
Erste Schritte mit AWS

Hosten einer Webanwendung für Linux



Erste Schritte mit AWS: Hosten einer Webanwendung für Linux

Copyright © 2013 Amazon Web Services, Inc. or its affiliates. All rights reserved.

The following are trademarks or registered trademarks of Amazon: Amazon, Amazon.com, Amazon.com Design, Amazon DevPay, Amazon EC2, Amazon Web Services Design, AWS, CloudFront, EC2, Elastic Compute Cloud, Kindle, and Mechanical Turk. In addition, Amazon.com graphics, logos, page headers, button icons, scripts, and service names are trademarks, or trade dress of Amazon in the U.S. and/or other countries. Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon.

All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Übersicht	1
Wie unterstützt Sie AWS?	2
Webanwendungshosting-Architektur	4
Erste Schritte	7
Schritt 1: Registrieren Sie sich für den Service	8
Schritt 2: Installieren der Befehlszeilen-Tools	8
Schritt 3: Erstellen eines Elastic Load Balancers	9
Aktueller Stand	14
Schritt 4: Erstellen und Konfigurieren Ihrer Amazon EC2-Sicherheitsgruppe	15
Schritt 5: Erstellen eines Schlüsselpaars	16
Schritt 6: Starten von Amazon EC2-Instances mithilfe von Auto Scaling	17
Aktueller Stand	19
Schritt 7: Erstellen eines CloudWatch-Alarms	20
Aktueller Stand	24
Schritt 8: Hinzufügen von Amazon RDS	24
Erstellen einer DB-Sicherheitsgruppe	25
Zugriff autorisieren	26
Starten einer Instance	26
Schritt 9: Bereitstellen Ihrer Anwendung	31
Herstellen einer Verbindung mit einer Amazon EC2-Instance über einen Web-Browser mithilfe des MindTerm-SSH-Clients	32
Herstellen einer Verbindung mit einer Amazon EC2-Instance von Windows mithilfe von PuTTY	33
Herstellen einer Verbindung mit einer Amazon EC2-Instance auf einem Linux/UNIX-Computer mithilfe eines eigenständigen SSH-Clients	38
Konfigurieren der Amazon EC2-Instance	39
Schritt 10: Erstellen eines benutzerdefinierten Amazon-Computerabbilds (AMI)	47
Schritt 11: Starten neuer Umgebungen mit AWS CloudFormation	48
Erstellen einer AWS CloudFormation-Vorlage	48
Ändern einer CloudFormation-Vorlage	51
Erstellen eines AWS CloudFormation-Stapels	53
Schritt 12: Bereinigen	58
Beenden Sie Ihre Amazon EC2-Instances in Ihrer Auto Scaling-Gruppe	58
Kündigen der DB-Instance	60
Löschen eines CloudWatch-Alarms	61
Löschen des Elastic Load Balancers	61
Löschen eines Schlüsselpaars	62
Löschen einer Amazon EC2-Sicherheitsgruppe	62
Delete Your Custom AMI	63
Amazon Route 53	64
Amazon CloudFront	65
Preise	66
Amazon EC2-Kostenaufschlüsselung	66
Amazon RDS-Kostenaufschlüsselung	69
Summe aller Kosten	71
Verwandte Ressourcen	73
Dokumentverlauf	75

Übersicht

Abstract

Enthält eine Grafik einer Beispielarchitektur einer auf AWS gehosteten Webanwendung sowie eine exemplarische Vorgehensweise zur Bereitstellung Ihrer Webanwendung für Linux mit AWS.

Sollten Sie Hardware kaufen, um Ihre Website zu betreiben, stellen Sie möglicherweise fest, dass hochverfügbares und skalierbares Webhosting ein komplexes und teures Vorhaben sein kann. Ihre Website unterliegt wahrscheinlich Spitzenzeiten mit hohem Datenverkehr und erheblichen Fluktuationen in den Datenverkehrsmustern. Dies würde dann zu niedrigen Auslastungsraten Ihrer teuren Hardware führen und es fallen möglicherweise hohe Betriebskosten an, um diese meist inaktive Hardware zu warten. Amazon Web Services (AWS) bietet eine zuverlässige, skalierbare, sichere und hochleistungsfähige Infrastruktur, die die Bedürfnisse der anspruchsvollsten Webanwendungen erfüllt. Das elastische Infrastrukturmodell von AWS kann vergrößert und verkleinert werden und die IT-Kosten werden den Änderungen der Datenverkehrsmuster der Kunden in Echtzeit angepasst.

Dieses Handbuch hilft Ihnen bei der Verwendung von AWS zum Erstellen skalierbarer, stabiler Webanwendungen, die komplexe Anforderungen und Arbeitslasten bewältigen können. Wir betrachten eine Beispielarchitektur einer auf AWS gehosteten Webanwendung und führen Sie durch die Schritte zur Bereitstellung einer Drupal-Beispielanwendung unter Verwendung mehrerer zentraler AWS-Services und der folgenden bewährten Methoden. (Drupal ist ein quelloffenes Content Management System.) Sie können dieses Beispiel bei Bedarf an Ihre spezifischen Anforderungen anpassen. Am Ende dieser schrittweisen Anleitung sollten Sie die folgenden Aufgaben ausführen können:

- Registrieren bei AWS
- Starten von Drupal, Herstellen einer Verbindung mit Drupal, Schützen und Bereitstellen von Drupal auf einem Computer in der [Cloud](#)
- Erstellen einer benutzerdefinierten Vorlage eines Computers mit der erforderliche Hardware, Software und Konfiguration
- Einrichten eines Load Balancers zum Verteilen des Datenverkehrs über mehrere Computer in der Cloud
- Skalieren Ihres Bestands an Computern in der Cloud
- Überwachen der Fehlerfreiheit Ihrer Anwendung und Computer
- Erstellen einer Datenbank-Instance und deren Verwendung mit Drupal
- Erstellen einer Vorlage für die erstellten AWS-Ressourcen
- Bereinigen Ihrer AWS-Ressourcen

Wenn Sie tiefere Einblicke in bewährte Methoden für AWS und die verschiedenen von AWS gebotenen Optionen erhalten möchten, empfehlen wir, dass Sie *Webanwendungshosting: Bewährte Methoden* in den [AWS Cloud Computing-Whitepapers](#) lesen.

Wenn Sie eine schnellere und einfachere Methode zur Bereitstellung Ihrer Webanwendungen suchen, können Sie einen Anwendungsverwaltungs-Service verwenden. AWS-Anwendungsverwaltungs-Services helfen Ihnen bei der Nutzung anderer AWS-Services und Sie müssen sie nicht mehr einzeln getrennt und manuell verwalten:

- Mit [AWS Elastic Beanstalk](#) können Sie sich auf den Code konzentrieren, während der Service den Rest verwaltet.
- [AWS OpsWorks](#) bietet Ihnen die Flexibilität, Ihren eigenen Software-Stack zu definieren und eine Reihe von Anwendungen und Architekturen bereitzustellen, zu betreiben und zu automatisieren.

Weitere Informationen zur Bereitstellung und Ressourcenverwaltung auf AWS finden Sie unter [Bereitstellung und Verwaltung bei AWS](#).

Sollten Sie in diesem Handbuch nicht die gewünschten Informationen finden, schauen Sie in den folgenden Dokumenten nach:

- [Erste Schritte mit AWS](#) – Liefert Informationen über Amazon Web Services, einschließlich hilfreicher Links.
- [Getting Started with AWS Free Usage Tier](#) – Liefert Informationen über die ersten Schritte zur Verwendung des kostenlosen Nutzungskontingents.
- [Hosten von Websites auf Amazon S3](#) im *Entwicklerhandbuch für Amazon Simple Storage Service* – Bietet eine exemplarische Vorgehensweise zum raschen Bereitstellen einer statischen Website, die nicht die Ausführung einer Anwendung erfordert.
- [Erste Schritte mit AWS CloudFormation](#) im *AWS CloudFormation User Guide* – Erleichtert Ihnen den Einstieg in die Verwendung einer Blog-Beispielvorlage von WordPress in der AWS CloudFormation. Sie müssen dabei die Reihenfolge, in der die AWS-Services bereitgestellt werden, und die erforderlichen Einheiten, damit die Abhängigkeiten funktionieren, nicht selbst herausfinden.
- [Amazon Elastic Compute Cloud Getting Started Guide](#) – Bietet eine schrittweise Anleitung durch den Vorgang des Startens einer Amazon EC2-Linux-Instance und des Herstellens einer Verbindung mit dieser Instance. Informationen zur Konfiguration von Software für die Ausführung auf Instances (z. B. MySQL, Tomcat, Python) erhalten Sie unter [Amazon Machine Images \(AMI\)](#) in *Amazon Elastic Compute Cloud User Guide*.
- [Getting Started with AWS-Computing-Grundlagen für Linux](#) – Enthält eine Einführung in eine Reihe von wichtigen AWS-Services und -Komponenten und beschreibt, was diese Services genau sind, warum sie wichtig sind und wie sie verwendet werden. Das Handbuch bietet außerdem eine einfache Beispielarchitektur auf einer Linux-Plattform und umfasst eine schrittweise Anleitung für eine Bereitstellung mit dieser Architektur. Sie erfahren außerdem, wie Sie MySQL Server installieren und eine Datenbank auf einer EC2-Instance konfigurieren.

Wie unterstützt Sie AWS?

Wenn Sie für die Ausführung einer Webanwendung verantwortlich sind, sind Sie mit einer Reihe von Infrastruktur- und Architekturproblemen konfrontiert, für die AWS Ihnen einfache, nahtlose und kosteneffektive Lösungen bereitstellen kann. Dieser Abschnitt enthält eine Liste von Amazon Web Services und Komponenten. Es wird erklärt, was diese Services und Komponenten zur Bewältigung der Herausforderungen beitragen können, die in dieser Beispiellösung gestellt werden.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Wie unterstützt Sie AWS?

Herausforderungen	Amazon Web Services	Vorteile
Server müssen bereitgestellt werden, um Kapazitäten zu Spitzenzeiten zu verarbeiten. Zu anderen Zeiten werden die nicht genutzten Zyklen verschwendet.	<ul style="list-style-type: none"> • Amazon Elastic Compute Cloud (EC2) • Amazon Elastic Load Balancing • Auto Scaling • Amazon CloudWatch 	<ul style="list-style-type: none"> • Amazon EC2 führt den Webserver und die Anwendungsserver aus. • Elastic Load Balancing unterstützt Zustandsprüfungen auf Hosts, Verteilung des Datenverkehrs an Amazon EC2-Instances in mehreren Availability Zones sowie das dynamische Hinzufügen und Entfernen von Amazon EC2-Hosts aus der lastenverteilten Rotation. • Auto Scaling erstellt Kapazitätsgruppen von Servern, die bei Bedarf vergrößert oder verkleinert werden können. • Amazon CloudWatch meldet Metrikdaten für Amazon EC2-Instances und die erfassten Metriken werden von Auto Scaling verwendet.
Ein Netzwerk zur Bereitstellung von Inhalten (Content Delivery Network, CDN) ist erforderlich, das Datenübertragung mit hohen Geschwindigkeiten und geringer Latenz bietet, damit Endbenutzer keine unnötigen Wartezeiten erleben.	<ul style="list-style-type: none"> • Amazon CloudFront • Amazon Simple Storage Service (Amazon S3) 	<ul style="list-style-type: none"> • Amazon CloudFront beschleunigt das Laden oder Streamen von heruntergeladenen Inhalten durch Zwischenspeichern der Inhalte über einen lokalen Edge-Cache an einem Standort mit der niedrigsten Latenz. • Amazon S3 speichert Datensicherungen von der relationalen Datenbank, dem Web und Anwendungsservern und für die Amazon CloudFront-Verteilung.
Anwendungen erfordern möglicherweise eine Datenbank, ein Dateisystem oder Zugriff auf unformatierten Speicher auf Blockebene.	Amazon Elastic Block Store (Amazon EBS)	Amazon EBS bietet ein persistentes Dateisystem für Web- und Anwendungsserver.
Die Pflege einer Datenbank kann teuer und zeitaufwändig sein.	Amazon Relational Database Service (Amazon RDS)	Amazon RDS stellt kosteneffiziente und individuell anpassbare Kapazität zur Verfügung und übernimmt zeitraubende Verwaltungsaufgaben für Datenbanken.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Webanwendungshosting-Architektur

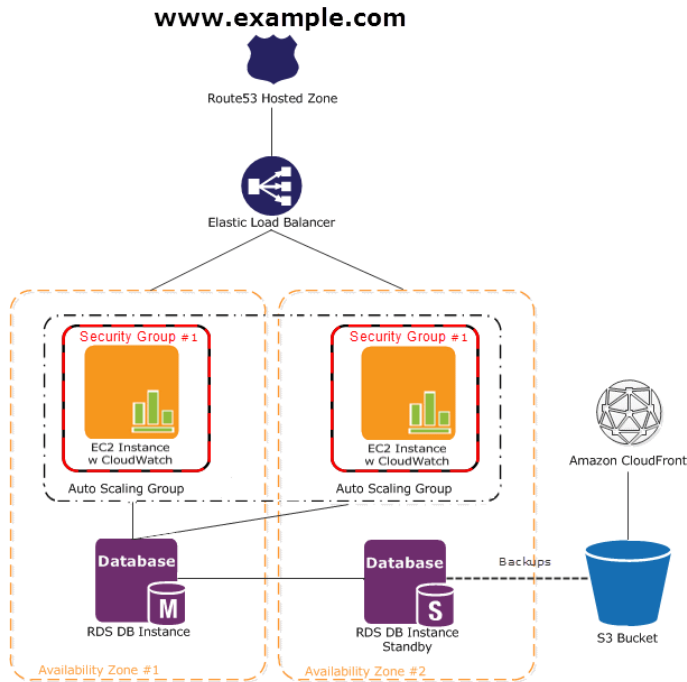
Herausforderungen	Amazon Web Services	Vorteile
Entwickler und Unternehmen brauchen eine zuverlässige und kostengünstige Methode, um Endbenutzer auf Internetanwendungen weiterzuleiten.	Amazon Route 53	Amazon Route 53 ist ein hochverfügbarer und skalierbarer Domain Name System (DNS)-Webservice. Entwickler und Unternehmen bietet es eine außerordentlich zuverlässige und kostengünstige Möglichkeit, Endbenutzer zu Internetanwendungen zu routen. Dazu werden menschenlesbare Namen wie www.beispiel.de in numerische IP-Adressen wie 192.0.2.1 übersetzt, die Computer zur gegenseitigen Vernetzung verwenden.
Die Reihenfolge, in der Amazon Web Services bereitgestellt werden, muss geplant werden, unter Berücksichtigung der Abhängigkeiten zwischen den Services.	AWS CloudFormation	AWS CloudFormation bietet Entwicklern und Systemadministratoren die Möglichkeit, ohne viel Aufwand eine Sammlung von zugehörigen AWS-Ressourcen zu erstellen und in geordneter und transparenter Form bereitzustellen.

Herausforderungen	AWS-Komponenten	Vorteile
Sicherheit muss gewährleistet werden, um Anwendungsserver vor böswilligen Benutzern von außen zu schützen.	Amazon-Sicherheitsgruppe	Eine Amazon-Sicherheitsgruppe erlaubt Ihnen, Protokolle, Ports und Quell-IP-Adressbereiche zu bestimmen, die berechtigt sind, auf Ihre Amazon EC2-Instances zuzugreifen.
Bei der Entwicklung muss Failover berücksichtigt werden.	Availability Zones	Availability Zones sind eigenständige Standorte, die so konzipiert wurden, dass Sie vor Fehlern in anderen Availability Zones geschützt sind. Jede Availability Zone bietet eine kostengünstige Netzwerkkonnektivität mit geringer Latenz zu anderen Availability Zones in der gleichen Region.

Webanwendungshosting-Architektur

Das folgende Diagramm zeigt eine Beispielarchitektur einer Webanwendung, die die im vorherigen Abschnitt erwähnten AWS-Ressourcen verwendet.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Webanwendungshosting-Architektur



In diesem Diagramm führen Amazon EC2-Instances die Anwendung und den Web-Server aus und gehören einer Amazon EC2-Sicherheitsgruppe an. Die Amazon EC2-Sicherheitsgruppe fungiert als äußere Firewall für die Amazon EC2-Instances. Eine Auto Scaling-Gruppe wird verwendet, um eine Flotte von Amazon Amazon EC2-Instances zu warten, die die auftretende Last abarbeiten kann. Diese Auto Scaling-Gruppe umfasst mehrere Availability Zones zum Schutz vor potenziellen Fehlern, falls eine Availability Zone ausfällt. Um den Datenverkehr gleichmäßig über die Amazon EC2-Instances zu verteilen, wird ein Elastic Load Balancer mit der Auto Scaling-Gruppe verknüpft. Wenn die Auto Scaling-Gruppe Instances basierend auf der Last startet oder beendet, passt sich der Elastic Load Balancer automatisch entsprechend an. Die Datenbankstufe besteht aus Amazon RDS-Datenbank-Instances, einschließlich Master und lokaler Slave, die in mehreren Availability Zones Schutz vor Failovers bieten. Amazon RDS bietet automatisierte Sicherungen auf Amazon S3. Amazon S3 speichert Sicherungen und statische Inhalte. Da die Benutzer dieser Anwendung weltweit verteilt sein könnten oder möglicherweise viele Benutzer die Website gleichzeitig aufrufen, werden große Volumes an statischen Inhalten mithilfe der Amazon CloudFront in einem Edge-Cache gespeichert, um eine bessere Leistung zu erzielen. Amazon Route 53 kann für eine sichere und zuverlässige Weiterleitung an Ihre Infrastruktur, die Amazon Web Services ausführt, verwendet werden.

Eine schrittweise Anleitung zum Erstellen dieser Architektur finden Sie unter [Erste Schritte \(p. 7\)](#). In dieser Anleitung erfahren Sie, wie Sie die folgenden Aufgaben ausführen:

- Registrieren bei AWS
- Starten von Drupal, Herstellen einer Verbindung mit Drupal, Schützen und Bereitstellen von Drupal auf einer Amazon EC2-Instance
- Erstellen eines benutzerdefinierten Amazon-Computerabbaus (AMI)
- Einrichten eines Elastic Load Balancers zum Verteilen des Datenverkehrs auf Ihre Amazon EC2-Instances
- Automatisches Skalieren Ihres Instance-Bestands mithilfe von Auto Scaling
- Überwachen Ihrer AWS-Ressourcen mithilfe von Amazon CloudWatch
- Erstellen einer Datenbank-Instance und deren Verwendung mit Drupal
- Erstellen einer AWS CloudFormation-Vorlage basierend auf den erstellten Ressourcen
- Bereinigen Ihrer AWS-Ressourcen

Weitere Informationen zur Verwendung von Amazon CloudFront in dieser Architektur erhalten Sie unter [Amazon CloudFront \(p. 65\)](#). Weitere Informationen zur Verwendung von Amazon Route 53 in dieser Architektur erhalten Sie unter [Amazon Route 53 \(p. 64\)](#).

Erste Schritte

Topics

- [Schritt 1: Registrieren Sie sich für den Service \(p. 8\)](#)
- [Schritt 2: Installieren der Befehlszeilen-Tools \(p. 8\)](#)
- [Schritt 3: Erstellen eines Elastic Load Balancers \(p. 9\)](#)
- [Schritt 4: Erstellen und Konfigurieren Ihrer Amazon EC2-Sicherheitsgruppe \(p. 15\)](#)
- [Schritt 5: Erstellen eines Schlüsselpaares \(p. 16\)](#)
- [Schritt 6: Starten von Amazon EC2-Instances mithilfe von Auto Scaling \(p. 17\)](#)
- [Schritt 7: Erstellen eines CloudWatch-Alarms \(p. 20\)](#)
- [Schritt 8: Hinzufügen von Amazon RDS \(p. 24\)](#)
- [Schritt 9: Bereitstellen Ihrer Anwendung \(p. 31\)](#)
- [Schritt 10: Erstellen eines benutzerdefinierten Amazon-Computerabbilds \(AMI\) \(p. 47\)](#)
- [Schritt 11: Starten neuer Umgebungen mit AWS CloudFormation \(p. 48\)](#)
- [Schritt 12: Bereinigen \(p. 58\)](#)

Nehmen wir an, Sie möchten eine CMS-Anwendung (Content Management System) erstellen. Sie möchten die zuverlässige, skalierbare, sichere und hochleistungsfähige Infrastruktur, die AWS bietet, optimal nutzen. Die ersten Schritte sind einfach und für die meisten Aufgaben können wir die [AWS Management Console](#) verwenden. In diesem Thema lernen Sie die Schritte kennen, die zum Bereitstellen Ihrer Webanwendung an AWS erforderlich sind. Es gibt viele verschiedene Arten der Bereitstellung für Webanwendungen, diese schrittweise Anleitung konzentriert sich jedoch auf ein Beispiel, das bewährte Methoden befolgt und viele der AWS-Services nutzt, sodass Sie sehen können, wie die Services zusammen funktionieren. Also starten wir!



Note

In diesem Beispiel gehen wir die Schritte in einer bestimmten Reihenfolge durch, um die Zeit für kostenpflichtige Services zu minimieren. Wenn Sie allerdings Ihre Anwendung bereitstellen, werden Sie wahrscheinlich damit beginnen, Ihre Amazon EC2-Instance zu starten, Ihre Anwendung und Datenbank zu konfigurieren, ein benutzerdefiniertes AMI zu erstellen und dann Ihre Anwendung zu skalieren.

Schritt 1: Registrieren Sie sich für den Service

Wenn Sie noch kein AWS-Konto haben, müssen Sie eines eröffnen. Mit Ihrem AWS-Konto haben Sie Zugriff auf alle Services, aber es werden Ihnen nur die Ressourcen in Rechnung gestellt, die Sie nutzen. Für dieses Beispiel werden die Gebühren minimal sein.

Um sich für AWS anzumelden:

1. Gehen Sie zu <http://aws.amazon.com/> und klicken Sie auf Anmelden.
2. Folgen Sie den Anweisungen auf dem Bildschirm.

Sie werden per E-Mail von AWS benachrichtigt, wenn Ihr Konto aktiv ist und verwendet werden kann.

Sie können über Ihr AWS-Konto Ressourcen innerhalb von AWS bereitstellen und verwalten. Wenn Sie anderen Personen den Zugriff auf Ihre Ressourcen erlauben, möchten Sie wahrscheinlich kontrollieren, wer Zugriff hat und was er tun kann. AWS Identity and Access Management (IAM) ist ein Webservice, der den Zugriff auf Ihre Ressourcen durch andere Leute steuert. In IAM erstellen Sie Benutzer, die andere Leute verwenden können, um Zugriff und Berechtigungen zu erhalten, die Sie definieren. Weitere Informationen zu IAM finden Sie unter [Verwendung von IAM](#).

Schritt 2: Installieren der Befehlszeilen-Tools

Wir müssen einige Befehlszeilen-Tools für Auto Scaling installieren. Hierdurch können Sie vor allem die Nutzung von gebührenpflichtigen Services minimieren.

Zum Installieren der Auto Scaling-Befehlszeilen-Tools auf Ihrem lokalen Computer rufen Sie [Using the Command Line Tools](#) in *Entwicklerhandbuch für Auto Scaling* auf. Nachdem Sie die Befehlszeilen-Tools installiert haben, probieren Sie einige Befehle aus, um sicherzustellen, dass sie funktionieren. Geben Sie beispielsweise an der Eingabeaufforderung den Befehl `as-cmd` ein.

```
PROMPT>as-cmd
```

Dieser Befehl gibt eine Liste aller Auto Scaling-Befehle mit ihren Beschreibungen zurück. Die Ausgabe sollte wie in der folgenden Abbildung oder ähnlich aussehen.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Schritt 3: Erstellen eines Elastic Load Balancers

Command Name	Description
as-create-auto-scaling-group	Create a new Auto Scaling group.
as-create-launch-config	Creates a new launch configuration.
as-create-or-update-tags	Create or update tags.
as-delete-auto-scaling-group	Deletes the specified Auto Scaling group.
as-delete-launch-config	Deletes the specified launch configuration.
as-delete-notification-configuration	Deletes the specified notification configuration.
as-delete-policy	Deletes the specified policy.
as-delete-scheduled-action	Deletes the specified scheduled action.
as-delete-tags	Delete the specified tags
as-describe-adjustment-types	Describes all policy adjustment types.
as-describe-auto-scaling-groups	Describes the specified Auto Scaling groups.
as-describe-auto-scaling-instances	Describes the specified Auto Scaling instances.
as-describe-auto-scaling-notification-types	Describes all Auto Scaling notification types.
as-describe-launch-configs	Describes the specified launch configurations.
as-describe-metric-collection-types	Describes all metric colle... metric granularity types.
as-describe-notification-configurations	Describes all notification...given Auto Scaling groups.
as-describe-policies	Describes the specified policies.
as-describe-process-types	Describes all Auto Scaling process types.
as-describe-scaling-activities	Describes a set of activit...ties belonging to a group.
as-describe-scheduled-actions	Describes the specified scheduled actions.
as-describe-tags	Describes tags
as-describe-termination-policy-types	Describes all Auto Scaling termination policy types.
as-disable-metrics-collection	Disables collection of Auto Scaling group metrics.
as-enable-metrics-collection	Enables collection of Auto Scaling group metrics.
as-execute-policy	Executes the specified policy.
as-put-notification-configuration	Creates or replaces notifi...or the Auto Scaling group.
as-put-scaling-policy	Creates or updates an Auto Scaling policy.
as-put-scheduled-update-group-action	Creates or updates a scheduled update group action.
as-resume-processes	Resumes all suspended Auto... given Auto Scaling group.
as-set-desired-capacity	Sets the desired capacity of the Auto Scaling group.
as-set-instance-health	Sets the health of the instance.
as-suspend-processes	Suspends all Auto Scaling ... given Auto Scaling group.
as-terminate-instance-in-auto-scaling-group	Terminates a given instance.
as-update-auto-scaling-group	Updates the specified Auto Scaling group.
help	
version	Prints the version of the CLI tool and the API.

For help on a specific command, type '<commandname> --help'

user ~ %_

Nachdem Sie die Befehlszeilen-Tools installiert haben, können Sie mit der Erstellung Ihrer AWS-Ressourcen beginnen. Sie können nun über den Start Ihrer Amazon EC2-Instances nachdenken. Obwohl Sie für die Zwecke dieses Tutorials nur eine Amazon EC2-Instance ausführen, möchten Sie eventuell mehrere Amazon EC2-Instances über mehrere Availability Zones ausführen. Auf diese Weise kann der Datenverkehr, falls eine Availability Zone ausfällt, an eine andere Availability Zone umgeleitet werden. Um uns auf die Eventualität mehrerer Amazon EC2-Instances vorzubereiten, werden wir unsere Elastic Load Balancer-Ressource erstellen. Im Schritt "AWS CloudFormation" können wir vertikal skalieren, um unseren Elastic Load Balancer zu nutzen. Fahren wir mit dem nächsten Schritt fort und erstellen wir unseren [Elastic Load Balancer](#).

Schritt 3: Erstellen eines Elastic Load Balancers

Elastic Load Balancing is a cost-effective and easy-to-use web service to help you improve the availability and scalability of your application. It makes it easy for you to distribute application loads between two or more Amazon EC2 instances. Elastic Load Balancing enables availability through redundancy and supports traffic growth of your application.

Elastic Load Balancing lets you automatically distribute and balance the incoming application traffic among all the instances you are running. The service also makes it easy to add new instances when you need to increase the capacity of your application. You can dynamically register or deregister instances from the load balancer as the capacity requirements of your application change with time.

As soon as your load balancer becomes available, you're billed for each hour or partial hour that you keep the load balancer running. For more information about Elastic Load Balancing, see the [Elastic Load Balancing](#) details page.

In this step, we will create a load balancer for an HTTP service. We'll specify that the load balancer listens on port 80 and distributes traffic to port 80 on the instances.



Note

We'll go ahead and create our Elastic Load Balancer resource so that in the future when you have multiple instances running, your traffic will be load balanced between your instances. Elastic Load Balancing is a small cost relative to instance hours. In [Schritt 11: Starten neuer Umgebungen mit AWS CloudFormation \(p. 48\)](#) we'll use AWS CloudFormation to create a template for our resources and add instances to our Auto Scaling Group.

For more information about elastic load balancers, go to the [Elastic Load Balancing Documentation](#).

Erstellen Sie einen Load Balancer wie folgt:

1. Definieren Sie einen Load Balancer.
 - a. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
 - b. Wählen Sie in der oberen Navigationsleiste US East (N. Virginia) aus der Regionenauswahl.
 - c. Klicken Sie im linken Navigationsbereich auf Load Balancers.
 - d. Klicken Sie auf Create Load Balancer.
 - e. Geben Sie im Assistenten Create a New Load Balancer auf der Seite Define Load Balancer einen Namen für den Load Balancer ein. Geben Sie für dieses Beispiel **MyLB** ein.

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	80

- f. Belassen Sie die Einstellung von Create LB Inside auf EC2-Classical.
 - g. Belassen Sie die Werte der Listener Configuration auf der Standardeinstellung.
 - h. Klicken Sie auf Weiter.
2. Konfigurieren Sie die Zustandsprüfung für Ihren Load Balancer. Der Elastic Load Balancing überprüft regelmäßig den Zustand der jeweiligen lastverteilten Amazon EC2-Instance. Wenn Elastic Load Balancing eine fehlerhafte Instance findet, wird kein Verkehr mehr an diese Instance gesendet und der Verkehr wird an eine fehlerfreie Instance umgeleitet.

- a. Belassen Sie in diesem Beispiel den Wert für Ping Protocol auf der Standardeinstellung HTTP. Wenn Sie Ihre Anwendung zu einem späteren Zeitpunkt bereitstellen, können Sie HTTPS angeben. Weitere Information zur Verwendung von HTTPS mit Elastic Load Balancing finden Sie unter [Elastic Load Balancing Security Features](#) in *Entwicklerhandbuch für Elastic Load Balancing*.
- b. Belassen Sie in diesem Beispiel den Wert für Ping Port auf der Standardeinstellung 80.

Elastic Load Balancing verwendet den Ping Port zum Senden von Zustandsprüfungsabfragen an Ihre Amazon EC2-Instances.



Note

Wenn Sie einen Portwert festlegen, müssen Ihre Amazon EC2-Instances den eingehenden Verkehr an dem Port, den Sie für die Zustandsprüfung angegeben haben, akzeptieren. Sie können einen anderen Portwert als 80 einstellen und Sie können diesen Wert auch zu einem späteren Zeitpunkt festlegen. Stellen Sie ihn aber für dieses Beispiel auf 80 ein.

- c. Ersetzen Sie den Standardwert im Feld Ping Path durch einen Schrägstrich ("/").

Elastic Load Balancing sendet Zustandsprüfungsabfragen an den Pfad, den Sie in Ping Path angeben. Dieses Beispiel verwendet einen Schrägstrich, damit Elastic Load Balancing die Anforderung an die Standardstartseite des HTTP-Servers sendet, unabhängig davon, ob die Standardseite den Namen `index.html`, `default.html` oder einen anderen Namen hat. Erwägen Sie beim Bereitstellen Ihrer Anwendung, eine spezielle einfache Datei zu erstellen, die nur auf die Zustandsprüfung reagiert. Hierdurch kann zwischen Datenverkehr, der auf Ihrer Website eingeht, und Reaktionen auf den Load Balancer unterschieden werden.

- d. Stellen Sie den Healthy Threshold auf 2. Belassen Sie die übrigen Advanced Details auf ihren Standardwerten.

Create Load Balancer

1. Define Load Balancer **2. Configure Health Check** 3. Add EC2 Instances 4. Review

Configure Health Check

Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to healthy instances. If an instance fails the health check, it is automatically removed from the load balancer. Configure health checks based on your specific needs.

Ping Protocol

Ping Port

Ping Path

Advanced Details

Response Timeout ⓘ	<input type="text" value="5"/>	seconds
Health Check Interval ⓘ	<input type="text" value="30"/>	seconds
Unhealthy Threshold ⓘ	<input type="text" value="2"/>	
Healthy Threshold ⓘ	<input type="text" value="2"/>	

e. Klicken Sie auf Weiter.

3. Klicken Sie auf der Seite Add Instances to Load Balancer auf Continue.
4. Überprüfen Sie die Einstellungen. Sie können die Einstellungen ändern, indem Sie auf den Link Edit für einen bestimmten Schritt klicken.

Create Load Balancer [Close]

1. Define Load Balancer 2. Configure Health Check 3. Add EC2 Instances **4. Review**

Review

Please review the load balancer details before continuing

▼ **Define Load Balancer** [Edit load balancer definition](#)

Load Balancer name: MyLB
Scheme: internet-facing
Port Configuration: 80 (HTTP) forwarding to 80 (HTTP)

▼ **Configure Health Check** [Edit health check](#)

Ping Target: HTTP:80/
Timeout: 5 seconds
Interval: 30 seconds
Unhealthy Threshold: 2
Healthy Threshold: 2

▼ **Add EC2 Instances** [Edit instances](#)

Cross-Zone Load Balancing: Enabled
Connection Draining: Enabled, 300 seconds
Instances:

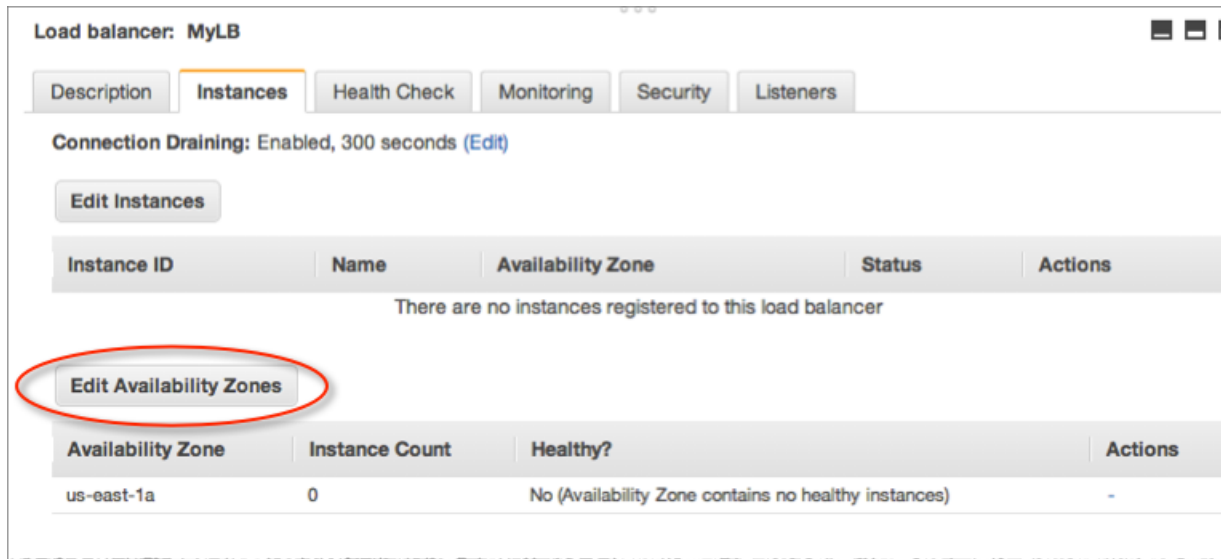
[Back](#) [Create](#)

5. Klicken Sie auf Create. Klicken Sie auf der Bestätigungsseite Create Load Balancer auf Close.

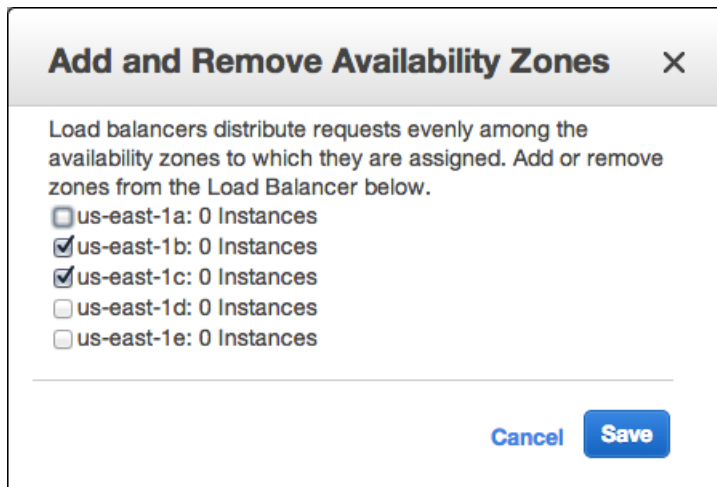
Der neue Load Balancer wird nun in der Liste angezeigt.

Als bewährte Methode sollten genügend Instances in den Availability Zones vorhanden sein, um den Ausfall einer Availability Zone zu überbrücken. Daher sollten Sie im nächsten Schritt sicherstellen, dass der Load Balancer auf mehrere Availability Zones zeigt.

6. Hinzufügen einer Availability Zone.
 - a. Klicken Sie in der Liste Load Balancers auf MyLB.
 - b. Klicken Sie auf die Registerkarte Instances.
 - c. Klicken Sie auf Edit Availability Zones.



- d. Führen Sie im Dialogfeld Add and Remove Availability Zones folgende Schritte aus:
- Klicken Sie auf us-east-1b: 0 instances.
 - Klicken Sie auf us-east-1c: 0 instances.
 - Klicken Sie auf Speichern.



Die Spalte Availability Zones des Load Balancers zeigt nun die Availability Zones an, die Sie ausgewählt haben.

Aktueller Stand

Zurzeit befinden Sie sich bei der Erstellung Ihrer Architektur an der folgenden Position.



Fahren wir mit dem nächsten Thema fort und erstellen wir Ihre Amazon EC2-Sicherheitsgruppe. Sie müssen eine Amazon EC2-Sicherheitsgruppe erstellen, um auf Ihrer Instance Ports öffnen zu können. Ihre Sicherheitsgruppe funktioniert im Wesentlichen wie eine Firewall.

Schritt 4: Erstellen und Konfigurieren Ihrer Amazon EC2-Sicherheitsgruppe

Abstract

Erstellen und konfigurieren Sie Ihre Amazon EC2-Sicherheitsgruppe, um den Zugriff für eine Gruppe von Instances zu steuern.

Eine Amazon EC2-Sicherheitsgruppe agiert als Firewall, die den zulässigen Datenverkehr steuert, der in eine Instance-Gruppe eingeht. Wenn Sie eine Amazon EC2-Instance starten, können Sie diese einer oder mehreren Sicherheitsgruppen zuweisen. Für jede Sicherheitsgruppe fügen Sie Regeln hinzu, die den zulässigen eingehenden Datenverkehr zu Instances in der Gruppe steuern. Anderer eingehender Datenverkehr wird abgewiesen. Sie können die Regeln für eine Sicherheitsgruppe jederzeit ändern. Die neuen Regeln werden automatisch für alle vorhandenen und künftigen Instances in der Gruppe durchgesetzt.

In diesem Schritt werden wir Folgendes tun:

- Eine Amazon EC2-Sicherheitsgruppe erstellen
- Eine Amazon EC2-Sicherheitsgruppe konfigurieren

Erstellen und konfigurieren Sie Ihre Sicherheitsgruppe wie folgt:

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Stellen Sie sicher, dass US East (N. Virginia) in der Regionenauswahl in der Navigationsleiste ausgewählt ist.



Note

Für die Zwecke dieser Anleitung werden wir immer die Region US East (N. Virginia) verwenden. Wenn Sie allerdings Ihre Anwendung bereitstellen, wählen Sie die Region, die Ihrem Standort am nächsten liegt.

3. Klicken Sie im linken Navigationsbereich auf Load Balancers.
4. Wählen Sie den Load Balancer aus, den Sie zuvor erstellt haben, und klicken Sie auf die Registerkarte Security. Kopieren oder schreiben Sie den Namen der Sicherheitsgruppe, die zum Load Balancer gehört, in das Feld Source Security Group. Sie benötigen diesen Namen, um die Regeln für die Sicherheitsgruppe Ihrer Instance zu konfigurieren.
5. Klicken Sie im Navigationsbereich auf Security Groups und dann auf Create Security Group.
6. Geben Sie im Dialogfeld Create Security Group `webappsecuritygroup` in das Feld Security group name ein und geben Sie eine Beschreibung Ihrer Wahl in das Feld Description ein.
7. Klicken Sie auf der Registerkarte Inbound auf Add Rule und wählen Sie HTTP aus der Liste Type aus.
8. Wählen Sie Custom IP im Feld Source aus und geben Sie den Namen der Sicherheitsgruppe ein, die zum Load Balancer gehört, beispielsweise `amazon-e1b/amazon-e1b-sg`. Wenn Sie diese Quelle auswählen, kann sich nur der Datenverkehr, der über den Elastic Load Balancer eingeht, mit Ihrer Amazon EC2-Instance verbinden.

- Klicken Sie auf Add Rule.
- Wählen Sie SSH aus der Liste Type, um sich mit Ihren Amazon EC2-Instances zu verbinden. Wählen Sie Anywhere aus der Liste "Source" aus.

Type	Protocol	Port Range	Source
HTTP	TCP	80	Custom IP : amazon-elb/amazon-
SSH	TCP	22	Anywhere : 0.0.0.0/0



Important

Die Sicherheitsgruppeneinstellungen sind so konfiguriert, dass der Zugriff von überall aus möglich ist: 0.0.0.0/0. Das ist kein bewährtes Verfahren und wir wählen diese Einstellung nur für die Zwecke dieser Übung. Die bewährte Methode wäre, Regeln einzurichten, die den Zugriff nur auf die Computer oder Netzwerke beschränken, die den Zugriff auf diesen Dienst erfordern. Die Zahl nach dem "/" gibt einen Adressbereich an.

- Klicken Sie auf Create.

Ihre Amazon EC2-Sicherheitsgruppe wird noch nicht durchgesetzt. Wir setzen die Gruppe durch, wenn wir die Auto Scaling-Gruppe erstellen. Sie können allerdings auch eine Amazon EC2-Sicherheitsgruppe auf eine Amazon EC2-Instance anwenden. Weitere Informationen erhalten Sie unter [Using Security Groups](#) in *Amazon Elastic Compute Cloud User Guide*.

Nachdem wir jetzt unsere Amazon EC2-Sicherheitsgruppe erstellt haben, benötigen wir eine Möglichkeit, auf unsere Amazon EC2-Instance zuzugreifen, um unsere Anwendung bereitzustellen. Öffentliche AMI Instances verwenden ein Schlüsselpaar aus einem privaten und einem öffentlichen Schlüssel anstatt eines Passworts für die Anmeldung. Im nächsten Abschnitt werden wir unser Schlüsselpaar erstellen.

Schritt 5: Erstellen eines Schlüsselpaares

Sie können ein Schlüsselpaar erstellen, um die Verbindung mit Ihren Amazon EC2-Instances herzustellen. Öffentliche AMI Instances verwenden ein Schlüsselpaar aus einem privaten und einem öffentlichen Schlüssel anstatt eines Passworts für die Anmeldung. Die öffentliche Hälfte des Schlüsselpaares ist in Ihrer Instance eingebettet, damit Sie den privaten Schlüssel für die sichere Anmeldung ohne Passwort verwenden können. Nachdem Sie Ihre AMIs erstellt haben, können Sie andere Methoden auswählen,

um sich sicher an Ihren neuen Instances anzumelden. In diesem Schritt werden wir ein Schlüsselpaar mithilfe der AWS Management Console erstellen.

Generieren Sie ein Schlüsselpaar wie folgt:

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie in der oberen Navigationsleiste in der Regionenauswahl auf US East (N. Virginia).
3. Klicken Sie im linken Navigationsbereich unter Network and Security auf Key Pairs.
4. Klicken Sie auf Create Key Pair.
5. Geben Sie `mykeypair` in das neue Feld Key Pair Name ein und klicken Sie anschließend auf Yes.
6. Laden Sie die Datei mit dem privaten Schlüssel `mykeypair.pem` herunter und speichern Sie sie an einem sicheren Ort. Sie müssen auf alle Instances, die Sie starten, mit diesem Schlüsselpaar zugreifen.



Important

Wenn Sie das Schlüsselpaar verlieren, können Sie nicht auf Ihre Amazon EC2-Instances zugreifen.

Weitere Informationen über Schlüsselpaare finden Sie unter [Amazon EC2 Key Pairs](#) in *Amazon Elastic Compute Cloud User Guide*.

Da Ihr Datenverkehr variieren kann, sollte AWS die Anzahl der Instances entsprechend skalieren können. Hierzu sollten Sie [Auto Scaling](#) verwenden, um eine Auto Scaling-Gruppe zu erstellen. Fahren wir nun mit dem nächsten Schritt fort, um unsere Auto Scaling-Gruppe zu erstellen und sie mit unserem Elastic Load Balancer zu verbinden.

Schritt 6: Starten von Amazon EC2-Instances mithilfe von Auto Scaling

Auto Scaling wurde entwickelt, um Amazon EC2-Instances basierend auf benutzerdefinierten Richtlinien, Zeitplänen und Alarmen automatisch zu starten oder zu beenden. Sie können Auto Scaling verwenden, um eine Flotte von Amazon Amazon EC2-Instances zu warten, die jede auftretende Last abarbeiten können. Wie der Name schon sagt, reagiert Auto Scaling automatisch auf sich ändernde Bedingungen. Sie müssen nur festlegen, wie auf diese Änderungen reagiert werden soll. Sie können beispielsweise Auto Scaling die Anweisung erteilen, zusätzliche Instances zu starten, wenn die CPU-Auslastung zehn Minuten lang über 60 Prozent liegt, oder Sie können Auto Scaling konfigurieren, die Hälfte der Instances Ihrer Website über das Wochenende zu beenden, wenn niedriger Datenverkehr zu erwarten ist. Sie können mithilfe von Auto Scaling außerdem sicherstellen, dass die Instances in Ihrer Flotte optimal funktionieren, damit Ihre Anwendungen weiterhin effizient ausgeführt werden. Auto Scaling-Gruppen funktionieren sogar über mehrere Availability Zones hinweg – das sind unterschiedliche physische Standorte für die gehosteten Amazon EC2-Instances. Wenn eine Availability Zone ausfällt, verteilt Auto Scaling die Anwendungen automatisch an eine andere Availability Zone. Auto Scaling gewährleistet, dass immer mindestens eine funktionierende Instance ausgeführt wird. Weitere Informationen finden Sie im Abschnitt [Auto Scaling](#).

In diesem Beispiel konfigurieren wir die Basisinfrastruktur, die erforderlich ist, um Auto Scaling für die meisten Anwendungen zu starten. Wir richten eine Amazon EC2-Anwendung ein, die mit mindestens einer Instance und mit maximal einer Instance lastenverteilt und automatisch skaliert wird, damit nur für eine Instance Kosten verrechnet werden. Wenn Sie die Website erstellen, sollten Sie jedoch das bewährte Verfahren einhalten und in den Availability Zones genügend Instances haben, um den Verlust einer Availability Zone zu überstehen. Erhöhen Sie außerdem die maximale Anzahl der Instances, die größer als die Mindestanzahl sind, um die Auto Scaling-Funktion zu verwenden. Sie können auch die maximale

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Schritt 6: Starten von Amazon EC2-Instances mithilfe von
Auto Scaling

Anzahl von Instances angeben, um die Größe Ihrer Flotte zu steuern. In diesem Beispiel wird Auto Scaling so konfiguriert, dass bei Änderung der Kapazität eine Instance hinzugefügt wird. In diesem Thema legen wir die Richtlinie fest und erstellen dann im nächsten Abschnitt einen CloudWatch-Alarm, der die Richtlinie anwendet, wenn die durchschnittliche CPU-Auslastung zehn Minuten lang einen Grenzwert von 60 Prozent überschreitet. Auto Scaling und Amazon CloudWatch arbeiten zusammen, um Instances basierend auf der von Ihnen erstellten Richtlinie zu starten oder abzubrechen. Um Zeit zu sparen, erstellen wir nur eine Richtlinie. Sie können jedoch weitere Richtlinien erstellen, beispielsweise eine Richtlinie zum Verringern der Anzahl von Instances.

Wenn Sie die Auto Scaling-Befehlszeilen-Tools nicht bereits installiert haben, führen Sie dies wie im Abschnitt zur [Verwendung der Befehlszeilen-Tools](#) im *Auto Scaling Developer Guide* beschrieben durch. Wir richten das Auto Scaling mithilfe von Befehlszeilen-Tools ein.

Konfigurieren Sie eine automatisch skalierte, lastenverteilte Amazon EC2-Anwendung wie folgt:

1. Öffnen Sie ein Befehlszeilenfenster. Starten Sie in Microsoft Windows die Befehlszeilenanwendung (klicken Sie im Menü Start auf Programs, klicken Sie auf Accessories und dann auf Command Prompt).
2. Verwenden Sie den Auto Scaling-Befehl `as-create-launch-config`. In diesem Beispiel wird ein öffentlich verfügbares Linux AMI verwendet, auf dem ein Content Management System (CMS) ausgeführt wird. Wir verwenden einen `t1.micro`-Instance-Typ und die Sicherheitsgruppe und das Schlüsselpaar, die wir in den vorherigen Schritten erstellt haben. In diesem Beispiel befindet sich die Schlüsselpaardatei in dem Verzeichnis, in dem wir unsere Auto Scaling-Gruppe erstellen. Eine Region wird nicht festgelegt, weil die Standardregion verwendet werden soll.



Note

Für das Starten einer Amazon EC2-Instance entstehen Kosten. Die Gebühren in diesem Beispiel sind minimal. Weitere Informationen über Amazon EC2-Preise finden Sie auf der Seite [Amazon Elastic Compute Cloud \(Amazon EC2\)](#).

```
PROMPT>as-create-launch-config MyLC --image-id ami-7813e011 --instance-type  
t1.micro --group webappsecuritygroup --key mykeypair
```

Die Ausgabe von Auto Scaling ähnelt der Ausgabe im folgenden Beispiel:

```
OK-Created launch config
```



Note

Sie können die Befehle aus dem Dokument kopieren und in das Befehlszeilenfenster einfügen. Verwenden Sie die rechte Maustaste, um die Inhalte in das Befehlszeilenfenster einzufügen. Sollten die Befehle nicht ordnungsgemäß ausgeführt werden, sind sie wahrscheinlich nicht korrekt eingefügt worden.

3. Verwenden Sie den Auto Scaling-Befehl `as-create-auto-scaling-group`. In diesem Beispiel legen wir zwei Availability Zones fest. Dies ist ein bewährtes Verfahren zum Erstellen von fehlertoleranten Anwendungen. Fällt eine Availability Zone aus, wird der Datenverkehr in eine andere Availability Zone umgeleitet. Die Anzahl der Instances, die in der Auto Scaling-Gruppe gestartet werden, wird gleichmäßig auf die Availability Zones verteilt.

```
PROMPT>as-create-auto-scaling-group MyAutoScalingGroup --launch-configuration  
MyLC --availability-zones us-east-1b, us-east-1c --min-size 1 --max-size  
1 --load-balancers MyLB
```

Auto Scaling gibt Folgendes zurück:

```
OK-Created AutoScalingGroup
```

4. Verwenden Sie den Auto Scaling Befehl `as-put-scaling-policy` zum Erstellen einer Richtlinie, die Ihre Instances-Flotte vergrößert.

```
PROMPT>as-put-scaling-policy MyScaleUpPolicy --auto-scaling-group MyAutoScalingGroup --adjustment=1 --type ChangeInCapacity --cooldown 300
```

Die Ausgabe von Auto Scaling ähnelt der Ausgabe im folgenden Beispiel:

```
POLICY-ARN arn:aws:autoscaling:us-east-1:012345678901:scalingPolicy:cbe7da4e-5d00-4882-900a-2f8113431e30:autoScalingGroupName/MyAutoScalingGroup:policyName/MyScaleUpPolicy
```



Note

Um Zeit zu sparen, haben wir nur eine Richtlinie zum Vergrößern der Anzahl von Instances erstellt. Normalerweise würden Sie auch eine Richtlinie zum Verringern der Anzahl von Instances erstellen. Auto Scaling verringert die Anzahl von Instances, wenn Ihre Anwendung die Ressourcen nicht benötigt, sodass Sie Geld sparen. Um eine Richtlinie zum Verringern der Instances zu erstellen, ändern Sie den Richtliniennamen und die Anpassung von 1 auf -1.

5. Überprüfen Sie mit dem Befehl `as-describe-auto-scaling-groups`, ob Ihre Auto Scaling-Gruppe vorhanden ist.

```
PROMPT>as-describe-auto-scaling-groups MyAutoScalingGroup --headers
```

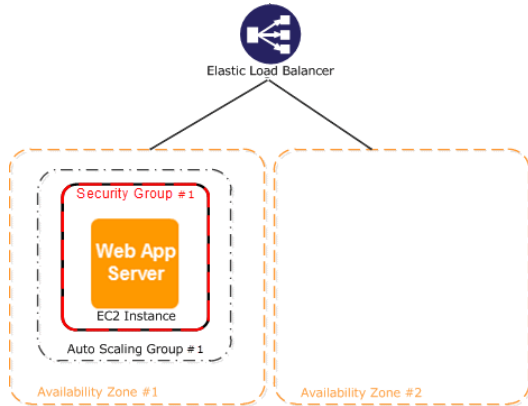
Auto Scaling gibt Folgendes zurück:

```
AUTO-SCALING-GROUP  GROUP-NAME          LAUNCH-CONFIG  AVAILABILITY-ZONES
  MIN-SIZE  MAX-SIZE  DESIRED-CAPACITY
AUTO-SCALING-GROUP  MyAutoScalingGroup  MyLC           us-east-1b,us-east-
1c  1          1          1
INSTANCE  INSTANCE-ID  AVAILABILITY-ZONE  STATE      STATUS  LAUNCH-CONFIG
INSTANCE  i-xxxxxxx   us-east-1c        InService  Healthy  MyLC
```

Ihre Amazon EC2-Anwendung wurde als automatisch skalierte, lastenverteilte Anwendung gestartet. Weitere Informationen zu Auto Scaling finden Sie in der [Auto Scaling Documentation](#). Solange die Amazon EC2-Instances ausgeführt werden, entstehen Ihnen weitere Kosten. Wenn Sie diese Instances beenden möchten, finden Sie weitere Informationen unter [Beenden Sie Ihre Amazon EC2-Instances in Ihrer Auto Scaling-Gruppe \(p. 58\)](#).

Aktueller Stand

Zurzeit befinden Sie sich bei der Erstellung Ihrer Architektur an der folgenden Position.



Nachdem Sie nun Ihre Auto Scaling-Gruppe erstellt haben und Ihre Amazon EC2-Instance reibungslos funktioniert, möchten Sie den Status Ihrer Instance überwachen. Im nächsten Schritt erstellen wir einen Amazon CloudWatch-Alarm, damit wir die soeben erstellte Auto Scaling-Richtlinie nachverfolgen können.

Schritt 7: Erstellen eines CloudWatch-Alarms

Amazon Cloudwatch ist ein Web-Service, mit dem Sie verschiedene Metriken überwachen, verwalten und veröffentlichen und Alarmaktionen anhand der Daten von Metriken konfigurieren können.

Amazon CloudWatch 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 AWS-Ressourcen – 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 Alarme erstellen, die in Ihrem Namen Aktionen des Auto Scaling-Services und des Amazon Simple Notification Service (Amazon SNS) starten.

Amazon CloudWatch wird normalerweise dazu eingesetzt, die Leistungsfähigkeit und Effizienz Ihrer Anwendungen und Services zu wahren. Sie können damit beispielsweise herausfinden, wann Ihre Website am besten läuft. Dies kann dann der Fall sein, wenn der Netzwerkdatenverkehr in Ihren Amazon EC2-Instances unter einem bestimmten Grenzwert bleibt. Sie können dann ein automatisiertes Verfahren erstellen, um sicherzugehen, dass immer die richtige Anzahl von Instances für den aktuellen Datenverkehr vorhanden ist. Sie können Amazon CloudWatch auch zum Diagnostizieren von Problemen verwenden, indem Sie die Systemleistung vor und nach dem Auftreten eines Problems analysieren. Amazon CloudWatch unterstützt Sie durch Erfassen der Leistung in Echtzeit bei der Ermittlung der Fehlerursache und bei der Überprüfung der Fehlerbehebung. Sie können Amazon CloudWatch beispielsweise so einrichten, dass Sie eine E-Mail erhalten, sobald Ihre Anwendung langsamer wird. Danach wird ermittelt, welche Datenbank überlastet war, und später wird die Beschleunigung der Reaktionszeiten überwacht. Weitere Informationen über die Erstellung von CloudWatch-Alarmen erhalten Sie unter [Erstellen von CloudWatch-Alarmen](#) im *Amazon CloudWatch Developer Guide*.

In der vorherigen Aufgabe haben wir eine Auto Scaling-Richtlinie erstellt, um die Anzahl der Instances zu vergrößern. In dieser Aufgabe müssen Sie die Auto Scaling-Richtlinie mit einer Alarmaktion verknüpfen, um Änderungen an Ihren Ressourcen durchzuführen. In diesem Thema werden Sie schrittweise durch die Erstellung eines CloudWatch-Alarms geführt, der die Anwendung warnt, wenn der Grenzwert überschritten wird. Um in dieser Übung Zeit zu sparen, erstellen wir nur einen Alarm. Sie können dieselbe Vorgehensweise jedoch auch zur Erstellung weiterer Alarme anwenden. Sie könnten beispielsweise

einen weiteren Alarm zum Verringern der Anzahl Ihrer Instances erstellen. Weitere Informationen über Amazon CloudWatch finden Sie auf der Detailseite von [Amazon CloudWatch](#).

Erstellen Sie einen Amazon CloudWatch-Alarm wie folgt:

1. Wählen Sie eine Metrik für Ihren Alarm.
 - a. Öffnen Sie die Amazon CloudWatch-Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
 - b. Stellen Sie sicher, dass US East (N. Virginia) in der Regionenauswahl in der oberen Navigationsleiste ausgewählt ist.
 - c. Wählen Sie im Navigationsbereich Alarm unter Alarms aus.
 - d. Klicken Sie auf Create Alarm.
 - e. Wählen Sie im Assistenten Create Alarm Wizard auf der Seite Select Metric aus dem Dropdown-Menü Viewing die Option EC2: Aggregated by Auto Scaling Group aus.

The screenshot shows the 'Create Alarm Wizard' in the AWS CloudWatch console. The wizard is in the 'SELECT METRIC' step. The 'Viewing' dropdown is set to 'EC2: Aggregated by Auto Scaling Group'. A list of metrics is displayed, with 'CPUUtilization' selected. A graph shows the CPU utilization over time. The 'Continue' button is visible at the bottom.

Viewing:	Search	1 to 28 of 28 Metrics
myAutoScalingGroup		DiskReadOps
MyAutoScalingGroup3		DiskWriteBytes
MyAutoScalingGroup3		DiskWriteOps
MyAutoScalingGroup3		NetworkIn
MyAutoScalingGroup3		NetworkOut
MyAutoScalingGroup		CPUUtilization
MyAutoScalingGroup		DiskReadBytes
MyAutoScalingGroup		DiskReadOps
MyAutoScalingGroup		DiskWriteBytes
MyAutoScalingGroup		DiskWriteOps
MyAutoScalingGroup		NetworkIn
MyAutoScalingGroup		NetworkOut

- f. Klicken Sie auf die Zeile MyAutoScalingGroup/CPU Utilization.
 - g. Klicken Sie auf Weiter.
2. Definieren Sie den Alarm.
 - a. Geben Sie auf der Seite Define Alarm des Assistenten Create Alarm den Namen **MyHighCPU-Alarm** in das Feld Name ein.
 - b. Geben Sie im Feld Description eine Beschreibung ein.
 - c. Wählen Sie im Abschnitt Define Alarm Threshold die Option > aus und geben Sie dann für dieses Beispiel im ersten Feld 60 und im Minutenfeld 10 ein. Sie können für Ihre Anwendung einige Lasttests ausführen, um festzustellen, welcher Wert für Ihre Anwendung am sinnvollsten ist.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Schritt 7: Erstellen eines CloudWatch-Alarms

Create Alarm Wizard Cancel

SELECT METRIC **DEFINE ALARM** CONFIGURE ACTIONS REVIEW

Provide the details and threshold for your alarm. Use the graph below to help set the appropriate threshold.

Identify Your Alarm
Assign your alarm a name and description.

Name:

Description:

Define Alarm Threshold
Alarms have three states: ALARM, OK, and INSUFFICIENT DATA. The state of your alarm changes according to a threshold you specify. First, define the criterion for entering the ALARM state. Later, you can specify an action to be taken when your alarm enters any of the three states.

This alarm will enter the ALARM state when CPUUtilization is > for minutes.

CPUUtilization (Percent)

Time	CPU Utilization (%)
9/22 19:00	~50
9/22 20:00	~50
9/22 21:00	~50
9/22 22:00	~50
9/22 23:00	~50
9/23 00:00	~75

< Back Continue

- d. Klicken Sie auf Weiter.
3. Definieren Sie Ihre Aktionen.
 - a. Wählen Sie auf der Seite Configure Actions des Assistenten Create Alarm die Option Alarm aus dem Dropdown-Menü When Alarm state is aus.
 - b. Wählen Sie Auto Scaling Policy im Dropdown-Menü Take Action aus.
 - c. Wählen Sie MyAutoScalingGroup im Dropdown-Menü Auto Scaling Group aus.
 - d. Wählen Sie MyScaleUpPolicy (Add 1 instance) im Dropdown-Menü Policy aus.
 - e. Klicken Sie auf Add Action.
 - f. Wählen Sie Alarm im Dropdown-Menü When Alarm state is aus.
 - g. Wählen Sie Send Notification im Dropdown-Menü Take Action aus.
 - h. Wählen Sie als Thema Create New Email Topic aus. Geben Sie dann einen Namen für das Thema in das Feld Topic ein.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Schritt 7: Erstellen eines CloudWatch-Alarms

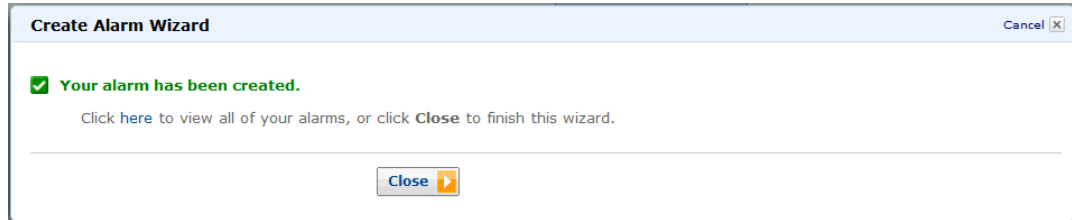
When Alarm state is	Take action	Action details	
ALARM	Auto Scaling Policy	Auto Scaling Group: MyAutoScalingGroup Policy: MyScaleUpPolicy (Add 1)	REMOVE
ALARM	Send Notification	Topic: MyHighCPUAlarmTopic Email(s):	ADD ACTION

- i. Geben Sie eine E-Mail-Adresse in das Feld Email(s) ein.
- j. Klicken Sie auf Add Action.
- k. Klicken Sie auf Weiter.

4. Klicken Sie auf der Seite Review auf Create Alarm.

When Alarm state is	Take action	Action details	
ALARM	Auto Scaling Policy	Auto Scaling Group: MyAutoScalingGroup Policy: MyScaleUpPolicy (Add 1)	REMOVE
ALARM	Send Notification	Topic: MyHighCPUAlarmTopic Email(s):	ADD ACTION

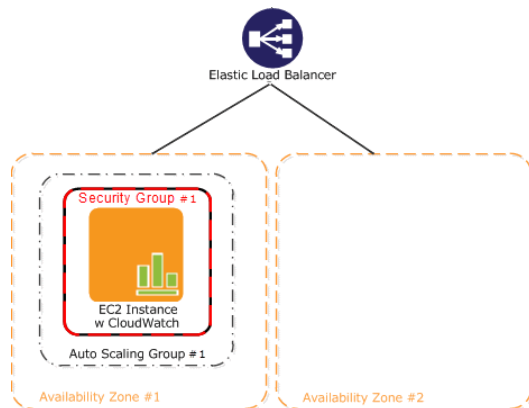
5. Klicken Sie auf Close.



Der neue Alarm wird in der Liste angezeigt. Wenn Sie Ihre MyScaleDownPolicy erstellen, können Sie einen anderen Alarm festlegen, indem Sie die gleichen Schritte ausführen.

Aktueller Stand

Zurzeit befinden Sie sich bei der Erstellung Ihrer Architektur an der folgenden Position.



Als nächstes fügen wir der Webanwendung eine Datenbank hinzu. Amazon bietet eine Reihe von Datenbankoptionen, aber für dieses Beispiel verwenden wir den [Amazon Relational Database Service \(Amazon RDS\)](#), weil er einfach zu bedienen ist und uns die für Datenbanken typische administrative Mehrarbeit erspart.

Schritt 8: Hinzufügen von Amazon RDS

Topics

- [Erstellen einer DB-Sicherheitsgruppe \(p. 25\)](#)
- [Zugriff autorisieren \(p. 26\)](#)
- [Starten einer Instance \(p. 26\)](#)

Jetzt sind wir bereit, die Amazon Relational Database (Amazon RDS) zu unserer Architektur hinzuzufügen. In diesem Schritt starten wir eine Multi-AZ RDS Instance. Wenn Sie Ihre DB-Instance zur Ausführung als Multi-AZ-Bereitstellung erstellen oder ändern, stellt Amazon RDS automatisch eine synchrone Standby-Replikation Ihrer DB-Instance in einer anderen Availability Zone bereit und verwaltet diese. Aktualisierungen Ihrer DB-Instance werden synchron über mehrere Availability Zones an den Standby-Daten repliziert, damit Ihre aktuellsten Datenbank-Updates synchronisiert und gegen einen Ausfall der DB-Instance geschützt bleiben. Bei bestimmten Arten der geplanten Wartung oder im unwahrscheinlichen Fall eines Fehlers der DB-Instances oder in der Availability Zone führt Amazon RDS automatisch ein Failover zum Standby aus, sodass die Lese-/Schreibvorgänge der Datenbank wieder aufgenommen werden können, sobald das Standby hochgestuft wurde. Da der Namensdatensatz für Ihre DB-Instance gleich bleibt, kann

Ihre Anwendung den Datenbankbetrieb ohne manuellen administrativen Eingriff fortsetzen. Mit Multi-AZ-Bereitstellungen erhalten Sie eine transparente Replizierung: Sie interagieren nicht direkt mit der Standby-Instanz, und diese kann nicht zur Unterstützung des Lesedatenverkehrs genutzt werden.



Important

Die DB-Instance, die Sie gleich starten werden, wird live sein (sie wird nicht in einer Sandbox ausgeführt). Es werden die standardmäßigen Amazon RDS-Nutzungsgebühren für die Instance anfallen, bis Sie sie beenden. Die Gesamtkosten werden minimal sein (meistens geringer als ein Dollar), wenn Sie diese Übung in einer Sitzung durchlaufen und die DB-Instance beenden, sobald Sie die Übung abgeschlossen haben. Weitere Informationen über Amazon RDS-Nutzungsgebühren finden Sie auf der [Produktseite von Amazon RDS](#).



Note

Dies ist ein optionaler Schritt. Wenn Sie diesen Schritt auslassen möchten, können Sie mit [Schritt 9: Bereitstellen Ihrer Anwendung \(p. 31\)](#) fortfahren.

Führen Sie folgende Schritte aus, um Ihre Amazon RDS-Datenbank einzurichten:

- Erstellen einer DB-Sicherheitsgruppe
- Autorisieren Ihrer DB-Instance
- Starten einer DB-Instance

Erstellen einer DB-Sicherheitsgruppe

Abstract

So erstellen Sie eine DB-Sicherheitsgruppe, bevor Sie den Zugriff auf Ihre Amazon EC2-Sicherheitsgruppe gewähren können.

Um eine DB-Sicherheitsgruppe zu erstellen, müssen Sie einen Namen und eine Beschreibung eingeben.

Erstellen Sie eine DB-Sicherheitsgruppe wie folgt:

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die Amazon RDS-Konsole unter <https://console.aws.amazon.com/rds/>.
2. Stellen Sie sicher, dass US East (N. Virginia) in der Regionenauswahl in der Navigationsleiste ausgewählt ist.
3. Klicken Sie im linken Navigationsbereich auf Security Groups.
4. Klicken Sie auf Create DB Security Group.
5. Geben Sie den Namen der neuen DB-Sicherheitsgruppe ein. Für dieses Beispiel, geben Sie `mydb-securitygroup` ein.
6. Geben Sie eine Beschreibung für die neue DB-Sicherheitsgruppe in das Textfeld Description ein.
7. Klicken Sie auf Yes, Create.

Jetzt sind Sie bereit, den Zugriff auf die Amazon EC2-Sicherheitsgruppe zu autorisieren.

Zugriff autorisieren

Abstract

Autorisieren Sie den Zugriff, sodass Ihre Amazon EC2-Sicherheitsgruppe auf Ihre Sicherheitsgruppe zugreifen kann.

Nun müssen Sie der Amazon EC2-Sicherheitsgruppe den Zugriff auf die DB-Sicherheitsgruppe gewähren.

Autorisieren Sie Ihre Amazon EC2-Sicherheitsgruppe für den Zugriff auf Ihre DB-Sicherheitsgruppe wie folgt:

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die Amazon RDS-Konsole unter <https://console.aws.amazon.com/rds/>.
2. Stellen Sie sicher, dass US East (N. Virginia) in der Regionenauswahl in der Navigationsleiste ausgewählt ist.
3. Klicken Sie im linken Navigationsbereich auf Security Groups.
4. Wählen Sie mydbsecuritygroup.
5. Wählen Sie im unteren Bereich EC2 Security Group für Connection Type aus.
6. Ihre AWS-Konto-ID erscheint in der rechten Hälfte des unteren Bereichs. Wenn Sie eine andere AWS-ID autorisieren möchten, diese DB-Sicherheitsgruppe zu verwenden, wählen Sie Another account und geben Sie dann Ihre ID in das Feld AWS Account ID ein.



Note

Achten Sie darauf, bei der Eingabe Ihrer Konto-ID die Bindestriche wegzulassen.

7. Wählen Sie als EC2 Security Group Name webappsecuritygroup aus.
8. Klicken Sie auf Authorize.

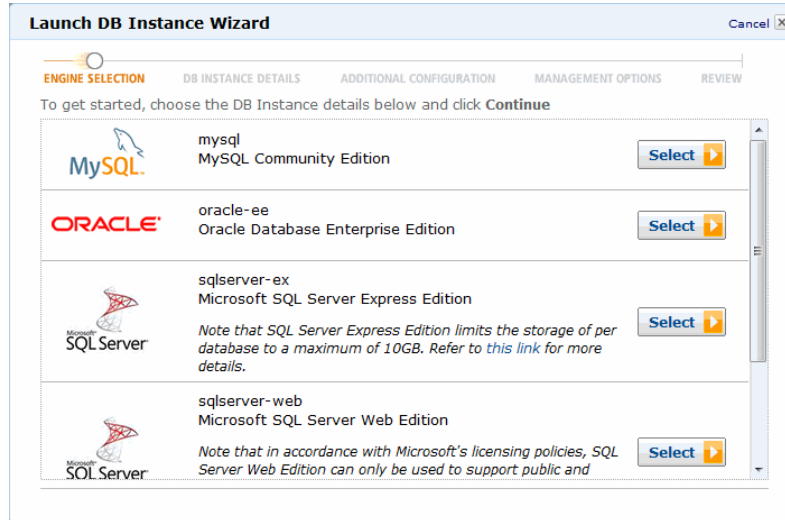
Die Autorisierung wird einige Augenblicke dauern. Nachdem die Sicherheitsgruppe autorisiert wurde, zeigt die Spalte Status im oberen Bereich authorized an. Fahren Sie im nächsten Schritt mit dem Start Ihrer ersten Amazon RDS-Datenbank fort.

Starten einer Instance

So starten Sie eine Instance

1. Starten Sie den Startassistenten:
 - a. Melden Sie sich bei AWS Management Console an und öffnen Sie die Amazon RDS-Konsole unter <https://console.aws.amazon.com/rds/>.
 - b. Stellen Sie sicher, dass US East (N. Virginia) in der Regionenauswahl in der oberen Navigationsleiste ausgewählt ist.
 - c. Klicken Sie im linken Navigationsbereich auf DB instances.
 - d. Klicken Sie im Amazon RDS Console Dashboard auf Launch a DB Instance.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Starten einer Instance



2. Klicken Sie auf Select neben MySQL Community Edition.
3. Geben Sie auf der Seite DB Instance Details Details zu Ihrer DB-Instance an, wie in der folgenden Tabelle gezeigt. Klicken Sie dann auf Continue.

Für diesen Parameter...	Vorgehensweise
License Model	Belassen Sie den Standardwert: general-public-license.
DB Engine Version	Belassen Sie den Standardwert: 5.5.27 (default).
DB-Instance-Klasse	Wählen Sie db.m1.small aus. Die Klasse DB Instance legt die CPU und Speicherkapazität für die Datenbank-Instance fest.
Multi-AZ-Bereitstellung	Wählen Sie Yes. Obwohl die Multi-AZ-Bereitstellung teurer ist, ist sie eine bewährte Methode.
Auto Minor Version Upgrade	Belassen Sie die Standardeinstellung für dieses Beispiel auf Yes. Mit der Option Auto Minor Version Upgrade kann Ihre DB-Instance automatisch kleinere Upgrades der Engine-Version erhalten, wenn sie verfügbar werden.
Allocated Storage	Sie können angeben, wie viel Speicher in Gigabyte sie anfänglich Ihrer DB-Instance zuweisen möchten. Geben Sie für dieses Beispiel 5 ein.
Use Provisioned IOPS	Lassen Sie das Kontrollkästchen deaktiviert. Diese Option schaltet Provisioned IOPS (E/A-Operationen pro Sekunde) ein, eine hochleistungsfähige Speicheroption in RDS, die für E/A-intensive, transaktionale (OLTP-) Datenbank-Arbeitslasten optimiert wurde. Weitere Information über Hochleistungsspeicherung finden Sie unter Provisioned IOPS .

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Starten einer Instance

Für diesen Parameter...	Vorgehensweise
DB Instance Identifier	Die DB-Instance ist ein Name für Ihre DB-Instance, der für Ihr Konto in einer Region eindeutig ist. Geben Sie mydbinstance in das Textfeld DB Instance Identifier ein.
Master Username	Geben Sie einen Namen für Ihren Masterbenutzer in das Textfeld Master Username ein. Sie können sich mit dem Masterbenutzernamen an Ihrer DB-Instance mit allen Datenbankberechtigungen anmelden.
Masterpasswort	Geben Sie ein Passwort für Ihren Masterbenutzer in das Textfeld Master User Passwort ein.



Important

Sie müssen ein Passwort angeben, dass zwischen 4 und 16 nur alphanumerische Zeichen enthält.

Launch DB Instance Wizard Cancel

ENGINE SELECTION **DB INSTANCE DETAILS** ADDITIONAL CONFIGURATION MANAGEMENT OPTIONS REVIEW

To get started, choose a DB engine below and click **Continue**

DB Engine: mysql
License Model: General Public License
DB Engine Version: MySQL 5.5.27 (default)
DB Instance Class: db.m1.small
Multi-AZ Deployment: Yes
Auto Minor Version Upgrade: Yes No

Provide the details for your RDS Database Instance.

Allocated Storage:* 5 GB (Minimum: 5 GB, Maximum: 3072 GB) Higher allocated storage [may improve](#) IOPS performance.
Use Provisioned IOPS:

DB Instance Identifier:* mydbinstance (e.g. mydbinstance)
Master Username:* awsuser (e.g. awsuser)
Master Password:* ●●●●●●●● (e.g. mypassword)

< Back Continue

4. Geben Sie zusätzliche Konfigurationsinformationen für Ihre DB-Instance an:
 - a. Geben Sie **mydb** in das Feld Database Name ein.

Sie geben einen Datenbanknamen an, sodass Amazon RDS eine Standarddatenbank auf Ihrer neuen DB-Instance erstellen kann. Wenn Sie diesen Schritt überspringen, wird Amazon RDS keine Datenbank auf Ihrer DB-Instance erstellen.

- b. Wählen Sie **mydbsecuritygroup** im Feld DB Security Groups aus.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Starten einer Instance

Launch DB Instance Wizard Cancel X

ENGINE SELECTION DB INSTANCE DETAILS **ADDITIONAL CONFIGURATION** MANAGEMENT OPTIONS REVIEW

Provide the optional additional configuration details below.

Database Name: mydb (e.g. mydb)

Note: if no database name is specified then no initial MySQL database will be created on the DB Instance.

Database Port: 3306

Choose a VPC: Not in VPC Only VPCs with a DB Subnet Group(s) are allowed

Availability Zone: No Preference MultiAZ deployment selected

Option Group: defaultmysql-5-5

If you have custom DB Parameter Groups or DB Security Groups you would like to associate with this DB Instance, select them below, otherwise proceed with default settings.

Parameter Group: defaultmysql5.5

DB Security Group(s): mydbsecuritygroup

< Back Continue >

- c. Akzeptieren Sie die Standardwerte für die übrigen verfügbaren Parameter auf dieser Seite und klicken Sie dann auf Continue.
5. Geben Sie auf der Seite Management Options Sicherungs- und Wartungsoptionen für Ihre DB-Instance an. Akzeptieren Sie für dieses Beispiel die Standardwerte und klicken Sie dann auf Continue.

Launch DB Instance Wizard Cancel X

ENGINE SELECTION DB INSTANCE DETAILS ADDITIONAL CONFIGURATION **MANAGEMENT OPTIONS** REVIEW

Enabled Automatic Backups: Yes No

The number of days for which automated backups are retained.

Please note that automated backups are currently **supported for InnoDB storage engine only** . If you are using MyISAM, refer to details [here](#) .

Backup Retention Period: 1 days

The daily time range during which automated backups are created if automated backups are enabled

Backup Window: Select Window No Preference

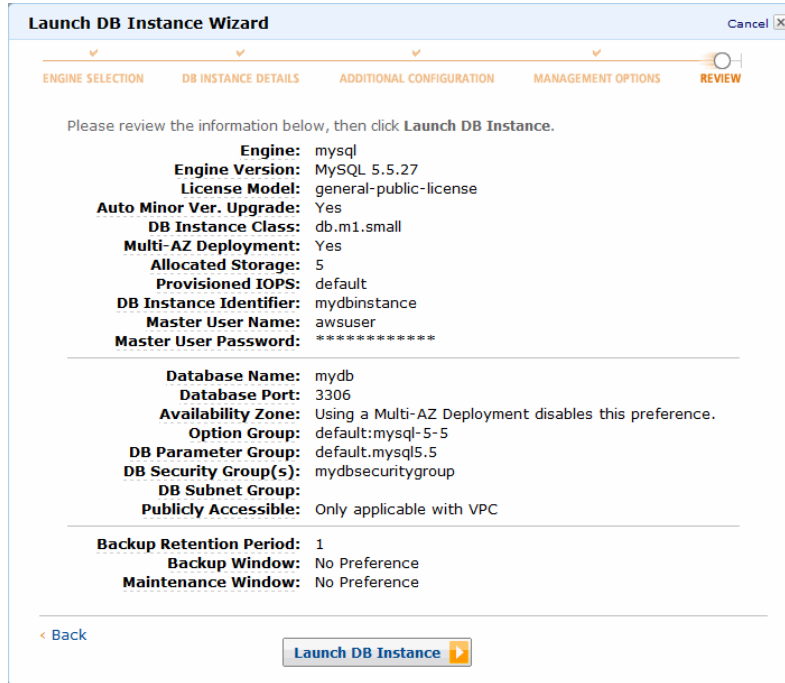
The weekly time range (in UTC) during which system maintenance can occur.

Maintenance Window: Select Window No Preference

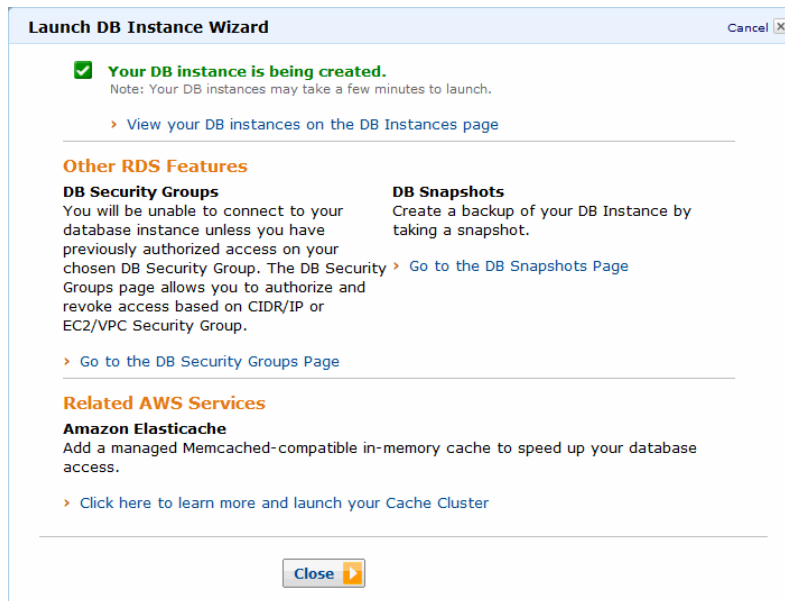
< Back Continue >

6. Überprüfen Sie die Optionen für Ihre DB-Instance. Wenn Sie Optionen korrigieren müssen, klicken Sie auf Back, um zu vorherigen Seiten zurückzukehren und Korrekturen vorzunehmen. Wenn Sie fertig sind, klicken Sie auf Launch DB Instance, um Ihre neue DB-Instance zu starten.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Starten einer Instance



7. Dieser Vorgang kann einige Minuten dauern. Wenn Sie die Benachrichtigung sehen, dass Ihre Instance erstellt wurde, klicken Sie auf Close.



Ihre DB-Instance wird in der Liste auf dieser Seite mit dem Status creating angezeigt, bis die DB-Instance erstellt wurde und verwendet werden kann.

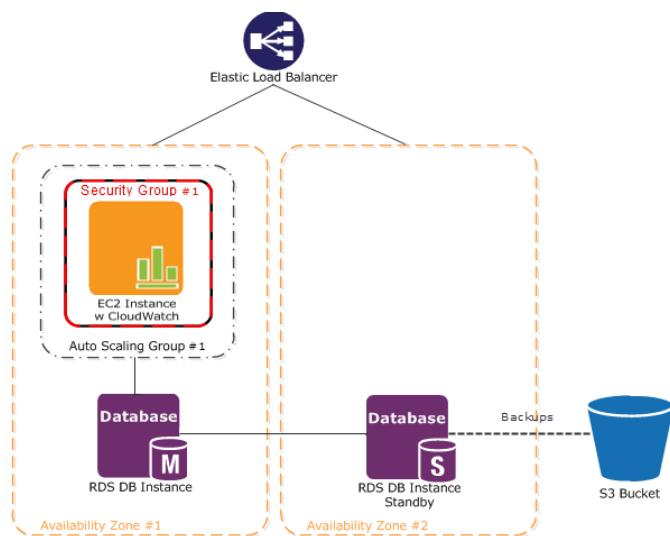
Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Schritt 9: Bereitstellen Ihrer Anwendung

DB Instance	VPC	Multi-AZ	Class	Status	Storage	IOPS	Security Groups	Engine	Zone
mydbinstance		Yes	db.m1.small	creating	5 GB		mydbsecuritygroup (active)	mysql	us-east-1a

Nachdem Ihre DB-Instance auf den Status available gewechselt hat, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, in der die DB-Instance läuft (auch wenn die DB-Instance nicht aktiv ist).

Aktueller Stand

Zurzeit befinden Sie sich bei der Erstellung Ihrer Architektur an der folgenden Position.



Nachdem Sie jetzt Ihre Amazon RDS-Datenbank gestartet haben, sind Sie bereit, Ihre Muster-Webanwendung bereitzustellen.

Schritt 9: Bereitstellen Ihrer Anwendung

Topics

- [Herstellen einer Verbindung mit einer Amazon EC2-Instance über einen Web-Browser mithilfe des MindTerm-SSH-Clients \(p. 32\)](#)
- [Herstellen einer Verbindung mit einer Amazon EC2-Instance von Windows mithilfe von PuTTY \(p. 33\)](#)
- [Herstellen einer Verbindung mit einer Amazon EC2-Instance auf einem Linux/UNIX-Computer mithilfe eines eigenständigen SSH-Clients \(p. 38\)](#)
- [Konfigurieren der Amazon EC2-Instance \(p. 39\)](#)

Nachdem Sie jetzt alle AWS-Ressourcen erstellt haben, ist es an der Zeit, die Anwendung für Ihre Amazon EC2-Instance bereitzustellen. In diesem Schritt stellen wir zuerst die Verbindung mit der Amazon EC2-Instance her und konfigurieren anschließend die Instance mit einer Beispielanwendung, die bereits auf dem Linux AMI verfügbar ist.

Herstellen einer Verbindung mit einer Amazon EC2-Instance über einen Web-Browser mithilfe des MindTerm-SSH-Clients

Führen Sie die folgenden Schritte aus, um über Ihren Web-Browser eine Verbindung mit einer Linux-/UNIX-Instance herzustellen:

1. [Installieren und Aktivieren von Java in Ihrem Browser \(p. 32\)](#)
2. [Herstellen einer Verbindung über den Java-SSH-Client \(p. 32\)](#)

Installieren und Aktivieren von Java in Ihrem Browser

Um eine Verbindung zu Ihrer Instance von der Amazon Elastic Compute Cloud (Amazon EC2)-Konsole herzustellen, muss Java installiert und in Ihrem Browser aktiviert sein. Zum Installieren und Aktivieren von Java führen Sie die von Oracle bereitgestellten unten stehenden Schritte aus oder bitten Sie Ihren IT-Administrator, Java zu installieren und in Ihrem Web-Browser zu aktivieren.



Note

Auf einem Windows- oder Mac-Client muss der Web-Browser mit Administrator-Anmeldeinformationen ausgeführt werden. Für Linux können weitere Schritte notwendig sein, wenn Sie nicht mit dem Stammkonto angemeldet sind.

1. Installieren Sie Java (siehe http://java.com/en/download/help/index_installing.xml).
2. Aktivieren Sie Java in Ihrem Web-Browser (siehe http://java.com/en/download/help/enable_browser.xml).

Herstellen einer Verbindung über den Java-SSH-Client

Stellen Sie über einen Web-Browser eine Verbindung zu Ihrer Instance wie folgt her:

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie im Navigationsbereich auf Instances.
3. Wählen Sie Ihre Instance aus und klicken Sie dann auf Connect.
4. Klicken Sie auf A Java SSH client directly from my browser (Java required). AWS erkennt automatisch die öffentliche DNS-Adresse Ihrer Instance und den Schlüsselpaarnamen, mit dem Sie die Instance gestartet haben.
5. Geben Sie in das Feld User name den Benutzernamen ein, um sich bei der Instance anzumelden. In diesem Beispiel geben Sie `ec2-user` ein.



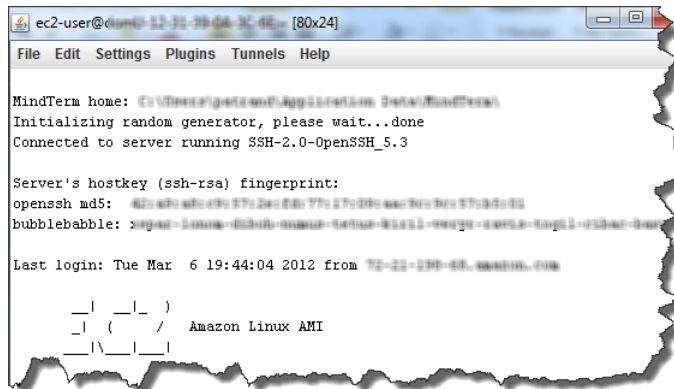
Note

Für eine Amazon Linux-Instance lautet der Standardbenutzername "ec2-user". Für Ubuntu lautet der Standardbenutzername "ubuntu". Bei einigen AMIs können Sie sich als root anmelden.

6. Das Feld Key name wird automatisch für Sie gefüllt.
7. Geben Sie in das Feld Private key path den vollqualifizierten Pfad zu Ihrer PEM-Datei mit dem privaten Schlüssel ein.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Herstellen einer Verbindung mit einer Amazon EC2-Instance
von Windows mithilfe von PuTTY

8. Klicken Sie auf Save key location und auf Stored in browser cache, um den Speicherort des Schlüssels im Browser-Cache zu speichern. Solange Sie den Browser-Cache nicht löschen, wird der Speicherort des Schlüssels auf diese Weise in nachfolgenden Browser-Sitzungen erkannt.
9. Klicken Sie auf Launch SSH Client.
10. Klicken Sie, falls erforderlich, auf Yes, um das Zertifikat als vertrauenswürdig einzustufen.
11. Klicken Sie auf Run, um den MindTerm-Client auszuführen.
12. Wenn Sie den Lizenzbestimmungen zustimmen, klicken Sie auf Accept.
13. Wenn Sie MindTerm zum ersten Mal ausführen, werden Sie aufgefordert, in einer Reihe von Dialogfeldern das Setup Ihres Startverzeichnisses und andere Einstellungen zu bestätigen.
14. Bestätigen Sie die Einstellungen für das MindTerm-Setup.
15. Es wird ein Bildschirm angezeigt, der etwa wie der folgende aussieht, und die Verbindung zu Ihrer Instance wird hergestellt.



Sollten bei der Verbindung mit MindTerm Schwierigkeiten auftreten, überprüfen Sie Folgendes:

- Ist Java installiert und in Ihrem Browser aktiviert?
- Verwenden Sie den richtigen Benutzernamen?
- Haben Sie das richtige Schlüsselpaar und den korrekten Pfad zu Ihrem privaten Schlüssel angegeben?
- Haben Sie Ihre Sicherheitsgruppe so konfiguriert, dass Sie sich mit Ihrer Instance verbinden können? .
- Sollten weiterhin Probleme auftreten, suchen Sie in den [AWS-Foren](#) nach einer Lösung.

16. Starten Sie den Web-Server mit dem Befehl `sudo service httpd start`.

```
sudo service httpd start
```

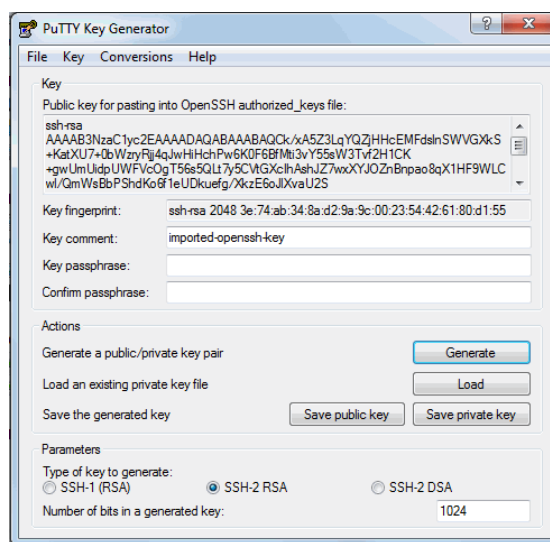
Herstellen einer Verbindung mit einer Amazon EC2-Instance von Windows mithilfe von PuTTY

Wenn Sie Windows auf Ihrem lokalen Computer ausführen, ist Secure Shell (SSH) nicht vorhanden, sodass Sie PuTTY und PuTTYGen installieren müssen. Zum Herstellen einer Verbindung mit einer Linux-Instance müssen Sie zunächst das ursprüngliche Administratorpasswort abrufen und es dann mit Microsoft Remote Desktop verwenden. Sie benötigen den Inhalt der Datei mit dem privaten Schlüssel, die Sie in [Schritt 5: Erstellen eines Schlüsselpaares](#) (p. 16) erstellt haben (beispielsweise `mykeypair.pem`).

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Herstellen einer Verbindung mit einer Amazon EC2-Instance
von Windows mithilfe von PuTTY

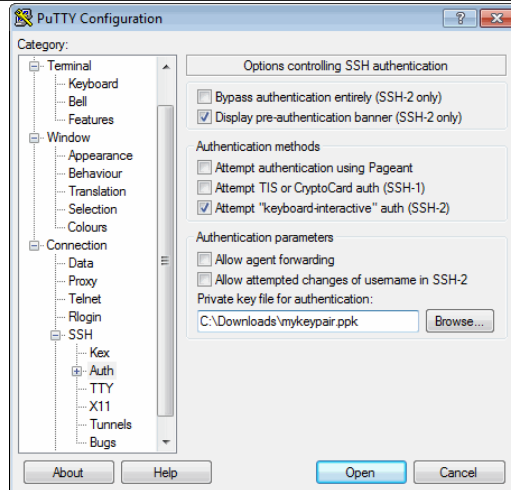
Stellen Sie eine Verbindung mit einer Amazon EC2-Instance auf einem Windows-Computer wie folgt her:

1. Installieren Sie PuTTY und PuTTYGen.
 - Laden Sie PuTTY herunter und installieren Sie es. Eine Liste mit Download-Sites erhalten Sie, indem Sie mit Google nach "download Putty" suchen. Achten Sie darauf, sowohl PuTTY als auch PuTTYGen zu installieren, da sie beide benötigen.
2. Konvertieren Sie das Schlüsselpaar mithilfe von PuTTYGen. (Weitere Informationen über Schlüsselpaare erhalten Sie unter [Schritt 5: Erstellen eines Schlüsselpaares \(p. 16\).](#))
 - a. Starten Sie PuTTYGen und wählen Sie Import Key im Menü Conversions aus.
 - b. Suchen Sie nach `mykeypair.pem` und importieren Sie den Schlüssel.

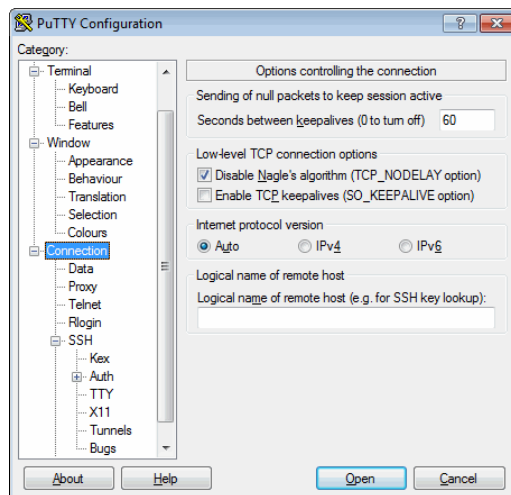


- c. Klicken Sie auf Save private key. Ignorieren Sie das Dialogfeld, in dem Sie gefragt werden, ob Sie dies ohne Pass-Phrase tun möchten. Speichern Sie den Schlüssel unter dem Namen `mykeypair.ppk`.
 - d. Schließen Sie PuTTYGen.
3. Konfigurieren Sie die SSH-Einstellungen.
 - a. Starten Sie PuTTY, erweitern Sie den SSH-Knoten und klicken Sie auf Auth.
 - b. Geben Sie den Speicherort für `mykeypair.ppk` ein.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Herstellen einer Verbindung mit einer Amazon EC2-Instance
von Windows mithilfe von PuTTY

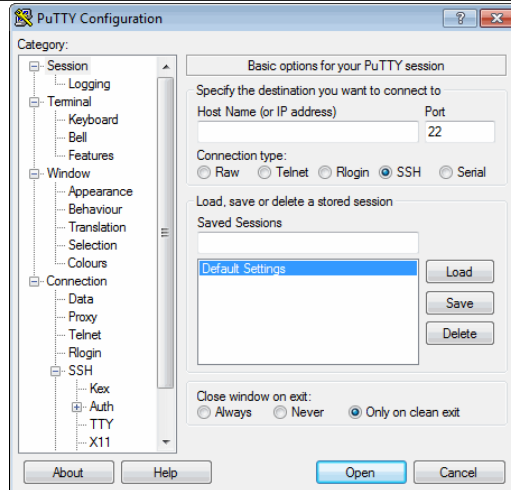


4. Ändern Sie die keepalive-Einstellung.
 - a. Klicken Sie im Fenster "PuTTY Configuration" im Bereich Category auf Connection.
 - b. Geben Sie im Feld Seconds between keepalives (0 to turn off) den Wert 60 ein. Wenn Sie diesen Wert nicht ändern, wird Ihre Sitzung durch Timeout beendet.



5. Übernehmen Sie die anderen Sitzungseinstellungen.
 - a. Klicken Sie im Fenster "PuTTY Configuration" im Bereich Category auf Session.
 - b. Klicken Sie im Feld Load, save, or delete a stored session auf Default Settings und dann auf Save.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Herstellen einer Verbindung mit einer Amazon EC2-Instance
von Windows mithilfe von PuTTY

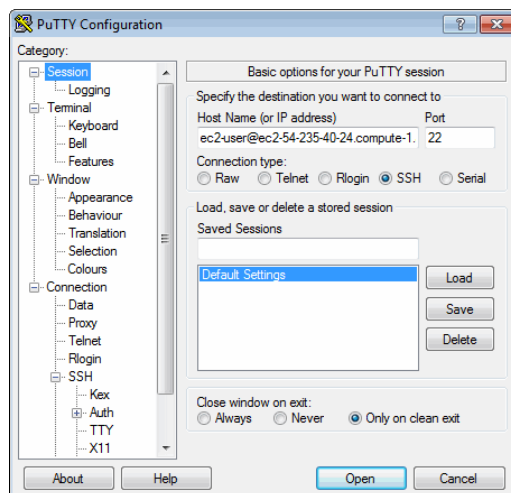


6. Geben Sie die DNS-Adresse Ihrer Amazon EC2-Instance ein.
 - a. Melden Sie sich bei AWS Management Console an und öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
 - b. Stellen Sie sicher, dass US East (N. Virginia) in der Regionenauswahl in der Navigationsleiste ausgewählt ist.
 - c. Klicken Sie im Navigationsbereich auf Instances.
 - d. Wählen Sie die aktive Instance und notieren Sie sich die im unteren Bereich angegebene öffentliche DNS-Adresse.
 - e. Klicken Sie im Fenster "PuTTY Configuration" im Bereich Category auf Sessions. Geben Sie im Feld Host Name (or IP address) den Wert `<ec2-user@DNS address of your Amazon EC2 instance>` ein.



Note

Wir geben `ec2-user` vor dem DNS-Namen ein, da der Benutzername für das AMI `ec2-user` lautet.



Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Herstellen einer Verbindung mit einer Amazon EC2-Instance
von Windows mithilfe von PuTTY

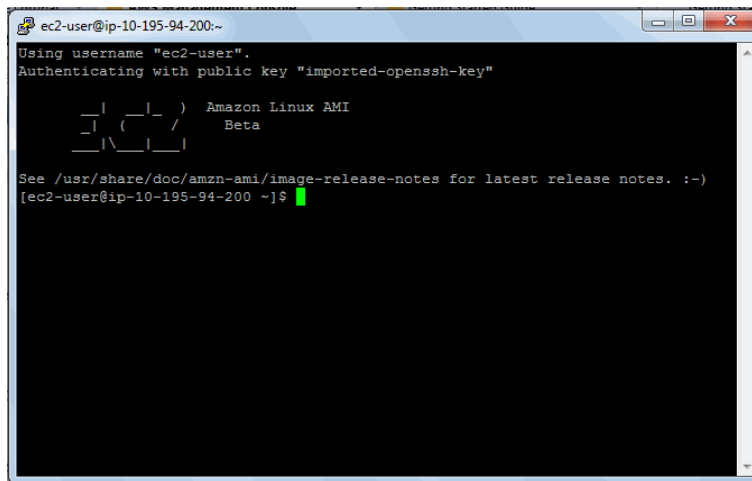
- f. Klicken Sie auf Open. Das Dialogfeld "PuTTY Security Alert" wird geöffnet. Klicken Sie darin auf Yes, um zu bestätigen, dass der Fingerprint korrekt ist. Das Fenster "SSH PuTTY" wird geöffnet.



Note

Der SSH-Fingerprint wird möglicherweise im Systemprotokoll angezeigt. Der SSH-Fingerprint kann zum Schutz vor einem Man-In-the-Middle-Angriff als Vergleichskriterium dienen.

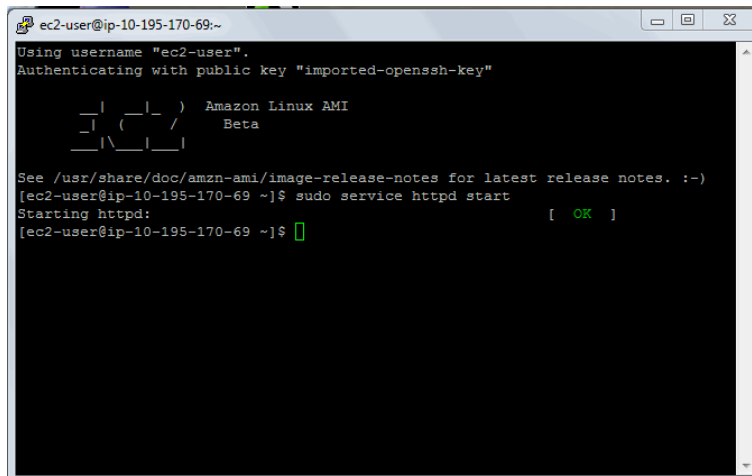
Ihr Bildschirm sollte wie folgt aussehen:



- g. Starten Sie den Web-Server mit dem Befehl `sudo service httpd start`.

```
sudo service httpd start
```

Ihr Bildschirm sollte wie folgt aussehen:



Sie haben sich jetzt erfolgreich bei Ihrem AMI angemeldet und können dieses nun konfigurieren. Anleitungen zum Konfigurieren eines AMI finden Sie unter [Konfigurieren der Amazon EC2-Instance \(p. 39\)](#).

Herstellen einer Verbindung mit einer Amazon EC2-Instance auf einem Linux/UNIX-Computer mithilfe eines eigenständigen SSH-Clients

Verwenden Sie auf einem Linux/UNIX-Computer den Befehl `ssh`, um eine Verbindung mit Ihren Linux-/UNIX-Instances herzustellen.



Note

Die meisten Linux- und UNIX-Computer enthalten standardmäßig einen SSH-Client. Falls dies bei Ihnen nicht der Fall ist, bietet das OpenSSH-Projekt eine kostenlose Implementierung der umfassenden Palette von SSH-Tools. Weitere Informationen erhalten Sie unter <http://www.openssh.org>.

Verwenden Sie SSH für die Verbindung wie folgt:

1. Ändern Sie die Verzeichnisse in einer Befehlszeilen-Shell in den Speicherort der Datei des privaten Schlüssels, die Sie unter [Schritt 5: Erstellen eines Schlüsselpaares \(p. 16\)](#) erstellt haben.
2. Verwenden Sie den Befehl `chmod`, um sicherzustellen, dass die Datei des privaten Schlüssels nicht öffentlich sichtbar ist. Geben Sie beispielsweise für `mykeypair.pem` Folgendes ein:

```
chmod 400 mykeypair.pem
```

3. Melden Sie sich bei AWS Management Console an und öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
4. Stellen Sie sicher, dass US East (N. Virginia) in der Regionenauswahl in der Navigationsleiste ausgewählt ist.
5. Klicken Sie im linken Navigationsbereich auf Instances.
6. Wählen Sie Ihre Instance aus und klicken Sie dann auf Connect.
7. Klicken Sie auf A Java SSH client directly from my browser (Java required). AWS erkennt automatisch die öffentliche DNS-Adresse Ihrer Instance und den Schlüsselpaarnamen, mit dem Sie die Instance gestartet haben.
8. Stellen Sie eine Verbindung mit Ihrer Instance unter deren öffentlichem DNS-Namen her. Lautet beispielsweise der Schlüsselname `mykeypair.pem` und der DNS-Name der Instance `ec2-107-20-66-228.compute-1.amazonaws.com`, verwenden Sie den folgenden Befehl.

```
ssh -i mykeypair.pem ec2-user@ec2-107-20-66-228.compute-1.amazonaws.com
```



Note

In dieser Übung für dieses AMI ist `ec2-user` der Benutzername.

Sie erhalten eine Antwort wie die folgende.

```
The authenticity of host 'ec2-107-20-66-228.compute-1.amazonaws.com
(10.254.142.33)'
can't be established.
RSA key fingerprint is 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00.
Are you sure you want to continue connecting (yes/no)? yes
```

9. Geben Sie **yes** ein.

Sie erhalten eine Antwort wie die folgende.

```
Warning: Permanently added 'ec2-107-20-66-228.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
```

10. Starten Sie den Web-Server mit dem Befehl `sudo service httpd start`.

```
sudo service httpd start
```

Sie erhalten eine Antwort wie die folgende.

```
Starting httpd [OK]
```

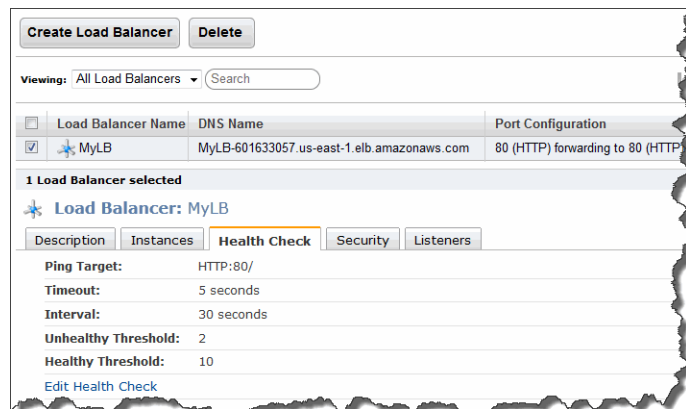
Sie haben sich jetzt erfolgreich bei Ihrem AMI angemeldet und können dieses nun konfigurieren. Anleitungen zum Konfigurieren eines AMI finden Sie unter [Konfigurieren der Amazon EC2-Instance](#) (p. 39).

Konfigurieren der Amazon EC2-Instance

In diesem Thema wird das aktive AMI konfiguriert. Zuerst konfigurieren wir die Zustandsprüfung für den Load Balancer, damit wir die Verbindung mit der Instance über den Load Balancer herstellen können. Wir ändern die Zustandsprüfung vorübergehend, damit diese auf das Skript "install.php" zeigt, bis unsere Instance konfiguriert ist. Nachdem die Instance konfiguriert wurde, lassen wir die Zustandsprüfung wieder auf den Stamm zeigen.

Konfigurieren Sie die Zustandsprüfung wie folgt, sodass Sie auf das Installationskript zeigt:

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Stellen Sie sicher, dass US East (N. Virginia) in der Regionenauswahl in der Navigationsleiste ausgewählt ist.
3. Klicken Sie im linken Navigationsbereich auf Load Balancers.
4. Klicken Sie auf Ihren Load Balancer und dann auf die Registerkarte Health Check.



5. Klicken Sie auf Edit Health Check.

Erste Schritte mit AWS Hosten einer Webanwendung für Linux Konfigurieren der Amazon EC2-Instance

6. Geben Sie im Dialogfeld Configure Health Check den Wert `/install.php` im Feld Ping Path ein. Klicken Sie dann auf Save.

Configure Health Check Cancel

Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer.

Ping Protocol: HTTP

Ping Port: 80

Ping Path: /install.php

Response Timeout: 5 Seconds

Health Check Interval: 0.5 Minutes

Unhealthy Threshold: 2

Healthy Threshold: 2

Save

Konfigurieren der Anwendung

Create Load Balancer Delete

Viewing: All Load Balancers 1 to 7 of 7 Items

Load Balancer Name	DNS Name	Port Configuration	Availability Zones
<input checked="" type="checkbox"/> MyLB	MyLB-832486740.us-east-1.elb.amazonaws.com	80 (HTTP) forwarding to 80 (HTTP)	us-east-1b, us-east-1c

1 Load Balancer selected

Load Balancer: MyLB

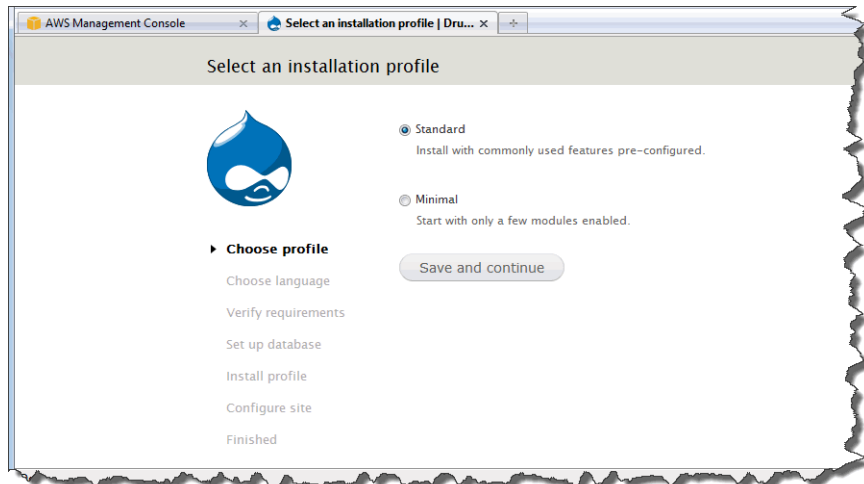
Description Instances Health Check Security Listeners

DNS Name: MyLB-832486740.us-east-1.elb.amazonaws.com (A Record)
ipvb.MyLB-601633057.us-east-1.elb.amazonaws.com (AAAA Record)
dualstack.MyLB-601633057.us-east-1.elb.amazonaws.com (A or AAAA Record)

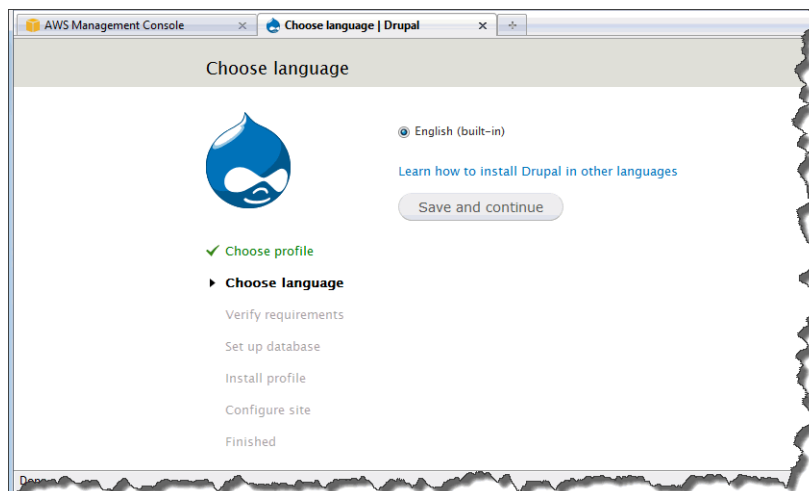
Note: Because the set of IP addresses associated with a LoadBalancer can change over time, you should never create an "A" record with any specific IP address. If you want to use a friendly DNS name for your LoadBalancer instead of the name generated by the Elastic Load Balancing service, you should create a CNAME record for the LoadBalancer DNS name, or use Amazon Route 53 to create a hosted zone. For more information, see the [Using Domain Names With Elastic Load Balancing](#)

1. Klicken Sie bei immer noch ausgewähltem Load Balancer auf die Registerkarte Description. Notieren Sie sich den öffentlichen DNS-Namen unter DNS Name.
2. Öffnen Sie Ihren Web-Browser, und geben Sie die öffentliche DNS-Adresse des Load Balancers in die Adressleiste ein.
3. Klicken Sie auf der Seite Choose profile auf Standard und dann auf Save and continue.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Konfigurieren der Amazon EC2-Instance

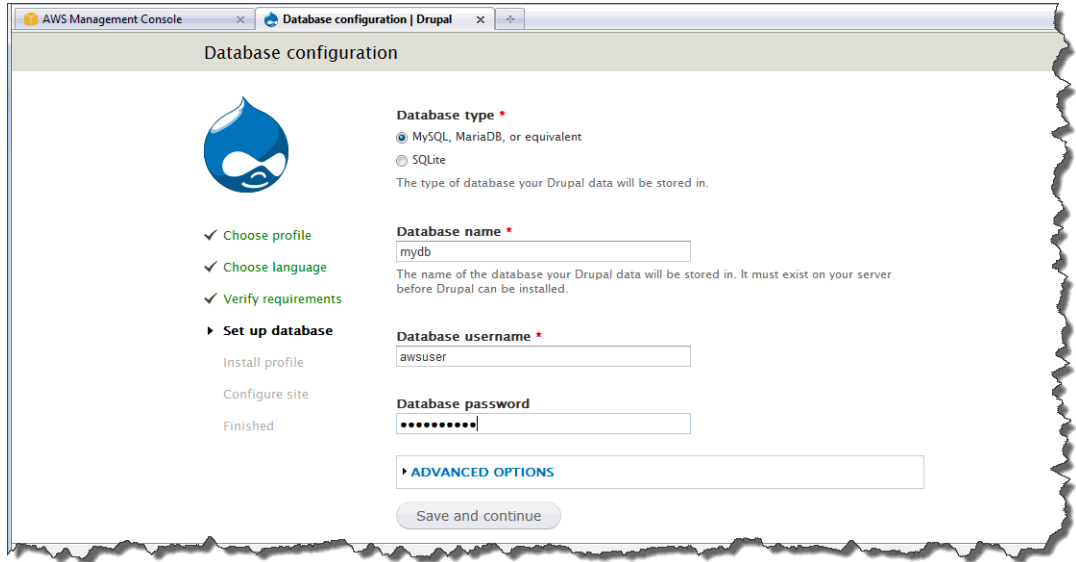


4. Klicken Sie auf der Seite Choose language auf English und dann auf Save and continue.



5. Geben Sie auf der Seite Set up database Folgendes ein:
- Klicken Sie für den Datenbanktyp auf MySQL, MariaDB, or equivalent.
 - Geben Sie im Feld Database name den Namen Ihrer Datenbank ein. In diesem Beispiel wird **mydb** verwendet.
 - Geben Sie im Feld Database username den Benutzernamen für Ihre Datenbank ein. In diesem Beispiel wird **awsuser** verwendet.
 - Geben Sie im Feld Database password das Passwort für Ihre Datenbank ein. In diesem Beispiel wird **mypassword** verwendet.

Erste Schritte mit AWS Hosten einer Webanwendung für Linux Konfigurieren der Amazon EC2-Instance



Database configuration

Choose profile
Choose language
Verify requirements
Set up database
Install profile
Configure site
Finished

Database type *
 MySQL, MariaDB, or equivalent
 SQLite
The type of database your Drupal data will be stored in.

Database name *
mydb
The name of the database your Drupal data will be stored in. It must exist on your server before Drupal can be installed.

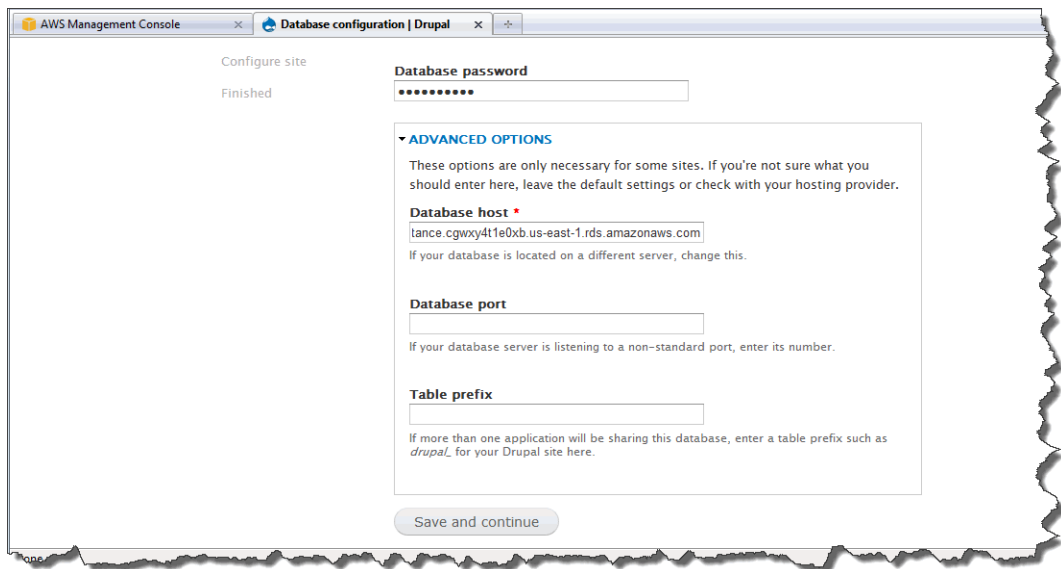
Database username *
awsuser

Database password

ADVANCED OPTIONS

Save and continue

- e. Klicken Sie auf ADVANCED OPTIONS.
- f. Geben Sie im Feld Database host den Amazon RDS-Endpoint ein.



Configure site
Finished

Database password

ADVANCED OPTIONS
These options are only necessary for some sites. If you're not sure what you should enter here, leave the default settings or check with your hosting provider.

Database host *
lance.cgwxy4t1e0xb.us-east-1.rds.amazonaws.com
If your database is located on a different server, change this.

Database port
If your database server is listening to a non-standard port, enter its number.

Table prefix
If more than one application will be sharing this database, enter a table prefix such as *drupal_* for your Drupal site here.

Save and continue



Note

Sie finden den Amazon RDS-Endpoint in der Amazon RDS-Konsole auf der Seite "My DB Instances", wie in der folgenden Abbildung dargestellt.

Erste Schritte mit AWS Hosten einer Webanwendung für Linux

Konfigurieren der Amazon EC2-Instance

The screenshot shows the AWS Management Console interface for an Amazon RDS DB Instance. The instance is named 'mydbinstance' and is in the 'available' state. Key details include:

- DB Instance Name: mydbinstance
- DB Engine: mysql
- DB Engine Version: 5.5.27
- DB Status: available
- DB Storage: 20GiB
- Zone: us-east-1e
- License Model: general-public-license
- Multi-AZ Deployment: No
- DB Security Groups: default (active)
- VPC Security Groups: **Not a VPC Instance**
- Secondary Zone:
- DB Instance Class: db.m1.small
- Endpoint: mydbinstance.cg034hpkmj1t.us-east-1.rds.amazonaws.com (circled in red)
- Port: 3306
- IOPS:
- DB Name: mydb (circled in red)
- Master Username: awsuser (circled in red)
- Created Time: 2012 August 2 10:31:49 UTC-7

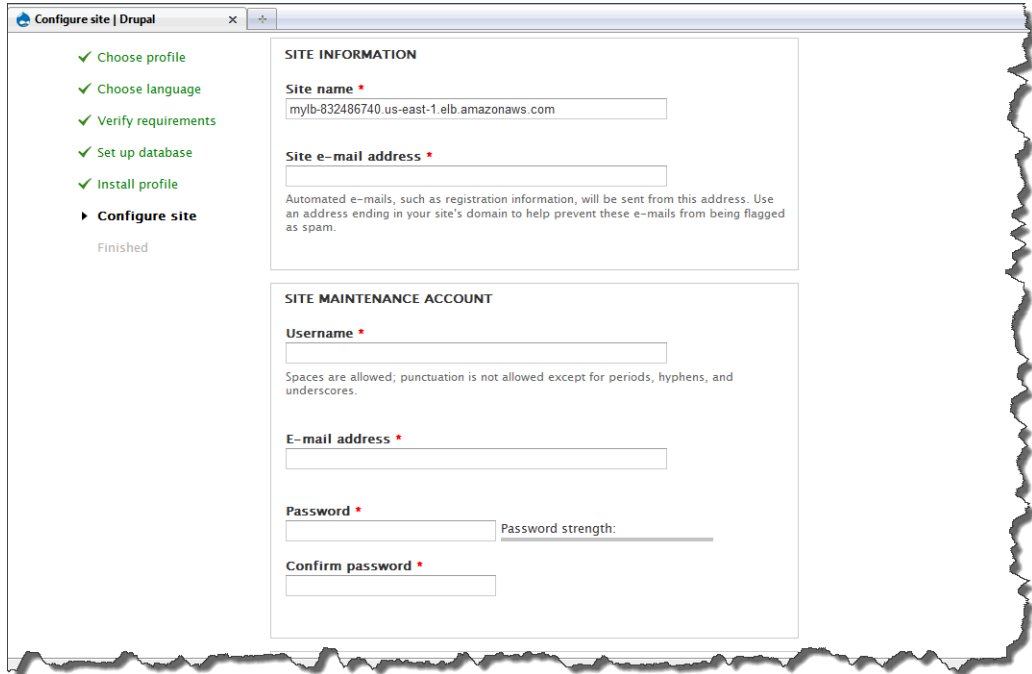


Note

Stellen Sie sicher, dass Ihre Datenbank-Instance reibungslos funktioniert, bevor Sie mit dem nächsten Schritt fortfahren. Der Status sollte available lauten (siehe Diagramm oben).

- g. Klicken Sie auf Save and continue. Die Seite Configure site wird angezeigt.
-
6. Geben Sie auf der Seite Configure site Folgendes ein.
 - a. Geben Sie im Feld Site name die DNS-Adresse des Load Balancers wie zu Beginn dieser Vorgehensweise ein.
 - b. Geben Sie im Feld Site e-mail address eine E-Mail-Adresse ein.
 - c. Geben Sie im Feld Username einen Benutzernamen ein.
 - d. Geben Sie im Feld Password ein Passwort ein.
 - e. Geben Sie im Feld Confirm password das Passwort erneut ein.

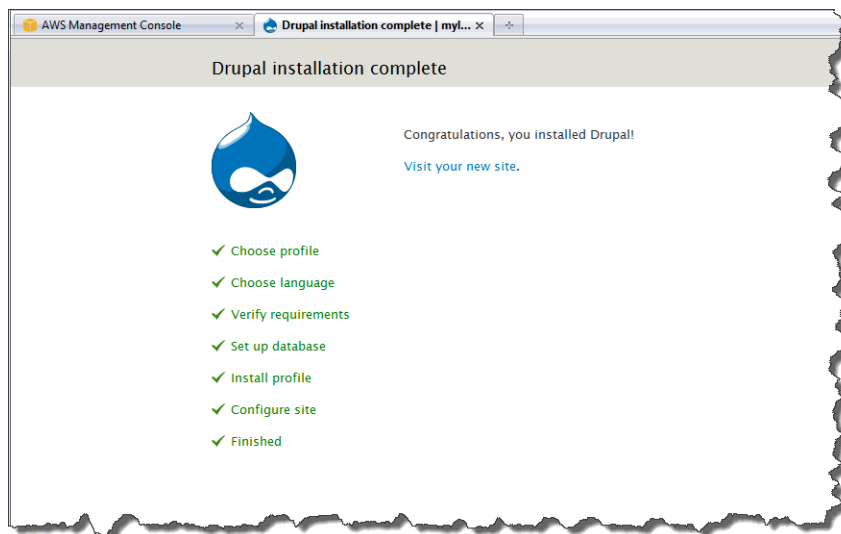
Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Konfigurieren der Amazon EC2-Instance



The screenshot shows the 'Configure site | Drupal' interface. On the left, a progress list includes: 'Choose profile', 'Choose language', 'Verify requirements', 'Set up database', 'Install profile', and 'Configure site' (which is expanded to show 'Finished'). The main content area is divided into two sections: 'SITE INFORMATION' and 'SITE MAINTENANCE ACCOUNT'. The 'SITE INFORMATION' section contains a 'Site name' field with the value 'mylb-832486740.us-east-1.elb.amazonaws.com' and an empty 'Site e-mail address' field. Below this is a note: 'Automated e-mails, such as registration information, will be sent from this address. Use an address ending in your site's domain to help prevent these e-mails from being flagged as spam.' The 'SITE MAINTENANCE ACCOUNT' section contains fields for 'Username', 'E-mail address', 'Password', and 'Confirm password'. The 'Password' field includes a 'Password strength' indicator.

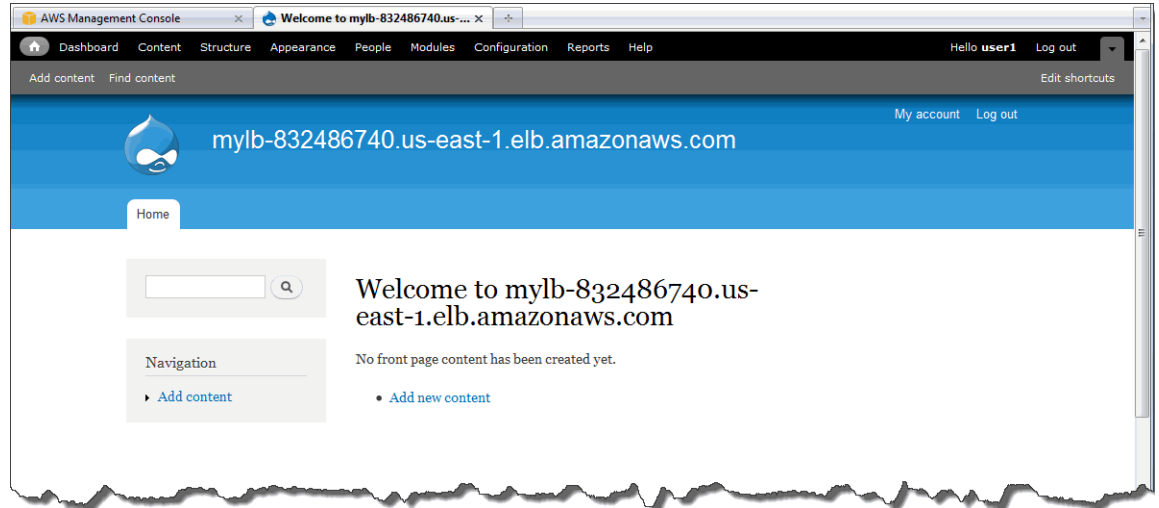
f. Klicken Sie auf Save and continue.

Die Installation ist abgeschlossen.



7. Klicken Sie auf Visit your new site. Ihre neue Website wird angezeigt.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Konfigurieren der Amazon EC2-Instance



8. Um Ihrer neuen Website einen neuen Artikel hinzuzufügen, klicken Sie auf Add new content.

Nachdem Sie nun Ihre neue Website erstellt haben, können Sie Ihre Datenbank sichern.

Sichern Sie Ihre Datenbank wie folgt:

1. Geben Sie an der Eingabeaufforderung den folgenden Befehl ein, um zum Startverzeichnis zu wechseln.

```
cd
```

2. Geben Sie an der Eingabeaufforderung den folgenden Befehl ein, um einen neuen Ordner namens "backups" zu erstellen.

```
mkdir backups
```

3. Geben Sie an der Eingabeaufforderung den folgenden Befehl ein, um Ihre vorhandene Datenbank zu sichern.

```
mysqldump -u awsuser -pmypassword mydb --host=your Amazon RDS endpoint > backups/backup.sql
```

Stellen Sie sicher, dass Sie den Amazon RDS-Endpunkt durch *Ihren Amazon RDS-Endpunkt*, den Sie im vorherigen Schritt notiert haben, ersetzen.

4. Überprüfen Sie, ob die Sicherung vorhanden ist.
 - a. Geben Sie an der Eingabeaufforderung den folgenden Befehl ein, um zum Ordner "backups" zu wechseln.

```
cd backups
```

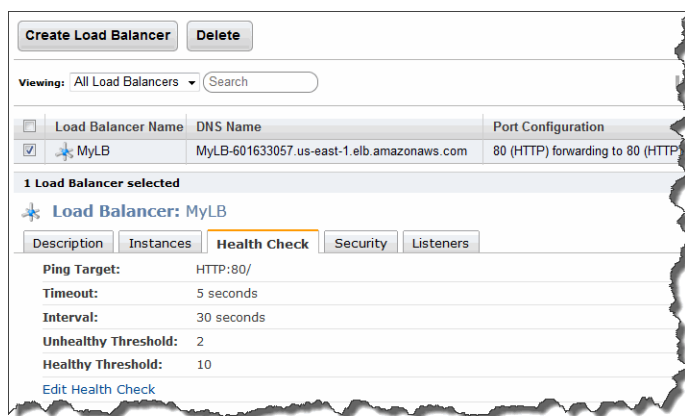
- b. Geben Sie an der Eingabeaufforderung den folgenden Befehl ein, um den Inhalt des Verzeichnisses aufzulisten.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Konfigurieren der Amazon EC2-Instance

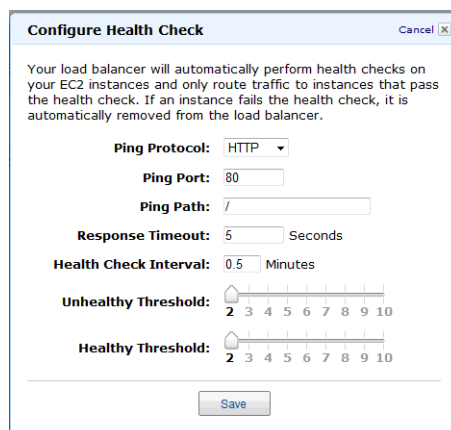
```
ls -al
```

Konfigurieren Sie die Zustandsprüfung wie folgt, sodass sie auf das Stammdokument zeigt:

1. Klicken Sie in der AWS Management Console auf die Registerkarte "Amazon EC2".
2. Klicken Sie im Navigationsbereich auf Instances.
3. Wählen Sie im Navigationsbereich aus dem Dropdown-Menü Region die Region US East (N. Virginia) aus.
4. Klicken Sie im Bereich Load Balancers auf Ihren Load Balancer und dann auf die Registerkarte Health Check.



5. a. Klicken Sie auf Edit Health Check.
b. Geben Sie im Dialogfeld Configure Health Check / im Feld Ping Path ein.



Herzlichen Glückwunsch! Sie haben Ihre Webanwendung erfolgreich mit Amazon Web Services bereitgestellt! Wenn Sie jetzt zukünftig weitere Instances starten möchten, werden Sie nicht jede einzeln anpassen wollen. Fahren wir damit fort, ein benutzerdefiniertes AMI mit all unseren Konfigurationsänderungen zu erstellen.

Schritt 10: Erstellen eines benutzerdefinierten Amazon-Computerabbilds (AMI)

Nachdem wir nun unsere Amazon EC2-Instance angepasst haben, können wir dieses Amazon-Computerabbild (AMI) speichern und zukünftige Umgebungen mit dieser gespeicherten Konfiguration mithilfe von AWS CloudFormation starten. Dieser Schritt ist optional. Wenn Sie das Tutorial lieber jetzt beenden möchten, können Sie ihn überspringen und Ihre AWS-Ressourcen in [Schritt 12: Bereinigen \(p. 58\)](#) bereinigen.

Erstellen eines AMI von einer laufenden Amazon EBS-gestützten Instance

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Stellen Sie sicher, dass US East (N. Virginia) in der Regionenauswahl in der Navigationsleiste ausgewählt ist.
3. Klicken Sie im Navigationsbereich auf Instances.
4. Klicken Sie mit der rechten Maustaste auf der Seite Instances auf die ausgeführte Instance und wählen Sie Create Image aus.
5. Geben Sie einen eindeutigen Namen für das Abbild und optional eine Beschreibung des Abbildes (bis zu 255 Zeichen) ein und klicken Sie auf Create Image.



Tip

Wenn Sie mit Amazon EC2-Instance-Speicher-gestützten AMIs vertraut sind, wissen Sie, dass der Name des Abbildes den Namen des Manifests ersetzt (z. B. `s3_bucket/something_of_your_choice.manifest.xml`), der jeden Amazon Amazon EC2-Instance-Speicher-gestützten AMI eindeutig identifiziert.

Amazon EC2 fährt die Instance herunter, erstellt Abbilder von allen zugewiesenen Volumes, erstellt und registriert das AMI und startet die Instance anschließend neu.

6. Gehen Sie zur Seite AMIs und zeigen Sie den Status des AMI an. Während das neue AMI erstellt wird, hat es den Status *pending*.

Es dauert einige Minuten, bis der Prozess abgeschlossen ist.

7. Sobald der Status des neuen AMI *available* lautet, navigieren Sie zur Seite Snapshots, um den Snapshot anzuzeigen, der für das neue AMI erstellt wurde. Alle Instances, die Sie über das neue AMI starten, verwenden diesen Snapshot als Root-Gerät-Datenträger. Sie können Ihre Auto Scaling-Gruppe mit dem neuen AMI aktualisieren, wir werden dies allerdings im AWS CloudFormation-Schritt ohnehin tun.

Name	Snapshot ID	Capacity	Description	Status	Started	Pr
empty	snap-c9d88f8e	8 GiB	Created by CreateImage(i-...) for ami-5ee27837 from vol-566cc025	completed	2013-03-28 13:18 PDT	ava

Wir haben bislang viele Schritte unternommen, um all unsere AWS-Ressourcen zu erstellen, unsere Anwendung bereitzustellen und unser AMI zu definieren. Wäre es nicht großartig, wenn wir all diese Informationen speichern und neue Umgebungen starten könnten, ohne alles manuell neu konfigurieren zu müssen? Genau das können wir! [AWS CloudFormation](#) ist eine Möglichkeit, Umgebungen auf einfache Weise zu starten. Das bedeutet, wenn Sie eine AWS CloudFormation-Umgebung starten, können Sie spezifische AMIs mit bestimmten Schlüsselpaaren starten, auf vordefinierten Instance-Größen und hinter

Load Balancern. Und wenn irgendein Teil Ihrer Umgebung nicht startet, setzt sich die Umgebung selbst zurück und beendet dabei alle Teile. Fahren wir mit dem nächsten Thema fort und beginnen wir, AWS CloudFormation zu verwenden.

Schritt 11: Starten neuer Umgebungen mit AWS CloudFormation

Topics

- [Erstellen einer AWS CloudFormation-Vorlage \(p. 48\)](#)
- [Ändern einer CloudFormation-Vorlage \(p. 51\)](#)
- [Erstellen eines AWS CloudFormation-Stapels \(p. 53\)](#)

Sie können AWS CloudFormation verwenden, um vorhersagbar und wiederholt AWS-Infrastrukturen zu erstellen und bereitzustellen. Entwickeln Sie mit AWS CloudFormation äußerst zuverlässige, hoch skalierbare und kostengünstige Anwendungen, ohne sich Gedanken über die Erstellung und Konfiguration der zugrunde liegenden Infrastruktur zu machen. AWS CloudFormation besteht aus Vorlagedateien, mit denen Sie Ressourcensammlungen als eine einzelne Einheit (einen AWS CloudFormation-Stapel) erstellen und löschen können. Mit AWS CloudFormation können Sie weitere Services nutzen, zum Beispiel Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Block Store (Amazon EBS), Amazon Simple Notification Service (Amazon SNS), Elastic Load Balancing und Auto Scaling.

In diesem Beispiel verwenden wir das CloudFormer-Tool, um eine Vorlage basierend auf den gerade erstellten AWS-Ressourcen zu generieren. CloudFormer soll einen Startpunkt für Ihre Vorlage erstellen. Nachdem Sie die Vorlage erstellt haben, passen Sie diese an, um eine neue Umgebung mit mehreren Instances zu starten, die sich über mehrere Availability Zones erstreckt, um eine Fehlertoleranzarchitektur zu errichten.

Dieser Schritt ist optional. Wenn Sie diesen Schritt überspringen möchten, können Sie mit [Schritt 12: Bereinigen \(p. 58\)](#) fortfahren, um mit dem Löschen Ihrer Ressourcen zu beginnen.

Erstellen einer AWS CloudFormation-Vorlage

Zunächst müssen Sie eine Vorlage basierend auf den gerade erstellten Ressourcen erstellen. Dafür verwenden Sie ein Tool namens CloudFormer, das Informationen über alle laufenden Ressourcen sammelt und eine Vorlage erstellt. CloudFormer ist ein Prototyp, der Sie zu Beginn unterstützt. Sie können dann einige Änderungen an der Vorlage vornehmen, bevor Sie Ihren neuen Stapel erstellen. Besuchen Sie die [AWS-Foren](#), um mehr zu erfahren und um das Tool auszuführen.

Nachdem Sie die Vorlage generiert und einige Änderungen vorgenommen haben, sieht das Ergebnis möglicherweise wie folgt aus.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "elbMyLB": {
      "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties": {
        "AvailabilityZones": [
          "us-east-1b",
          "us-east-1c"
        ]
      }
    }
  }
}
```

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Erstellen einer AWS CloudFormation-Vorlage

```
    ],
    "HealthCheck": {
      "HealthyThreshold": "2",
      "Interval": "30",
      "Target": "HTTP:80/",
      "Timeout": "5",
      "UnhealthyThreshold": "2"
    },
    "Listeners": [
      {
        "InstancePort": "80",
        "LoadBalancerPort": "80",
        "Protocol": "HTTP",
        "PolicyNames": [
        ]
      }
    ]
  }
},
"asgMyAutoScalingGroup": {
  "Type": "AWS::AutoScaling::AutoScalingGroup",
  "Properties": {
    "AvailabilityZones": [
      "us-east-1b",
      "us-east-1c"
    ],
    "Cooldown": "300",
    "DesiredCapacity": "1",
    "MaxSize": "1",
    "MinSize": "1",
    "LaunchConfigurationName": {
      "Ref": "lcMyLC"
    },
    "LoadBalancerNames": [
      {
        "Ref": "elbMyLB"
      }
    ]
  }
},
"lcMyLC": {
  "Type": "AWS::AutoScaling::LaunchConfiguration",
  "Properties": {
    "ImageId": "ami-45b77f2c",
    "InstanceType": "t1.micro",
    "KeyName": "mykeypair",
    "SecurityGroups": [
      {
        "Ref": "sgwebappsecuritygroup"
      }
    ]
  }
},
"aspMyScaleUpPolicy" : {
```

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Erstellen einer AWS CloudFormation-Vorlage

```
"Type" : "AWS::AutoScaling::ScalingPolicy",
"Properties" : {
  "AdjustmentType" : "ChangeInCapacity",
  "AutoScalingGroupName" : { "Ref" : "asgMyAutoScalingGroup" },
  "Cooldown" : "300",
  "ScalingAdjustment" : "1"
}
},

"cwCPUAlarmHigh": {
  "Type": "AWS::CloudWatch::Alarm",
  "Properties": {
    "AlarmDescription": "Scale-up if CPU > 60% for 10 minutes",
    "MetricName": "CPUUtilization",
    "Namespace": "AWS/EC2",
    "Statistic": "Average",
    "Period": "300",
    "EvaluationPeriods": "2",
    "Threshold": "60",
    "AlarmActions": [ { "Ref": "aspMyScaleUpPolicy" } ],
    "Dimensions": [
      {
        "Name": "AutoScalingGroupName",
        "Value": { "Ref": "asgMyAutoScalingGroup" }
      }
    ],
    "ComparisonOperator": "GreaterThanThreshold"
  }
},

"rdsmydbinstance": {
  "Type": "AWS::RDS::DBInstance",
  "Properties": {
    "AllocatedStorage": "5",
    "BackupRetentionPeriod": "1",
    "DBInstanceClass": "db.m1.small",
    "DBName": "MyDatabase",
    "DBParameterGroupName": "default.mysql5.1",
    "Engine": "mysql",
    "EngineVersion": "5.1.57",
    "MasterUsername": "awsuser",
    "MasterUserPassword": "awsuser",
    "Port": "3306",
    "PreferredBackupWindow": "10:00-10:30",
    "PreferredMaintenanceWindow": "sun:05:00-sun:05:30",
    "MultiAZ": "true",
    "DBSecurityGroups": [
      {
        "Ref": "dbsgmydbsecuritygroup"
      }
    ]
  }
},

"sgwebappsecuritygroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "this is a security group for demo",
```

```
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": "80",
        "ToPort": "80",
        "SourceSecurityGroupName": "amazon-elb-sg",
        "SourceSecurityGroupOwnerId": "amazon-elb"
      },
      {
        "IpProtocol": "tcp",
        "FromPort": "22",
        "ToPort": "22",
        "CidrIp": "0.0.0.0/0"
      }
    ]
  },
},
}

"dbsgmydbsecuritygroup": {
  "Type": "AWS::RDS::DBSecurityGroup",
  "Properties": {
    "GroupDescription": "my database security group",
    "DBSecurityGroupIngress": [
      {
        "EC2SecurityGroupName": {
          "Ref": "sgwebappsecuritygroup"
        },
        "EC2SecurityGroupOwnerId": "123456789012"
      }
    ]
  }
},
}

>Description": ""
}
```

Sie sollten jetzt einige Änderungen an dieser Vorlage vornehmen, bevor Sie Ihre neue Umgebung starten. In diesem Tutorial haben Sie nur eine Amazon EC2-Instance gestartet. Es ist allerdings eine bewährte Methode, mehrere Instances über mehrere Availability Zones zu starten; daher sollten Sie Ihre Auto Scaling-Gruppe aktualisieren, um mehrere Instances zu starten. Sie sollten auch eine neue Umgebung mit Ihrem benutzerdefinierten AMI starten. Schließlich sollten Sie Ihre Datenbankinformationen so ändern, dass sie Namen und Passwort Ihrer Datenbank beinhalten.

Ändern einer CloudFormation-Vorlage

Nachdem jetzt die Vorlage erstellt wurde, wollen wir sie so ändern, dass Sie eine neue Umgebung mit dem benutzerdefinierten AMI starten können, sodass Sie mehrere Instances über mehrere Availability Zones verteilen können.

Starten Sie einen neuen Stapel mit einer geänderten Vorlage wie folgt:

1. Öffnen Sie die Vorlage, die Sie mit CloudFormer erstellt haben.
2. Ändern Sie Min Size, Max Size und Desired Capacity in der Gruppe Auto Scaling in 2.

```
"asgMyAutoScalingGroup": {
  "Type": "AWS::AutoScaling::AutoScalingGroup",
  "Properties": {
    "AvailabilityZones": [
      "us-east-1b",
      "us-east-1c"
    ],
    "Cooldown": "300",
    "DesiredCapacity": "2",
    "MaxSize": "2",
    "MinSize": "2",
    "LaunchConfigurationName": {
      "Ref": "lcMyLC"
    },
    "LoadBalancerNames": [
      {
        "Ref": "elbMyLB"
      }
    ]
  }
},
```

3. Ändern Sie die Image ID in der Gruppe Launch Configuration in das benutzerdefinierte AMI, das Sie in [Schritt 10: Erstellen eines benutzerdefinierten Amazon-Computerabbilds \(AMI\) \(p. 47\)](#) erstellt haben.



Note

Ihre AMI ID wird sich von der unterscheiden, die Sie unten sehen.

Fügen Sie die UserData-Informationen wie unten gezeigt hinzu, damit der Web-Server beim Booten gestartet wird.



Note

Stellen Sie sicher, dass das Komma direkt nach] und unmittelbar vor `UserData` steht.

```
"lcMyLC": {
  "Type": "AWS::AutoScaling::LaunchConfiguration",
  "Properties": {
    "ImageId": "ami-91b270f8",
    "InstanceType": "t1.micro",
    "KeyName": "mykeypair",
    "SecurityGroups": [
      {
        "Ref": "sgwebappsecuritygroup"
      }
    ],
    "UserData" : { "Fn::Base64" : { "Fn::Join" : [",", [
      "#!/bin/bash -v\n",
      "sed -i 's/AllowOverride None/AllowOverride All/g' /etc/httpd/conf/ht
      tpd.conf\n",
    ]
    }
  }
},
```

```
    "service httpd start\n"  
  ]]]  
}  
,
```

4. Aktualisieren Sie die folgenden Parameter in der Gruppe Database.

- Ändern Sie DBName in **awsuser**.
- Ändern Sie MasterUserPassword in **mypassword**.

```
"mydbinstance": {  
  "Type": "AWS::RDS::DBInstance",  
  "Properties": {  
    "AllocatedStorage": "5",  
    "BackupRetentionPeriod": "1",  
    "DBInstanceClass": "db.m1.small",  
    "DBName": "mydb",  
    "DBParameterGroupName": "default.mysql5.1",  
    "Engine": "mysql",  
    "EngineVersion": "5.1.57",  
    "MasterUsername": "awsuser",  
    "MasterUserPassword": "mypassword",  
    "Port": "3306",  
    "PreferredBackupWindow": "08:30-09:00",  
    "PreferredMaintenanceWindow": "fri:03:30-fri:04:00",  
    "MultiAZ": "true",  
    "DBSecurityGroups": [  
      {  
        "Ref": "dbsgmydbsecuritygroup"  
      }  
    ]  
  }  
},
```

Nachdem Sie jetzt die Vorlage geändert haben, fahren wir im nächsten Schritt damit fort, Ihre neue Umgebung mithilfe Ihrer Vorlage zu starten.

Erstellen eines AWS CloudFormation-Stapels

Nachdem Sie nun Ihre AWS CloudFormation-Vorlage geändert haben, wollen wir einen neuen Stapel erstellen, um Ihre neue Umgebung zu starten. Bevor Sie Ihren neuen Stapel starten, können Sie prüfen, ob er funktioniert, indem Sie all Ihre AWS-Ressourcen bereinigen, *außer* Ihrem Schlüsselpaar und benutzerdefinierten AMI. Anleitungen zur Bereinigung Ihrer Ressourcen finden Sie unter [Schritt 12: Bereinigen](#) (p. 58).



Note

AWS CloudFormation ist ein kostenloser Service. Allerdings werden Ihnen die AWS-Ressourcen, die Sie in Ihren Stapeln verwenden, zu den jeweils aktuellen Tarifen in Rechnung gestellt. Weitere

Informationen über AWS-Preise finden Sie auf der Detailseite für jedes Produkt unter <http://aws.amazon.com/pricing>.

Erstellen Sie einen AWS CloudFormation-Stapel wie folgt:

1. Starten Sie den Assistenten "Create Stack".
 - a. Öffnen Sie die AWS CloudFormation-Konsole unter <https://console.aws.amazon.com/cloudformation/>.
 - b. Stellen Sie sicher, dass US East (N. Virginia) in der Regionenauswahl in der Navigationsleiste ausgewählt ist.
 - c. Klicken Sie auf Create Stack.

2. Wählen Sie eine Vorlage aus.
 - a. Geben Sie auf der Seite SELECT TEMPLATE des Assistenten Create Stack einen Namen für den Stapel in das Feld Stack Name ein.
 - b. Klicken Sie auf Upload a Template URL und geben Sie den Speicherort ein, an dem Sie Ihre geänderte Vorlage gespeichert haben.
 - c. Klicken Sie auf Show Advanced Options.
 - d. Wählen Sie Create a new SNS topic im Dropdown-Menü Amazon SNS Topic aus. Sie erhalten Benachrichtigungen per E-Mail, wenn Ressourcen erstellt und wenn sie gelöscht werden.
 - e. Geben Sie CRM in das Feld New topic name ein.
 - f. Geben Sie Ihre E-Mail-Adresse in das Feld Email ein und übernehmen Sie die übrigen Standardwerte.
 - g. Klicken Sie auf Continue.

Create Stack Cancel

SELECT TEMPLATE SPECIFY PARAMETERS ADD TAGS REVIEW

AWS CloudFormation gives you an easier way to create a collection of related AWS resources (a stack) by describing your requirements in a template. To create a stack, fill in the name for your stack and select a template. You may choose one of the sample templates to get started quickly, or one of your own templates stored in S3 or on your local hard drive.

Stack Name:
MyCloudFormationStack

Template:

Use a sample template

Upload a Template File
C:\MyFolder\MyModifie

Provide a Template URL

Show Advanced Options

Notifications (optional):
Amazon SNS Topic Create a new SNS topic
New topic name: CRM Email: inedoe@example.com
Creation Timeout (minutes): none
Rollback on Failure: Yes No

3. Klicken Sie auf der Seite "Add Tags" auf Continue.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Erstellen eines AWS CloudFormation-Stapels

- Überprüfen Sie die Einstellungen. Sie können die Einstellungen ändern, indem Sie auf den Link "Edit" für einen bestimmten Schritt klicken.

Create Stack Cancel

SELECT TEMPLATE SPECIFY PARAMETERS ADD TAGS REVIEW

Please review the information below, then click Continue to create the stack.

Stack Information Edit Stack

Stack Name: MyCloudFormationStack

Stack Description:

Template: https://cf-templates-111111111111-us-east-1.s3.amazonaws.com/2013087G9Y-cloudformer.template

IAM Acknowledgement: false

Estimated Cost: Cost

Notification Edit Notification

Notification: CRM

Creation Timeout (minutes): none

Rollback on Failure: true

< Back **Continue** >

- Klicken Sie auf Continue.

Ein Bestätigungsfenster wird geöffnet.

Create Stack Cancel

Your stack is being created. You can select the stack and, using the tabs shown below, you can see the current state and track the progress of the stack creation. When your stack is created, it will have a status of CREATE_COMPLETE. The Events tab will show you what is currently being created and will also contain any error messages if there are problems creating your stack.

1 Stack selected

Stack: USEast1

Description Outputs Resources **Events** Template Parameters

Stack Events

Time	Type
2011-02-16 09:21 PST	AWS::Stack
2011-02-16 09:21 PST	AWS::AutoScaling::AutoScalingGroup
2011-02-16 09:20 PST	AWS::AutoScaling::AutoScalingGroup
2011-02-16 09:20 PST	AWS::AutoScaling::LaunchConfiguration
2011-02-16 09:20 PST	AWS::AutoScaling::LaunchConfiguration

Close >

- Klicken Sie auf Close.

Das Bestätigungsfenster wird geschlossen und Sie kehren zur Seite CloudFormation zurück. Ihre neue AWS CloudFormation-Vorlage wird in der Liste mit dem Status CREATE_IN_PROGRESS angezeigt.

CloudFormation Stacks (Showing 5 of 5)

Create Stack Update Stack Delete Stack Viewing: Active Show/Hide Refresh

	Name	Created	Status	Description
<input checked="" type="checkbox"/>	MyCloudFormationStack	2013-03-28 14:24:37 UTC-7	CREATE_IN_PROGRESS	



Note

Es dauert einige Minuten, bis Ihr Stapel erstellt ist. Klicken Sie auf der Seite "Stacks" auf Refresh, um zu sehen, ob die Vorlage erfolgreich erstellt wurde.

Nachdem Ihr Stapel erstellt wurde, können Sie überprüfen, ob er funktioniert.

Überprüfen Sie den Stapel "AWS CloudFormation" wie folgt:

1. Verbinden Sie sich mit einer der neu erstellten Amazon EC2-Instances wie unter [Herstellen einer Verbindung mit einer Amazon EC2-Instance auf einem Linux/UNIX-Computer mithilfe eines eigenständigen SSH-Clients \(p. 38\)](#) oder [Herstellen einer Verbindung mit einer Amazon EC2-Instance von Windows mithilfe von PuTTY \(p. 33\)](#) beschrieben.
2. Wechseln Sie zur Amazon RDS-Datenbank (Amazon Relational Database Service).
 - a. Navigieren Sie zur Amazon RDS-Konsole und rufen Sie Ihren neuen Amazon RDS-Endpunkt ab, der von AWS CloudFormation erstellt wurde. Befolgen Sie das gleiche Verfahren wie unter [Schritt 9: Bereitstellen Ihrer Anwendung \(p. 31\)](#).
 - b. Geben Sie im SSH-Fenster an der Eingabeaufforderung den folgenden Befehl ein.

```
cd
```

- c. Geben Sie an der Eingabeaufforderung den folgenden Befehl ein.

```
mysql -u awsuser -pmypassword --database=mydb --host=your Amazon RDS endpoint < backups/backup.sql
```

Stellen Sie sicher, dass Sie den Amazon RDS-Endpunkt durch Ihren Amazon RDS-Endpunkt ersetzen. Dies ist der Endpunkt, den Sie im vorherigen Schritt abgerufen haben.

- d. Geben Sie an der Eingabeaufforderung den folgenden Befehl ein.

```
cd /var/www/html/sites/default
```

- e. Geben Sie an der Eingabeaufforderung den folgenden Befehl ein, um den Inhalt des Verzeichnisses aufzulisten.

```
ls
```

- f. Geben Sie an der Eingabeaufforderung den folgenden Befehl ein, um die Datei "settings.php" zu öffnen.

```
vi settings.php
```

- g. Navigieren Sie mit der Bild ab-Taste zu einem Abschnitt, der ähnlich wie der folgende aussieht.

```
$databases = array (  
  'default' =>  
    array (  

```

```
'default' =>
array (
  'database' => 'mydb',
  'username' => 'awsuser',
  'password' => 'mypassword',
  'host' => 'mydbinstance.cgwxy4t1e0xb.us-east-1.rds.amazonaws.com',

  'port' => '',
  'driver' => 'mysql',
  'prefix' => '',
),
),
);
```

- h. Drücken Sie die i-Taste, um in den Einfügemodus zu wechseln.
- i. Ersetzen Sie `mydbinstance.cgwxy4t1e0xb.us-east-1.rds.amazonaws.com` durch Ihren neuen Amazon RDS-Endpunkt.



Note

Klicken Sie mit der rechten Maustaste, um den Inhalt einzufügen, wenn Sie es vorziehen, mit Kopieren/Einfügen zu arbeiten.

```
$databases = array (
'default' =>
  array (
    'default' =>
      array (
        'database' => 'mydb',
        'username' => 'awsuser',
        'password' => 'mypassword',
        'host' => 'mycloudformationstack-rdsmydbinstance-1276gxy2eians.cg
wxy4t1e0xb.us-east-1.rds.amazonaws.com',
        'port' => '',
        'driver' => 'mysql',
        'prefix' => '',
      ),
    ),
  ),
);
```

- j. Drücken Sie Esc und dann `:wq`, um die Datei zu speichern und das Fenster zu verlassen.
- k. Führen Sie die gleichen Schritte für die andere Amazon EC2-Instance aus.
- l. Überprüfen Sie, dass alles ordnungsgemäß funktioniert, indem Sie zur Amazon EC2-Konsole navigieren und den DNS-Namen für den neuen Load Balancer abrufen, der von Ihrem Amazon CloudFormation-Stapel erstellt wurde, wie Sie es bereits unter [Schritt 9: Bereitstellen Ihrer Anwendung \(p. 31\)](#) gemacht haben.

Aktueller Stand

Herzlichen Glückwunsch! Sie haben gerade Ihre neue Umgebung mithilfe der AWS-Ressourcen, die Sie in diesem Tutorial erstellt haben, gestartet. Ihr Elastic Load Balancer zeigt jetzt auf Ihre beiden Amazon EC2-Instances über mehrere Availability Zones hinweg.

Löschen Sie einen AWS CloudFormation-Stapel wie folgt:

1. Öffnen Sie die AWS CloudFormation-Konsole unter <https://console.aws.amazon.com/cloudformation/>.
2. Klicken Sie auf den Stapel, den Sie löschen möchten, und dann auf Delete Stack.

Schritt 12: Bereinigen

Topics

- [Beenden Sie Ihre Amazon EC2-Instances in Ihrer Auto Scaling-Gruppe \(p. 58\)](#)
- [Kündigen der DB-Instance \(p. 60\)](#)
- [Löschen eines CloudWatch-Alarms \(p. 61\)](#)
- [Löschen des Elastic Load Balancers \(p. 61\)](#)
- [Löschen eines Schlüsselpaars \(p. 62\)](#)
- [Löschen einer Amazon EC2-Sicherheitsgruppe \(p. 62\)](#)
- [Delete Your Custom AMI \(p. 63\)](#)

Herzlichen Glückwunsch! Sie haben soeben Ihre Webanwendung bereitgestellt. Um das Anfallen weiterer Gebühren zu vermeiden, beenden Sie jetzt Ihre Umgebungen und bereinigen Sie Ihre Ressourcen.

Beenden Sie Ihre Amazon EC2-Instances in Ihrer Auto Scaling-Gruppe

In diesem Abschnitt werden Sie zunächst die Amazon EC2-Instance entfernen, dann die Auto Scaling-Gruppe und schließlich die Startkonfiguration löschen.

Sie müssen alle Amazon EC2-Instances in einer Auto Scaling-Gruppe beenden, bevor Sie die Gruppe löschen können. Sie können alle Instances in einer Gruppe einfach beenden, indem Sie die Gruppe so aktualisieren, dass sowohl die minimale als auch die maximale Größe auf Null eingestellt ist.

Entfernen Sie die Amazon EC2-Instance aus der Auto Scaling-Gruppe wie folgt:

1. Öffnen Sie ein Befehlszeilenfenster: Klicken Sie bei einem Windows-Computer auf Start. Geben Sie in das Suchfeld `cmd` ein und drücken Sie dann die Eingabetaste.
2. Verwenden Sie den Befehl `as-update-auto-scaling-group`, um die Auto Scaling-Gruppe zu aktualisieren, die wir vorher erstellt hatten. Geben Sie an der Eingabeaufforderung Folgendes ein und drücken Sie anschließend die Eingabetaste:

```
PROMPT>as-update-auto-scaling-group MyAutoScalingGroup --min-size 0 --max-size 0
```

Auto Scaling gibt Folgendes zurück:

```
OK-Updated AutoScalingGroup
```

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Beenden Sie Ihre Amazon EC2-Instances in Ihrer Auto
Scaling-Gruppe

3. Verwenden Sie jetzt den Befehl `as-describe-auto-scaling-groups`, um zu bestätigen, dass Auto Scaling die Instance aus `MyAutoScalingGroup` entfernt hat.

Es kann ein paar Minuten dauern, bis die Instance beendet wird, daher müssen Sie den Status eventuell mehr als einmal überprüfen. Geben Sie an der Eingabeaufforderung Folgendes ein und drücken Sie anschließend die Eingabetaste:

```
PROMPT>as-describe-auto-scaling-groups MyAutoScalingGroup --headers
```

Wenn die Beendigung der Instance noch nicht abgeschlossen ist, gibt Auto Scaling Informationen ähnlich der Folgenden zurück. (Ihr Wert für `INSTANCE-ID` wird abweichen):

```
AUTO-SCALING-GROUP  GROUP-NAME          LAUNCH-CONFIG  AVAILABILITY-ZONES
LOAD-BALANCERS     MIN-SIZE  MAX-SIZE  DESIRED-CAPACITY
AUTO-SCALING-GROUP  MyAutoScalingGroup  MyLC          us-east-1b,us-east-
1c  MyLB          0          0          0
INSTANCE  INSTANCE-ID  AVAILABILITY-ZONE  STATE      STATUS  LAUNCH-CONFIG
INSTANCE  i-xxxxxxx   us-east-1c        InService  Healthy  MyLC
```



Note

Sie können auch auf Instances in der Amazon EC2-Konsole klicken, um den Status Ihrer Instances anzuzeigen.

Wenn keine Instances in `MyAutoScalingGroup` vorhanden sind, können Sie die Gruppe löschen.

Löschen Sie die Auto Scaling-Gruppe wie folgt:

- Geben Sie an der Eingabeaufforderung Folgendes ein und drücken Sie anschließend die Eingabetaste:

```
PROMPT>as-delete-auto-scaling-group MyAutoScalingGroup
```

Um das Löschen zu bestätigen, geben Sie `Y` ein und drücken Sie dann die Eingabetaste.

```
Are you sure you want to delete this MyAutoScalingGroup? [Ny]
```

Auto Scaling gibt Folgendes zurück:

```
OK-Deleted MyAutoScalingGroup
```

Jetzt müssen Sie nur noch die Startkonfiguration löschen, die Sie für diese Auto Scaling-Gruppen erstellt haben.

Löschen Sie die Startkonfiguration wie folgt:

- Geben Sie an der Eingabeaufforderung Folgendes ein und drücken Sie anschließend die Eingabetaste:

```
PROMPT>as-delete-launch-config MyLC
```

Um das Löschen zu bestätigen, geben Sie `Y` ein und drücken Sie dann die Eingabetaste.

```
Are you sure you want to delete this launch configuration? [Ny]
```

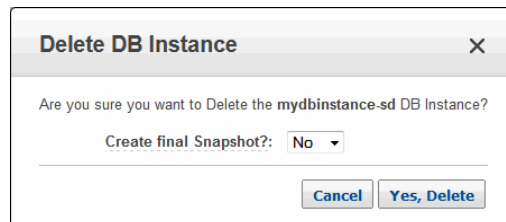
Auto Scaling gibt Folgendes zurück:

```
OK-Deleted launch configuration
```

Kündigen der DB-Instance

Kündigen Sie Ihre DB-Instance wie folgt:

1. Öffnen Sie die Amazon RDS-Konsole unter <https://console.aws.amazon.com/rds/>.
2. Stellen Sie sicher, dass US East (N. Virginia) in der Regionenauswahl in der Navigationsleiste ausgewählt ist.
3. Klicken Sie im linken Navigationsbereich auf DB Instances.
4. Finden Sie die DB-Instance in Ihrer Liste von DB-Instances.
5. Aktivieren Sie das Kontrollkästchen neben der DB-Instance und wählen Sie dann Delete aus der Dropdown-Liste Instance Actions oben auf der Seite.
6. Wählen Sie No in der Dropdown-Liste Create final snapshot?.



Wenn dies keine Übung wäre, könnten Sie noch einen letzten Snapshot erstellen, bevor Sie die DB-Instance löschen, sodass Sie die DB-Instance später wiederherstellen könnten.



Note

Die Erstellung eines letzten Snapshots verursacht zusätzliche Kosten für die Speicherung.

7. Klicken Sie auf Yes, Delete.

Amazon RDS beginnt, die Instance zu beenden. Sobald der Status der DB-Instance auf deleted wechselt, fallen für diese DB-Instance keine Gebühren mehr an.

Löschen Sie Ihre DB-Sicherheitsgruppe wie folgt:

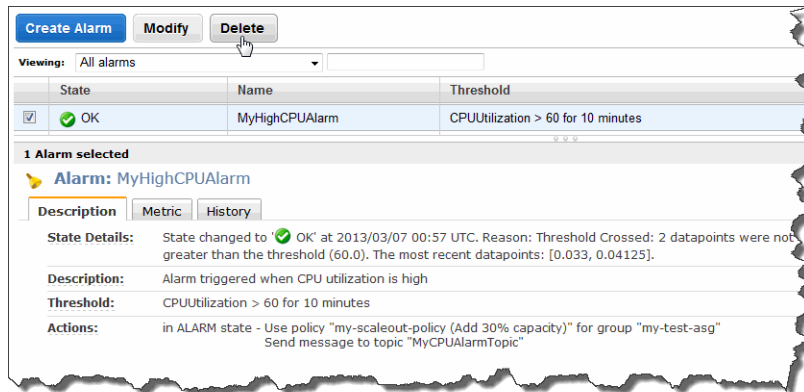
1. Öffnen Sie die Amazon RDS-Konsole unter <https://console.aws.amazon.com/rds/>.
2. Klicken Sie im linken Navigationsbereich auf DB Security Groups.
3. Wählen Sie eine DB-Sicherheitsgruppe aus und klicken Sie auf Delete DB Security Group.
4. Klicken Sie auf Yes, Delete.

Löschen eines CloudWatch-Alarms

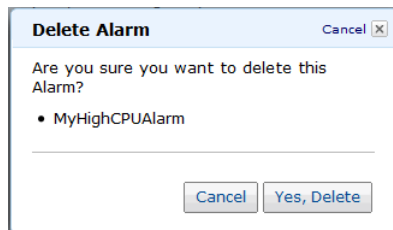
Wenn Sie entscheiden, dass Sie den Alarm nicht mehr benötigen, können Sie ihn löschen.

Löschen Sie den Alarm wie folgt:

1. Öffnen Sie die Amazon CloudWatch-Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Klicken Sie in der oberen Navigationsleiste in der Regionenauswahl auf US East (N. Virginia).
3. Klicken Sie im linken Navigationsbereich auf Alarms.
4. Aktivieren Sie das Kontrollkästchen neben dem zu löschenden Alarm und klicken Sie dann auf Delete.



5. Klicken Sie in der eingblendeten Bestätigungsmeldung auf Yes, Delete.



Löschen des Elastic Load Balancers

Sobald Ihr Load Balancer verfügbar ist, berechnet Ihnen AWS jede ganze oder angebrochene Stunde, in der der Load Balancer läuft. Wenn Sie entscheiden, dass Sie den Load Balancer nicht mehr benötigen, können Sie ihn löschen.

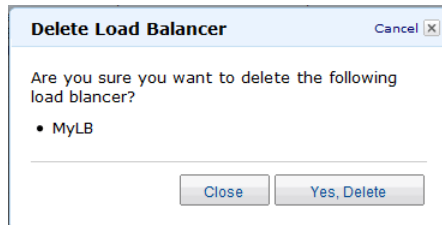
Löschen Sie den Load Balancer wie folgt:

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie in der oberen Navigationsleiste in der Regionenauswahl auf US East (N. Virginia).
3. Klicken Sie im linken Navigationsbereich auf Load Balancers.
4. Aktivieren Sie das Kontrollkästchen neben dem zu löschenden Load Balancer und klicken Sie dann auf Delete.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Löschen eines Schlüsselpaars



5. Klicken Sie in der eingeblendeten Bestätigungsmeldung auf Yes, Delete.



Elastic Load Balancing löscht den Load Balancer. Sobald der Load Balancer gelöscht ist, fallen keine weiteren Kosten für diesen Load Balancer mehr an.



Caution

Auch nachdem Sie einen Load Balancer gelöscht haben, laufen die Amazon EC2-Instances weiter, die dem Load Balancer zugeordnet sind. Während die Amazon EC2-Instances laufen, entstehen Ihnen weitere Kosten.

Löschen eines Schlüsselpaars

Dieser Schritt ist optional. Ihnen wird die Beibehaltung eines Schlüsselpaars nicht berechnet. Sie können das Schlüsselpaar zu einem späteren Zeitpunkt wiederverwenden.

Löschen Sie Schlüsselpaare wie folgt:

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie in der oberen Navigationsleiste in der Regionenauswahl auf US East (N. Virginia).
3. Klicken Sie im linken Navigationsbereich auf Key Pairs.
4. Aktivieren Sie das Kontrollkästchen neben dem zu löschenden Schlüsselpaar und klicken Sie dann auf Delete.
5. Klicken Sie in der eingeblendeten Bestätigungsmeldung auf Yes.

Löschen einer Amazon EC2-Sicherheitsgruppe

Löschen Sie eine Sicherheitsgruppe wie folgt:

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie in der oberen Navigationsleiste in der Regionenauswahl auf US East (N. Virginia).
3. Klicken Sie im linken Navigationsbereich auf Security Groups.
4. Wählen Sie im Bereich "Details" unter Security Groups eine Sicherheitsgruppe aus, die Sie löschen möchten, und klicken Sie dann auf Delete.
5. Klicken Sie auf Yes, Delete.

Delete Your Custom AMI

To delete an AMI and a snapshot

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Make sure US East (N. Virginia) is selected in the region selector in the navigation bar.
3. In the left navigation pane, click AMIs.
4. Right-click your AMI, and select Deregister. When prompted, click Continue.

The image is deregistered, which means it is deleted and can no longer be launched.

5. Go to the Snapshots page, right-click the snapshot, and select Delete Snapshot. When prompted, click Yes, Delete.

The snapshot is deleted.

Amazon Route 53

Amazon Route 53 ist ein skalierbarer Domain Name System (DNS)-Webservice. Er bietet sicheres und zuverlässiges Routing in Ihre Infrastruktur, die Amazon Web Services (AWS)-Produkte verwendet, wie etwa Amazon Elastic Compute Cloud (Amazon EC2), Elastic Load Balancing oder Amazon Simple Storage Service (Amazon S3). Sie können Amazon Route 53 auch dafür verwenden, um Benutzer auf Ihre Infrastruktur außerhalb von AWS zu leiten.

Amazon Route 53 leitet Anforderungen automatisch zu dem nächsten DNS-Server in einem globalen Netzwerk von DNS-Servern um, was zu geringer Latenz führt. Er ist ein autoritatives DNS-System, das bedeutet, dass er freundliche Domainnamen wie `www.example.com` in IP-Adressen wie `192.0.2.1` übersetzt.

Sie können Ihre DNS-Datensätze über die Amazon Route 53-API verwalten oder Benutzer- und Zugriffsmanagement auf Kontoebene über die Identity and Access Management (IAM)-API einstellen. Wie bei den anderen AWS-Produkten gibt es auch für die Nutzung von Amazon Route 53 keine Verträge oder Mindestverpflichtungen. Sie zahlen nur für die Domains, die Sie konfigurieren, und für die Anzahl der Abfragen, die der Service beantwortet. Weitere Preisinformationen zu Amazon Route 53 erhalten Sie unter [Amazon Route 53 Pricing](#).

Das folgende Verfahren erklärt die High-Level-Schritte, die Sie unternehmen müssen, um Route 53 für dieses Beispiel zu verwenden. Anleitungen dazu, wie Sie diese vier Schritte durchführen, finden Sie unter [Handbuch "Erste Schritte" für Amazon Route 53](#). Weitere Informationen darüber, wie Sie einen Alias im letzten Schritt erstellen, finden Sie unter [How to Create an Alias Record Set](#) in *Entwicklerhandbuch für Amazon Route 53*.

Verwenden Sie Amazon Route 53 wie folgt:

1. Erstellen Sie eine gehostete Zone für `example.com`.
2. Erstellen Sie einen neuen DNS-Datensatz für Ihre statischen Inhalte (z. B. `static.example.com`), der auf Ihre CloudFront-Verteilung zeigt (z. B. `d18k4jybr69gw2.cloudfront.net`).
3. Erstellen Sie einen neuen DNS-Datensatz für Ihre Website (z. B. `www.example.com`), der auf Ihren Elastic Load Balancer CNAME zeigt.
4. Bestätigen Sie, dass Ihre Anforderungen vollständig sind.
5. Aktualisieren Sie die Datensätze des Namensservers der Vergabestelle.
6. Erstellen Sie einen Alias für Ihren Load Balancer, der auf Anforderungen für `example.com` und `www.example.com` reagiert. Sie verwenden die gehostete Zonen-ID für den Load Balancer (z. B.: `Z3DZXE0Q79N41H`).

Amazon CloudFront

[Amazon CloudFront](#) ist ein Service für die Bereitstellung von Inhalten von Amazon Web Services, der Ihnen hilft, die Leistung, Zuverlässigkeit und Verfügbarkeit Ihrer Websites und Anwendungen zu verbessern. Der Inhalt, den Sie mit Amazon CloudFront bereitstellen, wird auf einem Server gespeichert, der als *Ursprungs-Server* bezeichnet wird. Amazon CloudFront verteilt Ihre Webinhalte (beispielsweise Bilder, Videos usw.) über ein Netzwerk von Edge-Standorten weltweit. Die Inhalte werden aus Ihrem konfigurierten Amazon S3-Bucket oder dem Ursprungs-Server zunächst an den Edge-Standort geleitet, der dem Anforderungsenden Benutzer am nächsten ist. In diesem Beispiel würden wir für die Verwendung von Amazon CloudFront unsere statischen Inhalte (Bilder, HTML usw.) in einem Amazon S3-Bucket speichern und dann eine CloudFront-Verteilung von unserem S3-Bucket erstellen. Nach der Erstellung unserer CloudFront-Verteilung ändern wir den Code einfach dahingehend, dass unsere statischen Inhalte auf die CloudFront-Verteilung zeigen. Weitere Informationen dazu erhalten Sie unter [Start Using CloudFront with Amazon S3](#) in *Handbuch "Erste Schritte" für Amazon CloudFront*.

Preise

Topics

- [Amazon EC2-Kostenaufschlüsselung \(p. 66\)](#)
- [Amazon RDS-Kostenaufschlüsselung \(p. 69\)](#)
- [Summe aller Kosten \(p. 71\)](#)

[AWS – Einfacher Monatsrechner](#) schätzt Ihre monatliche Rechnung. Der Rechner bietet eine Kostenaufschlüsselung pro Service sowie eine Schätzung der monatlichen Gesamtkosten. Sie können das Tool auch verwenden, um eine Kostenschätzung und -aufschlüsselung für häufige Lösungen zu erhalten. Dieses Thema führt Sie durch ein Beispiel für die Verwendung von "AWS – Einfacher Monatsrechner" zur Schätzung Ihrer monatlichen Abrechnung für die Beispielanwendung, die wir gerade erstellt haben.



Note

Die AWS-Preisinformationen in dieser Dokumentation sind zum Zeitpunkt der letzten Aktualisierung dieses Dokuments aktuell. Eine Übersicht mit aktuellen Preisinformationen erhalten Sie unter [AWS Service Pricing Overview](#).

Amazon EC2-Kostenaufschlüsselung

In der folgenden Tabelle sind die Merkmale für Amazon EC2 angegeben, die wir für die Hosting-Architektur dieser Webanwendung ermittelt haben.

Merkmal	Metrik	Beschreibung
Betriebszeit	24 Stunden/Tag	Unter der Annahme, der Monat hat im Durchschnitt 30,5 Tage, läuft die Instance 732 Stunden/Monat.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Amazon EC2-Kostenaufschlüsselung

Merkmal	Metrik	Beschreibung
Computermerkmale	ti.micro-Instance	613 MB Arbeitsspeicher, bis zu 2 ECUs (für kurze periodische Bursts), nur EBS-Speicher, 32-Bit- oder 64-Bit-Plattform Eine Liste mit Instance-Typen finden Sie unter http://aws.amazon.com/ec2/instance-types/ .
Zusätzlicher Speicher	1 EBS-Volume Speicher: 10 GB/Monat 100 IOPS	Das AMI ist EBS-gestützt, das Volume verfügt über 10 GB bereitgestellten Speicher und 100 E/A-Anforderungen werden pro Sekunde an das Volume gesendet.
Datenübertragung	Eingehende Daten: 0,005 GB/Tag Ausgehende Daten: 0,05 GB/Tag	Es sind annähernd 1 000 Zugriffe pro Tag und jede Antwort umfasst etwa 50 KB und jede Anforderung etwa 5 KB.
Skalierung der Instances	2	Im Durchschnitt laufen an einem gegebenen Tag 2 Instances.
Elastic Load Balancing	Nutzung in Stunden: 732 Stunden/Monat Verarbeitete Daten: 1 525 GB/Monat	ELB ist 24 Stunden/Tag an 7 Tagen/Woche im Einsatz. ELB verarbeitet insgesamt 0,055 GB/Tag (ein- und ausgehende Daten).
Detaillierte Überwachung	3,50 USD pro Instance pro Monat	Wir haben für unsere Amazon EC2-Instances eine detaillierte Überwachung eingerichtet.

In der folgenden Abbildung ist die von "AWS – Einfacher Monatsrechner" ermittelte Kostenaufschlüsselung für Amazon EC2 dargestellt.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Amazon EC2-Kostenaufschlüsselung

Choose region: US-East / US Standard (Northern Virginia) Inbound Data Transfer is Free and Outbound Data Transfer is 1 GB free per region per month

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. Amazon Elastic Block Store (EBS) provides persistent storage to Amazon EC2 instances. Clear Form

Compute: Amazon EC2 On-Demand Instances:

Description	Instances	Usage	Instance Type	Operating System	Tenancy	Detailed Monitoring
	2	24 Hours/Day	Micro	Linux	Default	<input checked="" type="checkbox"/>

Compute: Amazon EC2 Reserved Instances:

Description	Instances	Usage	Instance Type	Operating System	Offering and Term	Tenancy	Detailed Monitoring
	0	0 Hours/Month	Small	Linux	Medium Utilization 3yr term	Default	<input type="checkbox"/>

Storage: Amazon EBS Volumes:

Description	Volumes	Volume Type	Storage	IOPS	Snapshot Storage
	1	Standard	10 GB	100	0 GB-month of Storage

Elastic IP:

Number of Additional Elastic IPs:

Elastic IP Non-attached Time: Hours/Month

Number of Elastic IP Remaps: Per Month

Data Transfer:

Inter-Region Data Transfer Out: GB/Month

Data Transfer Out: GB/Day

Data Transfer In: GB/Day

Intra-Region Data Transfer: GB/Month

Public IP/Elastic IP Data Transfer: GB/Month

Elastic Load Balancing:

Number of Elastic LBs:

Total Data Processed by all ELBs: GB/Day

Die monatlichen Gesamtkosten ergeben sich aus der Summe der Kosten für die aktiven Instances, die EBS-Volumes und E/A-Anforderungen, die Elastic Load Balancer, die von den Elastic Load Balancern verarbeiteten Daten und Amazon CloudWatch-Metriken.

Variable	Formel	Berechnung
Instance-Kosten	Instance-Kosten pro Stunde	0,02 USD
	Anzahl der Instances	2
	x Betriebszeit in Stunden	x 732
	-----	-----
		29,28 USD
Zusätzlicher Speicher	Speichergebühr x Speichermenge (GB)	0,10 x 10 USD
	+ (E/A-Anforderungsrate x Sekunden pro Monat x Anforderungsrate (pro 1M Anforderungen))	+ (100 x ~2,6M x 0,10 USD)/1M
	-----	-----
	-----	27,35 USD

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Amazon RDS-Kostenaufschlüsselung

Variable	Formel	Berechnung
Elastic Load Balancing	Genutzte Stunden x stündliche Gebühr + (verarbeitete Daten (GB) x Verarbeitungsgebühr) -----	732 x 0,025 USD + 1,6775 x 0,008 USD ----- 18,31 USD
Amazon CloudWatch	Anzahl der Instances x Detaillierte Überwachungsrate -----	2 x 3,50 USD ----- 7,00 USD

Wir verwenden "AWS – Einfacher Monatsrechner" für diese Schätzung. Laut der Berechnungen des Rechners belaufen sich die Gesamtkosten für Amazon EC2 auf 81,94 USD.

Amazon RDS-Kostenaufschlüsselung

In der folgenden Tabelle sind die Merkmale für Amazon RDS angegeben, die wir für die Hosting-Architektur dieser Webanwendung ermittelt haben.

Merkmal	Metrik	Beschreibung
Betriebszeit	24 Stunden/Tag	24 Angenommen, der Monat hat im Durchschnitt 30,5 Tage, dann läuft die Instance 732 Stunden/Monat
Datenbankmerkmale	Kleine Amazon RDS-Instance	1,7 GB Arbeitsspeicher, 1 ECU (1 virtueller Kern mit 1 ECU), 64-Bit-Plattform, mittlere E/A-Kapazität
Bereitgestellter Speicher	5 GB/Monat	Amazon stellt 5 GB bis 1 TB zugewiesener Speicherkapazität für Ihre primären Datenbestände zur Verfügung.
Anforderungen	2 M IOP/Monat	Wir haben 1 000 Zugriffe pro Tag bei einer Rate von 5 IOP pro Zugriff auf die Seite. Angenommen, der Monat hat im Durchschnitt 30,5 Tage. Das sind insgesamt 152 500 E/A-Anforderungen pro Monat oder 1 M (gerundet auf die nächste Million), aber da die Schreib-E/A-Anforderungen sich verdoppeln, da die Daten auch auf die Standby-Instance repliziert werden, haben wir insgesamt 2 M.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Amazon RDS-Kostenaufschlüsselung

Merkmal	Metrik	Beschreibung
Art der Bereitstellung	Multi-AZ	Wir führen unsere Datenbank-Instance über mehrere Availability Zones hinweg aus.
Zusätzlicher Sicherungsspeicher	Keine	Wir verwenden bis zu der bereitgestellten Menge von 5 GB.
Datenübertragung	Eingehende Daten: 0 GB Ausgehende Daten: 0 GB	Es gibt keine Datenübertragung von RDS zum Internet.
Skalierung der Datenbank-Instance	1	Wir brauchen eine Datenbank-Instance.

In der folgenden Abbildung ist die von "AWS – Einfacher Monatsrechner" ermittelte Kostenaufschlüsselung für Amazon RDS dargestellt.

Choose region: US-East / US Standard (Northern Virginia) | Inbound Data Transfer is Free and Outbound Data Transfer is 1 GB free per region per month

Amazon RDS is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. Clear Form

Amazon RDS On-demand DB Instances:

Description	DB Instances	Usage	DB Engine and License	Class and Deployment	Storage	IOPS
	1	24 Hours/Day	MySQL	Small Multi-AZ	Provisioned 5 GB	01

Additional Backup Storage (Free backup storage up to 100% of provisioned Storage):

Backup Storage: 0 GB-month of Storage

Amazon RDS Reserved DB Instances:

Description	DB Instances	Usage	DB Engine and License	Class and Deployment	Offering and Term	Storage	IOPS
	0	0 Hours/Month	MySQL	Small Standard (Single-AZ)	Medium Utilization 3 yr term	Standard 20 GB	0

Data Transfer:

Data Transfer Out: 0 GB/Month

Data Transfer In: 0 GB/Month

Intra-Region Data Transfer: 0 GB/Month

Da wir keine Datenübertragung in beide Richtungen und keinen Sicherungsspeicher haben, sind die monatlichen Gesamtkosten die Summe der Kosten für die laufenden Instances, bereitgestellten Speicher und E/A-Anforderungen.

Variable	Formel	Berechnung
Instance-Kosten	Instance-Kosten pro Stunde	\$0.153
	Anzahl der Instances	1
	x Betriebszeit in Stunden	x 732
	-----	-----
		\$112.00

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Summe aller Kosten

Variable	Formel	Berechnung
Bereitgestellter Speicher	Speichertarif	\$0.20
	x Speichermenge (GB)	x 5
	-----	-----
		\$1.00
E/A-Anforderungen	E/A-Tarif	0,10 USD
	x Anzahl Anforderungen (Millionen)	x ~2
	-----	-----
		0,22 USD

Wir verwenden "AWS – Einfacher Monatsrechner" für diese Schätzung. Laut der Berechnungen des Rechners belaufen sich die Gesamtkosten für Amazon RDS auf 113,52 USD.

Summe aller Kosten

Um die Gesamtkosten für dieses Szenario zu berechnen, fügen wir die Kosten für Amazon EC2, Amazon RDS und der ausgehenden Datenübertragungen von RDS hinzu und ziehen dann sämtliche Rabatte ab, die im kostenlosen Nutzungskontingent von AWS gewährt werden.

Die gesamten ausgehenden Datenübertragungen von AWS setzen sich aus den aggregierten ausgehenden Datenübertragungen von Amazon EC2 und Amazon RDS zusammen. Für Amazon EC2 sind dies 0,05 GB pro Tag, was ungefähr 1,525 GB pro Monat entspricht. Für Amazon RDS wurden von AWS keine ausgehenden Daten übertragen. Da von den ausgehenden Daten bis zu 1 GB pro Monat kostenlos sind, verbleiben insgesamt 0,525 GB im Monat.

Variable	Formel	Berechnung
AWS-Datenübertragung	(eingehende Daten (GB) x Gebühr für eingehende Daten)	0,1525 x 0,00 USD
	+ (ausgehende Daten (GB) x Gebühr für ausgehende Daten)	+ (0,525) x 0,12 USD
	-----	-----
		0,06 USD

In der folgenden Abbildung ist ein Beispiel für Ihre geschätzten monatlichen Kosten dargestellt.

Erste Schritte mit AWS Hosten einer Webanwendung für
Linux
Summe aller Kosten

Services		Estimate of your Monthly Bill (\$ 160.95)	
Estimate of Your Monthly Bill			
<input checked="" type="checkbox"/> Show First Month's Bill (include all one-time fees, if any)			
<p>With AWS, You only pay for what you use. Below you will see an estimate of your monthly bill. Expand each line item to see cost breakout of each service. To save this bill and input values, click on 'Save and Share' button. To remove the service from the estimate, click on the red cross.</p>			
			Save and Share
<input type="checkbox"/> Amazon EC2 Service (US-East)			\$ 81.94
Compute:	\$	36.28	
Intra-Region Data Transfer:	\$	0.00	
EBS Volumes:	\$	1.00	
EBS IOPS:	\$	26.35	
EBS Snapshots:	\$	0.00	
Reserved Instances (One-time Fee):	\$	0.00	
Elastic IPs:	\$	0.00	
Elastic LBs:	\$	18.30	
Data Processed by Elastic LBs:	\$	0.01	
Dedicated Per Region Fee:	\$	0.00	
Inter-Region Data Transfer Out	\$	0.00	
<input type="checkbox"/> Amazon RDS Service (US-East)			\$ 113.52
<input type="checkbox"/> AWS Data Transfer In			\$ 0.00
<input type="checkbox"/> AWS Data Transfer Out			\$ 0.06
<input type="checkbox"/> AWS Support (Basic)			\$ 0.00
Free Tier Discount:	\$		-34.58
Total One-Time Payment:	\$		0.00
Total Monthly Payment:	\$		160.95

Die Gesamtkosten für diese Webanwendung werden auf monatlich 160,95 USD geschätzt, einschließlich der Rabatte für kostenlose Nutzungskontingente.

Verwandte Ressourcen

Die folgende Tabelle listet einige der AWS-Ressourcen auf, die Ihnen bei Ihrer Arbeit mit AWS nützlich sein werden.

Ressource	Beschreibung
AWS-Produkte und -Services	Informationen über die Produkte und Services, die AWS bietet.
AWS-Dokumentation	Offizielle Dokumentation für jedes AWS-Produkt, einschließlich Service-Einführungen, Service-Funktionen und API-Referenz.
AWS-Diskussionsforen	Community-basiertes Forum zur Diskussion technischer Fragen zu Amazon Web Services.
AWS Support	Die Homepage für AWS Support, mit Zugang zu unseren Diskussionsforen, technischen FAQs und AWS Support Center.
Kontakt	Dieses Formular ist <i>ausschließlich</i> für Fragen gedacht, die Ihr Konto betreffen. Technische Fragen stellen Sie bitte im Diskussionsforum.
AWS-Architekturzentrum	Bietet die erforderlichen Anleitungen und bewährte Methoden, um hochskalierbare und zuverlässige Anwendungen in der AWS-Cloud zu erstellen. Diese Ressourcen helfen Ihnen dabei, die AWS-Plattform, ihre Services und Funktionen zu verstehen. Sie bieten auch architektonische Anleitungen für Design und Implementierung von Systemen, die auf der AWS-Infrastruktur laufen.
AWS-Sicherheitszentrum	Bietet Informationen über Sicherheitsfunktionen und -ressourcen.
AWS Economics Center	Bietet Zugriff auf Informationen, Tools und Ressourcen für den Vergleich der Kosten für Amazon Web Services mit denen für alternative IT-Infrastrukturen.
Technische Whitepaper zu AWS	Bietet technische Whitepaper zu Themen wie Architektur, Sicherheit und Wirtschaftlichkeit. Diese Whitepaper wurden vom Amazon-Team, Kunden und Lösungsanbietern geschrieben.

Ressource	Beschreibung
AWS-Blogs	Bietet Blog-Posts, die neue Services und Aktualisierungen bestehender Services behandeln.
AWS-Podcast	Bietet Podcasts, die neue und bestehende Services behandeln und Tipps geben.

Dokumentverlauf

Dieser Dokumentverlauf ist mit der Version vom 30.9.2011 von "Erste Schritte" verknüpft. Dieses Handbuch wurde zuletzt am 8. März 2012 aktualisiert.

Änderung	Beschreibung	Veröffentlichungsdatum
Neuer Inhalt	Neues Dokument erstellt	30. September 2011
Neuer Abschnitt hinzugefügt	Neuer Abschnitt hinzugefügt, um AWS-Identitäts- und -Kontenverwaltung zu behandeln	24. Oktober 2011
Geänderte Anwendung und Linux AMI	Änderung des Beispiels, um zu zeigen, wie Drupal auf einem Amazon Linux AMI eingerichtet wird	23. November 2011
Neuer Abschnitt hinzugefügt	Neuer Abschnitt über die Herstellung einer Verbindung mit Amazon EC2 über den MindTerm-Client hinzugefügt	8. März 2012