt, die Sie für die Aufzeichnung wählen können: Application (Anwendung), ApplicationVersion (Anwendungsversion) und Environment (Umgebung).

The screenshot shows the 'Settings' page for AWS Config. At the top, it says 'Recording is on' with a 'Turn off' button. Below that is a section titled 'Resource types to record' with a note about selecting AWS resources for configuration changes. It has two sections: 'All resources' and 'Specific types'. Under 'All resources', there are two checkboxes: 'Record all resources supported in this region' (unchecked) and 'Include global resources (e.g., AWS IAM resources)' (unchecked). Under 'Specific types', there is a dropdown menu with 'Beanstalk' selected. A dropdown menu is open, showing 'ElasticBeanstalk' as the selected item, along with 'Application', 'ApplicationVersion', and 'Environment'. At the bottom, there is an 'Amazon S3 bucket' section and a note about creating a bucket.

Nachdem Sie einige Ressourcentypen ausgewählt haben, sieht die Liste Specific types (Spezifische Typen) folgendermaßen aus.

Resource types to record

Select the types of AWS resources for which you want AWS Config to record configuration changes. By default, AWS Config records configuration changes for all supported resources. You can also choose to record configuration changes for supported global resources in this region.

All resources Record all resources supported in this region ⓘ Include global resources (e.g., AWS IAM resources) ⓘ

Specific types ElasticBeanstalk: Application ElasticBeanstalk: Environment EC2: Instance

Informationen über regionale im Unterschied zu globalen Ressourcen sowie zum Anpassungsverfahren insgesamt finden Sie unter [Auswählen der Ressourcen, die AWS Config aufzeichnet](#).

Anzeigen von Elastic Beanstalk-Konfigurationsdetails in der AWS Config-Konsole

Sie können die AWS Config-Konsole verwenden, um nach Elastic Beanstalk-Ressourcen zu suchen und aktuelle und historische Details über deren Konfigurationen zu erhalten. Das folgende Beispiel zeigt, wie Sie Informationen über eine Elastic Beanstalk-Umgebung finden.

So finden Sie eine Elastic Beanstalk-Umgebung in der AWS Config-Konsole

1. Öffnen Sie die [AWS Config-Konsole](#).
2. Wählen Sie Resources aus.
3. Wählen Sie auf der Seite Resource inventory die Option Resourcesaus.
4. Öffnen Sie das Menü Resource type (Ressourcentyp) blättern Sie zu ElasticBeanstalk und wählen Sie einzelne oder mehrere Elastic Beanstalk-Ressourcentypen aus. Siehe 1 in der folgenden Abbildung.

Note

Wählen Sie weitere Ressourcentypen, um Konfigurationsdetails für andere als die von Elastic Beanstalk für die Anwendung erstellten Ressourcen anzuzeigen. Sie können beispielsweise Instance unter EC2 wählen.

5. Wählen Sie Look up. Siehe 2 in der folgenden Abbildung.

The screenshot shows the AWS Config Resource inventory interface. On the left, there is a sidebar with a tree view of resource types. The 'ElasticBeanstalk' node is expanded, and its children 'Application', 'ApplicationVersion', and 'Environment' are selected (indicated by checked checkboxes). A yellow circle labeled '1' points to the 'Environment' node. On the right, there is a search bar labeled 'Resource identifier (optional)' and a blue 'Look up' button. A yellow circle labeled '2' points to the 'Look up' button. Below the search area, there is a table with columns: 'Config timeline' (with a timeline icon), 'Compliance' (with a compliance icon), and 'Manage resource' (with a link icon). The table contains three rows of data.

Config timeline	Compliance	Manage resource
i-0abae959f6fb4b133	Compliant	Edit
arn:aws:elasticbeanstalk:us-east-1:270205402845:application/config-demo	--	Edit
e-yaumygtbwr	--	Edit

- Wählen Sie eine Ressourcen-ID in der Liste der Ressourcen, die AWS Config anzeigt.

The screenshot shows the AWS Config Resource inventory interface. On the left, there is a sidebar with a tree view of resource types. The 'EC2: Instance, ElasticBeanstalk: ...' node is selected (indicated by a checked checkbox). A yellow circle labeled '1' points to the 'EC2: Instance, ElasticBeanstalk: ...' node. Below the sidebar, there is a search bar labeled 'Resource identifier (optional)', a checkbox labeled 'Include deleted resources', and a blue 'Look up' button. A yellow circle labeled '2' points to the 'Look up' button. Below the search area, there is a table with columns: 'Resource type' (with a dropdown arrow), 'Config timeline' (with a timeline icon), 'Compliance' (with a compliance icon), and 'Manage resource' (with a link icon). The table contains three rows of data.

Resource type	Config timeline	Compliance	Manage resource
EC2 Instance	i-0abae959f6fb4b133	Compliant	Edit
ElasticBeanstalk Application	arn:aws:elasticbeanstalk:us-east-1:270205402845:application/config-demo	--	Edit
ElasticBeanstalk Environment	e-yaumygtbwr	--	Edit

AWS Config zeigt Konfigurationsdetails und andere Informationen über die von Ihnen gewählte Ressource an.

AWS Elastic Beanstalk Entwicklerhandbuch
Anzeigen von Elastic Beanstalk-
Konfigurationsdetails in der AWS Config-Konsole

The screenshot shows the AWS Config console interface. At the top, it displays the environment name "ElasticBeanstalk Environment e-yaumygtbwr" and the date "on February 09, 2018 4:03:54 PM Pacific Standard Time (UTC-08:00)". Below this is a timeline showing three configuration snapshots: "05th February 2018 4:34:35 PM", "06th February 2018 3:43:45 PM" (with a "Change" indicator), and "07th February 2018 11:43:44 PM". To the right are buttons for "Now" and "Changes".

Configuration Details (View Details):

- Amazon Resource Name: arn:aws:elasticbeanstalk:us-east-1:270205402845:environment/config-demo/ConfigDemo-env
- Resource type: AWS::ElasticBeanstalk::Environment
- Resource ID:** e-yaumygtbwr (highlighted with a yellow box)
- Resource name: ConfigDemo-env
- Availability zone: Not Applicable
- Created on: February 05, 2018 3:45:05 PM
- Tags (3): elasticbeanstalk:env..., elasticbeanstalk:envi..., Name:ConfigDemo...

Relationships (5)

Changes (7)

CloudTrail Events (0)

Die vollständigen Details der aufgezeichneten Konfiguration können Sie mit View Details (Details anzeigen) anzeigen.

A modal window is open, displaying the detailed configuration in JSON format. The "View Details" button is highlighted with a yellow box. The JSON content includes:

```
Object
version: "1.3"
accountId: "270205402845"
configurationItemCaptureTime: "2018-02-08T07:43:44.957Z"
configurationItemStatus: "OK"
configurationStateId: "1518075824957"
configurationItemMD5Hash: ""
arn:
"arn:aws:elasticbeanstalk:us-east-1:270205402845:environment/config-demo/ConfigDemo-env"
resourceType: "AWS::ElasticBeanstalk::Environment"
resourceId: "e-yaumygtbwr"
resourceName: "ConfigDemo-env"
awsRegion: "us-east-1"
availabilityZone: "Not Applicable"
resourceCreationTime: "2018-02-05T23:45:05.861Z"
tags: Object
elasticbeanstalk:environment-name: "ConfigDemo-env"
```

Buttons at the bottom of the modal include "Close" and "Show JSON".

Weitere Informationen darüber, wie Sie eine Ressource finden und Informationen auf dieser Seite anzeigen können, finden Sie unter [Anzeigen von AWS-Ressourcenkonfigurationen und Historie](#) im AWS Config-Entwicklerhandbuch.

Evaluierung von Elastic Beanstalk-Ressourcen mit AWS Config-Regeln

Sie können AWS Config-Regeln erstellen, die die idealen Konfigurationseinstellungen für Ihre Elastic Beanstalk-Ressourcen darstellen. Sie können vordefinierte verwaltete AWS Config-Regeln verwenden oder benutzerdefinierte Regeln erstellen. AWS Config verfolgt die Änderungen der Ressourcenkonfiguration kontinuierlich, um zu ermitteln, ob diese Änderungen gegen eine der Bedingungen in den Regeln verstößen. Die AWS Config-Konsole zeigt den Konformitätsstatus Ihrer Regeln und Ressourcen an.

Wenn eine Ressource gegen eine Regel verstößt und als nicht konform gekennzeichnet ist, kann AWS Config Sie mit einem [Amazon Simple Notification Service \(Amazon SNS\)](#)-Thema benachrichtigen. Um die Daten in diesen AWS Config-Benachrichtigungen programmgesteuert zu nutzen, verwenden Sie eine [Amazon Simple Queue Service \(Amazon SQS\)](#)-Warteschlange als Benachrichtigungsendpunkt für das Amazon SNS-Thema. Sie können beispielsweise Code schreiben, der einen Workflow startet, wenn die Auto Scaling-Gruppenkonfiguration der Umgebung geändert wird.

Weitere Informationen zum Einrichten und Verwenden von Regeln finden Sie unter [Auswerten von Ressourcen mit AWS Config-Regeln](#) im AWS Config-Entwicklerhandbuch.

Elastic Beanstalk mit Amazon DynamoDB verwenden

Amazon DynamoDB ist ein vollständig verwalteter NoSQL-Datenbank-Service, der schnelle und planbare Leistung mit nahtloser Skalierbarkeit bereitstellt. Als Entwickler können Sie mit DynamoDB eine Datenbanktabelle erstellen, um beliebige Datenmengen zu speichern und abzurufen und jeglichen Anforderungsdatenverkehr zu verarbeiten. DynamoDB verteilt die Daten und den Datenverkehr für die Tabelle automatisch auf eine ausreichende Anzahl von Servern, um die vom Kunden angegebene Anforderungskapazität zu erreichen und die gespeicherte Datensumme zu verarbeiten und dabei gleichzeitig eine konsistente und schnelle Leistung beizubehalten. Alle Datenelemente werden in Festkörperlaufwerken (Solid State Drives, SSDs) gespeichert und automatisch über mehrere Availability Zones in einer AWS-Region repliziert, um eine integrierte Hochverfügbarkeit und Datenhaltbarkeit bereitzustellen.

Wenn Sie [regelmäßige Aufgaben](#) (p. 519) in einer Worker-Umgebung verwenden, erstellt Elastic Beanstalk eine DynamoDB-Tabelle und verwendet diese, um die Leader-Wahl durchzuführen und Informationen über die Aufgabe zu speichern. Jede Instance in der Umgebung versucht alle paar Sekunden in die Tabelle zu schreiben, um Leader zu werden und die Aufgabe wie geplant durchzuführen.

Sie können [Konfigurationsdateien](#) (p. 722) zum Erstellen einer DynamoDB-Tabelle für Ihre Anwendung verwenden. Unter [eb-node-express-sample](#) auf GitHub finden Sie eine Node.js-Beispielanwendung, die eine Tabelle mit einer Konfigurationsdatei erstellt und diese mit dem AWS SDK for Node.js verbindet. Eine Beispielanleitung für die Verwendung von DynamoDB mit PHP finden Sie unter [Beispiel: DynamoDB, CloudWatch und SNS](#) (p. 760). Ein Beispiel, das AWS SDK for Java verwendet, finden Sie unter [Manage Tomcat Session State with DynamoDB](#) in der AWS SDK for Java-Dokumentation.

Beim Erstellen einer DynamoDB-Tabelle mit Konfigurationsdateien wird die Tabelle nicht mit dem Lebenszyklus der Umgebung verknüpft und nicht gelöscht, wenn Sie Ihre Umgebung beenden. Um sicherzustellen, dass personenbezogene Informationen nicht unnötig beibehalten werden, löschen Sie alle Datensätze, die Sie nicht mehr benötigen, oder löschen die Tabelle.

Weitere Informationen über DynamoDB finden Sie im [Entwicklerhandbuch von DynamoDB](#).

Verwenden von Elastic Beanstalk mit Amazon ElastiCache

Amazon ElastiCache ist ein Webservice, mit dem verteilte In-Memory-Cache-Umgebungen in der Cloud eingerichtet, verwaltet und skaliert werden können. Es bietet einen hochleistungsfähigen, anpassbaren und wirtschaftlichen In-Memory-Cache ohne die Komplexität, die mit der Bereitstellung und Verwaltung einer Umgebung eines verteilten Cache verbunden ist. ElastiCache ist protokollkompatibel mit Memcached- und Redis-Umgebungen. Deshalb werden der Code, die Anwendungen und die meisten Tools, die Sie heute in vorhandenen Memcached- oder Redis-Umgebungen verwenden, nahtlos mit dem Service zusammenarbeiten. Weitere Informationen zu ElastiCache finden Sie auf der Produktseite von [Amazon ElastiCache](#).

So verwenden Sie Elastic Beanstalk mit Amazon ElastiCache

1. Erstellen Sie ein ElastiCache-Cluster.
 - Anweisungen zum Erstellen eines ElastiCache-Clusters mit Redis finden Sie unter [Erste Schritte mit Amazon ElastiCache for Redis](#) im ElastiCache für Redis-Benutzerhandbuch.
 - Anweisungen zum Erstellen eines ElastiCache-Clusters mit Memcached finden Sie unter [Erste Schritte mit Amazon ElastiCache for Memcached](#) im ElastiCache for Memcached-Benutzerhandbuch.
2. Konfigurieren Sie Ihre ElastiCache-Sicherheitsgruppe, um Zugriff von der Amazon EC2-Sicherheitsgruppe zu erlauben, die von Ihrer Elastic Beanstalk-Anwendung verwendet wird. Anweisungen dazu, wie Sie den Namen Ihrer EC2-Sicherheitsgruppe mithilfe der AWS-Managementkonsole finden können, finden Sie unter [Sicherheitsgruppen \(p. 535\)](#) auf der Dokumentenseite EC2-Instances.
 - Weitere Informationen zu Redis finden Sie im ElastiCache für Redis-Benutzerhandbuch unter [Zugriff autorisieren](#).
 - Weitere Informationen zu Memcached finden Sie im ElastiCache für Memcached-Benutzerhandbuch unter [Zugriff autorisieren](#).

Sie können Konfigurationsdateien verwenden, um Ihre Elastic Beanstalk-Umgebung an ElastiCache anzupassen. Beispiele für Konfigurationsdateien, die ElastiCache mit Elastic Beanstalk integrieren, finden Sie unter [Beispiel: ElastiCache \(p. 752\)](#).

Elastic Beanstalk mit Amazon Elastic File System verwenden

Mit Amazon Elastic File System (Amazon EFS) können Sie Netzwerk-Dateisysteme erstellen, die von Instances über mehrere Availability Zones hinweg gemountet werden können. Ein Amazon EFS-Dateisystem ist eine AWS-Ressource, die Sicherheitsgruppen zum Steuern des Zugriffs über das Netzwerk in Ihrem Standard- oder benutzerdefinierten VPC verwendet.

In einer Elastic Beanstalk-Umgebung können Sie mit Amazon EFS ein freigegebenes Verzeichnis erstellen, in dem Dateien gespeichert werden, die vom Benutzer Ihrer Anwendung hochgeladen oder geändert werden. Ihre Anwendung kann ein bereitgestelltes Amazon EFS-Volume wie einen lokalen Speicher behandeln, sodass Sie Ihren Anwendungscode nicht ändern müssen, um auf mehrere Instances zu erweitern.

Weitere Informationen über Amazon EFS finden Sie im [Amazon Elastic File System-Benutzerhandbuch](#).

Abschnitte

- [Konfigurationsdateien \(p. 902\)](#)
- [Verschlüsselte Dateisysteme \(p. 902\)](#)
- [Beispielanwendungen \(p. 902\)](#)
- [Bereinigen von Dateisystemen \(p. 903\)](#)

Konfigurationsdateien

Elastic Beanstalk bietet [Konfigurationsdateien \(p. 722\)](#), die Sie zum Erstellen und Einbinden von Amazon EFS-Dateisystemen verwenden können. Sie können ein Amazon EFS-Volume als Teil Ihrer Umgebung erstellen oder ein Amazon EFS-Volume mounten, welches Sie unabhängig von Elastic Beanstalk erstellt haben.

- [storage-efs-createfilesystem.config](#) – Verwendet den Resources-Schlüssel zum Erstellen eines neuen Dateisystems und zum Mounten von Punkten in Amazon EFS. Alle Instances in Ihrer Umgebung können eine Verbindung zum selben Dateisystem herstellen, um einen gemeinsamen, skalierbaren Speicher zu erhalten. Das Mounting des Dateisystems auf der jeweiligen Instance erfolgt über [storage-efs-mountfilesystem.config](#).

Interne Ressourcen

Alle Ressourcen, die Sie mit Konfigurationsdateien erstellen, sind an den Lebenszyklus Ihrer Umgebung gebunden und gehen verloren, wenn Sie Ihre Umgebung beenden oder die Konfigurationsdatei entfernen.

- [storage-efs-mountfilesystem.config](#) – Mounten eines Amazon EFS-Dateisystems an einen lokalen Pfad auf den Instances in Ihrer Umgebung. Sie können das Volume als Teil der Umgebung mit [storage-efs-createfilesystem.config](#) oder außerhalb Ihrer Umgebung über die Amazon EFS-Konsole, AWS-CLI oder das AWS-SDK erstellen.

Um die Konfigurationsdateien zu verwenden, beginnen Sie mit dem Erstellen Ihres Amazon EFS-Dateisystems mit [storage-efs-createfilesystem.config](#). Befolgen Sie die Anweisungen in der Konfigurationsdatei und fügen Sie sie zum [.ebextensions \(p. 722\)](#)-Verzeichnis in Ihrem Quellcode hinzu, um das Dateisystem in Ihrer VPC zu erstellen.

Stellen Sie den aktualisierte Quellcode für Ihre Elastic Beanstalk-Umgebung bereit, um zu bestätigen, dass das Dateisystem erfolgreich erstellt wurde. Fügen Sie anschließend die [storage-efs-mountfilesystem.config](#) hinzu, um das Dateisystem für die Instances in Ihrer Umgebung bereitzustellen. Erfolgt dies in zwei separaten Bereitstellungen, wird sichergestellt, dass das Dateisystem intakt bleibt, wenn der Bereitstellungsvorgang fehlschlägt. Wenn Sie beides in derselben Bereitstellung durchführen, verursacht ein Problem in einem der Schritte, dass das Dateisystem beendet wird wenn die Bereitstellung fehlschlägt.

Verschlüsselte Dateisysteme

Amazon EFS unterstützt verschlüsselte Dateisysteme. Die in diesem Thema beschriebene Konfigurationsdatei [storage-efs-createfilesystem.config](#) definiert zwei benutzerdefinierte Optionen, mit deren Hilfe Sie ein Amazon EFS-verschlüsseltes Dateisystem erstellen können. Für weitere Informationen befolgen Sie die Anweisungen in der Konfigurationsdatei.

Beispielanwendungen

Elastic Beanstalk bietet auch Beispielanwendungen, die Amazon EFS für gemeinsam genutzten Speicher verwenden. Die beiden Projekte sind Konfigurationsdateien, die Sie mit einem Standard-WordPress- oder

Drupal-Installationsprogramm verwenden können, um einen Blog oder ein anderes Content-Management-System in einer lastverteilten Umgebung auszuführen. Wenn ein Benutzer ein Foto oder anderen Medien hochlädt, werden diese auf einem Amazon EFS-Dateisystem gespeichert. Somit muss kein Plugin verwendet werden, um hochgeladene Dateien in Amazon S3 zu speichern.

- [Load Balanced WordPress](#) – Konfigurationsdateien für die sichere Installation und Ausführung von WordPress in einer lastverteilten AWS Elastic Beanstalk-Umgebung.
- [Load Balanced Drupal](#) – Konfigurationsdateien und Anweisungen für die sichere Installation und Ausführung von Drupal in einer lastverteilten AWS Elastic Beanstalk-Umgebung.

Bereinigen von Dateisystemen

Wenn Sie ein Amazon EFS-Dateisystem unter Verwendung einer Konfigurationsdatei als Teil Ihrer Elastic Beanstalk-Umgebung erstellt haben, entfernt Elastic Beanstalk das Dateisystem, wenn Sie die Umgebung terminieren. Um die Speicherkosten einer laufenden Anwendung zu minimieren, löschen Sie regelmäßig Dateien, die Ihre Anwendung nicht benötigt, oder stellen sicher, dass der Anwendungscode den Dateilebenszyklus korrekt beibehält.

Wenn Sie darüber hinaus ein Amazon EFS-Dateisystem außerhalb einer Elastic Beanstalk-Umgebung erstellt und in den Instances der Umgebung gemountet haben, beachten Sie außerdem, dass Elastic Beanstalk das Dateisystem nicht entfernt, wenn Sie die Umgebung beenden. Um sicherzustellen, dass personenbezogene Informationen nicht unnötig gespeichert werden, löschen Sie Dateien, die Ihre Anwendung gespeichert hat, wenn Sie sie nicht mehr benötigen, oder entfernen das Dateisystem.

Elastic Beanstalk mit AWS Identity and Access Management verwenden

Sorgen Sie mit AWS Identity and Access Management (IAM) für eine sichere Zugangskontrolle zu Ihren AWS-Ressourcen. In diesem Abschnitt finden Sie Referenzmaterialien für die Nutzung der Richtlinien, Instance-Profile und Servicerollen von IAM.

Eine Übersicht über die Berechtigungen finden Sie unter [Servicerollen, Instance-Profile und Benutzerrichtlinien \(p. 21\)](#). In den meisten Umgebungen verfügen die Service-Rolle und das Instance-Profil, zu deren Erstellung Sie von der Elastic Beanstalk-Konsole beim Starten der ersten Umgebung aufgefordert werden, über alle erforderlichen Berechtigungen. Ebenso enthalten auch die von Elastic Beanstalk bereitgestellten [verwalteten Richtlinien \(p. 926\)](#) für Vollzugriff und schreibgeschützten Zugriff alle Benutzerberechtigungen, die bei der täglichen Arbeit benötigt werden.

Umfassende Informationen zu AWS-Berechtigungen finden Sie im [IAM-Benutzerhandbuch](#).

Themen

- [Elastic Beanstalk Instance-Profile verwalten \(p. 904\)](#)
- [Rollen von Elastic Beanstalk Service verwalten \(p. 909\)](#)
- [Servicegebundene Rollen für Elastic Beanstalk verwenden \(p. 915\)](#)
- [Elastic Beanstalk-Benutzerrichtlinien verwalten \(p. 926\)](#)
- [Amazon-Ressourcennamen-Format für Elastic Beanstalk \(p. 933\)](#)
- [Ressourcen und Bedingungen für Elastic Beanstalk-Aktionen \(p. 934\)](#)
- [Tags zur Steuerung des Zugriffs auf Elastic Beanstalk-Ressourcen verwenden \(p. 960\)](#)
- [Beispielrichtlinien basierend auf verwalteten Richtlinien \(p. 963\)](#)
- [Beispielrichtlinien auf Basis von Ressourcenberechtigungen \(p. 966\)](#)

Elastic Beanstalk Instance-Profile verwalten

Ein Instance-Profil ist ein Container für eine AWS Identity and Access Management (IAM)-Rolle, mit dem eine Amazon EC2-Instance bei ihrem Start Rolleninformationen erhält. Beim Starten einer Umgebung mit der Elastic Beanstalk-Konsole oder der EB-CLI erstellt Elastic Beanstalk ein Instance-Standardprofil namens `aws-elasticbeanstalk-ec2-role` und weist diesem verwaltete Richtlinien mit Standardberechtigungen zu.

Elastic Beanstalk stellt drei verwaltete Richtlinien bereit: eine für die Webserverebene, eine für die Worker-Ebene und eine mit weiteren Berechtigungen, die für Multicontainer-Docker-Umgebungen benötigt werden. Die Konsole weist all diese Richtlinien der Rolle zu, die dem Instance-Standardprofil angefügt ist. Nachfolgend finden Sie die Richtlinien.

Richtlinien für verwaltetes Instance-Profil

- AWSElasticBeanstalkWebTier – Damit erhält die Anwendung die Berechtigungen, um Protokolle in Amazon S3 und Debugging-Informationen in AWS X-Ray hochzuladen.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "BucketAccess",  
            "Action": [  
                "s3:Get*",  
                "s3>List*",  
                "s3:PutObject"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:s3:::elasticbeanstalk-*",  
                "arn:aws:s3:::elasticbeanstalk-*/*"  
            ]  
        },  
        {  
            "Sid": "XRayAccess",  
            "Action": [  
                "xray:PutTraceSegments",  
                "xray:PutTelemetryRecords",  
                "xray:GetSamplingRules",  
                "xray:GetSamplingTargets",  
                "xray:GetSamplingStatisticSummaries"  
            ],  
            "Effect": "Allow",  
            "Resource": "*"  
        },  
        {  
            "Sid": "CloudWatchLogsAccess",  
            "Action": [  
                "logs:PutLogEvents",  
                "logs>CreateLogStream",  
                "logs:DescribeLogStreams",  
                "logs:DescribeLogGroups"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:logs:*:*:log-group:/aws/elasticbeanstalk*"  
            ]  
        },  
        {  
            "Sid": "ElasticBeanstalkHealthAccess",  
            "Action": [  
                "elasticbeanstalk:PutInstanceStatistics"  
            ]  
        }  
    ]  
}
```

```

        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:elasticbeanstalk::*:application/*",
            "arn:aws:elasticbeanstalk::*:environment/*"
        ]
    }
}

```

- AWSElasticBeanstalkWorkerTier – Damit werden die Berechtigungen zum Hochladen von Protokollen, Debuggen, Veröffentlichen von Metriken sowie für Worker-Instance-Aufgaben wie Warteschlangenverwaltung, Leader-Wahl und regelmäßige Aufgaben erteilt.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "MetricsAccess",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Sid": "XRayAccess",
            "Action": [
                "xray:PutTraceSegments",
                "xray:PutTelemetryRecords",
                "xray:GetSamplingRules",
                "xray:GetSamplingTargets",
                "xray:GetSamplingStatisticSummaries"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Sid": "QueueAccess",
            "Action": [
                "sns:ChangeMessageVisibility",
                "sns:DeleteMessage",
                "sns:ReceiveMessage",
                "sns:SendMessage"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Sid": "BucketAccess",
            "Action": [
                "s3:Get*",
                "s3>List*",
                "s3:PutObject"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-*",
                "arn:aws:s3:::elasticbeanstalk-*/*"
            ]
        },
        {
            "Sid": "DynamoPeriodicTasks",
            "Action": [
                "dynamodb:BatchGetItem",

```

```
        "dynamodb:BatchWriteItem",
        "dynamodb>DeleteItem",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:Query",
        "dynamodb:Scan",
        "dynamodb:UpdateItem"
    ],
    "Effect": "Allow",
    "Resource": [
        "arn:aws:dynamodb:*::table/*-stack-AWSEBWorkerCronLeaderRegistry*"
    ]
},
{
    "Sid": "CloudWatchLogsAccess",
    "Action": [
        "logs:PutLogEvents",
        "logs>CreateLogStream"
    ],
    "Effect": "Allow",
    "Resource": [
        "arn:aws:logs:*::log-group:/aws/elasticbeanstalk*"
    ]
},
{
    "Sid": "ElasticBeanstalkHealthAccess",
    "Action": [
        "elasticbeanstalk:PutInstanceStatistics"
    ],
    "Effect": "Allow",
    "Resource": [
        "arn:aws:elasticbeanstalk:*::application/*",
        "arn:aws:elasticbeanstalk:*::environment/*"
    ]
}
]
```

- AWSElasticBeanstalkMulticontainerDocker – Gewährt Berechtigungen für den Amazon Elastic Container Service zur Koordinierung von Cluster-Aufgaben.

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

Damit die EC2-Instances in der Umgebung die Rolle `aws-elasticbeanstalk-ec2-role` übernehmen können, gibt das Instance-Profil Amazon EC2 wie folgt als vertrauenswürdige Entität in der Vertrauensstellungsrichtlinie an:

```
{  
    "Version": "2008-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "ec2.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

Wenn Sie die Berechtigungen anpassen möchten, können Sie entweder Richtlinien zur Rolle hinzufügen, die dem Instance-Standardprofil angefügt ist, oder ein eigenes Instance-Profil mit eingeschränkten Berechtigungen erstellen.

Abschnitte

- [Überprüfen der dem Instance-Standardprofil zugeordneten Berechtigungen \(p. 907\)](#)
- [Aktualisieren eines veralteten Instance-Standardprofils \(p. 907\)](#)
- [Hinzufügen von Berechtigungen zum Instance-Standardprofil \(p. 908\)](#)
- [Erstellen eines Instance-Profils \(p. 908\)](#)
- [Instance-Profile mit Amazon Linux 2-Plattformen \(p. 909\)](#)

Überprüfen der dem Instance-Standardprofil zugeordneten Berechtigungen

Die Berechtigungen, die dem Instance-Standardprofil zugeordnet sind, können je nach Erstellungsdatum, Datum des letzten Umgebungsstarts und verwendetem Client unterschiedlich sein. Die Berechtigungen für das Instance-Standardprofil können Sie in der IAM-Konsole überprüfen.

So überprüfen Sie die Berechtigungen des Instance-Standardprofils

1. Öffnen Sie die Seite [Roles \(Rollen\)](#) in der IAM-Konsole.
2. Wählen Sie `aws-elasticbeanstalk-ec2-role` aus.
3. Prüfen Sie auf der Registerseite [Permissions \(Berechtigungen\)](#) die der Rolle angefügte Liste von Richtlinien.
4. Um die von einer Richtlinie erteilten Berechtigungen anzuzeigen, wählen Sie die Richtlinie aus.

Aktualisieren eines veralteten Instance-Standardprofils

Falls benötigte Berechtigungen für das Instance-Standardprofil fehlen, können Sie es aktualisieren, indem Sie in der Elastic Beanstalk Environment Management Console eine [neue Umgebung erstellen \(p. 435\)](#).

Alternativ können Sie die verwalteten Richtlinien manuell zu der dem Instance-Standardprofil angefügten Rolle hinzufügen.

Hinzufügen von verwalteten Richtlinien zur Rolle, die dem Instance-Standardprofil angefügt ist

1. Öffnen Sie die Seite [Roles \(Rollen\)](#) in der IAM-Konsole.

2. Wählen Sie aws-elasticbeanstalk-ec2-role aus.
3. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Option Attach policies (Richtlinien anfügen) aus.
4. Geben Sie **AWSElasticBeanstalk** ein, um die Richtlinien zu filtern.
5. Wählen Sie die folgenden Richtlinien und anschließend Attach Policies (Richtlinien anfügen) aus:
 - AWSElasticBeanstalkWebTier
 - AWSElasticBeanstalkWorkerTier
 - AWSElasticBeanstalkMulticontainerDocker

Hinzufügen von Berechtigungen zum Instance-Standardprofil

Falls die Anwendung auf AWS-APIs oder Ressourcen zugreift, für die im Instance-Standardprofil keine Berechtigungen erteilt werden, können Sie in der IAM-Konsole entsprechende Richtlinien hinzufügen.

Hinzufügen von Richtlinien zur Rolle, die dem Instance-Standardprofil angefügt ist

1. Öffnen Sie die Seite [Roles \(Rollen\)](#) in der IAM-Konsole.
2. Wählen Sie aws-elasticbeanstalk-ec2-role aus.
3. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Option Attach policies (Richtlinien anfügen) aus.
4. Wählen Sie die verwaltete Richtlinie für die zusätzlichen Services aus, die von der Anwendung verwendet werden. Zum Beispiel `AmazonS3FullAccess` oder `AmazonDynamoDBFullAccess`.
5. Wählen Sie Attach policy aus.

Erstellen eines Instance-Profil

Ein Instance-Profil ist ein Wrapper für eine IAM-Standardrolle, damit eine EC2-Instance die Rolle übernehmen kann. Sie können zusätzliche Instance-Profile erstellen, um Berechtigungen für unterschiedliche Anwendungen anzupassen, oder Sie können ein Instance-Profil erstellen, das keinerlei Berechtigungen für eine Worker-Ebene oder für Multicontainer-Docker-Umgebungen erteilt, sofern Sie diese Funktionen nicht nutzen.

So erstellen Sie ein Instance-Profil

1. Öffnen Sie die Seite [Roles \(Rollen\)](#) in der IAM-Konsole.
2. Wählen Sie Create role aus.
3. Wählen Sie in AWS service (AWS-Service) EC2 (EC2) aus.
4. Wählen Sie Next: Permissions aus.
5. Fügen Sie die entsprechenden von Elastic Beanstalk bereitgestellten verwalteten Richtlinien sowie weitere Richtlinien hinzu, um der Anwendung die erforderlichen Berechtigungen zu erteilen.
6. Wählen Sie Next: Tags (Weiter: Tags) aus.
7. (Optional) Fügen Sie der Rolle Tags hinzu.
8. Klicken Sie auf Weiter: Prüfen.
9. Geben Sie einen Namen für die Rolle ein.
10. Wählen Sie Create role aus.

Instance-Profile mit Amazon Linux 2-Plattformen

Amazon Linux 2-Plattformen benötigen ein Instance-Profil für den ordnungsgemäßen Betrieb. Beispielsweise aktivieren alle Amazon Linux 2-Plattformversionen standardmäßig die erweiterte Integrität während der Umgebungserstellung. Instances benötigen die richtigen Berechtigungen, um erweiterte Integritätsinformationen zu sammeln und zu melden.

Rollen von Elastic Beanstalk Service verwwalten

Um Ihre Umgebung zu verwalten und zu überwachen, führt AWS Elastic Beanstalk in Ihrem Namen Aktionen mit den Ressourcen der Umgebung durch. AWS Elastic Beanstalk benötigt bestimmte Berechtigungen zur Durchführung dieser Aktionen und übernimmt die AWS-Service Identity and Access Management (IAM)-Rollen, um diese Berechtigungen zu erhalten.

Beim Annehmen einer Servicerolle muss Elastic Beanstalk stets temporäre Sicherheitsanmeldeinformationen verwenden. Um diese Anmeldeinformationen zu erhalten, sendet Elastic Beanstalk eine Anfrage an AWS Security Token Service (AWS STS) auf einem regionsspezifischen Endpunkt. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen](#) im IAM-Benutzerhandbuch.

Note

Wenn der AWS STS-Endpunkt der Region Ihrer Umgebung deaktiviert ist, sendet Elastic Beanstalk die Anforderung an einen alternativen Endpunkt, der nicht deaktiviert werden kann. Dieser Endpunkt ist einer anderen Region zugeordnet und die Anforderung ist daher eine regionenübergreifende Anforderung. Weitere Informationen finden Sie unter [AWS STS in einer AWS Region aktivieren und deaktivieren](#) im IAM-Benutzerhandbuch.

Servicerollen mit der Elastic Beanstalk-Konsole und EB CLI verwalten

Die Elastic Beanstalk-Konsole und die EB-Befehlszeilenschnittstelle erleichtern Ihnen das Einrichten von Servicerollen für Ihre Umgebung mit ausreichenden Berechtigungen. Sie erstellen eine Standardservicerolle und verwenden darin verwaltete Richtlinien.

Richtlinien zu verwalteten Servicerollen

Elastic Beanstalk bietet eine verwaltete Richtlinie für verbesserte Statusüberwachung (p. 824) und eine mit zusätzlichen Berechtigungen für verwaltete Plattformaktualisierungen (p. 496). Die Konsole und die EB-Befehlszeilenschnittstelle weisen beide Richtlinien der Standardservicerolle zu, die sie für Sie erstellen.

AWSElasticBeanstalkEnhancedHealth

Diese Richtlinie erteilt Berechtigungen für die Überwachung der Instance- und Umgebungsintegrität durch Elastic Beanstalk.

```
{  
    "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",
        "sns:GetQueueAttributes",
        "sns:GetQueueUrl",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeAutoScalingInstances",
        "autoscaling:DescribeScalingActivities",
        "autoscaling:DescribeNotificationConfigurations",
        "sns:Publish"
    ],
    "Resource": [
        "*"
    ]
}
]
```

[AWSelasticBeanstalkService](#)

Diese Richtlinie erteilt Elastic Beanstalk Berechtigungen, um Umgebungen in Ihrem Namen zu aktualisieren, um verwaltete Plattformaktualisierungen durchzuführen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudformationOperationsOnElasticBeanstalkStacks",
            "Effect": "Allow",
            "Action": [
                "cloudformation:*"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:stack/awseb-*",
                "arn:aws:cloudformation:*:stack/eb-*"
            ]
        },
        {
            "Sid": "AllowS3OperationsOnElasticBeanstalkBuckets",
            "Effect": "Allow",
            "Action": [
                "s3:/*"
            ],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-*",
                "arn:aws:s3:::elasticbeanstalk-*/"
            ]
        },
        {
            "Sid": "AllowOperations",
            "Effect": "Allow",
            "Action": [
                "autoscaling:AttachInstances",
                "autoscaling>CreateAutoScalingGroup",
                "autoscaling>CreateLaunchConfiguration",
                "autoscaling>DeleteLaunchConfiguration",
                "autoscaling>DeleteAutoScalingGroup",
                "autoscaling>DeletePolicy",
                "autoscaling>DeleteScheduledAction",
                "autoscaling>DescribeAccountLimits",
                "autoscaling>DescribeAutoScalingGroups",
                "autoscaling>DescribeAutoScalingInstances",
                "autoscaling>DescribeLaunchConfigurations",
                "autoscaling>DescribeLoadBalancers",
                "autoscaling>DescribeMetricCollectionMetrics"
            ]
        }
    ]
}
```

```
"autoscaling:DescribeNotificationConfigurations",
"autoscaling:DescribeScalingActivities",
"autoscaling:DescribeScheduledActions",
"autoscaling:DetachInstances",
"autoscaling:PutNotificationConfiguration",
"autoscaling:PutScalingPolicy",
"autoscaling:PutScheduledUpdateGroupAction",
"autoscaling:ResumeProcesses",
"autoscaling:SetDesiredCapacity",
"autoscaling:SuspendProcesses",
"autoscaling:TerminateInstanceInAutoScalingGroup",
"autoscaling:UpdateAutoScalingGroup",
"cloudwatch:PutMetricAlarm",
"ec2:AuthorizeSecurityGroupEgress",
"ec2:AuthorizeSecurityGroupIngress",
"ec2>CreateLaunchTemplate",
"ec2>CreateLaunchTemplateVersion",
"ec2>CreateSecurityGroup",
"ec2>DeleteLaunchTemplate",
"ec2>DeleteLaunchTemplateVersions",
"ec2>DeleteSecurityGroup",
"ec2:DescribeAccountAttributes",
"ec2:DescribeImages",
"ec2:DescribeInstances",
"ec2:DescribeKeyPairs",
"ec2:DescribeLaunchTemplates",
"ec2:DescribeLaunchTemplateVersions",
"ec2:DescribeSecurityGroups",
"ec2:DescribeSubnets",
"ec2:DescribeVpcs",
"ec2:RevokeSecurityGroupEgress",
"ec2:RevokeSecurityGroupIngress",
"ec2:RunInstances",
"ec2:TerminateInstances",
"ecs>CreateCluster",
"ecs>DeleteCluster",
"ecs:DescribeClusters",
"ecs:RegisterTaskDefinition",
"elasticbeanstalk:*",
"elasticloadbalancing:ApplySecurityGroupsToLoadBalancer",
"elasticloadbalancing:ConfigureHealthCheck",
"elasticloadbalancing>CreateLoadBalancer",
"elasticloadbalancing>DeleteLoadBalancer",
"elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
"elasticloadbalancing:DescribeInstanceHealth",
"elasticloadbalancing:DescribeLoadBalancers",
"elasticloadbalancing:DescribeTargetHealth",
"elasticloadbalancing:RegisterInstancesWithLoadBalancer",
"iam>ListRoles",
"iam>PassRole",
"logs>CreateLogGroup",
"logs>PutRetentionPolicy",
"rds>DescribeDBInstances",
"rds>DescribeOrderableDBInstanceStateOptions",
"s3:CopyObject",
"s3:GetObject",
"s3:GetObjectAcl",
"s3:GetObjectMetadata",
"s3>ListBucket",
"s3:listBuckets",
"sns>CreateTopic",
"sns>GetTopicAttributes",
"sns>ListSubscriptionsByTopic",
"sns>Subscribe",
"sqs>GetQueueAttributes",
"sqs>GetQueueUrl"
```

```
        ],
        "Resource": [
            "*"
        ]
    }
}
```

Elastic Beanstalk-Konsole verwenden

Wenn Sie in der Elastic Beanstalk-Konsole eine Umgebung starten, erstellt die Konsole eine Standardservicerolle mit dem Namen `aws-elasticbeanstalk-service-role` und fügt dieser verwaltete Richtlinien mit Standardberechtigungen an.

Damit Elastic Beanstalk die Rolle `aws-elasticbeanstalk-service-role` übernehmen kann, legt die Servicerolle Elastic Beanstalk als vertrauenswürdige Einheit in der Vertrauensbeziehungsrichtlinie fest.

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

Wenn Sie für Ihre Umgebung [verwaltete Plattformaktualisierungen \(p. 496\)](#) aktivieren, übernimmt Elastic Beanstalk eine separate Servicerolle für verwaltete Updates, um verwaltete Updates auszuführen. Standardmäßig verwendet die Elastic Beanstalk-Konsole die gleiche generierte Servicerolle, `aws-elasticbeanstalk-service-role`, als Servicerolle für verwaltete Updates. Bei einer Änderung der Standard-Servicerolle legt die Konsole die Servicerolle für verwaltete Updates auf die Verwendung der serviceverknüpften Rolle für verwaltete Updates fest, `AWSServiceRoleForElasticBeanstalkManagedUpdates`. Weitere Informationen zu serviceverknüpften Rollen finden Sie unter [the section called “Verwenden von serviceverknüpften Rollen” \(p. 913\)](#).

Note

Aufgrund von Berechtigungsproblemen gelingt es dem Elastic Beanstalk-Service nicht immer, diese serviceverknüpfte Rolle für Sie zu erstellen. Daher versucht die Konsole, sie explizit zu erstellen. Um sicherzustellen, dass Ihr Konto über diese servicebezogene Rolle verfügt, erstellen Sie mindestens einmal mithilfe der Konsole eine Umgebung und konfigurieren Sie verwaltete Updates, die aktiviert werden, bevor Sie die Umgebung erstellen.

Verwenden der EB CLI

Wenn Sie eine Umgebung mit dem [the section called “eb create” \(p. 1053\)](#)-Befehl der Elastic Beanstalk-Befehlszeilschnittstelle (EB CLI) starten und keine Servicerolle über die `--service-role`-Option angeben, erstellt Elastic Beanstalk die standardmäßige Servicerolle `aws-elasticbeanstalk-service-role`. Wenn die standardmäßige Servicerolle bereits vorhanden ist, verwendet Elastic Beanstalk diese für die neue Umgebung. Dies ist ähnlich wie bei der Elastic Beanstalk-Konsole.

Im Gegensatz zur Konsole können Sie eine Servicerolle für verwaltete Updates nicht über eine EB-CLI-Befehloption angeben. Wenn Sie für Ihre Umgebung verwaltete Updates aktivieren, legen Sie mithilfe von Konfigurationsoptionen eine Servicerolle für verwaltete Updates fest. Im folgenden Beispiel werden verwaltete Updates aktiviert und die Standardservicerolle als Servicerolle für verwaltete Updates verwendet.

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
```

Servicerollen mit der Elastic Beanstalk API verwalten

Wenn Sie eine Umgebung mithilfe der `CreateEnvironment`-Aktion der Elastic Beanstalk-API erstellen, geben Sie eine Servicerolle mit der `ServiceRole`-Konfigurationsoption im `aws:elasticbeanstalk:environment` (p. 686)-Namespace an. Einzelheiten zur Verwendung der erweiterten Zustandsprüfung mit der Elastic Beanstalk API finden Sie unter [Verwenden der erweiterten Integritätsberichte mit der Elastic Beanstalk-API](#) (p. 855).

Wenn Sie für Ihre Umgebung [Verwaltete Plattformaktualisierungen](#) (p. 496) aktivieren, können Sie über die Option `ServiceRoleForManagedUpdates` des Namespace `aws:elasticbeanstalk:managedactions` (p. 693) eine Servicerolle für verwaltete Updates angeben.

Verwenden von serviceverknüpften Rollen

Eine serviceverknüpfte Rolle ist eine eindeutige, von Elastic Beanstalk vordefinierte Servicerolle, die alle Berechtigungen einschließt, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen benötigt. Die serviceverknüpfte Rolle ist Ihrem Konto zugeordnet. Sie wird von Elastic Beanstalk einmalig erstellt und dann zum Erstellen weiterer Umgebungen wiederverwendet. Einzelheiten zur Verwendung von servicegebundenen Rollen mit Elastic Beanstalk-Umgebungen finden Sie unter [Servicegebundene Rollen für Elastic Beanstalk verwenden](#) (p. 915).

Wenn Sie mit der Elastic Beanstalk-API eine Umgebung erstellen und keine Servicerolle angeben, erstellt Elastic Beanstalk eine [überwachende serviceverknüpfte Rolle](#) (p. 916) für Ihr Konto, sofern diese nicht bereits vorhanden ist, und verwendet sie für die neue Umgebung. Sie können die serviceverknüpfte Rolle Ihres Kontos auch im Vorhinein mit IAM erstellen. Wenn Ihr Konto eine mit einem servicegebundenen Rolle für den Überwachungs-Service hat, können Sie damit eine Umgebung mit der Elastic Beanstalk API, der Elastic Beanstalk-Konsole oder der EB CLI erstellen.

Wenn Sie für die Umgebung [Verwaltete Plattformaktualisierungen](#) (p. 496) aktivieren und `AWSServiceRoleForElasticBeanstalkManagedUpdates` als Wert für die Option `ServiceRoleForManagedUpdates` des Namespace `aws:elasticbeanstalk:managedactions` (p. 693) angeben, erstellt Elastic Beanstalk zusätzlich eine [serviceverknüpfte Rolle für verwaltete Updates](#) (p. 921) für Ihr Konto, wenn noch nicht vorhanden, und verwendet diese für die Ausführung verwalteter Aktualisierungen für die neue Umgebung.

Note

Wenn Elastic Beanstalk versucht, die serviceverknüpften Rollen für verwaltete Updates und Überwachung für Ihr Konto zu erstellen, wenn Sie eine Umgebung erstellen, müssen Sie die Berechtigung `iam:CreateServiceLinkedRole`-besitzen. Wenn Sie nicht über diese

Genehmigung verfügen, schlägt das Erstellen der Umgebung fehl und es wird eine Meldung angezeigt, die das Problem erklärt.

Alternativ kann ein anderer Benutzer, der über die Berechtigung zum Erstellen von serviceverknüpften Rollen verfügt, IAM verwenden, um die serviceverknüpfte Rolle vorab zu erstellen. Anschließend können Sie Ihre Umgebung auch ohne die `iam:CreateServiceLinkedRole`-Berechtigung selbst erstellen.

Überprüfung der Berechtigungen der Standard-Servicerolle

Die Berechtigungen, die der Standard/Servicerolle gewährt sind, können je nach Erstellungsdatum, Datum des letzten Umgebungsstarts und verwendetem Client unterschiedlich sein. Die Berechtigungen, die von der Standard-Servicerolle in der IAM-Konsole gewährt wurden, überprüfen.

So überprüfen Sie die Berechtigungen der Standard-Servicerolle

1. Öffnen Sie die Seite [Roles \(Rollen\)](#) in der IAM-Konsole.
2. Wählen Sie `aws-elasticbeanstalk-service-role` aus.
3. Prüfen Sie auf der Registerseite [Permissions \(Berechtigungen\)](#) die der Rolle angefügte Liste von Richtlinien.
4. Um die von einer Richtlinie erteilten Berechtigungen anzuzeigen, wählen Sie die Richtlinie aus.

Aktualisieren einer veralteten Standard-Servicerolle

Falls die benötigten Berechtigungen für die Standard-Servicerolle fehlen, können Sie sie aktualisieren, indem Sie in der Elastic Beanstalk-Environment Management Console [eine neue Umgebung erstellen \(p. 435\)](#).

Alternativ können Sie die verwalteten Richtlinien manuell zur Standard-Servicerolle hinzufügen.

So fügen Sie verwaltete Richtlinien zur Standard-Servicerolle hinzu

1. Öffnen Sie die Seite [Roles \(Rollen\)](#) in der IAM-Konsole.
2. Wählen Sie `aws-elasticbeanstalk-service-role` aus.
3. Wählen Sie auf der Registerkarte [Permissions \(Berechtigungen\)](#) die Option [Attach policies \(Richtlinien anfügen\)](#) aus.
4. Geben Sie **AWSElasticBeanstalk** ein, um die Richtlinien zu filtern.
5. Wählen Sie die folgenden Richtlinien und anschließend [Attach Policies \(Richtlinien anfügen\)](#) aus:
 - `AWSElasticBeanstalkEnhancedHealth`
 - `AWSElasticBeanstalkService`

Hinzufügen von Berechtigungen zur Standard-Servicerolle

Wenn Ihre Anwendung Konfigurationsdateien umfasst, die sich auf AWS-Ressourcen beziehen, für die Berechtigungen nicht in der Standard-Servicerolle enthalten sind, benötigt Elastic Beanstalk möglicherweise zusätzliche Berechtigungen zur Auflösung dieser Referenzen, wenn die Konfigurationsdateien während einer verwalteten Aktualisierung verarbeitet werden. Wenn Berechtigungen fehlen, fehlschlägt die Aktualisierung fehl und Elastic Beanstalk gibt eine Meldung zurück, in der angegeben wird, welche Berechtigung erforderlich ist. Fügen Sie Berechtigungen für zusätzliche Services zur Standard-Servicerolle in der IAM-Konsole hinzu.

So fügen Sie zusätzliche Richtlinien zur Standard-Servicerolle hinzu

1. Öffnen Sie die Seite [Roles \(Rollen\)](#) in der IAM-Konsole.

2. Wählen Sie aws-elasticbeanstalk-service-role aus.
3. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Option Attach policies (Richtlinien anfügen) aus.
4. Wählen Sie die verwaltete Richtlinie für die zusätzlichen Services aus, die von der Anwendung verwendet werden. Zum Beispiel `AmazonAPIGatewayAdministrator` oder `AmazonElasticFileSystemFullAccess`.
5. Wählen Sie Attach policy aus.

Erstellen einer -Servicerolle

Wenn Sie die Standard-Servicerolle nicht verwenden können, erstellen Sie eine Servicerolle.

So erstellen Sie eine Servicerolle

1. Öffnen Sie die Seite [Roles \(Rollen\)](#) in der IAM-Konsole.
2. Wählen Sie Create role (Rolle erstellen) aus.
3. Wählen Sie unter AWS-Service die Option AWS Elastic Beanstalk aus. Wählen Sie dann Ihren Anwendungsfall aus.
4. Wählen Sie Next: Permissions aus.
5. Fügen Sie die verwalteten Richtlinien `AWSElasticBeanstalkService` und `AWSElasticBeanstalkEnhancedHealth` sowie weitere Richtlinien hinzu, um der Anwendung die erforderlichen Berechtigungen zu erteilen.
6. Wählen Sie Next: Tags (Weiter: Tags) aus.
7. (Optional) Fügen Sie der Rolle Tags hinzu.
8. Klicken Sie auf Weiter: Prüfen.
9. Geben Sie einen Namen für die Rolle ein.
10. Wählen Sie Create role aus.

Sie können die benutzerdefinierte Servicerolle anwenden, wenn Sie eine Umgebung im [Assistenten zum Erstellen einer Umgebung \(p. 437\)](#) oder mit der Option `--service-role` für den `eb create (p. 1053)`-Befehl erstellen.

Servicegebundene Rollen für Elastic Beanstalk verwenden

AWS Elastic Beanstalk verwendet [servicegebundene Rollen](#) von AWS Identity and Access Management (IAM). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Elastic Beanstalk verknüpft ist. Serviceverknüpfte Rollen werden von Elastic Beanstalk vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert.

Elastic Beanstalk definiert einige Arten von servicegebundenen Rollen:

- Servicegebundene Überwachungsrolle – Ermöglicht es Elastic Beanstalk, den Status laufender Umgebungen zu überwachen und Benachrichtigungen über statusrelevante Ereignisse zu veröffentlichen.
- Servicegebundene Wartungsrolle – Ermöglicht es Elastic Beanstalk, regelmäßige Wartungsarbeiten für Ihre laufenden Umgebungen durchzuführen.
- Servicegebundene Rolle für verwaltete Updates – Ermöglicht es Elastic Beanstalk, geplante Plattformaktualisierungen Ihrer laufenden Umgebungen durchzuführen.

Themen

- [Überwachen serviceverknüpfter Rollen \(p. 916\)](#)
- [Wartung serviceverknüpfter Rollen \(p. 919\)](#)
- [Serviceverknüpfte Rolle für verwaltete Updates \(p. 921\)](#)

Überwachen serviceverknüpfter Rollen

AWS Elastic Beanstalk verwendet [servicegebundene Rollen](#) von AWS Identity and Access Management (IAM). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Elastic Beanstalk verknüpft ist. Serviceverknüpfte Rollen werden von Elastic Beanstalk vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von Elastic Beanstalk, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Elastic Beanstalk definiert die Berechtigungen seiner servicegebundenen Rollen. Sofern nicht anders definiert, kann nur Elastic Beanstalk diese Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre zugehörigen AWS-Ressourcen gelöscht wurden. Dies schützt Ihre Elastic Beanstalk-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter [AWS-Services, die mit IAM funktionieren](#). Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen für servicegebundene Rollen für Elastic Beanstalk

Elastic Beanstalk verwendet die servicegebundene Rolle mit dem Namen AWSServiceRoleForElasticBeanstalk – Sie ermöglicht es Elastic Beanstalk, den Status laufender Umgebungen zu überwachen und Benachrichtigungen über statusrelevante Ereignisse zu veröffentlichen.

Die serviceverknüpfte Rolle AWSServiceRoleForAccessAnalyzer vertraut, dass die folgenden Services die Rolle übernehmen:

- [elasticbeanstalk.amazonaws.com](#)

Die Berechtigungsrichtlinie der servicegebundenen Rolle AWSServiceRoleForElasticBeanstalk enthält alle Berechtigungen, die Elastic Beanstalk benötigt, um Aktionen in Ihrem Namen durchzuführen:

```
{  
    "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",  
        "sns:GetQueueAttributes",  
        "sns:GetQueueUrl",  
        "sns:Publish"  
    ],  
    "Resource": [  
        "*"  
    ]  
}  
}
```

Sie müssen Berechtigungen konfigurieren, damit eine IAM-Entität (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen können. Weitere Informationen finden Sie unter [Serviceverknüpfte Rollenberechtigung](#) im IAM-Benutzerhandbuch.

Alternativ können Sie eine verwaltete AWS-Richtlinie verwenden, um [Vollzugriff auf Elastic Beanstalk zu gewähren](#) (p. 926).

Servicegebundene Rolle für Elastic Beanstalk erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine Elastic Beanstalk-Umgebung mit der Elastic Beanstalk API erstellen und keine Servicerolle angeben, erstellt Elastic Beanstalk die servicegebundene Rolle für Sie.

Important

Wenn Sie den Elastic Beanstalk-Service vor dem 27. September 2017 verwendet haben (dem Zeitpunkt der Einführung der Rolle AWSServiceRoleForElasticBeanstalk), und Ihr Konto die Rolle benötigte, erstellte Elastic Beanstalk die Rolle AWSServiceRoleForElasticBeanstalk in Ihrem Konto. Weitere Informationen finden Sie unter [In meinem IAM-Konto wird eine neue Rolle angezeigt](#).

Wenn Elastic Beanstalk versucht, beim Erstellen einer Umgebung die servicegebundene Rolle AWSServiceRoleForElasticBeanstalk für Ihr Konto zu erstellen, müssen Sie über die Berechtigung `iam:CreateServiceLinkedRole` verfügen. Wenn Sie nicht über diese Genehmigung verfügen, schlägt das Erstellen der Umgebung fehl und es wird eine Meldung angezeigt, die das Problem erklärt.

Alternativ kann ein anderer Benutzer, der über die Berechtigung zum Erstellen von serviceverknüpften Rollen verfügt, IAM verwenden, um die serviceverknüpfte Rolle vorab zu erstellen. Anschließend können Sie Ihre Umgebung auch ohne die `iam:CreateServiceLinkedRole`-Berechtigung selbst erstellen.

Sie (oder ein anderer Benutzer) können die IAM-Konsole verwenden, um eine servicegebundene Rolle mit dem Anwendungsfall Elastic Beanstalk zu erstellen. Erstellen Sie in der IAM CLI oder der IAM API eine serviceverknüpfte Rolle mit dem Servicenamen `elasticbeanstalk.amazonaws.com`. Weitere

Informationen finden Sie unter [Erstellen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch. Wenn Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie eine Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-API erstellen und keine Servicerolle angeben, erstellt Elastic Beanstalk die servicegebundene Rolle erneut für Sie.

Servicegebundene Rolle für Elastic Beanstalk bearbeiten

Elastic Beanstalk gestattet es Ihnen nicht, die servicegebundene Rolle `AWSServiceRoleForElasticBeanstalk` zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Servicegebundenen Rolle für Elastic Beanstalk bearbeiten

Wenn Sie eine Funktion oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie zunächst sicherstellen, dass alle Elastic Beanstalk-Umgebungen entweder eine andere Servicerolle verwenden oder beendet werden.

Note

Wenn der Elastic Beanstalk-Service die Rolle verwendet, wenn Sie versuchen, die Umgebungen zu beenden, tritt möglicherweise ein Fehler auf. Wenn das passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So terminieren Sie eine Elastic Beanstalk-Umgebung, die `AWSServiceRoleForElasticBeanstalk` verwendet (Konsole):

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Lesen Sie für Einzelheiten über das Terminieren einer Elastic Beanstalk-Umgebung mit der EB CLI [eb terminate \(p. 1094\)](#).

Details zum Beenden einer Elastic Beanstalk-Umgebung mithilfe der API erhalten Sie unter [TerminateEnvironment](#).

Manuelles Löschen der serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, die IAM-CLI oder die IAM-API, um die servicegebundene Rolle `AWSServiceRoleForElasticBeanstalk` zu löschen. Weitere Informationen finden Sie unter [Löschen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Unterstützte Regionen für servicegebundene Rollen für Elastic Beanstalk

Elastic Beanstalk unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter [AWS Elastic Beanstalk-Endpunkte und -Kontingente](#).

Wartung serviceverknüpfter Rollen

AWS Elastic Beanstalk verwendet [servicegebundene Rollen](#) von AWS Identity and Access Management (IAM). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Elastic Beanstalk verknüpft ist. Serviceverknüpfte Rollen werden von Elastic Beanstalk vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von Elastic Beanstalk, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Elastic Beanstalk definiert die Berechtigungen seiner servicegebundenen Rollen. Sofern nicht anders definiert, kann nur Elastic Beanstalk diese Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre zugehörigen AWS-Ressourcen gelöscht wurden. Dies schützt Ihre Elastic Beanstalk-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter [AWS-Services, die mit IAM funktionieren](#). Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen für servicegebundene Rollen für Elastic Beanstalk

Elastic Beanstalk verwendet die servicegebundene Rolle namens AWSServiceRoleForElasticBeanstalkMaintenance – Diese ermöglicht es Elastic Beanstalk, regelmäßige Wartungsarbeiten für Ihre laufenden Umgebungen durchzuführen.

Die servicegebundene Rolle AWSServiceRoleForElasticBeanstalkMaintenance vertraut bei der Übernahme dieser Rolle den folgenden Services:

- `maintenance.elasticbeanstalk.amazonaws.com`

Die Berechtigungsrichtlinie der servicegebundenen Rolle AWSServiceRoleForElasticBeanstalkMaintenance enthält alle Berechtigungen, die Elastic Beanstalk benötigt, um Aktionen in Ihrem Namen durchzuführen:

```
{  
    "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-*"  
            ]  
        }  
    ]  
}
```

```
}
```

Sie müssen Berechtigungen konfigurieren, damit eine IAM-Entität (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen können. Weitere Informationen finden Sie unter [Serviceverknüpfte Rollenberechtigung](#) im IAM-Benutzerhandbuch.

Alternativ können Sie eine verwaltete AWS-Richtlinie verwenden, um [Vollzugriff auf Elastic Beanstalk zu gewähren](#) (p. 926).

Servicegebundene Rolle für Elastic Beanstalk erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-API erstellen und kein Instance-Profil angeben, erstellt Elastic Beanstalk die servicegebundene Rolle für Sie.

Important

Diese serviceverknüpfte Rolle kann in Ihrem Konto erscheinen, wenn Sie eine Aktion in einem anderen Service abgeschlossen haben, der die von dieser Rolle unterstützten Funktionen verwendet. Wenn Sie Elastic Beanstalk Service vor dem 18. April 2019 genutzt haben (der Einführung der servicegebundenen Rolle AWSServiceRoleForElasticBeanstalkMaintenance), und Ihr Konto diese benötigte, hat Elastic Beanstalk die Rolle AWSServiceRoleForElasticBeanstalkMaintenance in Ihrem Konto erstellt. Weitere Informationen finden Sie unter [In meinem IAM-Konto wird eine neue Rolle angezeigt](#).

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie eine Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-API erstellen und kein Instance-Profil angeben, erstellt Elastic Beanstalk die servicegebundene Rolle erneut für Sie.

Servicegebundene Rolle für Elastic Beanstalk bearbeiten

Elastic Beanstalk gestattet Ihnen die Bearbeitung der servicegebundenen Rolle AWSServiceRoleForElasticBeanstalkMaintenance nicht. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Servicegebundenen Rolle für Elastic Beanstalk bearbeiten

Wenn Sie eine Funktion oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie IAM zum Löschen einer serviceverknüpften Rolle verwenden können, müssen Sie zuerst alle Elastic Beanstalk-Umgebungen beenden, die die Rolle verwenden.

Note

Wenn der Elastic Beanstalk-Service die Rolle verwendet, wenn Sie versuchen, die Umgebungen zu beenden, tritt möglicherweise ein Fehler auf. Wenn das passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So terminieren Sie eine Elastic Beanstalk-Umgebung, die die Rolle AWSServiceRoleForElasticBeanstalkMaintenance (Konsole) verwendet:

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.

2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).
4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Lesen Sie für Einzelheiten über das Terminieren einer Elastic Beanstalk-Umgebung mit der EB CLI [eb terminate \(p. 1094\)](#).

Details zum Beenden einer Elastic Beanstalk-Umgebung mithilfe der API erhalten Sie unter [TerminateEnvironment](#).

Manuelles Löschen der serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, die IAM-CLI oder die IAM-API, um die servicegebundene Rolle `AWSServiceRoleForElasticBeanstalkMaintenance` zu löschen. Weitere Informationen finden Sie unter [Löschen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Unterstützte Regionen für servicegebundene Rollen für Elastic Beanstalk

Elastic Beanstalk unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter [AWS Elastic Beanstalk-Endpunkte und -Kontingente](#).

Serviceverknüpfte Rolle für verwaltete Updates

AWS Elastic Beanstalk verwendet [servicegebundene Rollen](#) von AWS Identity and Access Management (IAM). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Elastic Beanstalk verknüpft ist. Serviceverknüpfte Rollen werden von Elastic Beanstalk vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von Elastic Beanstalk, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Elastic Beanstalk definiert die Berechtigungen seiner servicegebundenen Rollen. Sofern nicht anders definiert, kann nur Elastic Beanstalk diese Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre zugehörigen AWS-Ressourcen gelöscht wurden. Dies schützt Ihre Elastic Beanstalk-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter [AWS-Services, die mit IAM funktionieren](#). Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen für servicegebundene Rollen für Elastic Beanstalk

Elastic Beanstalk verwendet diese servicegebundene Rolle mit dem Namen `AWSServiceRoleForElasticBeanstalkManagedUpdates` – Sie ermöglicht es Elastic Beanstalk, geplante Plattformaktualisierungen Ihrer laufenden Umgebungen durchzuführen.

Die servicegebundene Rolle `AWSServiceRoleForElasticBeanstalkManagedUpdates` vertraut den folgenden Services, die diese Rolle übernehmen:

- managedupdates.elasticbeanstalk.amazonaws.com

Die Berechtigungsrichtlinie der servicegebundenen Rolle

AWSServiceRoleForElasticBeanstalkManagedUpdates enthält alle Berechtigungen, die Elastic Beanstalk benötigt, um verwaltete Aktualisierungsaktionen in Ihrem Namen durchzuführen:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPassRoleToElasticBeanstalkAndDownstreamServices",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringLikeIfExists": {
                    "iam:PassedToService": [
                        "elasticbeanstalk.amazonaws.com",
                        "ec2.amazonaws.com",
                        "autoscaling.amazonaws.com",
                        "elasticloadbalancing.amazonaws.com",
                        "ecs.amazonaws.com",
                        "cloudformation.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "SingleInstanceAPIs",
            "Effect": "Allow",
            "Action": [
                "ec2:releaseAddress",
                "ec2:allocateAddress",
                "ec2:DisassociateAddress",
                "ec2:AssociateAddress"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ECS",
            "Effect": "Allow",
            "Action": [
                "ecs:RegisterTaskDefinition",
                "ecs:DeRegisterTaskDefinition",
                "ecs>List*",
                "ecs:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ElasticBeanstalkAPIs",
            "Effect": "Allow",
            "Action": [
                "elasticbeanstalk:)"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ReadOnlyAPIs",
            "Effect": "Allow",
            "Action": [
                "cloudformation:Describe*",
                "cloudformation>List*",
                "ec2:Describe*",
                "cloudformation:List*"
            ]
        }
    ]
}
```

```

        "autoscaling:Describe",
        "elasticloadbalancing:Describe"
    ],
    "Resource": "*"
},
{
    "Sid": "ASG",
    "Effect": "Allow",
    "Action": [
        "autoscaling:AttachInstances",
        "autoscaling>CreateAutoScalingGroup",
        "autoscaling>CreateLaunchConfiguration",
        "autoscaling>DeleteAutoScalingGroup",
        "autoscaling>DeleteLaunchConfiguration",
        "autoscaling>DeleteScheduledAction",
        "autoscaling:DetachInstances",
        "autoscaling:PutNotificationConfiguration",
        "autoscaling:PutScalingPolicy",
        "autoscaling:PutScheduledUpdateGroupAction",
        "autoscaling:ResumeProcesses",
        "autoscaling:SuspendProcesses",
        "autoscaling:TerminateInstanceInAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup"
    ],
    "Resource": [
        "arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/
awseb-e-*",
        "arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/awseb-e-*"
    ]
},
{
    "Sid": "CFN",
    "Effect": "Allow",
    "Action": [
        "cloudformation>CreateStack",
        "cloudformation>DeleteStack",
        "cloudformation>GetTemplate",
        "cloudformation>UpdateStack"
    ],
    "Resource": "arn:aws:cloudformation:*:*:stack/awseb-e-*"
},
{
    "Sid": "EC2",
    "Effect": "Allow",
    "Action": [
        "ec2:TerminateInstances"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
        "StringLike": {
            "ec2:ResourceTag/aws:cloudformation:stack-id":
"arn:aws:cloudformation:*:*:stack/awseb-e-*"
        }
    }
},
{
    "Sid": "S3Obj",
    "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": "S3Bucket",
    "Effect": "Allow",
    "Action": [
        "s3:GetBucketLocation",
        "s3:GetBucketPolicy",
        "s3>ListBucket",
        "s3:PutBucketPolicy"
    ],
    "Resource": "arn:aws:s3:::elasticbeanstalk-*"
},
{
    "Sid": "CWL",
    "Effect": "Allow",
    "Action": [
        "logs>CreateLogGroup",
        "logs>DeleteLogGroup",
        "logs:PutRetentionPolicy"
    ],
    "Resource": "arn:aws:logs:**:log-group:/aws/elasticbeanstalk/*"
},
{
    "Sid": "ELB",
    "Effect": "Allow",
    "Action": [
        "elasticloadbalancing:RegisterTargets",
        "elasticloadbalancing:DeRegisterTargets",
        "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
        "elasticloadbalancing:RegisterInstancesWithLoadBalancer"
    ],
    "Resource": [
        "arn:aws:elasticloadbalancing:**:targetgroup/awseb-*",
        "arn:aws:elasticloadbalancing:**:loadbalancer/awseb-e-*"
    ]
}
]
```

Sie müssen Berechtigungen konfigurieren, damit eine IAM-Entität (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen können. Weitere Informationen finden Sie unter [Serviceverknüpfte Rollenberechtigung](#) im IAM-Benutzerhandbuch.

Alternativ können Sie eine verwaltete AWS-Richtlinie verwenden, um [Vollzugriff auf Elastic Beanstalk zu gewähren](#) (p. 926).

Servicegebundene Rolle für Elastic Beanstalk erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-API erstellen, verwaltete Aktualisierungen aktivieren und `AWSServiceRoleForElasticBeanstalkManagedUpdates` als Wert für die `ServiceRoleForManagedUpdates`-Option des [aws:elasticbeanstalk:managedactions](#) (p. 693)-Namespaces angeben, erstellt Elastic Beanstalk die servicegebundene Rolle für Sie.

Wenn Elastic Beanstalk versucht, die servicegebundene Rolle `AWSServiceRoleForElasticBeanstalkManagedUpdates` für Ihr Konto zu erstellen, wenn Sie eine Umgebung erstellen, müssen Sie die Berechtigung `iam>CreateServiceLinkedRole` haben. Wenn Sie nicht über diese Genehmigung verfügen, schlägt das Erstellen der Umgebung fehl und es wird eine Meldung angezeigt, die das Problem erklärt.

Alternativ kann ein anderer Benutzer, der über die Berechtigung zum Erstellen von serviceverknüpften Rollen verfügt, IAM verwenden, um die serviceverknüpfte Rolle vorab zu erstellen. Anschließend können Sie Ihre Umgebung auch ohne die `iam:CreateServiceLinkedRole`-Berechtigung selbst erstellen.

Sie (oder ein anderer Benutzer) können die IAM-Konsole verwenden, um eine servicegebundene Rolle mit dem Anwendungsfall Elastic Beanstalk Managed Updates zu erstellen. Erstellen Sie in der IAM CLI oder der IAM API eine serviceverknüpfte Rolle mit dem Servicenamen `managedupdates.elasticbeanstalk.amazonaws.com`. Weitere Informationen finden Sie unter [Erstellen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch. Wenn Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie eine Elastic Beanstalk-Umgebung mit der Elastic Beanstalk-API erstellen, verwaltete Updates aktivieren und `AWSServiceRoleForElasticBeanstalkManagedUpdates` als Wert für die `ServiceRoleForManagedUpdates`-Option des [aws:elasticbeanstalk:managedactions \(p. 693\)](#)-Namespaces angeben, erstellt Elastic Beanstalk die servicegebundene Rolle erneut für Sie.

Servicegebundene Rolle für Elastic Beanstalk bearbeiten

Elastic Beanstalk gestattet es Ihnen nicht, die servicegebundene Rolle `AWSServiceRoleForElasticBeanstalkManagedUpdates` zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Servicegebundenen Rolle für Elastic Beanstalk bearbeiten

Wenn Sie eine Funktion oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie zunächst sicherstellen, dass Elastic Beanstalk-Umgebungen mit aktivierte verwalteten Updates entweder eine andere Servicerolle verwenden oder beendet werden.

Note

Wenn der Elastic Beanstalk-Service die Rolle verwendet, wenn Sie versuchen, die Umgebungen zu beenden, tritt möglicherweise ein Fehler auf. Wenn das passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So terminieren Sie eine Elastic Beanstalk-Umgebung, die die Rolle `AWSServiceRoleForElasticBeanstalkManagedUpdates` verwendet (Konsole):

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#), und wählen Sie in der Liste Regionen Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Klicken Sie auf Actions (Aktionen) und dann auf Terminate Environment (Umgebung beenden).

4. Verwenden Sie das Dialogfeld auf dem Bildschirm, um die Beendigung der Umgebung zu bestätigen.

Lesen Sie für Einzelheiten über das Terminieren einer Elastic Beanstalk-Umgebung mit der EB CLI [eb terminate \(p. 1094\)](#).

Details zum Beenden einer Elastic Beanstalk-Umgebung mithilfe der API erhalten Sie unter [TerminateEnvironment](#).

Manuelles Löschen der serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, die IAM-CLI oder die IAM-API, um die servicegebundene Rolle AWSServiceRoleForElasticBeanstalkManagedUpdates zu löschen. Weitere Informationen finden Sie unter [Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch](#).

Unterstützte Regionen für servicegebundene Rollen für Elastic Beanstalk

Elastic Beanstalk unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter [AWS Elastic Beanstalk-Endpunkte und -Kontingente](#).

Elastic Beanstalk-Benutzerrichtlinien verwalten

AWS Elastic Beanstalk bietet zwei verwaltete Richtlinien, mit denen Sie allen Elastic Beanstalk-Ressourcen Vollzugriff oder Nur-Lese-Zugriff zuweisen können. Sie können die Richtlinien an AWS Identity and Access Management (IAM)-Benutzer oder -Gruppen anhängen.

Verwaltete Benutzerrichtlinien

- **AWSElasticBeanstalkFullAccess** - Benutzer können Elastic Beanstalk-Anwendungen sowie Anwendungsversionen, Konfigurationseinstellungen, Umgebungen und zugrunde liegende Ressourcen erstellen, ändern und löschen.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "elasticbeanstalk:*",  
                "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": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "iam>AddRoleToInstanceProfile",
        "iam>CreateInstanceProfile",
        "iam>CreateRole"
    ],
    "Resource": [
        "arn:aws:iam::*:role/aws-elasticbeanstalk*",
        "arn:aws:iam::*:instance-profile/aws-elasticbeanstalk*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "iam>CreateServiceLinkedRole"
    ],
    "Resource": [
        "arn:aws:iam::*:role/aws-service-role/autoscaling.amazonaws.com/
AWSServiceRoleForAutoScaling"
    ],
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": "autoscaling.amazonaws.com"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iam>CreateServiceLinkedRole"
    ],
    "Resource": [
        "arn:aws:iam::*:role/aws-service-role/elasticbeanstalk.amazonaws.com/
AWSServiceRoleForElasticBeanstalk"
    ],
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": "elasticbeanstalk.amazonaws.com"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iam>AttachRolePolicy"
    ],
    "Resource": "*",
    "Condition": {
        "StringLike": {
            "iam:PolicyArn": [
                "arn:aws:iam::aws:policy/AWSElasticBeanstalk*",
                "arn:aws:iam::aws:policy/service-role/AWSElasticBeanstalk*"
            ]
        }
    }
}
```

```
    ]  
}
```

- AWSElasticBeanstalkReadOnlyAccess – Benutzer können Anwendungen und Umgebungen anzeigen, aber keine Vorgänge für diese ausführen. Sie ermöglicht den Nur-Lese-Zugriff auf alle Elastic Beanstalk-Ressourcen. Beachten Sie, dass mit schreibgeschütztem Zugriff keine Aktionen – wie z. B. Elastic Beanstalk-Protokolle zum Lesen herunterladen – möglich sind. Das liegt daran, dass die Protokolle im Amazon S3-Bucket bereitgestellt werden, wo Elastic Beanstalk eine Schreibberechtigung erfordern würde. Weitere Informationen zum Einrichten des Zugriffs auf Elastic Beanstalk-Protokolle finden Sie im Beispiel am Ende dieses Themas.

```
{  
  "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*",  
        "cloudformation:Describe*",  
        "cloudformation:Get*",  
        "cloudformation>List*",  
        "cloudformation:Validate*",  
        "cloudformation:Estimate*",  
        "rds:Describe*",  
        "sns:Get*",  
        "sns>List*"  
      ],  
      "Resource": "*"  
    }  
  ]  
}
```

Steuern des Zugriffs mit verwalteten Richtlinien

Mit verwalteten Richtlinien können Sie Vollzugriff oder schreibgeschützten Zugriff auf Elastic Beanstalk gewähren. Diese Richtlinien werden von Elastic Beanstalk automatisch aktualisiert, wenn weitere Berechtigungen für den Zugriff auf neue Funktionen benötigt werden.

So wenden Sie eine verwaltete Richtlinie auf IAM-Benutzer oder -Gruppen an

1. Öffnen Sie die [Seite Policies \(Richtlinien\)](#) in der IAM-Konsole.
2. Geben Sie im Suchfeld **AWSElasticBeanstalk** ein, um die Richtlinien zu filtern.
3. Wählen Sie in der Liste der Richtlinien das Kontrollkästchen neben AWSElasticBeanstalkReadOnlyAccess oder AWSElasticBeanstalkFullAccess aus.
4. Klicken Sie auf Policy actions und anschließend auf Attach.

5. Wählen Sie einen oder mehrere Benutzer und Gruppen aus, denen Sie die Richtlinie anfügen möchten. Über das Menü Filter und das Suchfeld können Sie die Liste der Prinzipal-Entitäten filtern.
6. Wählen Sie Attach policy aus.

Erstellen einer benutzerdefinierten Benutzerrichtlinie

Sie können Ihre eigene IAM-Richtlinie erstellen, um bestimmte Elastic Beanstalk-API-Aktionen für bestimmte Elastic Beanstalk-Ressourcen zuzulassen oder zu verweigern. Weitere Informationen zum Anfügen einer Richtlinie zu einem Benutzer oder einer Gruppe finden Sie im Artikel zum [Arbeiten mit Richtlinien](#) im Thema zum Verwenden von AWS Identity and Access Management.

Note

Sie können einschränken, wie Benutzer mit Elastic Beanstalk-APIs interagieren können. Jedoch gibt es derzeit keine effektive Möglichkeit, mit der Sie Benutzer mit Berechtigungen zum Erstellen der benötigten zugrunde liegenden Ressourcen daran hindern können, weitere Ressourcen in Amazon EC2 und anderen Services zu erstellen.

Betrachten Sie diese Richtlinien als ein effektives Mittel zur Verteilung von Elastic Beanstalk-Verantwortlichkeiten, nicht als ein Mittel zur Sicherung aller zugrundeliegenden Ressourcen.

Im November 2019 führte Elastic Beanstalk die Unterstützung für [Amazon EC2-Startvorlagen](#) ein. Dies ist ein neuer Ressourcentyp, den die Auto Scaling-Gruppe Ihrer Umgebung zum Starten von Amazon EC2-Instances verwenden kann, und er erfordert neue Berechtigungen. Die meisten Kunden sollten davon nicht betroffen sein, da Umgebungen weiterhin die Legacy-Ressource verwenden und Konfigurationen starten können, auch wenn Ihre Benutzerrichtlinie hierfür nicht die erforderlichen Berechtigungen besitzt. Bei dem Versuch, eine neue Funktion, für die Amazon EC2-Startvorlagen benötigt werden, mit einer benutzerdefinierten Richtlinie zu verwenden, kann die Erstellung oder Aktualisierung Ihrer Umgebung fehlschlagen. Stellen Sie in diesem Fall sicher, dass Ihre benutzerdefinierte Richtlinie über die folgenden Berechtigungen verfügt.

Erforderliche Berechtigungen für Amazon EC2-Startvorlagen

- `EC2:CreateLaunchTemplate`
- `EC2:CreateLaunchTemplateVersions`
- `EC2>DeleteLaunchTemplate`
- `EC2>DeleteLaunchTemplateVersions`
- `EC2:DescribeLaunchTemplate`
- `EC2:DescribeLaunchTemplateVersions`

Eine IAM-Richtlinie enthält Richtlinienanweisungen, mit denen die zu erteilenden Berechtigungen beschrieben werden. Um eine Richtlinienanweisung für Elastic Beanstalk zu erstellen, müssen Sie wissen, wie die folgenden vier Elemente einer Richtlinienanweisung verwendet werden:

- Effect gibt an, ob die Aktionen in der Anweisung zugelassen oder verweigert werden.
- Action gibt die zu steuernden [API-Operationen](#) an. Beispielsweise können Sie mit `elasticbeanstalk>CreateEnvironment` den Vorgang `createEnvironment` angeben. Bestimmte Vorgänge, wie z. B. das Erstellen einer Umgebung, erfordern zusätzliche Berechtigungen für die Aktionsausführung. Weitere Informationen finden Sie unter [Ressourcen und Bedingungen für Elastic Beanstalk-Aktionen \(p. 934\)](#).

Note

Geben Sie zur Verwendung der API-Operation `UpdateTagsForResource` anstelle des Namens der API-Operation eine (oder beide) der folgenden beiden virtuellen Aktionen an:

elasticbeanstalk:AddTags

Steuert die Berechtigung zum Aufrufen von `UpdateTagsForResource` und übergeben einer Liste mit Tags, die dem Parameter `TagsToAdd` hinzugefügt werden.

elasticbeanstalk:RemoveTags

Steuert die Berechtigung zum Aufrufen von `UpdateTagsForResource` und übergeben einer Liste mit Tag-Schlüsseln, die aus dem Parameter `TagsToRemove` entfernt werden.

- Resource gibt die Ressourcen an, bei denen der Zugriff gesteuert werden soll. Um Elastic Beanstalk-Ressourcen anzugeben, führen Sie die [Amazon-Ressourcenname \(p. 933\)](#) (ARN) jeder Ressource auf.
- (Optional) Condition gibt Einschränkungen für die in der Anweisung erteilten Berechtigungen an. Weitere Informationen finden Sie unter [Ressourcen und Bedingungen für Elastic Beanstalk-Aktionen \(p. 934\)](#).

Die folgenden Abschnitte zeigen einige Fälle, in denen Sie eine benutzerdefinierte Benutzerrichtlinie in Betracht ziehen könnten.

Begrenzte Erstellung einer Elastic Beanstalk-Umgebung ermöglichen

Die Richtlinie im folgenden Beispiel ermöglicht einem Benutzer, die `CreateEnvironment`-Aktion aufzurufen, um eine Umgebung zu erstellen, deren Name mit **Test** beginnt, mit der angegebenen Anwendung und Anwendungsversion.

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

Die obige Richtlinie zeigt, wie man beschränkten Zugang zu Elastic Beanstalk-Operationen gewährt. Um eine Umgebung tatsächlich starten zu können, muss der Benutzer auch über die Berechtigungen zum Erstellen der in der Umgebung benötigten AWS-Ressourcen verfügen. Beispielsweise wird mit der folgenden Richtlinie Zugriff auf den Standardsatz der Ressourcen für eine Webserverumgebung gewährt:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:*",  
                "ecs:*",  
                "elasticloadbalancing:*",  
                "autoscaling:*",  
                "cloudwatch:",  
                "s3:*",  
                "sns:*",  
                "cloudformation:*",  
                "sqs:*"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Zugriff auf Elastic Beanstalk-Protokolle in Amazon S3 ermöglichen

Die Richtlinie im folgenden Beispiel ermöglicht es einem Benutzer, Elastic Beanstalk-Protokolle abzurufen, sie in Amazon S3 bereitzustellen und abzurufen.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "s3:DeleteObject",  
                "s3:GetObjectAcl",  
                "s3:PutObjectAcl"  
            ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:s3:::elasticbeanstalk-*"  
        }  
    ]  
}
```

Note

Um diese Berechtigungen nur auf die Protokollpfade einzuschränken, verwenden Sie das folgende Ressourcenformat.

```
"arn:aws:s3:::elasticbeanstalk-us-east-2-123456789012/resources/environments/logs/  
*"
```

Verwaltung einer bestimmten Elastic Beanstalk-Anwendung ermöglichen

Die Richtlinie im folgenden Beispiel ermöglicht einem Benutzer das Verwalten von Umgebungen und anderen Ressourcen innerhalb einer bestimmten Elastic Beanstalk-Anwendung. Die Richtlinie verweigert Elastic Beanstalk-Aktionen auf Ressourcen anderer Anwendungen und verweigert auch die Erstellung und Löschung von Elastic Beanstalk-Anwendungen.

Note

Die Richtlinie verweigert nicht den Zugriff auf Ressourcen durch andere Services. Es zeigt eine effektive Möglichkeit zum Verteilen von Verantwortlichkeiten für die Verwaltung von Elastic Beanstalk-Anwendungen über verschiedene Benutzer, nicht als Möglichkeit zum Sichern der zugrundeliegenden Ressourcen.

```
{  
    "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"  
                ]  
            }  
        }  
    ]  
}
```

Amazon-Ressourcennamen-Format für Elastic Beanstalk

Sie geben eine Ressource für eine IAM-Richtlinie an, wobei Sie den Amazon-Ressourcename (ARN) der Ressource verwenden. Für Elastic Beanstalk hat der ARN das folgende Format.

```
arn:aws:elasticbeanstalk:region:account-id:resource-type/resource-path
```

Wobei gilt:

- *region* ist die Region, in der sich die Ressource befindet (z. B. **us-west-2**).
- *account-id* ist die AWS-Konto-ID ohne Bindestriche (zum Beispiel **123456789012**).
- *resource-type* identifiziert den Typ der Elastic Beanstalk-Ressource, z. B. `environment`. In der nachstehenden Tabelle finden Sie eine Liste aller Ressourcentypen von Elastic Beanstalk.
- *resource-path* ist der Teil, der die jeweilige Ressource über den Namen identifiziert. Eine Ressource vom Typ Elastic Beanstalk hat einen Pfad, der diese Ressource eindeutig identifiziert. In der folgenden Tabelle finden Sie das Format des Ressourcenpfads für jeden Ressourcentyp. Eine Umgebung ist beispielsweise immer einer Anwendung zugeordnet. Der Ressourcenpfad für die Umgebung **myEnvironment** in der Anwendung **myApp** würde wie folgt aussehen:

```
myApp/myEnvironment
```

Elastic Beanstalk verfügt über mehrere Arten von Ressourcen, die Sie in einer Richtlinie angeben können. In der folgenden Tabelle sind die jeweiligen ARN-Formate für jeden Ressourcentyp sowie ein Beispiel aufgeführt.

Ressourcentyp	Format für ARN
application	<pre>arn:aws:elasticbeanstalk:<i>region</i>:<i>account-id</i>:application/<i>application-name</i></pre> <p>Beispiel: arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App</p>
applicationversion	<pre>arn:aws:elasticbeanstalk:<i>region</i>:<i>account-id</i>:applicationversion/<i>application-name</i>/<i>version-label</i></pre> <p>Beispiel: arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version</p>
configurationtemplate	<pre>arn:aws:elasticbeanstalk:<i>region</i>:<i>account-id</i>:configurationtemplate/<i>application-name</i>/<i>template-name</i></pre> <p>Beispiel: arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template</p>
environment	<pre>arn:aws:elasticbeanstalk:<i>region</i>:<i>account-id</i>:environment/<i>application-name</i>/<i>environment-name</i></pre> <p>Beispiel: arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/MyEnvironment</p>
platform	<pre>arn:aws:elasticbeanstalk:<i>region</i>:<i>account-id</i>:platform/<i>platform-name</i>/<i>platform-version</i></pre>

Ressourcentyp	Format für ARN
	Beispiel: arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/MyPlatform/1.0
solutionstackarn	arn:aws:elasticbeanstalk:<i>region</i>::solutionstack/<i>solutionstack-name</i> Beispiel: arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7

Eine Umgebung, Anwendungsversion und eine Konfigurationsvorlage sich immer in einer bestimmten Anwendung vorhanden. Sie werden feststellen, dass diese Ressourcen alle über einen Anwendungsnamen in ihrem Ressourcenpfad verfügen, damit sie eindeutig über ihren Ressourcennamen und die Anwendung, die sie enthält, identifiziert werden können. Obwohl Lösungs-Stacks von Konfigurationsvorlagen und Umgebungen verwendet werden, sind diese nicht spezifisch für eine Anwendung oder ein AWS-Konto und enthalten die Anwendung oder der AWS-Konto nicht in ihren ARNs.

Ressourcen und Bedingungen für Elastic Beanstalk-Aktionen

In diesem Abschnitt werden die Ressourcen und Bedingungen beschrieben, die Sie in Richtlinienanweisungen zum Erteilen von Berechtigungen verwenden können, mit denen spezifische Elastic Beanstalk-Aktionen auf bestimmten Elastic Beanstalk-Ressourcen ausgeführt werden können.

Mit Bedingungen können Sie Berechtigungen für Ressourcen festlegen, die die Aktion ausführen muss. Wenn Sie beispielsweise die `CreateEnvironment`-Aktion aufrufen, müssen Sie auch die bereitzustellende Anwendungsversion und die Anwendung angeben, die den Anwendungsnamen enthält. Wenn Sie Berechtigungen für die `CreateEnvironment`-Aktion festlegen, geben Sie die Anwendung und die Anwendungsversion an, auf die die Aktion einwirken soll, indem Sie die `InApplication`- und `FromApplicationVersion`-Bedingungen verwenden.

Außerdem können Sie die Umgebungskonfiguration mit einem Lösungs-Stack (`FromSolutionStack`) oder einer Konfigurationsvorlage (`FromConfigurationTemplate`) verwenden. Mit der folgenden Richtlinienanweisung kann die `CreateEnvironment`-Aktion eine Umgebung mit Namen `myenv` (angegeben durch `Resource`) in der Anwendung `My App` (durch die `InApplication`-Bedingung angegeben) erstellen. Dabei wird die Anwendungsversion `My Version` (`FromApplicationVersion`) mit einer `32bit Amazon Linux running Tomcat 7`-Konfiguration (`FromSolutionStack`) verwendet:

```
{
  "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

Die meisten Bedingungsschlüsse in diesem Thema gelten speziell für Elastic Beanstalk und ihre Namen enthalten das Präfix `elasticbeanstalk:`. Aus Gründen der Übersichtlichkeit wird dieses Präfix der Bedingungsschlüsselnamen in den folgenden Abschnitten ausgelassen. Wir geben also zum Beispiel `InApplication` anstelle des vollständigen Namens `elasticbeanstalk:InApplication` an.

Wir nennen aber einige Bedingungsschlüsse, die in allen AWS-Services verwendet werden, und fügen deren Präfix `aws:` hinzu, um diese Ausnahmen hervorzuheben.

In Richtlinienbeispielen werden die stets die vollständigen Bedingungsschlüsse mit Präfix angezeigt.

Abschnitte

- [Richtlinieninformationen für Elastic Beanstalk-Aktionen \(p. 935\)](#)
- [Bedingungsschlüsse für Elastic Beanstalk-Aktionen \(p. 957\)](#)

Richtlinieninformationen für Elastic Beanstalk-Aktionen

In der folgenden Tabelle werden alle Elastic Beanstalk-Aktionen, die Ressource, auf die jede Aktion einwirkt, und zusätzliche Kontextinformationen aufgelistet, die mithilfe von Bedingungen bereitgestellt werden können.

Richtlinieninformationen für Elastic Beanstalk-Aktionen, einschließlich Ressourcen, Bedingungen, Beispiele und Abhängigkeiten

Ressource	Bedingungen	Beispielanweisung
Aktion: AbortEnvironmentUpdate		
application environment	<code>aws:ResourceTag/ key-name</code> (Optional) <code>aws:TagKeys</code> (Optional)	Mit der folgenden Richtlinie können Benutzer Vorgänge zur Aktualisierung einer Umgebung auf Umgebungen in einer Anwendung namens <code>My App</code> abbrechen. <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:AbortEnvironmentUpdate"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] }] }</pre>
Aktion: CheckDNSAvailability		
"*"	-	{

Ressource	Bedingungen	Beispielanweisung
		<pre>"Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CheckDNSAvailability"], "Effect": "Allow", "Resource": "*" }]</pre>
Aktion: ComposeEnvironments		
application	aws:ResourceTag/ <i>key-name</i> (Optional)	Die folgende Richtlinie erlaubt einem Benutzer, Umgebungen aufzubauen, die zu einer Anwendung mit dem Namen My App gehören.
	aws:TagKeys (Optional)	<pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:ComposeEnvironments"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App"] }] }</pre>
Aktion: CreateApplication		
application	aws:RequestTag/ <i>key-name</i> (Optional)	In diesem Beispiel kann die CreateApplication-Aktion Anwendungen erstellen, deren Namen mit DivA beginnen:
	aws:TagKeys (Optional)	<pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk>CreateApplication"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/DivA*"] }] }</pre>
Aktion: CreateApplicationVersion		

Ressource	Bedingungen	Beispielanweisung
applicationversion	<pre>inApplication aws:RequestTag/key-name (Optional) aws:TagKeys (Optional)</pre>	<p>In diesem Beispiel kann die <code>CreateApplicationVersion</code>-Aktion Anwendungsversionen mit einem beliebigen Namen (*) in der Anwendung My App erstellen:</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CreateApplicationVersion"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/*"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } }] } }</pre>

Aktion: [CreateConfigurationTemplate](#)

Ressource	Bedingungen	Beispielanweisung
configurationtemplate	<pre> IsApplication FromApplication FromApplicationVersion FromConfigurationTemplate { FromEnvironment FromSolutionStack aws:RequestTag/ key-name (Optional) aws:TagKeys (Optional) </pre>	<p>Mit der folgenden Richtlinie kann die <code>CreateConfigurationTemplate</code>-Aktion Konfigurationsvorlagen, deren Name mit My Template (My Template*) beginnt, in der Anwendung My App erstellen:</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>
Aktion: CreateEnvironment		

Ressource	Bedingungen	Beispielanweisung
environment	<p>InApplication FromApplicationVersion FromConfigurationTemplate FromSolutionStack aws:RequestTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)</p>	<p>Die folgende Richtlinie erlaubt der CreateEnvironment-Aktion die Erstellung einer Umgebung mit dem Namen myenv in der Anwendung My App sowie die Verwendung des Lösungs-Stacks 32bit Amazon Linux running Tomcat 7.</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>

Aktion: [CreatePlatformVersion](#)

Ressource	Bedingungen	Beispielanweisung
platform	<p>aws:RequestTag/ key-name (Optional)</p> <p>aws:TagKeys (Optional)</p>	<p>In diesem Beispiel darf die CreatePlatformVersion-Aktion Plattformversionen erstellen, die auf die us-east-2-Region ausgerichtet sind und deren Name mit us-east-2_ beginnt:</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>
Aktion: CreateStorageLocation		
"*"	-	<pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk>CreateStorageLocation"], "Effect": "Allow", "Resource": "*" }] }</pre>
Aktion: DeleteApplication		
application	<p>aws:ResourceTag/ key-name (Optional)</p> <p>aws:TagKeys (Optional)</p>	<p>Mit dieser Richtlinie kann die DeleteApplication-Aktion die Anwendung My App löschen:</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk>DeleteApplication"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us- east-2:123456789012:application/My App"] }] }</pre>
Aktion: DeleteApplicationVersion		

Ressource	Bedingungen	Beispielanweisung
applicationversion	inApplication aws:ResourceTag/ key-name (Optional) aws:TagKeys (Optional)	Mit der folgenden Richtlinie kann die DeleteApplicationVersion-Aktion eine Anwendungsversion, deren Name My Version lautet, in der Anwendung My App löschen: <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:DeleteApplicationVersion"], "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"] } }] } }</pre>
Aktion: DeleteConfigurationTemplate		
configurationtemplate	inApplication (Optional) aws:ResourceTag/ key-name (Optional) aws:TagKeys (Optional)	Mit der folgenden Richtlinie kann die DeleteConfigurationTemplate-Aktion eine Konfigurationsvorlage, deren Name My Template lautet, in der Anwendung My App löschen. Die Angabe des Anwendungsnamens als Bedingung ist optional. <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>
Aktion: DeleteEnvironmentConfiguration		

Ressource	Bedingungen	Beispielanweisung
environment	InApplication (Optional)	<p>Die folgende Richtlinie ermöglicht der DeleteEnvironmentConfiguration-Aktion das Löschen eines Konfigurationsentwurfs für die Umgebung myenv in der Anwendung My App. Die Angabe des Anwendungsnamens als Bedingung ist optional.</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>
Aktion: DeletePlatformVersion		
platform	aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>Mit der folgenden Richtlinie kann die DeletePlatformVersion-Aktion Plattformversionen löschen, die auf die us-east-2-Region ausgerichtet sind und deren Name mit us-east-2_ beginnt:</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>
Aktion: DescribeApplications		

Ressource	Bedingungen	Beispielanweisung
application	<p><code>aws:ResourceTag/ <i>key-name</i></code> (Optional)</p> <p><code>aws:TagKeys</code> (Optional)</p>	<p>Mit dieser Richtlinie kann die <code>DescribeApplications</code>-Aktion die Anwendung "My App" beschreiben.</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:DescribeApplications"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] }] }</pre>
Aktion: DescribeApplicationVersions		
applicationversion	<p><code>ArnApplication</code> (Optional)</p> <p><code>aws:ResourceTag/ <i>key-name</i></code> (Optional)</p> <p><code>aws:TagKeys</code> (Optional)</p>	<p>Die folgende Richtlinie erlaubt der <code>DescribeApplicationVersions</code>-Aktion, die Anwendungsversion My Version in der Anwendung My App zu beschreiben. Die Angabe des Anwendungsnamens als Bedingung ist optional.</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:DescribeApplicationVersions"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"] }] }</pre>
Aktion: DescribeConfigurationOptions		

Ressource	Bedingungen	Beispielanweisung
environment configurationtemplate solutionstack	InApplication (Optional) aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>Die folgende Richtlinie ermöglicht der <code>DescribeConfigurationOptions</code>-Aktion das Beschreiben der Konfigurationsoptionen für die Umgebung myenv in der Anwendung My App. Die Angabe des Anwendungsnamens als Bedingung ist optional.</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>
Aktion: DescribeConfigurationSettings		
environment configurationtemplate	InApplication (Optional) aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>Die folgende Richtlinie ermöglicht der <code>DescribeConfigurationSettings</code>-Aktion das Beschreiben der Konfigurationseinstellungen für die Umgebung myenv in der Anwendung My App. Die Angabe des Anwendungsnamens als Bedingung ist optional.</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>
Aktion: DescribeEnvironmentHealth		

Ressource	Bedingungen	Beispielanweisung
environment	<p><code>aws:ResourceTag/ <i>key-name</i></code> (Optional)</p> <p><code>aws:TagKeys</code> (Optional)</p>	<p>Mit der folgenden Richtlinie wird die Verwendung von <code>DescribeEnvironmentHealth</code> zum Abrufen von Zustandsinformationen für eine Umgebung namens <code>myenv</code> zugelassen.</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>
Aktion: DescribeEnvironmentResources		
environment	<p><code>InApplication</code> (Optional)</p> <p><code>aws:ResourceTag/ <i>key-name</i></code> (Optional)</p> <p><code>aws:TagKeys</code> (Optional)</p>	<p>Die folgende Richtlinie ermöglicht der <code>DescribeEnvironmentResources</code>-Aktion das Zurückgeben einer Liste mit AWS-Ressourcen für die Umgebung <code>myenv</code> in der Anwendung <code>My App</code>. Die Angabe des Anwendungsnamens als Bedingung ist optional.</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": "elasticbeanstalk:DescribeEnvironmentResources", "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us- east-2:123456789012:environment/My App/myenv"] }] }</pre>
Aktion: DescribeEnvironments		

Ressource	Bedingungen	Beispielanweisung
environment	InApplication (Optional) aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	Die folgende Richtlinie erlaubt der <code>DescribeEnvironments</code> -Aktion, die Umgebungen <code>myenv</code> und <code>myotherenv</code> in der Anwendung <code>My App</code> zu beschreiben. Die Angabe des Anwendungsnamens als Bedingung ist optional. <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>
Aktion: <code>DescribeEvents</code>		
application	InApplication	Die folgende Richtlinie ermöglicht der <code>DescribeEvents</code> -Aktion das Auflisten von Ereignisbeschreibungen für die Umgebung <code>myenv</code> und die Anwendungsversion <code>My Version</code> in der Anwendung <code>My App</code> .
applicationversion configuration environment	aws:ResourceTag/ <i>key-name</i> aws:TagKeys (Optional)	<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:applicationversion/My App/My Version"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } }] } }</pre>
Aktion: <code>DescribeInstancesHealth</code>		

Ressource	Bedingungen	Beispielanweisung
environment	-	<p>Mit der folgenden Richtlinie wird die Verwendung von <code>DescribeInstancesHealth</code> zum Abrufen von Zustandsinformationen für Instances in einer Umgebung namens <code>myenv</code> zugelassen.</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>
	Aktion: DescribePlatformVersion	
platform	<p><code>aws:ResourceTag/ <i>key-name</i></code> (Optional)</p> <p><code>aws:TagKeys</code> (Optional)</p>	<p>Die folgende Richtlinie ermöglicht der <code>DescribePlatformVersion</code>-Aktion, Plattformversionen mit Ausrichtung auf die <code>us-east-2</code>-Region zu beschreiben, deren Name mit <code>us-east-2_</code> beginnt:</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>
	Aktion: ListAvailableSolutionStacks	

Ressource	Bedingungen	Beispielanweisung
solutionstack	-	<p>Mit der folgenden Richtlinie kann die <code>ListAvailableSolutionStacks</code>-Aktion nur den Lösungs-Stack 32bit Amazon Linux running Tomcat 7 zurückgeben.</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>
Aktion: ListPlatformVersions		
platform	<p><code>aws:RequestTag/ <i>key-name</i></code> (Optional)</p> <p><code>aws:TagKeys</code> (Optional)</p>	<p>In diesem Beispiel darf die <code>CreatePlatformVersion</code>-Aktion Plattformversionen erstellen, die auf die us-east-2-Region ausgerichtet sind und deren Name mit us-east-2_ beginnt:</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>
Aktion: ListTagsForResource		

Ressource	Bedingungen	Beispielanweisung
application applicationversion configurationtemplate environment platform	aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>Die folgende Richtlinie ermöglicht der ListTagsForResource-Aktion das Auflisten von Tags von vorhandenen Umgebungen nur, wenn sie über ein Tag mit Namen stage mit dem Wert test verfügen:</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk>ListTagsForResource"], "Effect": "Allow", "Resource": "*", "Condition": { "StringEquals": { "aws:ResourceTag stage": ["test"] } } }] }</pre>
Aktion: RebuildEnvironment		
environment	InApplication aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>Die folgende Richtlinie erlaubt der RebuildEnvironment-Aktion, die Umgebung myenv in der Anwendung My App neu zu erstellen.</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>
Aktion: RequestEnvironmentInfo		

Ressource	Bedingungen	Beispielanweisung
environment	InApplication aws:ResourceTag/ key-name (Optional) aws:TagKeys (Optional)	Die folgende Richtlinie erlaubt der RequestEnvironmentInfo-Aktion, Informationen zur Umgebung myenv in der Anwendung My App zusammenzustellen.
		<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>
Aktion: RestartAppServer		
environment	InApplication	Die folgende Richtlinie ermöglicht der RestartAppServer-Aktion, den Anwendungscontainer-Server für die Umgebung myenv in der Anwendung My App neu zu starten.
		<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>
Aktion: RetrieveEnvironmentInfo		

Ressource	Bedingungen	Beispielanweisung
environment	InApplication aws:ResourceTag/ key-name (Optional) aws:TagKeys (Optional)	Die folgende Richtlinie erlaubt der RetrieveEnvironmentInfo-Aktion, die zusammengestellten Informationen über die Umgebung myenv in der Anwendung My App abzurufen. <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>
Aktion: SwapEnvironmentCNAMEs		
environment	InApplication (Optional) FromEnvironment (Optional)	Mit der folgenden Richtlinie kann die SwapEnvironmentCNAMEs-Aktion die CNAMEs für die Umgebungen mysrcenv und mydestenv austauschen. <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>
Aktion: TerminateEnvironment		

Ressource	Bedingungen	Beispielanweisung
environment	InApplication aws:ResourceTag/ key-name (Optional) aws:TagKeys (Optional)	Die folgende Richtlinie erlaubt der TerminateEnvironment-Aktion, die Umgebung myenv in der Anwendung My App zu beenden. <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:TerminateEnvironment"], "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>
Aktion: UpdateApplication		
application	aws:ResourceTag/ key-name (Optional) aws:TagKeys (Optional)	Mit dieser Richtlinie kann die UpdateApplication-Aktion Eigenschaften der Anwendung My App aktualisieren. <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:UpdateApplication"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] }] }</pre>
Aktion: UpdateApplicationResourceLifecycle		

Ressource	Bedingungen	Beispielanweisung
application	<p><code>aws:ResourceTag/ <i>key-name</i></code> (Optional)</p> <p><code>aws:TagKeys</code> (Optional)</p>	<p>Die folgende Richtlinie erlaubt der <code>UpdateApplicationResourceLifecycle</code>-Aktion das Aktualisieren von Lebenszykluseinstellungen der Anwendung My App.</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:UpdateApplicationResourceLifecycle"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] }] }</pre>
Aktion: UpdateApplicationVersion		
applicationversion	<p><code>InApplication</code></p> <p><code>aws:ResourceTag/ <i>key-name</i></code> (Optional)</p> <p><code>aws:TagKeys</code> (Optional)</p>	<p>Die folgende Richtlinie erlaubt der <code>UpdateApplicationVersion</code>-Aktion, die Eigenschaften der Anwendungsversion My Version in der Anwendung My App zu aktualisieren.</p> <pre>{ "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"] } }] }</pre>
Aktion: UpdateConfigurationTemplate		

Ressource	Bedingungen	Beispielanweisung
configurationtemplate	<p>In Application</p> <p>aws:ResourceTag/ key-name (Optional)</p> <p>aws:TagKeys (Optional)</p>	<p>Die folgende Richtlinie erlaubt der UpdateConfigurationTemplate-Aktion, die Eigenschaften oder Optionen der Konfigurationsvorlage My Template in der Anwendung My App zu aktualisieren.</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>

Aktion: [UpdateEnvironment](#)

Ressource	Bedingungen	Beispielanweisung
environment	InApplication FromApplicationVersion FromConfigurationTemplate aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>Die folgende Richtlinie ermöglicht der UpdateEnvironment-Aktion das Aktualisieren der Umgebung myenv in der Anwendung My App, indem die Anwendungsversion My Version bereitgestellt wird.</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>
Aktion: UpdateTagsForResource – AddTags		

Ressource	Bedingungen	Beispielanweisung
application applicationversion configurationtemplate environment platform	aws:ResourceTag/ <i>key-name</i> (Optional) aws:RequestTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>Die AddTags-Aktion ist eine von zwei virtuellen Aktionen, die mit der UpdateTagsForResource-API verknüpft sind.</p> <p>Die folgende Richtlinie ermöglicht der AddTags-Aktion das Ändern von Tags von vorhandenen Umgebungen nur, wenn sie über ein Tag mit Namen stage mit dem Wert test verfügen:</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:AddTags"], "Effect": "Allow", "Resource": "*", "Condition": { "StringEquals": { "aws:ResourceTag/stage": ["test"] } } }] }</pre>
Aktion: UpdateTagsForResource - RemoveTags		
application applicationversion configurationtemplate environment platform	aws:ResourceTag/ <i>key-name</i> (Optional) aws:TagKeys (Optional)	<p>Die RemoveTags-Aktion ist eine von zwei virtuellen Aktionen, die mit der UpdateTagsForResource-API verknüpft sind.</p> <p>Die folgende Richtlinie verweigert der RemoveTags-Aktion, die Entfernung eines Tags mit Namen stage aus bestehenden Umgebungen anzufordern:</p> <pre>{ "Version": "2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:RemoveTags"], "Effect": "Deny", "Resource": "*", "Condition": { "ForAnyValue:StringEquals": { "aws:TagKeys": ["stage"] } } }] }</pre>
Aktion: ValidateConfigurationSettings		

Ressource	Bedingungen	Beispielanweisung
template environment	InApplication aws:ResourceTag/ key-name (Optional) aws:TagKeys (Optional)	<p>Mit dieser Richtlinie kann die ValidateConfigurationSettings-Aktion die Konfigurationseinstellungen für die Umgebung myenv in der Anwendung My App überprüfen.</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>

Bedingungsschlüssel für Elastic Beanstalk-Aktionen

Mit Schlüsseln können Sie Bedingungen festlegen, die Abhängigkeiten ausdrücken, Berechtigungen beschränken oder Einschränkungen bei den Eingabeparametern für eine Aktion angeben. Elastic Beanstalk unterstützt die folgenden Schlüssel.

InApplication

Gibt die Anwendung an, die die Ressource enthält, auf der die Aktion ausgeführt wird.

Im folgenden Beispiel kann die UpdateApplicationVersion-Aktion die Eigenschaften der Anwendungsversion **My Version** aktualisieren. Die InApplication-Bedingung gibt **My App** als Container für **My Version** an.

```
{
    "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

Gibt eine Anwendungsversion als eine Abhängigkeit oder eine Einschränkung auf einem Eingabeparameter an.

Das folgende Beispiel erlaubt der UpdateEnvironment-Aktion, die Umgebung **myenv** in der Anwendung **My App** zu aktualisieren. Die FromApplicationVersion-Bedingung schränkt den VersionLabel-Parameter so ein, dass nur die Anwendungsversion **My Version** die Umgebung aktualisieren kann.

```
{
  "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"]
        }
      }
    ]
}
```

FromConfigurationTemplate

Gibt eine Konfigurationsvorlage als eine Abhängigkeit oder eine Einschränkung auf einem Eingabeparameter an.

Das folgende Beispiel erlaubt der UpdateEnvironment-Aktion, die Umgebung **myenv** in der Anwendung **My App** zu aktualisieren. Die FromConfigurationTemplate-Bedingung schränkt den TemplateName-Parameter so ein, dass nur die Konfigurationsvorlage **My Template** die Umgebung aktualisieren kann.

```
{
  "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

Gibt eine Umgebung als eine Abhängigkeit oder eine Einschränkung auf einem Eingabeparameter an.

Im folgenden Beispiel kann die `SwapEnvironmentCNAMES`-Aktion die CNAMEs in **My App** für alle Umgebungen austauschen, deren Namen mit `mysrcenv` und `mydestenv` beginnen. Dies gilt jedoch nicht für Umgebungen, deren Namen mit `mysrcenvPROD*` und `mydestenvPROD*` beginnen.

```
{
    "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

Gibt einen Lösungs-Stack als eine Abhängigkeit oder eine Einschränkung auf einem Eingabeparameter an.

Mit der folgenden Richtlinie kann die `CreateConfigurationTemplate`-Aktion Konfigurationsvorlagen, deren Name mit **My Template** (**My Template***) beginnt, in der Anwendung **My App** erstellen. Die `FromSolutionStack`-Bedingung schränkt den `solutionstack`-Parameter so ein, dass nur der Lösungs-Stack **32bit Amazon Linux running Tomcat 7** als Eingabewert für diesen Parameter zulässig ist.

```
{
    "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`

Geben Sie Tag-basierte Bedingungen an. Details dazu finden Sie unter [Tags zur Steuerung des Zugriffs auf Elastic Beanstalk-Ressourcen verwenden \(p. 960\)](#).

Tags zur Steuerung des Zugriffs auf Elastic Beanstalk-Ressourcen verwenden

Bedingungen in AWS Identity and Access Management (IAM)-Benutzerrichtlinienanweisungen sind Teil der Syntax, die Sie verwenden, um Berechtigungen für Ressourcen anzugeben, die Elastic Beanstalk-Aktionen ausführen müssen. Weitere Informationen zum Festlegen von Bedingungen für Richtlinienanweisungen finden Sie unter [Ressourcen und Bedingungen für Elastic Beanstalk-Aktionen \(p. 934\)](#). Das Verwenden von Tags in Bedingungen ist eine Möglichkeit zur Kontrolle des Zugriffs auf Ressourcen und Anfragen. Informationen zum Tagging von Elastic Beanstalk-Ressourcen finden Sie unter [Markieren von Elastic Beanstalk-Anwendungsressourcen \(p. 420\)](#). Dieses Thema behandelt die Tag-basierte Zugriffskontrolle.

Wenn Sie IAM-Richtlinien entwerfen, können Sie präzise abgestufte Berechtigungen festlegen, indem Sie Zugriff auf bestimmte Ressourcen gewähren. Mit zunehmender Anzahl der Ressourcen, die Sie verwalten, wird diese Aufgabe erschwert. Durch Markieren von Ressourcen und Verwenden von Tags in Richtlinienanweisungsbedingungen lässt sich diese Aufgabe vereinfachen. Sie erteilen Massenzugriff auf eine beliebige Ressource mit einem bestimmten Tag. Anschließend wenden Sie dieses Tag während der Erstellung oder zu einem späteren Zeitpunkt wiederholt auf relevante Ressourcen an.

Tags können an die Ressource angehängt oder in der Anfrage an Services weitergeleitet werden, die das Markieren unterstützen. In Elastic Beanstalk können Ressourcen Tags haben, und einige Aktionen können Tags enthalten. Wenn Sie eine IAM-Richtlinie erstellen, können Sie Tag-Bedingungsschlüssel verwenden, um Folgendes zu kontrollieren:

- Welche Benutzer Aktionen für eine Umgebung ausführen können, basierend auf bereits vorhandenen Tags.
- Welche Tags in der Anforderung einer Aktion übergeben werden können.
- Ob bestimmte Tag-Schlüssel in einer Anforderung verwendet werden können.

Die vollständige Syntax und Semantik der Tag-Bedingungsschlüssel finden Sie unter [Steuern des Zugriffs mit Tags im IAM-Benutzerhandbuch](#).

Die folgenden Beispiele zeigen, wie Sie Tag-Bedingungen in Richtlinien für Elastic Beanstalk-Benutzer festlegen können.

Example 1: Einschränken von Aktionen basierend auf Tags in der Anforderung

Die von Elastic Beanstalk AWSElasticBeanstalkFullAccess verwaltete Benutzerrichtlinie gibt Benutzern die uneingeschränkte Erlaubnis, jede Elastic Beanstalk-Aktion für jede Ressource durchzuführen.

Mit der folgenden Richtlinie wird nicht autorisierten Benutzern die Berechtigung verweigert, Elastic Beanstalk-Produktionsumgebungen zu erstellen. Dazu verweigert es die `CreateEnvironment`-Aktion, wenn die Anforderung ein Tag mit dem Namen `stage` und einem der Werte `gamma` oder `prod` festlegt. Darüber hinaus verhindert diese Richtlinie, dass nicht autorisierte Benutzer die Phase der Produktionsumgebungen manipulieren, indem sie Tag-Änderungsaktionen verhindert, bei denen die gleichen Tag-Werte betroffen sind oder bei denen das `stage`-Tag vollständig entfernt wird. Der Administrator eines Kunden muss diese IAM-Richtlinie nicht autorisierten IAM-Benutzern hinzufügen, zusätzlich zu der verwalteten Benutzerrichtlinie.

```
{  
    "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: Einschränken von Aktionen basierend auf Ressourcen-Tags

Die von Elastic Beanstalk AWSElasticBeanstalkFullAccess verwaltete Benutzerrichtlinie gibt Benutzern die uneingeschränkte Erlaubnis, jede Elastic Beanstalk-Aktion für jede Ressource durchzuführen.

Mit der folgenden Richtlinie wird Benutzern die Berechtigung verweigert, Aktionen für Elastic Beanstalk-Produktionsumgebungen auszuführen. Dazu verweigert es bestimmte Aktionen, wenn die Umgebung über ein Tag mit dem Namen `stage` und einem der Werte `gamma` oder `prod` verfügt. Der Administrator eines Kunden muss diese IAM-Richtlinie nicht autorisierten IAM-Benutzern hinzufügen, zusätzlich zu der verwalteten Benutzerrichtlinie.

```
{  
    "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: Zulassen von Aktionen basierend auf Tags in der Anforderung

Mit der folgenden Richtlinie wird Benutzern die Berechtigung zum Erstellen von Elastic Beanstalk-Entwicklungsanwendungen erteilt.

Dazu werden die Aktionen `CreateApplication` und `AddTags` zugelassen, wenn die Anforderung ein Tag mit dem Namen `stage` mit dem Wert `development` angibt. Die `aws:TagKeys`-Bedingung stellt sicher, dass der Benutzer keine anderen Tag-Schlüssel hinzufügen kann. Insbesondere wird sichergestellt, dass die Groß-/Kleinschreibung des `stage`-Tag-Schlüssels berücksichtigt wird. Beachten Sie, dass sich diese Richtlinie für IAM-Benutzer eignet, denen die verwaltete Benutzerrichtlinie `Elastic Beanstalk AWSElasticBeanstalkFullAccess` nicht angefügt ist. Die verwaltete Richtlinie erteilt Benutzern die uneingeschränkte Berechtigung, eine Elastic Beanstalk-Aktion auf allen Ressourcen auszuführen.

```
{
    "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: Zulassen von Aktionen basierend auf Ressourcen-Tags

Mit der folgenden Richtlinie wird Benutzern die Berechtigung zum Ausführen von Aktionen auf und Abrufen von Informationen zu Elastic Beanstalk-Entwicklungsanwendungen gewährt.

Dazu werden bestimmte Aktionen zugelassen, wenn die Anwendung ein Tag mit dem Namen `stage` mit dem Wert `development` enthält. Die `aws:TagKeys`-Bedingung stellt sicher, dass der Benutzer keine

anderen Tag-Schlüssel hinzufügen kann. Insbesondere wird sichergestellt, dass die Groß-/Kleinschreibung des stage-Tag-Schlüssels berücksichtigt wird. Beachten Sie, dass sich diese Richtlinie für IAM-Benutzer eignet, denen die die verwaltete Benutzerrichtlinie Elastic Beanstalk AWSElasticBeanstalkFullAccess nicht angefügt ist. Die verwaltete Richtlinie erteilt Benutzern die uneingeschränkte Berechtigung, eine Elastic Beanstalk-Aktion auf allen Ressourcen auszuführen.

```
{  
    "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"]  
                }  
            }  
        }  
    ]  
}
```

Beispielrichtlinien basierend auf verwalteten Richtlinien

In diesem Abschnitt wird erläutert, wie Sie Benutzerzugriff auf AWS Elastic Beanstalk steuern, und es werden Beispielrichtlinien gezeigt, die den erforderlichen Zugriff für gängige Szenarien bereitstellen. Diese Richtlinien sind von den von Elastic Beanstalk verwalteten Richtlinien abgeleitet. Weitere Informationen zum Anfügen von verwalteten Richtlinien für Benutzer und Gruppen finden Sie unter [Elastic Beanstalk-Benutzerrichtlinien verwalten \(p. 926\)](#).

In diesem Szenario ist Example Corp. ein Software-Unternehmen mit drei Teams, die für die Unternehmens-Website verantwortlich sind: Administratoren, die die Infrastruktur verwalten, Entwickler, die die Software für die Website erstellen, und ein QA-Team, das die Website testet. Damit die Berechtigungen für ihre Elastic Beanstalk-Ressourcen einfacher verwaltet werden können, erstellt Example Corp. drei Gruppen, zu denen Mitglieder der jeweiligen Teams gehören: Administratoren, Entwickler und Tester. Example Corp. möchte, dass die Admins-Gruppe vollständigen Zugriff auf alle Anwendungen, Umgebungen und ihre zugrundeliegenden Ressourcen hat, damit sie alle Elastic Beanstalk-Komponenten erstellen, beheben und löschen können. Entwickler benötigen Berechtigungen zur Anzeige aller Elastic Beanstalk-Komponenten und zum Erstellen und Bereitstellen von Anwendungsversionen. Entwickler sollten nicht in der Lage sein, neue Anwendungen oder Umgebungen zu erstellen oder laufende Umgebungen zu beenden. Tester müssen alle Elastic Beanstalk-Ressourcen anzeigen, um Anwendungen überwachen und testen zu können. Die Tester sollten nicht in der Lage sein, Änderungen an Elastic Beanstalk-Ressourcen vorzunehmen.

Die folgenden Beispielrichtlinien bieten die erforderlichen Berechtigungen für jede Gruppe.

Beispiel 1: Admins-Gruppe – Alle Elastic Beanstalk-APIs und APIs verwandter Services

Mit der folgenden Richtlinie erhalten Benutzer Berechtigungen für alle Aktionen, die zur Verwendung von Elastic Beanstalk erforderlich sind. Mit dieser Richtlinie kann Elastic Beanstalk auch Ressourcen in Ihrem

Namen in den folgenden Services bereitstellen und verwalten. Elastic Beanstalk nutzt diese zusätzlichen Services zur Bereitstellung zugrunde liegender Ressourcen für die Umgebungserstellung.

- 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

Beachten Sie, dass diese Richtlinie als Beispiel dient. Sie bietet umfangreiche Berechtigungen für AWS-Services, die von Elastic Beanstalk zur Anwendungs- und Umgebungsverwaltung herangezogen werden. Zum Beispiel erlaubt `ec2:*` einem AWS Identity and Access Management (IAM)-Benutzer, jede beliebige Aktion für eine beliebige Amazon EC2-Ressource im AWS-Konto durchzuführen. Diese Berechtigungen sind nicht auf Ressourcen beschränkt, die mit Elastic Beanstalk verwendet werden. Als bewährte Methode sollten Sie Personen nur die Berechtigungen erteilen, die sie für ihre Aufgaben benötigen.

```
{  
    "Version" : "2012-10-17",  
    "Statement" : [  
        {  
            "Effect" : "Allow",  
            "Action" : [  
                "elasticbeanstalk:*",  
                "ec2:*",  
                "elasticloadbalancing:*",  
                "autoscaling:*",  
                "cloudwatch:*",  
                "s3:*",  
                "sns:*",  
                "rds:*",  
                "cloudformation:/*"  
            ],  
            "Resource" : "*"  
        }  
    ]  
}
```

Beispiel 2: Entwicklergruppe – Alle Operationen außer solchen mit hoher Privilegierung

Mit der folgenden Richtlinie wird die Berechtigung zum Erstellen von Anwendungen und Umgebungen abgelehnt und alle anderen Elastic Beanstalk-Aktionen werden zugelassen.

Beachten Sie, dass diese Richtlinie als Beispiel dient. Sie bietet umfangreiche Berechtigungen für AWS-Produkte, die von Elastic Beanstalk zur Anwendungs- und Umgebungsverwaltung herangezogen werden. Beispielsweise kann ein IAM-Benutzer mit `ec2:*` alle Aktionen für alle Amazon EC2-Ressourcen des AWS-Kontos ausführen. Diese Berechtigungen sind nicht auf Ressourcen beschränkt, die mit Elastic Beanstalk verwendet werden. Als bewährte Methode sollten Sie Personen nur die Berechtigungen erteilen, die sie für ihre Aufgaben benötigen.

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

Beispiel 3: Tester – Nur anzeigen

Mit der folgenden Richtlinie wird Lesezugriff für alle Anwendungen, Anwendungsversionen, Veranstaltungen und Umgebungen zugelassen. Das Ausführen von Aktionen wird nicht zugelassen.

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

}]

Beispielrichtlinien auf Basis von Ressourcenberechtigungen

In diesem Abschnitt wird ein Anwendungsfall für die Steuerung der Benutzerberechtigungen für Elastic Beanstalk-Aktionen durchgegangen, die auf bestimmte Elastic Beanstalk-Ressourcen zugreifen.

Dabei werden die Beispielrichtlinien für diesen Anwendungsfall vorgestellt. Weitere Informationen zu den Richtlinien für Elastic Beanstalk-Ressourcen finden Sie unter [Erstellen einer benutzerdefinierten Benutzerrichtlinie \(p. 929\)](#). Weitere Informationen zum Verknüpfen von Richtlinien mit Benutzern und Gruppen finden Sie im Abschnitt [Managing IAM Policies](#) unter Using AWS Identity and Access Management.

In diesem Anwendungsfall entwickelt das kleine Beratungsunternehmen Example Corp. Anwendungen für zwei verschiedene Kunden. Als Entwicklungsmanager ist John für die Entwicklung der beiden Elastic Beanstalk-Anwendungen "app1" und "app2" zuständig. John entwickelt die beiden Anwendungen und führt einige Tests aus; nur er kann die Produktionsumgebung für die beiden Anwendungen aktualisieren. Für "app1" und "app2" benötigt er folgende Berechtigungen:

- Anzeigen der Anwendung, Anwendungsversionen, Umgebungen und Konfigurationsvorlagen
- Erstellen von Anwendungsversionen sowie deren Bereitstellung in der Staging-Umgebung
- Aktualisieren der Produktionsumgebung
- Erstellen und Beenden von Umgebungen

Jill ist Testerin und muss die folgenden Ressourcen anzeigen können, um die beiden Anwendungen zu überwachen und zu testen: Anwendungen, Anwendungsversionen, Umgebungen und Konfigurationsvorlagen. Sie sollte jedoch nicht in der Lage sein, Änderungen an Elastic Beanstalk-Ressourcen vorzunehmen.

Jack ist der Entwickler von "app1" und muss alle Ressourcen für "app1" anzeigen können. Zudem soll er Anwendungsversionen für "app1" erstellen und diese in der Staging-Umgebung bereitstellen.

Judy ist der Administrator des AWS-Kontos von Example Corp. Sie hat die IAM-Benutzer für John, Jill und Jack erstellt und weist folgende Richtlinien den Benutzern zu, um die benötigten Berechtigungen für die Anwendungen „app1“ und „app2“ zu erteilen.

Beispiel 1: John – Entwicklungsmanager für app1, app2

Wir haben die Richtlinie von John in drei einzelne Richtlinien unterteilt, denn diese können einfacher gelesen und verwaltet werden. Gemeinsam geben sie John die Berechtigungen, die er benötigt, um Entwicklungs-, Test- und Bereitstellungsaktionen für die beiden Anwendungen durchzuführen.

Die erste Richtlinie legt Aktionen für Auto Scaling, Amazon S3, Amazon EC2, CloudWatch, Amazon SNS, Elastic Load Balancing, Amazon RDS und AWS CloudFormation fest. Elastic Beanstalk nutzt diese zusätzlichen Services zur Bereitstellung zugrunde liegender Ressourcen für die Umgebungserstellung.

Beachten Sie, dass diese Richtlinie als Beispiel dient. Sie bietet umfangreiche Berechtigungen für AWS-Produkte, die von Elastic Beanstalk zur Anwendungs- und Umgebungsverwaltung herangezogen werden. Beispielsweise kann ein IAM-Benutzer mit `ec2 : *` alle Aktionen für alle Amazon EC2-Ressourcen des AWS-Kontos ausführen. Diese Berechtigungen sind nicht auf Ressourcen beschränkt, die mit Elastic Beanstalk verwendet werden. Als bewährte Methode sollten Sie Personen nur die Berechtigungen erteilen, die sie für ihre Aufgaben benötigen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*",
                "ecs:*",
                "ecr:*",
                "elasticloadbalancing:*",
                "autoscaling:*",
                "cloudwatch:*",
                "s3:*",
                "sns:*",
                "cloudformation:*",
                "dynamodb:*",
                "rds:*",
                "sns:*",
                "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": "*"
        }
    ]
}
```

In der zweiten Richtlinie werden die Elastic Beanstalk-Aktionen definiert, die John für die Ressourcen von "app1" und "app2" ausführen kann. Mit der Anweisung `AllCallsInApplications` können alle Elastic Beanstalk-Aktionen ("elasticbeanstalk:") für alle Ressourcen in "app1" und "app2" ausgeführt werden (z. B. `elasticbeanstalk>CreateEnvironment`). Mit der Anweisung `AllCallsOnApplications` können alle Elastic Beanstalk-Aktionen ("elasticbeanstalk:") für die Anwendungsressourcen von "app1" und "app2" ausgeführt werden (z. B. `elasticbeanstalk:DescribeApplications`, `elasticbeanstalk:UpdateApplication` usw.). Mit der Anweisung `AllCallsOnSolutionStacks` können alle Elastic Beanstalk-Aktionen ("elasticbeanstalk:") für Lösungs-Stack-Ressourcen ausgeführt werden (z. B. `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/*"  
    ]  
}  
]  
}
```

Die dritte Richtlinie legt die Elastic Beanstalk-Aktionen fest, für die die zweite Richtlinie Berechtigungen benötigt, um diese Elastic Beanstalk-Aktionen abzuschließen. Mit der Anweisung `AllNonResourceCalls` kann die `elasticbeanstalk:CheckDNSAvailability`-Aktion ausgeführt werden, die für den Aufruf von `elasticbeanstalk>CreateEnvironment` und anderen Aktionen erforderlich ist. Zudem kann damit die `elasticbeanstalk>CreateStorageLocation`-Aktion ausgeführt werden, die für `elasticbeanstalk>CreateApplication`, `elasticbeanstalk>CreateEnvironment` und weitere Aktionen benötigt wird.

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

Beispiel 2: Jill – Tester für app1, app2

Wir haben die Richtlinie von Jill in drei einzelne Richtlinien unterteilt, denn diese können einfacher gelesen und verwaltet werden. Mit diesen drei Richtlinien erhält Jill die benötigten Berechtigungen, um die Aktionen für das Testen und Überwachen der beiden Anwendungen auszuführen.

Die erste Richtlinie legt die Aktionen `Describe*`, `List*` und `Get*` für Auto Scaling, Amazon S3, Amazon EC2, CloudWatch, Amazon SNS, Elastic Load Balancing, Amazon RDS und AWS CloudFormation (für nicht veraltete Containertypen) fest, sodass die Elastic Beanstalk-Aktionen in der Lage sind, die relevanten Informationen über die zugrundeliegenden Ressourcen der Anwendungen `app1` und `app2` abzurufen.

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

In der zweiten Richtlinie werden die Elastic Beanstalk-Aktionen definiert, die Jill für die Ressourcen von "app1" und "app2" ausführen kann. Mit der Anweisung `AllReadCallsInApplications` können die `Describe*`-Aktionen sowie die Aktionen für Umgebungsinformationen ausgeführt werden. Mit der Anweisung `AllReadCallsOnApplications` können die Aktionen `DescribeApplications` und `DescribeEvents` für die Anwendungsressourcen von "app1" und "app2" ausgeführt werden. Mit der Anweisung `AllReadCallsOnSolutionStacks` können Anzeigektionen für Lösungs-Stack-Ressourcen ausgeführt werden (`ListAvailableSolutionStacks`, `DescribeConfigurationOptions` und `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/*"
    ]
}
]
```

Die dritte Richtlinie legt die Elastic Beanstalk-Aktionen fest, für die die zweite Richtlinie Berechtigungen benötigt, um diese Elastic Beanstalk-Aktionen abzuschließen. Mit der Anweisung `AllNonResourceCalls` kann die `elasticbeanstalk:CheckDNSAvailability`-Aktion ausgeführt werden, die für einige Anzeigektionen erforderlich ist.

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

Beispiel 3: Jack – Entwickler für app1

Wir haben die Richtlinie von Jack in drei einzelne Richtlinien unterteilt, denn diese können einfacher gelesen und verwaltet werden. Gemeinsam geben sie Jack die Berechtigungen, die er benötigt, um Test-, Überwachungs- und Bereitstellungsaktionen auf der app1-Ressource durchzuführen.

Die erste Richtlinie legt die Aktionen für Auto Scaling, Amazon S3, Amazon EC2, CloudWatch, Amazon SNS, Elastic Load Balancing, Amazon RDS und AWS CloudFormation (für nicht veraltete Containertypen) fest, sodass die Elastic Beanstalk-Aktionen in der Lage sind, die zugrundeliegenden Ressourcen von app1 anzuseigen und mit ihnen zu arbeiten. Eine Liste mit unterstützten aktuellen Containertypen finden Sie unter [the section called “Warum sind einige Plattformversionen als Legacy gekennzeichnet?” \(p. 502\)](#).

Beachten Sie, dass diese Richtlinie als Beispiel dient. Sie bietet umfangreiche Berechtigungen für AWS-Produkte, die von Elastic Beanstalk zur Anwendungs- und Umgebungsverwaltung herangezogen werden. Beispielsweise kann ein IAM-Benutzer mit `ec2:*` alle Aktionen für alle Amazon EC2-Ressourcen des AWS-Kontos ausführen. Diese Berechtigungen sind nicht auf Ressourcen beschränkt, die mit Elastic Beanstalk verwendet werden. Als bewährte Methode sollten Sie Personen nur die Berechtigungen erteilen, die sie für ihre Aufgaben benötigen.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:*",  
                "elasticloadbalancing:*",  
                "autoscaling:*",  
                "cloudwatch:*",  
                "s3:*",  
                "sns:*",  
                "rds:*",  
                "cloudformation:*"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

In der zweiten Richtlinie werden die Elastic Beanstalk-Aktionen definiert, die Jack für die "app1"-Ressource ausführen kann.

```
{  
    "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/*"
    ]
}
]
}

```

Die dritte Richtlinie legt die Elastic Beanstalk-Aktionen fest, für die die zweite Richtlinie Berechtigungen benötigt, um diese Elastic Beanstalk-Aktionen abzuschließen. Mit der Anweisung AllNonResourceCalls kann die elasticbeanstalk:CheckDNSAvailability-Aktion ausgeführt werden, die für den Aufruf von elasticbeanstalk>CreateEnvironment und anderen Aktionen erforderlich ist. Zudem kann damit die elasticbeanstalk>CreateStorageLocation-Aktion ausgeführt werden, die für elasticbeanstalk>CreateEnvironment und weitere Aktionen benötigt wird.

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

}

Verwenden von Elastic Beanstalk mit Amazon RDS

AWS Elastic Beanstalk bietet Unterstützung für die Ausführung von [Amazon Relational Database Service \(Amazon RDS\)](#)-Instances in Ihrer Elastic Beanstalk-Umgebung. Weitere Informationen hierzu finden Sie unter [the section called “Datenbank” \(p. 609\)](#). Dies eignet sich hervorragend für Entwicklungs- und Testumgebungen, ist jedoch nicht ideal für eine Produktionsumgebung, da der Lebenszyklus der Datenbank-Instance an den Lebenszyklus Ihrer Anwendungsumgebung gebunden wird.

Note

Wenn Sie zuvor keine DB-Instance mit Ihrer Anwendung verwendet haben, versuchen Sie zunächst mithilfe der Elastic Beanstalk-Konsole eine DB-Instance zu einer Testumgebung hinzuzufügen. Dadurch können Sie prüfen, ob Ihre Anwendung Umgebungseigenschaften lesen, eine Verbindungszeichenfolge erstellen und eine Verbindung zu einer DB-Instance herstellen kann, bevor Sie [Amazon Virtual Private Cloud \(Amazon VPC\)](#) und die Sicherheitsgruppenkonfiguration hinzufügen. Details dazu finden Sie unter [Hinzufügen einer Datenbank zu Ihrer Elastic Beanstalk-Umgebung \(p. 609\)](#).

Wenn Sie Ihre Datenbank-Instance aus Ihrer Umgebung entkoppeln möchten, können Sie eine Datenbank-Instance in Amazon RDS ausführen und Ihre Anwendung so konfigurieren, dass Sie beim Start verbunden wird. Auf diese Weise können Sie mehrere Umgebungen mit einer Datenbank verbinden, eine Umgebung beenden, ohne dass dies Auswirkungen auf die Datenbank hat, und nahtlose Updates mit Blau/Grün-Bereitstellungen durchführen. Eine ausführliche Anleitung finden Sie unter [Wie kann ich eine Amazon RDS-Instance von einer AWS Elastic Beanstalk-Umgebung ohne Ausfallzeiten, Probleme bei der Datenbanksynchronisierung oder Datenverlust entkoppeln?](#).

Damit die Amazon EC2-Instances in Ihrer Umgebung eine Verbindung mit einer externen Datenbank herstellen können, müssen Sie die Auto Scaling-Gruppe der Umgebung mit einer zusätzlichen Sicherheitsgruppe konfigurieren. Die Sicherheitsgruppe, die Sie an Ihre Umgebung anfügen, kann identisch sein mit der, die Sie an Ihre Datenbank-Instance angefügt haben. Es kann sich auch um eine separate Sicherheitsgruppe handeln, von der die Sicherheitsgruppe der Datenbank eingehenden Zugriff erlaubt.

Note

Sie können Ihre Umgebung mit einer Datenbank verbinden, indem Sie eine Regel zur Sicherheitsgruppe Ihrer Datenbank hinzufügen, die eingehenden Zugriff von der automatisch generierten Sicherheitsgruppe erlaubt, die Elastic Beanstalk an die Auto Scaling-Gruppe Ihre Umgebung anfügt. Beachten Sie jedoch, dass dadurch eine Abhängigkeit zwischen den beiden Sicherheitsgruppen entsteht. Wenn Sie anschließend versuchen, die Umgebung zu beenden, kann Elastic Beanstalk die Sicherheitsgruppe der Umgebung nicht beenden, da die Sicherheitsgruppe der Datenbank davon abhängig ist.

Nach dem Starten Ihrer Datenbank-Instance und der Konfiguration der Sicherheitsgruppen können Sie die Verbindungsinformationen (Endpunkt, Passwort usw.) an die Anwendung übergeben, indem Sie Umgebungseigenschaften verwenden. Hierbei handelt es sich um denselben Mechanismus, den Elastic Beanstalk verwendet, wenn Sie eine Datenbank-Instance in Ihrer Umgebung ausführen.

Für zusätzliche Sicherheit können Sie Ihre Verbindungsinformationen in Amazon S3 speichern und Elastic Beanstalk so konfigurieren, dass sie während der Bereitstellung abgerufen werden. Mit [Konfigurationsdateien \(.ebextensions\) \(p. 722\)](#) können Sie die Instances in Ihrer Umgebung so konfigurieren, dass sie sicher Dateien aus Amazon S3 abrufen, wenn Sie Ihre Anwendung bereitstellen.

Themen

- [Starten von und Verbinden mit einer externen Amazon RDS-Instance in einer Standard-VPC \(p. 974\)](#)
- [Starten von und Verbinden mit einer externen Amazon RDS-Instance in EC2 Classic \(p. 979\)](#)
- [Speichern der Verbindungszeichenfolge in Amazon S3 \(p. 983\)](#)
- [Bereinigen einer externen Amazon RDS-Instance \(p. 985\)](#)

Starten von und Verbinden mit einer externen Amazon RDS-Instance in einer Standard-VPC

Um eine externe Datenbank mit einer Anwendung zu verwenden, die in Elastic Beanstalk ausgeführt wird, starten Sie zuerst eine DB-Instance mit Amazon RDS. Jede Instance, die Sie mit Amazon RDS starten, ist völlig unabhängig von Elastic Beanstalk und Ihren Elastic Beanstalk-Umgebungen und ist für die Konfiguration nicht von Elastic Beanstalk abhängig. Das bedeutet, dass Sie jede DB-Engine und jeden Instance-Typ verwenden können, der von Amazon RDS unterstützt wird, auch wenn er nicht von Elastic Beanstalk verwendet wird.

Die folgenden Verfahren beschreiben den Prozess für eine [Standard-VPC](#). Der Prozess ist identisch, wenn Sie eine benutzerdefinierte VPC verwenden. Die einzigen zusätzlichen Voraussetzungen sind, dass sich Ihre Umgebung und DB-Instance im selben Subnetz oder in Subnetzen befindet, die miteinander kommunizieren dürfen. Weitere Informationen zum Konfigurieren einer benutzerdefinierten VPC für die Verwendung mit Elastic Beanstalk finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon VPC \(p. 988\)](#).

So starten Sie eine RDS DB-Instance in einer Standard-VPC

1. Öffnen Sie die [Amazon RDS-Konsole](#).
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie Create database (Datenbank erstellen) aus.
4. Wählen Sie Standard Create (Standardeinstellung).

Important

Wählen Sie nicht Easy Create (Einfache Erstellung) aus. Sie können nicht die erforderlichen Einstellungen zum Start dieser RDS-DB konfigurieren.

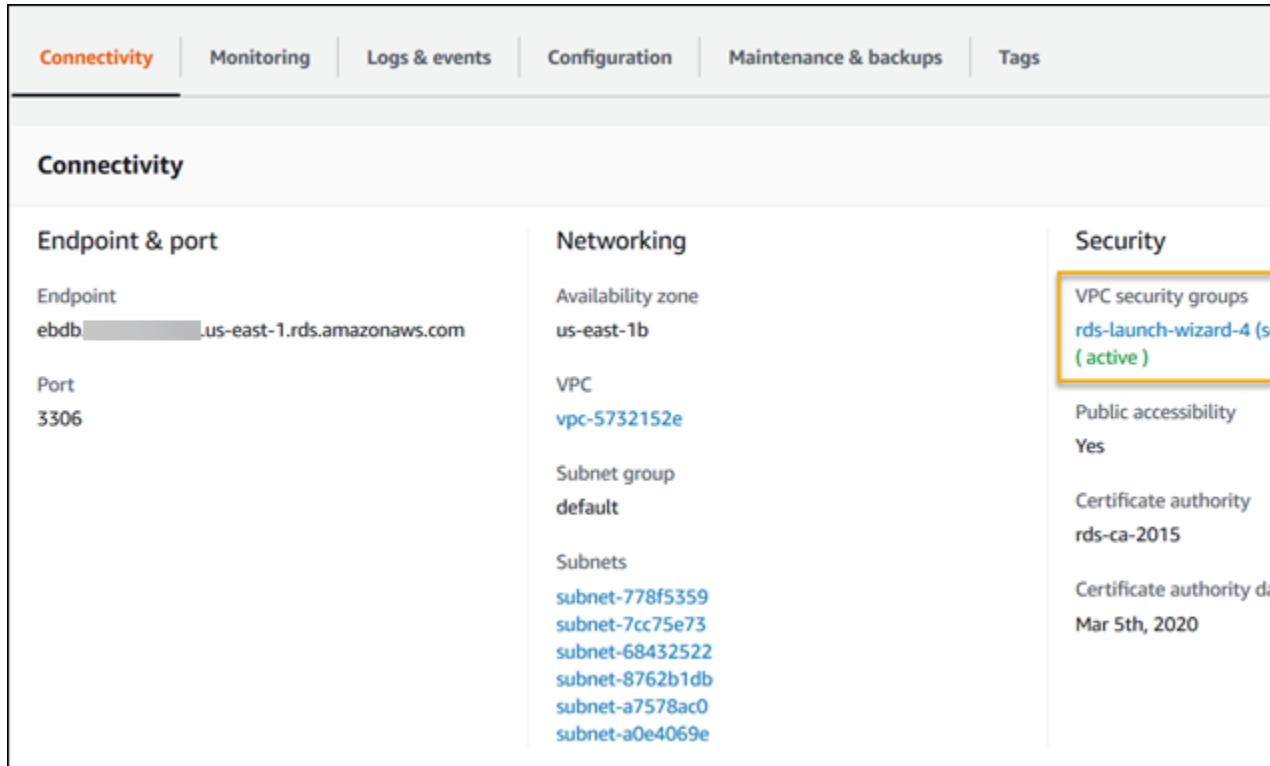
5. Geben Sie unter Additional configuration (Zusätzliche Konfiguration) für Initial database name (Anfangsdatenbankname) „**ebdb**“ ein.
6. Überprüfen Sie die Standardeinstellungen sorgfältig und passen Sie sie bei Bedarf an. Achten Sie auf die folgenden Optionen:
 - DB instance class (DB-Instance-Klasse) – Wählen Sie eine Instance-Größe, die eine geeignete Speichermenge und CPU-Leistung für den Workload bereitstellt.
 - Multi-AZ deployment (Multi-AZ-Bereitstellung) – Um hohe Verfügbarkeit zu gewährleisten, legen Sie dies unter Create an Aurora Replica/Reader node in a different AZ (Einen Aurora-Replikats-/Reader-Knoten in einer anderen AZ erstellen) fest.
 - Master username (Master-Benutzername) und Master password (Master-Passwort) – Der Benutzername und das Passwort für die Datenbank. Notieren Sie sich diese Einstellungen, um sie später zu verwenden.
7. Überprüfen Sie die Standardeinstellungen für die übrigen Optionen und wählen Sie dann Create database (Datenbank starten) aus.

Als Nächstes bearbeiten Sie die Sicherheitsgruppe für die DB-Instance, damit eingehender Datenverkehr über den entsprechenden Port zulässig ist. Es handelt sich um die gleiche Sicherheitsgruppe, die später

mit der Elastic Beanstalk-Umgebung verknüpft wird. Die von Ihnen hinzugefügte Regel gewährt damit Eingangsberechtigungen für andere Ressourcen in derselben Sicherheitsgruppe.

So ändern Sie die Eingangsregeln für die Sicherheitsgruppe der RDS-Instance

1. Öffnen Sie die [Amazon RDS-Konsole](#).
2. Wählen Sie Databases (Datenbanken) aus.
3. Wählen Sie den Namen der DB-Instance, um deren Details anzuzeigen.
4. Notieren Sie im Abschnitt Connectivity (Konnektivität) die auf dieser Seite angezeigten Subnets (Subnetze), Security groups (Sicherheitsgruppen) und den Endpoint (Endpunkt), damit Sie diese Informationen später verwenden können.
5. In Security (Sicherheit) wird die mit der DB-Instance verknüpfte Sicherheitsgruppe angezeigt. Öffnen Sie den Link, um die Sicherheitsgruppe in der Amazon EC2-Konsole aufzurufen.



The screenshot shows the 'Connectivity' tab of the AWS RDS console for a specific DB instance. The tab navigation bar includes Connectivity, Monitoring, Logs & events, Configuration, Maintenance & backups, and Tags. The 'Connectivity' tab is selected, indicated by an orange underline. Below the tabs, the word 'Connectivity' is repeated in a larger font. The main content area is divided into four columns: 'Endpoint & port', 'Networking', and 'Security' (with a 'Subnets' section below it). The 'Endpoint & port' column contains the endpoint address (ebdb...us-east-1.rds.amazonaws.com) and port (3306). The 'Networking' column contains the availability zone (us-east-1b) and VPC (vpc-5732152e). The 'Subnets' section lists several subnet IDs. The 'Security' column contains the VPC security group (rds-launch-wizard-4 (active)), which is highlighted with a yellow box. It also shows public accessibility (Yes), certificate authority (rds-ca-2015), and a certificate expiration date (Mar 5th, 2020).

Endpoint & port	Networking	Security
Endpoint ebdb...us-east-1.rds.amazonaws.com	Availability zone us-east-1b	VPC security groups rds-launch-wizard-4 (active)
Port 3306	VPC vpc-5732152e	Public accessibility Yes
	Subnet group default	Certificate authority rds-ca-2015
	Subnets subnet-778f5359 subnet-7cc75e73 subnet-68432522 subnet-8762b1db subnet-a7578ac0 subnet-a0e4069e	Certificate authority date Mar 5th, 2020

6. Klicken Sie in den Details der Sicherheitsgruppe auf Inbound (Eingehend).
7. Wählen Sie Edit (Bearbeiten) aus.
8. Klicken Sie auf Add Rule.
9. Wählen Sie unter Type (Typ) die von der Anwendung genutzte DB-Engine aus.
10. Geben Sie für Source (Quelle) **sg-** ein, um eine Liste der verfügbaren Sicherheitsgruppen anzuzeigen. Wählen Sie die Sicherheitsgruppe aus, die mit der Auto Scaling-Gruppe einer Elastic Beanstalk-Umgebung verknüpft ist, damit Amazon EC2-Instances in der Umgebung Zugriff auf die Datenbank haben.

11. Wählen Sie Save (Speichern) aus.

Als Nächstes fügen Sie die Sicherheitsgruppe der DB-Instance zur ausgeführten Umgebung hinzu. Dadurch stellt Elastic Beanstalk alle Instances in der Umgebung mit der zusätzlich zugeordneten Sicherheitsgruppe erneut bereit.

So fügen Sie eine Sicherheitsgruppe zur Umgebung hinzu

- Führen Sie eine der folgenden Aufgaben aus:
 - So fügen Sie eine Sicherheitsgruppe mit der Elastic Beanstalk-Konsole hinzu
 - a. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
 - b. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

- c. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
 - d. Wählen Sie in der Konfigurationskategorie Instances die Option Edit (Bearbeiten).
 - e. Wählen Sie unter EC2 security groups (EC2-Sicherheitsgruppen) neben der Instance-Sicherheitsgruppe, die Elastic Beanstalk erstellt, die Sicherheitsgruppe aus, die an die Instances angehängt werden soll.
 - f. Wählen Sie Apply (Anwenden) aus.
 - g. Lesen Sie die Warnmeldung und klicken Sie dann auf Bestätigen.
- Um eine Sicherheitsgruppe mit einer [Konfigurationsdatei \(p. 722\)](#) hinzuzufügen, verwenden Sie die Beispieldatei [securitygroup-addexisting.config](#).

Anschließend werden die Verbindungsinformationen mithilfe von Umgebungseigenschaften an die Umgebung übergeben. Wenn Sie mit der Elastic Beanstalk-Konsole [der Umgebung eine DB-Instance hinzufügen \(p. 609\)](#), verwendet Elastic Beanstalk Umgebungseigenschaften, wie RDS_HOSTNAME, um Verbindungsinformationen an die Anwendung zu übergeben. Sie können die gleichen Eigenschaften verwenden, mit denen Sie denselben Anwendungscode mit integrierten DB-Instances und externen DB-Instances verwenden können, oder Ihre eigenen Eigenschaftnamen wählen.

So konfigurieren Sie Umgebungseigenschaften für eine Amazon RDS-DB-Instance

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Definieren Sie im Abschnitt Environment properties (Umgebungseigenschaften) die Variablen, die Ihre Anwendung liest, um eine Verbindungszeichenfolge zu konstruieren. Für die Kompatibilität mit Umgebungen, die eine integrierte RDS-DB-Instance haben, verwenden Sie die folgenden Namen und Werte. Sie können alle Werte außer Ihrem Passwort in der [RDS-Konsole](#) finden.

Eigenschaftenname	Beschreibung	Eigenschaftswert
RDS_HOSTNAME	Der Hostname der DB-Instance.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) in der Amazon RDS-Konsole: Endpoint (Endpunkt).
RDS_PORT	Der Port, über den die DB-Instance Verbindungen annimmt. Die DB-Engines haben unterschiedliche Standardwerte.	Auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) auf der Amazon RDS-Konsole: Port.
RDS_DB_NAME	Der Datenbankname, ebdb	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: DB Name (DB-Name).
RDS_USERNAME	Der Benutzername, der für die Datenbank konfiguriert wurde.	Auf der Registerkarte Configuration (Konfiguration) in der Amazon RDS-Konsole: Master username (Master-Benutzername).
RDS_PASSWORD	Das Passwort, das für die Datenbank konfiguriert wurde.	In der Amazon RDS-Konsole nicht als Referenz verfügbar.

Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	ebdb
RDS_HOSTNAME	webapp-db.jxccb5mpan
RDS_PORT	5432
RDS_USERNAME	webapp-admin
RDS_PASSWORD	kUj5uKxmWDMYc403

[Cancel](#)

6. Wählen Sie Apply (Anwenden) aus.

Wenn Sie Ihre Anwendung noch nicht für das Programmieren von Umgebungseigenschaften und Erstellen einer Verbindungszeichenfolge programmiert haben, finden Sie Anweisungen in den folgenden sprachspezifischen Themen:

- Java SE – [Verbinden mit einer Datenbank \(Java SE-Plattformen\)](#) (p. 136)
- Java mit Tomcat – [Verbinden mit einer Datenbank \(Tomcat-Plattformen\)](#) (p. 137)
- Node.js – [Verbinden mit einer Datenbank](#) (p. 285)
- .NET – [Verbinden mit einer Datenbank](#) (p. 219)
- PHP – [Verbinden mit einer Datenbank mit einem PDO oder MySQLi](#) (p. 351)
- Python – [Verbinden mit einer Datenbank](#) (p. 380)
- Ruby – [Verbinden mit einer Datenbank](#) (p. 399)

Je nachdem, wann Ihre Anwendung Umgebungsvariablen liest, müssen Sie den Anwendungsserver auf den Instances in Ihrer Umgebung möglicherweise neu starten.

So starten Sie die App-Server der Umgebung neu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.

2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) und dann Restart app server(s) (App-Server neu starten) aus.

Starten von und Verbinden mit einer externen Amazon RDS-Instance in EC2 Classic

Wenn Sie EC2 Classic (ohne VPC) mit AWS Elastic Beanstalk verwenden, ändert sich die Prozedur aufgrund von Unterschieden in der Funktionsweise der Sicherheitsgruppen leicht. In EC2 Classic können DB-Instances keine EC2-Sicherheitsgruppen verwenden, sodass sie eine DB-Sicherheitsgruppe erhalten, die nur mit Amazon RDS funktioniert.

Sie können Regeln zu einer DB-Sicherheitsgruppe hinzufügen, die den eingehenden Zugriff von EC2-Sicherheitsgruppen erlaubt. Sie können keine DB-Sicherheitsgruppe zur Auto Scaling-Gruppe Ihrer Umgebung hinzufügen. Wenn Sie vermeiden möchten, eine Abhängigkeit zwischen der DB-Sicherheitsgruppe und Ihrer Umgebung zu erstellen, müssen Sie eine dritte Sicherheitsgruppe in Amazon EC2 erstellen. Außerdem müssen Sie eine Regel in der DB-Sicherheitsgruppe hinzufügen, um eingehenden Zugriff auf die neue Sicherheitsgruppe zu erlauben, und sie dann der Auto Scaling-Gruppe in Ihrer Elastic Beanstalk-Umgebung zuweisen.

So starten Sie eine RDS-Instance in EC2-Classic (ohne VPC)

1. Öffnen Sie die [RDS-Managementkonsole](#).
2. Wählen Sie Create database (Datenbank erstellen) aus.
3. Führen Sie die Schritte im Assistenten aus. Beachten Sie die Werte, die Sie für die folgenden Optionen eingeben:
 - Master Username
 - Masterpasswort
4. Wenn Sie Configure advanced settings (Erweiterte Einstellungen konfigurieren) erreichen, wählen Sie in den Einstellungen für Network and Security (Netzwerk und Sicherheit) Folgendes aus:
 - VPC – **Not in VPC**. Wenn diese Option nicht verfügbar ist, unterstützt Ihr Konto **EC2 Classic** möglicherweise nicht oder Sie haben einen [Instance-Typ, der nur in VPC verfügbar ist](#).
 - Availability Zone (Verfügbarkeitszone) – **No Preference**
 - DB Security Group(s) (DB-Sicherheitsgruppe[n]) – **Create new Security Group**
5. Konfigurieren Sie die übrigen Optionen und wählen Sie Create database (Datenbank erstellen) aus. Beachten Sie die Werte, die Sie für die folgenden Optionen eingeben:
 - Datenbank-Name
 - Database Port (Datenbankport)

In EC2-Classic hat Ihre DB-Instance eine DB-Sicherheitsgruppe anstelle einer VPC-Sicherheitsgruppe. Sie können keine DB-Sicherheitsgruppe an Ihre Elastic Beanstalk-Umgebung anhängen, also müssen Sie eine neue Sicherheitsgruppe erstellen, die Sie für den Zugriff auf die DB-Instance autorisieren und an Ihre Umgebung anhängen können. Wir bezeichnen dies als eine Brücken-Sicherheitsgruppe und nennen sie **webapp-bridge**.

So erstellen Sie eine Brücken-Sicherheitsgruppe

1. Öffnen Sie die [Amazon EC2-Konsole](#).
2. Wählen Sie in der Navigationsleiste unter Network & Security (Netzwerk und Sicherheit) die Option Security Groups (Sicherheitsgruppen) aus.
3. Wählen Sie Create Security Group aus.
4. Geben Sie für Security group name (Name der Sicherheitsgruppe) folgenden Namen ein: **webapp-bridge**.
5. Geben Sie für Description (Beschreibung) den Text **Provide access to DB instance from Elastic Beanstalk environment instances**. ein.
6. Übernehmen Sie für VPC den ausgewählten Standardwert.
7. Wählen Sie Create (Erstellen) aus

Als Nächstes bearbeiten Sie die Sicherheitsgruppe für die DB-Instance, damit eingehender Datenverkehr von der Brücken-Sicherheitsgruppe zulässig ist.

So ändern Sie die Eingangsregeln für die Sicherheitsgruppe der RDS-Instance

1. Öffnen Sie die [Amazon RDS-Konsole](#).
2. Wählen Sie Databases (Datenbanken) aus.
3. Wählen Sie den Namen der DB-Instance, um deren Details anzuzeigen.
4. Im Abschnitt Connectivity (Konnektivität) wird in Security (Sicherheit) die mit der DB-Instance verknüpfte Sicherheitsgruppe angezeigt. Öffnen Sie den Link, um die Sicherheitsgruppe in der Amazon EC2-Konsole aufzurufen.
5. Legen Sie in den Details der Sicherheitsgruppe Connection Type (Verbindungstyp) auf EC2 Security Group (EC2-Sicherheitsgruppe) fest.
6. Legen Sie EC2 Security Group Name (EC2-Sicherheitsgruppenname) auf den Namen der Brücken-Sicherheitsgruppe fest, die Sie erstellt haben.
7. Klicken Sie auf Authorize.

Als Nächstes fügen Sie die Brücken-Sicherheitsgruppe zur ausgeführten Umgebung hinzu. Dieses Verfahren sieht vor, dass alle Instances in der Umgebung mit der zusätzlich zugeordneten Sicherheitsgruppe erneut bereitgestellt werden.

So fügen Sie eine Sicherheitsgruppe zur Umgebung hinzu

- Führen Sie eine der folgenden Aufgaben aus:
 - So fügen Sie eine Sicherheitsgruppe mit der Elastic Beanstalk-Konsole hinzu
 - a. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
 - b. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

- c. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
- d. Wählen Sie in der Konfigurationskategorie Instances die Option Edit (Bearbeiten).
- e. Wählen Sie unter EC2 security groups (EC2-Sicherheitsgruppen) neben der Instance-Sicherheitsgruppe, die Elastic Beanstalk erstellt, die Sicherheitsgruppe aus, die an die Instances angehängt werden soll.

- f. Wählen Sie Apply (Anwenden) aus.
- g. Lesen Sie die Warnmeldung und klicken Sie dann auf Bestätigen.
- Um eine Sicherheitsgruppe mit einer [Konfigurationsdatei \(p. 722\)](#) hinzuzufügen, verwenden Sie die Beispieldatei [securitygroup-addexisting.config](#).

Anschließend werden die Verbindungsinformationen mithilfe von Umgebungseigenschaften an die Umgebung übergeben. Wenn Sie mit der Elastic Beanstalk-Konsole [der Umgebung eine DB-Instance hinzufügen \(p. 609\)](#), verwendet Elastic Beanstalk Umgebungseigenschaften, wie RDS_HOSTNAME, um Verbindungsinformationen an die Anwendung zu übergeben. Sie können die gleichen Eigenschaften verwenden, mit denen Sie denselben Anwendungscode mit integrierten DB-Instances und externen DB-Instances verwenden können, oder Ihre eigenen Eigenschaftsnamen wählen.

So konfigurieren Sie Umgebungseigenschaften

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.
2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
4. Wählen Sie in der Software-Konfigurationskategorie Edit (Bearbeiten) aus.
5. Definieren Sie im Abschnitt Environment Properties (Umgebungseigenschaften) die Variablen, die Ihre Anwendung liest, um eine Verbindungszeichenfolge zu konstruieren. Für die Kompatibilität mit Umgebungen, die eine integrierte RDS-Instance haben, verwenden Sie Folgendes:
 - RDS_DB_NAME – Der DB Name, der in der Amazon RDS-Konsole angezeigt wird.
 - RDS_USERNAME – Der Master Username (Hauptbenutzername), den Sie eingeben, wenn Sie die Datenbank zu Ihrer Umgebung hinzufügen.
 - RDS_PASSWORD – Das Master Password (Master-Passwort), das Sie eingeben, wenn Sie die Datenbank zu Ihrer Umgebung hinzufügen.
 - RDS_HOSTNAME – Der Endpoint (Endpunkt) der DB-Instance, die in der Amazon RDS-Konsole gezeigt wird.
 - RDS_PORT – Der Port, der in der Amazon RDS-Konsole angezeigt wird.

Environment Properties

The following properties are passed into the application as environment variables. [Learn more.](#)

Property Name	Property Value
RDS_DB_NAME	ebdb
RDS_HOSTNAME	webapp-db.jxccb5mpan
RDS_PORT	5432
RDS_USERNAME	webapp-admin
RDS_PASSWORD	kUj5uKxmWDMYc403

[Cancel](#)

6. Wählen Sie Apply (Anwenden) aus.

Wenn Sie Ihre Anwendung noch nicht für das Programmieren von Umgebungseigenschaften und Erstellen einer Verbindungszeichenfolge programmiert haben, finden Sie Anweisungen in den folgenden sprachspezifischen Themen:

- Java SE – [Verbinden mit einer Datenbank \(Java SE-Plattformen\)](#) (p. 136)
- Java mit Tomcat – [Verbinden mit einer Datenbank \(Tomcat-Plattformen\)](#) (p. 137)
- Node.js – [Verbinden mit einer Datenbank](#) (p. 285)
- .NET – [Verbinden mit einer Datenbank](#) (p. 219)
- PHP – [Verbinden mit einer Datenbank mit einem PDO oder MySQLi](#) (p. 351)
- Python – [Verbinden mit einer Datenbank](#) (p. 380)
- Ruby – [Verbinden mit einer Datenbank](#) (p. 399)

Je nachdem, wann Ihre Anwendung Umgebungsvariablen liest, müssen Sie den Anwendungsserver auf den Instances in Ihrer Umgebung möglicherweise neu starten.

So starten Sie die App-Server der Umgebung neu

1. Öffnen Sie die [Elastic Beanstalk-Konsole](#) und wählen Sie in der Liste Regions (Regionen) Ihre AWS-Region aus.

2. Wählen Sie im Navigationsbereich Umgebungen aus und wählen Sie dann den Namen Ihrer Umgebung in der Liste aus.

Note

Wenn Sie viele Umgebungen haben, verwenden Sie die Suchleiste, um die Umgebungsliste zu filtern.

3. Wählen Sie Environment actions (Umgebungsaktionen) und dann Restart app server(s) (App-Server neu starten) aus.

Speichern der Verbindungszeichenfolge in Amazon S3

Die Angabe von Verbindungsinformationen für Ihre Anwendung mit Umgebungseigenschaften ist eine gute Möglichkeit, Passwörter aus Ihrem Code herauszuhalten, es ist aber nicht die perfekte Lösung. Umgebungseigenschaften sind in der [Environment Management Console \(p. 426\)](#) erkennbar und können von jedem Benutzer mit der Berechtigung zum [Beschreiben der Konfigurationseinstellungen](#) auf Ihrer Umgebung angezeigt werden. Abhängig von der Plattform können Umgebungseigenschaften auch in [Instance-Protokollen \(p. 869\)](#) angezeigt werden.

Sie können Ihre Verbindungsinformationen sperren, indem Sie sie in einem Amazon S3-Bucket speichern, den Sie kontrollieren. Die grundlegenden Schritte sind wie folgt:

- Laden Sie eine Datei mit der Verbindungszeichenfolge in einen Amazon S3-Bucket hoch.
- Gewähren Sie dem EC2 Instance-Profil die Berechtigung zum Lesen der Datei.
- Konfigurieren Sie Ihre Anwendung, um die Datei während der Bereitstellung herunterzuladen.
- Lesen Sie die Datei in Ihrem Anwendungscode.

Zunächst erstellen Sie einen Bucket zum Speichern der Datei, die Ihre Verbindungszeichenfolge enthält. In diesem Beispiel verwenden wir eine JSON-Datei mit einem einzelnen Schlüssel und Wert. Der Wert ist eine JDBC-Verbindungszeichenfolge für eine PostgreSQL-DB-Instance in Amazon RDS.

`beanstalk-database.json`

```
{  
    "connection": "jdbc:postgresql://mydb.b5uacpxznijm.us-west-2.rds.amazonaws.com:5432/ebdb?  
user=username&password=mypassword"  
}
```

Die markierten Teile der URL entsprechen dem Endpunkt, Port, DB-Namen, Benutzernamen und Passwort für die Datenbank.

So erstellen Sie einen Bucket und laden eine Datei hoch

1. Öffnen Sie die [Amazon S3-Konsole](#).
2. Klicken Sie auf Create Bucket.
3. Geben Sie einen Bucket Name (Bucket-Namen) ein und wählen Sie eine Region aus.
4. Wählen Sie Create aus.
5. Öffnen Sie den Bucket und wählen Sie Upload (Hochladen) aus.
6. Folgen Sie den Anweisungen, um die Datei hochzuladen.

Standardmäßig besitzt Ihr Konto die Datei und die Berechtigung, sie zu verwalten. IAM-Benutzer und -Rollen haben diese jedoch nicht, es sei denn, Sie gewähren den Zugriff explizit. Gewähren Sie die Instances in Ihrer Elastic Beanstalk-Umgebung, indem Sie dem [Instance Profile \(Instance-Profil\) \(p. 22\)](#) eine Richtlinie hinzufügen.

Das Standardprofil der Instance hat den Namen `aws-elasticbeanstalk-ec2-role`. Wenn Sie nicht sicher sind, wie der Name Ihres Instance-Profils lautet, finden Sie ihn auf der Seite Configuration (Konfiguration) in der [Environment Management Console \(p. 531\)](#).

So fügen Sie Berechtigungen zum Instance-Profil hinzu

1. Öffnen Sie die [IAM-Konsole](#).
2. Wählen Sie Roles.
3. Wählen Sie `aws-elasticbeanstalk-ec2-role` aus.
4. Wählen Sie Add inline Policy (Inline-Richtlinie auswählen) aus.
5. Fügen Sie eine Richtlinie hinzu, mit der die Instance die Datei abrufen kann.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "database",  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET-123456789012/beanstalk-database.json"  
            ]  
        }  
    ]  
}
```

Ersetzen Sie die Bucket- und Objektnamen mit den Namen des Buckets und Objekts.

Als Nächstes fügen Sie eine [Konfigurationsdatei \(p. 722\)](#) zu Ihrem Quellcode hinzu, der Elastic Beanstalk darüber informiert, die Datei während der Bereitstellung von Amazon S3 herunterzuladen.

`~/my-app/.ebextensions/database.config`

```
Resources:  
  AWSEBAutoScalingGroup:  
    Metadata:  
      AWS::CloudFormation::Authentication:  
        S3Auth:  
          type: "s3"  
          buckets: ["DOC-EXAMPLE-BUCKET-123456789012"]  
          roleName: "aws-elasticbeanstalk-ec2-role"  
  
    files:  
      "/tmp/beanstalk-database.json" :  
        mode: "000644"  
        owner: root  
        group: root  
        authentication: "S3Auth"  
        source: https://s3-us-west-2.amazonaws.com/DOC-EXAMPLE-BUCKET-123456789012/beanstalk-database.json
```

Diese Konfigurationsdatei macht zwei Dinge. Der `Resources`-Schlüssel fügt eine Authentifizierungsmethode zu den Auto Scaling-Gruppen-Metadaten der Umgebung hinzu, die Elastic Beanstalk für den Zugriff auf Amazon S3 verwenden kann. Der `files`-Schlüssel informiert Elastic Beanstalk, die Datei aus Amazon S3 herunterzuladen und sie während der Bereitstellung lokal in `/tmp/` zu speichern.

Stellen Sie Ihre Anwendung mit der Konfigurationsdatei im `.ebextensions`-Ordner auf der Stammebene Ihres Quellcodes bereit. Wenn Sie die Berechtigungen korrekt konfiguriert haben, ist die Bereitstellung erfolgreich und die Datei wird auf alle Instances in Ihrer Umgebung heruntergeladen. Andernfalls schlägt die Bereitstellung fehl.

Fügen Sie Code zu Ihrer Anwendung hinzu, um die JSON-Datei zu lesen, und verwenden Sie die Verbindungszeichenfolge, um eine Verbindung mit der Datenbank herzustellen.

Bereinigen einer externen Amazon RDS-Instance

Wenn Sie eine externe Amazon RDS-Instance mit Ihrer Elastic Beanstalk-Umgebung verbinden, ist die Datenbank-Instance nicht an den Lebenszyklus Ihrer Umgebung gebunden und wird nicht gelöscht, wenn Sie Ihre Umgebung beenden. Um sicherzustellen, dass personenbezogene Informationen, die Sie möglicherweise in der Datenbankinstanz gespeichert haben, nicht unnötig beibehalten werden, löschen Sie nicht mehr benötigte Datensätze oder löschen die Datenbank-Instance.

Verwenden von Elastic Beanstalk mit Amazon S3

Amazon Simple Storage Service (Amazon S3) bietet eine langfristige, fehlertolerante Datenspeicherung.

Elastic Beanstalk erstellt einen Amazon S3-Bucket mit dem Namen `elasticbeanstalk-region-account-id` für jede Region, in der Sie Umgebungen erstellen. Elastic Beanstalk verwendet diesen Bucket zum Speichern von Objekten, beispielsweise von temporären Konfigurationsdateien, die für die ordnungsgemäße Ausführung Ihrer Anwendung erforderlich sind.

Elastic Beanstalk aktiviert nicht die Standardverschlüsselung für den erstellten Amazon S3-Bucket. Das bedeutet, dass Objekte standardmäßig unverschlüsselt im Bucket gespeichert werden (und nur autorisierte Benutzer auf sie zugreifen können). Einige Anwendungen erfordern die Verschlüsselung von allen Objekten, wenn sie – auf einer Festplatte, in einer Datenbank usw. – gespeichert werden (auch bekannt als Verschlüsselung im Ruhezustand). Wenn diese Anforderung zutrifft, können Sie die Buckets in Ihrem Konto für die Standardverschlüsselung konfigurieren. Weitere Informationen finden Sie unter [Amazon S3-Standardverschlüsselung für S3-Buckets](#) im Amazon Simple Storage Service-Entwicklerhandbuch.

Inhalt des Elastic Beanstalk Amazon S3-Buckets

In der folgenden Tabelle sind einige Objekte aufgelistet, die Elastic Beanstalk in Ihrem `elasticbeanstalk-*` Amazon S3-Bucket speichert. Die Tabelle zeigt auch, welche Objekte manuell gelöscht werden müssen. Um unnötige Speicherkosten zu vermeiden und sicherzustellen, dass personenbezogene Daten nicht gespeichert werden, sollten Sie diese Objekte manuell löschen, wenn Sie sie nicht mehr benötigen.

Objekt	Wann wird gespeichert?	Wann gelöscht?
Anwendungsver-	Wenn Sie eine Umgebung erstellen oder Ihren Anwendungscode in eine bestehende Umgebung implementieren, speichert Elastic Beanstalk eine Anwendungsversion in Amazon S3 und ordnet sie der Umgebung zu.	Beim Löschen der Anwendung und gemäß Lebenszyklus der Version (p. 408) .
Quell- Bundles (p. 406)	Wenn Sie eine neue Anwendungsversion über die Elastic Beanstalk-Konsole oder die EB CLI hochladen, speichert Elastic Beanstalk eine Kopie davon in Amazon S3 und legt sie als Quell-Bundle Ihrer Umgebung fest.	Manuell. Wenn Sie eine Anwendungsversion löschen, können Sie Delete versions from Amazon S3 (Versionen von Amazon S3 löschen) auswählen, um auch das zugehörige Quell-Bundle zu löschen. Details

Objekt	Wann wird gespeichert?	Wann gelöscht?
		dazu finden Sie unter Verwalten von Anwendungsversionen (p. 406) .
Custom platforms (p. 1117)	Wenn Sie eine benutzerdefinierte Plattform erstellen, speichert Elastic Beanstalk die zugehörigen Daten vorübergehend in Amazon S3.	Nach erfolgreichem Abschluss der Erstellung der benutzerdefinierten Plattform.
Protokolldateien (p. 169)	Sie können Elastic Beanstalk auffordern, Instance-Protokolldateien (Protokollfragmente oder Bundle-Protokolle) abzurufen und in Amazon S3 zu speichern. Sie können auch die Protokollrotation aktivieren und Ihre Umgebung so konfigurieren, dass Protokolle nach der Rotation automatisch in Amazon S3 veröffentlicht werden.	Protokollfragmente und Bundle-Protokolle: 15 Minuten nach ihrer Erstellung. Rotierte Protokolle: Manuell.
Gespeicherte Konfigurationen (p. 764)	Manuell.	Manuell.

Löschen von Objekten im Elastic Beanstalk Amazon S3-Bucket

Wenn Sie eine Umgebung beenden oder eine Anwendung löschen, löscht Elastic Beanstalk die meisten zugehörigen Objekte aus Amazon S3. Um die Speicherkosten einer laufenden Anwendung zu minimieren, löschen Sie regelmäßig Objekte, die Ihre Anwendung nicht benötigt. Achten Sie außerdem auf Objekte, die Sie manuell löschen müssen, wie unter [Inhalt des Elastic Beanstalk Amazon S3-Buckets \(p. 985\)](#) aufgelistet. Um sicherzustellen, dass private Informationen nicht unnötig beibehalten werden, löschen Sie diese Objekte, wenn Sie sie nicht mehr benötigen.

- Löschen Sie Anwendungsversionen, von denen Sie nicht mehr erwarten, dass sie in Ihrer Anwendung verwendet werden. Wenn Sie eine Anwendungsversion löschen, können Sie [Delete versions from Amazon S3 \(Versionen aus Amazon S3 löschen\)](#) auswählen, um auch das zugehörige Quellcode-Bundle zu löschen – eine Kopie des Quellcodes und der Konfigurationsdateien Ihrer Anwendung, die Elastic Beanstalk nach Amazon S3 hochgeladen hat, wenn Sie eine Anwendung bereitgestellt oder eine Anwendungsversion hochgeladen haben. Weitere Informationen zum Löschen einer Anwendungsversion finden Sie unter [Verwalten von Anwendungsversionen \(p. 406\)](#).
- Löschen Sie rotierte Protokolle, die Sie nicht benötigen. Alternativ können Sie sie herunterladen oder zur weiteren Analyse nach Amazon S3 Glacier verschieben.
- Löschen Sie gespeicherte Konfigurationen, die Sie in keiner Umgebung mehr verwenden werden.

Löschen des Elastic Beanstalk Amazon S3-Buckets

Elastic Beanstalk wendet eine Bucket-Richtlinie auf Buckets an, die erstellt werden, damit Umgebungen in den Bucket schreiben können und ein versehentliches Löschen verhindern. Wenn Sie einen Bucket löschen müssen, den Elastic Beanstalk erstellt hat, löschen Sie zuerst die Bucket-Richtlinie aus dem Abschnitt Permissions (Berechtigungen) der Bucket-Eigenschaften in der Amazon S3-Konsole.

Warnung

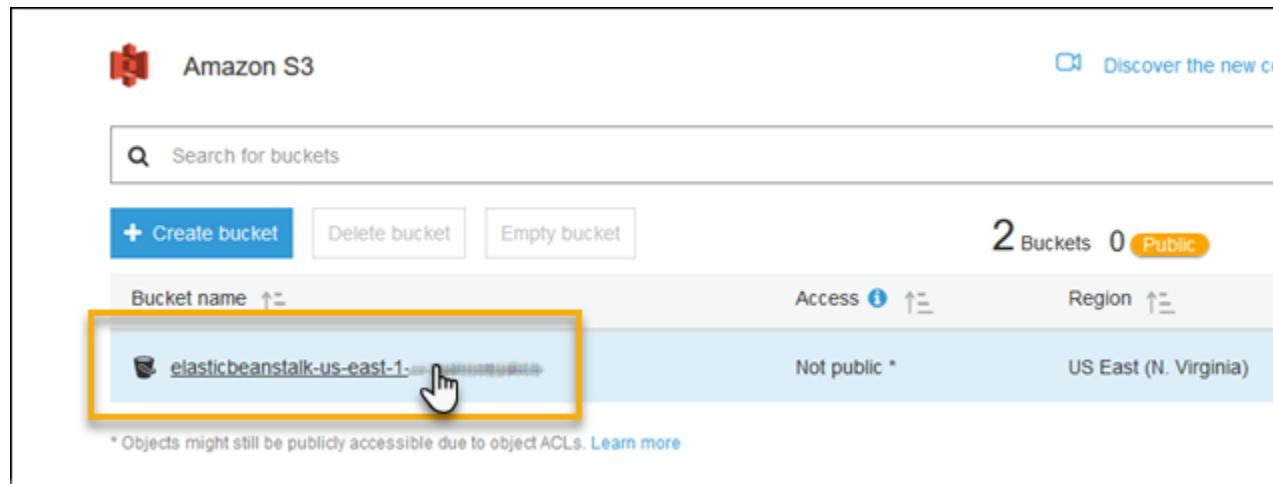
Wenn Sie einen Bucket löschen, den Elastic Beanstalk in Ihrem Konto erstellt hat, und Sie noch vorhandene Anwendungen und laufende Umgebungen in der entsprechenden Region haben, funktionieren Ihre Anwendungen möglicherweise nicht mehr richtig. Zum Beispiel:

- Wenn eine Umgebung skaliert wird, sollte Elastic Beanstalk in der Lage sein, die Anwendungsversion der Umgebung im Amazon S3-Bucket zu finden und sie zum Starten neuer Amazon EC2-Instances zu verwenden.
- Wenn Sie eine benutzerdefinierte Plattform erstellen, verwendet Elastic Beanstalk während des Erstellungsprozesses eine temporäre Amazon S3-Speicherung.

Wir empfehlen, dass Sie bestimmte unnötige Objekte aus Ihrem Elastic Beanstalk Amazon S3-Bucket löschen, statt den gesamten Bucket löschen.

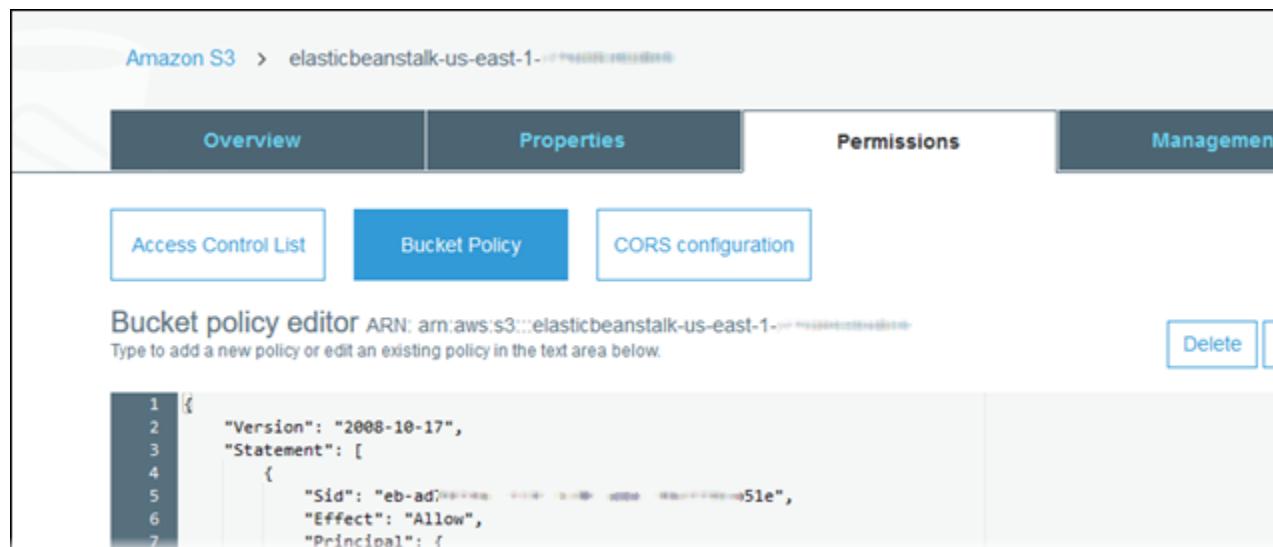
So löschen Sie einen Elastic Beanstalk-Speicher-Bucket (Konsole)

1. Öffnen Sie die [Amazon S3-Konsole](#).
2. Öffnen Sie die Seite des Elastic Beanstalk-Speicher-Buckets, indem Sie den Bucket-Namen auswählen.



The screenshot shows the Amazon S3 console interface. At the top, there's a search bar labeled "Search for buckets" and three buttons: "+ Create bucket", "Delete bucket", and "Empty bucket". To the right, it displays "2 Buckets" and "0 Public". Below the buttons, a table lists buckets. The first row shows a thumbnail, the bucket name "elasticbeanstalk-us-east-1", its access status "Not public", and its region "US East (N. Virginia)". A yellow box highlights the bucket name "elasticbeanstalk-us-east-1", and a cursor arrow points to the end of the bucket name. A small note at the bottom says "* Objects might still be publicly accessible due to object ACLs. [Learn more](#)".

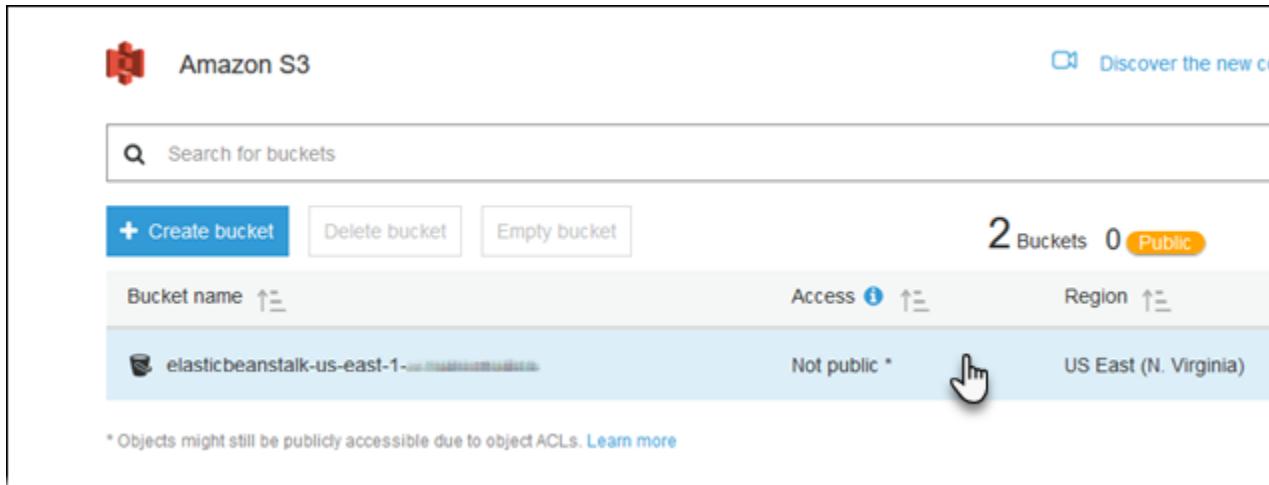
3. Wählen Sie die Registerkarte Permissions aus.
4. Wählen Sie Bucket Policy aus.
5. Wählen Sie Delete.



The screenshot shows the "Properties" tab of the Amazon S3 bucket "elasticbeanstalk-us-east-1". Below the tabs, there are three buttons: "Access Control List", "Bucket Policy" (which is highlighted), and "CORS configuration". Under "Bucket policy editor", there is a text area containing a JSON policy. The policy starts with "Version": "2008-10-17", followed by a "Statement" array containing one item. Item 1 has "Sid": "eb-ad", "Effect": "Allow", and "Principal": "*". On the far right of the policy editor, there is a blue "Delete" button.

```
1 {
2     "Version": "2008-10-17",
3     "Statement": [
4         {
5             "Sid": "eb-ad",
6             "Effect": "Allow",
7             "Principal": "*"
8         }
9     ]
10 }
```

6. Gehen Sie zurück zur Hauptseite der Amazon S3-Konsole und wählen Sie dann den Elastic Beanstalk-Speicher-Bucket aus, indem Sie auf dessen Zeile außerhalb des Bucket-Namens klicken.



7. Wählen Sie Delete Bucket (Bucket löschen).
8. Geben Sie den Namen des Buckets ein und klicken Sie dann auf Confirm (Bestätigen).

Verwenden von Elastic Beanstalk mit Amazon VPC

Sie können eine [Amazon Virtual Private Cloud](#) (Amazon VPC) verwenden, um ein sicheres Netzwerk für Ihre Elastic Beanstalk-Anwendung und zugehörige AWS-Ressourcen zu erstellen. Beim Erstellen Ihrer Umgebung wählen Sie VPC, Subnetze und Sicherheitsgruppen für Ihre Anwendungs-Instances und Load Balancer aus. Sie können die VPC-Konfiguration beliebig festlegen, solange sie die folgenden Voraussetzungen erfüllt.

VPC-Anforderungen

- Internetzugriff – Instances müssen über eine der folgenden Methoden auf das Internet zugreifen können:
 - Öffentliches Subnetz – Instances besitzen eine öffentliche IP-Adresse und verwenden für den Zugriff auf das Internet ein Internet-Gateway.
 - Privates Subnetz – Instances verwenden für den Zugriff auf das Internet ein NAT-Gerät.

Note

Wenn Sie in Ihrer VPC [VPC-Endpunkte \(p. 1002\)](#) für die Herstellung von Verbindungen mit `elasticbeanstalk` und `elasticbeanstalk-healthd` konfigurieren, ist der Internetzugriff optional und nur erforderlich, wenn die Anwendung sie spezifisch benötigt. Wenn keine VPC-Endpunkte vorhanden sind, muss Ihre VPC Zugriff auf das Internet haben.
Die Standard-VPC, die Elastic Beanstalk für Sie einrichtet, stellt einen Internetzugang bereit.

Elastic Beanstalk unterstützt keine Proxy-Einstellungen wie `HTTPS_PROXY` für die Konfiguration eines Web-Proxys.

- NTP – Instances in Ihrer Elastic Beanstalk-Umgebung verwenden das Network Time Protocol (NTP), um die Systemuhr zu synchronisieren. Wenn Instances nicht auf UDP-Port 123 kommunizieren können, ist die Uhr möglicherweise nicht synchron, wodurch Probleme mit den Elastic Beanstalk-Integritätsberichten verursacht werden. Stellen Sie sicher, dass Ihre VPC-Sicherheitsgruppen und Netzwerk-ACLs ein- und ausgehenden UDP-Datenverkehr auf Port 123 zulassen, um diese Probleme zu verhindern.

Das Repository [elastic-beanstalk-samples](#) bietet AWS CloudFormation-Vorlagen, mit denen Sie eine VPC für die Verwendung mit Ihren Elastic Beanstalk-Umgebungen erstellen können.

So erstellen Sie Ressourcen mit einer AWS CloudFormation-Vorlage

1. Klonen Sie das Beispiel-Repository oder laden Sie über die Links in der [README](#)-Datei eine Vorlage herunter.
2. Öffnen Sie die [AWS CloudFormation-Konsole](#).
3. Wählen Sie Create stack aus.
4. Wählen Sie Upload a template to Amazon S3 (Vorlage auf Amazon S3 hochladen) aus.
5. Wählen Sie Upload file (Datei hochladen) aus und laden Sie die Vorlagendatei von Ihrem lokalen Computer hoch.
6. Klicken Sie auf Next (Weiter) und befolgen Sie die Anweisungen zum Erstellen eines Stapels mit den Ressourcen in der Vorlage.

Nach Abschluss der Stack-Erstellung werden auf der Registerkarte Outputs (Ausgaben) die VPC-ID und die Subnetz-IDs angezeigt. Verwenden Sie diese zum Konfigurieren der VPC in der neuen [Netzwerkkonfigurationskategorie \(p. 452\)](#) des Umgebungsassistenten.

Themen

- [Öffentliche VPC \(p. 989\)](#)
- [Öffentliche/private VPC \(p. 990\)](#)
- [Private VPC \(p. 990\)](#)
- [Beispiel: Starten einer Elastic Beanstalk-Anwendung in einer VPC mit Bastion-Hosts \(p. 992\)](#)
- [Beispiel: Starten einer Elastic Beanstalk-Anwendung in einer VPC mit Amazon RDS \(p. 997\)](#)
- [Verwenden von Elastic Beanstalk mit VPC-Endpunkten \(p. 1002\)](#)

Öffentliche VPC

AWS CloudFormation-Vorlage – [vpc-public.yaml](#)

Einstellungen (mit Load Balancing)

- Load balancer visibility (Load Balancer-Sichtbarkeit) – Öffentlich
- Load balancer subnets (Load Balancer-Subnetze) – Beide öffentliche Subnetze
- Instance public IP (Öffentliche IP der Instance) – Aktiviert
- Instance subnets (Instance-Subnetze) – Beide öffentliche Subnetze
- Instance security groups (Instance-Sicherheitsgruppen) – Standard-Sicherheitsgruppe hinzufügen

Einstellungen (einzelne Instance)

- Instance subnets (Instance-Subnetze) – Eines der öffentlichen Subnetze
- Instance security groups (Instance-Sicherheitsgruppen) – Standard-Sicherheitsgruppe hinzufügen

Ein einfaches rein öffentliches VPC-Layout umfasst mindestens ein öffentliches Subnetz, ein Internet-Gateway und eine Standardsicherheitsgruppe, die den Datenverkehr zwischen den Ressourcen in der VPC zulässt. Wenn Sie in der VPC eine Umgebung erstellen, generiert Elastic Beanstalk zusätzliche Ressourcen, die sich je nach Typ der Umgebung unterscheiden.

VPC-Ressourcen

- Einzelne Instance – Elastic Beanstalk erstellt eine Sicherheitsgruppe für die Anwendungs-Instance, die Datenverkehr aus dem Internet auf Port 80 zulässt, und weist der Instance eine Elastic IP-Adresse zu,

damit sie eine öffentliche IP-Adresse hat. Der Domänenname der Umgebung wird zur öffentlichen IP-Adresse der Instance aufgelöst.

- Lastausgleich – Elastic Beanstalk erstellt eine Sicherheitsgruppe für den Load Balancer, der Datenverkehr aus dem Internet auf Port 80 zulässt, und eine Sicherheitsgruppe für die Anwendungs-Instances, die Datenverkehr aus der Sicherheitsgruppe für den Load Balancer zulassen. Der Domänenname der Umgebung wird zum öffentlichen Domänenamen für den Load Balancer aufgelöst.

Dies ähnelt der Art und Weise, wie Elastic Beanstalk die Netzwerke verwaltet, wenn Sie die Standard-VPC verwenden. Die Sicherheit in einem öffentlichen Subnetz hängt vom Load Balancer und von den Instance-Sicherheitsgruppen ab, die von Elastic Beanstalk erstellt wurden. Dies ist die kostengünstigste Konfiguration, da kein NAT-Gateway erforderlich ist.

Öffentliche/private VPC

AWS CloudFormation-Vorlage – [vpc-privatepublic.yaml](#)

Einstellungen (mit Load Balancing)

- Load balancer visibility (Load Balancer-Sichtbarkeit) – Öffentlich
- Load balancer subnets (Load Balancer-Subnetze) – Beide öffentliche Subnetze
- Instance public IP (Öffentliche IP der Instance) – Deaktiviert
- Instance subnets (Instance-Subnetze) – Beide private Subnetze
- Instance security groups (Instance-Sicherheitsgruppen) – Standard-Sicherheitsgruppe hinzufügen

Fügen Sie Ihrer VPC zur Erhöhung der Sicherheit private Subnetze hinzu, um ein öffentlich/privat-Layout zu erstellen. Dieses Layout erfordert einen Load Balancer und ein NAT-Gateway in den öffentlichen Subnetzen. Es bietet Ihnen die Möglichkeit, Anwendungs-Instances, Datenbanken und andere Ressourcen in privaten Subnetzen auszuführen. Instances in privaten Subnetzen können nur über den Load Balancer und das NAT-Gateway mit dem Internet kommunizieren.

Private VPC

AWS CloudFormation-Vorlage – [vpc-private.yaml](#)

Einstellungen (mit Load Balancing)

- Load balancer visibility (Load Balancer-Sichtbarkeit) – Privat
- Load balancer subnets (Load Balancer-Subnetze) – Beide private Subnetze
- Instance public IP (Öffentliche IP der Instance) – Deaktiviert
- Instance subnets (Instance-Subnetze) – Beide private Subnetze
- Instance security groups (Instance-Sicherheitsgruppen) – Standard-Sicherheitsgruppe hinzufügen

Im Fall interner Anwendungen, auf die kein Zugriff aus dem Internet möglich sein darf, können Sie alle Vorgänge in privaten Subnetzen ausführen und den Load Balancer so konfigurieren, dass er nur intern verfügbar ist. Hierzu ändern Sie den Wert für Load balancer visibility (Load Balancer-Sichtbarkeit) in Internal (Intern). Diese Vorlage erstellt eine VPC ohne öffentliche Subnetze und ohne Internet-Gateway. Verwenden Sie dieses Layout für Anwendungen, auf die der Zugriff nur über dieselbe VPC oder ein zugehöriges VPN möglich sein soll.

Ausführung einer Elastic Beanstalk-Umgebung in einer privaten VPC

Wenn Sie Ihre Elastic Beanstalk-Umgebung in einer privaten VPC erstellen, hat die Umgebung keinen Zugriff auf das Internet. Ihre Anwendung benötigt möglicherweise Zugriff auf den Elastic Beanstalk-Service oder andere Services. Ihre Umgebung könnte z. B. erweiterte Integritätsberichte verwenden. In diesem Fall senden die Umgebungs-Instances Integritätsdaten an den erweiterten Integritätsservice. Der Elastic Beanstalk-Code auf den Umgebungs-Instances sendet Datenverkehr an andere AWS-Services und weiteren Datenverkehr an andere als AWS-Endpunkte (um beispielsweise Abhängigkeitspakete für Ihre Anwendung herunterzuladen). Im Folgenden finden Sie einige Schritte, die Sie in diesem Fall möglicherweise ausführen müssen, um sicherzustellen, dass Ihre Umgebung ordnungsgemäß funktioniert.

- Konfigurieren von VPC-Endpunkten für Elastic Beanstalk – Elastic Beanstalk und der erweiterte Integritätsservice unterstützen VPC-Endpunkte. Diese stellen sicher, dass der Datenverkehr zu diesen Services ausschließlich innerhalb des Amazon-Netzwerks ausgeführt wird und keinen Internetzugriff erfordert. Weitere Informationen finden Sie unter [the section called “VPC-Endpunkte” \(p. 1002\)](#).
- Konfigurieren von VPC-Endpunkten für zusätzliche Services – Elastic Beanstalk-Instances senden in Ihrem Namen Datenverkehr an mehrere andere AWS-Services: Amazon Simple Storage Service (Amazon S3), Amazon Simple Queue Service (Amazon SQS), AWS CloudFormation und Amazon CloudWatch Logs. Sie müssen auch für diese Services VPC-Endpunkte konfigurieren. Detaillierte Informationen zu VPC-Endpunkten einschließlich Links pro Service finden Sie unter [VPC-Endpunkte](#) im Amazon VPC-Benutzerhandbuch.

Note

Einige AWS-Services einschließlich Elastic Beanstalk unterstützen VPC-Endpunkte in einer begrenzten Anzahl von AWS-Regionen. Überprüfen Sie daher während der Erstellung Ihrer privaten VPC-Lösung, ob Elastic Beanstalk und die anderen hier genannten abhängigen Services VPC-Endpunkte in der von Ihnen ausgewählten AWS-Region unterstützen.

- Bereitstellen eines privaten Docker-Image – In einer [Docker \(p. 46\)](#)-Umgebung könnte Code auf den Umgebungs-Instances versuchen, während der Erstellung der Umgebung das konfigurierte Docker-Image aus dem Internet abzurufen, und fehlschlagen. Um diesen Fehler zu vermeiden, [erstellen Sie in Ihrer Umgebung ein benutzerdefiniertes Docker-Image \(p. 57\)](#) oder verwenden ein in [Amazon Elastic Container Registry](#) (Amazon ECR) gespeichertes Docker-Image und [konfigurieren einen VPC-Endpunkt für den Amazon ECR-Service](#).
- Aktivieren von DNS-Namen – Elastic Beanstalk-Code auf Umgebungs-Instances sendet an alle AWS-Services Datenverkehr unter Verwendung ihrer öffentlichen Endpunkte. Damit dieser Datenverkehr das Ziel erreicht, wählen Sie während der Konfigurierung der VPC-Schnittstellenendpunkte die Option **Enable DNS name (DNS-Name aktivieren)** aus. Hierdurch wird ein DNS-Eintrag in Ihrer VPC hinzugefügt, der den Endpunkt für den öffentlichen Service dem VPC-Schnittstellenendpunkt zuordnet.

Important

Wenn Ihre VPC nicht privat ist, Zugang zum öffentlichen Internet hat und **Enable DNS name (DNS-Name aktivieren)** für einen VPC-Endpunkt deaktiviert ist, wird der Datenverkehr zum jeweiligen Dienst über das öffentliche Internet transportiert. Das wird wahrscheinlich nicht von Ihnen gewünscht. Sie können dieses Problem bei einer privaten VPC leicht erkennen, da dies verhindert, dass der Datenverkehr das Ziel erreicht, und Ihnen Fehler angezeigt werden. Bei einer öffentlichen VPC erhalten Sie jedoch keine Hinweise.

- Einschluss von Anwendungsabhängigkeiten – Wenn Ihre Anwendung Abhängigkeiten wie Sprachlaufzeitpakete besitzt, versucht sie möglicherweise, diese während der Erstellung der Umgebung aus dem Internet herunterzuladen und zu installieren, und schlägt fehl. Um diesen Fehler zu vermeiden, sollten Sie alle Abhängigkeitspakete in das Quellpaket Ihrer Anwendung einschließen.
- Verwenden einer aktuellen Plattformversion – Stellen Sie sicher, dass Ihre Umgebung eine Plattformversion verwendet, die am 24. Februar 2020 oder später veröffentlicht wurde. Sie sollten

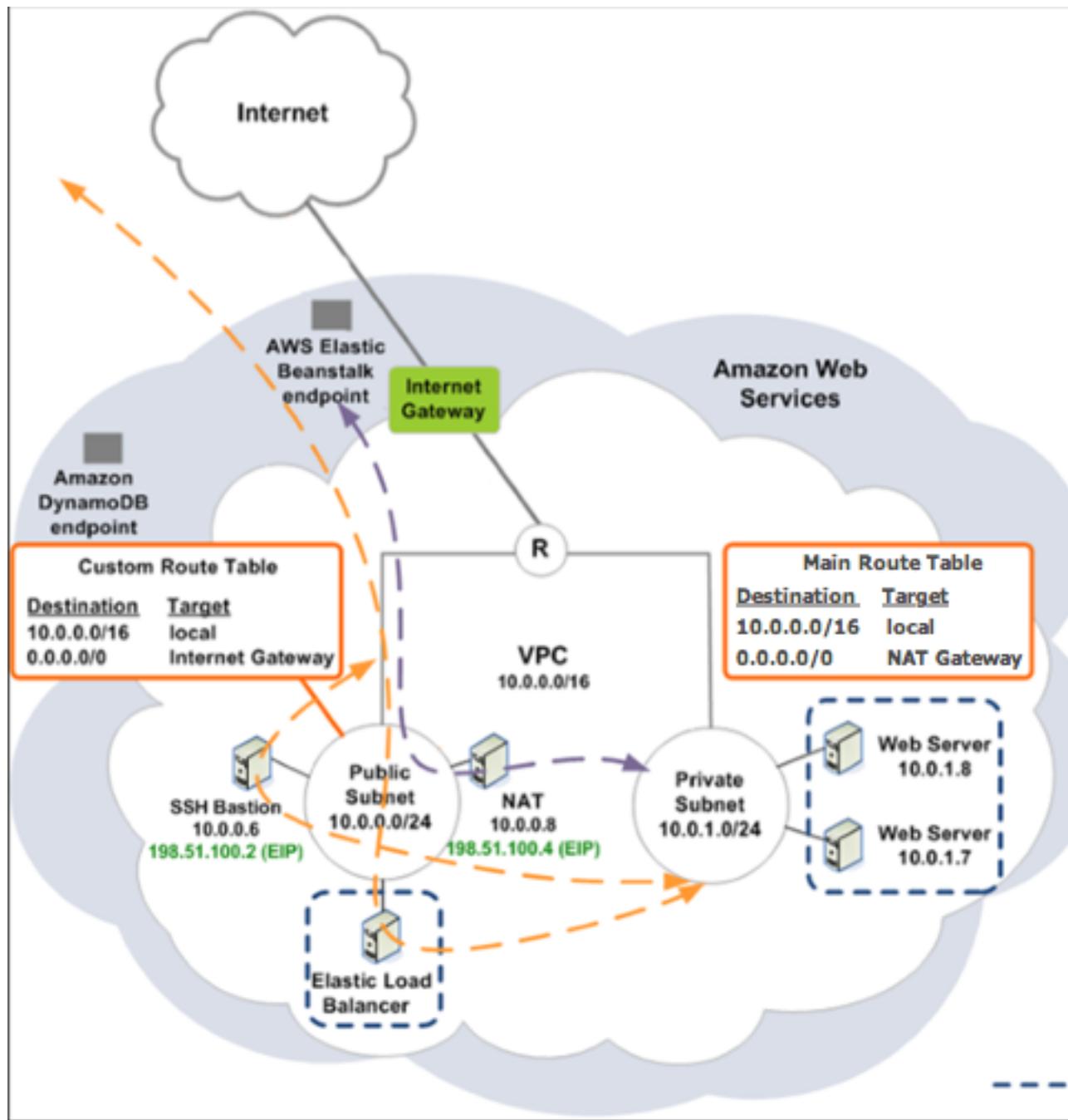
insbesondere eine Plattformversion verwenden, die mit oder nach einem der folgenden beiden Updates veröffentlicht wurde: [Linux Update 2020-02-28](#), [Windows Update 2020-02-24](#).

Note

Eine aktuelle Plattformversion ist deswegen notwendig, weil in älteren Versionen ein Problem auftrat, das verhinderte, dass mit der Option Enable DNS name (DNS-Name aktivieren) erstellte DNS-Einträge für Amazon SQS ordnungsgemäß funktionieren.

Beispiel: Starten einer Elastic Beanstalk-Anwendung in einer VPC mit Bastion-Hosts

Wenn sich Ihre Amazon EC2-Instances in einem privaten Subnetz befinden, können Sie keine Remote-Verbindung zu ihnen herstellen. Zum Herstellen einer Verbindung können Sie Bastion-Server im öffentlichen Subnetz herstellen, die als Proxys fungieren. Beispiel: Sie können SSH-Port-Weiterleitungen oder RDP-Gateways im öffentlichen Subnetz einrichten, um für den Datenverkehr an Ihre Datenbankserver aus Ihrem eigenen Netzwerk einen Proxy bereitzustellen. In diesem Abschnitt finden Sie ein Beispiel dafür, wie Sie eine VPC mit einem privaten und öffentlichen Subnetz erstellen. Die Instances befinden sich im privaten Subnetz und Bastion-Host, NAT-Gateway und Load Balancer befinden sich im öffentlichen Subnetz. Ihre Infrastruktur sollte dann wie das folgende Diagramm aussehen:



Um eine Elastic Beanstalk-Anwendung in einer VPC mithilfe eines Bastion-Host bereitzustellen, führen Sie die in den folgenden Unterabschnitten beschriebenen Schritte aus.

Schritte

- Erstellen einer VPC mit einem öffentlichen und privaten Subnetz (p. 994)
- Erstellen und Konfigurieren der Bastion-Host-Sicherheitsgruppe (p. 994)
- Aktualisieren der Instance-Sicherheitsgruppe (p. 996)
- Erstellen eines Bastion-Host (p. 996)

Erstellen einer VPC mit einem öffentlichen und privaten Subnetz

Führen Sie alle Verfahren in [Öffentliche/private VPC \(p. 990\)](#) durch. Wenn Sie die Anwendung bereitstellen, müssen Sie ein Amazon EC2-Schlüsselpaar für die Instances angeben, sodass Sie eine Remote-Verbindung herstellen können. Weitere Informationen darüber, wie Sie das Instance-Schlüsselpaar angeben können, finden Sie unter [Die Amazon EC2-Instances Ihrer Elastic Beanstalk-Umgebung \(p. 531\)](#).

Erstellen und Konfigurieren der Bastion-Host-Sicherheitsgruppe

Erstellen Sie eine Sicherheitsgruppe für den Bastion-Host und fügen Sie Regeln hinzu, mit denen eingehender SSH-Datenverkehr aus dem Internet und ausgehender SSH-Datenverkehr an das private Subnetz mit den Amazon EC2-Instances zulässig ist.

So erstellen Sie die Bastion-Host-Sicherheitsgruppe

1. Öffnen Sie die Amazon VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie im Navigationsbereich Security Groups aus.
3. Wählen Sie Create Security Group aus.
4. Geben Sie im Dialogfeld Create Security Group (Sicherheitsgruppe erstellen) Folgendes ein und wählen Sie Yes, Create (Ja, erstellen) aus.

Name tag (Namens-Tag) (optional)

Geben Sie einen Namens-Tag für die Sicherheitsgruppe ein.

Group name (Gruppenname)

Geben Sie den Namens der Sicherheitsgruppe ein.

Beschreibung

Geben Sie eine Beschreibung für die Sicherheitsgruppe ein.

VPC

Wählen Sie Ihre VPC aus.

Die Sicherheitsgruppe wird erstellt und auf der Seite Security Groups (Sicherheitsgruppen) angezeigt. Die Gruppe ist mit einer ID versehen (z. B. sg-xxxxxxxx). Möglicherweise müssen Sie die Spalte Group ID (Gruppen-ID) aktivieren, indem Sie oben rechts auf der Seite auf Show/Hide (Einblenden/Ausblenden) klicken.

Konfigurieren der Bastion-Host-Sicherheitsgruppe

1. Aktivieren Sie in der Liste der Sicherheitsgruppen das Kontrollkästchen für die Sicherheitsgruppe, die Sie gerade für Ihren Bastion-Host erstellt haben.
2. Wählen Sie auf der Registerkarte Inbound Rules die Option Edit aus.
3. Wählen Sie bei Bedarf Add another rule (Weitere Regel hinzufügen) aus.
4. Wenn der Bastion-Host eine Linux-Instance ist, wählen Sie unter Type (Typ) die Option SSH aus.

Wenn der Bastion-Host eine Windows-Instance ist, wählen Sie unter Type (Typ) die Option RDP aus.

5. Geben Sie den gewünschten Quell-CIDR-Bereich im Feld Source (Quelle) ein und wählen Sie Save (Speichern) aus.

Type	Protocol	Port Range	Source	Remove
SSH	TCP (6)	22	0.0.0.0/0	

6. Wählen Sie auf der Registerkarte Outbound Rules (Ausgehende Regeln) die Option Edit (Bearbeiten) aus.
7. Wählen Sie bei Bedarf Add another rule (Weitere Regel hinzufügen) aus.
8. Wählen Sie unter Type (Typ) den für die eingehende Regel angegebenen Typ aus.
9. Geben Sie in das Feld Source (Quelle) den CIDR-Bereich des Subnetzes der Hosts im privaten Subnetz der VPC ein.

So suchen Sie:

- a. Öffnen Sie die Amazon VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
- b. Wählen Sie im Navigationsbereich Subnets aus.
- c. Beachten Sie den Wert unter IPv4 CIDR für jede Availability Zone, in der sich die Hosts befinden, zu denen der Bastion-Host weiterleiten soll.

Note

Wenn Sie Hosts in mehreren Availability Zones haben, erstellen Sie eine Regel für ausgehenden Datenverkehr für jede dieser Availability Zones.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4
	subnet-9de72a91	available	vpc-7523e317	172.31.0.0/20	4091
	subnet-04575af9	available	vpc-7523e317	172.31.16.0/20	4090
	subnet-3d33734c	available	vpc-7523e317	172.31.32.0/20	4091
	subnet-caa74a67	available	vpc-7523e317	172.31.48.0/20	4091
	subnet-9dfcbf9f5	available	vpc-7523e317	172.31.64.0/20	4091

10. Wählen Sie Save aus.

Aktualisieren der Instance-Sicherheitsgruppe

Standardmäßig erlaubt die Sicherheitsgruppe, die Sie für Ihre Instances erstellt haben, eingehenden Datenverkehr nicht. Während Elastic Beanstalk die Standardgruppe so ändert, dass die Instances SSH-Datenverkehr zulassen, müssen Sie Ihre benutzerdefinierte Instance-Sicherheitsgruppe für das Zulassen von RDP-Datenverkehr ändern, wenn Ihre Instances Windows-Instances sind.

So aktualisieren Sie die Instance-Sicherheitsgruppe für RDP

1. Aktivieren Sie in der Liste der Sicherheitsgruppen das Kontrollkästchen für die Instance-Sicherheitsgruppe.
2. Klicken Sie auf die Registerkarte Inbound und wählen Sie Edit aus.
3. Wählen Sie bei Bedarf Add another rule (Weitere Regel hinzufügen) aus.
4. Geben Sie die folgenden Werte ein und wählen Sie Save (Speichern) aus.

Typ

RDP

Protocol (Protokoll)

TCP

Port Range (Port-Bereich)

3389

Source (Quelle)

Geben Sie die ID der Bastion-Host-Sicherheitsgruppe ein (z. B. sg-8a6f71e8) und wählen Sie Save (Speichern) aus.

Erstellen eines Bastion-Host

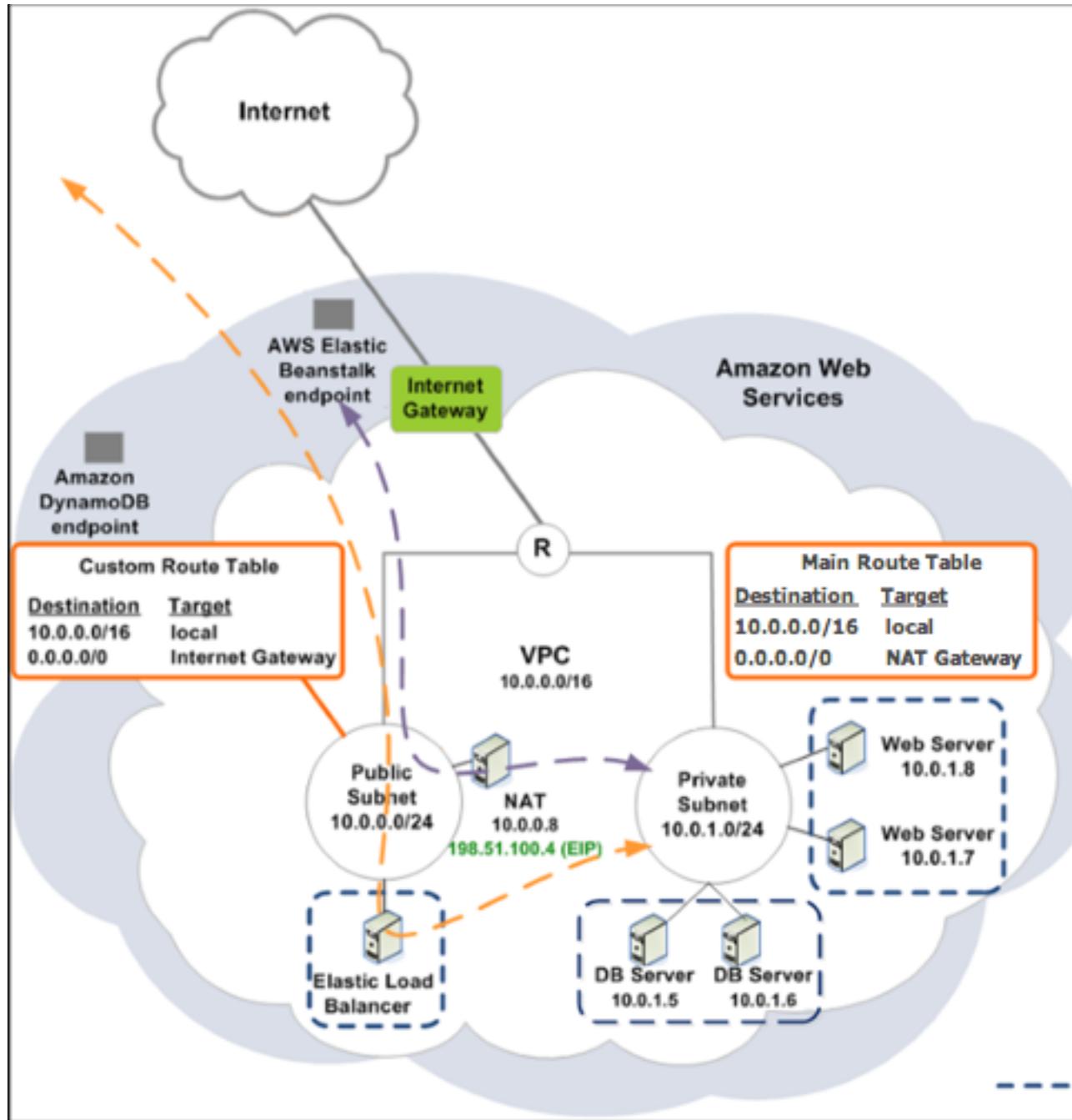
Zum Erstellen eines Bastion-Host starten Sie eine Amazon EC2-Instance in Ihrem öffentlichen Subnetz, das als Bastion-Host fungiert.

Weitere Informationen zum Einrichten eines Bastion-Host für Windows-Instances im privaten Subnetz finden Sie unter [Controlling Network Access to EC2 Instances Using a Bastion Server](#).

Weitere Informationen zum Einrichten eines Bastion-Host für Linux-Instances im privaten Subnetz finden Sie unter [Securely Connect to Linux Instances Running in a Private Amazon VPC](#).

Beispiel: Starten einer Elastic Beanstalk-Anwendung in einer VPC mit Amazon RDS

In diesem Abschnitt werden Sie schrittweise durch die Aufgaben geführt, die für die Bereitstellung einer Elastic Beanstalk-Anwendung mit Amazon RDS in einer VPC über ein NAT-Gateway notwendig sind. Ihre Infrastruktur sollte dann wie das folgende Diagramm aussehen.



Note

Wenn Sie noch keine DB-Instance mit Ihrer Anwendung verwendet haben, versuchen Sie, eine zu einer Testumgebung hinzuzufügen (p. 609) und eine Verbindung mit einer externen DB-Instance herzustellen (p. 973), bevor Sie eine VPC-Konfiguration hinzufügen.

Erstellen einer VPC mit einem öffentlichen und privaten Subnetz

Sie können die [Amazon VPC-Konsole](#) zum Erstellen einer VPC verwenden.

So erstellen Sie eine VPC

1. Melden Sie sich bei der [Amazon VPC-Konsole](#) an.
2. Wählen Sie im Navigationsbereich VPC Dashboard (VPC-Dashboard) aus. Wählen Sie dann die Option Create VPC (VPC erstellen).
3. Klicken Sie auf VPC with Public and Private Subnets (VPC mit öffentlichen und privaten Subnetzen) und anschließend auf Select (Auswählen).

Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

VPC with Public and Private Subnets

VPC with Public and Private Subnets and Hardware VPN Access

VPC with a Private Subnet Only and Hardware VPN Access

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).

Creates:

A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via Network Address Translation (NAT). (Hourly charges for NAT devices apply.)

Select

Cancel and Exit

4. Der Elastic Load Balancing-Load Balancer und die Amazon EC2-Instances müssen sich in derselben Availability Zone befinden, um miteinander kommunizieren zu können. Wählen Sie in jeder Liste Availability Zone dieselbe Availability Zone aus.

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: Public subnet

Private subnet's IPv4 CIDR: * 10.0.1.0/24 (251 IP addresses available)

Availability Zone: *

Private subnet name: Private subnet

You can add more subnets after AWS creates the VPC.

Specify the details of your NAT gateway (NAT gateway rates apply).

Elastic IP Allocation ID:

Service endpoints

Enable DNS hostnames: * Yes No

Hardware tenancy: *

Enable ClassicLink: * Yes No

Cancel and Exit Back Create VPC

5. Wählen Sie eine Elastic IP-Adresse für Ihr NAT-Gateway aus.
6. Wählen Sie Create VPC aus.

Der Assistent erstellt Ihre VPC, Subnetze und das Internet-Gateway. Außerdem wird die Haupt-Routing-Tabelle aktualisiert und eine benutzerdefinierte Routing-Tabelle erstellt. Zum Schluss erstellt der Assistent ein NAT-Gateway im öffentlichen Subnetz.

Note

Sie können auswählen, ob eine NAT-Instance statt im NAT-Gateway im öffentlichen Subnetz starten soll. Weitere Informationen finden Sie unter [Szenario 2: VPC mit öffentlichen und privaten Subnetzen \(NAT\)](#) im Amazon VPC-Benutzerhandbuch.

7. Nachdem die VPC erfolgreich erstellt wurde, erhalten Sie eine VPC-ID. Sie benötigen diesen Wert im nächsten Schritt. Um die VPC-ID anzuzeigen, wählen Sie Your VPCs (Ihre VPCs) im linken Bereich der [Amazon VPC-Konsole](#) aus.

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with options: 'Virtual Private Cloud', 'Your VPCs' (which is highlighted with a red box), 'Subnets', and 'Route Tables'. The main area is titled 'Create VPC' and has an 'Actions' dropdown. Below it is a search bar and a table with columns: Name, VPC ID, State, IPv4 CIDR, DHCP options set, Route table, and Network ACL. One row is visible, showing 'vpc-f56cff91' in the VPC ID column, which is also highlighted with a red box. The state is 'available'.

Erstellen einer DB-Subnetzgruppe

Eine DB-Subnetzgruppe für eine VPC ist eine Sammlung von Subnetzen (in der Regel private Subnetze), die Sie für Ihre Backend-RDS-DB-Instances bestimmen sollten. Jede DB-Subnetzgruppe sollte über mindestens ein Subnetz für jede Availability Zone in einer bestimmten AWS-Region verfügen. Weitere Informationen erhalten Sie unter [Erstellen eines Subnetzes in Ihrer VPC](#).

Erstellen einer DB-Subnetzgruppe

1. Öffnen Sie die [Amazon RDS-Konsole](#).
2. Wählen Sie im Navigationsbereich Subnetzgruppe aus.
3. Wählen Sie DB-Subnetzgruppe erstellen aus.
4. Wählen Sie Name und geben Sie dann den Namen Ihrer DB-Subnetzgruppe ein.
5. Wählen Sie Description (Beschreibung) und beschreiben Sie dann Ihre DB-Subnetzgruppe.
6. Wählen Sie für VPC die ID der zuvor erstellten VPC aus.
7. Wählen Sie im Abschnitt Add subnets (Subnetze hinzufügen) die Option Add all the subnets related to this VPC (Alle zu dieser VPC gehörenden Subnetze hinzufügen).

Availability zone	Subnet ID	CIDR block	Action
us-east-2c	subnet-da3408ae	10.0.1.0/24	<button>Remove</button>
us-east-2c	subnet-db3408af	10.0.0.0/24	<button>Remove</button>
us-east-2b	subnet-4f195024	10.0.2.0/24	<button>Remove</button>
us-east-2a	subnet-fe064f95	10.0.3.0/24	<button>Remove</button>

- Wenn Sie fertig sind, wählen Sie Erstellen aus.

Ihre neue DB-Subnetzgruppe wird in der Liste der Subnetzgruppen in der Amazon RDS-Konsole angezeigt. Sie können sie wählen, um die Details im Details-Bereich unten auf der Seite anzeigen zu lassen – einschließlich aller Subnetze, die dieser Gruppe angehören.

Bereitstellen in Elastic Beanstalk

Nach dem Einrichten Ihrer VPC können Sie Ihre Umgebung darin erstellen und Ihre Anwendung für Elastic Beanstalk bereitstellen. Sie können dies mit der Elastic Beanstalk-Konsole tun oder die AWS-Toolkits, AWS CLI, EB CLI oder Elastic Beanstalk-API verwenden. Wenn Sie die Elastic Beanstalk-Konsole verwenden, laden Sie einfach Ihre .war- oder .zip-Datei hoch und wählen Sie die VPC-Einstellung im Assistenten aus. Elastic Beanstalk erstellt Ihre Umgebung in Ihrer VPC und stellt Ihre Anwendung bereit. Alternativ können Sie zum Bereitstellen Ihrer Anwendung auch die AWS-Toolkits, die AWS CLI, die EB CLI oder die Elastic Beanstalk-API verwenden. Dazu müssen Sie Ihre VPC-Optionseinstellungen in einer Konfigurationsdatei definieren und diese Datei mit Ihrem Quell-Bundle bereitstellen. Dieses Thema enthält Anleitungen für beide Methoden.

Bereitstellen mit der Elastic Beanstalk-Konsole

Die Elastic Beanstalk-Konsole führt Sie durch die Schritte zum Erstellen Ihrer neuen Umgebung in der VPC. Sie müssen eine .war-Datei (für Java-Anwendungen) bzw. eine .zip-Datei (für alle anderen Anwendungen) angeben. Nehmen Sie auf der Seite VPC Configuration (VPC-Konfiguration) des Elastic Beanstalk-Umgebungsassistenten folgende Einstellungen vor:

VPC

Wählen Sie Ihre VPC aus.

VPC Security Group (VPC-Sicherheitsgruppe)

Wählen Sie die weiter oben erstellte Instance-Sicherheitsgruppe aus.

ELB visibility (ELB-Sichtbarkeit)

Wählen Sie External aus, wenn Ihr Load Balancer öffentlich verfügbar sein soll, oder Internal, wenn der Load Balancer nur in Ihrer VPC zur Verfügung stehen soll.

Wählen Sie die Subnetze für Ihren Load Balancer und die EC2 Instances aus. Stellen Sie sicher, dass Sie das öffentliche Subnetz für den Load Balancer und das private Subnetz für Ihre Amazon EC2-Instances auswählen. Standardmäßig erstellt der VPC-Assistent das öffentliche Subnetz in 10.0.0.0/24 und das private Subnetz in 10.0.1.0/24.

Sie können die Subnetz-IDs anzeigen, indem Sie Subnets (Subnetze) in der [Amazon VPC-Konsole](#) auswählen.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4 IPs	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	

Summary

Subnet ID: subnet-ec18feb4
IPv4 CIDR: 172.31.16.0/20
IPv6 CIDR:
State: available
VPC: vpc-f56cff91
Available IPs: 4089

Availability Zone: us-east-1d
Route table: rtb-4f0f472b
Network ACL: acl-ca059fae
Default subnet: yes
Auto-assign Public IP: yes
Auto-assign IPv6 address: no

Bereitstellen über die AWS Toolkits, EB-CLI, AWS-CLI oder API

Beim Bereitstellen Ihrer Anwendung für Elastic Beanstalk unter Verwendung der AWS-Toolkits, der EB CLI, der AWS CLI oder der API können Sie Ihre VPC-Optionseinstellungen in einer Datei angeben und mit Ihrem Quell-Bundle bereitstellen. Weitere Informationen finden Sie unter [Erweiterte Umgebungsanpassung mit Konfigurationsdateien \(.ebextensions\)](#) (p. 722).

Wenn Sie die Optionseinstellungen aktualisieren, müssen Sie mindestens Folgendes angeben:

- VPCId (VPC-ID) – Enthält die ID der VPC.
- Subnets (Subnetze) – Enthält die ID des Auto Scaling-Gruppensubnetzes. In diesem Beispiel ist dies die ID des privaten Subnetzes.
- ELBSubnets (ELB-Subnetze) – Enthält die ID des Subnetzes für den Load Balancer. In diesem Beispiel ist dies die ID des öffentlichen Subnetzes.
- SecurityGroups (Sicherheitsgruppen) – Enthält die ID der Sicherheitsgruppen.
- DBSubnets (DB-Subnetze) – Enthält die ID der DB-Subnetze.

Note

Wenn Sie DB-Subnetze verwenden, müssen Sie zusätzliche Subnetze in Ihrer VPC erstellen, um alle Availability Zones in der AWS-Region abzudecken.

Optional können Sie auch die folgenden Informationen angeben:

- ELBScheme (ELB-Schema) – Geben Sie `internal` an, wenn ein interner Load Balancer in Ihrer VPC erstellt werden soll, damit nur innerhalb der VPC auf die Elastic Beanstalk-Anwendung zugegriffen werden kann.

Nachfolgend finden Sie ein Beispiel für die Optionseinstellungen, die Sie bei der Elastic Beanstalk-Anwendungsbereitstellung in einer VPC verwenden können. Weitere Informationen zu VPC-

Optionseinstellungen (einschließlich Beispiele, wie diese angegeben werden, Standardwerte und gültige Werte) finden Sie in der aws:ec2:vpc-Namespace-Tabelle in [Konfigurationsoptionen \(p. 646\)](#).

```
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

Wenn Sie DB-Subnetze verwenden, müssen Sie zusätzliche Subnetze in Ihrer VPC erstellen, um alle Availability Zones in der AWS-Region abzudecken.

Verwenden von Elastic Beanstalk mit VPC-Endpunkten

Ein VPC-Endpunkt ermöglicht Ihnen, eine private Verbindung zwischen Ihrer VPC und unterstützten AWS-Services und VPC-Endpunktsservices mit AWS PrivateLink einzurichten, ohne dass ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder eine AWS Direct Connect-Verbindung erforderlich sind.

Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit Ressourcen im Service zu kommunizieren. Der Datenverkehr zwischen Ihrer VPC und dem anderen Service verlässt das Amazon-Netzwerk nicht. Vollständige Informationen zu VPC-Endpunkten finden Sie unter [VPC-Endpunkte](#) im Amazon VPC-Benutzerhandbuch.

AWS Elastic Beanstalk unterstützt AWS PrivateLink. Diese Lösung stellt eine private Verbindung mit dem Elastic Beanstalk-Service her und verhindert den Transport von Daten über das öffentliche Internet. Damit Ihre Anwendung über AWS PrivateLink Anforderungen an Elastic Beanstalk senden kann, müssen Sie einen VPC-Endpunkttyp konfigurieren, der als Schnittstellen-VPC-Endpunkt (Schnittstellenendpunkt) bezeichnet wird. Weitere Informationen finden Sie unter [Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#) im Amazon VPC-Benutzerhandbuch.

Note

Elastic Beanstalk unterstützt AWS PrivateLink und Schnittstellen-VPC-Endpunkte in einer begrenzten Anzahl von AWS-Regionen. Wir arbeiten daran, die Unterstützung in naher Zukunft auf weitere AWS-Regionen zu erweitern.

Einrichten eines VPC-Endpunkts für Elastic Beanstalk

Zur Erstellung des Schnittstellen-VPC-Endpunkts für den Elastic Beanstalk-Service in Ihrer VPC folgen Sie der Prozedur [Erstellung eines Schnittstellenendpunkts](#). Wählen Sie in Service Name (Servicename) die Option com.amazonaws.**Region**.elasticBeanstalk aus.

Wenn Ihre VPC mit einem Zugang zum öffentlichen Internet konfiguriert ist, kann Ihre Anwendung weiter unter Verwendung des öffentlichen Endpunkts elasticbeanstalk.**region**.amazonaws.com über das Internet auf Elastic Beanstalk zugreifen. Sie können dies durch die Aktivierung von Enable DNS name (DNS-Namen aktivieren) während der Endpunktterstellung verhindern (standardmäßig „true“). Hierdurch wird ein DNS-Eintrag in Ihrer VPC hinzugefügt, der den Endpunkt für den öffentlichen Service dem VPC-Schnittstellenendpunkt zuordnet.

Einrichtung eines VPC-Endpunkts zur Verbesserung der Integrität

Wenn Sie für Ihre Umgebung [erweiterte Integritätsberichte \(p. 824\)](#) aktiviert haben, können Sie diese so konfigurieren, dass erweiterte Integritätsinformationen ebenfalls über AWS PrivateLink gesendet werden. Erweiterte Integritätsinformationen werden vom Daemon healthd, einer Elastic Beanstalk-Komponente auf den Instances in Ihrer Umgebung, an einen separaten, erweiterten Elastic Beanstalk-Integritätsservice gesendet. Zur Erstellung des Schnittstellen-VPC-Endpunkts für diesen Service in Ihrer VPC folgen Sie der Prozedur [Erstellung eines Schnittstellenendpunkts](#). Wählen Sie in Service Name (Servicename) die Option com.amazonaws.**Region**.elasticbeanstalk-Gesundheit aus.

Important

Der Daemon healthd sendet erweiterte Integritätsdaten an den öffentlichen Endpunkt, elasticbeanstalk-health.**region**.amazonaws.com. Wenn Ihre VPC mit einem Zugang zum öffentlichen Internet konfiguriert und Enable DNS name (DNS-Name aktivieren) für den VPC-Endpunkt deaktiviert ist, werden erweiterte Integritätsdaten über das öffentliche Internet transportiert. Dies wird von Ihnen bei der Einrichtung eines VPC-Endpunkts für erweiterte Integrität wahrscheinlich nicht gewünscht. Stellen Sie daher sicher, dass Enable DNS name (DNS-Name aktivieren) aktiviert ist (standardmäßig „true“).

Verwenden von VPC-Endpunkten in einer privaten VPC

Eine private VPC (oder ein privates Subnetz in einer VPC) verfügt nicht über einen Zugang zum öffentlichen Internet. Möglicherweise möchten Sie Ihre Elastic Beanstalk-Umgebung in einer [privaten VPC \(p. 990\)](#) ausführen und Schnittstellen-VPC-Endpunkte konfigurieren, um die Sicherheit zu verbessern. In diesem Fall sollten Sie beachten, dass Ihre Umgebung möglicherweise versucht, aus anderen Gründen eine Verbindung zum Internet herzustellen, zusätzlich zur Kontaktaufnahme mit dem Elastic Beanstalk-Service. Weitere Informationen zur Ausführung einer Umgebung in einer privaten VPC finden Sie unter [the section called “Ausführung einer Elastic Beanstalk-Umgebung in einer privaten VPC” \(p. 991\)](#).

Verwenden von Endpunktrichtlinien zur Steuerung des Zugriffs mit VPC-Endpunkten

Standardmäßig ermöglicht ein VPC-Endpunkt den vollständigen Zugriff auf den Service, dem er zugeordnet ist. Wenn Sie einen Endpunkt erstellen oder ändern, können Sie diesem eine Endpunktrichtlinie anfügen.

Eine Endpunktrichtlinie ist eine AWS Identity and Access Management (IAM)-Ressourcenrichtlinie, die den Zugriff vom Endpunkt aus auf den angegebenen Service steuert. Die Endpunktrichtlinie ist für den jeweiligen Endpunkt spezifisch. Sie ist von allen Benutzer- oder Instance-IAM-Richtlinien, die in Ihrer Umgebung möglicherweise vorhanden sind, getrennt; weder überschreibt sie diese noch ersetzt sie diese. Details zur Erstellung und Verwendung von VPC-Endpunktrichtlinien finden Sie unter [Steuerung des Zugriffs auf Services mit VPC-Endpunkten](#) im Amazon VPC-Benutzerhandbuch.

Das folgende Beispiel lehnt die Berechtigung zum Beenden einer Umgebung über den VPC-Endpunkt für alle Benutzer ab und ermöglicht den vollständigen Zugriff auf alle übrigen Aktionen.

```
{  
    "Statement": [  
        {  
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "elasticbeanstalk:TerminateEnvironment",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

Note

Zu diesem Zeitpunkt unterstützt nur der Elastic Beanstalk-Hauptservice die Anfügung einer Endpunkttrichtlinie an seinen VPC-Endpunkt. Der erweiterte Integritätsservice unterstützt Endpunkttrichtlinien nicht.

Konfigurieren des Entwicklungscomputers für die Verwendung mit Elastic Beanstalk

Auf dieser Seite erfahren Sie, wie Sie einen lokalen Computer für die Entwicklung einer AWS Elastic Beanstalk-Anwendung einrichten. Diese Informationen decken Ordnerstruktur, Versionsverwaltung und CLI-Tools ab.

Themen

- [Erstellen eines Projektordners \(p. 1005\)](#)
- [Einrichten der Quellsteuerung \(p. 1005\)](#)
- [Konfigurieren eines Remote-Repositorys \(p. 1006\)](#)
- [Installieren der EB CLI \(p. 1006\)](#)
- [Installieren der AWS-CLI \(p. 1007\)](#)

Erstellen eines Projektordners

Erstellen Sie einen Ordner für Ihr Projekt. Sie können den Ordner überall auf Ihrer lokalen Festplatte speichern, wenn Sie über die Berechtigung für Lese- und Schreibvorgänge verfügen. Das Erstellen eines Ordners in Ihrem Benutzerordner ist möglich. Wenn Sie mit mehreren Anwendungen arbeiten möchten, erstellen Sie Ihre Projektordner in einem anderen Ordner und nennen Sie ihn zum Beispiel `workspace` oder `projects`, damit alles gut organisiert ist:

```
workspace/
|-- my-first-app
`-- my-second-app
```

Der Inhalt Ihres Projektordners variiert je nach Webcontainer oder Framework, den bzw. das Ihre Anwendung nutzt.

Note

Vermeiden Sie Ordner und Pfade mit einfachen Anführungszeichen ('') oder doppelten Anführungszeichen ("") im Ordnernamen oder einem Pfadelement. Einige Elastic Beanstalk-Befehle können in einem Ordner mit diesen Zeichen im Namen nicht ausgeführt werden.

Einrichten der Quellsteuerung

Richten Sie die Versionsverwaltung ein, um sich vor einem versehentlichen Löschen von Dateien oder Code in Ihrem Projektordner zu schützen und um Änderungen rückgängig zu machen, die Ihr Projekt beschädigen.

Wenn Sie kein System zur Versionsverwaltung haben, können Sie Git verwenden, eine kostenlose und benutzerfreundliche Option, die sich problemlos in die Elastic Beanstalk-Befehlszeilenschnittstelle integrieren lässt. Besuchen Sie die [Git-Homepage](#), um Git zu installieren.

Befolgen Sie die Anweisungen auf der Git-Website zum Installieren und Konfigurieren von Git und führen Sie dann `git init` in Ihrem Projektordner aus, um ein lokales Repository einzurichten:

```
~/workspace/my-first-app$ git init
Initialized empty Git repository in /home/local/username/workspace/my-first-app/.git/
```

Wenn Sie Inhalte zu Ihrem Projektordner hinzufügen und Inhalte aktualisieren, führen Sie ein Commit für diese Änderungen im Git-Repository durch:

```
~/workspace/my-first-app$ git add default.jsp
~/workspace/my-first-app$ git commit -m "add default JSP"
```

Jedes Mal, wenn Sie ein Commit durchführen, erstellen Sie einen Snapshot Ihres Projekts, den Sie später wiederherstellen können, wenn etwas schiefläuft. Viele weitere Informationen zu Git-Befehlen und -Workflows finden Sie in der [Git-Dokumentation](#).

Konfigurieren eines Remote-Repositorys

Was ist, wenn Ihre Festplatte abstürzt, oder Sie in Ihrem Projekt auf einem anderen Computer arbeiten möchten? Um Ihren Quellcode online zu sichern und von einem beliebigen Computer darauf zuzugreifen, konfigurieren Sie ein Remote-Repository, an das Sie Ihre Commits übertragen können.

Mit AWS CodeCommit können Sie ein privates Repository in der AWS Cloud erstellen. CodeCommit ist im [kostenlosen Kontingent für AWS](#) für bis zu fünf AWS Identity and Access Management (IAM)-Benutzer in Ihrem Konto enthalten. Preisdetails finden Sie unter [AWS CodeCommit – Preise](#).

Besuchen Sie das [AWS CodeCommit-Benutzerhandbuch](#), um Anweisungen zum Einrichten zu erhalten.

GitHub ist eine andere beliebte Option für das Speichern Ihres Projektcodes online. Sie können damit kostenlos ein öffentliches Online-Repository erstellen. Gegen eine monatliche Gebühr werden auch private Repositorys unterstützt. Registrieren Sie sich unter [github.com](#) für GitHub.

Nachdem Sie ein Remote-Repository für Ihr Projekt erstellt haben, fügen Sie es mit `git remote add` an Ihr lokales Repository an:

```
~/workspace/my-first-app$ git remote add origin ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/my-repo
```

Installieren der EB CLI

Verwenden Sie die [EB CLI \(p. 1008\)](#) für die Verwaltung Ihrer Elastic Beanstalk-Umgebungen und Überwachung des Zustands über die Befehlszeile. Installationsanweisungen finden Sie unter [Install the EB CLI \(Installieren Sie die EB CLI\) \(p. 1009\)](#).

Standardmäßig packt die EB CLI alles in Ihren Projektordner und lädt in auf Elastic Beanstalk als Quell-Bundle hoch. Wenn Sie Git und die EB CLI gemeinsam verwenden, können Sie mit `.gitignore` verhindern, dass erstellte Klassendateien in der Quelle gespeichert werden. Mit `.ebignore` verhindern Sie, dass Quelldateien bereitgestellt werden.

Sie können auch [die EB CLI so konfigurieren, dass sie ein Build-Artefakt bereitstellt \(p. 1019\)](#) (eine WAR- oder ZIP-Datei) anstatt die Inhalte Ihres Projektordners.

Installieren der AWS-CLI

Die AWS-Befehlszeilenschnittstelle (Command Line Interface, AWS-CLI) ist ein einheitliches Client für AWS-Services, der Befehle für alle öffentlichen API-Operationen bietet. Diese Befehle sind auf niedrigerer Ebene als die der EB CLI, sodass es oft mehr Befehle für eine Operation mit der AWS-CLI erfordert. Andererseits können Sie mit der AWS-Befehlszeilenschnittstelle mit jeder Anwendung oder Umgebung in Ihrem Konto arbeiten, ohne dass ein Repository auf Ihrem lokalen Computer eingerichtet werden muss. Verwenden Sie die AWS-CLI zum Erstellen von Skripts, die operative Aufgaben vereinfachen oder automatisieren.

Weitere Informationen zu unterstützten Services und zum Herunterladen der AWS-Befehlszeilenschnittstelle finden Sie unter [AWS-Befehlszeilenschnittstelle](#).

Verwenden des Elastic Beanstalk Command Line Interface (EB CLI)

Die EB CLI ist eine Befehlszeilenschnittstelle für AWS Elastic Beanstalk, die interaktive Befehle bereitstellt, wodurch Sie Umgebungen aus einem lokalen Repository einfach erstellen, aktualisieren und überwachen können. Verwenden Sie die EB CLI als Teil des täglichen Entwicklungs- und Testzyklus alternativ zur Elastic Beanstalk-Konsole.

Note

Die aktuelle Version der EB CLI weist andere grundlegende Befehle auf als die Versionen vor 3.0. Wenn Sie eine ältere Version verwenden, finden Sie unter [Zu EB CLI 3 und CodeCommit migrieren \(p. 1098\)](#) weitere Informationen zur Migration.

Nachdem Sie [die EB-CLI installiert \(p. 1009\)](#) und ein Projektverzeichnis konfiguriert haben, können Sie Umgebungen mit nur einem Befehl erstellen:

```
~/my-app$ eb create my-env
```

Der Quellcode für die EB-CLI ist ein Open-Source-Projekt. Es befindet sich im GitHub-Repository [aws/aws-elastic-beanstalk-cli](#). Sie können sich beteiligen, indem Sie Probleme melden, Vorschläge machen und Pull-Anforderungen übermitteln. Wir schätzen Ihre Beiträge! Für eine Umgebung, in der Sie nur die EB-CLI verwenden möchten, empfehlen wir die Installation mit einem der EB-CLI-Einrichtungs-Skripts (siehe [the section called “Installieren der EB-CLI mit Setup-Skripts” \(p. 1009\)](#)).

Bisher unterstützte Elastic Beanstalk eine separate CLI – unter dem Namen [Elastic Beanstalk API CLI \(p. 1099\)](#) – die direkten Zugriff auf API-Operationen bot. Diese wurde durch die [AWS CLI \(p. 1007\)](#) ersetzt, die dieselbe Funktionalität für die APIs aller AWS-Services bietet.

Mit der AWS CLI haben Sie direkten Zugriff auf die Elastic Beanstalk-API. Die AWS CLI eignet sich gut für Skripts, lässt sich aber – aufgrund der Anzahl der auszuführenden Befehle und der Anzahl der Parameter für die einzelnen Befehle – über die Befehlszeile nicht so einfach verwenden. Beispielsweise ist zum Erstellen einer Umgebung eine ganze Reihe von Befehlen erforderlich:

```
~$ 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)"
```

Weitere Informationen zum Installieren der EB CLI, zum Konfigurieren eines Repositorys und zum Verwenden von Umgebungen finden Sie in den folgenden Themen.

Themen

- [Installieren der EB CLI \(p. 1009\)](#)
- [Konfigurieren der EB CLI \(p. 1017\)](#)
- [Elastic Beanstalk-Umgebungen mit der EB CLI verwalten \(p. 1020\)](#)
- [EB CLI mit AWS CodeBuild verwenden \(p. 1025\)](#)
- [Verwenden der EB CLI mit Git \(p. 1027\)](#)
- [Verwenden der EB CLI mit AWS CodeCommit \(p. 1028\)](#)

- [Verwenden der EB-CLI zur Überwachung des Umgebungszustands \(p. 1033\)](#)
- [Verwalten mehrerer Elastic Beanstalk-Umgebungen als Gruppe mit der EB CLI \(p. 1038\)](#)
- [Beheben von EB CLI-Fehlern \(p. 1040\)](#)
- [EB-CLI-Befehlsreferenz \(p. 1042\)](#)
- [EB-CLI 2.6 \(veraltet\) \(p. 1098\)](#)
- [Elastic Beanstalk API-Befehlszeilenschnittstelle \(veraltet\) \(p. 1099\)](#)

Installieren der EB CLI

Bei der AWS Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI) handelt es sich um einen Befehlszeilen-Client, mit dem Sie Elastic Beanstalk-Umgebungen erstellen, konfigurieren und verwalten können. Für weitere Informationen zur EB-CLI siehe [EB CLI \(p. 1008\)](#).

Themen

- [Installieren der EB-CLI mit Setup-Skripts \(p. 1009\)](#)
- [Manuelle Installation der EB-CLI \(p. 1009\)](#)

Installieren der EB-CLI mit Setup-Skripts

Die einfachste und zugleich empfohlene Methode zum Installieren der EB-CLI ist die Verwendung der auf GitHub verfügbaren [EB-CLI-Einrichtungsskripts](#). Verwenden Sie die Skripts zum Installieren der EB-CLI unter Linux, macOS oder Windows. Die Skripts installieren die EB CLI und ihre Abhängigkeiten, einschließlich Python und pip. Die Skripts erstellen auch eine virtuelle Umgebung für die EB CLI. Installationsanweisungen finden Sie im [aws/aws-elastic-beanstalk-cli-setup](#)-Repository auf GitHub.

Manuelle Installation der EB-CLI

Zur Installation der EB CLI empfehlen wir die Verwendung der [EB CLI Setupskripts](#). Wenn die Setupskripts mit Ihrer Entwicklungsumgebung nicht kompatibel sind, installieren Sie die EB CLI manuell.

Die primäre Verteilungsmethode für die EB CLI auf Linux, Windows und macOS ist pip. Dies ist ein Python-Paket-Manager, der eine einfache Möglichkeit zum Installieren, Aktualisieren und Entfernen von Python-Paketen und ihren Abhängigkeiten bietet. Für macOS können Sie auch die neueste Version der EB CLI mit Homebrew abrufen.

Kompatibilitätshinweis

Die EB CLI wurde für Python entwickelt und erfordert Python Version 2.7, 3.4, Version 3.4 oder eine neuere Version.

Note

Amazon Linux wird ab Version 2015.03 mit Python 2.7 und pip ausgeliefert.

Wir empfehlen die Verwendung der [EB CLI-Setupskripts](#) zur Installation der EB CLI und ihrer Abhängigkeiten. Wenn Sie die EB CLI manuell installieren, kann es schwierig sein, Abhängigkeitskonflikte in Ihrer Entwicklungsumgebung in den Griff zu bekommen.

Die EB CLI und die [AWS-Befehlszeilenschnittstelle](#) (AWS CLI) teilen sich eine Abhängigkeit vom [Botocore](#) Python-Paket. Aufgrund einer größeren Änderung in botocore basieren verschiedene Versionen dieser beiden CLI-Tools auf unterschiedlichen Versionen von botocore.

Die neuesten Versionen der beiden CLIs sind kompatibel. Wenn Sie eine ältere Version verwenden, finden Sie in der folgenden Tabelle eine verwendbare kompatible Version.

EB CLI-Version	Kompatible AWS CLI-Version
3.14.5 oder niedriger	1.16.9 oder niedriger
3.14.6 oder höher	1.16.11 oder höher

Installieren der EB CLI

Wenn Sie bereits über `pip` und eine unterstützte Version von Python verfügen, verwenden Sie das folgende Verfahren, um die EB CLI zu installieren.

Wenn Sie nicht über Python und `pip` verfügen, befolgen Sie die Anleitung für Ihr Betriebssystem.

- [Installieren von Python, "pip" und der EB CLI auf Linux \(p. 1011\)](#)
- [Installieren der EB CLI auf macOS \(p. 1013\)](#)
- [Installieren von Python, "pip" und der EB CLI auf Windows \(p. 1015\)](#)

So installieren Sie die EB CLI:

1. Führen Sie den folgenden Befehl aus.

```
$ pip install awsebcli --upgrade --user
```

Die Option `--upgrade` weist `pip` an, alle bereits installierten Anforderungen zu aktualisieren. Die Option `--user` bewirkt, dass `pip` das Programm in ein Unterverzeichnis Ihres Benutzerverzeichnisses installiert, damit keine vom Betriebssystem verwendeten Bibliotheken geändert werden.

Note

Falls bei der Installation der EB CLI mit `pip` Probleme auftreten, können Sie die [EB CLI in einer virtuellen Umgebung installieren \(p. 1016\)](#), um das Tool und seine Abhängigkeiten zu isolieren, oder eine Python-Version verwenden, die Sie normalerweise nicht nutzen.

2. Fügen Sie den Pfad der ausführbaren Datei zu Ihrer `PATH`-Variable hinzu:

- Unter Linux und macOS:

Linux – `~/.local/bin`

macOS – `~/Library/Python/3.7/bin`

So ändern Sie die `PATH`-Variable (Linux, Unix oder macOS):

- a. Suchen Sie das Profilskript für die Shell in Ihrem Benutzerordner. Wenn Sie nicht sicher sind, welche Shell Sie haben, führen Sie `echo $SHELL` aus.

```
$ ls -a ~
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash – `.bash_profile`, `.profile` oder `.bash_login`.
- Zsh – `.zshrc`
- Tcsh – `.tcshrc`, `.cshrc` oder `.login`.

- b. Fügen Sie dem Profilskript einen Exportbefehl hinzu. Im folgenden Beispiel wird der aktuellen PATH-Variablen der durch **LOCAL_PATH** repräsentierte Pfad hinzugefügt.

```
export PATH=LOCAL_PATH:$PATH
```

- c. Laden Sie das im ersten Schritt beschriebene Profilskript in die aktuelle Sitzung. Im folgenden Beispiel wird das von **PROFILE_SCRIPT** repräsentierte Profilskript geladen.

```
$ source ~/PROFILE_SCRIPT
```

- Unter Windows:

Python 3.7 – %USERPROFILE%\AppData\Roaming\Python\Python37\Scripts

Python frühere Versionen – %USERPROFILE%\AppData\Roaming\Python\Scripts

So ändern Sie Ihre PATH-Variable (Windows):

- a. Betätigen Sie die Windows-Taste und geben Sie anschließend **environment variables** ein.
- b. Wählen Sie Edit environment variables for your account (Umgebungsvariablen für Ihr Konto bearbeiten).
- c. Wählen Sie PATH (PFAD) und Edit (Bearbeiten) aus.
- d. Fügen Sie im Feld Variable value (Variablenwert) Pfade hinzu, getrennt durch Semikola.
Beispiel: **C:\item1\path;C:\item2\path**
- e. Klicken Sie zweimal auf OK, um die neuen Einstellungen anzuwenden.
- f. Schließen Sie alle laufenden Eingabeaufforderungsfenster und öffnen Sie dann erneut ein Eingabeaufforderungsfenster.

3. Überprüfen Sie, ob die EB CLI korrekt installiert wurde, indem Sie eb --version ausführen.

```
$ eb --version  
EB CLI 3.14.8 (Python 3.7)
```

Durch regelmäßige Aktualisierungen der EB CLI werden Funktionalitäten zur Unterstützung [der neuesten Elastic Beanstalk-Funktionen](#) hinzugefügt. Führen Sie das Installationsprogramm erneut aus, um auf die neueste Version der EB CLI zu aktualisieren.

```
$ pip install awsebcli --upgrade --user
```

Wenn Sie die EB CLI deinstallieren müssen, verwenden Sie pip uninstall.

```
$ pip uninstall awsebcli
```

Installieren von Python, "pip" und der EB CLI auf Linux

Die EB CLI erfordert Python Version 2.7, 3.4, oder eine neuere Version. Falls Ihre Verteilung kein Python bzw. eine frühere Version enthält, installieren Sie Python vor der Installation von pip und der EB CLI.

So installieren Sie Python 3.7 unter Linux:

1. Stellen Sie fest, ob Python bereits installiert ist.

```
$ python --version
```

Note

Ist Python in Ihrer Linux-Verteilung enthalten, müssen Sie möglicherweise das Python-Entwicklerpaket installieren, um die Header und Bibliotheken zu erhalten, die zum Kompilieren von Erweiterungen und Installieren der EB CLI erforderlich sind. Verwenden Sie Ihren Paket-Manager, um das Entwicklerpaket (üblicherweise `python-dev` oder `python-devel` genannt) zu installieren.

2. Wenn kein Python 2.7 oder höher installiert ist, installieren Sie Python 3.7 mit dem Paket-Manager Ihrer Verteilung. Der Befehl und der Paketname variieren:

- Auf Debian-Derivaten wie Ubuntu verwenden Sie `APT`.

```
$ sudo apt-get install python3.7
```

- Auf Red Hat und seinen Derivaten verwenden Sie `yum`.

```
$ sudo yum install python37
```

- Verwenden Sie auf SUSE und Derivaten `zypper`.

```
$ sudo zypper install python3-3.7
```

3. Öffnen Sie eine Eingabeaufforderung oder Shell, und führen Sie den folgenden Befehl aus, um zu überprüfen, ob Python korrekt installiert ist.

```
$ python3 --version
Python 3.7.3
```

Installieren Sie `pip` mithilfe des Skripts der Python Packaging Authority, und installieren Sie anschließend die EB CLI.

So installieren Sie `pip` und die EB CLI

1. Laden Sie das Installationsskript von [pypa.io](https://bootstrap.pypa.io/get-pip.py) herunter.

```
$ curl -O https://bootstrap.pypa.io/get-pip.py
```

Das Skript wird heruntergeladen und installiert die neueste Version von `pip` und ein anderes erforderliches Paket namens `setuptools`.

2. Führen Sie das Skript mit Python aus.

```
$ 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
```

Wenn Sie Python, Version 3, direkt mit dem Befehl `python3` und nicht mit `python` aufrufen, ist sichergestellt, dass `pip` am korrekten Speicherort installiert wird, auch wenn eine ältere Systemversion von Python auf dem System vorhanden ist.

3. Fügen Sie den ausführbaren Pfad, `~/local/bin`, zu Ihrer `PATH`-Variablen hinzu.

So ändern Sie die `PATH`-Variable (Linux, Unix oder macOS):

- a. Suchen Sie das Profilskript für die Shell in Ihrem Benutzerordner. Wenn Sie nicht sicher sind, welche Shell Sie haben, führen Sie `echo $SHELL` aus.

```
$ ls -a ~  
.. .bash_logout .bash_profile .bashrc Desktop Documents Downloads
```

- Bash – `.bash_profile`, `.profile` oder `.bash_login`.
- Zsh – `.zshrc`
- Tcsh – `.tcshrc`, `.cshrc` oder `.login`.

- b. Fügen Sie dem Profilskript einen Exportbefehl hinzu. Im folgenden Beispiel wird der aktuellen `PATH`-Variablen der durch `LOCAL_PATH` repräsentierte Pfad hinzugefügt.

```
export PATH=$LOCAL_PATH:$PATH
```

- c. Laden Sie das im ersten Schritt beschriebene Profilskript in die aktuelle Sitzung. Im folgenden Beispiel wird das von `PROFILE_SCRIPT` repräsentierte Profilskript geladen.

```
$ source ~/PROFILE_SCRIPT
```

4. Stellen Sie sicher, dass `pip` korrekt installiert ist.

```
$ pip --version  
pip 8.1.2 from ~/.local/lib/python3.7/site-packages (python 3.7)
```

5. Verwenden Sie `pip` zum Installieren der EB CLI.

```
$ pip install awsebcli --upgrade --user
```

6. Überprüfen Sie, ob die EB CLI ordnungsgemäß installiert wurde.

```
$ eb --version  
EB CLI 3.14.8 (Python 3.7)
```

Führen Sie das Installationsprogramm erneut aus, um auf die neueste Version zu aktualisieren.

```
$ pip install awsebcli --upgrade --user
```

Installieren der EB CLI auf macOS

Wenn Sie den Homebrew-Paket-Manager verwenden, können Sie die EB CLI installieren, indem Sie den `brew`-Befehl verwenden. Sie können auch Python und `pip` installieren und anschließend die EB CLI mit `pip` installieren.

Installieren der EB-CLI mit Homebrew

In der Regel ist die neueste EB CLI-Version für Homebrew einige Tage nach der Veröffentlichung für `pip` verfügbar.

So installieren Sie die EB CLI mit **Homebrew**:

1. Stellen Sie sicher, dass Sie über die neueste Version von Homebrew verfügen.

```
$ brew update
```

2. Führen Sie `brew install awsebcli` aus.

```
$ brew install awsebcli
```

3. Überprüfen Sie, ob die EB CLI ordnungsgemäß installiert wurde.

```
$ eb --version  
EB CLI 3.14.8 (Python 3.7)
```

Installieren von Python, "pip" und der EB CLI auf macOS

Sie können die neueste Version von Python und `pip` installieren und mit diesen dann die EB CLI installieren.

So installieren Sie die EB CLI auf macOS

1. Laden Sie Python von der [Downloads](#)-Seite von [Python.org](#) herunter und installieren Sie es. Wir verwenden zur Demonstration Version 3.7.

Note

Die EB CLI erfordert Python 2 Version 2.7 oder Python 3 in den Versionen 3.4 bis 3.7.

2. Installieren Sie `pip` mithilfe des Skripts der Python Packaging Authority.

```
$ curl -O https://bootstrap.pypa.io/get-pip.py  
$ python3 get-pip.py --user
```

3. Verwenden Sie `pip` zum Installieren der EB CLI.

```
$ pip3 install awsebcli --upgrade --user
```

4. Fügen Sie den ausführbaren Pfad, `~/Library/Python/3.7/bin`, zu Ihrer `PATH`-Variablen hinzu.

So ändern Sie die `PATH`-Variable (Linux, Unix oder macOS):

- a. Suchen Sie das Profilskript für die Shell in Ihrem Benutzerordner. Wenn Sie nicht sicher sind, welche Shell Sie haben, führen Sie `echo $SHELL` aus.

```
$ ls -a ~  
. . . .bash_logout .bash_profile .bashrc Desktop Documents Downloads
```

- Bash – `.bash_profile`, `.profile` oder `.bash_login`.
- Zsh – `.zshrc`
- Tcsh – `.tcshrc`, `.cshrc` oder `.login`.

- b. Fügen Sie dem Profilskript einen Exportbefehl hinzu. Im folgenden Beispiel wird der aktuellen `PATH`-Variablen der durch `LOCAL_PATH` repräsentierte Pfad hinzugefügt.

```
export PATH=$LOCAL_PATH:$PATH
```

- c. Laden Sie das im ersten Schritt beschriebene Profilskript in die aktuelle Sitzung. Im folgenden Beispiel wird das von `PROFILE_SCRIPT` repräsentierte Profilskript geladen.

```
$ source ~/PROFILE_SCRIPT
```

5. Überprüfen Sie, ob die EB CLI ordnungsgemäß installiert wurde.

```
$ eb --version
EB CLI 3.14.8 (Python 3.7)
```

Führen Sie das Installationsprogramm erneut aus, um auf die neueste Version zu aktualisieren.

```
$ pip3 install awsebcli --upgrade --user
```

Installieren von Python, "pip" und der EB CLI auf Windows

Die Python Software Foundation bietet Installationsprogramme für Windows, die `pip` beinhalten.

So installieren Sie Python 3.7 und `pip` (Windows):

1. Laden Sie die ausführbare Installationsdatei für Python 3.7 Windows x86-64 von der [Downloads-Seite](#) von [Python.org](#) herunter.
2. Führen Sie das Installationsprogramm aus.
3. Wählen Sie Add Python 3.7 to PATH (Python 3.7 zu PATH hinzufügen) aus.
4. Wählen Sie Install Now (Jetzt installieren) aus.

Das Installationsprogramm installiert Python in Ihrem Benutzerordner und fügt seine ausführbaren Verzeichnisse Ihrem Benutzerpfad hinzu.

So installieren Sie die AWS-CLI mit `pip` (Windows):

1. Öffnen Sie im Start-Menü ein Eingabeaufforderungsfenster.
2. Verwenden Sie die folgenden Befehle, um zu prüfen, ob Python und `pip` korrekt installiert sind.

```
C:\Windows\System32> python --version
Python 3.7.3
C:\Windows\System32> pip --version
pip 9.0.1 from c:\users\myname\appdata\local\programs\python\python37\lib\site-packages
(python 3.7)
```

3. Installieren Sie die EB CLI mit `pip`.

```
C:\Windows\System32> pip install awsebcli --upgrade --user
```

4. Fügen Sie den ausführbaren Pfad %USERPROFILE%\AppData\roaming\Python\Python37\scripts zu Ihrer PATH-Umgebungsvariablen hinzu. Der Speicherort ist abhängig davon, ob Sie Python für einen oder alle Benutzer installieren.

So ändern Sie Ihre PATH-Variable (Windows):

- a. Betätigen Sie die Windows-Taste und geben Sie anschließend **environment variables** ein.
- b. Wählen Sie Edit environment variables for your account (Umgebungsvariablen für Ihr Konto bearbeiten).
- c. Wählen Sie PATH (PFAD) und Edit (Bearbeiten) aus.
- d. Fügen Sie im Feld Variable value (Variablenwert) Pfade hinzu, getrennt durch Semikola. Beispiel:
`C:\item1\path;C:\item2\path`

- e. Klicken Sie zweimal auf OK, um die neuen Einstellungen anzuwenden.
 - f. Schließen Sie alle laufenden Eingabeaufforderungsfenster und öffnen Sie dann erneut ein Eingabeaufforderungsfenster.
5. Starten Sie eine neue Befehlszeile, um die neue PATH-Variable in Kraft zu setzen.
 6. Überprüfen Sie, ob die EB CLI ordnungsgemäß installiert wurde.

```
C:\Windows\System32> eb --version
EB CLI 3.14.8 (Python 3.7)
```

Führen Sie das Installationsprogramm erneut aus, um auf die neueste Version zu aktualisieren.

```
C:\Windows\System32> pip install awsebcli --upgrade --user
```

Installieren der EB-CLI in einer virtuellen Umgebung

Sie können Versionsanforderungskonflikte mit anderen pip-Paketen vermeiden, indem Sie die EB CLI in einer virtuellen Umgebung installieren.

So installieren Sie die EB CLI in einer virtuellen Umgebung:

1. Installieren von virtualenv mit pip.

```
$ pip install --user virtualenv
```

2. Erstellen Sie eine virtuelle Umgebung.

```
$ virtualenv ~/eb-ve
```

Mithilfe der Option -p können Sie eine andere als die standardmäßige ausführbare Python-Datei verwenden.

```
$ virtualenv -p /usr/bin/python3.7 ~/eb-ve
```

3. Aktivieren Sie die virtuelle Umgebung.

Linux, Unix oder macOS

```
$ source ~/eb-ve/bin/activate
```

Windows

```
$ %USERPROFILE%\eb-ve\Scripts\activate
```

4. Installieren Sie die EB CLI.

```
(eb-ve)$ pip install awsebcli --upgrade
```

5. Überprüfen Sie, ob die EB CLI ordnungsgemäß installiert wurde.

```
$ eb --version
EB CLI 3.14.8 (Python 3.7)
```

Sie können den Befehl `deactivate` verwenden, um die virtuelle Umgebung zu beenden. Führen Sie den Aktivierungsbefehl bei jedem Start einer neuen Sitzung erneut aus.

Führen Sie das Installationsprogramm erneut aus, um auf die neueste Version zu aktualisieren.

```
(eb-ve)~$ pip install awsebcli --upgrade
```

Konfigurieren der EB CLI

Nach der [Installation der EB CLI \(p. 1009\)](#) können Sie Ihr Projektverzeichnis und EB CLI konfigurieren, indem Sie `eb init` ausführen.

Das folgende Beispiel zeigt die Konfigurationsschritte, wenn `eb init` zum ersten Mal in einem Projektordner mit dem Namen `eb` ausgeführt wird.

So initialisieren Sie ein EB CLI-Projekt

1. Zunächst werden Sie von der EB CLI aufgefordert, eine Region auszuwählen. Geben Sie die Zahl für die Region ein, die Sie verwenden möchten, und drücken Sie dann die Eingabetaste.

```
~/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. Geben Sie dann Ihren Zugriffsschlüssel und geheimen Schlüssel an, sodass die EB CLI Ressourcen für Sie verwalten kann. Zugriffsschlüssel werden in der AWS Identity and Access Management Console erstellt. Wenn Sie keine Schlüssel haben, finden Sie weitere Informationen unter [Wie erhalte ich Sicherheitsanmeldeinformationen?](#) in der Allgemeinen Amazon Web Services-Referenz.

```
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): 5ZR1rtTM4ciIAvd4EXAMPLEdtm+PiPSzpoK
```

3. Eine Anwendung in Elastic Beanstalk ist eine Ressource mit einer Reihe von Anwendungsversionen (Quelle), Umgebungen und gespeicherten Konfigurationen, die einer einzelnen Webanwendung zugeordnet sind. Jedes Mal, wenn Sie Ihren Quellcode mit der EB CLI in Elastic Beanstalk bereitstellen, wird eine neue Anwendungsversion erstellt und der Liste hinzugefügt.

```
Select an application to use
1) [ Create new Application ]
(default is 1): 1
```

4. Der Standard-Anwendungsname ist der Name des Ordners, in dem Sie `eb init` ausführen. Geben Sie einen beliebigen Namen ein, der Ihr Projekt beschreibt.

```
Enter Application Name
(default is "eb"): eb
Application eb has been created.
```

5. Wählen Sie eine Plattform, die mit der Sprache oder dem Framework, in der bzw. in dem Ihre Webanwendung entwickelt wurde, übereinstimmt. Wenn Sie noch keine Anwendung entwickelt haben, wählen Sie eine Plattform, die Sie interessiert. Wir zeigen Ihnen in Kürze, wie Sie eine Beispielanwendung starten. Sie können diese Einstellung zu einem späteren Zeitpunkt jederzeit ändern.

```
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. Wählen Sie Yes (Ja), um den Instances in Ihrer Elastic Beanstalk-Umgebung ein SSH-Schlüsselpaar zuzuweisen. Auf diese Weise können Sie eine direkte Verbindung für die Fehlerbehebung herstellen.

```
Do you want to set up SSH for your instances?  
(y/n): y
```

7. Wählen Sie ein vorhandenes Schlüsselpaar oder erstellen Sie ein neues. Damit mit eb init ein neues Schlüsselpaar erstellt werden kann, müssen Sie ssh-keygen auf Ihrem lokalen Computer installiert haben und über die Befehlszeile darauf zugreifen können. Die EB CLI registriert das neue Schlüsselpaar für Sie mit Amazon EC2 und speichert den privaten Schlüssel lokal in einem Ordner mit dem Namen .ssh in Ihrem Benutzerverzeichnis.

```
Select a keypair.  
1) [ Create new KeyPair ]  
(default is 1): 1
```

Ihre EB CLI-Installation ist nun konfiguriert und einsatzbereit. Anweisungen zum Erstellen und Verwenden von Elastic Beanstalk-Umgebungen finden Sie unter [Elastic Beanstalk-Umgebungen mit der EB CLI verwalten \(p. 1020\)](#).

Erweiterte Konfiguration

- [Ignorieren von Dateien, die .ebignore verwenden \(p. 1018\)](#)
- [Verwenden von benannten Profilen \(p. 1019\)](#)
- [Bereitstellen eines Artefakts anstelle des Projektordners \(p. 1019\)](#)
- [Konfigurationseinstellungen und Vorrang \(p. 1019\)](#)
- [Instance-Metadaten \(p. 1020\)](#)

Ignorieren von Dateien, die .ebignore verwenden

Sie können die EB CLI anweisen, bestimmte Dateien in Ihrem Projektverzeichnis zu ignorieren, indem Sie dem Verzeichnis die Datei `.ebignore` hinzufügen. Diese Datei funktioniert wie eine `.gitignore`-Datei. Bei der Bereitstellung Ihres Projektverzeichnisses für Elastic Beanstalk und der Erstellung einer neuen Anwendungsversion schließt die EB CLI keine Dateien ein, die als `.ebignore` im von ihr erstellten Quell-Bundle angegeben werden.

Wenn keine `.ebignore` vorhanden ist, jedoch eine `.gitignore`, ignoriert die EB CLI die Dateien, die in `.gitignore` angegeben sind. Wenn `.ebignore` vorhanden ist, liest die EB CLI `.gitignore` nicht.

Wenn `.ebignore` vorhanden ist, verwendet die EB CLI keine git-Befehle zum Erstellen Ihres Quell-Bundles. Das bedeutet, EB CLI ignoriert die in `.ebignore` angegebenen Dateien und bindet alle anderen Dateien ein. Insbesondere bindet sie nicht festgeschriebene Quelldateien ein.

Note

Unter Windows bewirkt das Hinzufügen von `.ebignore`, dass die EB CLI symbolischen Links folgt und die verknüpfte Datei bei der Erstellung eines Quell-Bundle einschließt. Das Problem ist bekannt und wird in einem der nächsten Updates behoben.

Verwenden von benannten Profilen

Wenn Sie Ihre Anmeldeinformationen in einer `credentials`- oder `config`-Datei als benanntes Profil speichern, können Sie die [--profile \(p. 1097\)](#)-Option zur expliziten Angabe eines Profils verwenden. Mit dem folgenden Befehl wird beispielsweise eine neue Anwendung mit dem `user2`-Profil erstellt.

```
$ eb init --profile user2
```

Sie können auch das Standard-Profil ändern, indem Sie die `AWS_EB_PROFILE`-Umgebungsvariable festlegen. Wenn diese Variable festgelegt ist, liest die EB CLI Anmeldeinformationen aus dem angegebenen Profil anstatt `default` oder `eb-cli`.

Linux, macOS oder Unix

```
$ export AWS_EB_PROFILE=user2
```

Windows

```
> set AWS_EB_PROFILE=user2
```

Bereitstellen eines Artefakts anstelle des Projektordners

Sie können die EB CLI anweisen, eine ZIP- oder WAR-Datei bereitzustellen, die Sie als Teil eines separaten Erstellungsprozess generieren, indem Sie die folgenden Zeilen zu `.elasticbeanstalk/config.yml` in Ihrem Projektordner hinzufügen.

```
deploy:  
  artifact: path/to/buildartifact.zip
```

Wenn Sie die EB CLI in Ihrem [Git-Repository \(p. 1027\)](#) konfigurieren und das Artifact nicht in Ihrer Quelle bereitstellen, verwenden Sie die `--staged`-Option zur Bereitstellung des neuesten Builds.

```
~/eb$ eb deploy --staged
```

Konfigurationseinstellungen und Vorrang

Die EB CLI verwendet eine Anbieterkette, um an mehreren Positionen – unter anderem in System- und Benutzerumgebungsvariablen sowie lokalen AWS-Konfigurationsdateien – nach AWS-Anmeldeinformationen zu suchen.

Die EB CLI sucht in der folgenden Reihenfolge nach Anmeldeinformationen und Konfigurationseinstellungen:

1. Befehlszeilenoptionen – Geben Sie ein benanntes Profil mit `--profile` an, um die Standardeinstellungen zu überschreiben.
2. Umgebungsvariablen – `AWS_ACCESS_KEY_ID` und `AWS_SECRET_ACCESS_KEY`.
3. Die Datei mit den AWS-Anmeldeinformationen – Gespeichert in `~/.aws/credentials` auf Linux- und OS X-Systemen oder in `C:\Users\USERNAME\.aws\credentials` auf Windows-Systemen. Diese Datei kann zusätzlich zu einem Standardprofil mehrere benannte Profile enthalten.
4. Die [AWS CLI-Konfigurationsdatei](#) – Gespeichert in `~/.aws/config` auf Linux- und OS X-Systemen oder in `C:\Users\USERNAME\.aws\config` auf Windows-Systemen. Diese Datei kann ein Standardprofil, [benannte Profile](#) und AWS CLI-spezifische Konfigurationsparameter für jedes Profil umfassen.
5. Bestehende EB CLI-Konfigurationsdatei – Gespeichert in `~/.elasticbeanstalk/config` auf Linux- und OS X-Systemen oder in `C:\Users\USERNAME\.elasticbeanstalk\config` auf Windows-Systemen.
6. Instance-Profilanmeldeinformationen – Diese Anmeldeinformationen können in Amazon EC2-Instances mit einer zugewiesenen Instance-Rolle verwendet werden. Ihre Bereitstellung übernimmt der Amazon EC2-Metadatenservice. Das [Instance-Profil \(p. 22\)](#) muss über die Berechtigung zur Nutzung von Elastic Beanstalk verfügen.

Wenn die Datei mit den Anmeldeinformationen ein benanntes Profil mit dem Namen "eb-cli" enthält, zieht die EB CLI dieses Profil dem Standard-Profil vor. Wenn keine Profile gefunden werden oder ein Profil gefunden wird, dieses jedoch nicht über die Berechtigung zur Nutzung von Elastic Beanstalk verfügt, fordert die EB CLI Sie auf, Schlüssel einzugeben.

Instance-Metadaten

Sie können die EB CLI aus einer Amazon EC2-Instance verwenden. Dazu erstellen Sie eine Rolle, die Zugriff auf die benötigten Ressourcen hat, und weisen die Rolle der Instance bei deren Start zu. Starten Sie die Instance und installieren Sie die EB CLI mit `pip`.

```
~$ sudo pip install awsebcli
```

`pip` ist auf Amazon Linux vorinstalliert.

Die EB CLI liest die Anmeldeinformationen aus den Instance-Metadaten. Weitere Informationen finden Sie im Thema zum [Gewähren von Zugriff auf AWS-Ressourcen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Elastic Beanstalk-Umgebungen mit der EB CLI verwalten

Nachdem Sie die EB CLI [installiert \(p. 1009\)](#) und das Projektverzeichnis [konfiguriert \(p. 1017\)](#) haben, können Sie nun eine Elastic Beanstalk-Umgebung mit der EB CLI erstellen, Quell- und Konfigurationsaktualisierungen bereitstellen sowie Protokolle und Ereignisse abrufen.

Note

Zum Erstellen von Umgebungen mit der EB-CLI ist eine [Service-Rolle \(p. 21\)](#) erforderlich. Eine Servicerolle generieren Sie, indem Sie eine Umgebung in der Elastic Beanstalk-Konsole erstellen.

Falls keine Servicerolle vorhanden ist, versucht die EB-CLI eine Rolle zu erstellen, wenn Sie eb create ausführen.

Die EB CLI gibt einen Beendigungscode gleich Null (0) für alle erfolgreichen Befehle zurück und einen Beendigungscode ungleich Null, wenn Fehler aufgetreten sind.

In den folgenden Beispielen wird ein leerer Projektordner mit dem Namen eb verwendet, der mit der EB CLI für eine Docker-Beispielanwendung initialisiert wurde.

Grundlegende Befehle

- [Eb create \(p. 1021\)](#)
- [Eb status \(p. 1021\)](#)
- [Eb health \(p. 1022\)](#)
- [Eb events \(p. 1022\)](#)
- [Eb logs \(p. 1023\)](#)
- [Eb open \(p. 1023\)](#)
- [Eb deploy \(p. 1023\)](#)
- [Eb config \(p. 1024\)](#)
- [Eb terminate \(p. 1024\)](#)

Eb create

Zum Erstellen Ihrer ersten Umgebung führen Sie [eb create \(p. 1053\)](#) aus und folgen Sie den Anweisungen. Falls das Projektverzeichnis Quellcode enthält, wird dieser von der EB CLI gebündelt und in der Umgebung bereitgestellt. Andernfalls wird eine Beispielanwendung verwendet.

```
~/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.
```

Es kann einige Minuten dauern, bis die Umgebung einsatzbereit ist. Drücken Sie Strg+C, um zur Befehlszeile zurückzukehren, während die Umgebung erstellt wird.

Eb status

Führen Sie eb status aus, um den aktuellen Status der Umgebung aufzurufen. Lautet der Status ready, steht die Beispielanwendung unter elasticbeanstalk.com zur Verfügung und die Umgebung kann aktualisiert werden.

```
~/eb$ eb status
Environment details for: elasticBeanstalkExa-env
  Application name: elastic-beanstalk-example
  Region: us-west-2
  Deployed Version: Sample Application
  Environment ID: e-gbzqc3jcr
  Platform: 64bit Amazon Linux 2015.03 v1.4.3 running Docker 1.6.2
  Tier: WebServer-Standard
  CNAME: elasticbeanstalkxa-env.elasticbeanstalk.com
  Updated: 2015-06-30 01:47:45.589000+00:00
  Status: Ready
  Health: Green
```

Eb health

Mit dem Befehl eb health zeigen Sie [Zustandsinformationen \(p. 824\)](#) zu den Instances der Umgebung sowie den allgemeinen Umgebungsstatus an. Verwenden Sie die Option --refresh, um den Zustand in einer aktiven Ansicht anzusehen, die alle 10 Sekunden aktualisiert wird.

```
~/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         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

Verwenden Sie eb events, um eine Liste der von Elastic Beanstalk ausgegebenen Ereignisse anzuzeigen.

```
~/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

Verwenden Sie eb logs, um Protokolle von einer Instance der Umgebung abzurufen. Standardmäßig werden mit eb logs Protokolle von der zuerst gestarteten Instance abgerufen und in der Standardausgabe angezeigt. Sie können eine Instance-ID mit der --instance-Option angeben und so Protokolle von einer bestimmten Instance abrufen.

Über die Option --all werden die Protokolle von allen Instances abgerufen und unter .elasticbeanstalk/logs in Unterverzeichnissen gespeichert.

```
~/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

Verwenden Sie eb open, um die Website der Umgebung in einem Browser zu öffnen:

```
~/eb$ eb open
```

In einer Windowed-Umgebung wird der Standardbrowser in einem neuen Fenster geöffnet. In einer Terminal-Umgebung wird ein Befehlszeilen-Browser (z. B. w3m) verwendet (sofern vorhanden).

Eb deploy

Sobald die Umgebung einsatzbereit ist, können Sie diese mit eb deploy aktualisieren.

Dieser Befehl ist besser geeignet, wenn Quellcode gebündelt und bereitgestellt wird, daher wurde in diesem Beispiel eine Dockerfile-Datei mit folgendem Inhalt im Projektverzeichnis erstellt:

~/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.zip

EXPOSE 80
```

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

Mit der Dockerfile-Datei wird ein Ubuntu 12.04-Image bereitgestellt und das Spiel 2048 installiert. Führen Sie eb deploy aus, um die Anwendung in die Umgebung hochzuladen:

```
~/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.
```

Bei Ausführung von eb deploy werden die Inhalte des Projektverzeichnisses von der EB CLI gebündelt und in der Umgebung bereitgestellt.

Note

Sofern Sie ein Git-Repository im Projektordner initialisiert haben, wird stets der neueste Commit von der EB CLI bereitgestellt, auch im Falle ausstehender Änderungen. Übergeben Sie die Änderungen, bevor Sie diese mit eb deploy in der Umgebung bereitstellen.

Eb config

Zeigen Sie mit dem Befehl eb config die Konfigurationsoptionen an, die für Ihre ausgeführte Umgebung verfügbar sind:

```
~/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:
        LowerBreathScaleIncrement: '-1'
    AWSEBAutoScalingScaleUpPolicy.aws:autoscaling:trigger:
        UpperBreathScaleIncrement: '1'
    AWSEBCloudwatchAlarmHigh.aws:autoscaling:trigger:
        UpperThreshold: '6000000'
    ...
...
```

Mit diesem Befehl wird eine Liste der verfügbaren Konfigurationsoptionen in einem Texteditor angezeigt. Viele der angezeigten Optionen weisen den Wert null auf, das heißt, sie werden nicht standardmäßig gesetzt, sondern können zur Aktualisierung der Umgebungsressourcen geändert werden. Weitere Informationen zu diesen Optionen finden Sie unter [Konfigurationsoptionen \(p. 646\)](#).

Eb terminate

Falls Sie die Umgebung nicht mehr benötigen, können Sie diese mit eb terminate beenden.

```
~/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-jc8t3pm SCN-stack-
AWSEBCloudwatchAlarmHigh-1XLMU7DNCBV6Y
INFO: Deleted CloudWatch alarm named: awseb-e-jc8t3pm SCN-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)
```

Eine vollständige Liste der verfügbaren EB CLI-Befehle finden Sie unter [EB-CLI-Befehlsreferenz \(p. 1042\)](#).

EB CLI mit AWS CodeBuild verwenden

[AWS CodeBuild](#) kompiliert den Quellcode, führt Einheitentests aus und erzeugt Artefakte, die bereitgestellt werden können. Sie können CodeBuild zusammen mit der EB CLI verwenden, um die Erstellung Ihrer Anwendung aus ihrem Quellcode zu automatisieren. Die Erstellung der Umgebung und alle anschließenden Bereitstellungen beginnen mit einem Erstellungsschritt, nach dem die so erstellte Anwendung bereitgestellt wird.

Note

Einige Regionen bieten kein CodeBuild an. Die Integration zwischen Elastic Beanstalk und CodeBuild funktioniert in diesen Regionen nicht.

Weitere Informationen zu den AWS-Services, die in den einzelnen Regionen angeboten werden, finden Sie in der [Tabelle der Regionen](#).

Erstellen einer Anwendung

So erstellen Sie eine Elastic Beanstalk-Anwendung, die CodeBuild verwendet:

1. Fügen Sie in Ihren Anwendungsordner eine Spezifikationsdatei für die CodeBuild-Erstellung ein, [buildspec.yml](#).
2. Fügen Sie der Datei einen `eb_codebuild_settings`-Eintrag mit spezifischen Optionen für Elastic Beanstalk hinzu.
3. Führen Sie [eb init \(p. 1067\)](#) im Ordner aus.

Elastic Beanstalk erweitert das [CodeBuild Build-Spezifikationsdateiformat](#) um die folgenden zusätzlichen Einstellungen:

```
eb_codebuild_settings:
  CodeBuildServiceRole: role-name
  ComputeType: size
  Image: image
  Timeout: minutes
```

CodeBuildServiceRole

Der ARN oder der Name der AWS Identity and Access Management (IAM)-Servicerolle, die CodeBuild verwenden kann, um in Ihrem Namen mit abhängigen AWS-Services zu interagieren. Dieser Wert ist erforderlich. Wenn Sie diesen auslassen, schlagen alle folgenden eb create- oder eb deploy-Befehle fehl.

Weitere Informationen zum Erstellen einer Servicerolle für CodeBuild finden Sie unter [Erstellen einer CodeBuild-Servicerolle](#) im AWS CodeBuild-Benutzerhandbuch für CodeBuild.

Note

Sie benötigen außerdem Berechtigungen für das Ausführen von Aktionen in CodeBuild selbst. Die von Elastic Beanstalk `AWSElasticBeanstalkFullAccess` verwaltete Benutzerrichtlinie enthält alle erforderlichen CodeBuild-Aktionsberechtigungen. Wenn Sie die verwaltete Richtlinie nicht verwenden, müssen Sie in Ihrer Benutzerrichtlinie die folgenden Berechtigungen erteilen:

```
"codebuild:CreateProject",
"codebuild:DeleteProject",
"codebuild:BatchGetBuilds",
"codebuild:StartBuild"
```

Details dazu finden Sie unter [Elastic Beanstalk-Benutzerrichtlinien verwalten \(p. 926\)](#).

ComputeType

Die Zahl der Ressourcen, die vom Docker-Container in der CodeBuild-Erstellungsumgebung verwendet werden. Gültige Werte sind `BUILD_GENERAL1_SMALL`, `BUILD_GENERAL1_MEDIUM` und `BUILD_GENERAL1_LARGE`.

Image

Der Name des Docker-Hubs oder Amazon ECR-Images, das CodeBuild für die Build-Umgebung verwendet. Dieses Docker-Image sollte alle Tools und Laufzeitbibliotheken enthalten, die zum Erstellen Ihres Codes erforderlich sind, und sollte mit der Zielplattform Ihrer Anwendung übereinstimmen. CodeBuild verwaltet und pflegt einen Satz von Images, die speziell für die Verwendung mit Elastic Beanstalk vorgesehen sind. Es wird empfohlen, eines dieser Images zu verwenden. Weitere Einzelheiten finden Sie unter [Von CodeBuild bereitgestellte Docker-Images](#) im AWS CodeBuild-Benutzerhandbuch.

Der `Image`-Wert ist optional. Wenn Sie diesen nicht angeben, wählt der Befehl `eb init` das am besten mit Ihrer Zielplattform übereinstimmende Image aus. Wenn Sie `eb init` im interaktiven Modus ausführen und der Befehl kein Image auswählen kann, werden Sie aufgefordert, ein Image auszuwählen. Am Ende einer erfolgreichen Initialisierung schreibt `eb init` das ausgewählte Image in die Datei `buildspec.yml`.

Timeout

Die Dauer (in Minuten), in der der CodeBuild-Build ausgeführt wird, bevor ein Timeout erfolgt. Dieser Wert ist optional. Details zu gültigen und Standardwerten finden Sie unter [Erstellen eines Build-Projekts in CodeBuild](#).

Note

Diese Zeitüberschreitung steuert die maximale Dauer einer CodeBuild-Ausführung. Die EB CLI berücksichtigt sie als Teil des ersten Schritts bei der Erstellung einer Anwendungsversion. Es handelt sich nicht um den Wert, den Sie mittels der Option `--timeout` der Befehle `eb create (p. 1053)` oder `eb deploy (p. 1063)` angeben können. Dieser Wert steuert die Zeit, die die EB CLI auf die Erstellung oder Aktualisierung einer Umgebung maximal warten soll.

Erstellen und Bereitstellen Ihres Anwendungscodes

Wenn Ihr Anwendungscode bereitgestellt werden muss, verwendet die EB CLI, um mit CodeBuild einen Build auszuführen, und stellt anschließend die so erstellten Build-Artefakte in Ihrer Umgebung bereit. Dies ist bei Erstellung einer Elastic Beanstalk-Umgebung für Ihre Anwendung über den Befehl `eb create (p. 1053)` und bei jeder späteren Bereitstellung von Codeänderungen in der Umgebung über den Befehl `eb deploy (p. 1063)` der Fall.

Wenn der CodeBuild-Schritt fehlschlägt, wird die Erstellung oder Bereitstellung der Umgebung nicht gestartet.

Verwenden der EB CLI mit Git

Die EB CLI bietet Integration in Git. In diesem Abschnitt finden Sie eine Übersicht über die Verwendung von Git mit der EB CLI.

So installieren Sie Git und initialisieren Ihr Git-Repository

1. Laden Sie die neueste Version von Git unter <http://git-scm.com> herunter.
2. Initialisieren Sie Ihr Git-Repository, indem Sie den folgenden Befehl eingeben:

```
~/eb$ git init
```

EB CLI wird nun erkennen, dass Ihre Anwendung mit Git eingerichtet wurde.

3. Wenn Sie eb init noch nicht ausgeführt haben, tun Sie dies jetzt:

```
~/eb$ eb init
```

Zuordnen von Git-Verzweigungen zu Elastic Beanstalk-Umgebungen

Sie können jede Verzweigung Ihres Codes einer anderen Umgebung zuordnen. Wenn Sie eine Verzweigung abschließen, werden Änderungen für die zugeordnete Umgebung bereitgestellt. Sie können beispielsweise Folgendes eingeben, um Ihre Produktionsumgebung Ihrer Hauptverzweigung, und eine separate Entwicklungsumgebung Ihrer Entwicklungsverzweigung zuzuordnen:

```
~/eb$ git checkout master
~/eb$ eb use prod
~/eb$ git checkout develop
~/eb$ eb use dev
```

Bereitstellen von Änderungen

Standardmäßig stellt die EB CLI den neuesten Commit in der aktuellen Verzweigung bereit, wobei die Commit-ID und Nachricht als die Bezeichnung bzw. Beschreibung der Anwendungsversion verwendet wird. Wenn Sie ohne Commit in Ihrer Umgebung bereitstellen möchten, können Sie die --staged-Option verwenden, um Änderungen bereitzustellen, die zum Staging-Bereich hinzugefügt wurden.

So stellen Sie Änderungen ohne Commit bereit

1. Fügen Sie neue und geänderte Dateien zum Staging-Bereich hinzu:

```
~/eb$ git add .
```

2. Bereitstellen von zwischengespeicherten Änderungen mit eb deploy:

```
~/eb$ eb deploy --staged
```

Wenn Sie die EB CLI so konfiguriert haben, dass [ein Artefakt bereitgestellt wird \(p. 1019\)](#), und Sie das Artefakt nicht in Ihrem git-Repository bereitstellen, verwenden Sie die --staged-Option zur Bereitstellung des neuesten Builds.

Verwenden von Git-Submodulen

Einige Code-Projekte profitieren davon, dass sie Git-Submodul-Repositorys innerhalb des Top-Level-Repository haben. Wenn Sie Ihren Code mithilfe von eb create oder eb deploy bereitstellen, kann die EB CLI Submodule in die ZIP-Datei der Anwendungsversion einfügen und sie mit den Rest des Codes hochladen.

Sie können die Einbeziehung von Submodulen kontrollieren, indem Sie die Option `include_git_submodules` im Abschnitt `global` der EB CLI-Konfigurationsdatei `.elasticbeanstalk/config.yml` in Ihrem Projektordner verwenden.

Um die Submodule einzubeziehen, legen Sie diese Option auf `true` fest:

```
global:  
  include_git_submodules: true
```

Wenn die Option `include_git_submodules` fehlt oder auf `false` festgelegt ist, bezieht EB CLI keine Submodule in die hochgeladene ZIP-Datei ein.

Weitere Informationen über Git-Submodule finden Sie unter [Git Tools - Submodules](#).

Standardverhalten

Wenn Sie eb init zum Konfigurieren Ihres Projekts ausführen, fügt die EB CLI die Option `include_git_submodules` hinzu und legt sie auf `true` fest. Auf diese Weise wird sichergestellt, dass alle Submodule, die Sie in Ihrem Projekt haben, in Ihre Bereitstellungen einbezogen werden.

Die EB CLI hat die Einbeziehung von Submodulen nicht immer unterstützt. Um eine versehentliche und unerwünschte Änderung an vorher bereits vorhandenen Projekten zu verhindern, haben wir Submodul-Support hinzugefügt. Die EB CLI bezieht keine Submodule ein, wenn die Option `include_git_submodules` fehlt. Wenn Sie eines dieser vorhandenen Projekte haben und Sie Submodule in Ihren Bereitstellungen einbeziehen möchten, fügen Sie die Option hinzu und legen Sie sie auf `true` fest, wie in diesem Abschnitt erklärt.

CodeCommit-Verhalten

Die Integration von Elastic Beanstalk mit [CodeCommit \(p. 1028\)](#) unterstützt derzeit keine Submodule. Wenn Sie Ihre Umgebung für die Integration mit CodeCommit aktiviert haben, werden Submodule nicht in Ihren Bereitstellungen einbezogen.

Zuweisen von Git-Tags zu Ihrer Anwendungsversion

Sie können ein Git-Tag als Versionsbezeichnung verwenden, um zu identifizieren, welche Anwendungsversion in Ihrer Umgebung ausgeführt wird. Geben Sie zum Beispiel Folgendes ein:

```
~/eb$ git tag -a v1.0 -m "My version 1.0"
```

Verwenden der EB CLI mit AWS CodeCommit

Sie können die EB CLI verwenden, um Ihre Anwendung direkt aus dem AWS CodeCommit-Repository bereitzustellen. Mit CodeCommit können Sie bei der Bereitstellung nur Ihre Änderungen und nicht das gesamte Projekt in das Repository hochladen. Auf diese Weise sparen Sie Zeit und Bandbreite, wenn Sie ein großes Projekt oder eine eingeschränkte Internetverbindung haben. Die EB CLI überträgt Ihre lokalen Commits und nutzt sie zum Erstellen von Anwendungsversionen, wenn Sie eb create oder eb deploy verwenden.

Um Ihre Änderungen bereitzustellen, müssen Sie für die CodeCommit-Integration zuerst die Änderungen bestätigen. Wenn Sie jedoch entwickeln oder debuggen möchten Sie möglicherweise Änderungen, die Sie nicht bestätigt haben, übermitteln. Mithilfe von Staging und eb deploy --staged (führt eine Standard-Bereitstellung durch) können Sie verhindern, dass Ihre Änderungen bestätigt werden. Sie können Ihre Änderungen auch zuerst in eine Entwicklungs- oder Testverzweigung bestätigen sie erst in Ihrer Masterverzweigung zusammenführen, wenn Ihr Code bereit ist. Mit eb use können Sie die EB-Befehlszeilschnittstelle so konfigurieren, dass von Ihrer Entwicklungsverzweigung in eine und von Ihrer Masterverzweigung in eine andere Umgebung bereitgestellt wird.

Note

CodeCommit wird nicht in allen Regionen angeboten. Die Integration zwischen Elastic Beanstalk und CodeCommit funktioniert in diesen Regionen nicht.

Weitere Informationen zu den AWS-Services, die in den einzelnen Regionen angeboten werden, finden Sie in der [Tabelle der Regionen](#).

Abschnitte

- [Voraussetzungen \(p. 1029\)](#)
- [Erstellen eines CodeCommit-Repositorys mit der EB CLI \(p. 1029\)](#)
- [Bereitstellen von Ihrem CodeCommit-Repository aus \(p. 1030\)](#)
- [Konfigurieren von zusätzlichen Verzweigungen und Umgebungen \(p. 1031\)](#)
- [Verwenden eines vorhandenen CodeCommit-Repositorys \(p. 1032\)](#)

Voraussetzungen

Zur Verwendung von CodeCommit mit AWS Elastic Beanstalk benötigen Sie ein lokales Git-Repository (entweder ein bestehendes oder ein neues) mit mindestens einem Commit, [der Berechtigung zur Nutzung von CodeCommit](#) und eine Elastic Beanstalk-Umgebung in einer Region, die CodeCommit unterstützt. Ihre Umgebung und das Repository müssen sich in derselben Region befinden.

So initialisieren Sie ein Git-Repository

1. Führen Sie git init in Ihrem Projektordner aus.

```
~/my-app$ git init
```

2. Erstellen Sie Stufen für Ihre Projektdateien mit git add.

```
~/my-app$ git add .
```

3. Übergeben Sie Änderung mit git commit.

```
~/my-app$ git commit -m "Elastic Beanstalk application"
```

Erstellen eines CodeCommit-Repositorys mit der EB CLI

Um mit CodeCommit zu beginnen, führen Sie [eb init \(p. 1067\)](#) aus. Während der Repository-Konfiguration werden Sie von der EB CLI aufgefordert, CodeCommit zu verwenden, um Ihren Code zu speichern und Bereitstellungen zu beschleunigen. Auch wenn Sie Ihr Projekt zuvor mit eb init konfiguriert haben, können Sie es erneut zum Konfigurieren von CodeCommit ausführen.

So erstellen Sie ein CodeCommit-Repository mit der EB CLI

1. Führen Sie eb init in Ihrem Projektordner aus. Während der Konfiguration werden Sie von der EB CLI gefragt, ob Sie CodeCommit verwenden möchten, um Ihren Code zu speichern und Bereitstellungen zu beschleunigen. Auch wenn Sie Ihr Projekt zuvor mit eb init konfiguriert haben, können Sie es trotzdem erneut zum Konfigurieren von CodeCommit ausführen. Geben Sie **y** bei der Eingabeaufforderung ein, um CodeCommit einzurichten.

```
~/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. Wählen Sie Create new Repository (Neues Repository erstellen) aus.

```
Select a repository
1) my-repo
2) [ Create new Repository ]
(default is 2): 2
```

3. Geben Sie einen Repository-Namen ein oder drücken Sie die Eingabetaste zum Übernehmen des Standardnamens.

```
Enter Repository Name
(default is "codecommit-origin"): my-app
Successfully created repository: my-app
```

4. Wählen Sie eine vorhandene Verzweigung für Ihre Commits oder verwenden Sie die EB-Befehlszeilenschnittstelle zum Erstellen einer neuen Verzweigung.

```
Enter Branch Name
***** Must have at least one commit to create a new branch with CodeCommit *****
(default is "master"): ENTER
Successfully created branch: master
```

Bereitstellen von Ihrem CodeCommit-Repository aus

Wenn Sie CodeCommit mit Ihrem EB CLI-Repository konfigurieren, verwendet die EB CLI den Inhalt des Repository zum Erstellen der Quell-Bundles. Wenn Sie eb deploy oder eb create ausführen, überträgt die EB CLI neue Commits und verwendet die HEAD-Revision Ihrer Verzweigung für die Erstellung des Archivs, das in den EC2-Instances in Ihrer Umgebung bereitgestellt wird.

So verwenden Sie die CodeCommit-Integration mit der EB CLI

1. Erstellen Sie eine neue Umgebung mit 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.
...
```

Die EB CLI nutzt den aktuellen Commit in der nachverfolgten Verzweigung zum Erstellen der Anwendungsversion, die in der Umgebung bereitgestellt wird.

2. Wenn Sie neue lokale Commits haben, verwenden Sie eb deploy, um diese zu übertragen und in Ihrer Umgebung bereitzustellen.

```
~/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. Um die Änderungen zu testen, bevor Sie sie bestätigen, verwenden Sie die --staged-Option zum Bereitstellen von Änderungen, die Sie mit git add zum Staging-Bereich hinzugefügt haben.

```
~/my-app$ git add new-file
~/my-app$ eb deploy --staged
```

Durch das Bereitstellen mit der --staged-Option wird eine Standardbereitstellung durchgeführt und CodeCommit umgangen.

Konfigurieren von zusätzlichen Verzweigungen und Umgebungen

Die CodeCommit-Konfiguration gilt für eine einzelne Verzweigung. Sie können eb use und eb codesource verwenden, um zusätzliche Verzweigungen zu konfigurieren oder die Konfiguration der aktuellen Verzweigung zu ändern.

So konfigurieren Sie die CodeCommit-Integration mit der EB CLI

1. Zum Ändern der Remote-Verzweigung verwenden Sie die --source-Option des Befehls [eb use \(p. 1097\)](#).

```
~/my-app$ eb use test-env --source my-app/test
```

2. Um eine neue Verzweigung und Umgebung zu erstellen, checken Sie eine neue Verzweigung aus, übertragen Sie sie an CodeCommit, erstellen Sie die Umgebung und verbinden Sie die lokale Verzweigung, Remote-Verzweigung und Umgebung dann mit eb use.

```
~/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. Zum interaktiven Konfigurieren von CodeCommit verwenden Sie [eb codesource codecommit \(p. 1049\)](#).

```
~/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) master
2) test
3) [ Create new Branch with local HEAD ]
(default is 1): 1
```

- Zum Deaktivieren der CodeCommit-Integration verwenden Sie [eb codesource local \(p. 1049\)](#).

```
~/my-app$ eb codesource local
Current CodeCommit setup:
  Repository: my-app
  Branch: master
Default set to use local sources
```

Verwenden eines vorhandenen CodeCommit-Repositorys

Wenn Sie bereits ein CodeCommit-Repository haben und es mit Elastic Beanstalk verwenden möchten, führen Sie eb init am Stamm des lokalen Git-Repositorys aus.

So verwenden Sie ein vorhandenes CodeCommit-Repository mit der EB CLI

- Klonen Sie Ihr CodeCommit-Repository.

```
~$ git clone ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/my-app
```

- Checken Sie eine Verzweigung aus und übertragen Sie sie zur Verwendung mit der Elastic Beanstalk-Umgebung.

```
~/my-app$ git checkout -b dev-env
~/my-app$ git push --set-upstream origin dev-env
```

- Führen Sie eb init. Wählen Sie für Region, Repository und Verzweigung den Namen, den Sie derzeit verwenden.

```
~/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) master
2) dev-env
3) [ Create new Branch with local HEAD ]
(default is 2): 2
```

Weitere Informationen zur Verwendung von eb init finden Sie unter [Konfigurieren der EB CLI \(p. 1017\)](#).

Verwenden der EB-CLI zur Überwachung des Umgebungszustands

[Elastic Beanstalk Command Line Interface \(Befehlszeilenschnittstelle\) \(p. 1008\)](#) (EB CLI) ist ein Befehlszeilen-Tool für die Verwaltung von AWS Elastic Beanstalk-Umgebungen. Sie können die EB CLI auch zum Überwachen des Zustands der Umgebung in Echtzeit einsetzen. Dabei bietet sie mehr Granularität, als derzeit in der Elastic Beanstalk-Konsole verfügbar ist.

Nach der [Installation \(p. 1009\)](#) und [Konfiguration \(p. 1017\)](#) der EB CLI können Sie [eine neue Umgebung starten \(p. 1020\)](#) und Ihren Code mit dem eb create-Befehl für sie bereitstellen. Wenn Sie bereits über eine Umgebung verfügen, die Sie in der Elastic Beanstalk-Konsole erstellt haben, können Sie die EB CLI anfügen, indem Sie eb init in einem Projektordner ausführen und die Anweisungen auf dem Bildschirm befolgen (der Projektordner kann leer sein).

Important

Stellen Sie sicher, dass Sie die neueste Version der EB CLI verwenden, indem Sie pip install mit der --upgrade-Option ausführen:

```
$ sudo pip install --upgrade awsebcli
```

Vollständige Anweisungen zur EB CLI-Installation finden Sie unter [Installieren der EB CLI \(p. 1009\)](#).

Zum Verwenden der EB CLI für die Zustandsüberwachung Ihrer Umgebung müssen Sie zuerst einen lokalen Projektordner konfigurieren, indem Sie eb init ausführen und die Anweisungen befolgen. Vollständige Anweisungen finden Sie unter [Konfigurieren der EB CLI \(p. 1017\)](#).

Wenn Sie bereits über eine Umgebung in Elastic Beanstalk verfügen und die EB CLI für ihre Zustandsüberwachung verwenden möchten, fügen Sie sie mit diesem Verfahren die vorhandene Umgebung an.

So fügen Sie die EB CLI an eine vorhandene Umgebung an

1. Öffnen Sie ein Befehlszeilen-Terminal und navigieren Sie zu Ihrem Benutzerordner.
2. Erstellen und öffnen Sie einen neuen Ordner für Ihre Umgebung.
3. Führen Sie den eb init-Befehl aus und wählen Sie dann die Anwendung und die Umgebung, deren Zustand Sie überwachen möchten. Wenn Sie nur eine Umgebung haben, auf der die ausgewählte

Anwendung ausgeführt wird, wählt die EB CLI diese automatisch aus, und Sie müssen die Umgebung nicht auswählen, wie im folgenden Beispiel gezeigt.

```
~/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
```

So überwachen Sie den Zustand mit der EB CLI

1. Öffnen Sie eine Befehlszeile und navigieren Sie zu Ihrem Projektordner.
2. Führen Sie den eb health-Befehl zum Anzeigen des Zustands der Instances in Ihrer Umgebung aus. In diesem Beispiel gibt es fünf Instances, die in einer Linux-Umgebung ausgeführt werden.

```
~/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        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
  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%
  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							
<hr/>								
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			

In diesem Beispiel gibt es eine Instance, die in einer Windows-Umgebung ausgeführt wird.

```

~/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         0

  instance-id           status       cause
                                         health
                                         Overall      Ok
i-065716fba0e08a351      Ok

  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-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
    i-065716fba0e08a351 t2.large  1b    4 hours            0.2                      0.1
99.7

  instance-id           status      id    version
                                         deployments
                                         ago
    i-065716fba0e08a351 Deployed    2    Sample Application    4 hours

```

Lesen der Ausgabe

Die Ausgabe zeigt den Namen der Umgebung, den Gesamtzustand der Umgebung und das aktuelle Datum oben im Bildschirm an.

```
elasticBeanstalkExa-env          Ok  
2015-07-08 23:13:20
```

Die nächsten drei Zeilen enthalten die Art der Umgebung (in diesem Fall "WebServer"), die Konfiguration (in diesem Fall Ruby 2.1 mit Puma) und eine Übersicht darüber, wie viele Instances sich in jedem der sieben Zustände befinden.

Der Rest der Ausgabe ist in vier Abschnitte unterteilt. Der erste zeigt den Status und den Grund des Status für die gesamte Umgebung und anschließend für jede Instance an. Das folgende Beispiel zeigt zwei Instances in der Umgebung mit dem Status `Info` und einem Grund, der besagt, dass eine Bereitstellung gestartet wurde.

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	

Weitere Informationen über Zustand und Farben finden Sie unter [Farben und Status in Zustandsangaben \(p. 841\)](#).

Der Abschnitt `requests` enthält Informationen aus den Webserverprotokollen zu jeder Instance. In diesem Beispiel nimmt jede Instance Anfragen normal an und es gibt keine Fehler.

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	2.4	100.0	0.0	0.0	0.0	1.102*	0.865	0.601	0.413
0.091	2.7	100.0	0.0	0.0	0.0	0.842*	0.788	0.480	0.305
0.062	4.1	100.0	0.0	0.0	0.0	1.520*	1.088	0.883	0.524
0.104	2.2	100.0	0.0	0.0	0.0	1.334*	0.791	0.760	0.344
0.197	2.3	100.0	0.0	0.0	0.0	1.162*	0.867	0.698	0.477
0.076									

Der Abschnitt `cpu` enthält Betriebssystemmetriken für jede Instance. Die Ausgabe unterscheidet sich je nach Betriebssystem. Hier sehen Sie die Ausgabe für Linux-Umgebungen.

instance-id	type	az	running	load 1	load 5	user%	nice%	system%	idle
% iowait%									
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								

Und hier die Ausgabe für Windows-Umgebungen.

instance-id	type	az	running	% user time	% privileged time	%
idle time						
i-065716fba0e08a351	t2.large	1b	4 hours	0.2		0.0
99.8						

Weitere Informationen über die dargestellten Server- und Betriebssystemmetriken finden Sie unter [Instance-Metriken \(p. 843\)](#).

Der letzte Abschnitt deployments zeigt den Bereitstellungsstatus jeder Instance. Wenn eine fortlaufende Bereitstellung fehlschlägt, können Sie die Bereitstellung-ID, den Status und die angezeigte Versionsbezeichnung verwenden, um Instances in Ihrer Umgebung zu identifizieren, die mit der falschen Version ausgeführt werden.

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

Interaktive Ansicht des Zustands

Der eb health-Befehl zeigt einen Snapshot des Zustands Ihrer Umgebung. Um die angezeigten Informationen alle zehn Sekunden zu aktualisieren, verwenden Sie die --refresh-Option.

\$ 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
p10				requests					
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	0.0
99.7 0.1									
i-ba65c144	t2.micro	1a	12 mins	0.0	0.03	0.3	0.0	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	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: Help,Quit, # # # #

```

Das folgende Beispiel zeigt eine Umgebung, die kürzlich von einer auf fünf Instances hochskaliert wurde. Der Skalierungsvorgang wurde erfolgreich abgeschlossen und alle Instances bestehen nun die Zustandsprüfungen und können Anfragen annehmen. Im interaktiven Modus wird der Zustandsstatus alle zehn Sekunden aktualisiert. In der rechten oberen Ecke zählt ein Timer bis zum nächsten Update herunter.

In der linken unteren Ecke zeigt der Bericht eine Liste der Optionen an. Um den interaktiven Modus zu verlassen, drücken Sie Q. Drücken Sie die Pfeiltasten, um zu scrollen. Für eine Liste der zusätzlichen Befehle drücken Sie H.

Optionen für die Interaktive Ansicht des Zustands

Bei der interaktiven Anzeige des Zustands der Umgebung können Sie die Ansicht über die Tastatur anpassen und Elastic Beanstalk anweisen, einzelne Instances zu ersetzen oder neu zu starten. Für eine Liste der verfügbaren Befehle während der Anzeige der Zustandsberichte im interaktiven Modus drücken Sie 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)

Verwalten mehrerer Elastic Beanstalk-Umgebungen als Gruppe mit der EB CLI

Sie können mit der EB-CLI Gruppen von AWS Elastic Beanstalk-Umgebungen erstellen, die jeweils eine separate Komponente einer Service-orientierten Architekturanwendung ausführen. Die EB-CLI verwaltet solche Gruppen unter Verwendung der [ComposeEnvironments](#)-API.

Note

Umgebungsgruppen unterscheiden sich von dem Konzept mehrerer Container in einer Multicontainer Docker-Umgebung. Mit Umgebungsgruppen wird jede Komponente Ihrer Anwendung in einer separaten Elastic Beanstalk-Umgebung ausgeführt, mit jeweils einem eigenen Satz von Amazon EC2-Instances. Jede Komponente kann separat skaliert werden. Mit Multicontainer Docker kombinieren Sie mehrere Komponenten einer Anwendung in einer einzigen Umgebung. Alle Komponenten verwenden den gleichen Satz an Amazon EC2-Instances, wobei jede Instance mehrere Docker-Container ausführt. Wählen Sie eine dieser Architekturen entsprechend den Anforderungen Ihrer Anwendung aus.

Details zu Multicontainer Docker finden Sie unter [Verwenden der Multicontainer-Docker-Plattform \(Amazon Linux AMI\) \(p. 64\)](#).

Organisieren Sie Ihre Anwendungskomponenten in der folgenden Ordnerstruktur:

```
~/project-name
|-- component-a
|   `-- env.yaml
`-- component-b
    `-- env.yaml
```

Jeder Unterordner enthält den Quellcode für eine unabhängige Komponente einer Anwendung, die in ihrer eigenen Umgebung ausgeführt, sowie eine Umgebungsdefinitionsdatei mit dem Namen `env.yaml`. Weitere Informationen zum Format `env.yaml` finden Sie unter [Umgebungsmanifest \(`env.yaml`\) \(p. 771\)](#).

Zur Verwendung der `Compose Environments`-API führen Sie zuerst `eb init` aus dem Projektordner aus und geben Sie die einzelnen Komponenten mit dem Namen des Ordners an, der sie mit der `--modules`-Option enthält:

```
~/workspace/project-name$ eb init --modules component-a component-b
```

Sie werden von der EB CLI aufgefordert, [jede Komponente zu konfigurieren \(p. 1017\)](#). Anschließend wird das `.elasticbeanstalk`-Verzeichnis in jedem Komponentenordner erstellt. EB CLI erstellt keine Konfigurationsdateien im übergeordneten Verzeichnis.

```
~/project-name
|-- component-a
|   |-- .elasticbeanstalk
|   |   `-- env.yaml
`-- component-b
    |-- .elasticbeanstalk
    |   `-- env.yaml
```

Führen Sie im nächsten Schritt den `eb create`-Befehl mit einer Liste von zu erstellenden Umgebungen aus, einen für jede Komponente:

```
~/workspace/project-name$ eb create --modules component-a component-b --env-group-suffix group-name
```

Mit diesem Befehl wird eine Umgebung für jede Komponente erstellt. Die Namen der Umgebungen werden erstellt, indem der `EnvironmentName`, der in der `env.yaml`-Datei angegeben ist, mit dem Gruppennamen zusammengefügt wird (durch einen Bindestrich getrennt). Die Gesamtlänge dieser beiden Optionen und der Bindestrich dürfen nicht größer als die maximal zulässige Umgebungsnamenlänge von 23 Zeichen.

Verwenden Sie den `eb deploy`-Befehl, um die Umgebung zu aktualisieren:

```
~/workspace/project-name$ eb deploy --modules component-a component-b
```

Sie können jede Komponente einzeln oder als Gruppe aktualisieren. Geben Sie die Komponenten an, die Sie mit der `--modules`-Option aktualisieren möchten.

Die EB CLI speichert den Gruppennamen, den Sie mit `eb create` im `branch-defaults`-Abschnitt der EB CLI-Konfigurationsdatei unter `/elasticbeanstalk/config.yml` verwendet haben. Zum Bereitstellen Ihrer Anwendung für eine andere Gruppe verwenden Sie die `--env-group-suffix`-Option, wenn Sie `eb deploy` ausführen. Wenn die Gruppe noch nicht vorhanden ist, erstellt die EB CLI eine neue Gruppe von Umgebungen:

```
~/workspace/project-name$ eb deploy --modules component-a component-b --env-group-suffix group-2-name
```

Zum Beenden der Umgebungen führen Sie `eb terminate` im Ordner für jedes Modul aus. Standardmäßig zeigt die EB CLI eine Fehlermeldung an, wenn Sie versuchen, eine Umgebung zu beenden, von der eine andere laufende Umgebung abhängt. Beenden Sie zuerst die abhängige Umgebung oder überschreiben Sie mit der `--ignore-links`-Option das Standardverhalten:

```
~/workspace/project-name/component-b$ eb terminate --ignore-links
```

Beheben von EB CLI-Fehlern

In diesem Thema werden häufig auftretende Fehlermeldungen im Zusammenhang mit der EB CLI-Verwendung sowie mögliche Lösungen aufgeführt. Falls Ihnen eine Fehlermeldung angezeigt wird, die hier nicht enthalten ist, teilen Sie uns dies bitte über die Feedback-Links mit.

FEHLER: An error occurred while handling git command. Error code: 128 Error: fatal: Not a valid object name HEAD

Ursache: Diese Fehlermeldung wird angezeigt, wenn Sie ein Git-Repository initialisiert, aber noch nicht übergeben haben. Wenn der Projektordner ein Git-Repository enthält, sucht die EB CLI nach einer HEAD-Revision.

Lösung: Fügen Sie die Dateien in Ihrem Projektordner zum Staging-Bereich hinzu und führen Sie den Commit aus:

```
~/my-app$ git add .
~/my-app$ git commit -m "First commit"
```

FEHLER: This branch does not have a default environment. You must either specify an environment by typing "eb status my-env-name" or set a default environment by typing "eb use my-env-name".

Ursache: Wenn Sie in Git eine neue Verzweigung erstellen, wird dieser nicht standardmäßig einer Elastic Beanstalk-Umgebung zugewiesen.

Lösung: Führen Sie `eb list` aus, um eine Liste der verfügbaren Umgebungen aufzurufen. Führen Sie anschließend `eb use env-name` aus, um eine der verfügbaren Umgebungen zu verwenden.

FEHLER: 2.0+ Platforms require a service role. You can provide one with --service-role option

Ursache: Wenn Sie einen Umgebungsnamen mit `eb create` angeben (z. B. `eb create my-env`), erstellt die EB CLI keine Servicerolle für Sie. Ist die standardmäßige Servicerolle nicht vorhanden, wird die obige Fehlermeldung angezeigt.

Lösung: Führen Sie `eb create` ohne den Umgebungsnamen aus und folgen Sie den Anweisungen, um die standardmäßige Servicerolle zu erstellen.

Beheben von Bereitstellungsfehlern

Falls die Elastic Beanstalk-Bereitstellung nicht so problemlos wie geplant ausgeführt wurde, sehen Sie anstelle Ihrer Website möglicherweise die Fehlermeldung 404 (wenn die Anwendung nicht gestartet werden konnte) oder 500 (wenn die Anwendung zur Laufzeit ausgefallen ist). Zahlreiche häufig auftretende Fehler können Sie beheben, indem Sie mit der EB CLI den Status der Bereitstellung prüfen, die Protokolle anzeigen, über SSH auf die EC2-Instance zugreifen oder für die Anwendungsumgebung die entsprechende Seite in der AWS Management Console aufrufen.

So setzen Sie die EB CLI zum Beheben von Bereitstellungsfehlern ein

1. Führen Sie eb status aus, um den Status der aktuellen Bereitstellung sowie den Zustand der EC2-Hosts anzuzeigen. Beispiel:

```
$ 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

Mit --verbose erhalten Sie detaillierte Informationen über den Status der ausgeführten Instances. Andernfalls werden nur allgemeine Umgebungsinformationen von eb status ausgegeben.

2. Führen Sie eb health aus, um Zustandsinformationen zur Umgebung anzuzeigen:

```
$ 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         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				

Im obigen Beispiel wird eine Umgebung mit fünf Instances dargestellt, in der die Bereitstellung der Version "v2" auf der dritten Instance fehlerhaft verlaufen ist. Nach der fehlgeschlagenen Bereitstellung wird die erwartete Version auf die letzte erfolgreiche Version zurückgesetzt, in diesem Falle auf "Sample Application" aus der ersten Bereitstellung. Weitere Informationen finden Sie unter [Verwenden der EB-CLI zur Überwachung des Umgebungszustands \(p. 1033\)](#).

- Führen Sie eb logs aus, um die Protokolle zur Anwendungsbereitstellung herunterzuladen und aufzurufen.

```
$ eb logs
```

- Führen Sie eb ssh aus, um die Verbindung zur EC2-Instance, auf der die Anwendung ausgeführt wird, herzustellen und diese direkt zu überprüfen. Auf der Instance finden Sie die bereitgestellte Anwendung im Verzeichnis /opt/python/current/app und die Python-Umgebung in /opt/python/run/venv/.
- Führen Sie eb console aus, um die Anwendungsumgebung in der [AWS-Managementkonsole](#) anzuzeigen. Über die Weboberfläche können Sie schnell zahlreiche Bereitstellungsaspekte untersuchen, darunter Konfiguration, Status, Ereignisse und Protokolle der Anwendung. Außerdem können Sie die aktuellen oder beendeten Anwendungsversionen herunterladen, die Sie auf dem Server bereitgestellt haben.

EB-CLI-Befehlsreferenz

Mit der Elastic Beanstalk-Befehlszeilenschnittstelle (EB CLI) können Sie zahlreiche Vorgänge für die Bereitstellung und Verwaltung Ihrer Elastic Beanstalk-Anwendungen und -Umgebungen ausführen. Die EB CLI kann in Git integriert werden, sodass Sie Anwendungsquellcode mit der Git-Quellüberwachung bereitstellen können. Weitere Informationen finden Sie unter [Verwenden des Elastic Beanstalk Command Line Interface \(EB CLI\) \(p. 1008\)](#) und [Verwenden der EB CLI mit Git \(p. 1027\)](#).

Befehle

- [eb abort \(p. 1043\)](#)
- [eb appversion \(p. 1044\)](#)
- [eb clone \(p. 1047\)](#)
- [eb codesource \(p. 1049\)](#)
- [eb config \(p. 1050\)](#)
- [eb console \(p. 1052\)](#)
- [eb create \(p. 1053\)](#)
- [eb deploy \(p. 1063\)](#)
- [eb events \(p. 1064\)](#)
- [eb health \(p. 1066\)](#)
- [eb init \(p. 1067\)](#)
- [eb labs \(p. 1070\)](#)
- [eb list \(p. 1070\)](#)
- [eb local \(p. 1071\)](#)
- [eb logs \(p. 1074\)](#)
- [eb open \(p. 1076\)](#)
- [eb platform \(p. 1077\)](#)
- [eb printenv \(p. 1084\)](#)
- [eb restore \(p. 1085\)](#)
- [eb scale \(p. 1086\)](#)
- [eb setenv \(p. 1086\)](#)
- [eb ssh \(p. 1087\)](#)
- [eb status \(p. 1089\)](#)
- [eb swap \(p. 1091\)](#)
- [eb tags \(p. 1092\)](#)
- [eb terminate \(p. 1094\)](#)
- [eb upgrade \(p. 1096\)](#)
- [eb use \(p. 1097\)](#)
- [Allgemeine Optionen \(p. 1097\)](#)

eb abort

Beschreibung

Bricht ein Upgrade ab, während die Umgebungskonfigurationsänderungen für die Instances noch ausgeführt werden.

Note

Falls für mehr als zwei Umgebungen eine Aktualisierung ausgeführt wird, werden Sie aufgefordert, den Namen der Umgebung anzugeben, für die Änderungen zurückgesetzt werden sollen.

Syntax

eb abort

eb abort **environment-name**

Optionen

Name	Beschreibung
Allgemeine Optionen (p. 1097)	

Ausgabe

Mit dem Befehl wird eine Liste der Umgebungen angezeigt, die gerade aktualisiert werden. Zudem werden Sie aufgefordert, die abzubrechende Aktualisierung auszuwählen. Sofern derzeit nur eine Umgebung aktualisiert wird, muss kein Umgebungsname angegeben werden. Bei einer erfolgreichen Ausführung werden mit diesem Befehl die Umgebungskonfigurationsänderungen zurückgesetzt. Der Rollbackvorgang wird fortgesetzt, bis alle Instances der Umgebung wieder die vorherige Umgebungskonfiguration aufweisen oder bis im Rollback Fehler auftreten.

Beispiel

Mit dem folgenden Beispiel wird die Plattformaktualisierung abgebrochen.

```
$ eb abort
Aborting update to environment "tmp-dev".
<list of events>
```

eb appversion

Beschreibung

Verwaltet Ihre Elastic Beanstalk-Anwendungsversionen, z. B. das Löschen einer Version der Anwendung oder das Erstellen der Anwendungsversions-Lebenszyklus-Richtlinie. Wenn Sie den Befehl ohne Optionen aufrufen, wird dieser in den [interaktiven Modus \(p. 1045\)](#) versetzt.

Verwenden Sie die `--delete`-Option zum Löschen einer Version der Anwendung.

Verwenden Sie die `lifecycle`-Option zum Anzeigen oder Erstellen der Anwendungsversions-Lebenszyklus-Richtlinie. Weitere Informationen finden Sie unter [the section called “Lebenszyklus der Version” \(p. 408\)](#).

Syntax

eb appversion

eb appversion [-d | --delete] **version-label**

eb appversion lifecycle [-p | --print]

Optionen

Name	Beschreibung
-d version-label	Löschen Sie die Version version-label der Anwendung.

Name	Beschreibung
oder --delete <i>version-label</i>	
Lebenszyklus	Rufen Sie den Standard-Editor zum Erstellen einer neuen Anwendungsversions-Lebenszyklus-Richtlinie auf. Verwenden Sie diese Richtlinie, um die Grenze für das Kontingent der Anwendungsversion zu vermeiden.
lifecycle -p oder lifecycle --print	Zeigen Sie die aktuelle Anwendungs-Lebenszyklus-Richtlinie an.
Allgemeine Optionen (p. 1097)	

Interaktives Verwenden des Befehls

Der Befehl ohne Argumente zeigt die Versionen der Anwendung an, von der neuesten bis zu ältesten. Beispiele dazu, wie der Bildschirm aussieht, finden Sie im Abschnitt Examples (Beispiele). Beachten Sie die Statuszeile am Ende der Anzeige. Sie enthält kontextbezogene Informationen, die Ihnen helfen können.

Drücken Sie d zum Löschen einer Anwendungsversion, drücken Sie l zum Verwalten der Lebenszyklus-Richtlinie für Ihre Anwendung oder drücken Sie q, um den Vorgang zu beenden, ohne Änderungen vorzunehmen.

Note

Wenn die Version in einer beliebigen Umgebung bereitgestellt wird, können Sie diese Version nicht löschen.

Ausgabe

Der Befehl mit der --delete *version-label*-Option zeigt eine Meldung an, in der bestätigt wird, dass die Anwendungsversion gelöscht wurde.

Beispiele

Das folgende Beispiel zeigt das interaktive Fenster für eine Anwendung ohne Bereitstellungen.

```
No Environment Specified
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, ▼▲◀▶)
```

Das folgende Beispiel zeigt das interaktive Fenster für eine Anwendung mit der vierten Version, wobei die Versionsbezeichnung Sample Application (Beispielanwendung) bereitgestellt ist.

```
Sample-env Application
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, ▼▲◀▶)
```

Das folgende Beispiel zeigt die Ausgabe eines eb appversion lifecycle -p-Befehls, wobei **ACCOUNT-ID** die Konto-ID des Benutzers ist:

```
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

Beschreibung

Klont eine Umgebung in einer neuen Umgebung, damit beide identische Umgebungseinstellungen haben.

Note

Standardmäßig erstellt der eb clone-Befehl die Klonumgebung mit dem neuesten Lösungs-Stack, unabhängig von der Lösungs-Stack-Version der Umgebung, aus der Sie den Klon erstellen. Sie können dies unterdrücken, indem Sie die --exact-Option einschließen, wenn Sie den Befehl ausführen.

Syntax

eb clone

eb clone **environment-name**

Optionen

Name	Beschreibung
-n string (Zeichenfolge) oder --clone_name string (Zeichenfolge)	Gewünschter Name für die geklonte Umgebung.
-c string (Zeichenfolge) oder --cname string (Zeichenfolge)	Gewünschtes CNAME-Präfix für die geklonte Umgebung.
--envvars	Umgebungseigenschaften in einer durch Kommas getrennten Liste mit dem Format name (Name)=value (Wert) . Typ: Zeichenkette Einschränkungen: <ul style="list-style-type: none">• Schlüssel-Wert-Paare müssen durch Kommas getrennt werden.• Schlüssel und Werte können alphabetische Zeichen in jeder beliebigen Sprache, numerische Zeichen, Leerzeichen,

Name	Beschreibung
	<ul style="list-style-type: none"> unsichtbare Trennzeichen und die folgenden Symbole sein: _ . : / + \ - @ Schlüssel können bis zu 128 Zeichen enthalten. Werte können bis zu 256 Zeichen enthalten. Bei Schlüsseln und Werten wird die Groß-/Kleinschreibung berücksichtigt. Werte können nicht mit dem Umgebungsnamen übereinstimmen. Werte können nicht <code>aws:</code> oder <code>elasticbeanstalk:</code> enthalten. Die Größe aller Umgebungseigenschaften darf nicht mehr als 4096 Byte sein.
<code>--exact</code>	Verhindert, dass Elastic Beanstalk die Lösungs-Stack-Version für die neue Klonumgebung auf die neueste verfügbare Version aktualisiert (für die Plattform der ursprünglichen Umgebung).
<code>--scale number (Zahl)</code>	Die Anzahl der Instances, die in der Klonumgebung ausgeführt werden sollen, wenn diese gestartet wird.
<code>--tags name (Name)=value (Wert)</code>	Tags (p. 617) für die Ressourcen in Ihrer Umgebung in einer durch Kommas getrennten Liste mit dem Format <code>name (Name)=value (Wert)</code> .
<code>--timeout</code>	Die Anzahl der Minuten, bevor ein Timeout für den Befehl auftritt.
Allgemeine Optionen (p. 1097)	

Ausgabe

Wenn der Befehl erfolgreich war, wird eine Umgebung erstellt, die dieselben Einstellungen wie die ursprüngliche Umgebung hat oder Änderungen aufweist, wie durch eb clone-Optionen angegeben.

Beispiel

Im folgenden Beispiel wird die angegebene Umgebung geklont.

```
$ 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-vjvqrqn5pv
  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

Beschreibung

Mit diesem Befehl wird die EB CLI für die [Bereitstellung aus einem CodeCommit-Repository \(p. 1028\)](#) konfiguriert oder die CodeCommit-Integration wird deaktiviert und das Quell-Bundle wird vom lokalen Rechner hochgeladen.

Note

CodeCommit wird nicht in allen AWS-Regionen angeboten. Die Integration zwischen Elastic Beanstalk und CodeCommit funktioniert in diesen Regionen nicht.
Weitere Informationen zu den AWS-Services, die in den einzelnen Regionen angeboten werden, finden Sie in der [Tabelle der Regionen](#).

Syntax

```
eb codesource
eb codesource codecommit
eb codesource local
```

Optionen

Name	Beschreibung
Allgemeine Optionen (p. 1097)	

Ausgabe

Mit eb codesource werden Sie aufgefordert, die Wahl zwischen einer CodeCommit-Integration und Standardbereitstellungen zu treffen.

Mit eb codesource codecommit wird die interaktive Repository-Konfiguration für die CodeCommit-Integration gestartet.

Mit eb codesource local wird die ursprüngliche Konfiguration angezeigt und die CodeCommit-Integration wird deaktiviert.

Beispiele

Mit eb codesource codecommit konfigurieren Sie die CodeCommit-Integration für die aktuelle Verzweigung.

```
~/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) master
```

```
2) test
3) [ Create new Branch with local HEAD ]
(default is 1): 1
```

Mit eb codesource local deaktivieren Sie die CodeCommit-Integration für die aktuelle Verzweigung.

```
~/my-app$ eb codesource local
Current CodeCommit setup:
  Repository: my-app
  Branch: master
Default set to use local sources
```

eb config

Beschreibung

Ändert die Einstellungen der Umgebungskonfiguration. Mit diesem Befehl werden die Einstellungen der Umgebungskonfiguration sowie Uploads, Downloads oder in Listen gespeicherte Konfigurationen gespeichert.

Wenn das Stammverzeichnis eine `platform.yaml`-Datei enthält, die eine benutzerdefinierte Plattform angibt, ändert dieser Befehl auch die Builder-Konfigurationseinstellungen, basierend auf den Werten, die in `platform.yaml` festgelegt sind.

Note

eb config zeigt keine Umgebungseigenschaften. Zum Festlegen von Umgebungseigenschaften, die Sie in Ihrer Anwendung lesen können, verwenden Sie [eb setenv \(p. 663\)](#).

Syntax

eb config

eb config **environment-name**

In der folgenden Tabelle wird die Syntax für den eb config-Befehl für die Arbeit mit gespeicherten Konfigurationen beschrieben. Beispiele finden Sie im Abschnitt [Beispiele \(p. 1051\)](#) weiter unten in diesem Thema.

- eb config delete **filename** – Löscht die benannte gespeicherte Konfiguration.
- eb config get **filename** – Lädt die benannte gespeicherte Konfiguration herunter.
- eb config list – Listet die Konfigurationen auf, die Sie in Amazon S3 gespeichert haben.
- eb config put **filename** – Lädt die benannte gespeicherte Konfiguration in einen Amazon S3-Bucket hoch. Der **Dateiname** muss die Dateierweiterung `.cfg.yaml` haben. Um den Dateinamen ohne Pfad anzugeben, können Sie die Datei im Ordner `.elasticbeanstalk` oder im Ordner `.elasticbeanstalk/saved_configs/` speichern, bevor Sie den Befehl ausführen. Alternativ können Sie den **Dateinamen** festlegen, indem Sie den vollständigen Pfad angeben.
- eb config save – Speichert die Einstellungen der Umgebungskonfiguration für die aktuell ausgeführte Umgebung in `.elasticbeanstalk/saved_configs/` mit dem Dateinamen `[configuration-name].cfg.yaml`. Standardmäßig speichert die EB CLI Konfigurationseinstellungen mit einem **configuration-name** basierend auf dem Umgebungsnamen. Sie können einen anderen Konfigurationsnamen angeben, indem Sie die `--cfg`-Option in den gewünschten Konfigurationsnamen einschließen, wenn Sie den Befehl ausführen.

Sie können Ihre gespeicherte Konfiguration unter Verwendung der `--tags`-Option mit Tags markieren.

Optionen

Name	Beschreibung
--cfg <i>config-name</i>	Der für eine gespeicherte Konfiguration zu verwendende Name (den Sie zu einem späteren Zeitpunkt angeben können, um eine Umgebung aus einer gespeicherten Konfiguration zu erstellen oder zu aktualisieren). Diese Option funktioniert nur mit dem Befehl eb config save.
--tags <i>key1=value1[,key2=value2]</i>	Tags zum Hinzufügen zu Ihrer gespeicherten Konfiguration. Tags werden als durch Komma getrennte Liste von key=value-Paaren angegeben. Weitere Informationen finden Sie unter Markieren von gespeicherten Konfigurationen (p. 769) . Diese Option funktioniert nur mit dem Befehl eb config save.
--timeout <i>timeout</i>	Die Anzahl der Minuten, bevor ein Timeout für den Befehl auftritt.
Allgemeine Optionen (p. 1097)	

Ausgabe

Wenn der Befehl erfolgreich ohne Parameter ausgeführt wird, zeigt der Befehl Ihre aktuellen Optionseinstellungen im Text-Editor, die Sie als EDITOR-Umgebungsvariable konfiguriert haben. (Wenn Sie keine EDITOR-Umgebungsvariable konfiguriert haben, zeigt die EB CLI Ihre Optionseinstellungen im Standard-Editor für YAML-Dateien Ihres Computers an.) Wenn Sie Änderungen in der Datei speichern und den Editor schließen, wird die Umgebung mit den Optionseinstellungen in der Datei aktualisiert.

Wenn der Befehl erfolgreich mit dem `get`-Parameter ausgeführt wird, zeigt der Befehl den Speicherort der lokalen Kopie, die Sie heruntergeladen haben.

Wenn der Befehl erfolgreich mit dem `save`-Parameter ausgeführt wird, zeigt der Befehl den Speicherort der gespeicherten Datei.

Beispiele

In diesem Abschnitt wird beschrieben, wie Sie den Text-Editor ändern, den Sie zum Anzeigen und Bearbeiten Ihrer Optionseinstellungen verwenden.

Für Linux/UNIX wird im folgenden Beispiel der Editor auf vim geändert:

```
$ export EDITOR=vim
```

Für Linux/UNIX wird im folgenden Beispiel der Editor auf das geändert, was unter `/usr/bin/kate` installiert ist.

```
$ export EDITOR=/usr/bin/kate
```

Für Windows wird im folgenden Beispiel der Editor auf Notepad++ geändert:

```
> set EDITOR="C:\Program Files\Notepad++\Notepad++.exe"
```

Dieser Abschnitt zeigt Beispiele für den eb config-Befehl, wenn er mit Parametern ausgeführt wird.

Im folgenden Beispiel wird die gespeicherte Konfiguration mit dem Namen "app-tmp" gelöscht.

```
$ eb config delete app-tmp
```

Im folgenden Beispiel wird die gespeicherte Konfiguration mit dem Namen "app-tmp" aus Ihrem Amazon S3-Bucket heruntergeladen.

```
$ eb config get app-tmp
```

Im folgenden Beispiel werden die Namen der gespeicherten Konfigurationen aufgelistet, die in Ihrem Amazon S3-Bucket gespeichert sind.

```
$ eb config list
```

Im folgenden Beispiel wird die lokale Kopie der gespeicherten Konfiguration mit dem Namen "app-tmp" in Ihren Amazon S3-Bucket hochgeladen.

```
$ eb config put app-tmp
```

Im folgenden Beispiel werden Konfigurationseinstellungen aus der aktuell laufenden Umgebung gespeichert. Wenn Sie keinen Namen für die gespeicherte Konfiguration angeben, benennt Elastic Beanstalk die Konfigurationsdatei entsprechend des Umgebungsnamens. Beispiel: Eine Umgebung mit dem Namen "tmp-dev" würde als tmp-dev.cfg.yml bezeichnet werden. Elastic Beanstalk speichert die Datei im Ordner /.elasticbeanstalk/saved_configs/.

```
$ eb config save
```

Im folgenden Beispiel wird gezeigt, wie die --cfg-Option zum Speichern der Konfigurationseinstellungen aus der Umgebung "tmp-dev" in eine Datei mit dem Namen v1-app-tmp.cfg.yml gespeichert wird. Elastic Beanstalk speichert die Datei im Ordner /.elasticbeanstalk/saved_configs/. Wenn Sie keinen Umgebungsnamen angeben, speichert Elastic Beanstalk Konfigurationseinstellungen aus der aktuell laufenden Umgebung.

```
$ eb config save tmp-dev --cfg v1-app-tmp
```

eb console

Beschreibung

Öffnet einen Browser zum Anzeigen des Umgebungskonfigurations-Dashboards in der Elastic Beanstalk Management Console.

Wenn das Stammverzeichnis eine platform.yaml-Datei enthält, die eine benutzerdefinierte Plattform angibt, zeigt dieser Befehl auch die Builder-Umgebungskonfiguration, wie in platform.yaml angegeben, in der Elastic Beanstalk Management Console an.

Syntax

eb console

eb console **environment-name**

Optionen

Name	Beschreibung
Allgemeine Optionen (p. 1097)	

eb create

Beschreibung

Erstellt eine neue Umgebung und stellt eine Anwendungsversion darin bereit.

Note

- Um den Befehl eb create für eine .NET-Anwendung zu verwenden, müssen Sie ein Bereitstellungspaket wie in [Erstellen eines Quell-Bundles für eine .NET-Anwendung \(p. 418\)](#) beschrieben erstellen und die CLI-Konfiguration zum Bereitstellen des Pakets als Artefakt gemäß der Beschreibung in [Bereitstellen eines Artefakts anstelle des Projektordners \(p. 1019\)](#) einrichten.
- Zum Erstellen von Umgebungen mit der EB-CLI ist eine [Service-Rolle \(p. 21\)](#) erforderlich. Eine Servicerolle generieren Sie, indem Sie eine Umgebung in der Elastic Beanstalk-Konsole erstellen. Falls keine Servicerolle vorhanden ist, versucht die EB-CLI eine Rolle zu erstellen, wenn Sie eb create ausführen.

Sie können die Anwendungsversion aus unterschiedlichen Quellen bereitstellen:

- Standardmäßig wird die Anwendung aus dem Quellcode im lokalen Projektverzeichnis bereitgestellt.
- Mit der Option --version kann die Anwendung aus einer Anwendungsversion bereitgestellt werden, die bereits in Ihrer Anwendung vorhanden ist.
- Wenn Ihr Projektverzeichnis keinen Anwendungscode enthält oder wenn Sie die Option --sample verwenden, können Sie die Anwendung aus einer Beispielanwendung für die Plattform Ihrer Umgebung bereitstellen.

Syntax

eb create

eb create **environment-name**

Ein Umgebungsname muss zwischen 4 und 40 Zeichen lang sein und darf nur Buchstaben, Ziffern und Bindestriche enthalten. Ein Umgebungsname darf nicht mit einem Bindestrich beginnen oder enden.

Wenn Sie einen Umgebungsnamen in den Befehl aufnehmen, fordert die EB CLI Sie nicht auf, eine Auswahl zu treffen oder eine Servicerolle zu erstellen.

Wenn Sie den Befehl ohne Umgebungsnamensargument ausführen, wird er in einem interaktiven Ablauf ausgeführt und Sie werden aufgefordert, Werte für einige Einstellungen auszuwählen. Falls Sie eine Beispielanwendung bereitstellen, werden Sie in diesem interaktiven Ablauf von der EB CLI auch gefragt, ob Sie diese Beispielanwendung in Ihr lokales Projektverzeichnis herunterladen möchten. Dadurch können Sie die EB CLI später in der neuen Umgebung verwenden, um Operationen auszuführen, für die Anwendungscode erforderlich ist, wie [eb deploy \(p. 1063\)](#).

Einige interaktive Flow-Aufforderungen werden nur unter bestimmten Bedingungen angezeigt. Wenn Sie beispielsweise einen Application Load Balancer verwenden möchten und Ihr Konto mindestens

einen freigebbaren Application Load Balancer enthält, zeigt Elastic Beanstalk eine Aufforderung an, in der Sie gefragt werden, ob Sie einen freigegebenen Load Balancer verwenden möchten. Diese Eingabeaufforderung wird nicht angezeigt, wenn Ihr Konto keinen freigebbaren Application Load Balancer enthält.

Optionen

Keine dieser Optionen ist erforderlich. Wenn Sie eb create ohne Optionen ausführen, werden Sie von der EB CLI aufgefordert, einen Wert für jede Einstellung einzugeben oder auszuwählen.

Name	Beschreibung
-d oder --branch_default	Legen Sie die Umgebung als Standard-Umgebung für das aktuelle Repository fest.
--cfg <i>config-name</i>	Verwenden Sie Plattformmeinstellungen aus einer gespeicherten Konfiguration (p. 656) in .elasticbeanstalk/saved_configs/ oder Ihrem Amazon S3-Bucket. Geben Sie den Namen der Datei ohne die .cfg.yaml-Erweiterung an.
-c <i>subdomain-name</i> oder --cname <i>subdomain-name</i>	Der Name der Subdomäne für ein Präfix des CNAME DNS-Eintrags, der auf Ihre Website weiterleitet. Typ: Zeichenkette Standard: Der Umgebungsname
-db oder --database	Fügt eine Datenbank an die Umgebung an. Wenn Sie eb create mit der Option --database, jedoch ohne die Optionen --database.username und --database.password ausführen, werden Sie von EB CLI aufgefordert, den Master-Benutzernamen und das Master-Passwort der Datenbank einzugeben.
-db.engine <i>engine</i> oder --database.engine <i>engine</i>	Der Typ der Datenbank-Engine. Wenn Sie eb create mit dieser Option ausführen, startet EB CLI die Umgebung mit einer angefügten Datenbank, auch wenn Sie den Befehl nicht mit der --database-Option ausgeführt haben. Typ: Zeichenkette Zulässige Werte: mysql, oracle-se1, postgres, sqlserver-ex, sqlserver-web, sqlserver-se
-db.i <i>instance_type</i> oder --database.instance <i>instance_type</i>	Der Typ der Amazon EC2-Instance für die Datenbank. Wenn Sie eb create mit dieser Option ausführen, startet EB CLI die Umgebung mit einer angefügten Datenbank, auch wenn Sie den Befehl nicht mit der --database-Option ausgeführt haben. Typ: Zeichenkette Gültige Werte: Siehe im Abschnitt zu Optionswerten .

Name	Beschreibung
-db.pass <i>password</i> oder --database.password <i>password</i>	Das Passwort für die Datenbank. Wenn Sie eb create mit dieser Option ausführen, startet EB CLI die Umgebung mit einer angefügten Datenbank, auch wenn Sie den Befehl nicht mit der --database-Option ausgeführt haben.
-db.size <i>number_of_gigabytes</i> oder --database.size <i>number_of_gigabytes</i>	Die Anzahl der Gigabyte (GB) für die Datenbankspeicherung. Wenn Sie eb create mit dieser Option ausführen, startet EB CLI die Umgebung mit einer angefügten Datenbank, auch wenn Sie den Befehl nicht mit der --database-Option ausgeführt haben. Typ: Zahl Zulässige Werte: <ul style="list-style-type: none"> • MySQL – 5 bis 1024. Der Standardwert ist 5. • Postgres – 5 bis 1024. Der Standardwert ist 5. • Oracle – 10 bis 1024. Der Standardwert ist 10. • Microsoft SQL Server Express Edition – 30. • Microsoft SQL Server Web Edition – 30. • Microsoft SQL Server Standard Edition – 200.
-db.user <i>username</i> (<i>Benutzername</i>) oder --database.username <i>username</i> (<i>Benutzername</i>)	Der Benutzername für die Datenbank. Wenn Sie eb create mit dieser Option ausführen, startet EB CLI die Umgebung mit einer angefügten Datenbank, auch wenn Sie den Befehl nicht mit der --database-Option ausgeführt haben. Wenn Sie eb create mit der --database-Option, jedoch ohne die --database.username- und --database.password-Optionen ausführen, werden Sie von EB CLI aufgefordert, den Benutzernamen und das Passwort der Master-Datenbank einzugeben.
-db.version <i>Version</i> oder --database.version <i>Version</i>	Ermöglicht die Angabe der Datenbank-Engine-Version. Wenn diese Markierung vorhanden ist, startet die Umgebung mit einer Datenbank mit der angegebenen Versionsnummer, auch wenn die --database-Markierung nicht vorhanden ist.
--elb-type <i>Typ</i>	Der Load Balancer-Typ (p. 554) . Typ: Zeichenkette Zulässige Werte: <code>classic</code> , <code>application</code> , <code>network</code> Standard: <code>application</code>

Name	Beschreibung
-es oder --enable-spot	Aktivieren Sie Spot-Instance-Anforderungen für Ihre Umgebung. Weitere Informationen finden Sie unter Auto Scaling-Gruppe (p. 540) . Verwandte Optionen: <ul style="list-style-type: none">• --instance-types• --on-demand-base-capacity• --on-demand-above-base-capacity• --spot-max-price
--env-group-suffix <i>groupname</i>	Gruppenname, der an den Umgebungsnamen angefügt wird. Nur zur Verwendung mit Compose-Umgebungen (p. 1038) .
--envvars	Umgebungseigenschaften (p. 621) in einer durch Kommas getrennten Liste mit dem Format <i>name (Name)=value (Wert)</i> . Informationen zu den Beschränkungen finden Sie unter Konfigurieren von Umgebungseigenschaften (p. 623) .
-ip <i>profile_name</i> oder --instance_profile <i>profile_name</i>	Die Instance-Profil mit der IAM-Rolle mit den temporären Anmeldeinformationen, die Ihre Anwendung für den Zugriff auf temporäre AWS-Ressourcen benötigt.
-it oder --instance-types <i>type1[, type2 ...]</i>	Eine durch Kommas getrennte Liste von Amazon EC2-Instance-Typen, die in Ihrer Umgebung verwendet werden sollen. Wenn Sie diese Option nicht angeben, stellt Elastic Beanstalk Standard-Instance-Typen bereit. Weitere Informationen finden Sie unter Amazon EC2-Instances (p. 531) und Auto Scaling-Gruppe (p. 540) .
-i oder --instance_type	Der Amazon EC2-Instance-Typ, den Ihre Umgebung verwenden soll. Wenn Sie diese Option nicht angeben, stellt Elastic Beanstalk einen Standard-Instance-Typ bereit. Weitere Informationen finden Sie unter Amazon EC2-Instances (p. 531) . Note Die Option --instance_type ist veraltet. Sie wird durch die neuere und leistungsfähigere Option --instance-types ersetzt. Mit der neuen Option können Sie eine Liste einer oder mehrerer Instance-Typen für Ihre Umgebung angeben. Der erste Wert auf dieser List entspricht dem Wert der Option --instance_type. Als Methode zur Angabe von Instance-Typen wird die neue Option empfohlen.

Name	Beschreibung
<p>-k <i>key_name</i> oder --keyname <i>key_name</i></p>	<p>Der Name des Amazon EC2-Schlüsselpaars zur Verwendung mit dem Secure Shell (SSH)-Client zur sicheren Anmeldung bei den Amazon EC2-Instances, die Ihre Elastic Beanstalk-Anwendung ausführen. Wenn Sie diese Option mit dem eb create Befehl bereitstellen, überschreibt der Wert alle Schlüsselnamen, die Sie mit eb init angegeben haben.</p> <p>Zulässige Werte: Ein vorhandener Schlüsselname, der mit Amazon EC2 registriert ist.</p>
<p>-im <i>number-of-instances</i> oder --min-instances <i>number-of-instances</i></p>	<p>Die minimale Anzahl von Amazon EC2-Instances, die Sie für Ihre Umgebung benötigen.</p> <p>Typ: Zahl (Ganzzahl)</p> <p>Standard: 1</p> <p>Gültige Werte: 1 bis 10000</p>
<p>-ix <i>number-of-instances</i> oder --max-instances <i>number-of-instances</i></p>	<p>Die maximale Anzahl von Amazon EC2-Instances, die Sie für Ihre Umgebung zulassen.</p> <p>Typ: Zahl (Ganzzahl)</p> <p>Standard: 4</p> <p>Gültige Werte: 1 bis 10000</p>
--modules <i>component-a</i> <i>component-b</i>	Liste der zu erstellenden Komponentenumgebungen. Nur zur Verwendung mit Compose-Umgebungen (p. 1038) .
<p>-sb oder --on-demand-base-capacity</p>	<p>Die Mindestanzahl von On-Demand-Instances, die von der Auto Scaling-Gruppe bereitgestellt werden, bevor Spot-Instances bei der Skalierung Ihrer Umgebung in Betracht gezogen werden.</p> <p>Diese Option kann nur mit der Option --enable-spot angegeben werden. Weitere Informationen finden Sie unter Auto Scaling-Gruppe (p. 540).</p> <p>Typ: Zahl (Ganzzahl)</p> <p>Standard: 0</p> <p>Gültige Werte: 0 bis --max-instances (wenn nicht vorhanden: MaxSize-Option im aws:autoscaling:asg (p. 667) Namespace)</p>

Name	Beschreibung
-sp oder --on-demand-above-base-capacity	<p>Der Anteil der On-Demand-Instances als Teil der zusätzlichen Kapazität, den Ihre Auto Scaling-Gruppe über die durch die --on-demand-base-capacity-Option angegebene Anzahl von Instances hinaus bereitstellt.</p> <p>Diese Option kann nur mit der Option --enable-spot angegeben werden. Weitere Informationen finden Sie unter Auto Scaling-Gruppe (p. 540).</p> <p>Typ: Zahl (Ganzzahl)</p> <p>Standard: 0 für eine Einzel-Instance-Umgebung; 70 für eine lastverteilte Umgebung</p> <p>Gültige Werte: 0 bis 100</p>
-p <i>Plattformversion</i> oder --platform <i>Plattformversion</i>	<p>Die zu verwendende Plattformversion (p. 32). Sie können eine Plattform, Name und Version einer Plattform, einen Plattformzweig, einen Lösungs-Stack-Namen oder einen Lösungs-Stack-ARN angeben. Zum Beispiel:</p> <ul style="list-style-type: none"> php, PHP, node.js – Die aktuelle Plattformversion für die angegebene Plattform php-7.2, "PHP 7.2" – Die empfohlene (typischerweise neueste) PHP 7.2-Plattformversion "PHP 7.2 running on 64bit Amazon Linux" – Die empfohlene (typischerweise neueste) PHP-Plattformversion in diesem Plattformzweig "64bit Amazon Linux 2017.09 v2.6.3 running PHP 7.1" – Die PHP-Plattformversion, die von diesem Lösungs-Stack-Namen angegeben wird "arn:aws:elasticbeanstalk:us-east-2::platform/PHP 7.1 running on 64bit Amazon Linux/2.6.3" – Die PHP-Plattformversion, die durch diesen Lösungs-Stack-ARN angegeben wird <p>Verwenden Sie eb platform list (p. 1077), um eine Liste der verfügbaren Konfigurationen anzuzeigen.</p> <p>Wenn Sie die --platform-Option angeben, wird der durch eb init bereitgestellte Wert überschrieben.</p>
-pr oder --process	Vorverarbeiten und Validieren des Umgebungsmanifests und der Konfigurationsdateien im Quell-Bundle. Durch die Validierung der Konfigurationsdateien können Probleme vor der Bereitstellung der Anwendungsversion für eine Umgebung identifiziert werden.
-r <i>Region</i> oder --region <i>Region</i>	<p>Die AWS-Region, in der Sie die Anwendung bereitstellen möchten.</p> <p>Eine Liste der Werte, die Sie für diese Option festlegen können, finden Sie unter AWS Elastic Beanstalk Endpunkte und Kontingente in der Allgemeinen AWS-Referenz.</p>

Name	Beschreibung
--sample	Stellen Sie die Beispielanwendung für die neue Umgebung anstelle von Code in Ihrem Repository bereit.
--scale <i>number-of-instances</i>	Starten mit der angegebenen Anzahl von Instances
--service-role <i>servicerole</i>	Weisen Sie der Umgebung eine nicht standardmäßige Servicerolle zu. Note Geben Sie keinen ARN ein, nur den Namen für die Rolle. Elastic Beanstalk fügt den Rollennamen mit den richtigen Werten als Präfix hinzu, um den resultierenden ARN intern zu erstellen.
-ls <i>load-balancer</i> oder --shared-lb <i>load-balancer</i>	Konfigurieren Sie die Umgebung für die Verwendung eines freigegebenen Load Balancer. Geben Sie den Namen oder ARN eines freigebaren Load Balancers in Ihrem Konto an. Dabei muss es sich um einen Application Load Balancer handeln, den Sie ausdrücklich erstellt haben. Es darf sich nicht um einen Load Balancer handeln, der von einer anderen Elastic Beanstalk-Umgebung erstellt wurde. Details dazu finden Sie unter Geteile Application Load Balancer (p. 583) . Parameterbeispiele: <ul style="list-style-type: none">• FrontEndLB – Ein Load Balancer-Name.• arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/FrontEndLB/0dbf78d8ad96abbc – Ein Application Load Balancer-ARN. Sie können diese Option nur mit --elb-type application angeben. Wenn Sie diese Option ohne Angabe von --shared-lb angeben, erstellt Elastic Beanstalk einen dedizierten Load Balancer für die Umgebung.
-lp <i>port</i> oder --shared-lb-port <i>port</i>	Der standardmäßige Listener-Port des gemeinsam genutzten Load Balancers für diese Umgebung. Elastic Beanstalk fügt eine Listener-Regel hinzu, die den gesamten Datenverkehr von diesem Listener an den Standardumgebungsprozess weiterleitet. Details dazu finden Sie unter Geteile Application Load Balancer (p. 583) . Typ: Zahl (Ganzzahl) Standard: 80 Gültige Werte: Jede Ganzzahl, die einen Listener-Port des freigegebenen Load Balancer darstellt.

Name	Beschreibung
--single	<p>Erstellen Sie die Umgebung mit einer einzelnen Amazon EC2-Instance und ohne Load Balancer.</p> <p>Warning</p> <p>Eine Umgebung mit einer einzelnen Instance ist nicht geeignet für die Produktion. Falls die Instance während der Bereitstellung instabil wird, oder Elastic Beanstalk beendet wird und die Instance während eines Konfigurations-Update neu gestartet wird, steht Ihre Anwendung möglicherweise längere Zeit nicht zur Verfügung. Umgebungen mit einer einzelnen Instanz können Sie für die Entwicklung, das Testen oder ein Staging nutzen. Für die Produktion verwenden Sie Umgebungen mit Load Balancing.</p>
-sm oder --spot-max-price	<p>Der Höchstpreis pro Einheitsstunde in USD, den Sie für eine Spot-Instance zu zahlen bereit sind</p> <p>Diese Option kann nur mit der Option --enable-spot angegeben werden. Weitere Informationen finden Sie unter Auto Scaling-Gruppe (p. 540).</p> <p>Typ: Zahl (Gleitkomma)</p> <p>Standard: Der On-Demand-Preis</p> <p>Gültige Werte: 0 . 001 bis 20 . 0</p>
--tags <i>key1=value1[,key2=value2]</i>	<p>Kennzeichnen Sie die Ressourcen in Ihrer Umgebung. Tags werden als durch Komma getrennte Liste von key=value-Paaren angegeben.</p> <p>Weitere Informationen finden Sie unter Markieren von Umgebungen mit Tags (p. 617).</p>
-t worker oder --tier worker	Erstellen Sie eine Worker-Umgebung. Lassen Sie diese Option zum Erstellen einer Webserverumgebung aus.
--timeout <i>minutes</i>	Legen Sie Anzahl der Minuten fest, bevor ein Timeout für den Befehl auftritt.
--version <i>version_label</i>	<p>Gibt die Anwendungsversion an, die Sie für die Umgebung bereitgestellt haben möchten, im lokalen Projektverzeichnis an, anstelle des Anwendungsquellcodes.</p> <p>Typ: Zeichenkette</p> <p>Zulässige Werte: Eine vorhandene Anwendungsversionsbezeichnung</p>
--vpc	Konfigurieren Sie eine VPC für die Umgebung. Wenn Sie diese Option einschließen, werden Sie von der EB-Befehlszeilenschnittstelle aufgefordert, alle erforderlichen Einstellungen vor dem Start der Umgebung einzugeben.

Name	Beschreibung
--vpc.dbsubnets <i>subnet1, subnet2</i>	Gibt Subnetze für Datenbank-Instances in einer VPC an. Erforderlich, wenn --vpc.id angegeben ist.
--vpc.ec2subnets <i>subnet1, subnet2</i>	Gibt Subnetze für Amazon EC2-Instances in einer VPC an. Erforderlich, wenn --vpc.id angegeben ist.
--vpc.elbpublic	Startet den Elastic Load Balancing-Load Balancer in einem öffentlichen Subnetz in Ihrer VPC. Diese Option können Sie nicht mit den Optionen --tier worker oder --single angeben.
--vpc.elbsubnets <i>subnet1, subnet2</i>	Gibt Subnetze für den Elastic Load Balancing-Load Balancer in einer VPC an. Diese Option können Sie nicht mit den Optionen --tier worker oder --single angeben.
--vpc.id <i>ID</i>	Startet Ihre Umgebung in der angegebenen VPC.
--vpc.publicip	Startet die Amazon EC2-Instances in einem öffentlichen Subnetz in Ihrer VPC. Diese Option können Sie nicht mit der Option --tier worker angeben.
--vpc.securitygroups <i>securitygroup1, securitygroup2</i>	Gibt Sicherheitsgruppen-IDs an. Erforderlich, wenn --vpc.id angegeben ist.
Allgemeine Optionen (p. 1097)	

Ausgabe

Wenn der Befehl erfolgreich war, werden Sie aufgefordert, Fragen zu beantworten und gibt den Status des erstellten Vorgangs zurück. Wenn es Probleme beim Starten gibt, können Sie über die [eb events \(p. 1064\)](#)-Operation weitere Details erhalten.

Wenn Sie die CodeBuild-Unterstützung in Ihrer Anwendung aktiviert haben, zeigt eb create Informationen von CodeBuild an, wenn Ihr Code erstellt wird. Hinweise zur CodeBuild-Unterstützung in Elastic Beanstalk finden Sie unter [EB CLI mit AWS CodeBuild verwenden \(p. 1025\)](#).

Beispiele

Im folgenden Beispiel wird eine Umgebung im interaktiven Modus erstellt.

```
$ 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:
...
```

Im folgenden Beispiel wird auch eine Umgebung im interaktiven Modus erstellt. In diesem Beispiel enthält das Projektverzeichnis keinen Anwendungscode. Der Befehl stellt eine Beispielanwendung bereit und lädt diese in Ihr lokales Projektverzeichnis herunter.

```
$ 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:
...
```

Mithilfe den folgenden Befehlen können Sie eine Umgebung erstellen, ohne eine Eingabeaufforderung anzuzeigen.

```
$ 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:
...
```

Mit dem folgenden Befehl wird eine Umgebung in einer benutzerdefinierten VPC erstellt.

```
$ 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-pqkcip3mns  
  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

Beschreibung

Stellt das Quell-Bundle der Anwendung aus dem initialisierten Projektverzeichnis für die laufende Anwendung bereit.

Wenn Git installiert ist, verwendet EB CLI den `git archive`-Befehl zum Erstellen einer `.zip`-Datei aus dem Inhalt des letzten `git commit`-Befehls.

Wenn jedoch `.ebignore` in Ihrem Projektverzeichnis vorhanden ist, verwendet die EB CLI keine `git`-Befehle und -Semantik, um Ihr Quell-Bundle zu erstellen. Das bedeutet, EB CLI ignoriert die in `.ebignore` angegebenen Dateien und bindet alle anderen Dateien ein. Insbesondere bindet sie nicht festgeschriebene Quelldateien ein.

Note

Sie können die EB CLI so konfigurieren, dass sie ein Artefakt aus Ihrem Erstellungsprozess bereitstellt, anstatt eine ZIP-Datei Ihres Projektordners zu erstellen. Details dazu finden Sie unter [Bereitstellen eines Artefakts anstelle des Projektordners \(p. 1019\)](#).

Syntax

`eb deploy`

`eb deploy environment-name`

Optionen

Name	Beschreibung
<code>-l version_label</code> oder <code>--label version_label</code>	Geben Sie eine Bezeichnung für die Version an, die die EB CLI erstellt. Wenn die Bezeichnung bereits verwendet wurde, stellt die EB CLI die frühere Version mit dieser Bezeichnung erneut bereit.
<code>--env-group-suffix groupname</code>	Typ: Zeichenkette
<code>--env-group-suffix groupname</code>	Gruppenname, der an den Umgebungsnamen angefügt wird. Nur zur Verwendung mit Compose-Umgebungen (p. 1038) .

Name	Beschreibung
-m " <i>version_description</i> " oder --message " <i>version_description</i> "	Die Beschreibung für die Anwendungsversion, in doppelte Anführungszeichen gesetzt. Typ: Zeichenkette
--modules <i>component-a</i> <i>component-b</i>	Liste der zu aktualisierenden Komponenten. Nur zur Verwendung mit Compose-Umgebungen (p. 1038) .
-p oder --process	Vorverarbeiten und Validieren des Umgebungsmanifests und der Konfigurationsdateien im Quell-Bundle. Durch die Validierung der Konfigurationsdateien können Probleme vor der Bereitstellung der Anwendungsversion für eine Umgebung identifiziert werden.
--source codecommit/ <i>repository-name</i> / <i>branch-name</i>	CodeCommit-Repository und Verzweigung. Siehe Verwenden der EB CLI mit AWS CodeCommit (p. 1028) .
--staged	Bereitstellen von Dateien, die im Git-Index und nicht im HEAD-Commit zwischengespeichert sind.
--timeout <i>minutes</i>	Die Anzahl der Minuten, bevor ein Timeout für den Befehl auftritt.
--version <i>version_label</i>	Eine vorhandene Anwendungsversion zum Bereitstellen. Typ: Zeichenkette
Allgemeine Optionen (p. 1097)	

Ausgabe

Bei Erfolg gibt der Befehl den Status des deploy-Vorgangs aus.

Wenn Sie die CodeBuild-Unterstützung in Ihrer Anwendung aktiviert haben, zeigt eb deploy Informationen von CodeBuild an, wenn Ihr Code erstellt wird. Hinweise zur CodeBuild-Unterstützung in Elastic Beanstalk finden Sie unter [EB CLI mit AWS CodeBuild verwenden \(p. 1025\)](#).

Beispiel

Im folgenden Beispiel wird die aktuelle Anwendung bereitgestellt.

```
$ 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

Beschreibung

Gibt die aktuellen Ereignisse für die Umgebung zurück.

Falls im Stammverzeichnis eine `platform.yaml`-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, gibt dieser Befehl auch die aktuellen Ereignisse für die Builder-Umgebung zurück.

Syntax

`eb events`

`eb events environment-name`

Optionen

Name	Beschreibung
<code>-f</code> oder <code>--follow</code>	Streamt Ereignisse. Wenn Sie den Vorgang abbrechen möchten, drücken Sie STRG+C.

Ausgabe

Bei Erfolg gibt der Befehl die aktuellen Ereignisse zurück.

Beispiel

Das folgende Beispiel gibt die aktuellen Ereignisse zurück.

```
$ 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-
NSKUOK5X6Z9J
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-1E3G7PZKZPSOG
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-LA9YEW3O6WJO
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

Beschreibung

Gibt den aktuellen Zustand für die Umgebung zurück.

Falls im Stammverzeichnis eine `platform.yaml`-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, gibt dieser Befehl auch den aktuellen Zustand für die Builder-Umgebung zurück.

Syntax

`eb health`

`eb health environment-name`

Optionen

Name	Beschreibung
<code>-r</code> oder <code>--refresh</code>	Zeigen Sie Zustandsinformationen interaktiv an und aktualisieren alle 10 Sekunden, wenn neue Informationen gemeldet werden.
<code>--mono</code>	Keine Anzeigefarbe in der Ausgabe.

Ausgabe

Bei Erfolg gibt der Befehl den aktuellen Zustand zurück.

Beispiel

Das folgende Beispiel gibt die aktuellsten Zustandsinformationen für eine Linux-Umgebung zurück.

```
~/project $ eb health
elasticBeanstalkExa-env
2015-07-08 23:13:20
Ok
WebServer
 2.1 (Puma)
    total      ok     warning   degraded   severe     info     pending   unknown
      5        5         0          0          0         0         0         0
Ruby
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	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	cpu	load 1	load 5	user%	nice%	system%	idle
% iowait%	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

Beschreibung

Legt Standardwerte für Elastic Beanstalk-Anwendungen fest, die mit der EB CLI erstellt wurden, indem Sie eine Reihe von Fragen beantworten müssen.

Note

Die Werte, die Sie mit eb init festlegen, gelten nur für das aktuelle Verzeichnis und Repository auf dem aktuell verwendeten Computer.

Der Befehl erstellt nichts in Ihrem Elastic Beanstalk-Konto. Um eine Elastic Beanstalk-Umgebung zu erstellen, führen Sie [eb create \(p. 1053\)](#) nach Ausführung von eb init aus.

Syntax

`eb init`

`eb init application-name`

Optionen

Wenn Sie eb init ohne Angaben der `--platform`-Option ausführen, werden Sie von der EB CLI aufgefordert, einen Wert für jede Einstellung einzugeben oder auszuwählen.

Note

Damit mit eb init ein neues Schlüsselpaar erstellt werden kann, müssen Sie `ssh-keygen` auf Ihrem lokalen Computer installiert haben und über die Befehlszeile darauf zugreifen können.

Name	Beschreibung
<code>-i</code> <code>--interactive</code>	Zwingt EB CLI, Sie aufzufordern, einen Wert für jede eb init-Befehlsoption anzugeben. Note Der <code>init</code> -Befehl fordert Sie auf, Werte für eb init-Befehlsoptionen anzugeben, die keinen (Standard-)Wert haben. Nach der ersten Ausführung des eb init-Befehls in einem Verzeichnis fordert EB CLI Sie möglicherweise nicht hinsichtlich Befehlsoptionen auf. Aus diesem Grund verwenden Sie die <code>--interactive</code> -Option, wenn Sie eine Einstellung ändern möchten, die Sie zuvor festgelegt haben.
<code>-k <i>keyname</i></code> <code>--keyname <i>keyname</i></code>	Der Name des Amazon EC2-Schlüsselpaares zur Verwendung mit dem Secure Shell (SSH)-Client zur sicheren Anmeldung bei den Amazon EC2-Instances, die Ihre Elastic Beanstalk-Anwendung ausführen.
<code>--modules <i>folder-1</i> <i>folder-2</i></code>	Liste der untergeordneten Verzeichnisse zur Initialisierung. Nur zur Verwendung mit Compose-Umgebungen (p. 1038) .
<code>-p <i>Platformversion</i></code> <code>--platform <i>Platformversion</i></code>	Die zu verwendende Plattformversion (p. 32) . Sie können eine Plattform, Name und Version einer Plattform, einen Plattformzweig, einen Lösungs-Stack-Namen oder einen Lösungs-Stack-ARN angeben. Zum Beispiel: <ul style="list-style-type: none"> • <code>php</code>, <code>PHP</code>, <code>node.js</code> – Die aktuelle Plattformversion für die angegebene Plattform • <code>php-7.2</code>, "PHP 7.2" – Die empfohlene (typischerweise neueste) PHP 7.2-Plattformversion • "PHP 7.2 running on 64bit Amazon Linux" – Die empfohlene (typischerweise neueste) PHP-Plattformversion in diesem Plattformzweig • "64bit Amazon Linux 2017.09 v2.6.3 running PHP 7.1" – Die PHP-Plattformversion, die von diesem Lösungs-Stack-Namen angegeben wird • "arn:aws:elasticbeanstalk:us-east-2::platform/PHP 7.1 running on 64bit Amazon Linux/2.6.3" – Die PHP-Plattformversion, die durch diesen Lösungs-Stack-ARN angegeben wird Verwenden Sie eb platform list (p. 1077) , um eine Liste der verfügbaren Konfigurationen anzuzeigen. Geben Sie die Option <code>--platform</code> an, um die interaktive Konfiguration zu überspringen.

Name	Beschreibung	
	<p>Note</p> <p>Wenn Sie diese Option angeben, werden Sie von der EB CLI nicht zur Eingabe von Werten für andere Optionen aufgefordert. Stattdessen werden die Standardwerte für die einzelnen Optionen verwendet. Sie können Optionen für alles angeben, für das Sie keine Standardwerte verwenden möchten.</p>	
--source codecommit/ <i>repository-name</i> / <i>branch-name</i>	CodeCommit-Repository und Verzweigung. Siehe Verwenden der EB CLI mit AWS CodeCommit (p. 1028) .	
--tags <i>key1=value1[,key2=value2,...]</i>	<p>Markieren Sie Ihre Anwendung mit Tags. Tags werden als durch Komma getrennte Liste von key=value-Paaren angegeben.</p> <p>Weitere Informationen finden Sie unter Markieren von Anwendungen (p. 421).</p>	
Allgemeine Optionen (p. 1097)		

CodeBuild-Unterstützung

Wenn Sie eb init in einem Ordner mit einer `buildspec.yml`-Datei ausführen, analysiert Elastic Beanstalk die Datei auf einen `eb_codebuild_settings`-Eintrag mit für Elastic Beanstalk spezifischen Optionen. Weitere Informationen zur CodeBuild-Unterstützung in Elastic Beanstalk finden Sie unter [EB CLI mit AWS CodeBuild verwenden \(p. 1025\)](#).

Ausgangs-

Wenn der Befehl erfolgreich war, werden Sie durch die Einrichtung einer neuen Elastic Beanstalk-Anwendung mithilfe einer Reihe von Anweisungen geführt.

Beispiel

Die folgende Beispielanforderung initialisiert EB CLI und fordert Sie auf, Informationen zu Ihrer Anwendung einzugeben. Ersetzen Sie den `Platzhaltertext` durch eigene Werte.

```
$ 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

Beschreibung

Unterbefehle von eb labs unterstützen laufende oder experimentelle Funktionen. Diese Befehle werden möglicherweise in künftigen Versionen der EB CLI entfernt oder überarbeitet und bleiben nicht garantiert kompatibel.

Für eine Liste der verfügbaren Unterbefehle und Beschreibungen führen Sie eb labs --help aus.

eb list

Beschreibung

Listet alle Umgebungen in der aktuellen Anwendung oder alle Umgebungen in allen Anwendungen auf, wie in der --all-Option angegeben.

Falls im Stammverzeichnis eine `platform.yaml`-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, listet dieser Befehl auch die Builder-Umgebung auf.

Syntax

eb list

Optionen

Name	Beschreibung
<code>-a</code>	Listet alle Umgebungen von allen Anwendungen auf.

Name	Beschreibung
oder	
--all	
-v	Bietet detaillierte Informationen über alle Umgebungen, einschließlich Instances.
oder	
--verbose	
Allgemeine Optionen (p. 1097)	

Ausgabe

Wenn der Befehl erfolgreich war, gibt er eine Liste der Umgebungsnamen zurück, in denen Ihre aktuelle Umgebung mit einem Sternchen (*) gekennzeichnet ist.

Beispiel 1

Im folgenden Beispiel werden Ihre Umgebungen aufgelistet und angegeben, dass "tmp-dev" Ihre Standard-Umgebung ist.

```
$ eb list
* tmp-dev
```

Beispiel 2

Im folgenden Beispiel werden Ihre Umgebungen mit weiteren Informationen aufgelistet.

```
$ eb list --verbose
Region: us-west-2
Application: tmp
  Environments: 1
    * tmp-dev : ['i-c7ee492d']
```

eb local

Beschreibung

Verwenden Sie eb local run, um den Anwendungscontainer lokal in Docker auszuführen. Prüfen Sie den Status des Anwendungscontainers mit eb local status. Öffnen Sie die Anwendung mit eb local open in einem Webbrowser. Rufen Sie den Speicherort der Anwendungsprotokolle mit eb local logs ab.

Mit eb local setenv und eb local printenv können Sie Umgebungsvariablen einrichten und anzeigen, welche für die lokal mit eb local run ausgeführten Docker-Container bereitgestellt werden.

Sie müssen alle eb local-Befehle im Projektverzeichnis einer Docker-Anwendung ausführen, die mit eb init als EB CLI-Repository initialisiert wurde.

Note

Verwenden Sie eb local auf einem lokalen Computer mit Linux oder macOS. Der Befehl unterstützt Windows nicht.

Bevor Sie den Befehl unter macOS verwenden, installieren Sie Docker für Mac und stellen Sie sicher, dass boot2docker nicht installiert ist (oder sich nicht im Ausführungspfad) befindet. Der eb local-Befehl versucht boot2docker zu verwenden, sofern vorhanden, funktioniert damit jedoch unter macOS nicht gut.

Syntax

eb local run
eb local status
eb local open
eb local logs
eb local setenv
eb local printenv

Optionen

eb local run

Name	Beschreibung
--envvars key1=value1, key2=value2	Legt Umgebungsvariablen fest, die von der EB CLI an die lokalen Docker-Container übergeben werden. In Multicontainer-Umgebungen werden alle Variablen an alle Container übertragen.
--port hostport	Ordnet einen Host-Port dem entsprechenden Container-Port zu. Wenn Sie diese Option nicht angeben, verwendet die EB CLI denselben Port für Host und Container. Diese Option funktioniert nur mit Docker-Plattformanwendungen. Sie gilt nicht für die Multicontainer Docker-Plattform.
Allgemeine Optionen (p. 1097)	

eb local status
eb local open
eb local logs
eb local setenv
eb local printenv

Name	Beschreibung
Allgemeine Optionen (p. 1097)	

Ausgabe

eb local run

Statusmeldungen von Docker. Bleibt aktiv, solange die Anwendung ausgeführt wird. Drücken Sie Strg+C, um die Anwendung zu stoppen.

eb local status

Der Status der einzelnen Container, die von der Anwendung verwendet werden (ausgeführt oder nicht ausgeführt).

eb local open

Öffnet die Anwendung in einem Webbrowser und wird beendet.

eb local logs

Der Speicherort der Protokolle, die im Projektverzeichnis von lokal unter eb local run ausgeführten Anwendungen generiert werden.

eb local setenv

Keine

eb local printenv

Die Namen und Werte der mit eb local setenv festgelegten Umgebungsvariablen.

Beispiele

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

Zeigen Sie den Status der lokalen Container an:

```
~/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

Zeigen Sie den Protokollpfad für das aktuelle Projekt an:

```
~/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

Legen Sie Umgebungsvariablen zur Verwendung mit eb local run fest.

```
~/project$ eb local setenv PARAM1=value
```

Drucken Sie die mit eb local setenv festgelegten Umgebungsvariablen.

```
~/project$ eb local printenv  
Environment Variables:  
PARAM1=value
```

eb logs

Beschreibung

Der eb logs-Befehl hat zwei verschiedene Zwecke: zum Aktivieren oder Deaktivieren von Protokoll-Streaming nach CloudWatch Logs und zum Abrufen von Instance-Protokollen oder CloudWatch Logs-Protokollen. Mit der Option --cloudwatch-logs (-cw) aktiviert oder deaktiviert der Befehl das Protokoll-Streaming. Ohne diese Option werden Protokolle abgerufen.

Beim Abrufen der Protokolle geben Sie die Option --all, --zip oder --stream an, um vollständige Protokolle abzurufen. Wenn Sie keine dieser Optionen angeben, ruft Elastic Beanstalk Protokollfragmente ab.

Der Befehl verarbeitet Protokolle für die angegebene oder die Standardumgebung. Die entsprechenden Protokolle sind je nach Containertyp unterschiedlich. Falls im Stammverzeichnis eine platform.yaml-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, verarbeitet dieser Befehl auch die Protokolle für die Builder-Umgebung.

Weitere Informationen finden Sie unter [the section called “CloudWatch-Protokolle” \(p. 880\)](#).

Syntax

So aktivieren oder deaktivieren Sie das Protokoll-Streaming in CloudWatch Logs:

```
eb logs --cloudwatch-logs [enable | disable] [--cloudwatch-log-source instance | environment-health | all] [environment-name]
```

So rufen Sie Instance-Protokolle ab:

```
eb logs [-all | --zip | --stream] [--cloudwatch-log-source instance] [--instance instance-id] [--log-group log-group] [environment-name]
```

So rufen Sie Umgebungszustand-Protokolle ab:

```
eb logs [-all | --zip | --stream] --cloudwatch-log-source environment-health [environment-name]
```

Optionen

Name	Beschreibung
-cw [enable disable] oder --cloudwatch-logs [enable disable]	Aktiviert oder deaktiviert das Streamen von Protokollen in CloudWatch Logs. Falls kein Argument angegeben wird, ist das Streaming der Protokolle aktiviert. Wenn die Option --cloudwatch-log-source (-cls) nicht zusätzlich angegeben ist, ist das Instance-Protokoll-Streaming aktiviert oder deaktiviert.
-cls instance environment-health all oder --cloudwatch-log-source instance environment-health all	Gibt die Quelle für Protokolle bei der Arbeit mit CloudWatch Logs an. Mit der Aktivierungs- oder Deaktivierungsform des Befehls sind dies die Protokolle, für die das CloudWatch Logs-Streaming aktiviert oder deaktiviert werden soll. Mit der Abrufform des Befehls sind dies die Protokolle, die aus CloudWatch Logs abgerufen werden sollen. Zulässige Werte: <ul style="list-style-type: none">• Mit --cloudwatch-logs (Aktivieren oder Deaktivieren) – instance environment-health all• Ohne --cloudwatch-logs (Abrufen) – instance environment-health Bedeutungen der Werte: <ul style="list-style-type: none">• instance (Standard) – Instance-Protokolle• environment-health – Umgebungszustand-Protokolle (wird nur unterstützt, wenn erweiterte Zustandsberichte in der Umgebung aktiviert sind)• all – Beide Protokollquellen
-a oder --all	Ruft vollständige Protokolle ab und speichert sie im Verzeichnis .elasticbeanstalk/logs.
-z oder --zip	Ruft vollständige Protokolle ab, komprimiert sie in einer .zip-Datei und speichert diese im Verzeichnis .elasticbeanstalk/logs.
--stream	Streamt vollständige Protokolle (fortlaufende Ausgaben). Mit dieser Option wird der Befehl ausgeführt, bis Sie ihn abbrechen (drücken Sie hierzu Ctrl+C).
-i <i>instance-id</i> oder --instance <i>instance-id</i>	Ruft nur Protokolle für die angegebene Instance ab.
-g <i>log-group</i> oder	Gibt die CloudWatch Logs-Protokollgruppe an, aus der Protokolle abgerufen werden sollen. Die Option ist nur gültig, wenn das Instance-Protokoll-Streaming in CloudWatch Logs aktiviert ist.

Name	Beschreibung
--log-group <i>log-group</i>	<p>Wenn das Instance-Protokoll-Streaming aktiviert ist und Sie nicht die --log-group-Option angeben, ist die Standard-Protokollgruppe eine der folgenden:</p> <ul style="list-style-type: none"> Linux-Plattformen – /aws/elasticbeanstalk/<i>environment-name</i>/var/log/eb-activity.log Windows-Plattform – /aws/elasticbeanstalk/<i>environment-name</i>/EBDeploy-Log <p>Weitere Informationen über die Protokollgruppe für jede Protokolldatei finden Sie unter Wie Elastic Beanstalk CloudWatch Logs einrichtet (p. 883).</p>
Allgemeine Optionen (p. 1097)	

Ausgabe

Standardmäßig werden die Protokolle direkt im Terminal angezeigt. Verwendet ein Paginierungsprogramm, um die Ausgabe anzuzeigen. Drücken Sie zum Beenden **q** oder **q**.

Zeigt mit --stream vorhandene Protokolle im Terminal an und führt sie weiterhin aus. Drücken Sie **ctrl +c** zum Beenden.

Speichert mit --all und --zip die Protokolle in lokalen Dateien und zeigt den Speicherort der Datei.

Beispiele

Das folgende Beispiel aktiviert das Streaming von Instance-Protokollen in 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.
```

Das folgende Beispiel ruft die Instance-Protokolle in eine .zip Datei ab.

```
$ eb logs --zip
Retrieving logs...
Logs were saved to /home/workspace/environment/.elasticbeanstalk/logs/150622_173444.zip
```

eb open

Beschreibung

Öffnet die öffentliche URL der Website im Standardbrowser.

Syntax

eb open

eb open **environment-name**

Optionen

Name	Beschreibung
Allgemeine Optionen (p. 1097)	

Ausgabe

Der Befehl eb open erzeugt keine Ausgabe. Stattdessen wird die Anwendung in einem Browserfenster geöffnet.

eb platform

Beschreibung

Dieser Befehl unterstützt zwei unterschiedliche Workspaces:

[Plattform \(p. 1077\)](#)

Verwenden Sie diesen Workspace zum Verwalten von benutzerdefinierten Plattformen.

[Umgebung \(p. 1083\)](#)

Verwenden Sie diesen Workspace, um eine Standardplattform auszuwählen oder Informationen zur aktuellen Plattform anzuzeigen.

Elastic Beanstalk bietet den Schnellzugang ebp für eb platform.

Note

Windows PowerShell verwendet ebp als Befehls-Alias. Wenn Sie die EB CLI in Windows PowerShell ausführen, müssen Sie die Langform dieses Befehls verwenden – eb platform.

Verwenden der eb-Plattform für benutzerdefinierte Plattformen

Listet die Versionen der aktuellen Plattform auf und ermöglicht Ihnen die Verwaltung von benutzerdefinierten Plattformen.

Syntax

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]
```

Optionen

Name	Beschreibung
<code>create [<i>version</i>] [<i>options</i>]</code>	Erstellen Sie eine neue Version der Plattform. Weitere Informationen (p. 1079) .
<code>delete <i>version</i> [<i>options</i>]</code>	Löschen Sie eine Plattformversion. Weitere Informationen (p. 1080) .
<code>events [<i>version</i>] [<i>options</i>]</code>	Zeigen Sie die Ereignisse einer Plattformversion an. Weitere Informationen (p. 1080) .
<code>init [<i>platform</i>] [<i>options</i>]</code>	Initialisieren Sie ein Plattform-Repository. Weitere Informationen (p. 1081) .
<code>list [<i>options</i>]</code>	Listen Sie die Versionen der aktuellen Plattform auf. Weitere Informationen (p. 1081) .
<code>logs [<i>version</i>] [<i>options</i>]</code>	Zeigen Sie Protokolle aus der Builder-Umgebung für eine Plattformversion an. Weitere Informationen (p. 1082) .
<code>status [<i>version</i>] [<i>options</i>]</code>	Zeigen Sie den Status einer Plattformversion an. Weitere Informationen (p. 1082) .
<code>use [<i>platform</i>] [<i>options</i>]</code>	Wählen Sie eine andere Plattform, aus der neue Versionen erstellt werden. Weitere Informationen (p. 1082) .
Allgemeine Optionen (p. 1097)	

Allgemeine Optionen

Alle eb platform-Befehle umfassen die folgenden allgemeinen Optionen.

Name	Beschreibung
<code>-h</code>	Zeigt eine Hilfenachricht an und wird beendet.
ODER	
<code>--help</code>	
<code>--debug</code>	Zeigt zusätzliche Debugging-Ausgabe.
<code>--quiet</code>	Unterdrückt alle Ausgabedateien.
<code>-v</code>	Zeigt zusätzliche Ausgabedateien an.
ODER	

Name	Beschreibung
--verbose	
--profile <i>PROFILE</i>	Verwendet das angegebene <i>PROFILE</i> aus Ihren Anmeldeinformationen.
-r <i>REGION</i> ODER --region <i>REGION</i>	Verwenden Sie die Region <i>REGION</i> .
--no-verify-ssl	Überprüfen Sie keine AWS-SSL-Zertifikate.

Eb platform create

Erstellt eine neue Version der Plattform und gibt den ARN für die neue Version zurück. Wenn keine Builder-Umgebung in der aktuellen Region ausgeführt wird, startet dieser Befehl eine. Die *Versions*- und Erhöhungsoptionen (-M, -m und -p) schließen sich gegenseitig aus.

Optionen

Name	Beschreibung
<i>Version</i>	Wenn die <i>Version</i> nicht angegeben wird, wird eine neue Version basierend auf der aktuellen Plattform mit der erhöhten Patch-Version (N in n.n.N) erstellt.
-M ODER --major-increment	Erhöht die höhere Versionsnummer (die N in N.n.n).
-m ODER --minor-increment	Erhöht die kleinere Versionsnummer (die N in n.N.n).
-p ODER --patch-increment	Erhöht die Patch-Versionsnummer (die N in n.n.N).
-i <i>INSTANCE_TYPE</i> ODER --instance-type <i>INSTANCE_TYPE</i>	Verwenden Sie <i>INSTANCE_TYPE</i> als Instance-Typ, z. B. t1.micro .
-ip <i>INSTANCE_PROFILE</i> ODER --instance-profile <i>INSTANCE_PROFILE</i>	Verwenden Sie <i>INSTANCE_PROFILE</i> als Instance-Profil beim Erstellen von AMIs für eine benutzerdefinierte Plattform. Wenn die -ip-Option nicht angegeben ist, wird das Instance-Profil <code>aws-elasticbeanstalk-custom-platform-ec2-role</code> erstellt und für die benutzerdefinierte Plattform verwendet.

Name	Beschreibung
--tags <code>key1=value1[,key2=value2]</code>	Markieren Sie die benutzerdefinierte Plattformversion mit Tags. Tags werden als durch Komma getrennte Liste von key=value-Paaren angegeben. Weitere Informationen finden Sie unter Markieren von benutzerdefinierten Plattformversionen (p. 1127) .
--timeout <code>minutes</code>	Legen Sie Anzahl der Minuten fest, bevor ein Timeout für den Befehl auftritt.
--vpc.id <code>VPC_ID</code>	Die ID der VPC, in der Packer erstellt.
--vpc.subnets <code>VPC_SUBNETS</code>	Die VPC-Subnetze, in der Packer erstellt.
--vpc.publicip	Weist öffentliche IP-Adressen zu gestarteten EC2-Instances zu.

Eb platform delete

Löschen Sie eine Plattformversion. Die Version wird nicht gelöscht, wenn eine Umgebung diese Version verwendet.

Optionen

Name	Beschreibung
<code>version</code>	Die zu löschende Version. Dieser Wert ist erforderlich.
--cleanup	Entfernen Sie alle Plattformversionen mit dem Failed-Status.
--all-platforms	Wenn --cleanup angegeben ist, entfernen Sie alle Plattformversionen mit dem Failed-Status für alle Plattformen.
--force	Keine Bestätigung beim Löschen einer Version erforderlich.

Eb platform events

Zeigen Sie die Ereignisse einer Plattformversion an. Wenn `Version` angegeben ist, zeigen Sie die Ereignisse aus dieser Version an. Andernfalls werden die Ereignisse aus der aktuellen Version angezeigt.

Optionen

Name	Beschreibung
<code>Version</code>	Die Version, für die die Ereignisse angezeigt werden. Dieser Wert ist erforderlich.
-f ODER --follow	Ereignisse weiterhin anzeigen, wenn sie auftreten.

Eb platform init

Initialisieren Sie ein Plattform-Repository.

Optionen

Name	Beschreibung
platform	Der Name der zu initialisierenden Plattform. Dieser Wert ist erforderlich, es sei denn, -i (interaktiver Modus) ist aktiviert.
-i ODER --interactive	Verwenden Sie den interaktiven Modus.
-k KEYNAME ODER --keyname KEYNAME	Der Standard-EC2-Schlüsselname.

Sie können diesen Befehl in einem Verzeichnis ausführen, das zuvor initialisiert wurde. Sie können den Workspace-Typ jedoch nicht mehr ändern, wenn er in einem Verzeichnis ausgeführt wird, das zuvor initialisiert wurde.

Um mit anderen Optionen neu zu initialisieren, verwenden Sie die **-i**-Option.

Eb platform list

Listen Sie die Versionen der Plattform auf, die mit dem Workspace (Verzeichnis) oder einer Region verknüpft sind.

Der Befehl gibt unterschiedliche Ergebnisse zurück, abhängig von der Workspace-Art, die Sie ausführen:

- In einem Plattform-Workspace (ein Verzeichnis, das durch `eb platform init` initialisiert wurde) gibt der Befehl eine Liste aller Plattformversionen der in diesem Workspace definierten benutzerspezifischen Plattformen zurück. Fügen Sie die Option `--all-platforms` oder `--verbose` hinzu, um eine Liste aller Plattformversionen aller benutzerdefinierten Plattformen anzuzeigen, mit denen Ihr Konto in der Region mit dem Workspace verknüpft ist.
- In einem Anwendungs-Workspace (ein Verzeichnis, dass durch `eb init` initialisiert wurde) gibt der Befehl eine Liste aller Plattformversionen aus. Diese enthält die durch Elastic Beanstalk verwalteten Plattformen und die benutzerdefinierten Plattformen Ihres Kontos. Die Liste enthält die Kurznamen der Plattformversionen und einige Plattformversionsvarianten können zusammengefasst aufgelistet sein. Fügen Sie die Option `--verbose` hinzu, um eine detaillierte Liste mit den vollständigen Namen anzuzeigen, in der alle Versionsvarianten separat aufgelistet sind.
- In einem nicht initialisierten Verzeichnis funktioniert nur der Befehl mit der `--region`-Option. Sie gibt eine Liste aller von Elastic Beanstalk verwalteten Plattformversionen zurück, die in der Region unterstützt werden. Die Liste enthält die Kurznamen der Plattformversionen und einige Plattformversionsvarianten können zusammengefasst aufgelistet sein. Fügen Sie die Option `--verbose` hinzu, um eine detaillierte Liste mit den vollständigen Namen anzuzeigen, in der alle Versionsvarianten separat aufgelistet sind.

Optionen

Name	Beschreibung
-a ODER --all-platforms	Gilt nur in einem initialisierten Workspace (einem Verzeichnis, dass durch <code>eb platform init</code> oder <code>eb init</code> initialisiert wurde). Listet die Versionen aller benutzerdefinierten Plattformen auf, die Ihrem Konto zugeordnet sind.
-s <i>STATUS</i> ODER --status <i>STATUS</i>	Listen Sie nur Plattformen mit folgendem <i>STATUS</i> auf: <ul style="list-style-type: none"> • Bereit • Fehlgeschlagen • Deleting (Wird gelöscht) • Creating

Eb platform logs

Zeigen Sie Protokolle aus der Builder-Umgebung für eine Plattformversion an.

Optionen

Name	Beschreibung
<i>version</i>	Die Version der Plattform, für die Protokolle angezeigt werden. Wenn nicht angegeben, werden Protokolle aus der aktuellen Version angezeigt.
--stream	Streamen Sie Bereitstellungsprotokolle, die mit CloudWatch eingerichtet wurden.

Eb platform status

Zeigen Sie den Status einer Plattformversion an.

Optionen

Name	Beschreibung
<i>version</i>	Die Version der Plattform, für die der Status abgerufen wird. Wenn nicht angegeben, wird der Status der aktuellen Version angezeigt.

Eb platform use

Wählen Sie eine andere Plattform, aus der neue Versionen erstellt werden.

Optionen

Name	Beschreibung
<i>platform</i>	Gibt <i>platform</i> (<i>Plattform</i>) als die aktive Version für diesen Workspace an. Dieser Wert ist erforderlich.

Verwenden der eb-Plattform für Umgebungen

Listet die unterstützten Plattformen auf und ermöglicht Ihnen, die Standardplattform und die zu verwendende Plattformversion festzulegen, wenn Sie eine Umgebung starten. Verwenden Sie eb platform list, um eine Liste aller unterstützten Plattformen anzuzeigen. Verwenden Sie eb platform select zum Ändern der Plattform für Ihr Projekt. Verwenden Sie eb platform show zum Anzeigen der ausgewählten Plattform Ihres Projekts.

Syntax

```
eb platform list
eb platform select
eb platform show
```

Optionen

Name	Beschreibung
list	Listen Sie die Version der aktuellen Plattform auf.
select	Wählen Sie die Standardplattform.
show	Zeigen Sie Informationen zur aktuellen Plattform.

Beispiel 1

Im folgenden Beispiel werden die Namen aller Konfigurationen für alle Plattformen aufgelistet, die Elastic Beanstalk unterstützt.

```
$ 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
```

Beispiel 2

Im folgenden Beispiel werden Sie aufgefordert, aus einer Liste von Plattformen auszuwählen und die Version zu nehmen, die Sie für die angegebene Plattform bereitstellen möchten.

```
$ 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)
```

Beispiel 3

Im folgenden Beispiel werden Informationen über die aktuelle Standardplattform gezeigt.

```
$ 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

Beschreibung

Drückt alle Umgebungseigenschaften im Befehlsfenster.

Syntax

eb printenv

eb printenv **environment-name**

Optionen

Name	Beschreibung
Allgemeine Optionen (p. 1097)	

Ausgabe

Bei Erfolg gibt der Befehl den Status des printenv-Vorgangs aus.

Beispiel

Das folgende Beispiel druckt Umgebungseigenschaften für die angegebene Umgebung.

```
$ eb printenv
Environment Variables:
PARAM1 = Value1
```

eb restore

Beschreibung

Erstellt eine beendete Umgebung erneut, erstellt eine neue Umgebung mit demselben Namen, derselben ID und Konfiguration. Die Umgebungsname, Domänennamen und die Anwendungsversion müssen verwendet werden können, damit die Neuerstellung erfolgreich abgeschlossen werden kann.

Syntax

eb restore

eb restore **environment_id**

Optionen

Name	Beschreibung
Allgemeine Optionen (p. 1097)	

Ausgabe

Die EB CLI zeigt eine Liste der beendeten Umgebungen, die für Wiederherstellungen verfügbar sind.

Beispiel

```
$ eb restore
Select a terminated environment to restore

#    Name          ID           Application Version      Date Terminated      Ago
3    gamma        e-s7miemej8e9  app-77e3-161213_211138  2016/12/14 20:32 PST  13 mins
2    beta         e-sj28uu2wia  app-77e3-161213_211125  2016/12/14 20:32 PST  13 mins
1    alpha        e-gia8mphu6q  app-77e3-161213_211109  2016/12/14 16:21 PST  4 hours

(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

Beschreibung

Skaliert die Umgebung so, dass sie immer auf einer bestimmten Anzahl von Instances ausgeführt wird, wobei die minimale und maximale Anzahl an Instances auf die angegebene Zahl festgelegt wird.

Syntax

eb scale **number-of-instances**

eb scale **number-of-instances environment-name**

Optionen

Name	Beschreibung
--timeout	Die Anzahl der Minuten, bevor ein Timeout für den Befehl auftritt.
Allgemeine Optionen (p. 1097)	

Ausgabe

Wenn der Befehl erfolgreich war, wird die Anzahl der minimal und maximal auszuführenden Instances auf die angegebene Zahl aktualisiert.

Beispiel

Im folgenden Beispiel wird die Anzahl der Instances auf 2 festgelegt.

```
$ 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

Beschreibung

Damit werden [Umgebungseigenschaften \(p. 621\)](#) für die Standardumgebung festgelegt.

Syntax

eb setenv **key=value**

Sie können beliebig viele Eigenschaften einbeziehen, die Gesamtgröße aller Eigenschaften darf jedoch nicht mehr als 4 096 Bytes betragen. Sie können eine Variable löschen, indem Sie den Wert leer lassen. Informationen zu den Beschränkungen finden Sie unter [Konfigurieren von Umgebungseigenschaften \(p. 623\)](#).

Note

Falls value ein [Sonderzeichen](#) enthält, müssen Sie ein Escape-Zeichen in Form von \ voranstellen.

Optionen

Name	Beschreibung
--timeout	Die Anzahl der Minuten, bevor ein Timeout für den Befehl auftritt.
Allgemeine Optionen (p. 1097)	

Ausgabe

Bei Erfolg wird vom Befehl angezeigt, dass die Umgebungsaktualisierung erfolgreich ausgeführt werden konnte.

Beispiel

Im folgenden Beispiel wird die Umgebungsvariable "ExampleVar" festgelegt.

```
$ 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.
```

Mit dem folgenden Befehl werden mehrere Umgebungseigenschaften definiert. Er fügt die Umgebungseigenschaft foo hinzu, legt deren Wert auf bar fest, ändert den Wert der Eigenschaft JDBC_CONNECTION_STRING und löscht die Eigenschaften PARAM4 und PARAM5.

```
$ eb setenv foo=bar JDBC_CONNECTION_STRING=hello PARAM4= PARAM5=
```

eb ssh

Beschreibung

Note

Dieser Befehl kann nicht auf Umgebungen mit Windows Server-Instances ausgeführt werden.

Stellen Sie eine Verbindung mit einer Linux-Amazon EC2-Instance in Ihrer Umgebung mithilfe von Secure Shell (SSH) her. Wenn eine Umgebung mehrere Instances ausführt, werden Sie von der EB CLI aufgefordert anzugeben, mit welcher Instance Sie eine Verbindung herstellen möchten. Um diesen Befehl verwenden zu können, muss SSH auf Ihrem lokalen Computer installiert und über die Befehlszeile verfügbar sein. Dateien mit privaten Schlüsseln müssen sich in einem Ordner mit dem Namen .ssh in Ihrem Benutzerverzeichnis befinden und die EC2-Instances in Ihrer Umgebung müssen öffentliche IP-Adressen haben.

Falls im Stammverzeichnis eine `platform.yaml`-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, stellt dieser Befehl auch eine Verbindung zu Instances in der benutzerdefinierten Umgebung her.

SSH-Schlüssel

Wenn Sie SSH nicht zuvor konfiguriert haben, können Sie mit der EB CLI einen Schlüssel erstellen, wenn `eb init` ausgeführt wird. Wenn Sie den Befehl `eb init` bereits ausgeführt haben, führen Sie ihn erneut mit der `--interactive`-Option aus und wählen Sie Yes (Ja) und Create New Keypair (Neues Schlüsselpaar anlegen) aus, wenn Sie zum Einrichten von SSH aufgefordert werden. Schlüssel, die während dieses Prozesses erstellt wurden, werden von der EB CLI im richtigen Ordner gespeichert.

Mit diesem Befehl wird Port 22 vorübergehend in der Sicherheitsgruppe Ihrer Umgebung für eingehenden Datenverkehr von 0.0.0.0/0 (alle IP-Adressen) geöffnet, wenn keine Regeln für Port 22 vorhanden sind. Wenn Sie die Sicherheitsgruppe Ihrer Umgebung für mehr Sicherheit so konfiguriert haben, dass Port 22 für einen eingeschränkten CIDR-Bereich geöffnet wird, berücksichtigt die EB CLI diese Einstellung und verzichtet auf Änderungen an der Sicherheitsgruppe. Wenn Sie dieses Verhalten überschreiben und die EB CLI zwingen möchten, Port 22 für allen eingehenden Datenverkehr zu öffnen, verwenden Sie die `--force`-Option.

Weitere Informationen zum Konfigurieren der Sicherheitsgruppe Ihrer Umgebung finden Sie unter [Sicherheitsgruppen \(p. 535\)](#).

Syntax

`eb ssh`

`eb ssh environment-name`

Optionen

Name	Beschreibung
<code>-i</code> oder <code>--instance</code>	Gibt die Instance-ID der Instance an, zu der Sie eine Verbindung herstellen. Wir empfehlen die Verwendung dieser Option.
<code>-n</code> oder <code>--number</code>	Geben Sie durch eine Zahl die Instance an, zu der eine Verbindung hergestellt werden soll.
<code>-o</code> oder <code>--keep_open</code>	Lassen Sie Port 22 auf der Sicherheitsgruppe offen, nachdem die SSH-Sitzung beendet wurde.
<code>--command</code>	Führen Sie einen Shell-Befehl auf der angegebenen Instance aus, anstatt eine SSH-Sitzung zu starten.
<code>--custom</code>	Geben Sie einen SSH-Befehl an, der anstatt "ssh -i keyfile" verwendet werden soll. Verwenden Sie den Remote-Benutzer- und Hostnamen nicht.
<code>--setup</code>	Ändern Sie das Schlüsselpaar, das den Instances der Umgebung zugewiesen ist (erfordert das Ersetzen von Instances).

Name	Beschreibung
--force	<p>Öffnen Sie Port 22 für eingehenden Datenverkehr von 0.0.0.0/0 in der Sicherheitsgruppe der Umgebung, selbst wenn die Sicherheitsgruppe bereits für SSH konfiguriert ist.</p> <p>Verwenden Sie diese Option, wenn die Sicherheitsgruppe Ihrer Umgebung so konfiguriert ist, dass Port 22 für einen eingeschränkten CIDR-Bereich geöffnet wird, der die IP-Adresse, von der Sie die Verbindung herstellen möchten, nicht enthält.</p>
--timeout <i>minutes</i>	<p>Legen Sie Anzahl der Minuten fest, bevor ein Timeout für den Befehl auftritt.</p> <p>Kann nur mit dem Argument --setup verwendet werden.</p>
Allgemeine Optionen (p. 1097)	

Ausgabe

Wenn der Befehl erfolgreich war, wird eine SSH-Verbindung mit der Instance geöffnet.

Beispiel

Im folgenden Beispiel werden Sie mit der angegebenen Umgebung verbunden.

```
$ 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

Beschreibung

Bietet Informationen über den Status der Umgebung.

Falls im Stammverzeichnis eine `platform.yaml`-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, bietet dieser Befehl auch Informationen über die Builder-Umgebung.

Syntax

`eb status`

`eb status environment-name`

Optionen

Name	Beschreibung
<code>-v</code> oder <code>--verbose</code>	Hier finden Sie weitere Informationen zu den einzelnen Instances, wie zum Beispiel ihren Status mit dem Elastic Load Balancing-Load Balancer.
Allgemeine Optionen (p. 1097)	

Ausgabe

Bei Erfolg gibt der Befehl folgende Informationen zur Umgebung zurück:

- Environment name
- Anwendungsname
- Bereitgestellte Anwendungsversion
- Umgebungs-ID
- Plattform
- Umgebungsschicht
- CNAME
- Zeitpunkt der letzten Aktualisierung der Umgebung
- Status
- Integrität

Wenn Sie den Verbose-Modus verwenden, gibt die EB CLI Ihnen außerdem die Anzahl der laufenden Amazon EC2-Instances an.

Beispiel

Im folgenden Beispiel wird der Status der Umgebung "tmp-dev" angezeigt.

```
$ 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

Beschreibung

Damit wird der CNAME einer Umgebung mit dem CNAME einer anderen Umgebung ausgetauscht (z. B. um Ausfallzeiten beim Aktualisieren der Anwendungsversion zu verhindern).

Note

Bei mehr als zwei Umgebungen werden Sie aufgefordert, aus einer Liste den Namen der Umgebung auszuwählen, die derzeit den gewünschten CNAME hat. Um dies zu unterbinden, geben Sie den Namen der entsprechenden Umgebung an. Dazu binden Sie die Option `-n` ein, wenn Sie den Befehl ausführen.

Syntax

`eb swap`

`eb swap environment-name`

Note

Mit `environment-name` wird die Umgebung angegeben, für die ein anderer CNAME verwendet werden soll. Falls Sie `environment-name` bei der Ausführung von `eb swap` nicht als Befehlszeilenparameter angeben, aktualisiert die EB CLI den CNAME der Standardumgebung.

Optionen

Name	Beschreibung
<code>-n</code> oder <code>--destination_name</code>	Gibt den Namen der Umgebung an, mit welcher der CNAME ausgetauscht werden soll. Falls Sie <code>eb swap</code> ohne diese Option ausführen, werden Sie von der EB CLI aufgefordert, aus einer Liste eine Umgebung auszuwählen.
Allgemeine Optionen (p. 1097)	

Ausgabe

Bei Erfolg gibt der Befehl den Status des swap-Vorgangs aus.

Beispiele

Im folgenden Beispiel wird die Umgebung "tmp-dev" mit "live-env" ausgetauscht.

```
$ 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'.
```

Im folgenden Beispiel wird die Umgebung "tmp-dev" mit der Umgebung "live-env" ausgetauscht, aber Sie werden nicht aufgefordert, einen Wert für eine Einstellung einzugeben oder auszuwählen.

```
$ 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

Beschreibung

Sie können Tags einer Elastic Beanstalk-Ressource hinzufügen, löschen, aktualisieren und auflisten.

Weitere Informationen zum Ressourcen-Tagging in Elastic Beanstalk finden Sie unter [Markieren von Elastic Beanstalk-Anwendungsressourcen \(p. 420\)](#).

Syntax

```
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 ...]
```

Sie können die Unterbefehlsoptionen --add, --update und --delete in einem einzelnen Befehl kombinieren. Es ist mindestens eine der Optionen erforderlich. Diese drei Unterbefehlsoptionen können nicht mit --list kombiniert werden.

Ohne zusätzliche Argumente werden mit diesen Befehlen Tags der Standardumgebung in der Anwendung des aktuellen Verzeichnisses aufgelistet oder geändert. Mit dem Argument **environment-name** führen diese Befehle Tags dieser Umgebung auf oder ändern sie. Mit der --resource-Option führen die Befehle Tags einer beliebigen Elastic Beanstalk-Ressource – eine Anwendung, eine Umgebung, eine Anwendungsversion, eine gespeicherte Konfiguration oder eine benutzerdefinierte Plattformversion – auf oder ändern sie. Geben Sie die -Ressource anhand ihres Amazon Resource Name (ARN) an.

Optionen

Keine dieser Optionen ist erforderlich. Wenn Sie eb create ohne Optionen ausführen, werden Sie aufgefordert, einen Wert für jede Einstellung einzugeben oder auszuwählen.

Name	Beschreibung
-l oder --list	Listen Sie alle Tags auf, die derzeit auf die Umgebung angewendet werden.

Name	Beschreibung
-a <code>key1=value1[,key2=value2 ...]</code> oder --add <code>key1=value1[,key2=value2 ...]</code>]Wenden Sie neue Tags auf die Ressource an. Legen Sie Tags als durch Komma getrennte Liste von key=value-Paaren fest. Sie können keine Schlüssel für vorhandene Tags festlegen. ...] Folgende Werte sind zulässig: Siehe Markieren von Ressourcen (p. 420) .
-u <code>key1=value1[,key2=value2 ...]</code> oder --update <code>key1=value1[,key2=value2 ...]</code>]Aktualisieren Sie die Werte von vorhandenen Ressourcen-Tags. Legen Sie Tags als durch Komma getrennte Liste von key=value-Paaren fest. Sie müssen Schlüssel für vorhandene Tags festlegen. ...] Folgende Werte sind zulässig: Siehe Markieren von Ressourcen (p. 420) .
-d <code>key1[,key2 ...]</code> oder --delete <code>key1[,key2 ...]</code>	Löschen Sie vorhandene Ressourcen-Tags. Legen Sie Tags als durch Komma getrennte Liste von Schlüsseln fest. Sie müssen Schlüssel für vorhandene Tags festlegen. ...] Folgende Werte sind zulässig: Siehe Markieren von Ressourcen (p. 420) .
-r <code>region</code> oder --region <code>region</code>	Die AWS-Region, in der Ihre Ressource vorhanden ist. Standard: die konfigurierte Standardregion Eine Liste der Werte, die Sie für diese Option festlegen können, finden Sie unter AWS Elastic Beanstalk Endpunkte und Kontingente in der Allgemeinen AWS-Referenz.
--resource <code>ARN</code>	Der ARN der Ressource, die vom Befehl geändert wird oder für die Tags aufgelistet werden. Wenn Sie diesen Parameter nicht angeben, bezieht sich der Befehl auf die (Standard- oder angegebene) Umgebung im aktuellen Verzeichnis. Gültige Werte: Weitere Informationen finden Sie in einem der Unterthemen von Markieren von Ressourcen (p. 420) , das sich auf die entsprechende Ressource bezieht. Diese Themen zeigen, wie der ARN der Ressource aufgebaut ist, und erläutern, wie Sie eine Liste der ARNs dieser Ressource, die für Ihre Anwendung oder Ihr Konto vorhanden sind, abrufen.

Ausgabe

Mit der --list-Unterbefehloption können Sie eine Liste der Tags der Ressource anzeigen. Die Ausgabe enthält sowohl die Tags, die von Elastic Beanstalk standardmäßig angewendet werden, als auch benutzerdefinierte Tags.

```
$ 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

Für die Unterbefehlsoptionen --add, --update und --delete gibt es bei erfolgreicher Ausführung keine Ausgabe. Fügen Sie die Option --verbose hinzu, um eine detaillierte Ausgabe der Befehlsaktivitäten anzuzeigen.

```
$ eb tags --verbose --update "mytag=tag value"
Updated Tags:

Key           Value
mytag         tag value
```

Beispiele

Der folgende Befehl fügt ein Tag mit dem Schlüssel tag1 und dem Wert value1 der Standardumgebung der Anwendung hinzu und löscht gleichzeitig das Tag tag2.

```
$ eb tags --add tag1=value1 --delete tag2
```

Mit dem folgenden Befehl wird einer gespeicherten Konfiguration innerhalb einer Anwendung ein Tag hinzugefügt.

```
$ eb tags --add tag1=value1 \
--resource "arn:aws:elasticbeanstalk:us-east-2:my-account-
id:configurationtemplate/my-app/my-template"
```

Der folgende Befehl schlägt fehl, da versucht wird, ein nicht vorhandenes Tag zu aktualisieren.

```
$ eb tags --update tag3=newval
ERROR: Tags with the following keys can't be updated because they don't exist:
tag3
```

Der folgende Befehl schlägt fehl, da versucht wird, denselben Schlüssel zu aktualisieren und zu löschen.

```
$ 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

Beschreibung

Die ausgeführte Umgebung wird beendet, damit keine Kosten für ungenutzte AWS-Ressourcen entstehen.

Die Verwendung der --all-Option löscht die Anwendung, für die das aktuelle Verzeichnis mit [eb init \(p. 1067\)](#) initialisiert wurde. Der Befehl beendet alle Umgebungen in der Anwendung, beendet die [Anwendungsversionen \(p. 406\)](#) der Anwendung und die [gespeicherten Konfigurationen \(p. 764\)](#) und löscht dann die Anwendung.

Falls im Stammverzeichnis eine `platform.yaml`-Datei enthalten ist, die eine benutzerdefinierte Plattform angibt, beendet dieser Befehl die ausgeführte benutzerdefinierte Umgebung.

Note

Sie können zu einem späteren Zeitpunkt problemlos eine neue Umgebung mit derselben Version starten. Sofern Sie Daten aus einer Umgebung behalten möchten, erstellen Sie einen Snapshot der aktuellen DB-Instance, bevor Sie die Umgebung beenden. Diesen können Sie später bei der Erstellung einer neuen Umgebung als Basis für die neue DB-Instance nutzen. Weitere Informationen finden Sie unter [Erstellen von DB-Snapshots](#) im Amazon Relational Database Service-Benutzerhandbuch.

Syntax

eb terminate

eb terminate **environment-name**

Optionen

Name	Beschreibung
--all	Beendet alle Umgebungen in der Anwendung, die Anwendungsversionen (p. 406) der Anwendung sowie ihre gespeicherten Konfigurationen (p. 764) und löscht dann die Anwendung.
--force	Die Umgebung wird ohne Bestätigungsaufforderung beendet.
--ignore-links	Die Umgebung wird beendet, auch wenn abhängige Umgebungen mit Links zu dieser vorhanden sind. Weitere Informationen finden Sie im Abschnitt zu den Compose-Umgebungen (p. 1038) .
--timeout	Die Anzahl der Minuten, bevor ein Timeout für den Befehl auftritt.

Ausgabe

Bei Erfolg gibt der Befehl den Status des terminate-Vorgangs aus.

Beispiel

In der folgenden Beispielanforderung wird die Umgebung "tmp-dev" beendet.

```
$ 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-16V08YOF2KQ7U
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-SL4LHODMOMU
```

```
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

Beschreibung

Aktualisiert die Plattform Ihrer Umgebung auf die neueste Version der Plattform, auf der Sie derzeit ausgeführt wird.

Wenn das Stammverzeichnis eine `platform.yaml`-Datei enthält, die eine benutzerdefinierte Plattform angibt, aktualisiert dieser Befehl die Umgebung auf die neueste Version der benutzerdefinierten Plattform, auf der sie derzeit ausgeführt wird.

Syntax

`eb upgrade`

`eb upgrade environment-name`

Optionen

Name	Beschreibung
<code>--force</code>	Nimmt eine Aktualisierung vor, ohne dass Sie den Umgebungsnamen bestätigen müssen, bevor Sie den Aktualisierungsvorgang starten.
<code>--noroll</code>	Aktualisiert alle Instances ohne fortlaufende Aktualisierungen, damit einige Instances während der Aktualisierung in Betrieb bleiben.
Allgemeine Optionen (p. 1097)	

Ausgabe

Der Befehl zeigt eine Übersicht über die Änderung und fordert Sie auf, die Aktualisierung zu bestätigen, indem Sie den Umgebungsnamen eingeben. Bei Erfolg wird Ihre Umgebung aktualisiert und dann mit der neuesten Version der Plattform gestartet.

Beispiel

Im folgenden Beispiel wird die aktuelle Plattformversion der angegebenen Umgebung auf die aktuell verfügbare Plattformversion aktualisiert.

```
$ 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

Beschreibung

Legt die angegebene Umgebung als Standardumgebung fest.

Bei der Verwendung von Git legt eb use die Standardumgebung für die aktuelle Verzweigung fest. Führen Sie diesen Befehl einmal in jeder Verzweigung aus, die Sie für Elastic Beanstalk bereitstellen möchten.

Syntax

eb use **environment-name**

Optionen

Name	Beschreibung
--source codecommit/ repository-name / branch-name	CodeCommit-Repository und Verzweigung. Siehe Verwenden der EB CLI mit AWS CodeCommit (p. 1028) .
-r region --region region	Ändern Sie die Region, in der Sie Umgebungen erstellen.
Allgemeine Optionen (p. 1097)	

Allgemeine Optionen

Sie können die folgenden Optionen mit allen EB CLI-Befehlen verwenden.

Name	Beschreibung
--debug	Ausdrucken von Informationen für das Debugging.
-h, --help	Anzeigen der Hilfenachricht. Typ: Zeichenkette Standard: keiner
--no-verify-ssl	Überspringen der SSL-Zertifikatprüfung. Verwenden Sie diese Option, wenn bei der Nutzung der CLI mit einem Proxy Probleme auftreten.
--profile	Verwenden eines bestimmten Profils aus den AWS-Anmeldeinformationen.
--quiet	Unterdrücken der gesamten Ausgabe des Befehls.
--region	Verwenden der angegebenen Region.

Name	Beschreibung
-v, --verbose	Anzeigen von ausführlichen Informationen.

EB-CLI 2.6 (veraltet)

Diese Version der EB CLI sowie die zugehörige Dokumentation wurden durch Version 3 ersetzt (in diesem Abschnitt steht EB CLI 3 für Version 3 und höher der EB CLI). Weitere Informationen zur neuen Version finden Sie unter [Verwenden des Elastic Beanstalk Command Line Interface \(EB CLI\) \(p. 1008\)](#).

Sie sollten auf die neueste Version von EB CLI 3 migrieren. Es kann Umgebungen verwalten, die Sie mit EB CLI 2.6 oder früheren Versionen von EB CLI gestartet haben.

Unterschiede zu Version 3 von EB-CLI

Bei EB handelt es sich um ein Befehlszeilenschnittstellen-Tool für Elastic Beanstalk, die Sie verwenden können, um Anwendungen schnell und einfach bereitzustellen. Die neueste EB-Version wurde von Elastic Beanstalk in EB CLI 3 eingeführt. EB CLI ruft automatisch Einstellungen aus einer Umgebung auf, die mithilfe von EB erstellt wurden, wenn die Umgebung ausgeführt wird. Beachten Sie, dass EB CLI 3 keine Optionseinstellungen lokal speichern, wie in früheren Versionen.

EB CLI führt die Befehle eb create, eb deploy, eb open, eb console, eb scale, eb setenv, eb config, eb terminate, eb clone, eb list, eb use, eb printenv und eb ssh ein. In EB CLI 3.1 oder höher können Sie auch den Befehl eb swap verwenden. Nur in EB CLI 3.2 können Sie die Befehle eb abort, eb platform und eb upgrade verwenden. Zusätzlich zu diesen neuen Befehlen unterscheiden sich EB CLI 3-Befehle in mehreren Fällen von EB CLI 2.6-Befehlen:

- eb init – Verwenden Sie eb init zum Erstellen eines .elasticbeanstalk-Verzeichnisses in einem vorhandenen Projektverzeichnis und erstellen Sie eine neue Elastic Beanstalk-Anwendung für das Projekt. Im Gegensatz zu früheren Versionen werden Sie von EB CLI 3 und späteren Versionen nicht aufgefordert, eine Umgebung zu erstellen.
- eb start – EB CLI 3 enthält den Befehl eb start nicht. Verwenden Sie eb create zum Erstellen einer Umgebung.
- eb stop – EB CLI 3 enthält den Befehl eb stop nicht. Verwenden Sie eb terminate, um eine Umgebung vollständig zu beenden und zu bereinigen.
- eb push und `git aws.push` – EB CLI 3 enthält die Befehle eb push oder git aws.push nicht. Verwenden Sie eb deploy zum Aktualisieren Ihres Anwendungscodes.
- eb update – EB CLI 3 enthält den Befehl eb update nicht. Verwenden Sie eb config zum Aktualisieren einer Umgebung.
- eb branch – EB CLI 3 enthält den Befehl eb branch nicht.

Weitere Informationen zur Verwendung von EB CLI 3-Befehlen zum Erstellen und Verwalten einer Anwendung finden Sie unter [EB-CLI-Befehlsreferenz \(p. 1042\)](#). Eine Anleitung zur Bereitstellung einer Beispielanwendung mithilfe von EB CLI 3 finden Sie unter [Elastic Beanstalk-Umgebungen mit der EB CLI verwalten \(p. 1020\)](#).

Zu EB CLI 3 und CodeCommit migrieren

Elastic Beanstalk hat nicht nur EB CLI 2.6 ausgemustert, es werden auch einige 2.6-Funktionalitäten entfernt. Der größte Unterschied zu 2.6 besteht darin, dass EB CLI nicht mehr nativ inkrementelle Codeaktualisierungen (eb push, git aws.push) oder Verzweigungen (eb branch) unterstützt. In diesem Abschnitt wird beschrieben, wie Sie von EB CLI 2.6 auf die neueste Version von EB CLI migrieren und CodeCommit als Ihr Code-Repository verwenden.

Wenn Sie dies nicht bereits getan haben, erstellen Sie ein Code-Repository in CodeCommit, wie in [Migrate to CodeCommit](#) beschrieben.

Sobald Sie EB CLI [installiert \(p. 1009\)](#) und [konfiguriert \(p. 1017\)](#) haben, haben Sie zwei Möglichkeiten, Ihre Anwendung Ihrem CodeCommit-Repository zuzuordnen, einschließlich einer bestimmten Verzweigung.

- Wenn eb init ausgeführt wird, wie im folgenden Beispiel, in dem *myRepo* der Name Ihres CodeCommit-Repositories und *myBranch* die Verzweigung in CodeCommit ist.

```
eb init --source codecommit/myRepo/myBranch
```

- Wenn eb deploy ausgeführt wird, wie im folgenden Beispiel, in dem *myRepo* der Name Ihres CodeCommit-Repositories und *myBranch* die Verzweigung in CodeCommit ist.

```
eb deploy --source codecommit/myRepo/myBranch
```

Weitere Informationen, einschließlich Anweisungen zum Bereitstellen von inkrementellen Codeaktualisierungen für eine Elastic Beanstalk-Umgebung, ohne dass das gesamte Projekt erneut hochgeladen werden muss, finden Sie unter [Verwenden der EB CLI mit AWS CodeCommit \(p. 1028\)](#).

Elastic Beanstalk API-Befehlszeilenschnittstelle (veraltet)

Dieses Tool, die Elastic Beanstalk API-Befehlszeilenschnittstelle (API Command Line Interface, API CLI) wurde von der AWS CLI ersetzt, die API-ähnliche Befehle für alle AWS-Services bereitstellt. Informationen zu den ersten Schritten mit der AWS CLI finden Sie im AWS Command Line Interface-Benutzerhandbuch. Eine vereinfachte Befehlszeilenumgebung auf höherer Ebene bietet die [EB CLI \(p. 1008\)](#).

Konvertieren von Elastic Beanstalk API CLI-Skripten

Konvertieren Sie die veralteten EB API CLI-Skripts für die Verwendung der AWS CLI oder Tools for Windows PowerShell, damit Sie auf die neuesten Elastic Beanstalk-APIs zugreifen können. In der folgenden Tabelle werden die Elastic Beanstalk-API-basierten CLI-Befehle sowie deren Entsprechungen in der AWS CLI und den Tools for Windows PowerShell aufgeführt.

Elastic Beanstalk API CLI	AWS-CLI	AWS-Tools für Windows PowerShell
<code>elastic-beanstalk-check-dns-availability</code>	<code>check-dns-availability</code>	<code>Get-EBDNSAvailability</code>
<code>elastic-beanstalk-create-application</code>	<code>create-application</code>	<code>New-EBAApplication</code>
<code>elastic-beanstalk-create-application-version</code>	<code>create-application-version</code>	<code>New-EBAApplicationVersion</code>
<code>elastic-beanstalk-create-configuration-template</code>	<code>create-configuration-template</code>	<code>New-EBConfigurationTemplate</code>

Elastic Beanstalk API CLI	AWS-CLI	AWS-Tools für Windows PowerShell
<code>elastic-beanstalk-create-environment</code>	<code>create-environment</code>	<code>New-EBEnvironment</code>
<code>elastic-beanstalk-create-storage-location</code>	<code>create-storage-location</code>	<code>New-EBStorageLocation</code>
<code>elastic-beanstalk-delete-application</code>	<code>delete-application</code>	<code>Remove-EBAApplication</code>
<code>elastic-beanstalk-delete-application-version</code>	<code>delete-application-version</code>	<code>Remove-EBAApplicationVersion</code>
<code>elastic-beanstalk-delete-configuration-template</code>	<code>delete-configuration-template</code>	<code>Remove-EBConfigurationTemplate</code>
<code>elastic-beanstalk-delete-environment-configuration</code>	<code>delete-environment-configuration</code>	<code>Remove-EBEnvironmentConfiguration</code>
<code>elastic-beanstalk-describe-application-versions</code>	<code>describe-application-versions</code>	<code>Get-EBAApplicationVersion</code>
<code>elastic-beanstalk-describe-applications</code>	<code>describe-applications</code>	<code>Get-EBAApplication</code>
<code>elastic-beanstalk-describe-configuration-options</code>	<code>describe-configuration-options</code>	<code>Get-EBConfigurationOption</code>
<code>elastic-beanstalk-describe-configuration-settings</code>	<code>describe-configuration-settings</code>	<code>Get-EBConfigurationSetting</code>
<code>elastic-beanstalk-describe-environment-resources</code>	<code>describe-environment-resources</code>	<code>Get-EBEnvironmentResource</code>
<code>elastic-beanstalk-describe-environments</code>	<code>describe-environments</code>	<code>Get-EBEnvironment</code>
<code>elastic-beanstalk-describe-events</code>	<code>describe-events</code>	<code>Get-EEvent</code>
<code>elastic-beanstalk-list-available-solution-stacks</code>	<code>list-available-solution-stacks</code>	<code>Get-EBAvailableSolutionStack</code>
<code>elastic-beanstalk-rebuild-environment</code>	<code>rebuild-environment</code>	<code>Start-EBEnvironmentRebuild</code>

Elastic Beanstalk API CLI	AWS-CLI	AWS-Tools für Windows PowerShell
<code>elastic-beanstalk-request-environment-info</code>	<code>request-environment-info</code>	<code>Request-EBEnvironmentInfo</code>
<code>elastic-beanstalk-restart-app-server</code>	<code>restart-app-server</code>	<code>Restart-EBAppServer</code>
<code>elastic-beanstalk-retrieve-environment-info</code>	<code>retrieve-environment-info</code>	<code>Get-EBEnvironmentInfo</code>
<code>elastic-beanstalk-swap-environment-cnames</code>	<code>swap-environment-cnames</code>	<code>Set-EBEnvironmentCNAME</code>
<code>elastic-beanstalk-terminate-environment</code>	<code>terminate-environment</code>	<code>Stop-EBEnvironment</code>
<code>elastic-beanstalk-update-application</code>	<code>update-application</code>	<code>Update-EBAccount</code>
<code>elastic-beanstalk-update-application-version</code>	<code>update-application-version</code>	<code>Update-EBAccountVersion</code>
<code>elastic-beanstalk-update-configuration-template</code>	<code>update-configuration-template</code>	<code>Update-EBConfigurationTemplate</code>
<code>elastic-beanstalk-update-environment</code>	<code>update-environment</code>	<code>Update-EBEnvironment</code>
<code>elastic-beanstalk-validate-configuration-settings</code>	<code>validate-configuration-settings</code>	<code>Test-EBConfigurationSetting</code>

AWS Elastic Beanstalk – Sicherheit

Cloud-Sicherheit bei AWS ist unsere oberste Priorität. Als AWS-Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die zur Erfüllung der Anforderungen von Organisationen entwickelt wurden, für die Sicherheit eine kritische Bedeutung hat.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Im [Modell der übergreifenden Verantwortlichkeit](#) wird dies als Sicherheit der Cloud und Sicherheit in der Cloud bezeichnet.

Sicherheit der Cloud – AWS ist verantwortlich für den Schutz der Infrastruktur, die alle in der AWS Cloud angebotenen Services ausführt, und für die Bereitstellung von Services, die sicher verwendet werden können./// Sicherheit hat bei AWS höchste Priorität. Die Effektivität unserer Sicherheitsmaßnahmen wird regelmäßig von externen Prüfern im Rahmen der [AWS Compliance-Programme](#) überprüft. Sehen Sie sich die [AWS-Services, die die Sicherheitsgewährleistungsprogramme von AWS](#) bestanden haben, in Bezug auf Elastic Beanstalk an.///

Sicherheit in der Cloud – Ihr Verantwortungsbereich hängt von dem AWS-Service ab, den Sie verwenden, und anderen Faktoren, einschließlich der Sensibilität Ihrer Daten, den Anforderungen Ihres Unternehmens sowie den geltenden Gesetzen und Vorschriften. Diese Dokumentation beschreibt, wie Sie das Modell der übergreifenden Verantwortlichkeit bei der Verwendung von Elastic Beanstalk anwenden können.

Sehen Sie sich die folgenden Sicherheitsthemen an, um mehr über die Sicherheitsaufgaben zu erfahren, für die Elastic Beanstalk verantwortlich ist, und werfen Sie einen Blick auf die Sicherheitskonfigurationen, die Sie berücksichtigen sollten, wenn Sie Elastic Beanstalk verwenden, um Ihre Sicherheits- und Compliance-Ziele zu erreichen.

Themen

- [Datenschutz in Elastic Beanstalk \(p. 1102\)](#)
- [Identity and Access Management für Elastic Beanstalk \(p. 1104\)](#)
- [Protokollierung und Überwachung in Elastic Beanstalk \(p. 1105\)](#)
- [Compliance-Validierung für Elastic Beanstalk \(p. 1106\)](#)
- [Resilienz in Elastic Beanstalk \(p. 1107\)](#)
- [Infrastruktursicherheit in Elastic Beanstalk \(p. 1107\)](#)
- [Konfigurations- und Schwachstellenanalyse in Elastic Beanstalk \(p. 1107\)](#)
- [Bewährte Sicherheitsmethoden für Elastic Beanstalk \(p. 1108\)](#)

Datenschutz in Elastic Beanstalk

Das [Modell der übergreifenden Verantwortlichkeit](#) von AWS gilt für den Datenschutz in AWS Elastic Beanstalk. Wie in diesem Modell beschrieben, ist AWS verantwortlich für den Schutz der globalen Infrastruktur, auf der die gesamte AWS Cloud ausgeführt wird. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Dieser Inhalt enthält die Sicherheitskonfigurations- und Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#). Weitere Informationen zum Datenschutz in Europa enthält der Blog-Post [AWS Shared Responsibility Model and GDPR](#) im AWS Security Blog.

Aus Datenschutzgründen empfehlen wir Ihnen, die Anmeldeinformationen für AWS-Konten zu schützen und individuelle Benutzerkonten mit AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem sollten Sie die Daten mit folgenden Methoden schützen:

- Verwenden Sie die Multi-Factor Authentication (MFA) für jedes Konto.
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir empfehlen TLS 1.2 oder höher.
- Richten Sie die API- und Benutzeraktivitätsprotokollierung mit AWS CloudTrail ein.
- Verwenden Sie AWS-Verschlüsselungslösungen zusätzlich zu allen standardmäßigen Sicherheitskontrollen innerhalb von AWS-Services.
- Verwenden Sie moderne verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu sichern.
- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

Wir empfehlen dringend, in Freitextfeldern wie z. B. im Feld Name keine sensiblen, identifizierenden Informationen wie Kontonummern von Kunden einzugeben. Dies gilt auch, wenn Sie mit Elastic Beanstalk oder unter Verwendung von Konsole, API, AWS CLI oder AWS SDKs mit anderen AWS-Services arbeiten. Alle Daten, die Sie in Elastic Beanstalk oder andere Services eingeben, können in Diagnoseprotokolle aufgenommen werden. Wenn Sie eine URL für einen externen Server bereitstellen, schließen Sie keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL ein.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter [AWS Elastic Beanstalk – Sicherheit \(p. 1102\)](#).

Themen

- [Datenschutz durch Verschlüsselung \(p. 1103\)](#)
- [Richtlinie für den Datenverkehr zwischen Netzwerken \(p. 1104\)](#)

Datenschutz durch Verschlüsselung

Elastic Beanstalk speichert verschiedene Objekte in einem Amazon Simple Storage Service (Amazon S3)-Bucket, den es für jede AWS-Region erstellt, in der Sie Umgebungen erstellen. Details dazu finden Sie unter [the section called “Amazon S3” \(p. 985\)](#).

Sie stellen einige der gespeicherten Objekte bereit und senden sie an Elastic Beanstalk, z. B. Anwendungsversionen und Quell-Bundles. Elastic Beanstalk generiert andere Objekte, z. B. Protokolldateien. Zusätzlich zu den Daten, die Elastic Beanstalk speichert, kann Ihre Anwendung Daten im Rahmen ihrer Operation übertragen und/oder speichern.

Datenschutz bezieht sich auf den Schutz von Daten während der Übertragung (auf dem Weg von und nach Elastic Beanstalk) und im Ruhezustand (während sie in AWS-Rechenzentren gespeichert sind).

Verschlüsselung während der Übertragung

Sie können den Datenschutz während der Übertragung auf zwei Arten erreichen: Verschlüsseln Sie die Verbindung mit Secure Sockets Layer (SSL) oder verwenden Sie die clientseitige Verschlüsselung (wobei das Objekt vor dem Senden verschlüsselt wird). Beide Methoden sind für den Schutz Ihrer Anwendungsdaten gültig. Um die Verbindung zu sichern, verschlüsseln Sie sie mit SSL, wenn Ihre Anwendung, ihre Entwickler und Administratoren und ihre Endbenutzer Objekte senden oder empfangen.

Weitere Informationen zum Verschlüsseln von Web-Datenverkehr zu und von Ihrer Anwendung finden Sie unter [the section called “HTTPS” \(p. 778\)](#).

Die clientseitige Verschlüsselung ist keine gültige Methode zum Schutz Ihres Quellcodes in Anwendungsversionen und Quell-Bundles, die Sie hochladen. Elastic Beanstalk benötigt Zugriff auf diese Objekte, sodass sie nicht verschlüsselt werden können. Stellen Sie daher sicher, dass Sie die Verbindung zwischen Ihrer Entwicklungs- oder Bereitstellungsumgebung und Elastic Beanstalk sichern.

Verschlüsselung im Ruhezustand

Um die Daten Ihrer Anwendung im Ruhezustand zu schützen, erfahren Sie mehr über den Datenschutz im Speicherservice, den Ihre Anwendung verwendet. Beispiele finden Sie unter [Datenschutz in Amazon RDS](#) im Amazon RDS-Benutzerhandbuch, [Datenschutz in Amazon S3](#) im Amazon Simple Storage Service-Entwicklerhandbuch oder [Verschlüsseln von Daten und Metadaten in EFS](#) im Amazon Elastic File System-Benutzerhandbuch.

Elastic Beanstalk aktiviert nicht die Standardverschlüsselung für den erstellten Amazon S3-Bucket. Das bedeutet, dass Objekte standardmäßig unverschlüsselt im Bucket gespeichert werden (und nur für Benutzer zugänglich sind, die zum Lesen des Buckets autorisiert sind). Wenn Ihre Anwendung eine Verschlüsselung im Ruhezustand erfordert, können Sie die Buckets Ihres Kontos für die Standardverschlüsselung konfigurieren. Weitere Informationen finden Sie unter [Amazon S3-Standardverschlüsselung für S3-Buckets](#) im Amazon Simple Storage Service-Entwicklerhandbuch.

Weitere Informationen zum Datenschutz enthält der Blog-Post [AWS Shared Responsibility Model and GDPR](#) im AWS Security Blog.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter [AWS Elastic Beanstalk – Sicherheit \(p. 1102\)](#).

Richtlinie für den Datenverkehr zwischen Netzwerken

Sie können Amazon Virtual Private Cloud (Amazon VPC) verwenden, um Grenzen zwischen Ressourcen in Ihrer Elastic Beanstalk-Anwendung zu erstellen und den Datenverkehr zwischen ihnen, Ihrem lokalen Netzwerk und dem Internet zu steuern. Details dazu finden Sie unter [the section called “Amazon VPC” \(p. 988\)](#).

Weitere Informationen zur Amazon VPC-Sicherheit finden Sie unter [Sicherheit](#) im Amazon VPC-Benutzerhandbuch.

Weitere Informationen zum Datenschutz enthält der Blog-Post [AWS Shared Responsibility Model and GDPR](#) im AWS Security Blog.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter [AWS Elastic Beanstalk – Sicherheit \(p. 1102\)](#).

Identity and Access Management für Elastic Beanstalk

AWS Identity and Access Management (IAM) ist ein AWS-Service, der es einem Administrator ermöglicht, den Zugriff auf AWS-Ressourcen sicher zu steuern. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (Berechtigungen besitzt) ist, AWS Elastic Beanstalk-Ressourcen zu nutzen. IAM ist ein AWS-Service, die Sie ohne zusätzliche Kosten verwenden können.

Weitere Informationen zum Arbeiten mit IAM finden Sie unter [Verwenden von Elastic Beanstalk mit AWS Identity and Access Management \(p. 903\)](#).

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter [AWS Elastic Beanstalk – Sicherheit \(p. 1102\)](#).

Protokollierung und Überwachung in Elastic Beanstalk

Die Überwachung ist wichtig für die Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von AWS Elastic Beanstalk und Ihrer AWS-Lösungen. Sie sollten Überwachungsdaten aus allen Teilen der AWS-Lösung erfassen, damit Sie Ausfälle, die sich über mehrere Punkte erstrecken, leichter debuggen können. AWS stellt mehrere Tools für die Überwachung Ihrer Elastic Beanstalk-Ressourcen und zur Reaktion auf potenzielle Vorfälle bereit:

Weitere Informationen zur Überwachung finden Sie unter [Überwachen einer Umgebung \(p. 817\)](#).

Andere Elastic Beanstalk-Sicherheitsthemen finden Sie unter [AWS Elastic Beanstalk – Sicherheit \(p. 1102\)](#).

Erweiterte Zustandsberichte

Erweiterte Zustandsberichte ist eine Funktion, die Sie auf Ihrer Umgebung aktivieren können, damit Elastic Beanstalk weitere Informationen über Ressourcen in Ihrer Umgebung erfassen kann. Elastic Beanstalk analysiert die gesammelten Informationen, um ein besseres Bild des Gesamtumgebungszustands zu erhalten und um bei der Identifizierung von Problemen zu helfen, die dazu führen können, dass Ihre Anwendung nicht mehr verfügbar ist. Weitere Informationen finden Sie unter [Erweiterte Zustandsberichte und Überwachung \(p. 824\)](#).

Amazon EC2-Instance-Protokolle

Die Amazon EC2-Instances der Elastic Beanstalk-Umgebung generieren Protokolle, die Sie zur Behebung von Fehlern mit der Anwendung oder mit Konfigurationsdateien aufrufen können. Protokolle, die vom Webserver, Anwendungsserver, Elastic Beanstalk-Plattform-Scripts und AWS CloudFormation erstellt werden, werden lokal auf einzelnen Instances gespeichert. Sie können diese über die [Environment Management Console \(p. 426\)](#) oder die EB CLI ganz einfach abrufen. Sie können Ihre Umgebung auch so konfigurieren, dass Protokolle in Echtzeit an Amazon CloudWatch Logs gestreamt werden. Weitere Informationen finden Sie unter [Protokolle von Amazon EC2-Instances in Ihrer Elastic Beanstalk Umgebung anzeigen \(p. 869\)](#).

Umgebungsbenachrichtigungen

Sie können Ihre Elastic Beanstalk Umgebung so konfigurieren, dass Sie Amazon Simple Notification Service (Amazon SNS) verwenden, um Sie über wichtige Ereignisse zu benachrichtigen, die Ihre Anwendung betreffen. Geben Sie eine E-Mail-Adresse während oder nach der Umgebungserstellung an, um E-Mails von AWS zu erhalten, wenn ein Fehler auftritt oder sich der Anwendungsstatus Ihrer Umgebung ändert. Weitere Informationen finden Sie unter [Elastic Beanstalk-Umgebungsbenachrichtigungen mit Amazon SNS \(p. 633\)](#).

Amazon CloudWatch-Alarne

Mit CloudWatch-Alarmen überwachen Sie eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum. Wenn die Metrik einen bestimmten Schwellenwert überschreitet, wird eine Benachrichtigung an ein Amazon SNS-Thema oder eine AWS Auto Scaling-Richtlinie gesendet. CloudWatch-Alarne rufen keine Aktionen auf, nur weil sie sich in einem bestimmten Status befinden. Stattdessen rufen

Alarme Aktionen auf, wenn sich der Status geändert hat und für eine bestimmte Anzahl von Zeiträumen aufrechterhalten wurde. Weitere Informationen finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon CloudWatch \(p. 879\)](#).

AWS CloudTrail-Protokolle

CloudTrail bietet eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem AWS-Service in Elastic Beanstalk durchgeführten Aktionen. Mit den von CloudTrail erfassten Informationen können Sie die an CloudTrail gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen. Weitere Informationen finden Sie unter [Protokollieren von Elastic Beanstalk-API-Aufrufen mit AWS CloudTrail \(p. 878\)](#).

AWS X-Ray-Debugging

X-Ray ist ein AWS-Service, der Daten über die Anforderungen erfasst, die Ihre Anwendung verwendet, um eine Servicezuordnung zu erstellen, die Sie verwenden können, um Probleme mit Ihrer Anwendung und Möglichkeiten zur Optimierung zu identifizieren. Sie können die AWS Elastic Beanstalk-Konsole oder eine Konfigurationsdatei verwenden, um den X-Ray-Daemon auf den Instances in Ihrer Umgebung auszuführen. Weitere Informationen finden Sie unter [Konfigurieren von AWS X-Ray Debugging \(p. 627\)](#).

Compliance-Validierung für Elastic Beanstalk

Die Sicherheit und Compliance von AWS Elastic Beanstalk wird von externen Prüfern im Rahmen mehrerer AWS-Compliance-Programme bewertet. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA. AWS bietet eine häufig aktualisierte Liste von AWS-Services, die von bestimmten Compliance-Programmen zugelassen sind. Diese sind unter [AWS-Services in Scope nach Compliance-Programm](#) aufgeführt.

Prüfungsberichte von Drittanbietern stehen Ihnen zum Download über AWS Artifact zur Verfügung. Weitere Informationen finden Sie unter [Herunterladen von Berichten in AWS Artifact](#).

Weitere Informationen zu AWS-Compliance-Programmen finden Sie unter [AWS-Compliance-Programme](#).

Welche Compliance-Verpflichtungen Sie bei der Verwendung von Elastic Beanstalk haben, hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihrer Organisation und den geltenden Gesetzen und Vorschriften ab. Wenn Ihre Verwendung von Elastic Beanstalk der Compliance mit Standards wie HIPAA, PCI oder FedRAMP unterliegt, stellt AWS Ressourcen zur Unterstützung bereit:

- [Kurzanleitungen zu Sicherheit und Compliance](#) – Bereitstellungsleitfäden, die Überlegungen zur Architektur enthalten und Schritte für die Bereitstellung von Baseline-Umgebungen mit dem Schwerpunkt auf Sicherheit und Compliance in AWS beschreiben.
- [Architekturen für Sicherheit und Compliance im Zusammenhang mit HIPAA](#) – Ein Whitepaper, in dem beschrieben wird, wie Unternehmen AWS verwenden können, um HIPAA-konforme Anwendungen zu erstellen.
- [AWS-Compliance-Ressourcen](#) – Eine Sammlung von Compliance-Arbeitsmappen und -Anleitungen, die für Ihre Branche und Ihren Standort möglicherweise relevant sind.
- [AWS Config](#) – Ein Service, der die Compliance Ihrer Ressourcenkonfigurationen mit internen Praktiken, Branchenrichtlinien und Vorschriften bewertet.
- [AWS Security Hub](#) – Eine umfassende Ansicht Ihres Sicherheitsstatus innerhalb von AWS, die Ihnen hilft, Ihre Compliance mit Standards und bewährten Methoden der Sicherheitsbranche zu überprüfen.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter [AWS Elastic Beanstalk – Sicherheit \(p. 1102\)](#).

Resilienz in Elastic Beanstalk

Im Zentrum der globalen AWS-Infrastruktur stehen AWS-Regionen und Availability Zones (Verfügbarkeitszonen, AZs).

AWS-Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über hoch redundante Netzwerke mit niedriger Latenz und hohen Durchsätzen verbunden sind.

Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS-Regionen und Availability Zones finden Sie unter [Weltweite AWS-Infrastruktur](#).

AWS Elastic Beanstalk verwaltet und automatisiert die Nutzung der globalen AWS-Infrastruktur in Ihrem Namen. Bei der Verwendung von Elastic Beanstalk profitieren Sie von den Verfügbarkeits- und Fehlertoleranzmechanismen, die AWS bietet.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter [AWS Elastic Beanstalk – Sicherheit \(p. 1102\)](#).

Infrastruktursicherheit in Elastic Beanstalk

Als verwalteter Service ist AWS Elastic Beanstalk durch die globalen Verfahren zur Gewährleistung der Netzwerksicherheit von AWS geschützt, die im Whitepaper [Amazon Web Services: Übersicht über die Sicherheitsprozesse](#) beschrieben sind.

Sie verwenden von AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Elastic Beanstalk zuzugreifen. Clients müssen Transport Layer Security (TLS) 1.0 oder höher unterstützen. Wir empfehlen TLS 1.2 oder höher. Clients müssen außerdem Cipher Suites mit PFS (Perfect Forward Secrecy) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. Die meisten modernen Plattformen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit dem [AWS Security Token Service \(AWS STS\)](#) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter [AWS Elastic Beanstalk – Sicherheit \(p. 1102\)](#).

Konfigurations- und Schwachstellenanalyse in Elastic Beanstalk

AWS und unseren Kunden haben eine gemeinsame Verantwortung, um ein hohes Maß an Sicherheit und Compliance für Softwarekomponenten zu erzielen. AWS Elastic Beanstalk unterstützt Sie bei Umsetzung Ihres Anteils am Modell übergreifender Verantwortlichkeit, indem eine Funktion für verwaltete Aktualisierungen bereitgestellt wird. Diese Funktion wendet automatisch Patches und kleinere Aktualisierungen für eine von Elastic Beanstalk unterstützte Platformversion an.

Weitere Informationen finden Sie unter [Modell der übergreifenden Verantwortlichkeit für die Wartung der Elastic Beanstalk-Plattform \(p. 29\)](#).

Weitere Themen zur Elastic Beanstalk-Sicherheit finden Sie unter [AWS Elastic Beanstalk – Sicherheit \(p. 1102\)](#).

Bewährte Sicherheitsmethoden für Elastic Beanstalk

AWS Elastic Beanstalk bietet mehrere Sicherheitsfunktionen, die Sie beim Entwickeln und Implementieren Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden bewährten Methoden sind allgemeine Richtlinien und keine vollständige Sicherheitslösung. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

Andere Elastic Beanstalk-Sicherheitsthemen finden Sie unter [AWS Elastic Beanstalk – Sicherheit \(p. 1102\)](#).

Bewährte Methoden für vorbeugende Sicherheitsmaßnahmen

Vorbeugende Sicherheitskontrollen versuchen, Vorfälle zu verhindern, bevor sie auftreten.

Implementieren der geringstmöglichen Zugriffsrechte

Elastic Beanstalk bietet von AWS Identity and Access Management (IAM) verwaltete Richtlinien für [Instance-Profile \(p. 904\)](#), [Servicerollen \(p. 909\)](#) und [IAM-Benutzer \(p. 926\)](#). Diese verwalteten Richtlinien geben alle Berechtigungen an, die möglicherweise für den korrekten Betrieb Ihrer Umgebung und Anwendung erforderlich sind.

Ihre Anwendung benötigt möglicherweise nicht alle Berechtigungen in unseren verwalteten Richtlinien. Sie können sie anpassen und nur die Berechtigungen erteilen, die für die Instances Ihrer Umgebung, den Elastic Beanstalk-Service und Ihre Benutzer zum Ausführen ihrer Aufgaben erforderlich sind. Dies ist besonders relevant für Benutzerrichtlinien, in denen unterschiedliche Benutzerrollen möglicherweise unterschiedliche Berechtigungsanforderungen haben. Die Implementierung der geringstmöglichen Zugriffsrechte ist eine grundlegende Voraussetzung zum Reduzieren des Sicherheitsrisikos und der Auswirkungen, die aufgrund von Fehlern oder böswilligen Absichten entstehen könnten.

Aktualisieren Sie Ihre Plattformen regelmäßig

Elastic Beanstalk veröffentlicht regelmäßig neue Plattformversionen, um alle Plattformen zu aktualisieren. Neue Plattformversionen bieten Betriebssystem-, Laufzeit-, Anwendungsserver- und Webserver-Updates sowie Updates für Elastic Beanstalk-Komponenten. Viele dieser Plattformaktualisierungen enthalten wichtige Sicherheitskorrekturen. Stellen Sie sicher, dass Ihre Elastic Beanstalk-Umgebungen auf einer unterstützten Plattformversion ausgeführt werden (in der Regel die neueste Version für Ihre Plattform). Details dazu finden Sie unter [Aktualisieren der Plattformversion für die Elastic Beanstalk-Umgebung \(p. 491\)](#).

Die einfachste Möglichkeit, die Plattform Ihrer Umgebung auf dem neuesten Stand zu halten, besteht darin, die Umgebung so zu konfigurieren, dass [verwaltete Plattformaktualisierungen \(p. 496\)](#) verwendet werden.

Erzwingen von IMDSv2 auf Umgebungs-Instances

Amazon Elastic Compute Cloud (Amazon EC2)-Instances in Ihren Elastic Beanstalk-Umgebungen verwenden den Instance Metadata Service (IMDS), eine Instances-Komponente, um sicher auf

Instances-Metadaten zuzugreifen. IMDS unterstützt zwei Methoden für den Datenzugriff: IMDSv1 und IMDSv2. IMDSv2 verwendet sitzungsorientierte Anfragen und mildert verschiedene Arten von Sicherheitsschwachstellen, über die versucht werden kann, auf das IMDS zuzugreifen. Ausführliche Informationen zu den Vorteilen von IMDSv2 finden Sie unter [Verbesserungen, um den EC2-Instance-Metadatendienst umfassend zu verteidigen](#).

IMDSv2 ist sicherer, daher empfiehlt es sich, die Verwendung von IMDSv2 auf Ihren Instances zu erzwingen. Um IMDSv2 zu erzwingen, stellen Sie sicher, dass alle Komponenten Ihrer Anwendung IMDSv2 unterstützen, und deaktivieren Sie dann IMDSv1. Weitere Informationen finden Sie unter [the section called "IMDS" \(p. 538\)](#).

Bewährte Methoden für aufdeckende Sicherheitsmaßnahmen

Aufdeckende Sicherheitskontrollen identifizieren Sicherheitsverstöße, nachdem sie aufgetreten sind. Sie können Ihnen dabei helfen, potenzielle Sicherheitsbedrohungen oder -vorfälle zu erkennen.

Implementieren der Überwachung

Die Überwachung ist ein wichtiger Teil der Aufrechterhaltung von Zuverlässigkeit, Verfügbarkeit und Performance Ihrer Elastic Beanstalk-Lösungen. AWS bietet verschiedene Tools und Services, die Sie bei der Überwachung Ihrer AWS-Services unterstützen.

Es folgen einige Beispiele für zu überwachende Elemente:

- Amazon CloudWatch-Metriken für Elastic Beanstalk – Legen Sie Alarme für wichtige Elastic Beanstalk-Metriken und für die benutzerdefinierten Metriken Ihrer Anwendung fest. Details dazu finden Sie unter [Verwenden von Elastic Beanstalk mit Amazon CloudWatch \(p. 879\)](#).
- AWS CloudTrail-Einträge – Verfolgen Sie Aktionen, die sich auf die Verfügbarkeit auswirken können, wie z. B. `UpdateEnvironment` oder `TerminateEnvironment`. Details dazu finden Sie unter [Protokollieren von Elastic Beanstalk-API-Aufrufen mit AWS CloudTrail \(p. 878\)](#).

AWS Config aktivieren

AWS Config bietet eine detaillierte Ansicht der Konfiguration der AWS-Ressourcen in Ihrem AWS-Konto. Sie können Ressourcenbeziehungen betrachten, einen Verlauf der Konfigurationsänderungen anzeigen und feststellen, wie sich Beziehungen und Konfigurationen im Laufe der Zeit ändern.

Mit AWS Config können Sie Regeln definieren, die Ressourcenkonfigurationen im Hinblick auf Datencompliance bewerten. Die AWS Config-Regeln stellen die idealen Konfigurationseinstellungen für Ihre Elastic Beanstalk-Ressourcen dar. Wenn eine Ressource gegen eine Regel verstößt und als nicht konform gekennzeichnet ist, kann AWS Config Sie mit einem Amazon Simple Notification Service (Amazon SNS)-Thema benachrichtigen. Details dazu finden Sie unter [Suchen und Verfolgen von Elastic Beanstalk-Ressourcen mit AWS Config \(p. 895\)](#).

Fehlersuche

In diesem Kapitel finden Sie eine Tabelle mit den am häufigsten auftretenden Elastic Beanstalk-Problemen sowie die Fehlerbehebung oder einen Workaround. Fehlermeldungen können als Ereignisse auf der Umgebungsübersichtsseite der Konsole, in Protokollen oder auf der Zustandsprüfungsseite angezeigt werden.

Ändert sich der Zustand der Umgebung in rot, führen Sie die folgenden Schritte aus:

- Überprüfen Sie die letzten [Ereignisse \(p. 864\)](#) in der Umgebung. Hier finden Sie häufig Meldungen von Elastic Beanstalk zu Bereitstellungs-, Auslastungs- und Konfigurationsproblemen.
- [Rufen Sie die Protokolle auf \(p. 869\)](#) und sehen Sie sich die letzten Protokolldateieinträge an. Die Webserverprotokolle können Informationen zu eingehenden Anforderungen und Fehlern enthalten.
- [Stellen Sie die Verbindung zu einer Instance \(p. 866\)](#) her und überprüfen Sie die Systemressourcen.
- Stellen Sie eine vorherige funktionierende Version der Anwendung [wieder her \(p. 471\)](#).
- Machen Sie die letzten Konfigurationsänderungen rückgängig oder stellen Sie eine [gespeicherte Konfiguration \(p. 651\)](#) wieder her.
- Stellen Sie eine neue Umgebung bereit. Wenn diese stabil wirkt, führen Sie einen [CNAME-Austausch \(p. 481\)](#) durch, um den Datenverkehr in die neue Umgebung umzuleiten, und setzen Sie die Fehlerbehebung in der alten Umgebung fort.

Themen

- [Konnektivität \(p. 1110\)](#)
- [Umgebungserstellung und Instance-Starts \(p. 1111\)](#)
- [Bereitstellungen \(p. 1111\)](#)
- [Integrität \(p. 1112\)](#)
- [Konfiguration \(p. 1112\)](#)
- [Fehlerbehebung bei Docker-Containern \(p. 1112\)](#)
- [Häufig gestellte Fragen \(p. 1113\)](#)

Konnektivität

Problem: Verbindung zu Amazon RDS aus Elastic Beanstalk nicht möglich.

Führen Sie die folgenden Schritte aus, um RDS mit Ihrer Elastic Beanstalk-Anwendung zu verbinden.

- Stellen Sie sicher, dass sich RDS in der derselben Region wie Ihre Elastic Beanstalk-Anwendung befindet.
- Stellen Sie sicher, dass die RDS-Sicherheitsgruppe für Ihre Instance über eine Autorisierung für die Amazon EC2-Sicherheitsgruppe verfügt, die Sie für Ihre Elastic Beanstalk-Umgebung verwenden. Anweisungen zum Suchen des Namens Ihrer EC2-Sicherheitsgruppe mit der AWS Management Console finden Sie unter [Sicherheitsgruppen \(p. 535\)](#). Weitere Informationen zum Konfigurieren Ihrer EC2-Sicherheitsgruppe finden Sie im Abschnitt "Autorisieren von Netzwerkzugriff auf eine Amazon EC2-Sicherheitsgruppe" unter [Arbeiten mit DB-Sicherheitsgruppen](#) im Amazon Relational Database Service-Benutzerhandbuch.
- Stellen Sie bei der Verwendung von Java sicher, dass sich die MySQL JAR-Datei in Ihrer WEB-INF/lib befindet. Weitere Details finden Sie unter [Hinzufügen einer Amazon RDS-DB-Instance zur Java-Anwendungsumgebung \(p. 134\)](#).

Problem: Server, die in der Elastic Beanstalk-Konsole erstellt wurden, werden nicht im Toolkit for Eclipse angezeigt

Sie können Server manuell importieren, indem Sie die Anweisungen unter [Importieren von vorhandenen Umgebungen in Eclipse \(p. 141\)](#) befolgen.

Umgebungserstellung und Instance-Starts

Ereignis: Umgebung konnte nicht gestartet werden

Dieses Ereignis tritt auf, wenn Elastic Beanstalk versucht, eine Umgebung zu starten und es dabei zu Fehlern kommt. Die vorherigen Ereignisse auf der Seite Events (Ereignisse) weisen auf die Ursache des Problems hin.

Ereignis: Die Erstellung der Umgebung wurde mit Befehls-Timeouts abgeschlossen. Try increasing the timeout period.

Falls Sie Konfigurationsdateien einsetzen, mit denen Befehle auf der Instance ausgeführt, große Dateien heruntergeladen oder Pakete installiert werden, kann die Anwendungsbereitstellung langwierig sein. Erhöhen Sie den [Timeout-Wert für Befehle \(p. 475\)](#), damit die Anwendung im Rahmen der Bereitstellungen mehr Zeit zum Ausführen hat.

Ereignis: Die folgende(n) Ressource(n) konnten nicht erstellt werden:
[AWSEBInstanceLaunchWaitCondition]

Diese Nachricht gibt an, dass die Amazon EC2-Instances der Umgebung Elastic Beanstalk nicht über den erfolgreichen Start informiert haben. Das kann passieren, wenn die Instances keine Verbindung zum Internet haben. Wenn die Umgebungskonfiguration das Starten der Instances in einem privaten VPC-Subnetz vorsieht, [stellen Sie sicher, dass das Subnetz über NAT verfügt \(p. 988\)](#), damit die Instances eine Verbindung zu Elastic Beanstalk herstellen können.

Ereignis: In dieser Region ist eine Servicerolle erforderlich. Please add a Service Role option to the environment.

Elastic Beanstalk nutzt eine Servicerolle zur Überwachung der Ressourcen in Ihrer Umgebung und zur Unterstützung von [verwalteten Plattformaktualisierungen \(p. 496\)](#). Weitere Informationen finden Sie unter [Rollen von Elastic Beanstalk Service verwalten \(p. 909\)](#).

Bereitstellungen

Problem: Anwendung steht während Bereitstellungen nicht zur Verfügung

Da Elastic Beanstalk ein Drop-in-Upgrade ausführt, kann es zu einer Ausfallzeit von einigen Sekunden kommen. Setzen Sie [fortlaufende Bereitstellungen \(p. 474\)](#) ein, um die Beeinträchtigung der Produktionsumgebungen aufgrund von Bereitstellungen zu minimieren.

Ereignis: AWS Elastic Beanstalk-Anwendungsversion konnte nicht erstellt werden

Möglicherweise ist das Quell-Bundle der Anwendung zu umfangreich oder das [Anwendungsversionskontingent \(p. 406\)](#) wurde erreicht.

Ereignis: Umgebungsaktualisierung wurde mit Befehls-Timeouts abgeschlossen. Try increasing the timeout period.

Falls Sie Konfigurationsdateien einsetzen, mit denen Befehle auf der Instance ausgeführt, große Dateien heruntergeladen oder Pakete installiert werden, kann die Anwendungsbereitstellung langwierig sein.

Erhöhen Sie den [Timeout-Wert für Befehle \(p. 475\)](#), damit die Anwendung im Rahmen der Bereitstellungen mehr Zeit zum Ausführen hat.

Integrität

Ereignis: CPU-Auslastung Über 95 %

Versuchen Sie, [mehr Instances auszuführen \(p. 540\)](#), oder [wählen Sie einen anderen Instance-Typ aus \(p. 531\)](#).

Ereignis: Elastic Load Balancer awseb-*myapp* hat keine fehlerfreien Instances

Sofern die Anwendung ordnungsgemäß ausgeführt werden kann, überprüfen Sie, ob die Zustandsprüfungs-URL für die Anwendung korrekt konfiguriert ist. Falls nicht, überprüfen Sie die Zustandsprüfung und die Umgebungsprotokolle, um weitere Informationen zu erhalten.

Ereignis: Elastic Load Balancer awseb-*myapp* wurde nicht gefunden

Der Load Balancer der Umgebung ist möglicherweise außerplanmäßig entfernt worden. Nehmen Sie Änderungen an den Umgebungsressourcen nur mit den von Elastic Beanstalk bereitgestellten Konfigurationsoptionen und [Erweiterungen \(p. 722\)](#) vor. Erstellen Sie Ihre Umgebung erneut oder starten Sie eine neue Umgebung.

Ereignis: EC2-Instance konnte nicht gestartet werden. Waiting for a New EC2 Instance to Launch...

Möglicherweise sind derzeit keine Instance-Typen für die Umgebung verfügbar oder das Instance-Kontingent für Ihr Konto wurde erreicht. Überprüfen Sie in der [Übersicht zum Servicestatus](#), ob der Elastic Compute Cloud (Amazon EC2)-Service den Status "Grün" aufweist, oder [fordern Sie eine Kontingenterhöhung an](#).

Konfiguration

Ereignis: Es kann keine Elastic Beanstalk-Umgebung konfiguriert werden, die Werte für die Optionen "Elastic Load Balancing Target (Elastic Load Balancing-Ziel)" und "Application Healthcheck URL (URL für die Zustandsprüfung der Anwendung)" enthält.

Die Option `Target` im Namespace `aws : elb : healthcheck` ist veraltet. Entfernen Sie die Namespace-Option `Target` aus der Umgebung und führen Sie die Aktualisierung erneut aus.

Ereignis: ELB kann nicht im selben AZ zu verschiedenen Subnetzen hinzugefügt werden.

Diese Meldung wird angezeigt, wenn Sie einen Load Balancer zwischen Subnetzen in derselben Availability Zone verschieben möchten. Sollen die Subnetze des Load Balancers geändert werden, muss dieser aus den ursprünglichen Availability Zones entfernt und mit den gewünschten Subnetzen wieder in diese eingefügt werden. Bei diesem Prozess werden alle Instances zwischen den Availability Zones migriert, was zu erheblichen Ausfallzeiten führt. Ziehen Sie stattdessen in Betracht, eine neue Umgebung zu erstellen und einen [CNAME-Austausch durchzuführen \(p. 481\)](#).

Fehlerbehebung bei Docker-Containern

Ereignis: Docker-Image :latest: konnte nicht abgerufen werden. Ungültiger Repository-Name (), nur [a-z0-9-] sind zulässig. Weitere Details finden Sie in der Protokollfragmentierung.

Prüfen Sie die Syntax der `dockerrun.aws.json`-Datei mit einer JSON-Validierung. Verifizieren Sie außerdem den dockerfile-Inhalt hinsichtlich der Anforderungen in [Docker-Konfiguration \(p. 53\)](#).

Ereignis: Keine EXPOSE-Anweisung in Dockerfile gefunden, Bereitstellung abbrechen

Die Dockerfile oder `dockerrun.aws.json`-Datei deklariert den Container-Port nicht. Verwenden Sie die EXPOSE-Anweisung (Dockerfile) oder den Ports-Block (`dockerrun.aws.json`-Datei), um einen Port für eingehenden Datenverkehr verfügbar zu machen.

Ereignis: **Repository** mit Authentifizierungs-Anmeldeinformationen konnte nicht aus dem **Bucket-Namen** heruntergeladen werden.

Die `dockerrun.aws.json` bietet ein ungültiges EC2-Schlüsselpaar und/oder einen ungültigen S3-Bucket für die `.dockercfg`-Datei. Oder das Instance-Profil hat keine GetObject-Autorisierung für den S3-Bucket. Überprüfen Sie, ob die `.dockercfg`-Datei einen gültigen S3-Bucket und ein gültiges EC2-Schlüsselpaar enthält. Erteilen Sie Berechtigungen für die Aktion `s3:GetObject` für die IAM-Rolle im Instance-Profil. Details finden Sie unter [Elastic Beanstalk Instance-Profile verwalten \(p. 904\)](#).

Ereignis: Aktivitätsausführung wegen Warnmeldung zur ungültigen "auth"-Konfigurationsdatei fehlgeschlagen

Die Authentifizierungsdatei (`config.json`) ist nicht ordnungsgemäß formatiert. Siehe [Verwenden von Images aus einem privaten Repository \(p. 90\)](#).

Häufig gestellte Fragen

Frage: Wie kann ich meine Anwendungs-URL von `myapp.us-west-2.elasticbeanstalk.com` in `www.myapp.com` ändern?

Registrieren Sie bei einem DNS-Server einen CNAME-Datensatz, wie z. B. `www.mydomain.com CNAME mydomain.elasticbeanstalk.com`.

Frage: Wie gebe ich eine bestimmte Availability Zone für meine Elastic Beanstalk-Anwendung an?

Sie können bestimmte Availability Zones mithilfe der APIs, CLI, des Eclipse-Plug-In oder des Visual Studio-Plug-In auswählen. Für Anweisungen zum Angeben einer Availability Zone mit der Elastic Beanstalk-Konsole siehe [Auto Scaling-Gruppe für Ihre Elastic Beanstalk-Umgebung \(p. 540\)](#).

Frage: Wie kann ich den Instance-Typ meiner Umgebung ändern?

Wählen Sie auf der Umgebungskonfigurationsseite die Option Edit (Bearbeiten) in der Konfigurationskategorie Instances. Wählen Sie einen neuen Instance-Typ aus und klicken Sie auf Apply (Anwenden) zum Aktualisieren Ihrer Umgebung. Elastic Beanstalk beendet alle laufenden Instances und ersetzt sie durch die neuen.

Frage: Kann ich verhindern, dass Amazon EBS-Volumes gelöscht werden, wenn Instances beendet werden?

Instances in Ihrer Umgebung verwenden Amazon EBS zur Speicherung. Das Stamm-Volume wird jedoch gelöscht, wenn eine Instance durch Auto Scaling beendet wird. Es ist nicht empfohlen, Status- oder andere Daten auf Ihren Instances zu speichern. Bei Bedarf können Sie mit der AWS CLI verhindern, dass Volumes gelöscht werden: `$ aws ec2 modify-instance-attribute -b '/dev/sdc=<vol-id>:false`, wie in der [AWS CLI Reference](#) beschrieben.

Frage: Wie lösche ich persönliche Informationen aus meiner Elastic Beanstalk-Anwendung?

AWS-Ressourcen, die von Ihrer Elastic Beanstalk-Anwendung genutzt werden, speichern möglicherweise personenbezogene Informationen. Wenn Sie eine Umgebung beenden, beendet Elastic Beanstalk die

erstellten Ressourcen. Die Ressourcen, die Sie mithilfe von [Konfigurationsdateien \(p. 722\)](#) hinzugefügt haben, werden ebenfalls beendet. Wenn Sie jedoch AWS-Ressourcen außerhalb Ihrer Elastic Beanstalk-Umgebung erstellt und mit Ihrer Anwendung verknüpft haben, müssen Sie möglicherweise manuell sicherstellen, dass personenbezogene Informationen, die Ihre Anwendung möglicherweise gespeichert hat, nicht unnötig gespeichert werden. In diesem Entwicklerhandbuch wird überall dort, wo wir über die Erstellung zusätzlicher Ressourcen sprechen, auch erwähnt, wann Sie diese löschen sollten.

Elastic Beanstalk-Ressourcen

Die folgenden verwandten Ressourcen bieten Ihnen nützliche Informationen für die Arbeit mit diesem Service.

- [Elastic Beanstalk-API-Referenz](#) Eine umfassende Beschreibung aller SOAP- und Abfrage-APIs. Zudem ist eine Liste aller SOAP-Datentypen enthalten.
- [elastic-beanstalk-samples auf GitHub](#) – Ein GitHub-Repository mit Elastic Beanstalk-Beispielkonfigurationsdateien (.ebextensions). Die README.md-Datei des Repositorys enthält Links zu weiteren GitHub-Repositories mit Beispielanwendungen.
- [Häufig gestellte technische Fragen zu Elastic Beanstalk](#) – Die wichtigsten Fragen, die Entwickler zu diesem Produkt gestellt haben.
- [Versionshinweise zu AWS Elastic Beanstalk](#) – Details zu neuen Funktionen, Updates und Fehlerbehebungen im Zusammenhang mit dem Elastic-Beanstalk-Service, der Plattform, der Konsole und der EB CLI.
- [Kurse und Workshops](#) – Links zu rollenbasierten und speziellen Kursen sowie Übungen im Selbststudium zur Verbesserung Ihrer AWS-Kompetenzen und für praktische Erfahrung.
- [AWS-Entwickler-Tools](#) – Links zu Entwickler-Tools, SDKs, IDE-Toolkits und Befehlszeilen-Tools für die Entwicklung und Verwaltung von AWS-Anwendungen.
- [AWS-Whitepaper](#) – Links zu einer umfangreichen Liste technischer AWS-Whitepaper zu Themen wie Architektur, Sicherheit und Wirtschaftlichkeit. Diese Whitepaper wurden von AWS-Lösungsarchitekten und anderen technischen Experten verfasst.
- [AWS Support Center](#) – Der zentrale Ort für die Erstellung und Verwaltung Ihrer AWS Support-Fälle. Bietet außerdem Links zu hilfreichen Ressourcen wie z. B. Foren, technischen Fragen und Antworten, Übersicht zum Servicestatus und AWS Trusted Advisor.
- [AWS Support](#) – Die Hauptwebsite mit Informationen zum AWS Support ist ein persönlicher und reaktionsschneller Support-Kanal, der Sie beim Konfigurieren und Ausführen von Anwendungen in der Cloud unterstützt.
- [Kontaktieren Sie uns](#) – Ein zentraler Kontaktpunkt für Anfragen zur AWS-Abrechnung, zu Konten, Ereignissen, Missbrauch und anderen Problemen.
- [Nutzungsbedingungen für die AWS-Website](#) – Detaillierte Informationen zu unseren Copyright- und Markenbestimmungen, Ihrem Konto, den Lizizenzen, Websitezugriff und anderen Themen.

Beispielanwendungen

Nachfolgend finden Sie Links zu den herunterladbaren Beispielanwendungen, die unter [Erste Schritte mit Elastic Beanstalk \(p. 3\)](#) bereitgestellt werden.

Note

In einigen Beispielen werden Funktionen verwendet, die möglicherweise nach der von Ihnen verwendeten Plattformversion veröffentlicht wurden. Falls das Beispiel nicht ausgeführt werden kann, versuchen Sie, Ihre Plattform wie unter [the section called “Unterstützte Plattformen” \(p. 32\)](#) beschrieben auf die neueste Version zu aktualisieren.

- Docker – [docker.zip](#)
- Multicontainer Docker – [docker-multicontainer-v2.zip](#)
- Vorkonfigurierter Docker (Glassfish) – [docker-glassfish-v1.zip](#)

- Go – [go.zip](#)
- Corretto – [corretto.zip](#)
- Tomcat – [tomcat.zip](#)
- .NET Core unter Linux – [dotnet-core-linux.zip](#)
- .NET – [dotnet-asp-v1.zip](#)
- Node.js – [nodejs.zip](#)
- PHP – [php.zip](#)
- Python – [python.zip](#)
- Ruby – [ruby.zip](#)

Plattformverlauf

Die Historie der AWS Elastic Beanstalk-Plattform wurde verschoben. Siehe [Plattform-Historie](#) im Dokument zu AWS Elastic Beanstalk-Plattformen.

Themen

- [Benutzerdefinierte Elastic Beanstalk-Plattformen \(p. 1117\)](#)

Benutzerdefinierte Elastic Beanstalk-Plattformen

AWS Elastic Beanstalk unterstützt benutzerdefinierte Plattformen. Eine benutzerdefinierte Plattform ist mehr eine erweiterte Anpassung als ein [benutzerdefiniertes Image \(p. 773\)](#). Eine benutzerdefinierte Plattform gestattet Ihnen, eine ganze Plattform von Grund auf neu zu entwickeln, das Betriebssystem anzupassen sowie zusätzliche Software und Skripts hinzuzufügen, die Elastic Beanstalk auf Plattform-Instances ausführt. Diese Flexibilität ermöglicht Ihnen den Aufbau einer Plattform für eine Anwendung, die eine Sprache oder andere Infrastruktur-Software verwendet, für die Elastic Beanstalk keine verwaltete Plattform bietet. Vergleichen Sie dies mit benutzerdefinierten Abbildern, bei denen Sie ein Amazon Machine Image (AMI) für die Verwendung mit einer vorhandenen Elastic Beanstalk-Plattform ändern und Elastic Beanstalk weiterhin die Plattform-Skripts bereitstellt und den Software-Stack der Plattform steuert. Darüber hinaus verwenden Sie bei benutzerdefinierten Plattformen eine automatisierte, skriptgesteuerte Methode, Ihre Anpassung zu erstellen und zu verwalten, während Sie bei benutzerdefinierten Abbildern die Änderungen manuell über eine laufende Instance ausführen.

Note

Benutzerdefinierte Elastic Beanstalk-Plattformen unterstützen ausschließlich das Erstellen eines AMI über AMIs auf der Basis von Amazon Linux AMI, RHEL 7, RHEL 6 oder Ubuntu 16.04. Amazon Linux 2 oder andere Betriebssysteme werden von der Funktionalität für benutzerdefinierte Plattformen nicht unterstützt.

Um eine benutzerdefinierte Plattform zu erstellen, erzeugen Sie ein AMI aus einem der unterstützten Betriebssysteme – Ubuntu, RHEL oder Amazon Linux (die genauen Versionsnummern finden Sie im Eintrag `flavor` unter [Platform.yaml-Dateiformat \(p. 1125\)](#)) – und fügen weitere Anpassungen hinzu. Sie erstellen Ihre eigene Elastic Beanstalk-Plattform mit [Packer](#), einem Open Source-Tool zum Erstellen von Machine Images für zahlreiche Plattformen, darunter auch AMIs für Amazon Elastic Compute Cloud (Amazon EC2). Eine Elastic Beanstalk-Plattform umfasst ein AMI, das zur Ausführung der anwendungsunterstützenden Software konfiguriert ist, sowie Metadaten, die benutzerdefinierte Konfigurationsoptionen und Standardeinstellungen der Konfigurationsoptionen enthalten können.

Elastic Beanstalk verwaltet Packer als separate integrierte Plattform und Sie brauchen sich nicht um die Konfiguration und Versionen von Packer zu kümmern.

Sie erstellen eine Plattform, indem Sie eine Packer-Vorlage sowie die von der Vorlage aufgerufenen Skripts und Dateien für die AMI-Erstellung in Elastic Beanstalk bereitstellen. Diese Komponenten werden mit einer [Plattformdefinitionsdatei \(p. 1118\)](#), in der die Vorlage und die Metadaten spezifiziert werden, in ein ZIP-Archiv verpackt, das als [Plattformdefinitionsarchiv \(p. 1123\)](#) bezeichnet wird.

Wenn Sie eine benutzerdefinierte Plattform erstellen, starten Sie eine Umgebung mit einer einzelnen Instance ohne eine Elastic IP, die Packer ausführt. Packer startet dann eine weitere Instance, um ein Image

zu erstellen. Sie können diese Umgebung für mehrere Plattformen und verschiedene Versionen jeder Plattform einsetzen.

Note

Benutzerdefinierte Plattformen sind AWS-regionsspezifisch. Falls Sie Elastic Beanstalk in mehreren Regionen nutzen, müssen Sie die Plattformen in jeder Region separat erstellen. In bestimmten Situationen werden vom Packer gestartete Instances nicht gelöscht und müssen manuell beendet werden. Informationen zum manuellen Bereinigen dieser Instances finden Sie unter [Bereinigen von Instances durch Packer \(p. 1124\)](#).

Die Benutzer in Ihrem Konto können die benutzerdefinierten Plattformen verwenden, indem sie während der Umgebungserstellung einen [Plattform-ARN \(p. 933\)](#) angeben. Diese ARNs werden vom Befehl `eb platform create` zurückgegeben, mit dem Sie die benutzerdefinierte Plattform erstellt haben.

Jedes Mal, wenn Sie eine benutzerdefinierte Plattform erstellen, generiert Elastic Beanstalk eine neue Plattformversion. Benutzer können den Namen einer Plattform angeben, damit nur die neueste Version der Plattform aufgerufen wird, oder eine Versionsnummer spezifizieren, um eine bestimmte Version zu erhalten.

Beispielsweise stellen Sie die benutzerdefinierte Plattform mit dem ARN `MyCustomPlatformARN` und der neuesten Version 3.0 mit folgendem EB CLI-Befehl bereit:

```
eb create -p MyCustomPlatformARN
```

Für die Bereitstellung von Version 2.1 sieht der EB CLI-Befehl wie folgt aus:

```
eb create -p MyCustomPlatformARN --version 2.1
```

Sie können eine benutzerdefinierte Plattformversion mit Tags markieren, wenn Sie sie erstellen, und Tags von vorhandenen benutzerdefinierten Plattformversionen bearbeiten. Details dazu finden Sie unter [Markieren von benutzerdefinierten Plattformversionen \(p. 1127\)](#).

Erstellen einer benutzerdefinierten Plattform

Damit Sie eine benutzerdefinierte Plattform erstellen können, muss der Anwendungsstamm eine Plattformdefinitionsdatei enthalten, `platform.yaml`. Diese definiert den Builder-Typ, mit dem die benutzerdefinierte Plattform erstellt wird. Das Format dieser Datei ist in [Platform.yaml-Dateiformat \(p. 1125\)](#) beschrieben. Sie können die benutzerdefinierte Plattform von Grund auf neu erstellen oder eines der [benutzerdefinierten Plattformbeispiele \(p. 1118\)](#) als Ausgangspunkt verwenden.

Verwenden eines benutzerdefinierten Plattformbeispiels

Als Alternative zum Erstellen einer eigenen benutzerdefinierten Plattform können Sie eines der Plattformdefinitionsarchiv-Beispiele für den Bootstrap der benutzerdefinierten Plattform nutzen. Die einzigen Elemente, die Sie in den Beispielen konfigurieren müssen, bevor Sie sie verwenden können, sind ein Quell-AMI und eine Region.

Note

Verwenden Sie kein nicht abgeändertes Beispiel für eine benutzerdefinierte Plattform in der Produktion. Das Ziel der Beispiele besteht darin, einige verfügbare Funktionen für eine

benutzerdefinierte Plattform zu veranschaulichen, sie sind jedoch nicht für den Einsatz in der Produktion gehärtet worden.

[NodePlatform_Ubuntu.zip](#)

Diese benutzerdefinierte Plattform basiert auf Ubuntu 16.04 und unterstützt Node.js 4.4.4. In den folgenden Beispielen wird diese benutzerdefinierte Plattform verwendet.

[NodePlatform_RHEL.zip](#)

Diese benutzerdefinierte Plattform basiert auf RHEL 7.2 und unterstützt Node.js 4.4.4.

[NodePlatform_AmazonLinux.zip](#)

Diese benutzerdefinierte Plattform basiert auf Amazon Linux 2016.09.1 und unterstützt Node.js 4.4.4.

[TomcatPlatform_Ubuntu.zip](#)

Diese benutzerdefinierte Plattform basiert auf Ubuntu 16.04 und unterstützt Tomcat 7/Java 8.

[CustomPlatform_NodeSampleApp.zip](#)

In diesem Node.js-Beispiel wird eine statische Webseite mithilfe von express und ejs angezeigt.

[CustomPlatform_TomcatSampleApp.zip](#)

In diesem Tomcat-Beispiel wird eine statische Webseite bei der Bereitstellung angezeigt.

Laden Sie das Plattformdefinitionsarchiv-Beispiel herunter: [NodePlatform_Ubuntu.zip](#). Diese Datei enthält eine Plattformdefinitionsdatei, eine Packer-Vorlage, Skripts, die während der Image-Erstellung von Packer ausgeführt werden, sowie Skripts und Konfigurationsdateien, die von Packer im Rahmen der Plattformerstellung auf die Builder-Instance kopiert werden.

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
```

Die Plattformdefinitionsdatei, `platform.yaml`, teilt Elastic Beanstalk den Namen der Packer-Vorlage mit, `custom_platform.json`.

```
version: "1.0"

provisioner:
  type: packer
  template: custom_platform.json
  flavor: ubuntu1604
```

Mithilfe der Packer-Vorlage wird Packer angewiesen, die AMIs für die Plattform zu erstellen, und zwar mit einem [Ubuntu-AMI](#) als Basis für das Plattform-Image von HVM-Instance-Typen. Der Abschnitt `provisioners` weist Packer an, alle Dateien im Archiv-Ordner `builder` auf die Instance zu kopieren und das Skript `builder.sh` auf der Instance auszuführen. Nach Abschluss des Skripts erstellt Packer ein Image von der geänderten Instance.

Elastic Beanstalk erstellt drei Umgebungsvariablen, mit denen AMIs in Packer getaggt werden können:

AWS_EB_PLATFORM_ARN

Der ARN der benutzerdefinierten Plattform.

AWS_EB_PLATFORM_NAME

Der Name der benutzerdefinierten Plattform.

AWS_EB_PLATFORM_VERSION

Die Version der benutzerdefinierten Plattform.

Die Beispieldatei `custom_platform.json` nutzt diese Variablen, um die folgenden in den Skripts verwendeten Werte zu definieren:

- `platform_name`, wird festgelegt von der Datei `platform.yaml`.
- `platform_version`, wird festgelegt von der Datei `platform.yaml`.
- `platform_arn`, wird festgelegt vom Build-Hauptskript `builder.sh`, das sich am Ende der Beispieldatei `custom_platform.json` befindet.

Die Datei `custom_platform.json` enthält zwei Eigenschaften, für die Sie Werte angeben müssen: `source_ami` und `region`. Weitere Informationen über die Auswahl der richtigen Werte für AMI und Region finden Sie unter [Aktualisieren der Packer-Vorlage](#) im GitHub-Repository `eb-custom-platforms-samples`.

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"
      ]
    }
  ]
}
```

Die Skripts und andere Dateien, die Sie in das Plattformdefinitionsarchiv einbinden, sind je nach den Änderungen, die Sie an der Instance vornehmen möchten, sehr unterschiedlich. Das Plattformbeispiel enthält die folgenden Skripte:

- `00-sync-apt.sh` – Auskommentiert: `apt -y update`. Wir haben den Befehl auskommentiert, weil er den Benutzer zu einer Eingabe auffordert, die die automatische Paketaktualisierung unterbricht. Dies könnte ein Ubuntu-Problem sein. Als bewährte Methode empfehlen wir jedoch weiterhin, `apt -y update` auszuführen. Aus diesem Grund haben wir den Befehl zur Referenz im Beispiel-Skript beibehalten.
- `01-install-nginx.sh` – Installiert nginx.

- 02-setup-platform.sh – Installiert wget, tree und git. Kopiert Hooks sowie [Protokollierungskonfigurationen \(p. 869\)](#) auf die Instance und erstellt die folgenden Verzeichnisse:
 - /etc/SampleNodePlatform – Gibt an, wohin die Container-Konfigurationsdatei während der Bereitstellung hochgeladen wird.
 - /opt/elasticbeanstalk/deploy/appsource/ – Gibt an, wohin der Anwendungsquellcode während der Bereitstellung vom 00-unzip.sh-Skript hochgeladen wird (weitere Informationen zu diesem Skript finden Sie im Abschnitt [Plattform-Skript-Tools \(p. 43\)](#)).
 - /var/app/staging/ – Gibt an, wo der Anwendungsquellcode während der Bereitstellung verarbeitet wird.
 - /var/app/current/ – Gibt an, wo der Anwendungsquellcode nach der Verarbeitung ausgeführt wird.
 - /var/log/nginx/healthd/ – Gibt an, wohin die Protokolle vom [Agent für erweiterte Integritätsberichte \(p. 827\)](#) geschrieben werden.
 - /var/nodejs – Gibt an, wohin die Node.js-Dateien während der Bereitstellung hochgeladen werden.

Verwenden Sie die EB CLI, um Ihre erste benutzerdefinierte Plattform mit dem Plattformdefinitionsarchiv-Beispiel zu erstellen.

So erstellen Sie eine benutzerdefinierte Plattform

1. [Installieren Sie die EB CLI \(p. 1009\)](#).
2. Erstellen Sie ein Verzeichnis, in welches das benutzerdefinierte Plattformbeispiel extrahiert wird.

```
~$ mkdir ~/custom-platform
```

3. Extrahieren Sie `NodePlatform_Ubuntu.zip` in das Verzeichnis und wechseln Sie dann zu diesem Verzeichnis.

```
~$ cd ~/custom-platform
~/custom-platform$ unzip ~/NodePlatform_Ubuntu.zip
~/custom-platform$ cd NodePlatform_Ubuntu
```

4. Bearbeiten Sie die Datei `custom_platform.json` und geben Sie Werte für die Eigenschaften `region` und `source_ami` an. Weitere Informationen finden Sie unter [Aktualisieren der Packer-Vorlage](#).
5. Führen Sie [eb platform init \(p. 1081\)](#) aus und folgen Sie den Anweisungen, um ein Plattform-Repository zu initialisieren.

Sie können eb platform zu ebp verkürzen.

Note

Windows PowerShell verwendet ebp als Befehls-Alias. Wenn Sie die EB CLI in Windows PowerShell ausführen, müssen Sie die Langform dieses Befehls verwenden eb platform.

```
~/custom-platform$ eb platform init
```

Mit diesem Befehl wird zudem das Verzeichnis `.elasticbeanstalk` im aktuellen Verzeichnis erstellt, dann wird die Konfigurationsdatei `config.yml` zum Verzeichnis hinzugefügt. Diese Datei wird von Elastic Beanstalk zum Erstellen der benutzerdefinierten Plattform benötigt, daher darf sie weder geändert noch gelöscht werden.

Standardmäßig verwendet eb platform init den Namen des aktuellen Ordners als Bezeichnung für die benutzerdefinierte Plattform, in diesem Beispiel also `custom-platform`.

6. Führen Sie [eb platform create \(p. 1079\)](#) aus, um die Packer-Umgebung zu starten und den ARN der benutzerdefinierten Plattform abzurufen. Diesen Wert benötigen Sie später zum Erstellen der Umgebung für die benutzerdefinierte Plattform.

```
~/custom-platform$ eb platform create
...
```

Standardmäßig erstellt Elastic Beanstalk das Instance-Profil `aws-elasticbeanstalk-custom-platform-ec2-role` für benutzerdefinierte Plattformen. Wenn Sie stattdessen ein vorhandenes Instance-Profil nutzen möchten, fügen Sie die Option `-ip INSTANCE_PROFILE` zum Befehl [eb platform create \(p. 1079\)](#) hinzu.

Note

Falls Sie das Instance-Standardprofil `aws-elasticbeanstalk-ec2-role` von Elastic Beanstalk nutzen, kann Packer die benutzerdefinierte Plattform nicht erstellen.

In der EB CLI wird die Ereignisausgabe der Packer-Umgebung angezeigt, bis die Erstellung abgeschlossen ist. Sie können die Ereignisanzeige verlassen, indem Sie Strg+C drücken.

7. Mit dem Befehl [eb platform logs \(p. 1082\)](#) können Sie die Protokolle auf Fehler überprüfen.

```
~/custom-platform$ eb platform logs
...
```

8. Mit [eb platform events \(p. 1080\)](#) können Sie den Prozess später prüfen.

```
~/custom-platform$ eb platform events
...
```

9. Überprüfen Sie den Status der Plattform mit [eb platform status \(p. 1082\)](#).

```
~/custom-platform$ eb platform status
...
```

Nach Abschluss des Vorgangs steht Ihnen eine Plattform zur Verfügung, auf der Sie eine Elastic Beanstalk-Umgebung starten können.

Sie können die benutzerdefinierte Plattform einsetzen, um eine neue Umgebung mit der Konsole zu erstellen. Siehe [Der Assistent zum Erstellen einer neuen Umgebung \(p. 437\)](#).

So starten Sie eine Umgebung auf der benutzerdefinierten Plattform

1. Erstellen Sie ein Verzeichnis für die Anwendung.

```
~$ mkdir custom-platform-app
~$ cd ~/custom-platform-app
```

2. Initialisieren Sie ein Anwendungs-Repository.

```
~/custom-platform-app$ eb init
...
```

3. Laden Sie die Beispieldatenbank [NodeSampleApp.zip](#) herunter.

4. Extrahieren Sie die Beispieldatenbank.

```
~/custom-platform-app$ unzip -/NodeSampleApp.zip
```

5. Führen Sie eb create -p **CUSTOM-PLATFORM-ARN** aus (wobei **CUSTOM-PLATFORM-ARN** der vom Befehl eb platform create zurückgegebene ARN ist), um eine Umgebung zu starten, in der die benutzerdefinierte Plattform ausgeführt wird.

```
~/custom-platform-app$ eb create -p CUSTOM-PLATFORM-ARN
...

```

Inhalte des Plattformdefinitionsarchivs

Ein Plattformdefinitionsarchiv ist die Plattformentsprechung vom [Quell-Bundle der Anwendung \(p. 412\)](#). Das Plattformdefinitionsarchiv ist eine ZIP-Datei und enthält eine Plattformdefinitionsdatei, eine Packer-Vorlage sowie die von der Packer-Vorlage verwendeten Skripts und Dateien für die Plattformerstellung.

Note

Wenn Sie die EB CLI zum Erstellen der benutzerdefinierten Plattform verwenden, generiert die EB CLI ein Plattformdefinitionsarchiv aus den Dateien und Ordern im Plattform-Repository, daher muss das Archiv nicht manuell angelegt werden.

Die Datei Plattformdefinitionsdatei weist das YAML-Format auf, muss den Namen `platform.yaml` haben und im Plattformdefinitionsarchiv-Stamm sein. Unter [Erstellen einer benutzerdefinierten Plattform \(p. 1118\)](#) finden Sie eine Liste der erforderlichen und optionalen Schlüssel, die in einer Plattformdefinitionsdatei unterstützt werden.

Die Packer-Vorlage muss nicht nach einer bestimmten Konvention benannt werden, jedoch muss dieser Dateiname mit der Provisioner-Vorlage, die in der Plattformdefinitionsdatei angegeben ist, übereinstimmen. Anleitungen zum Erstellen von Packer-Vorlagen finden Sie in der offiziellen [Packer-Dokumentation](#).

Bei den anderen Dateien im Plattformdefinitionsarchiv handelt es sich um Skripts und Dateien, die von der Vorlage vor der AMI-Erstellung zur Instance-Anpassung verwendet werden.

Benutzerdefinierte Plattform-Hooks

Elastic Beanstalk verwendet eine standardisierte Verzeichnisstruktur für Hooks auf benutzerdefinierten Plattformen. Dies sind Skripts, die bei Lebenszyklusereignissen und als Reaktion auf Verwaltungsvorgänge ausgeführt werden, z. B. wenn Instances in der Umgebung gestartet werden oder wenn ein Benutzer eine Bereitstellung initiiert oder die Funktion für den Neustart des Anwendungsservers nutzt.

Platzieren Sie Skripts, die Hooks auslösen sollen, in einem der Unterordner des Ordners `/opt/elasticbeanstalk/hooks/`.

Warning

Die Verwendung benutzerdefinierter Plattform-Hooks auf verwalteten Plattformen wird nicht unterstützt. Benutzerdefinierte Plattform-Hooks sind für benutzerdefinierte Plattformen bestimmt. Auf von Elastic Beanstalk verwalteten Plattformen funktionieren sie möglicherweise anders oder weisen Probleme auf und das Verhalten kann sich je nach Plattform unterscheiden. Auf Amazon Linux AMI-Plattformen (vor Amazon Linux 2) funktionieren sie möglicherweise immer noch auf nützliche Weise. Verwenden Sie sie jedoch mit Vorsicht.

Benutzerdefinierte Plattform-Hooks sind eine Legacy-Funktion auf Amazon Linux AMI-Plattformen. Auf Amazon Linux 2-Plattformen werden benutzerdefinierte Plattform-Hooks im Ordner `/opt/elasticbeanstalk/hooks/` vollständig eingestellt. Elastic Beanstalk liest oder führt sie nicht aus. Amazon Linux 2-Plattformen unterstützen eine neue Art von Plattform-Hooks, die speziell zur Erweiterung von verwalteten Elastic Beanstalk-Plattformen entwickelt wurden. Sie können benutzerdefinierte Skripte und Programme direkt zu einem Hook-Verzeichnis in Ihrem

Anwendungs-Quell-Bundle hinzufügen. Elastic Beanstalk führt sie während verschiedener Instance-Bereitstellungsphasen aus. Weitere Informationen erhalten Sie, indem Sie den Abschnitt Plattform-Hooks unter [the section called “Erweitern von Linux-Plattformen” \(p. 34\)](#) erweitern.

Hooks sind in den folgenden Ordnern organisiert:

- `appdeploy` – Skripts, die während der Anwendungsbereitstellung ausgeführt werden. Wenn neue Instances gestartet werden oder ein Client eine neue Versionsbereitstellung initiiert, führt Elastic Beanstalk eine Anwendungsbereitstellung aus.
- `configdeploy` – Skripts, die im Rahmen von clientseitigen Konfigurationsaktualisierungen, welche sich auf die Softwarekonfiguration der Instance auswirken (z. B. durch Festlegen von Umgebungseigenschaften oder Aktivieren der Protokollrotation an Amazon S3), ausgeführt werden.
- `restartappserver` – Skripts, die bei einem clientseitigen Neustart des Anwendungsservers ausgeführt werden.
- `preinit` – Skripts, die während des Instance-Bootstrapping ausgeführt werden.
- `postinit` – Skripts, die nach dem Instance-Bootstrapping ausgeführt werden.

Die Ordner `appdeploy`, `configdeploy` und `restartappserver` enthalten die Unterordner `pre`, `enact` und `post`. In den einzelnen Vorgangsphasen werden alle Skripts im Ordner `pre` in alphabetischer Reihenfolge ausgeführt, dann folgen diejenigen im Ordner `enact` und anschließend diejenigen im Ordner `post`.

Beim Starten einer Instance führt Elastic Beanstalk `preinit`, `appdeploy` und `postinit` in dieser Reihenfolge aus. Bei nachfolgenden Bereitstellungen auf ausgeführten Instances führt Elastic Beanstalk die `appdeploy`-Hooks aus. Die `configdeploy`-Hooks werden ausgeführt, wenn ein Benutzer die Instance-Softwarekonfigurationseinstellungen aktualisiert. Die `restartappserver`-Hooks werden nur ausgeführt, wenn ein Benutzer den Neustart des Anwendungsservers initiiert.

Falls die Skripts auf Fehler treffen, werden sie mit einem Status ungleich null beendet und schreiben den fehlgeschlagenen Vorgang in die Datei `stderr`. Die in die Datei `stderr` geschriebene Nachricht wird im Ereignis angezeigt, das im Falle eines fehlgeschlagenen Vorgangs ausgegeben wird. Zudem werden diese Informationen von Elastic Beanstalk in der Protokolldatei `/var/log/eb-activity.log` erfasst. Wenn der Vorgang nicht als fehlgeschlagen gelten soll, muss 0 (Null) zurückgegeben werden. Nachrichten, die in die Datei `stderr` oder `stdout` geschrieben werden, erscheinen in den [Bereitstellungsprotokollen \(p. 869\)](#), werden aber nur bei einem fehlgeschlagenen Vorgang im Ereignis-Stream angegeben.

Bereinigen von Instances durch Packer

Unter bestimmten Umständen, z. B. wenn der Packer-Erstellungsprozess vorzeitig abgebrochen wird, werden von Packer gestartete Instances nicht bereinigt. Diese Instances sind nicht Teil der Elastic Beanstalk-Umgebung und lassen sich nur mit dem Amazon EC2-Service anzeigen und beenden.

So bereinigen Sie diese Instances manuell

1. Öffnen Sie die [Amazon EC2-Konsole](#).
2. Achten Sie darauf, dass Sie in derselben AWS-Region sind, in der Sie die Instance mit Packer erstellt haben.
3. Wählen Sie unter Resources (Ressourcen) den Wert **N** Running Instances (Ausgeführte Instances) aus, wobei **N** für die Anzahl der ausgeführten Instances steht.
4. Klicken Sie in das Abfragetextfeld.
5. Wählen Sie das Tag Name aus.
6. Geben Sie `packer` ein.

Die Abfrage sollte wie folgt aussehen: tag:Name: packer

7. Wählen Sie die Instances aus, die mit der Abfrage übereinstimmen.
8. Wenn der Wert für Instance State (Instance-Status) mit running (wird ausgeführt) angegeben wird, wählen Sie Actions (Aktionen), Instance State (Instance-Status), Stop (Anhalten) und dann Actions (Aktionen), Instance State (Instance-Status), Terminate (Beenden) aus.

Platform.yaml-Dateiformat

Die `platform.yaml`-Datei hat folgendes Format:

```
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"
```

Ersetzen Sie die Platzhalter mit folgenden Werten:

version-number

Erforderlich. Die Version der YAML-Definition. Der Wert muss **1.0** sein.

provisioner-type

Erforderlich. Der Typ des Builders, der für die Erstellung der benutzerdefinierten Plattform verwendet wurde. Der Wert muss **packer** sein.

provisioner-template

Erforderlich. Die JSON-Datei mit den Einstellungen für **provisioner-type**.

provisioner-flavor

Optional. Das Basis-Betriebssystem für die AMI. Eine der beiden folgenden Komponenten:
amazon (Standard)

Amazon Linux. Wenn nicht angegeben, wird die aktuelle Version von Amazon Linux verwendet, wenn die Plattform erstellt wird.

Amazon Linux 2 ist keine unterstützte Betriebssystemvariante.

ubuntu1604

Ubuntu 16.04 LTS

rhel7

RHEL 7

rhel6

RHEL 6

metadata-maintainer

Optional. Kontaktinformationen für den Eigentümer der Plattform (100 Zeichen).

metadata-description

Optional. Beschreibung der Plattform (2000 Zeichen).

metadata-operating_system_name

Optional. Name der Plattform im Betriebssystem (50 Zeichen). Dieser Wert ist verfügbar, wenn die Ausgabe für die [ListPlatformVersions](#)-API gefiltert wird.

metadata-operating_system_version

Optional. Version der Plattform im Betriebssystem (20 Zeichen).

metadata-programming_language_name

Optional. Programmiersprache, die von der Plattform unterstützt wird (50 Zeichen)

metadata-programming_language_version

Optional. Version der Sprache im Betriebssystem (20 Zeichen).

metadata-framework_name

Optional. Der Name des Web-Framework, das von der Plattform verwendet wird (50 Zeichen).

metadata-framework_version

Optional. Version des Web-Framework im Betriebssystem (20 Zeichen).

option-def-namespace

Optional. Ein Namespace unter `aws:elasticbeanstalk:container:custom` (100 Zeichen)

option-def-option_name

Optional. Der Name der Option (100 Zeichen). Sie können bis zu 50 benutzerdefinierte Konfigurationsoptionen festlegen, die die Plattform Benutzern bereitstellt.

option-def-description

Optional. Beschreibung der Option (1024 Zeichen).

option-def-default_value

Optional. Der Standardwert wird verwendet, wenn der Benutzer keinen angibt.

Das folgende Beispiel erstellt die Option **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

Optional. Namespace der Option.

option-setting-option_name

Optional. Name der Option. Sie können bis zu 50 Optionen von Elastic Beanstalk (p. 666) angeben.

option-setting-value

Optional. Der Wert wird verwendet, wenn der Benutzer keinen angibt.

Das folgende Beispiel erstellt die Option **TEST**.

```
option_settings:
  - namespace: "aws:elasticbeanstalk:application:environment"
    option_name: "TEST"
    value: "This is a test"
```

Markieren von benutzerdefinierten Plattformversionen

Sie können Ihre benutzerdefinierten AWS Elastic Beanstalk-Plattformversionen mit Tags markieren. Tags sind mit AWS-Ressourcen verknüpfte Schlüssel-Wert-Paare. Weitere Informationen zum Elastic Beanstalk-Ressourcen-Tagging, zu Anwendungsfälle, Einschränkungen für Tag-Schlüssel und -Werte sowie zu unterstützten Ressourcentypen finden Sie unter [Markieren von Elastic Beanstalk-Anwendungsressourcen \(p. 420\)](#).

Sie können Tags angeben, wenn Sie eine benutzerdefinierte Plattformversion erstellen. In einer vorhandenen benutzerdefinierten Plattformversion können Sie Tags hinzufügen oder entfernen sowie die Werte von vorhandenen Tags aktualisieren. Sie können jeder benutzerdefinierten Plattformversion bis zu 50 Tags hinzufügen.

Hinzufügen von Tags beim Erstellen einer benutzerdefinierten Plattformversion

Wenn Sie mit der EB CLI Ihre benutzerdefinierte Plattform erstellen, verwenden Sie die `--tags`-Option mit dem Befehl [eb platform create \(p. 1079\)](#), um Tags hinzuzufügen.

```
~/workspace/my-app$ eb platform create --tags mytag1=value1,mytag2=value2
```

Fügen Sie Tags mit der AWS CLI oder anderen API-basierten Clients hinzu, indem Sie den `--tags`-Parameter mit dem Befehl [create-platform-version](#) verwenden.

```
$ 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
```

Verwalten von Tags einer vorhandenen benutzerdefinierten Plattformversion

In einer vorhandenen benutzerdefinierten Elastic Beanstalk-Plattformversion können Sie Tags hinzufügen, aktualisieren und löschen.

Wenn Sie Ihre benutzerdefinierte Plattformversion mit der EB CLI aktualisieren, verwenden Sie den Befehl [eb tags \(p. 1092\)](#) zum Hinzufügen, Aktualisieren, Löschen oder Auflisten von Tags.

Beispiel: Der folgende Befehl listet die Tags in einer benutzerdefinierten Plattformversion auf.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-platform/1.0.0"
```

Mit dem folgenden Befehl wird das Tag mytag1 aktualisiert und das Tag mytag2 gelöscht.

```
~/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"
```

Eine umfassende Liste der Optionen sowie weitere Beispiele finden Sie unter [eb tags \(p. 1092\)](#).

Wenn Sie die AWS CLI oder andere API-basierte Clients verwenden, nutzen Sie den Befehl [list-tags-for-resource](#), um die Tags einer benutzerdefinierten Plattformversion aufzulisten.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-platform/1.0.0"
```

Verwenden Sie den Befehl [update-tags-for-resource](#) zum Hinzufügen, Aktualisieren und Löschen von Tags in einer benutzerdefinierten Plattformversion.

```
$ 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"
```

Geben Sie sowohl hinzuzufügende als auch zu aktualisierende Tags im Parameter --tags-to-add des Befehls update-tags-for-resource an. Wenn ein Tag nicht vorhanden ist, wird es hinzugefügt, andernfalls wird es aktualisiert.

Note

Wenn Sie einige der EB CLI- und AWS CLI-Befehle mit einer benutzerdefinierten Elastic Beanstalk-Plattformversion verwenden möchten, benötigen Sie den ARN der benutzerdefinierten Plattformversion. Verwenden Sie den folgenden Befehl, um den ARN abzurufen.

```
$ aws elasticbeanstalk list-platform-versions
```

Verwenden Sie die --filters-Option zum Filtern der Leistung Ihrer benutzerdefinierten Plattform.