Erste Schritte mit AWS

Computing-Grundlagen für Linux

Erste Schritte mit AWS: Computing-Grundlagen für Linux

Table of Contents

Übersicht

Für die Bereitstellung eines beliebigen Anwendungstyps ist normalerweise folgende Vorgehensweise erforderlich:

- Sie richten einen Computer ein, der Ihre Anwendung ausführt.
- Sie schützen Ihre Anwendung und Ressourcen.
- Sie richten ein Netzwerk ein, damit Benutzer auf Ihre Anwendung zugreifen können.
- Sie skalieren Ihre Anwendung.
- Sie überwachen Ihre Anwendung und Ressourcen.
- Sie stellen sicher, dass Ihre Anwendung fehlertolerant ist.

In dieser Anleitung finden Sie eine Einführung in eine Reihe von wichtigen AWS-Services und -Komponenten, die Sie bei der Bereitstellung dieser grundlegenden Anforderungen unterstützen. Sie erfahren mehr über diese wesentlichen Services, warum sie bei der Bereitstellung einer Webanwendung so wichtig sind, und wie Sie sie verwenden.

Um Ihnen den Einstieg in die Anwendung der grundlegenden AWS-Services zu erleichtern, betrachten wir eine Beispielarchitektur einer auf AWS gehosteten Web-Anwendung und führen Sie durch die Schritte zur Bereitstellung von Drupal. (Drupal ist ein quelloffenes Content Management System.) Sie können dieses Beispiel bei Bedarf an Ihre spezifischen Anforderungen anpassen. Am Ende dieser schrittweisen Anleitung werden Sie in der Lage sein, die folgenden Aufgaben auszuführen:

- Registrieren bei AWS.
- Starten, Verbinden, Schützen und Bereitstellen von Drupal (einschließlich einer MySQL-Datenbank) auf einem Computer in der Cloud
- Erstellen einer benutzerdefinierten Vorlage für einen Computer mit der erforderliche Hardware, Software und Konfiguration
- Einrichten eines Load Balancers zum Verteilen des Datenverkehrs über mehrere Computer in der Cloud
- Skalieren Ihrer Computerflotte in der Cloud
- Überwachen der Fehlerfreiheit Ihrer Anwendung und Computer
- 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 Feinheiten, damit die Abhängigkeiten funktionieren, nicht selbst herausfinden.
- Hosten einer Getting Started with AWS-Webanwendung für Linux – Enthält eine detailliertere schrittweise Anleitung, die weitere Services nutzt, wie Amazon Relational Database Service (Amazon RDS) und Amazon Route 53.
- Amazon Elastic Compute Cloud Getting Started Guide – Enthält eine schrittweise Einführung in die Verwendung von Amazon EC2-Instances.

Introduction to AWS

Wenn Sie für die Ausführung einer Webanwendung verantwortlich sind, finden Sie eine Reihe von Infrastruktur- und Architekturproblemen vor, für die AWS einfache, nahtlose und kostengünstige Lösungen bietet. 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. Dabei nehmen wir eine Unterteilung in die folgenden Abschnitte vor: Computerressourcen, Sicherheit, Überwachung, Netzwerk und Fehlertoleranz.

Computing-Ressourcen

Wenn Sie eine lokale Lösung bereitstellen, müssen Sie entsprechend Ihren Anforderungen einen Computer mit einem Betriebssystem, Software und Hardware kaufen. Bei der Bereitstellung Ihrer Lösung auf Amazon Web Services wählen Sie ein Amazon-Computerabbild (AMI) aus und verwenden dieses dann zum Bereitstellen eines virtuellen Servers, der als Amazon Elastic Compute Cloud (EC2)-Instance bezeichnet wird. Ein AMI ist eine Vorlage, die eine Softwarekonfiguration (bestehend beispielsweise aus Betriebssystem, Anwendungsserver und Anwendungen) enthält. Ein AMI kann z. B. die gesamte Software enthalten, um als Web-Server zu fungieren (z. B. Linux, Apache und Ihre Website). Amazon und die Amazon EC2-Community stellen eine große Auswahl von öffentlichen AMIs bereit. Wählen Sie ein AMI aus, das Ihren

Anforderungen am ehesten entspricht, und passen sie es an. Sie können diese angepasste Konfiguration in einem anderen AMI speichern und damit bei Bedarf neue Amazon EC2-Instances starten.

Die Speicherung kann ein zentraler Bestandteil einer Amazon EC2-Instance oder eine unabhängige Komponente sein, deren Lebensdauer getrennt von der Lebensdauer der Instance verwaltet wird. Für jede Speicherstrategie gibt es AMIs, und Sie müssen entscheiden, welchen Typ Sie verwenden möchten. Wenn Sie Ihre Amazon EC2-Instances starten, können Sie die Daten des Root-Geräts auf dem Amazon Elastic Block Store (Amazon EBS) oder auf dem lokalen Instance-Speicher ablegen. Amazon Elastic Block Store (Amazon EBS) ist ein dauerhaftes Speicher-Volume auf Blockebene, das an eine einzige laufende Amazon EC2-Instance angehängt werden kann. Amazon EBS-Volumes verhalten sich wie unformatierte, externe Block-Geräte, die Sie anhängen können. Sie existieren unabhängig von der Betriebsdauer einer Amazon EC2-Instance. Alternativ dazu gibt es den lokalen Instance-Speicher, der als temporäres Speicher-Volume nur während der Lebensdauer der Instance existiert. Sie können Amazon EBS-gestützte Instances für Web- oder Datenbankserver verwenden, die den Status lokal speichern und für die die Daten verfügbar sein müssen, auch wenn die zugeordnete Instance abstürzt. Sie können die vom Amazon Instance-Speicher gestützten Instances zur Verwaltung des Datenverkehrs auf großen Websites verwenden, auf denen jede Instance einen Klon darstellt. So können Sie Instances kostengünstig starten, wenn keine Daten auf dem Root-Gerät gespeichert sind. So können die zwei Hauptunterschiede zwischen diesen AMIs zusammengefasst werden:

- Sie können eine Amazon EBS-gestützte Instance anhalten und neu starten. Eine vom Amazon EC2-Instance-Speicher gestützte Instance kann jedoch nur ausgeführt oder beendet werden.
- Standardmäßig gehen alle Daten auf dem Instance-Speicher verloren, wenn die Instance ausfällt oder beendet wird. Daten auf Amazon EBS-gestützten Instances sind auf einem Amazon EBS-Volume gespeichert, damit keine Daten verloren gehen, wenn die Instance beendet wird.

Weitere Informationen über die Unterschiede zwischen Instance-Speicher-gestützten und Amazon EBS-gestützten Instances finden Sie unter Grundlagen zu Amazon EBS-gestützten AMIs und Instances im *Amazon Elastic Compute Cloud User Guide*.

Sicherheit

Wenn Sie einen neuen Computer gekauft haben, erstellen Sie normalerweise einen Benutzernamen und ein Passwort für die Anmeldung auf dem Computer. In AWS verwenden Sie ein Schlüsselpaar, bestehend aus einem öffentlichen und einem privaten Schlüssel, zur Verbindungsherstellung mit Ihrer Amazon EC2-Instance. Der öffentliche Schlüssel ist in Ihre Instance eingebettet. Der private Schlüssel dient zur sicheren Anmeldung ohne Passwort. Zur ersten Anmeldung bei Amazon Linux-Instances verwenden Sie (je nach AMI) den Benutzernamen "ec2-user" oder "root".

Nachdem Sie Ihre Anwendung bereitgestellt haben, schützen Sie Ihr System. Für eine lokale Bereitstellung geben Sie normalerweise die Ports und die Protokolle an, mit denen die Benutzer auf Ihre Anwendung zugreifen können. In AWS gehen Sie auf die gleiche Weise vor. AWS verfügt über Sicherheitsgruppen, die wie Netzwerk-Firewalls für eingehenden Datenverkehr fungieren, sodass Sie festlegen können, welche Benutzer über welche Ports auf Ihre Amazon EC2-Instances zugreifen können.

Skalierung

Möglicherweise stellen Sie fest, dass der Anwendungsdatenverkehr im Tagesverlauf schwankt. Zum Beispiel liegen die Spitzenzeiten mit hohem Datenverkehrsaufkommen zwischen 9 Uhr und 17 Uhr. Zu den anderen Tageszeiten ist es wesentlich niedriger. Da sich das Datenverkehrsaufkommen ändert, wäre es ratsam, die Anzahl der Computer, auf denen Ihre Anwendung ausgeführt wird, entsprechend diesen Änderungen kontinuierlich anzupassen. Auto Scaling kann Instances für Sie gemäß den festgelegten Richtlinien automatisch starten und beenden. Wenn Sie ein AMI als Grundlage definiert haben, startet Auto Scaling neue Instances mit genau derselben Konfiguration. Auto Scaling kann Ihnen auch Benachrichtigungen senden, wenn Instances hinzugefügt oder entfernt werden.

Überwachung

Sie müssen stets die aktuelle Leistung und den aktuellen Status Ihrer Ressourcen kennen. Wenn die Ressourcen nicht den geeigneten Status aufweisen, den Datenverkehr nicht verarbeiten können oder inaktiv sind, müssen Sie darüber informiert werden, damit Sie entsprechende Maßnahmen ergreifen können. Amazon CloudWatch dient zur Überwachung der AWS-Cloudressourcen und der auf AWS ausgeführten Anwendungen. Sie können Metriken erfassen und nachverfolgen, die Daten analysieren und unverzüglich reagieren, um den reibungslosen Betrieb Ihrer Anwendungen und Ihres Geschäfts zu gewährleisten. Sie können anhand der Informationen von Amazon CloudWatch die Richtlinien anwenden, die Sie mithilfe von Auto Scaling festlegen. Sie können zum Beispiel einen Alarm erstellen, der Sie benachrichtigt, wenn die CPU-Auslastung 95 % überschreitet. Wenn der Grenzwert überschritten wird, löst Amazon CloudWatch einen Alarm aus und Auto Scaling ergreift anhand der von Ihnen festgelegten Richtlinie entsprechende Maßnahmen. In diesem Beispiel startet Auto Scaling eine neue Instance, um die höhere Last verarbeiten zu können. Gleichermaßen können Sie einen Alarm festlegen, der Sie benachrichtigt, wenn die CPU-Auslastung unter einen bestimmten Grenzwert fällt. In diesem Fall könnte Auto Scaling eine Instance beenden, sodass Sie Kosten einsparen.

Sie können den Status Ihrer Instances überwachen, indem Sie Statusüberprüfungen und geplante Ereignisse für Ihre Instances anzeigen. Mit automatischen von Amazon EC2 durchgeführten Statusüberprüfungen lässt sich erkennen, ob bestimmte Probleme Ihre Instances beeinflussen. Anhand der Daten der Statusüberprüfungen zusammen mit den durch Amazon CloudWatch gewonnenen Daten erhalten Sie einen tiefgreifenden Einblick in die Funktionalität Ihrer jeweiligen Instances.

Sie können auch den Status der bestimmten Ereignisse einsehen, die für Ihre Instances geplant sind. Geplante Ereignisse liefern Informationen zu anstehenden Aktivitäten (wie beispielsweise das Neustarten oder Beenden einer Instance), die für Ihre Instances geplant sind, zusammen mit den geplanten Start- und Endzeiten des jeweiligen Ereignisses. Weitere Informationen zum Instance-Status erhalten Sie unter Monitoring the Status of Your Instances im *Amazon Elastic Compute Cloud User Guide*.

Netzwerk

Wenn Sie mehrere Computer zum Hosten Ihrer Webanwendung benötigen, müssen Sie den Datenverkehr auf diesen Computern entsprechend ausgleichen und gleichmäßig verteilen. Elastic Load Balancing bietet diesen Service genauso wie ein lokaler Load Balancer. Sie können einen Load Balancer einer Auto Scaling-Gruppe zuordnen. Während Instances gestartet und beendet werden, leitet der Load Balancer den Datenverkehr automatisch an die laufenden Instances weiter. Elastic Load Balancing führt auch Zustandsprüfungen für jede Instance durch. Wenn eine Instance nicht reagiert, kann der Load Balancer den Datenverkehr automatisch an die fehlerfreien Instances umleiten.

AWS weist Ihren AWS-Ressourcen wie Elastic Load Balancer und Amazon EC2-Instances eine URL zu. Möglicherweise möchten Sie jedoch eine URL haben, die spezifischer und leichter zu merken ist, z. B. www.example.com. Hierzu müssen Sie einen Domain-Namen von einer Domain-Vergabestelle kaufen. Nach dem Kauf können Sie mit Amazon Route 53 Ihren Domain-Namen Ihrer AWS-Bereitstellung zuordnen.

Möglicherweise möchten Sie ein privates, isoliertes Netzwerk bereitstellen. Amazon Virtual Private Cloud (Amazon VPC) ermöglicht die Bereitstellung eines privaten, isolierten Bereichs der Amazon Web Services (AWS)-Cloud, in dem Sie AWS-Ressourcen in einem von Ihnen definierten virtuellen Netzwerk ausführen können. Wenn Sie zum Beispiel eine Multi-Tier-Webanwendung hosten, sollten Sie die Netzwerkkonfiguration so anpassen, dass Ihre Webserver öffentlich sind und sich Ihre Datenbank und Anwendungsserver in einem privaten Subnetz ohne Internetzugang befinden. Anwendungsserver und Datenbanken sind nicht direkt über das Internet zugänglich, können jedoch über eine NAT-Instance auf das Internet zugreifen, um beispielsweise Patches herunterzuladen.

Sie können den Zugriff zwischen Servern und Subnetzen mittels Filterung eingehender und ausgehender Datenpakete steuern, die von Netzwerk-Zugriffskontrolllisten und Sicherheitsgruppen bereitgestellt wird. Im Folgenden sind einige andere Fälle aufgeführt, in denen Sie Amazon VPC verwenden sollten:

- Hosten skalierbarer Webanwendungen in der AWS-Cloud, die mit Ihrem Rechenzentrum verbunden sind
- Einbeziehen des Unternehmensnetzwerks in die Cloud
- Notfallwiederherstellung

Weitere Informationen zu den ersten Schritten mithilfe von Amazon VPC finden Sie unter Get Started with Amazon VPC im *Handbuch "Erste Schritte" für Amazon Virtual Private Cloud*.

Fehlertoleranz

Um Ihre Webanwendung fehlertolerant zu machen, müssen Sie in Betracht ziehen, Ihre Computer an verschiedenen physischen Standorten bereitzustellen. Für eine lokale Lösung kann sich die Verwaltung von Hardware an verschiedenen physischen Standorten als kostspielig erweisen. AWS bietet Ressourcen in verschiedenen Availability Zones und Regionen. Availability Zones sind vergleichbar mit Rechenzentren. Sie können mehrere Instances in unterschiedlichen Availability Zones ausführen. Wenn eine Availability Zone ausfällt (z. B. aufgrund einer Naturkatastrophe) wird sämtlicher Datenverkehr an eine andere Availability Zone umgeleitet. Jede Region enthält mehrere Availability Zones.

Ein noch größerer Vorteil ergibt sich, wenn die Instances über Regionen verteilt werden. Wenn eine Region mit sämtlichen Availability Zones komplett ausfällt, wird der Datenverkehr an eine andere Region umgeleitet.

Übersicht

In der folgenden Tabelle werden die wichtigsten Herausforderungen zusammengefasst, auf die Sie bei der Entwicklung einer einfachen Webanwendung stoßen, sowie die AWS-Services, die diese Herausforderungen angehen.

Herausforderung	Amazon Web Services	Vorteil
Es sind Computer erforderlich, auf denen Ihre Anwendung ausgeführt wird.	Amazon Elastic Compute Cloud (EC2)	Amazon EC2 führt den Webserver und die Anwendungsserver aus.
Eingehender Datenverkehr muss zur Maximierung der Leistung gleichmäßig auf Computern verteilt werden.	Elastic Load Balancing	Elastic Load Balancing unterstützt Zustandsprüfungen auf Hosts, Verteilung des Datenverkehrs an Amazon EC2-Instances in mehreren Availability Zones sowie dynamisches Hinzufügen und Entfernen von Amazon EC2-Hosts aus der lastenverteilten Rotation.
Server müssen bereitgestellt werden, um Kapazitäten zu Spitzenzeiten zu verarbeiten, zu anderen Zeiten werden die nicht genutzten Zyklen jedoch verschwendet.	Auto Scaling	Auto Scaling erstellt Kapazitätsgruppen von Servern, die bei Bedarf vergrößert oder verkleinert werden können.

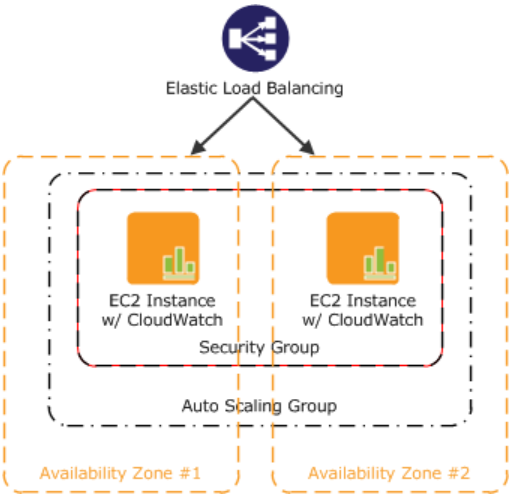
Herausforderung	Amazon Web Services	Vorteil
Server müssen auf Leistung und Zustand überwacht werden.	Amazon CloudWatch	Amazon CloudWatch meldet Metrikdaten für Amazon EC2-Instances, die dann von Auto Scaling verwendet werden.
Anwendungen benötigen möglicherweise einen persistenten Speicher.	Amazon Elastic Block Store (Amazon EBS)	Amazon EBS bietet ein persistentes Dateisystem für Web- und Anwendungsserver.

In der folgenden Tabelle werden zusätzliche Herausforderungen zusammengefasst, auf die Sie bei der Entwicklung einer einfachen Webanwendung stoßen, sowie die AWS-Komponenten, die diese Herausforderungen angehen.

Herausforderung	AWS-Komponente	Vorteil
Es ist ein sicherer Mechanismus zum Herstellen einer Verbindung mit dem Computer erforderlich.	Amazon-Schlüsselpaar	Schlüsselpaare sind Sicherheitsanmeldeinformationen, die Passwörtern entsprechen. Sie dienen zur sicheren Anmeldung bei einer ausgeführten Instance.
Sicherheit muss gewährleistet werden, um Anwendungsserver vor böswilligen Benutzern von außen zu schützen.	Amazon-Sicherheitsgruppe	Mit einer Amazon-Sicherheitsgruppe erhalten Sie Kontrolle über die Protokolle, Ports und Quell-IP-Adressbereiche, die zum Zugriff auf Ihre Amazon EC2-Instances berechtigt sind.
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.

Beispielarchitektur

Das folgende Diagramm zeigt eine Beispielarchitektur mit den im vorherigen Abschnitt erwähnten AWS-Ressourcen.



Als Beispiel führen wir Sie durch alle Schritte, die zur Bereitstellung einer einfachen Web-Anwendung erforderlich sind. Sofern Sie andere Aufgaben auszuführen haben, können Sie diese Beispielarchitektur an Ihre spezifische Situation anpassen. In diesem Diagramm führen Amazon EC2-Instances in einer Sicherheitsgruppe die Anwendung und den Web-Server aus. Die Amazon EC2-Sicherheitsgruppe fungiert als äußere Firewall für die Amazon EC2-Instances. Eine Auto Scaling-Gruppe verwaltet eine Flotte von Amazon EC2-Instances, die zum Abarbeiten der auftretenden Last automatisch hinzugefügt oder entfernt werden können. Diese Auto Scaling-Gruppe umfasst zwei Availability Zones zum Schutz vor potenziellen Fehlern in einer der Availability Zones. Um den Datenverkehr gleichmäßig auf die Amazon EC2-Instances zu verteilen, ist der Auto Scaling-Gruppe ein Elastic Load Balancer zugeordnet. Wenn die Auto Scaling-Gruppe Instances startet oder beendet, um auf Laständerungen zu reagieren, passt sich der Elastic Load Balancer automatisch entsprechend an.

Eine schrittweise Anleitung zum Erstellen dieser Architektur finden Sie unter Erste Schritte (p. 8). In dieser Anleitung erfahren Sie, wie Sie die folgenden Aufgaben ausführen:

- Registrieren bei AWS.
- Starten von Drupal, Herstellen einer Verbindung mit Drupal und Bereitstellen von Drupal auf einer Amazon EC2-Instance
- Erstellen eines benutzerdefinierten Amazon-Computerabbilds (AMI)
- Einrichten eines Elastic Load Balancers zum Verteilen des Datenverkehrs auf Ihre Amazon EC2-Instances
- Automatisches Skalieren Ihrer Instances-Flotte mithilfe von Auto Scaling
- Überwachen Ihrer AWS-Ressourcen mithilfe von Amazon CloudWatch
- Bereinigen Ihrer AWS-Ressourcen

Erste Schritte

Abstract

Anhand eines exemplarischen Beispiels lernen Sie im Folgenden die Schritte kennen, die in einer Linux-Umgebung erforderlich sind, um Ihre Web-Anwendung für AWS bereitzustellen.

Topics

Nehmen wir an, Sie möchten Drupal bereitstellen, ein quelloffenes Content Management System (CMS). Die ersten Schritte sind einfach, und für die meisten Aufgaben verwenden Sie die AWS Management Console. In diesem Thema lernen Sie die Schritte kennen, die erforderlich sind, um Ihre Web-Anwendung für AWS bereitzustellen. Es gibt viele verschiedene Möglichkeiten für die Bereitstellung Ihrer Web-Anwendung. Diese schrittweise Anleitung orientiert sich an bewährten Methoden und verwendet mehrere der Kern-Services, damit Sie sehen können, wie sie zusammenarbeiten.

Bevor Sie mit der Bereitstellung von Drupal mithilfe von AWS beginnen, müssen Sie sich mit einem AWS-Konto anmelden und die Kommandozeilen-Tools für das Auto Scaling installieren. Durch die Anmeldung bei AWS erhalten Sie Zugriff auf alle Services; bezahlen aber nur für solche, die Sie auch tatsächlich nutzen.

Zunächst werden Sie ein geeignetes AMI suchen, das Ihren Hardware- und Software-Anforderungen entspricht. Mit diesem AMI starten Sie eine Amazon EC2-Instance. Beim Start Ihrer Amazon EC2-Instance erstellen Sie ein neues Schlüsselpaar und eine Sicherheitsgruppe. Die Sicherheitsgruppe regelt, wer auf die Amazon EC2-Instance zugreifen darf. Ohne das Schlüsselpaar ist keine Verbindung mit der Amazon EC2-Instance möglich.

Ist Ihre Instance aktiv und gesichert, installieren Sie abschließend die erforderliche Software und konfigurieren dann die Drupal-Anwendung. Zur Vereinfachung des Starts neuer Amazon EC2-Instances, die bereits konfiguriert sind, erstellen Sie ein benutzerdefiniertes AMI als neue Grundlage.

Dann erstellen Sie einen Elastic Load Balancer, der den Datenverkehr auf mehrere Instances verteilt. Anschließend ändern Sie Ihre Sicherheitsgruppe, damit diese HTTP-Datenverkehr nur vom Load Balancer und nicht von jeder Adresse akzeptiert. Sie erstellen den Elastic Load Balancer, bevor Sie Ihre Instances starten, sodass Sie Ihr Auto Scaling-Gruppe mit Elastic Load Balancer verknüpfen können. Auf diese Weise kann der Load Balancer die Datenübertragung an deaktivierte Instances automatisch beenden bzw. eine Datenverbindung mit neu gestarteten Instances einrichten.

An diesem Punkt verwenden Sie Auto Scaling zum Starten Ihrer Amazon EC2-Instances. Sie erstellen dazu eine Auto Scaling-Richtlinie, die festlegt, wann Auto Scaling die Anzahl der Instances in Ihrer Gruppe erhöht oder verringert.

Zum Abschluss erstellen Sie einen CloudWatch-Alarm, mit dem die Instances in Ihrer Auto Scaling-Gruppe überwacht werden und der die Auto Scaling-Gruppe ggf. anweist, die Richtlinie anzuwenden.

Da es sich hierbei um eine Beispielbereitstellung handelt, sollten Sie alle von Ihnen erstellen AWS-Ressourcen beenden. Sobald Sie eine AWS-Ressource beenden, fallen dafür keine Gebühren mehr an.

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

Abstract

Installieren Sie die Befehlszeilen-Tools für Auto Scaling, sodass Sie mit der Erstellung Ihrer AWS-Ressourcen beginnen können.

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 im *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.

Nachdem Sie die Befehlszeilen-Tools installiert haben, können Sie mit der Erstellung Ihrer AWS-Ressourcen beginnen. Fahren Sie mit Schritt 3: Auswählen eines geeigneten AMI (p. 10) fort, um zu erfahren, wie Sie ein geeignetes AMI auswählen. Mit diesem AMI starten Sie Ihre Amazon EC2-Instance. Es dient auch als Grundlage zum Erstellen Ihres eigenen benutzerdefinierten AMI.

Schritt 3: Auswählen eines geeigneten AMI

Abstract

Beispiel zum Auswählen eines geeigneten AMI (Amazon Machine Image) für das Hosten Ihrer Web-Anwendung in einer Linux-Umgebung

Ein Amazon Machine Image (AMI) enthält alle erforderlichen Informationen zum Starten von Instances Ihrer Software. Ein AMI kann beispielsweise die gesamte Software enthalten, die erforderlich ist, um als Web-Server zu fungieren (etwa Linux, Apache und Ihre Website). Wir werden in dieser schrittweisen Anleitung eines dieser AMIs verwenden. Wenn Sie eine oder mehrere Amazon EC2-Instances aus einem AMI starten, stimmen alle diese Instances überein.

Amazon und die Amazon EC2-Community stellen eine große AMI-Auswahl bereit. Weitere Informationen finden Sie im AWS Marketplace.

Verwenden Sie die AWS Management Console (at http://console.aws.amazon.com) zur Suche nach AMIs, die bestimmte Kriterien erfüllen, und starten Sie dann Instances dieser AMIs. Sie finden damit zum Beispiel die von Amazon bereitgestellten Amis, die AMIs der EC2-Community oder AMIs, die ein bestimmtes Betriebssystem verwenden.

In diesem Schritt werden Sie ein Amazon Linux-AMI verwenden, in dem Apache, MySQL, PHP und Drupal installiert sind. Dieses AMI dient als Grundlage, die Sie anpassen und aus der Sie in einem späteren Schritt Ihr eigenes AMI erstellen.

Finden Sie ein geeignetes AMI wie folgt:

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Klicken Sie im Navigationsbereich auf AMIs.
3. Wählen Sie in den Filter-Listen Public images, dann Amazon images und dann Amazon Linux. Damit beschränken Sie die Anzeige auf AMIs, die durch Amazon Web Services bereitgestellt werden. Geben Sie im Textfeld **`drupal`** ein.
4. Wählen Sie ein AMI, in dem Drupal bereits installiert ist, und klicken Sie dann auf Launch.

Dieses AMI wird Ihre Grundlage sein. Mit einem Klick auf Launch rufen Sie den Startassistenten auf, der Ihre Instance konfiguriert und dann startet.

Schritt 4: Starten einer Instance

Abstract

Beispiel zum Starten einer Amazon EC2-Instance mithilfe des AMI beim Hosten Ihrer Web-Anwendung in einer Linux-Umgebung

Sie sind nun bereit zum Starten einer Amazon EC2-Instance mithilfe des AMI, das Sie im vorherigen Schritt ausgewählt haben. Zum Starten einer Instance sind folgende Aufgaben auszuführen:

- Konfigurieren der Instance
- Erstellen eines Schlüsselpaares
- Erstellen einer Sicherheitsgruppe.
- Starten der Instance

Im vorherigen Schritt haben Sie ein AMI ausgewählt und mit einem Klick auf Launch den Startassistenten aufgerufen. EC2 ermöglicht den Start einer Instance aber noch auf eine andere Weise. Wenn Sie im linken Navigationsbereich auf Instances klicken und dann auf Launch Instance, wird der Startassistent ebenfalls aufgerufen.

Da Sie bereits im vorherigen Schritt ein AMI ausgewählt haben, wird der Assistent erst mit Choose an Instance Type aufgerufen.

Important

Nachdem Sie die Instance gestartet haben, ist sie aktiv. Für die Instanz fallen die standardmäßigen Amazon EC2 Nutzungsgebühren an, bis Sie sie in der letzten Aufgabe dieser Übung beenden. Sofern Sie diese schrittweise Anleitung in einer Sitzung durchlaufen, sind die Kosten minimal (meistens geringer als ein Dollar). Weitere Informationen über Amazon EC2-Nutzungsgebühren finden Sie auf der Amazon EC2-Produktseite.

Starten Sie eine Amazon EC2-Instance wie folgt:

1. Wählen Sie auf der Seite Choose an Instance Type eine Hardware-Konfiguration und die Größe der zu startenden Instance. Übernehmen Sie in diesem Fall die Standardauswahl und klicken Sie auf Next: Configure Instance Details.
2. Ändern Sie die folgenden Einstellungen auf der Seite Configure Instance Details, falls erforderlich, übernehmen Sie die anderen Einstellungen und klicken Sie dann auf Next: Add Storage:

   - Network: Ihr Konto kann entweder die Plattformen die EC2-Classic und EC2-VPC unterstützen oder nur EC2-VPC. Wenn Ihr Konto EC2-Classic unterstützt, wählen Sie Launch into EC2-Classic und dann us-east-1b aus der Availability Zone-Liste. Wenn Ihr Konto nur EC2-VPC unterstützt, wählen Sie Ihre Standard-VPC und dann aus der Subnet-Liste ein Standardsubnetz der us-east-1b-Availability Zone.

   Klicken Sie in den nächsten Seiten des Assistenten auf Next, bis Sie die Seite Configure Security Group erreichen. Dann beginnen Sie mit dem nächsten Schritt.
3. Erstellen einer Sicherheitsgruppe:

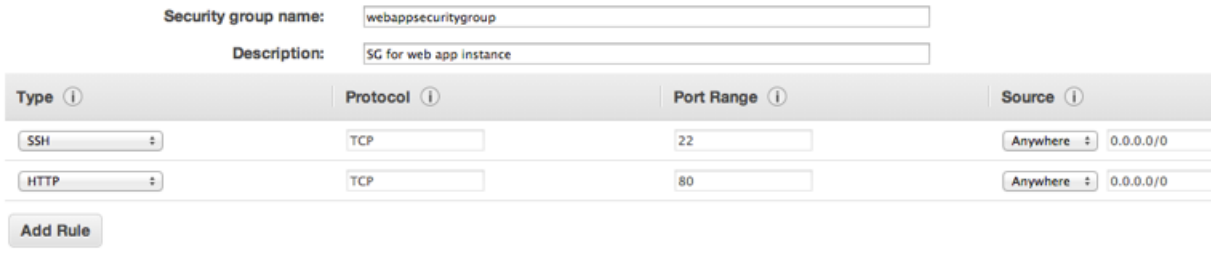
   Eine Sicherheitsgruppe definiert Firewall-Regeln für Ihre Instances. Diese Regeln legen fest, welcher eingehende Netzwerkverkehr zu Ihrer Instanz geliefert werden soll (z.B. Webverkehr auf Port 80 akzeptieren). Der gesamte übrige Verkehr wird ignoriert. Sie können die Regeln für eine Gruppe jederzeit ändern. Die neuen Regeln werden automatisch für alle laufenden Instanzen angewendet. Weitere Informationen über Sicherheitsgruppen finden Sie unter Using Security Groups in *Amazon Elastic Compute Cloud (Amazon EC2)*.

   Caution

   Der Startassistent erstellt standardmäßig eine Sicherheitsgruppe, die *allen* IP-Adressen den Zugriff auf Ihre Instance über SSH erlaubt. Dies ist für das kurze Beispiel in diesem Tutorial akzeptabel, aber nicht für Produktionsumgebungen. Für die Produktion wird nur eine bestimmte IP-Adresse bzw. ein bestimmter Adressbereich für den Zugriff auf Ihre Instance autorisiert.

   a. Ersetzen Sie im Feld Security group name den Namen der Standardsicherheitsgruppe durch **`webappsecuritygroup`**.
   b. Im Feld Description überschreiben Sie die Standardbeschreibung mit einem Text Ihrer Wahl.
   c. Klicken Sie auf Add Rule und wählen Sie HTTP aus der Liste Type.

   d. Klicken Sie auf Review and Launch.

   Die Sicherheitsgruppe wird erstellt und erhält eine ID (beispielsweise sg-48996e20). Ihre Instance wird in dieser neuen Sicherheitsgruppe gestartet.
4. Prüfen Sie Ihre Einstellungen und klicken Sie dann auf Launch. Sie werden aufgefordert, ein Schlüsselpaar auszuwählen oder zu erstellen. In dieser Übung erstellen Sie im nächsten Schritt ein neues Schlüsselpaar.

5. Erstellen Sie ein Schlüsselpaar:

   a. Aus einem öffentlichen AMI erstellte IAmazon EC2-Instances verwenden für die Anmeldung anstatt eines Kennworts ein Schlüsselpaar, bestehend aus einem privaten und einem öffentlichen Schlüssel. Der öffentliche Schlüssel ist in Ihrer Instance eingebettet. Mit dem privaten Schlüssel melden Sie sich ohne Passwort sicher an. Nachdem Sie Ihre AMIs erstellt haben, können Sie andere Methoden auswählen, um sich sicher an Ihren neuen Instances anzumelden.

      Wählen Sie Create a new key pair und geben Sie im Feld Key pair name den Namen **`mykeypair`** ein. Dieser Name wird der Name der persönlichen Schlüsseldatei, die mit dem Paar verknüpft ist (mit einer `.pem` Dateierweiterung).

   b. Klicken Sie auf Download Key Pair.

      Sie werden aufgefordert, Ihren persönlichen Schlüssel vom Schlüsselpaar zu Ihrem System zu speichern.

   c. Speichern Sie den privaten Schlüssel an einem sicheren Speicherort auf Ihrem System und notieren Sie sich, wo der Schlüssel gespeichert ist.

      Important

      Sie benötigen das Schlüsselpaar, um eine Verbindung mit Ihrer Amazon EC2-Instance herstellen zu können. Sie können die Datei mit dem Schlüsselpaar nicht erneut herunterladen. Nach einem Verlust der Datei haben Sie keine Möglichkeit mehr, eine Verbindung mit Ihrer Instance herzustellen.

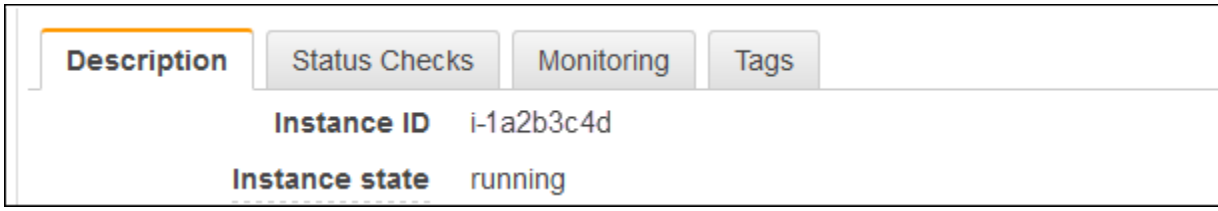
   d. Aktivieren Sie das Bestätigungsfeld und klicken Sie auf Launch Instances.

6. Klicken Sie in der angezeigten Bestätigungsmeldung auf View Instances. Es dauert einige Zeit, bis die Instance startet. Solange die Instance startet, wird ihr Status als *pending* angezeigt.

   Nach kurzer Zeit wechselt der Status der Instance auf *running* (in Ausführung). Mit einem Klick auf "Refresh" können Sie die Anzeige jederzeit selbst aktualisieren. Sobald für Ihre Instance der Status *running* angezeigt wird, stellen Sie eine Verbindung mit der Instance her und dann Ihre Anwendung bereit.

7. Notieren Sie den öffentlichen DNS-Namen für Ihre Instance:

   - Wählen Sie die aktive Instance und notieren Sie sich die im unteren Bereich angegebene öffentliche DNS-Adresse. Sie werden sie im nächsten Schritt benötigen.



Schritt 5: Bereitstellen der Anwendung

Topics

Nachdem Sie nun Ihre Amazon EC2-Instance gestartet haben, ist es Zeit, eine Verbindung vorzunehmen und dann Ihre Anwendung bereitzustellen. In diesen Schritt sorgen Sie zuerst für eine Verbindung mit Ihrer Amazon EC2-Instance und stellen dann die Anwendung Drupal bereit, die bereits im Linux AMI enthalten 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. 14)
2. Herstellen einer Verbindung mithilfe des MindTerm-Clients (SSH) (p. 14)

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 mithilfe des MindTerm-Clients (SSH)

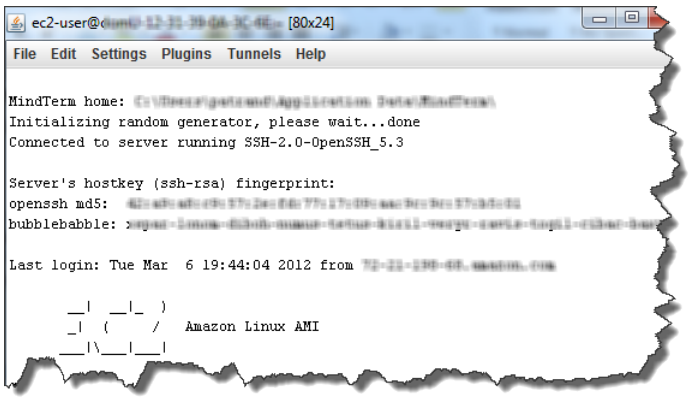
So stellen Sie die Verbindung zu Ihrer Instance über einen Web-Browser her

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 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 ist ec2-user der Standardbenutzername. Für Ubuntu ist ubuntu der Standardbenutzername. 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 persönlichen PEM-Schlüsseldatei an.
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 bei Ihrer Instance anmelden 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
```

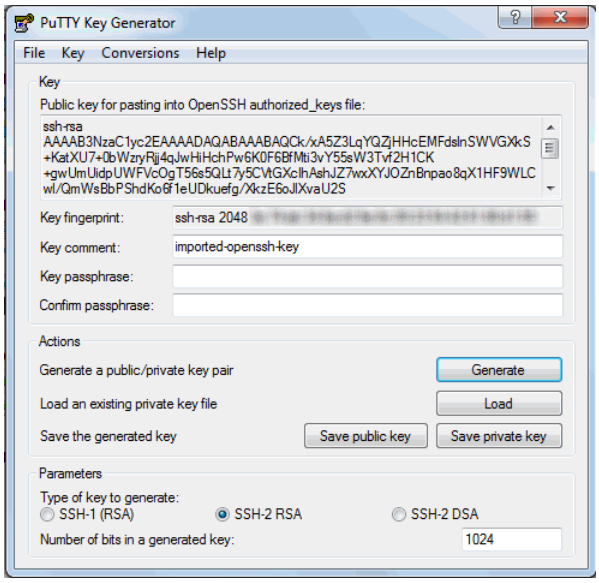
Herstellen einer Verbindung mit einer Amazon EC2-Instance auf einem Windows -Computer mithilfe von PuTTY

Wenn Sie Windows auf Ihrem lokalen Computer ausführen, ist Secure Shell (SSH) ist nicht vorhanden, sodass Sie PuTTY und PuTTYGen installieren müssen. Sie benötigen die Datei mit dem privaten Schlüssel, den Sie in Schritt 4: Starten einer Instance (p. 11) angelegt haben (beispielsweise `mykeypair.pem`).

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

1. Laden Sie PuTTY und PuTTYGen herunter und installieren Sie beide. 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 finden Sie unter Schritt 4: Starten einer Instance (p. 11).

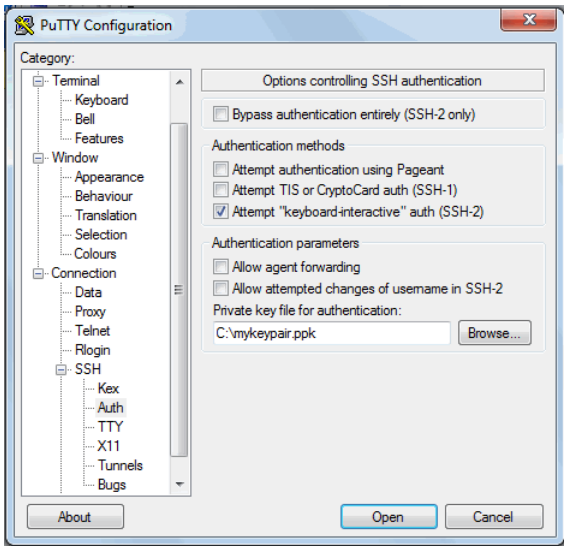
   a. Starten Sie PuTTYGen. Klicken Sie im Menü Conversions auf Import Key.
   b. Suchen Sie nach **`mykeypair.pem`** und klicken Sie dann auf Open.



   c. Klicken Sie auf Save private key. Wenn Sie gefragt werden, ob Sie den Schlüssel ohne Pass-Phrase speichern möchten, klicken Sie auf Yes. Speichern Sie den Schlüssel unter dem Namen **`mykeypair.ppk`**.
   d. Schließen Sie PuTTYGen.

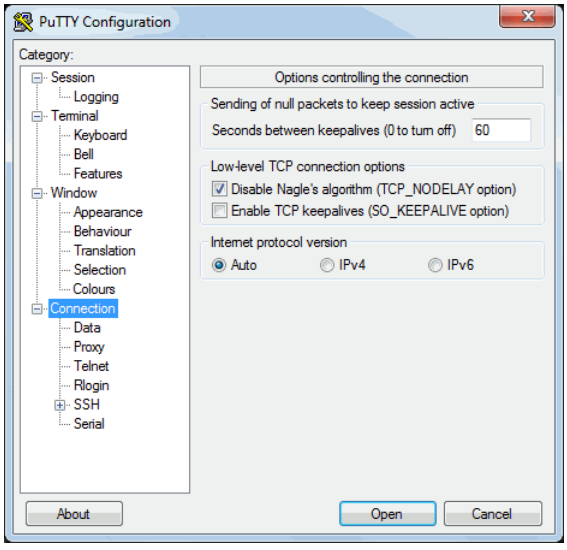
3. Konfigurieren der SSH-Einstellungen

   a. Starten Sie PuTTY, erweitern Sie den SSH-Knoten und klicken Sie dann auf Auth.
   b. Geben Sie im Feld Private key file for authentication den Speicherort von `mykeypair.ppk` ein.



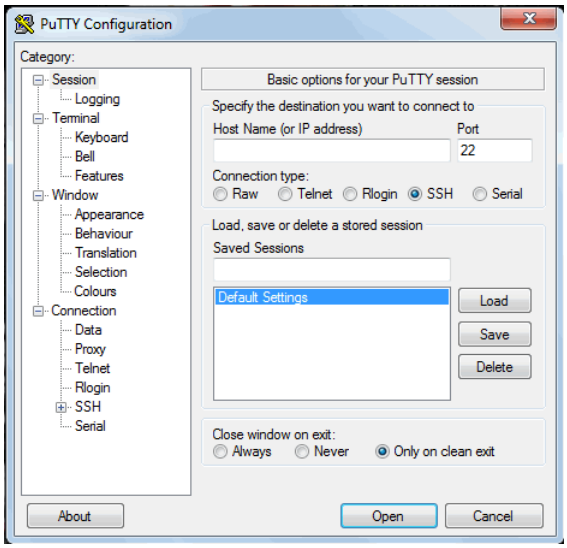
4. Ändern Sie die keepalive-Einstellung.

   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.

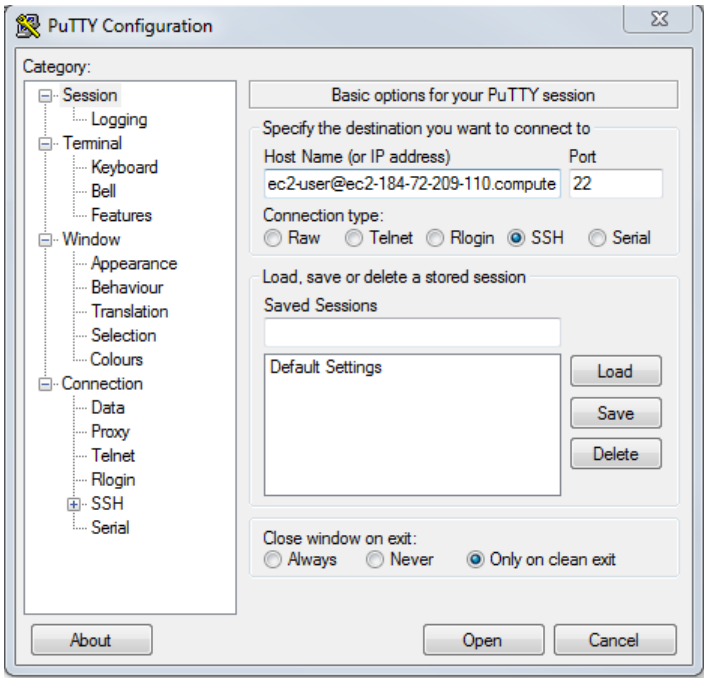
   a. Klicken Sie im Fenster PuTTY Configuration im Bereich Category auf Session.
   b. Klicken Sie im Feld Load, save, or delete a stored session auf Default Settings und dann auf Save.



6. Geben Sie die DNS-Adresse der Amazon EC2 -Instance ein, die Sie im vorherigen Schritt abgerufen haben.

   a. Klicken Sie im Fenster "PuTTY Configuration" im Bereich Category auf Sessions. Geben Sie im Feld Host Name (or IP address) folgendes ein: **`ec2-user`**@*<DNS-Adresse Ihrer Amazon EC2-Instance>*.

   > **Note**
   >
   > Der Benutzername für das AMI ist ec2-user.

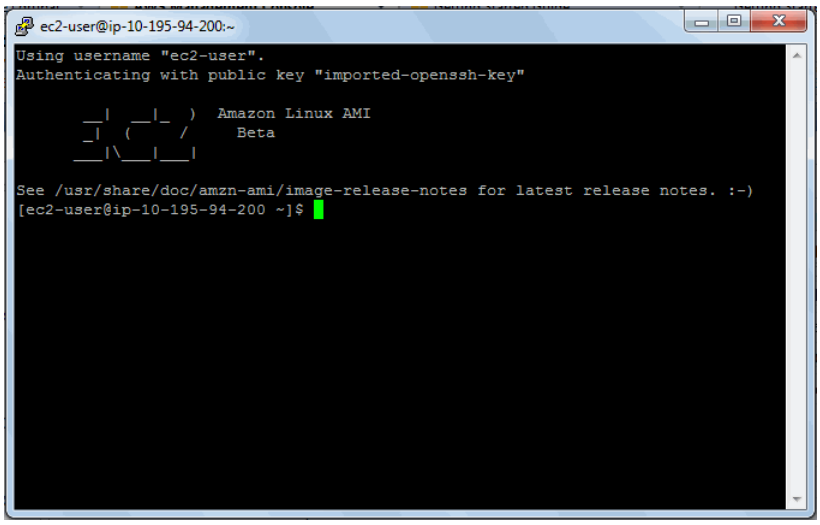


   b. Klicken Sie auf Öffnen. Das Dialogfeld PuTTY Security Alert wird geöffnet. Klicken Sie darin auf Yes, um zu bestätigen, dass der Fingerprint korrekt ist. Das Fenster SSH PuTTY wird geöffnet.

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. Weitere Informationen finden Sie unter Connecting Using PuTTY SSH in *Amazon Elastic Compute Cloud User Guide*.

Ihr Bildschirm sollte wie folgt aussehen:



Sie haben sich jetzt erfolgreich bei Ihrer Instance angemeldet und können diese nun konfigurieren. Anleitungen zum Konfigurieren einer Instance finden Sie unter Konfigurieren der Amazon EC2-Instance (p. 20).

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 Secure Shell (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.

Um SSH zur Verbindung zu verwenden

1. Geben Sie in einem Befehlszeilen-Tool den Speicherort der persönlichen Schlüsseldatei an, die Sie in Schritt 4: Starten einer Instance (p. 11) erstellt haben.
2. Verwenden Sie den Befehl `chmod`, um sicherzustellen, dass die persönliche Schlüsseldatei 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. Klicken Sie im linken Navigationsbereich auf Instances.
5. Wählen Sie Ihre Instance und klicken Sie dann auf Connect.
6. Klicken Sie auf A standalone SSH client. AWS erkennt automatisch die öffentliche DNS-Adresse Ihrer Instance und den Schlüsselpaarnamen, mit dem Sie die Instance gestartet haben.
7. Stellen Sie eine Verbindung mit Ihrer Instance unter deren öffentlichem Namen her. Lautet beispielsweise der Schlüsselname `mykeypair.pem` und der DNS-Name der Instance ec2-184-72-209-110.compute-1.amazonaws.com,, verwenden Sie den folgenden Befehl.

```
ssh -i mykeypair.pem ec2-user@ec2-184-72-209-110.compute-1.amazonaws.com
```

Note

In diesem Beispiel für dieses AMI ist **`ec2-user`** der Benutzername.

Sie erhalten eine Antwort wie die folgende.

```
The authenticity of host 'ec2-184-72-209-110.compute-1.amazonaws.com
(10.254.142.33)'
can't be established.
RSA key fingerprint is 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00.
Are you sure you want to continue connecting (yes/no)? yes
```

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. Weitere Informationen finden Sie unter Connect to Linux/UNIX Instances from Linux/UNIX with SSH in *Amazon Elastic Compute Cloud User Guide*.

8. Geben Sie **yes** ein.

   Sie erhalten eine Antwort wie die folgende.

```
Warning: Permanently added 'ec2-184-72-209-110.compute-1.amazonaws.com'
(RSA)
   to the list of known hosts.
```

Sie haben sich jetzt erfolgreich bei Ihrer Instance angemeldet und können diese nun konfigurieren. Anleitungen zum Konfigurieren einer Instance finden Sie unter Konfigurieren der Amazon EC2-Instance (p. 20).

Konfigurieren der Amazon EC2-Instance

In diesem Thema wird die aktive Instance konfiguriert. Dazu sind folgende Aufgaben auszuführen:

- Zuweisen von Berechtigungen für die Einstellungsdatei

- Installieren des MySQL-Servers
- Starten von Web-Server und MySQL
- Konfigurieren einer Datenbank
- Konfigurieren der Anwendung

Zur Vereinfachung dieses Tutorials erstellen wir eine Datenbank, die lokal auf einer Amazon EC2-Instance ausgeführt wird. Sie konfigurieren die Drupal-Anwendung so, dass sie diese Amazon EC2-Instance für Ihre Datenbank verwendet. Alle anderen Amazon EC2-Instances werden eine Verbindung mit dieser Instance herstellen, um auf die Datenbank zuzugreifen.

Sofern Sie mehr als eine Amazon EC2-Instance verwenden, richten Sie Ihre Datenbank in der Regel nicht auf dem Server ein, der Ihre Anwendung ausführt. Auf diese Weise werden die Information an nur einem Ort gespeichert, und alle Instances greifen auf die gleichen Daten zu – statt auf eine lokale Datenbank, die möglicherweise nicht synchronisiert ist.

Das Einrichten einer separaten Datenbank kann im Rahmen dieses Dokuments nicht behandelt werden. Weitere Informationen zum Einrichten von Amazon RDS für eine Web-Anwendung finden Sie unter Schritt 8: Hinzufügen von Amazon RDS in *Hosten einer Getting Started with AWS-Webanwendung für Linux*.

Weisen Sie Berechtigungen für die Einstellungsdatei wie folgt zu:

- Geben Sie auf Ihrer Amazon EC2-Instance in einer Eingabeaufforderung den folgenden Befehl ein, um Berechtigungen festzulegen:

```
sudo chmod 666 /var/www/html/sites/default/settings.php
```

Installieren Sie einen MySQL-Server wie folgt:

- Geben Sie auf Ihrer Amazon EC2-Instance in einer Eingabeaufforderung den folgenden Befehl ein, um einen MySQL-Server zu installieren:

```
sudo yum install mysql-server
```

  Nach der entsprechenden Aufforderung geben Sie "y" ein.

Starten Sie den Web-Server und MySQL wie folgt:

1. Starten Sie auf Ihrer Amazon EC2-Instance den Web-Server in einer Eingabeaufforderung und konfigurieren Sie ihn dann so, dass er bei einem Neustart automatisch ausgeführt wird:

```
sudo chkconfig httpd on
```

```
sudo service httpd start
```

   Sie erhalten eine Antwort wie die folgende.

```
Starting httpd [OK]
```

2. Starten Sie MySQL und legen Sie fest, dass es bei einem Neustart automatisch ausgeführt wird.

```
sudo chkconfig mysqld on
```

```
sudo service mysqld start
```

Sie erhalten eine Antwort wie die folgende.

```
Starting mysqld [OK]
```

Konfigurieren Sie eine Datenbank wie folgt:

1. Ändern Sie auf Ihrer Amazon EC2-Instance das Passwort für den "root"-Benutzer: In diesem Beispiel verwenden Sie das Passwort "root".

   ```
   mysqladmin -u root password root
   ```

2. Erstellen Sie eine Datenbank. In diesem Beispiel verwenden Sie den Datenbanknamen "mydb".

   ```
   mysqladmin -u root -p create mydb
   ```

   Wenn Sie nach einem Passwort gefragt werden, geben Sie "root" ein.
3. Melden Sie sich an und legen Sie die Zugriffsberechtigungen für die Datenbank fest.

   ```
   mysql -u root -p
   ```

   Wenn Sie nach einem Passwort gefragt werden, geben Sie "root" ein.
4. Legen Sie in der MySQL-Eingabeaufforderung die Berechtigungen mit folgendem Befehl fest: Ersetzen Sie <your public EC2 DNS address> durch die von Ihnen in Schritt 4: Starten einer Instance (p. 11) notierte öffentliche DNS-Adresse der Amazon EC2-Instance.

   ```
   GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, LOCK
   TABLES, CREATE TEMPORARY TABLES ON mydb.* TO 'awsuser'@'<your public EC2
   DNS address>' IDENTIFIED BY 'mypassword';
   ```

   Nach erfolgreicher Ausführung antwortet MySQL wie folgt:

   ```
   Query OK, 0 rows affected
   ```

5. Legen Sie in der MySQL-Eingabeaufforderung die Berechtigungen mit folgendem Befehl fest:

   ```
   GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, LOCK
   TABLES, CREATE TEMPORARY TABLES ON mydb.* TO 'awsuser'@'%' IDENTIFIED BY
   'mypassword';
   ```

   Nach erfolgreicher Ausführung antwortet MySQL wie folgt:

   ```
   Query OK, 0 rows affected
   ```

6. Geben Sie folgenden Befehl ein, um die neuen Berechtigungen zu aktivieren:

```
FLUSH PRIVILEGES;
```

   Nach erfolgreicher Ausführung antwortet MySQL wie folgt:

```
Query OK, 0 rows affected
```

7. Verlassen Sie die MySQL-Eingabeaufforderung mit folgendem Befehl:

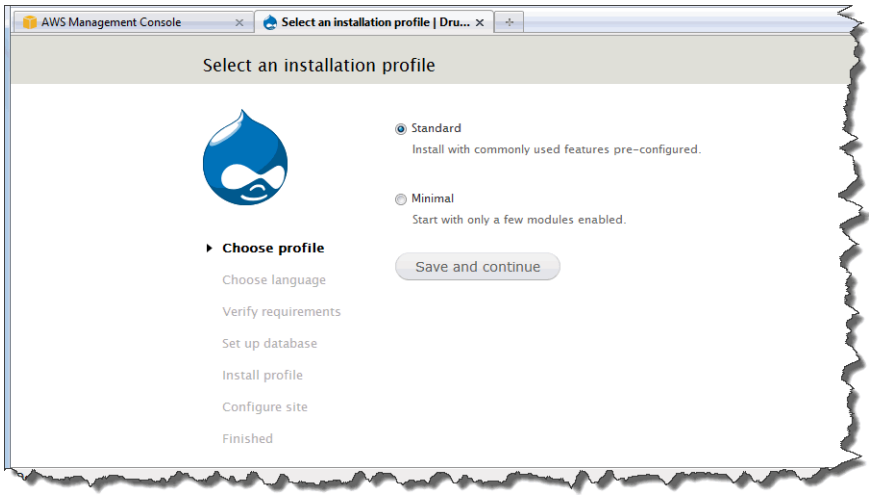
```
exit
```

   Der Server antwortet wie folgt:

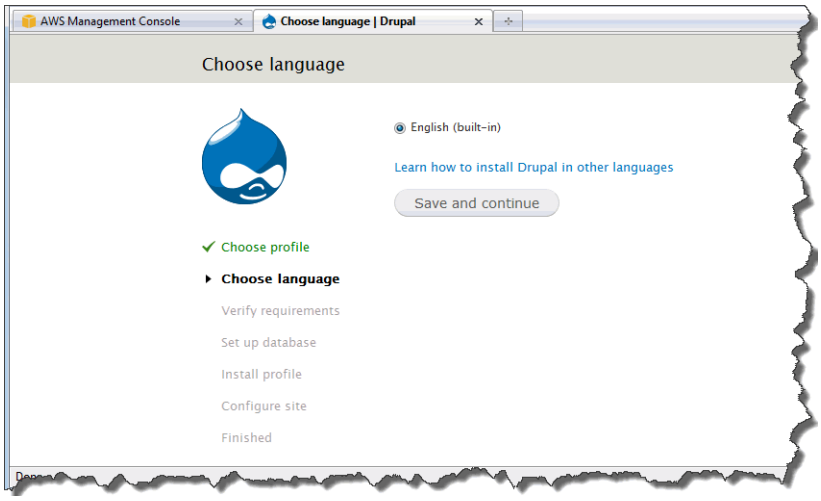
```
Bye
```

Konfigurieren Sie die Anwendung wie folgt:

1. Öffnen Sie Ihren Web-Browser. Geben Sie im Feld "Address" die von Ihnen in Schritt 4: Starten einer Instance (p. 11) aufgezeichnete öffentliche DNS-Adresse der Amazon EC2-Instance ein. Im Drupal-Installationsassistenten wird die Seite Choose profile geöffnet.
2. Klicken Sie auf der Seite Choose profile auf Standard und dann auf Save and continue.

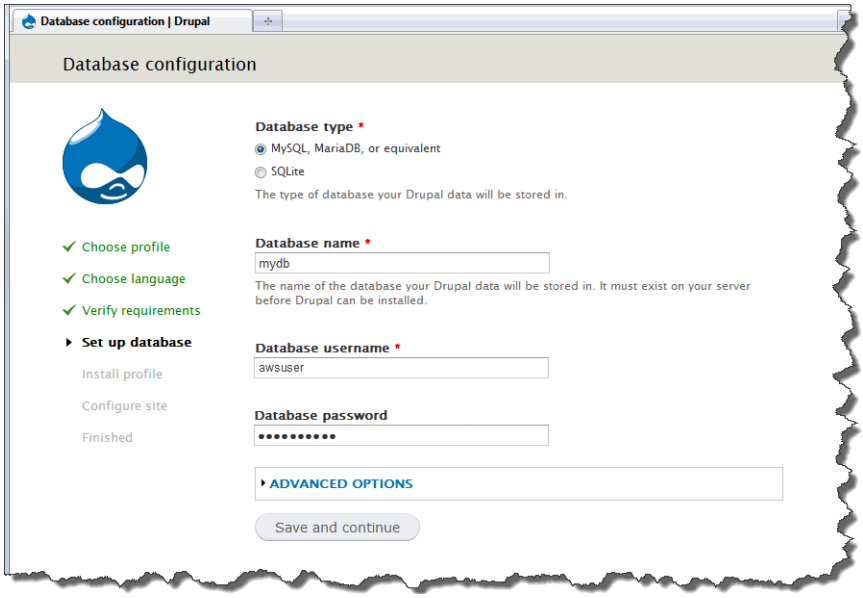


3. Klicken Sie auf der Seite Choose language auf English und dann auf Save and continue. Die Seite Set up database wird angezeigt.

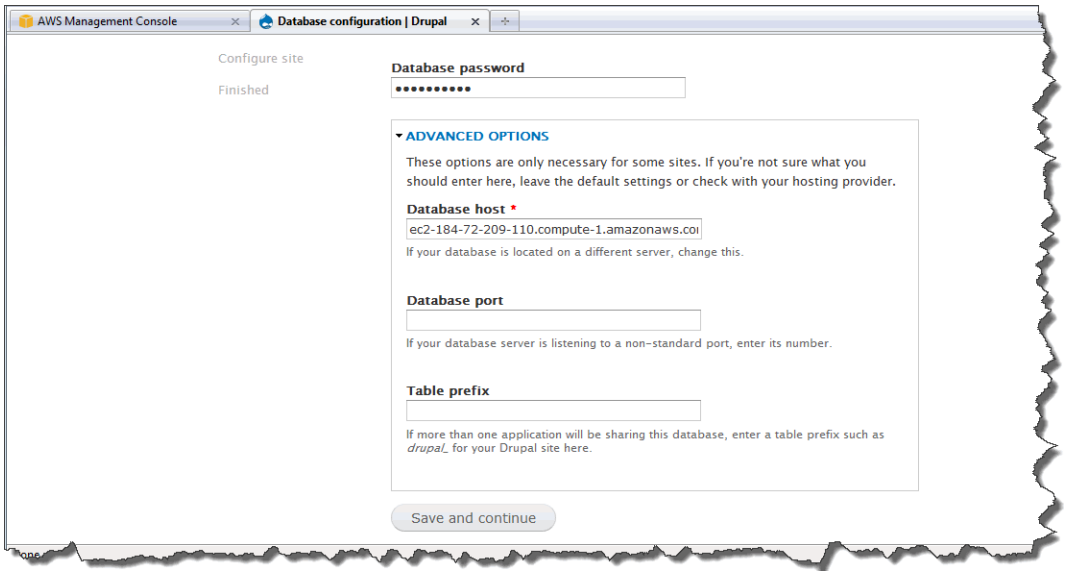


4. Geben Sie auf der Seite Set up database Folgendes ein:

   a. Klicken Sie auf der Seite Database type auf MySQL, MariaDB, or equivalent.
   b. Geben Sie im Feld Database name den Namen Ihrer Datenbank ein. In diesem Beispiel verwenden Sie **`mydb`**.
   c. Geben Sie im Feld Database username den Benutzernamen für Ihre Datenbank ein. In diesem Beispiel verwenden Sie **`awsuser`**.
   d. Geben Sie im Feld Database password das Passwort für Ihre Datenbank ein. In diesem Beispiel verwenden Sie **`mypassword`**.



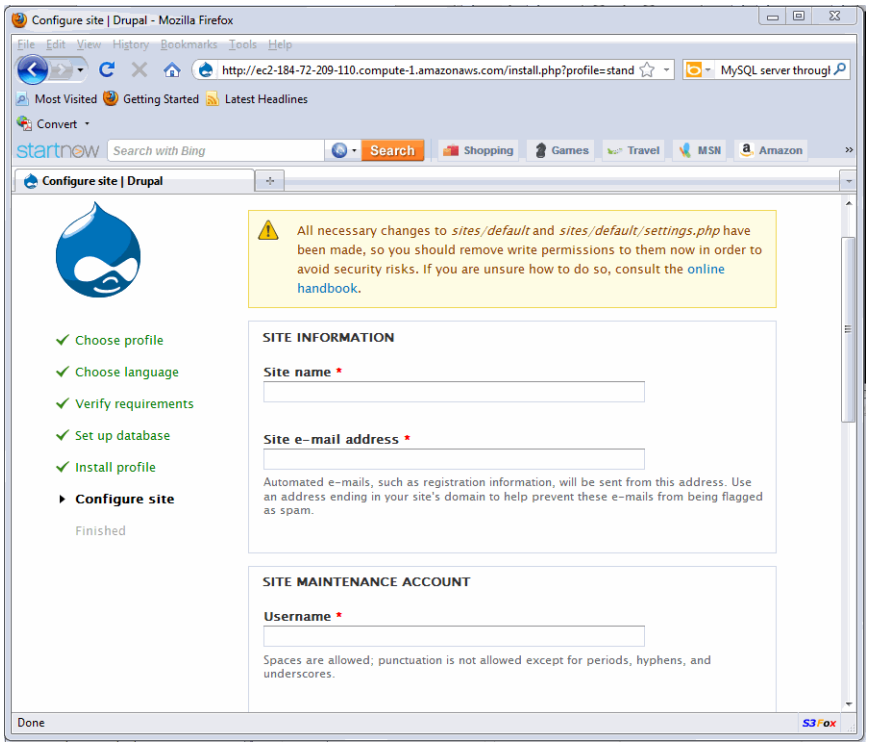
   e. Klicken Sie auf Advanced Options.
   f. Geben Sie im Feld Database host die von Ihnen in Schritt 4: Starten einer Instance (p. 11) notierte öffentliche DNS-Adresse der Amazon EC2-Instance ein.



g. Klicken Sie auf Save and continue.

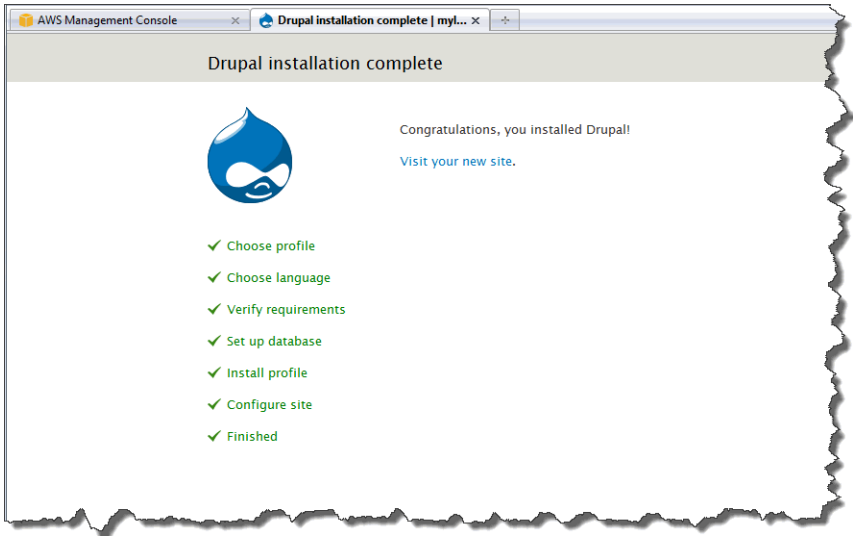
5. Geben Sie auf der Seite Configure site Folgendes ein:

   a. Geben Sie im Feld Site name den Namen Ihrer Site ein.
   b. Geben Sie im Feld Site e-mail address Ihre E-Mail-Adresse ein.
   c. Geben Sie im Feld Username einen Benutzernamen ein.
   d. Geben Sie im Feld Password das zum Benutzernamen gehörende Passwort ein.
   e. Geben Sie im Feld Confirm password das Passwort erneut ein.

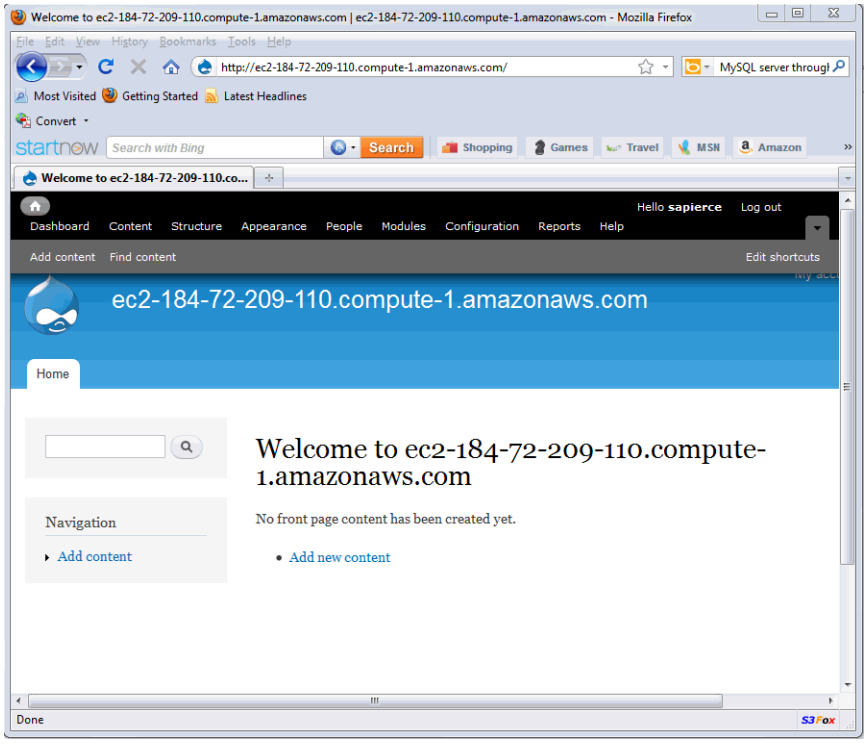


f. Klicken Sie auf Save and continue.

Die Installation ist abgeschlossen.



6. Klicken Sie auf Visit your new site.



7. Um Ihrer Site neue Inhalte hinzuzufügen, klicken Sie auf Add new content.

Herzlichen Glückwunsch! Sie haben Ihre Web-Anwendung erfolgreich mit Amazon Web Services bereitgestellt. Wenn Sie zukünftig weitere Instances starten, werden Sie nicht jede einzeln anpassen wollen. Wir werden deshalb ein benutzerdefiniertes AMI erstellen, das alle bisher vorgenommenen Konfigurationsänderungen enthält.

Schritt 6: Erstellen eines benutzerdefinierten Amazon-Computerabbilds (AMI)

Abstract

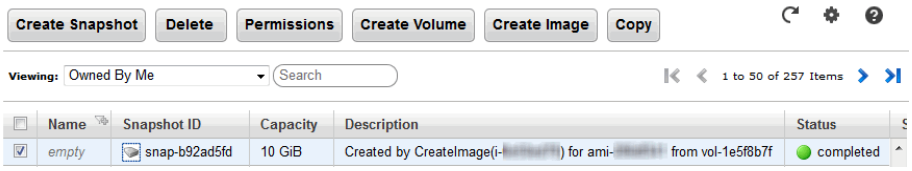
Erstellen Sie beim Hosten Ihrer Webanwendung in Windows ein benutzerdefiniertes Amazon-Computerabbild (AMI) und starten Sie zukünftige Umgebungen mit dieser gespeicherten Konfiguration.

Nachdem wir nun unsere Amazon EC2-Instance angepasst haben, können wir dieses Amazon-Computerabbild (AMI) speichern und zukünftige Umgebungen mit dieser gespeicherten Konfiguration starten.

Erstellen Sie ein AMI von einer laufenden Amazon EC2-Instance wie folgt:

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Klicken Sie im Navigationsbereich auf Instances.
3. Klicken Sie auf der Seite Instances mit der rechten Maustaste auf die laufende Instance und klicken Sie dann auf Create Image.
4. Geben Sie im Dialogfeld Create Image einen eindeutigen Abbildnamen und eine optionale Beschreibung für das Abbild ein (bis zu 255 Zeichen). Klicken Sie dann auf Create Image.

   &EC2 beendet die Instance, erfasst Abbilder von allen angefügten Volumes, erstellt und registriert das AMI und startet dann die Instance neu.
5. Im Dialogfeld Create Image wird die AMI-ID angezeigt. Notieren Sie sich diese ID, da Sie sie für eine spätere Aufgabe benötigen.
6. Um den Status des neuen AMI anzuzeigen, klicken Sie im Navigationsbereich auf AMIs. Während das neue AMI erstellt wird, hat es den Status *pending*. Es dauert einige Minuten, bis der Vorgang beendet ist.
7. Wenn das AMI den Status *available* annimmt, öffnen Sie die Seite Snapshots, indem Sie im Navigationsbereich auf Snapshots klicken. Sehen Sie sich den neuen Snapshot an, der für das AMI erstellt wurde. Alle Instances, die Sie über das neue AMI starten, verwenden diesen Snapshot als Root-Gerät-Datenträger.



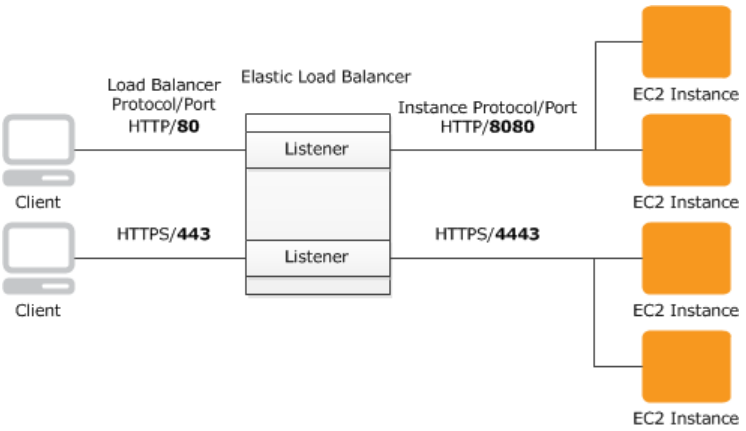
Möglicherweise möchten Sie mehrere Amazon EC2-Instances in mehreren Availability Zones ausführen. Wenn eine Availability Zone ausfällt, wird der Datenverkehr an eine andere Availability Zone umgeleitet. Ein Elastic Load Balancer verbessert die Verfügbarkeit Ihrer Anwendung, unabhängig davon, ob sich sämtliche Instances in der gleichen Availability Zone oder in mehreren Availability Zones befinden. Informationen zum Erstellen eines Elastic Load Balancers finden Sie unter Schritt 7: Erstellen eines Elastic Load Balancers (p. 27).

Schritt 7: Erstellen eines Elastic Load Balancers

Abstract

Erstellen Sie einen Elastic Load Balancer, um den eingehenden Datenverkehr der Webanwendung auf sämtliche Amazon EC2-Instances automatisch zu verteilen und dort auszugleichen.

Mit Elastic Load Balancing wird der eingehende Anwendungsdatenverkehr auf sämtliche laufende Instances automatisch verteilt und dort ausgeglichen, wodurch sich die Verfügbarkeit und Skalierbarkeit Ihrer Anwendung verbessern lässt. Der Service erleichtert auch das Hinzufügen neuer Instances oder das Entfernen unausgelasteter Instances, wenn Sie die Kapazität Ihrer Anwendung erhöhen oder verringern müssen. Im folgenden Diagramm wird die Funktionsweise des Load Balancers dargestellt. In diesem Diagramm enthält der Load Balancer zwei Listener. Der erste Listener akzeptiert mittels HTTP Datenverkehr auf Port 80 und leitet diese Anforderungen mittels HTTP auf Port 8080 an die Amazon EC2-Instances weiter. Der andere Listener akzeptiert mittels HTTPS Datenverkehr auf Port 443 und leitet diese Anforderungen mittels HTTPS auf Port 4443 an die Amazon EC2-Instances weiter.



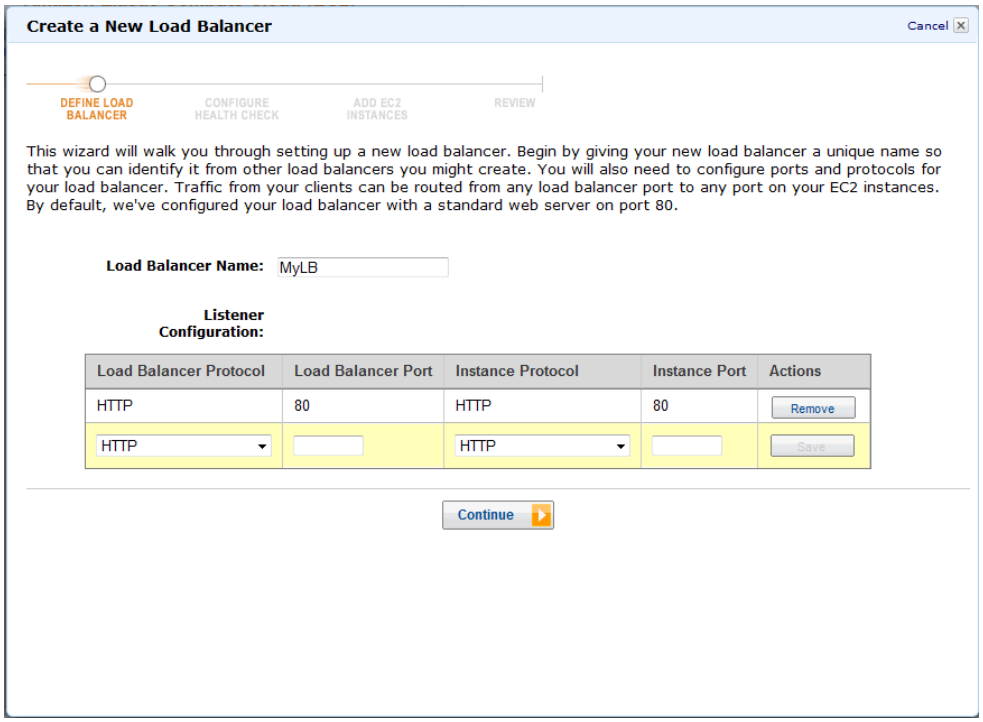
Sie können das Protokoll und den Port sowohl für den Client als auch die Amazon EC2-Instances angeben. In diesem Schritt erstellen wir einen Load Balancer für einen HTTP-Service. Wir legen fest, dass der Load Balancer auf Port 80 auf eingehenden Datenverkehr von Clients wartet und diesen dann auf Port 80 an die Instances verteilt.

Sobald der Load Balancer verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, in der der Load Balancer ausgeführt wird. Weitere Informationen zu Elastic Load Balancing-Preisen finden Sie auf der Detailseite Elastic Load Balancing.

Weitere Informationen zu Elastic Load Balancern finden Sie in der 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. Klicken Sie im linken Navigationsbereich auf Load Balancers.
   c. Klicken Sie auf Create Load Balancer.
   d. Geben Sie im Assistenten Create a New Load Balancer auf der Seite Define Load Balancer einen Namen für den Load Balancer ein. Geben Sie in diesem Beispiel **`MyLB`** ein.



e. Belassen Sie für dieses Beispiel die Standardeinstellung unter Listener Configuration. In den Feldern "Load Balancer Port" und "Load Balancer Protocol" werden der Port und das Protokoll angegeben, mit denen der Load Balancer auf den Datenverkehr der Clients wartet. In den Feldern "Instance Protocol" und "Instance Port" werden der Port und das Protokoll angegeben, mit denen der Load Balancer den Datenverkehr an die Instances weiterleitet. Wenn der Load Balancer zum Beispiel den Datenverkehr über Port 8080 an die Instances weiterleiten soll, können Sie dies hier festlegen.

> **Note**
>
> Einmal konfigurierte Listenerinformationen können nicht mehr geändert werden. Wenn Sie diese Informationen aktualisieren möchten, müssen Sie einen neuen Load Balancer erstellen.

f. Klicken Sie auf Weiter.

2. Konfigurieren Sie die Zustandsprüfung:

   Elastic Load Balancing überprüft regelmäßig die jeweilige lastenverteilte Amazon EC2-Instance auf Fehlerfreiheit. Wenn Elastic Load Balancing eine fehlerhafte Instance erkennt, wird der Datenverkehr nicht mehr an diese Instance gesendet und stattdessen an eine fehlerfreie Instance umgeleitet.

   a. Führen Sie auf der Seite Configure Health Check unter Configuration Options die folgenden Schritte aus:

      - Belassen Sie den Wert für Ping Protocol auf der Standardeinstellung HTTP. Wenn Sie zukünftig ein sicheres Protokoll verwenden möchten, mit dem der Load Balancer Ping-Anforderungen an die Instances sendet, können Sie HTTPS verwenden und einen anderen Port festlegen. Weitere Informationen zum Verwenden von HTTPS mit Elastic Load Balancing finden Sie im *Entwicklerhandbuch für Elastic Load Balancing* unter "Entwicklerhandbuch für Elastic Load Balancing".

- Belassen Sie den Wert für Ping Port auf der Standardeinstellung "80".

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

  Note

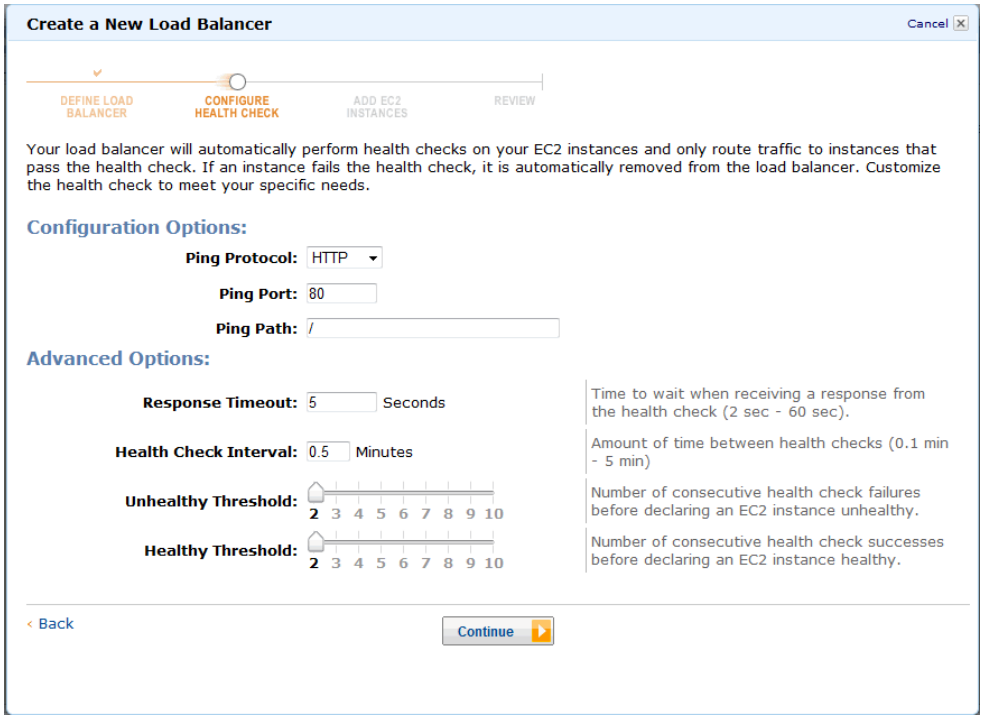
  Wenn Sie einen Wert für den Ping-Port angeben, müssen die Amazon EC2-Instances eingehenden Datenverkehr auf dem von Ihnen festgelegten Port annehmen. Sie können einen anderen Portwert als 80 festlegen und diesen Wert jederzeit ändern.

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

  Elastic Load Balancing sendet Zustandsprüfungsabfragen an den von Ihnen angegebenen Ping-Path. In diesem Beispiel wird ein Schrägstrich verwendet, 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.

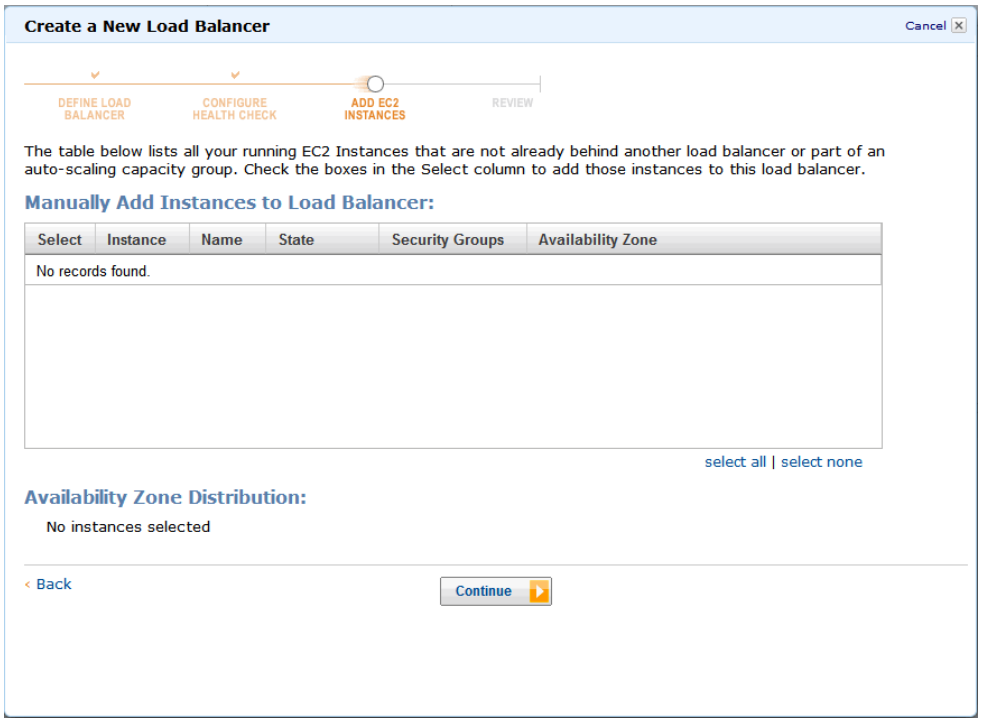
b. Stellen Sie unter Advanced Options den Wert für Healthy Threshold auf 2 ein. Übernehmen Sie die Standardwerte für die anderen Optionen.

   Normalerweise ist der Standardwert "10" ausreichend als Grenzwert für fehlerfreie Zustandsprüfungen. Um dieses Tutorial schneller auszuführen, legen wir "2" fest, sodass Sie nicht so lange warten müssen, bis die fehlerfreien Instances angezeigt werden.

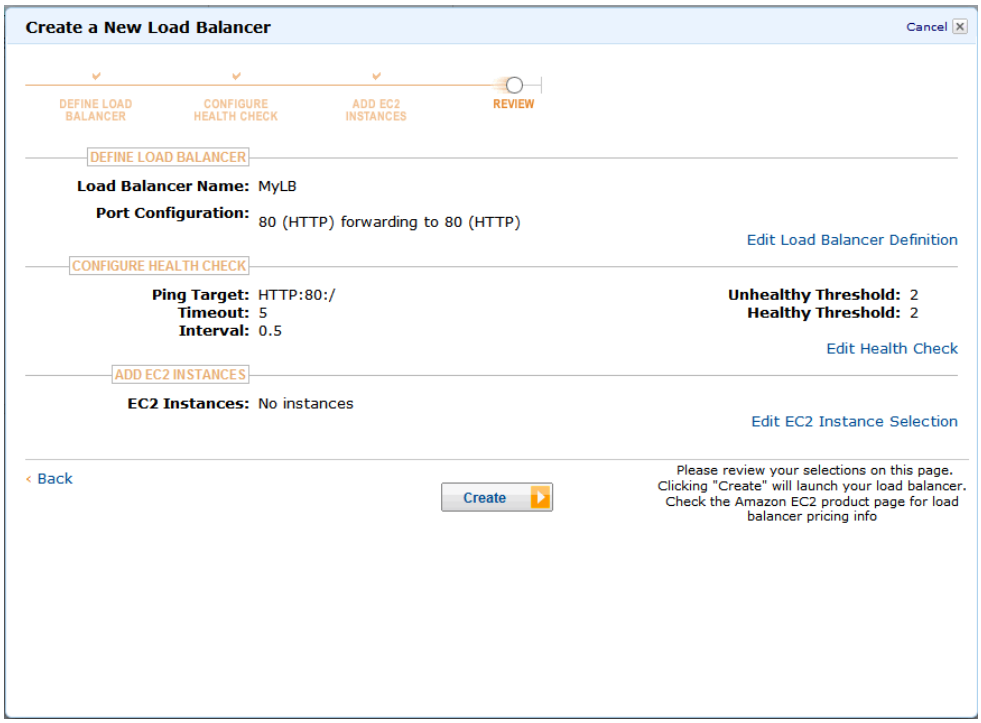


c. Klicken Sie auf Weiter.

3. Klicken Sie auf der Seite Add EC2 Instances auf Continue.



4. Überprüfen Sie die Einstellungen. Wenn Sie die Einstellungen ändern möchten, klicken Sie für einen bestimmten Schritt im Prozess auf den Link Edit.

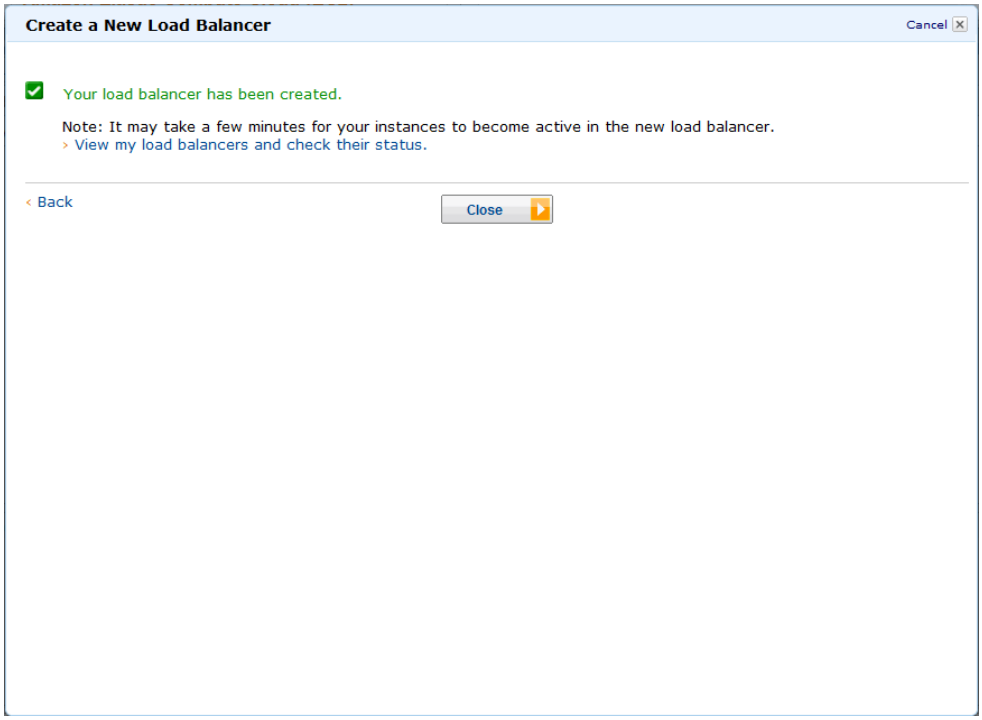


Important

Nach dem Erstellen eines Load Balancers können Sie jede beliebige Einstellung ändern, außer Load Balancer Name und Port Configuration. Wenn Sie einen Load Balancer umben-

ennen oder seine Portkonfiguration ändern möchten, erstellen Sie einen Ersatz-Load Balancer.

5. Klicken Sie auf Create.
6. Klicken Sie auf der Bestätigungsseite auf Close.

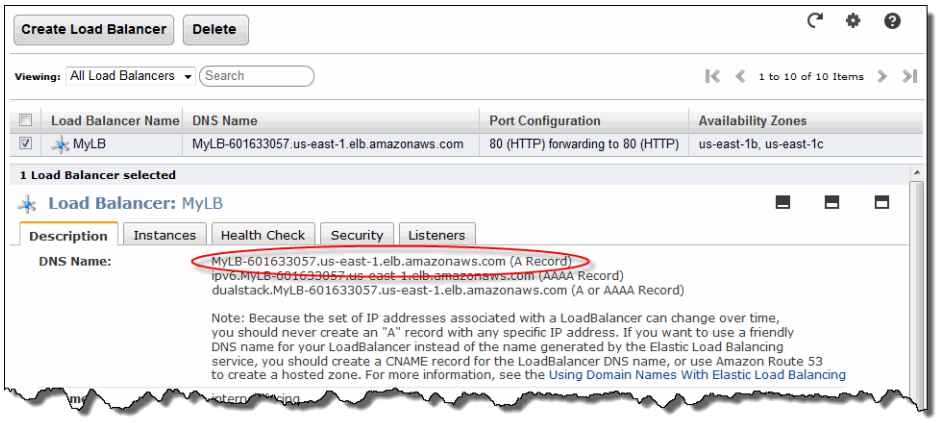


Der neue Load Balancer wird nun in der Liste angezeigt.

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

7. Notieren Sie sich die öffentliche DNS-Adresse:

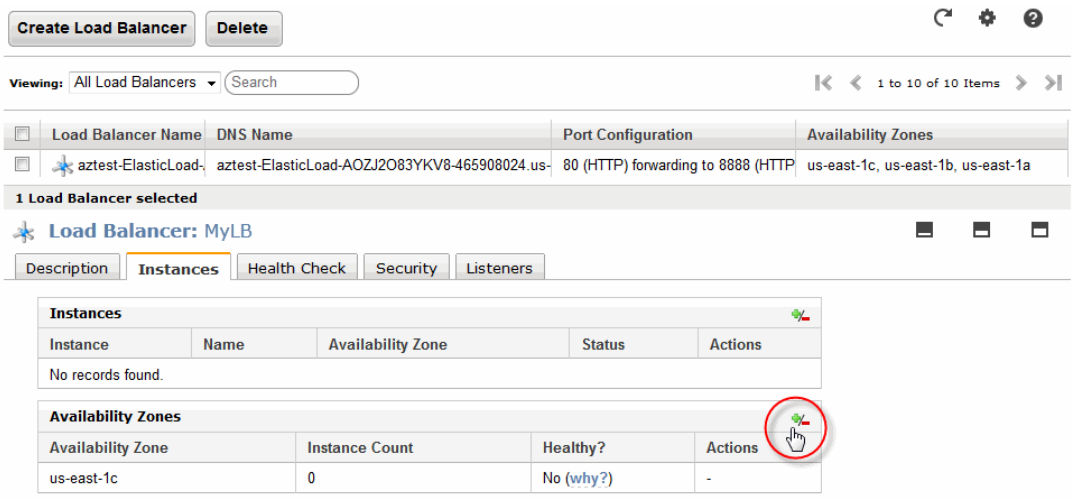
   a. Klicken Sie im Bereich Load Balancers auf MyLB.
   b. Klicken Sie auf die Registerkarte Description.



c. Notieren Sie sich die öffentliche DNS-Adresse. Sie benötigen sie zu einem späteren Zeitpunkt in diesem Tutorial.

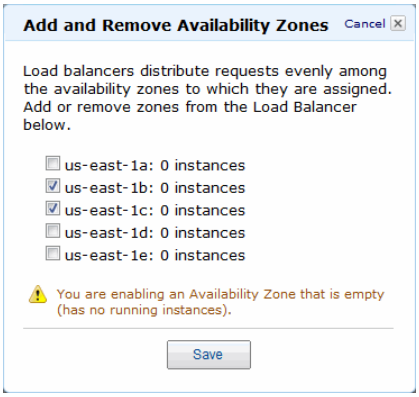
8. Fügen Sie eine Availability Zone hinzu:

   a. Klicken Sie in der Liste der Load Balancers auf MyLB.
   b. Klicken Sie auf die Registerkarte Instances.
   c. Klicken Sie auf das Plus-Symbol.



   d. Führen Sie im Dialogfeld Add and Remove Availability Zones folgende Schritte aus:

      - Aktivieren Sie das Kontrollkästchen us-east-1b: 0 instances.
      - Aktivieren Sie das Kontrollkästchen us-east-1c: 0 instances.
      - Klicken Sie auf Speichern.



In einer späteren Aufgabe starten Sie mithilfe von Auto Scaling die Instances in diesen zwei Availability Zones. Die Spalte "Availability Zones" des Load Balancers wurde für beide Availability Zones aktualisiert.

Aktueller Stand

Zurzeit befinden Sie sich bei der Erstellung Ihrer Architektur hier:

In Schritt 4: Starten einer Instance (p. 11) haben Sie eine Sicherheitsgruppe so eingerichtet, dass sämtlicher Datenverkehr über Port 80 (HTTP) eine Verbindung mit Ihrer Amazon EC2-Instance herstellt. Nachdem Sie nun einen Elastic Load Balancer erstellt haben, können Sie Ihre Sicherheitsgruppe so aktualisieren, dass nur eingehender HTTP-Datenverkehr vom Elastic Load Balancer zulässig ist. Fahren Sie mit Schritt 8: Aktualisieren Ihrer Amazon EC2-Sicherheitsgruppe (p. 34) fort.

Schritt 8: Aktualisieren Ihrer Amazon EC2-Sicherheitsgruppe

Abstract

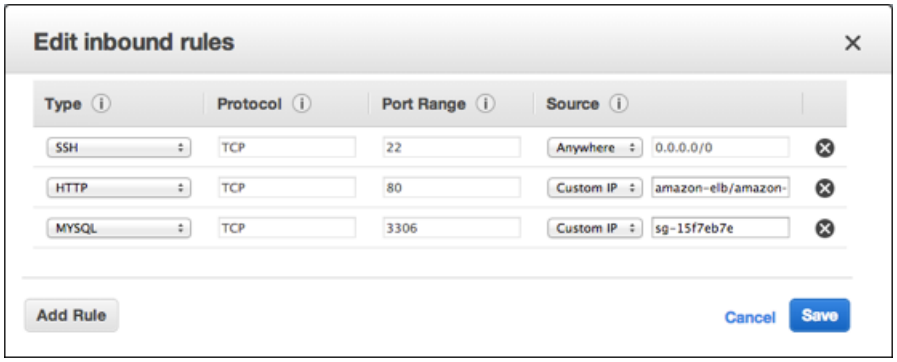
Beispiel zum Aktualisieren und Konfigurieren Ihrer Amazon EC2-Sicherheitsgruppe beim Hosten einer Web-Anwendung in einer Linux-Umgebung.

In Schritt 4: Starten einer Instance (p. 11) haben wir eine Sicherheitsgruppe erstellt, die HTTP über Port 80 aktiviert. Die Sicherheitsgruppe erlaubt dem gesamten Datenverkehr den Zugriff auf die Amazon EC2-Instance direkt über HTTP/80. Da Sie einen Elastic Load Balancer erstellt haben, ist es sicherer, nur diesem den Zugriff auf Ihre Amazon EC2-Instance zu gestatten. Wir haben zwei neue Instances mit unserer Auto Scaling-Gruppe gestartet. Damit die vom Benutzer abgerufenen Informationen synchron bleiben, sollen alle Instances nur auf die Informationen einer einzigen Datenbank zugreifen. Dazu müssen wir eine neue Regel einrichten, die es den Instances ermöglicht, die Datenbank auf der ursprünglichen Instance mithilfe von MySQL abzufragen. In diesem Schritt werden Sie Ihre Sicherheitsgruppe wie folgt aktualisieren: Nur der Load Balancer darf über HTTP/80 auf Ihre Amazon EC2-Instance zugreifen, und nur die Instances innerhalb der webappsecuritygroup akzeptieren über 3306/MySQL eingehenden Datenverkehr. Es gibt etliche Möglichkeiten zur Einrichtung einer Datenbank, einschließlich der Verwendung von Amazon RDS oder eines dedizierten Datenbankservers. Das Einrichten einer Datenbank kann im Rahmen dieses Dokuments nicht behandelt werden. Weitere Informationen zum Einrichten von Amazon RDS für eine Web-Anwendung finden Sie unter Schritt 8: Hinzufügen von Amazon RDS in *Hosten einer Getting Started with AWS-Webanwendung für Linux*.

Konfigurieren Sie Ihre Sicherheitsgruppe wie folgt:

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Klicken Sie im linken Navigationsbereich auf Load Balancers.
3. Wählen Sie den Load Balancer, 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 Ihre Sicherheitsgruppe zu ändern.
4. Klicken Sie im linken Navigationsbereich auf Security Groups.
5. Klicken Sie auf der Seite Security Groups auf die Sicherheitsgruppe`webappsecuritygroup`, die Sie im vorherigen Verfahren erstellt haben. Wenn die Sicherheitsgruppe nicht angezeigt wird, wählen Sie All security groups in der Filterliste.
6. Klicken Sie auf die Registerkarte Inbound und dann auf Edit.

7. In der Zeile, in der Port 80 (HTTP) angezeigt wird, wählen Sie Custom IP im Feld Source und geben den Namen der Sicherheitsgruppe ein, die zum Load Balancer gehört, beispielsweise **`amazon-elb/amazon-elb-sg`**.
8. Klicken Sie auf Add Rule.
9. Wählen Sie MYSQL aus der Liste Type.
10. Geben Sie **`webappsecuritygroup`** im Feld Source ein. Wählen Sie die Sicherheitsgruppen-ID für die webappsecuritygroup, nachdem sie angezeigt wurde.



11. Klicken Sie auf Speichern. Die Regeln für diese Sicherheitsgruppe werden nach dem Start der Instances erzwungen, die diese Regeln verwenden.

Nachdem Sie nun Ihre Amazon EC2-Sicherheitsgruppe konfiguriert haben, können Sie mit Schritt 9: Starten von Amazon EC2-Instances mithilfe von Auto Scaling (p. 35) fortfahren. Auto Scaling passt die Anzahl aktiver Instances an das jeweilige Datenverkehrsaufkommen an.

Schritt 9: Starten von Amazon EC2-Instances mithilfe von Auto Scaling

Auto Scaling startet und beendet Amazon EC2-Instances automatisch, basierend auf benutzerdefinierten Richtlinien, Zeitplänen und Alarmen. Mit Auto Scaling haben Sie die Möglichkeit, eine Flotte von Amazon Amazon EC2-Instances zu warten und an jedes auftretende Datenverkehrsaufkommen anzupassen. Sie können Auto Scaling auch zum gleichzeitigen Aktivieren mehrerer Instances in einer Gruppe verwenden.

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 Auto Scaling beispielsweise anweisen, zusätzliche Instances zu starten, wenn die CPU-Auslastung einer oder mehrerer der vorhandenen Instances zehn Minuten lang über 60 Prozent liegt. Oder Sie konfigurieren Auto Scaling so, dass die Hälfte der Instances Ihrer Website über das Wochenende beendet wird, wenn ein niedriger Datenverkehr zu erwarten ist.

Mit Auto Scaling stellen Sie sicher, dass die Instances in Ihrer Flotte optimale Leistung erbringen, damit Ihre Anwendungen weiterhin effizient ausgeführt werden. Auto Scaling-Gruppen funktionieren sogar über mehrere Availability Zones hinweg. Fällt eine Availability Zone aus, verteilt Auto Scaling den Datenverkehr automatisch an die Anwendungen in einer anderen Availability Zone. Auto Scaling gewährleistet, dass immer mindestens eine Instance korrekt ausgeführt wird. Weitere Informationen finden Sie im Abschnitt Auto Scaling.

In diesem Beispiel konfigurieren Sie die Basisinfrastruktur, die für meisten Anwendungen erforderlich ist, um Auto Scaling zu starten. Dazu ist Folgendes auszuführen:

- Erstellen einer Startkonfiguration
- Erstellen einer Auto Scaling-Gruppe
- Erstellen einer Richtlinie für Ihre Auto Scaling-Gruppe

Im Rahmen dieses Tutorials richten wir auf Amazon EC2 eine Anwendung ein, die mit der minimalen Anzahl von zwei Instances und der maximalen Anzahl von zwei Instances lastenverteilt und automatisch skaliert wird. Indem Sie für die minimale und die maximale Anzahl den gleichen Wert festlegen, stellen Sie sicher, dass stets die gewünschte Anzahl von Instances vorhanden ist, auch wenn eine Instance ausfällt. Bei der Erstellung Ihrer realen Website sollten Sie als bewährte Methode genügend Instances in verschiedenen Availability Zones starten, um den Verlust einer Availability Zone zu überstehen. Zudem gilt als Voraussetzung für die Verwendung der Auto Scaling-Funktion, dass die maximale Anzahl der Instances größer ist als die minimale.

Sie steuern die Größe Ihrer Flotte durch die Angabe einer maximalen Anzahl von Instances. In diesem Beispiel wird Auto Scaling so konfiguriert, dass bei einem Anstieg der Last eine Instance hinzugefügt wird. Wir legen die Richtlinie in diesem Thema fest und erstellen dann im nächsten Abschnitt einen CloudWatch-Alarm, der die Richtlinie anwendet, wenn der durchschnittliche NetworkOut-Wert 5 Minuten lang einen Grenzwert von 6 000 000 Bytes 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 Beenden von Instances für den Fall, dass die Last abnimmt.

Installieren Sie nun die Befehlszeilen-Tools für Auto Scaling, sofern Sie dies noch nicht getan haben. Weitere Informationen finden Sie unter Verwenden der Befehlszeilen-Tools im *Entwicklerhandbuch für Auto Scaling*. Sie verwenden Befehlszeilen-Tools, um Auto Scaling einzurichten.

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

1. Öffnen Sie ein Befehlszeilenfenster, indem Sie auf einem lokalen Windows-Computer auf Start klicken. Geben Sie im Feld Search den Befehl `cmd` ein und drücken Sie anschließend die Eingabetaste.
2. Bei der Startkonfiguration handelt es sich um eine Vorlage für die Instances, die Sie in Ihrer Auto Scaling-Gruppe starten. Zum Definieren der Startkonfiguration für dieses Beispiel verwenden wir den Befehl `as-create-launch-config`. Die folgenden Parameter definieren Ihre Startkonfiguration.

   - *image-id* ist die AMI-ID. Verwenden Sie die benutzerdefinierte AMI-ID, die Sie in Schritt 6: Erstellen eines benutzerdefinierten Amazon-Computerabbilds (AMI) (p. 27) erstellt haben.
   - *instance-type* enthält grundlegende Informationen zu der Instance, die gestartet werden soll, wie zum Beispiel Angaben über das Betriebssystem, den Arbeitsspeicher oder den lokalen Speicher. Verwenden Sie für dieses Beispiel den gleichen Instance-Typ wie beim ersten Start Ihrer Instance.
   - *key* ist das Schlüsselpaar für die Verbindungsherstellung mit Ihren Instances. Verwenden Sie das Schlüsselpaar, das Sie beim ersten Start Ihrer Instance erstellt haben.
   - *group* ist die Sicherheitsgruppe, in der Sie die Zugriffsregeln für Ihre Instance definiert haben. Verwenden Sie die Sicherheitsgruppe, die Sie beim ersten Start Ihrer Instance erstellt haben.

     Note

     Wenn Sie Ihre Instance in einer VPC gestartet und dort Ihre Sicherheitsgruppe erstellt haben, müssen Sie im Befehl die ID der Sicherheitsgruppe angeben und nicht den Namen.

   - *monitoring-disabled* aktiviert die grundlegende Überwachung anstelle der detaillierten Überwachung. Die detaillierte Überwachung ist standardmäßig aktiviert. Weitere Informationen über die grundlegende und die detaillierte Überwachung finden Sie unter Amazon CloudWatch.

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

```
PROMPT>as-create-launch-config MyLC --image-id ami-95ce1afc --instance-type
 t1.micro --group webappsecuritygroup --key mykeypair --monitoring-disabled
```

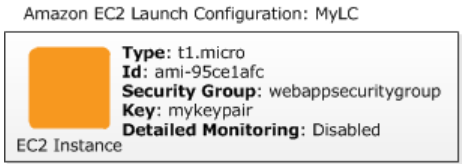
Auto Scaling antwortet wie im folgenden Beispiel (oder ähnlich):

```
OK-Created launch config
```

Note

Sie können die Befehle aus diesem Dokument kopieren und im Fenster der Eingabeaufforderung einfügen. Um die Inhalte im Fenster der Eingabeaufforderung einzufügen, klicken Sie mit der rechten Maustaste im Fenster und dann auf "Einfügen". Sollten die Befehle nicht ordnungsgemäß ausgeführt werden, sind sie wahrscheinlich nicht korrekt eingegeben worden.

Sie haben nun Ihre Startkonfiguration erstellt.



3. Zum Erstellen einer Auto Scaling-Gruppe, in der mehrere Amazon EC2-Instances gestartet werden können, verwenden Sie den `as-create-auto-scaling-group`-Befehl. Die folgenden Parameter definieren Ihre Auto Scaling-Gruppe.

   - *launch-configuration* ist der Name der Startkonfiguration, die Sie im vorangegangenen Schritt erstellt haben.
   - *availability-zones* bezeichnet die Availability Zones, in denen die Amazon EC2-Instances aus der Auto Scaling-Gruppe gestartet werden. In diesem Beispiel legen Sie zwei Availability Zones fest.

     Die Angabe mehrerer Availability Zones ist eine bewährte Methode zur Einrichtung fehlertoleranter Anwendungen. Fällt eine Availability Zone aus, wird der Verkehr in eine andere Availability Zone umgeleitet. Die in der Auto Scaling-Gruppe gestarteten Instances werden gleichmäßig auf die Availability Zones verteilt.
   - *min-size und max-size* bestimmen die minimale und die maximale Anzahl von Amazon EC2-Instances in der Auto Scaling-Gruppe. Indem Sie für die minimale und die maximale Anzahl den gleichen Wert angeben, bleibt die Anzahl der Instances in der Gruppe konstant. In diesem Beispiel wird sowohl die minimale und als auch die maximale Anzahl auf 2 festgelegt.
   - *load-balancer* ist der Name des Load Balancer, der für die Weiterleitung des Datenverkehrs zur Auto Scaling-Gruppe verwendet wird.

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

```
PROMPT>as-create-auto-scaling-group MyAutoScalingGroup --launch-configuration
 MyLC --availability-zones us-east-1b, us-east-1c  --min-size 2 --max-size
 2 --load-balancers MyLB
```

   Auto Scaling gibt Folgendes zurück:

```
OK-Created AutoScalingGroup
```

4. Zum Erstellen einer Richtlinie, die Ihre Instances-Flotte vergrößert, verwenden Sie den Auto Scaling-Befehl `as-put-scaling-policy`. Diese Richtlinie gilt für die Auto Scaling-Gruppe, die Sie im vorherigen Schritt erstellt haben. Die folgenden Parameter definieren Ihre Auto Scaling-Gruppe.

   - *auto-scaling-group* ist der Name der Auto Scaling-Gruppe, auf welche die Richtlinie angewendet werden soll. Verwenden Sie den Namen der Auto Scaling-Gruppe, die Sie im vorherigen Schritt erstellt haben.
   - *adjustment* ist der Wert, um den die Anzahl der Instances erhöht oder verringert werden soll. Wählen Sie 1 für dieses Beispiel.
   - *type* ist der Typ der zu erstellenden Richtlinie. Verwenden Sie für dieses Beispiel ChangeInCapacity, um die Flottengröße der Instances zu ändern.
   - *cooldown* ist die Anzahl der Sekunden, die im Anschluss an eine Aktion gewartet wird, bevor Auto Scaling die Bedingungen erneut auswertet.

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

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

   Auto Scaling antwortet wie im folgenden Beispiel (oder ähnlich):

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

   Note

   Um Zeit zu sparen, haben wir nur eine Richtlinie zum Hinzufügen einer einzelnen Instance erstellt. In den meisten Fällen würden Sie zusätzlich eine Richtlinie erstellen, die eine oder mehrere Instances beendet, sobald der Datenverkehr abnimmt. Auto Scaling kann die Anzahl der Instances verringern, wenn Ihre Anwendung die Ressourcen nicht benötigt, sodass Sie Geld sparen. Eine Richtlinie zum Beenden einer Instance erstellen Sie, indem Sie die gerade erstellte Richtlinie übernehmen, deren Namen ändern und statt des Anpassungswerts 1 den Wert -1 festlegen. Auf einem Windows-Computer verwenden Sie dazu **`"--adjustment=-1"`**.

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

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

5. Mit dem Befehl `as-describe-auto-scaling-groups` prüfen Sie, ob Ihre Auto Scaling-Gruppe vorhanden ist. Geben Sie Folgendes an der Eingabeaufforderung ein und drücken Sie anschließend die Eingabetaste:

   ```
   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  2         2         2
INSTANCE  INSTANCE-ID  AVAILABILITY-ZONE  STATE      STATUS   LAUNCH-CONFIG
INSTANCE  i-xxxxxxxx   us-east-1c         InService  Healthy  MyLC
INSTANCE  i-xxxxxxxx   us-east-1b         InService  Healthy  MyLC
```

Ihre Amazon EC2-Anwendung wurde als automatisch skalierte, lastenverteilte Anwendung gestartet.

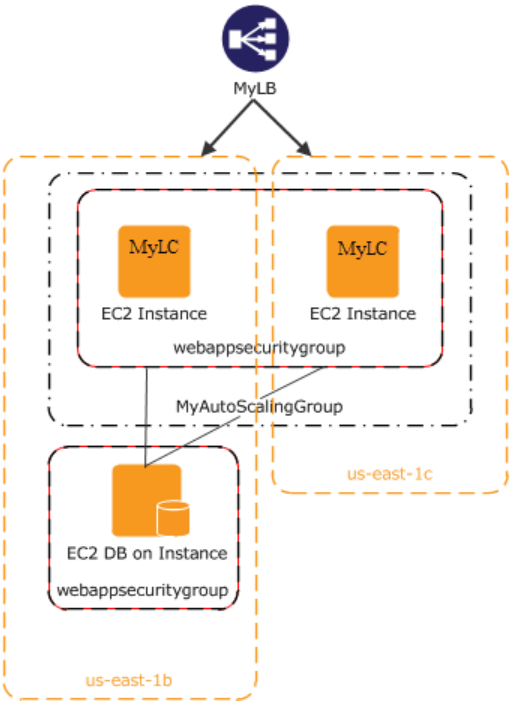
Weitere Informationen zu Auto Scaling finden Sie unter Auto Scaling Documentation.

Caution

Solange Amazon EC2-Instances ausgeführt werden, entstehen Ihnen Kosten. Informationen zum Beenden solcher Instances finden Sie unter Beenden Sie Ihre Amazon EC2-Instances in Ihrer Auto Scaling-Gruppe (p. 49).

Aktueller Stand

Zurzeit befinden Sie sich bei der Erstellung Ihrer Architektur hier:



Nachdem Sie nun die Auto Scaling-Gruppe erstellt haben und Ihre Amazon EC2-Instance verwendet werden kann, möchten Sie den Zustand Ihrer Instance überwachen. Im nächsten Schritt erstellen Sie einen Amazon CloudWatch-Alarm zum Nachverfolgen der soeben erstellten Auto Scaling-Richtlinie.

Schritt 10: Erstellen eines CloudWatch-Alarms

Amazon CloudWatch ist ein Web-Service, der Ihnen ermöglicht, verschiedene Metriken zu überwachen, zu verwalten oder zu veröffentlichen und Alarmaktionen auf der Grundlage dieser Metriken zu konfigurieren.

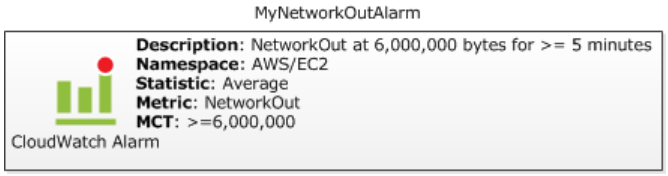
Amazon CloudWatch ermöglicht es Ihnen, System- und Anwendungsmetriken zu sammeln, zu analysieren und aufzurufen, um betriebliche und geschäftliche Entscheidungen schnell und sicher treffen zu können. Amazon CloudWatch sammelt automatisch Metriken über Ihre AWS-Ressourcen, wie zum Beispiel die Leistung Ihrer Amazon EC2-Instances. Sie können Ihre eigenen Metriken direkt in Amazon CloudWatch veröffentlichen.

Sie können Amazon CloudWatch 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. Sie können dann ermitteln, ob beispielsweise eine bestimmte Datenbank überlastet war. Wenn Sie das Problem behoben haben, können Sie mit Amazon CloudWatch überwachen, wie die Reaktionszeiten wieder in den normalen Bereich zurückkehren. Weitere Informationen über die Erstellung von CloudWatch-Alarmen erhalten Sie unter Erstellen von CloudWatch-Alarmen im Amazon CloudWatch Developer Guide.

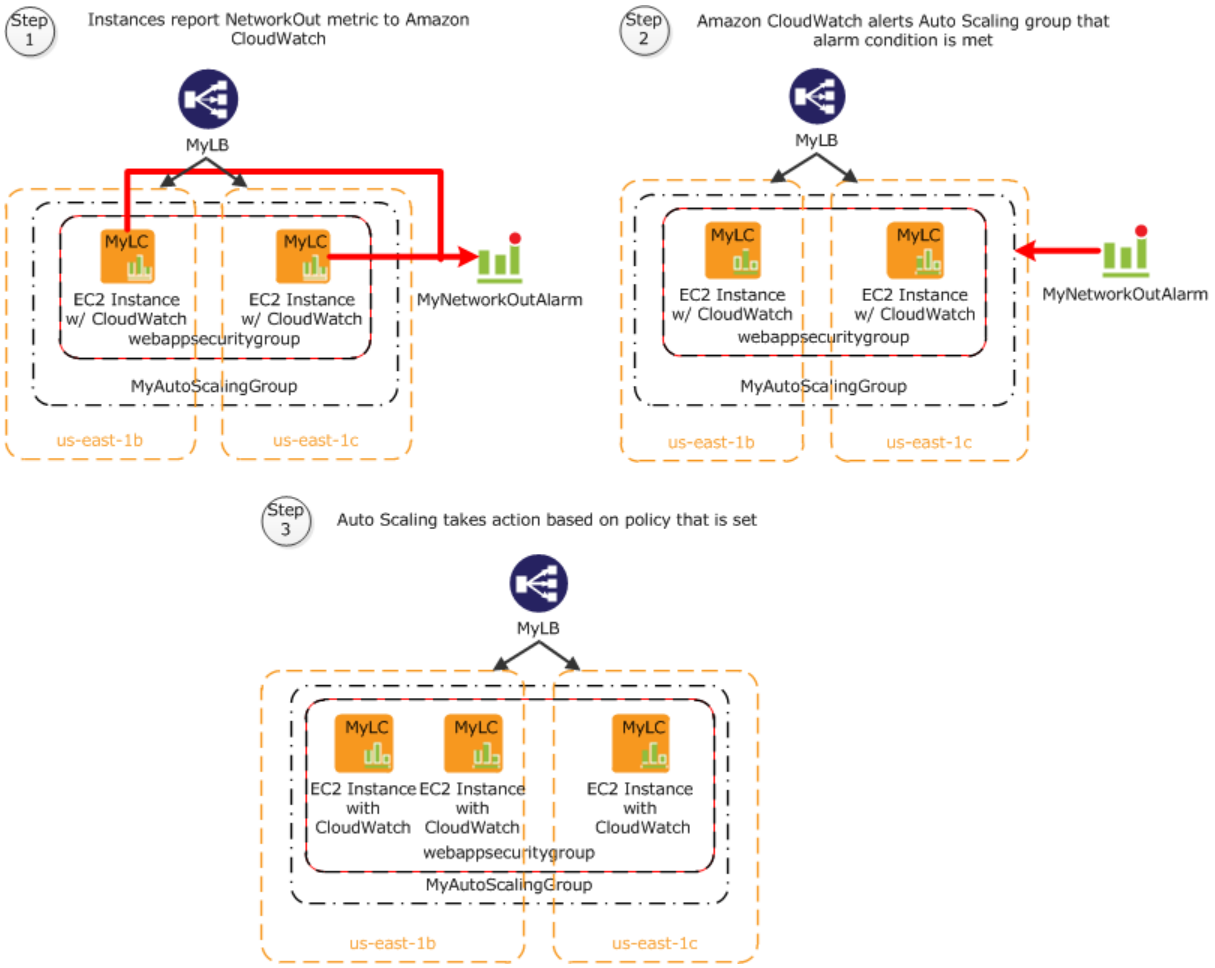
Amazon CloudWatch wird normalerweise dazu eingesetzt, die Leistungsfähigkeit und Effizienz Ihrer Anwendungen und Services zu wahren. Sie können damit beispielsweise herausfinden, wann Ihre Website am besten läuft. Dies kann dann der Fall sein, wenn der Netzwerkdatenverkehr in Ihren Amazon EC2-Instances unter einem bestimmten Grenzwert bleibt. Sie können dann eine Auto Scaling-Richtlinie erstellen, um sicherzugehen, dass immer die richtige Anzahl von Instances für den aktuellen Datenverkehr vorhanden ist.

In der vorherigen Aufgabe haben wir eine Auto Scaling-Richtlinie erstellt, um die Anzahl der laufenden Instances zu vergrößern. In dieser Aufgabe verknüpfen wir diese Richtlinie mit einer Alarmaktion. Bei Auslösung des Alarms wird die Auto Scaling-Richtlinie benachrichtigt und nimmt die entsprechenden Änderungen an Ihren Ressourcen vor.

Sie erstellen einen Alarm mit den folgenden Eigenschaften:



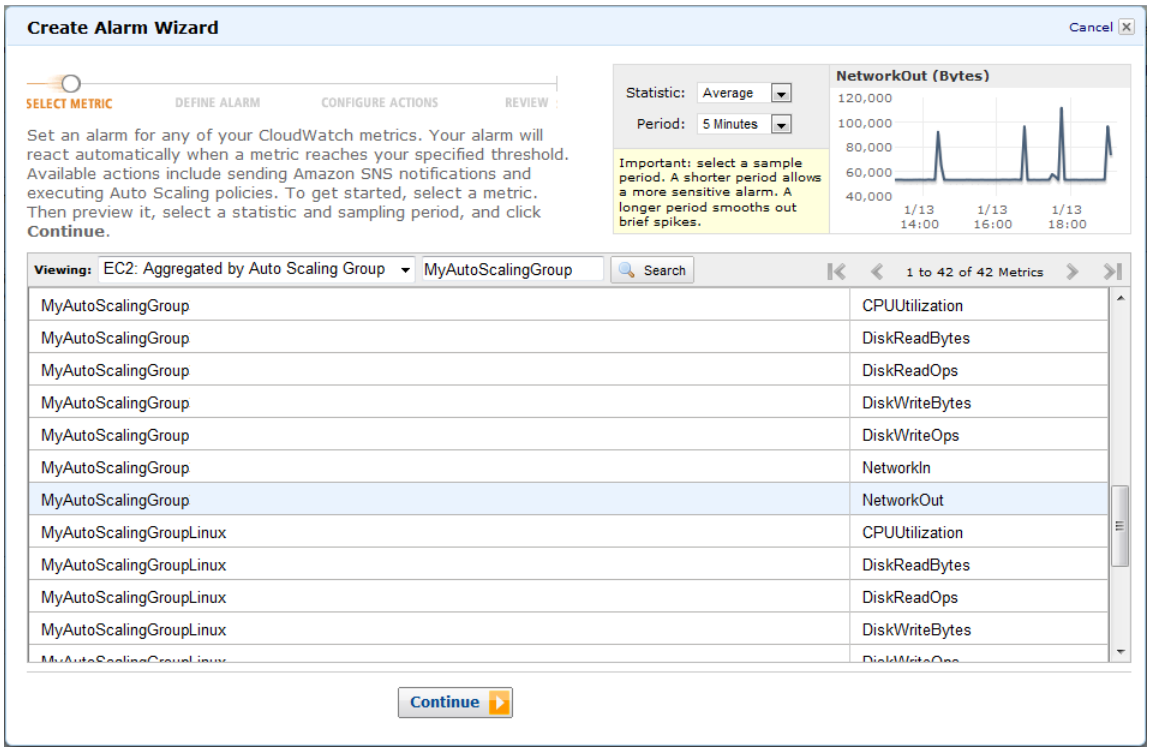
Im folgenden Diagramm wird gezeigt, wie Amazon CloudWatch und Auto Scaling zusammenarbeiten. Die Amazon EC2-Instance meldet ihre NetworkOut-Metrik an Amazon CloudWatch. Amazon CloudWatch löst einen Alarm aus, wenn der festgelegte Grenzwert überschritten wird, und meldet dies der Auto Scaling-Gruppe. Die Auto Scaling-Gruppe trifft dann Maßnahmen, die auf der festgelegten Richtlinie basieren.



In diesem Thema werden Sie schrittweise durch die Erstellung eines CloudWatch-Alarms geführt, der die Anwendung warnt, wenn der Grenzwert überschritten wird. Um für diese Anleitung 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 zur Benachrichtigung von Auto Scaling erstellen, wenn dieses eine Instance beenden muss. Weitere Informationen über Amazon CloudWatch finden Sie auf der Detailseite von Amazon CloudWatch.

Erstellen Sie Amazon CloudWatch-Alarme wie folgt:

1. Wählen Sie eine Metrik für den Alarm:
   a. Öffnen Sie die Amazon CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
   b. Klicken Sie im linken Navigationsbereich auf Alarm.
   c. Klicken Sie im Detailbereich auf Create Alarm.
   d. Wählen Sie im Assistenten Create Alarm Wizard auf der Seite Select Metric in der Liste Viewing die Option EC2: Aggregated by Auto Scaling Group.



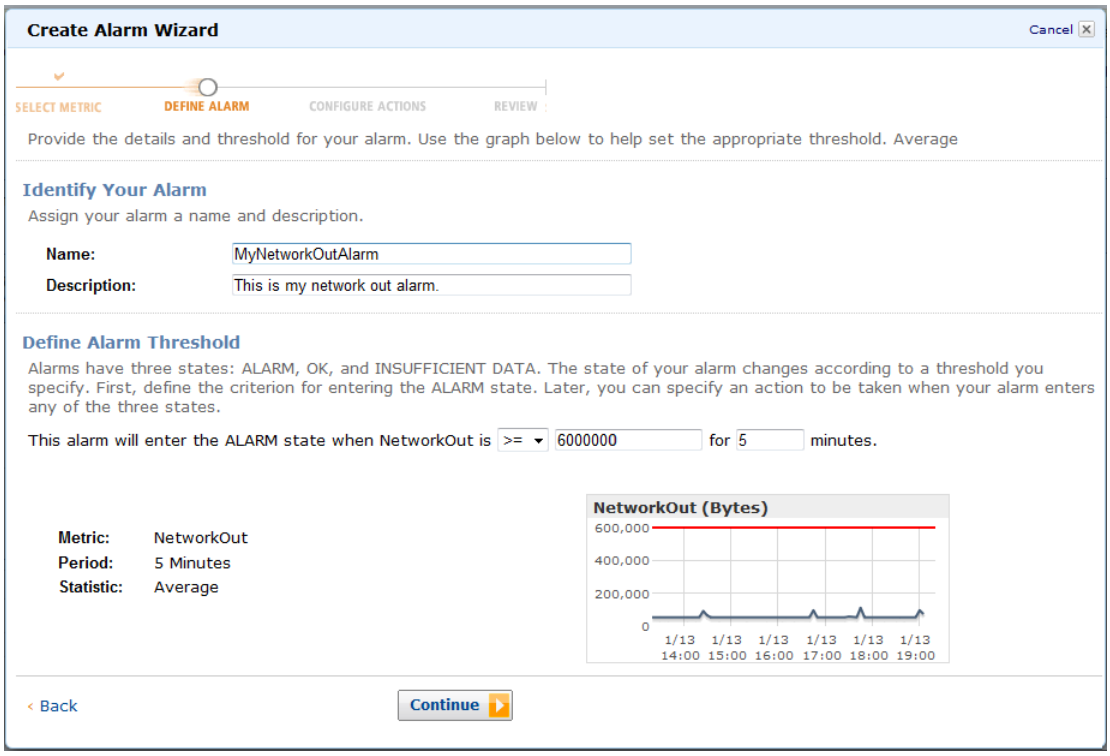
e. Klicken Sie in die Zeile MyAutoScalingGroup/NetworkOut und dann auf Continue.

> **Note**
>
> Es kann bis zu 15 Minuten dauern, bis die Auto Scaling-Gruppe in der Liste angezeigt wird. Wenn die Auto Scaling-Gruppe nicht angezeigt wird, warten Sie 15 Minuten und versuchen Sie es erneut.

2. Definieren Sie den Alarm:

   Führen Sie auf der Seite Define Alarm des Assistenten Create Alarm folgende Schritte aus und klicken Sie dann auf Continue:

   - Geben Sie im Feld Name den Eintrag **`MyNetworkOutAlarm`** ein.
   - Geben Sie im Feld Description eine Beschreibung ein.
   - Klicken Sie im Abschnitt Define Alarm Threshold auf **>=**. Geben Sie dann im ersten Feld **`6000000`** und im Minutenfeld **`5`** ein. Sie können für Ihre eigene Anwendung einige Lasttests ausführen, um festzustellen, welche Werte am sinnvollsten sind.



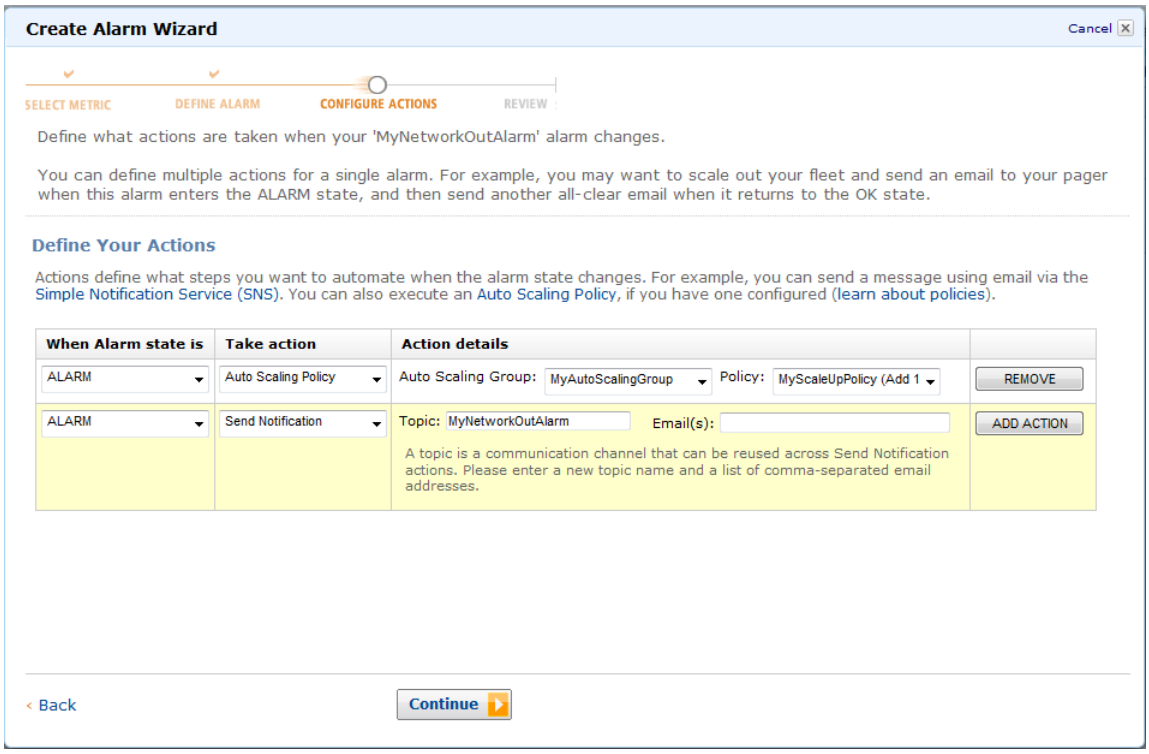
3. Definieren Sie die Aktionen:

   a. Führen Sie auf der Seite Configure Actions des Assistenten Create Alarm folgende Schritte aus und klicken Sie dann auf Add Action.

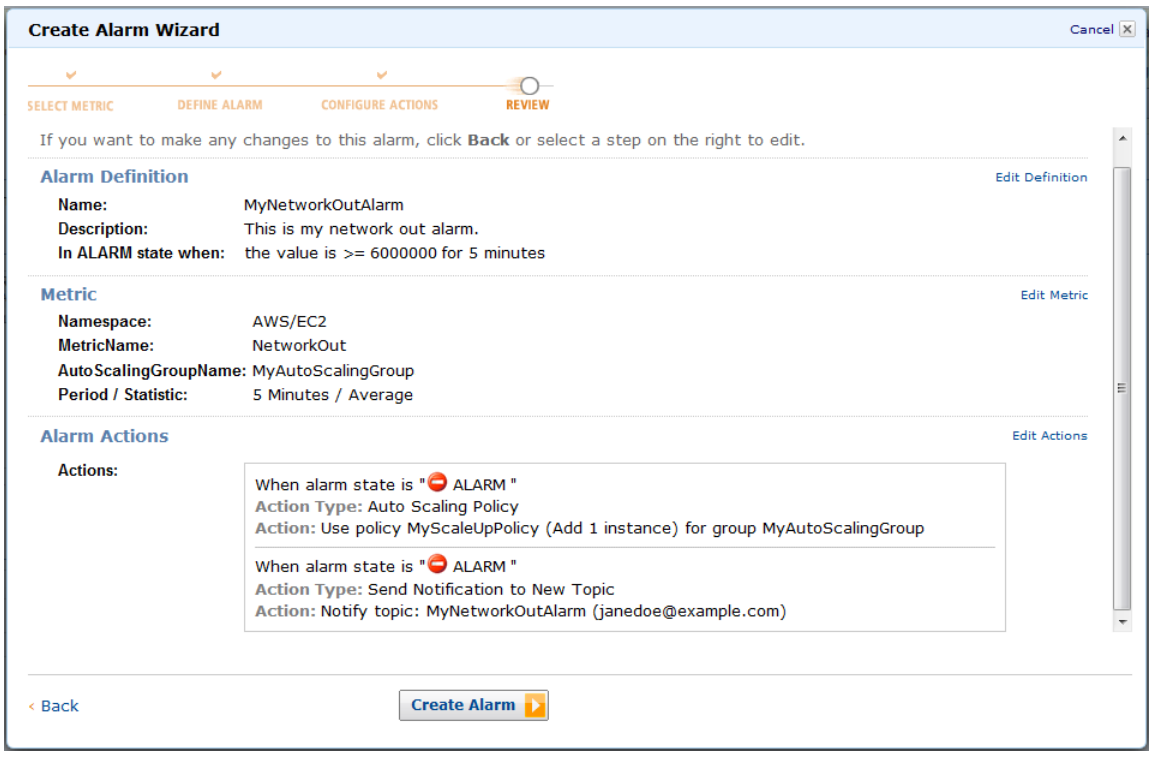
      - Klicken Sie unter When Alarm state is auf ALARM.
      - Klicken Sie unter der Liste Take Action auf Auto Scaling Policy.
      - Klicken Sie in der Liste Auto Scaling Group auf MyAutoScalingGroup.
      - Klicken Sie in der Liste Policy auf MyScaleUpPolicy (Add 1 instance).

   b. Führen Sie folgende Schritte aus und klicken Sie dann auf Continue.

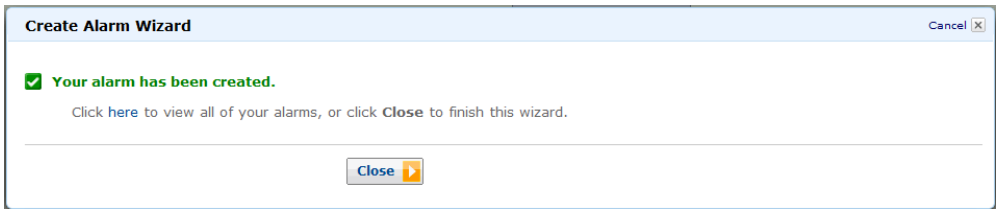
      - Klicken Sie in der neu erstellten Zeile unter When Alarm state is auf ALARM.
      - Klicken Sie unter der Liste Take Action auf Send Notification.
      - Klicken Sie im Feld Topic auf Create New Email Topic und geben Sie dann einen Namen für das Thema ein.
      - Geben Sie im Feld Email(s) die E-Mail-Adresse ein, an die die Benachrichtigungen gesendet werden sollen.



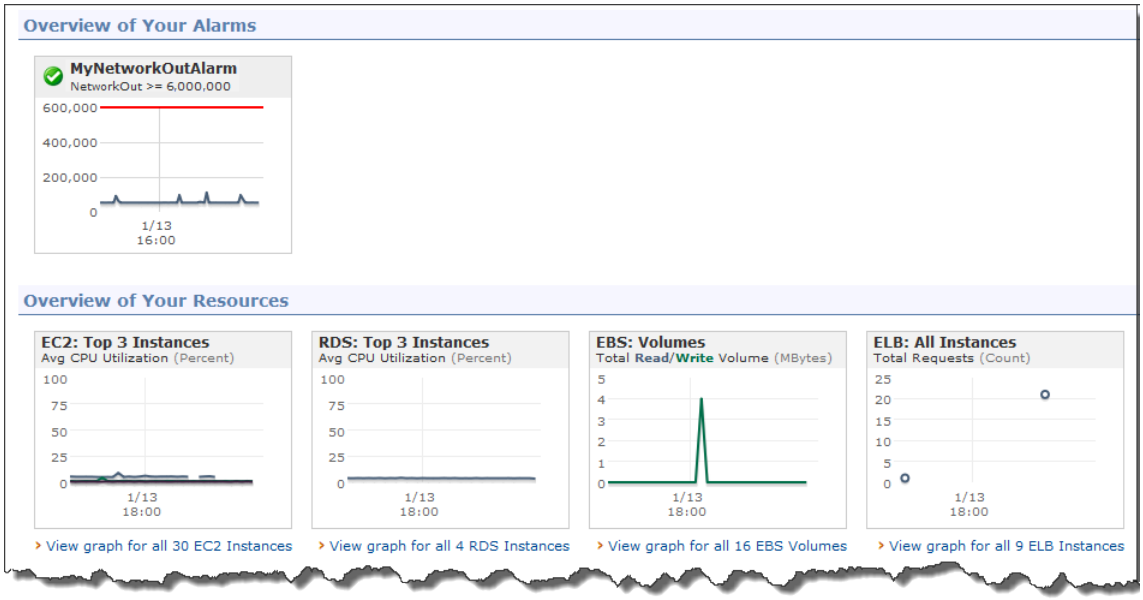
4. Überprüfen Sie die Einstellungen auf der Seite Review. Wenn die Einstellungen richtig sind, klicken Sie auf Create Alarm.



5. Klicken Sie auf der Bestätigungsseite auf Close.



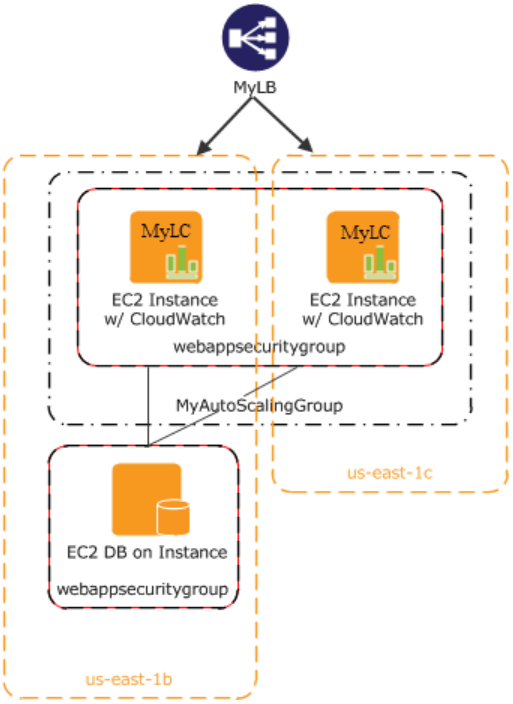
Der neue Alarm wird nun auf der Dashboard-Seite der Amazon CloudWatch-Konsole in der Liste angezeigt.



Wenn Sie eine MyScaleDownPolicy-Richtlinie erstellen, können Sie mit den gleichen Schritten einen anderen Alarm erstellen.

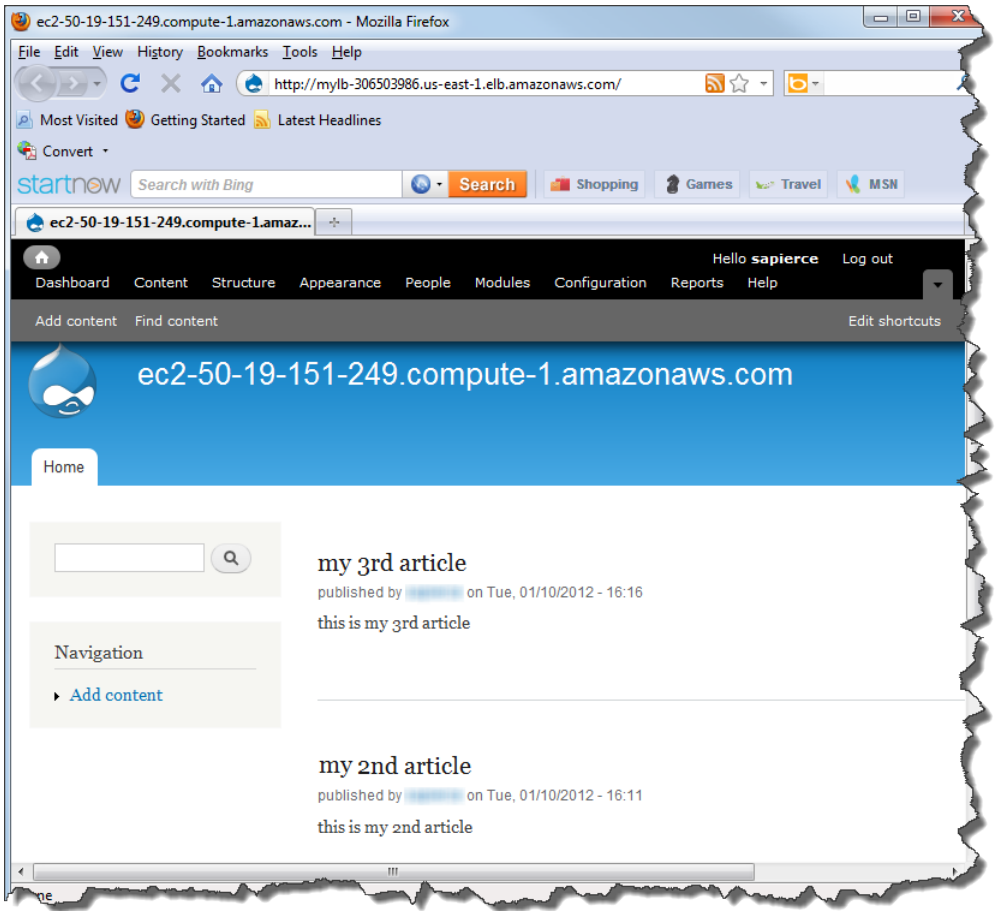
Aktueller Stand

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



Herzlichen Glückwunsch! Sie haben mithilfe grundlegender AWS-Bausteine Ihre Web-Anwendung erfolgreich für EC2 bereitgestellt. Zum Überprüfen einer reibungslosen Funktion gehen Sie wie folgt vor:

1. Aktualisieren Sie Ihren Browser. Sie sollten nicht mehr auf Ihre Website zugreifen können, da Sie Ihre Sicherheitsgruppe so geändert haben, dass ein Zugriff nur noch über den Elastic Load Balancer möglich ist.
2. Geben Sie die öffentliche DNS-Adresse Ihres Elastic Load Balancer ein, den Sie in Schritt 7: Erstellen eines Elastic Load Balancers (p. 27) notiert haben, um zu überprüfen, ob Ihre Anwendung angezeigt wird.



In diesem Tutorial haben Sie erfahren, wie Sie Ihre Webanwendung unter Verwendung der folgenden AWS-Produkte bereitstellen:

- Amazon EC2 zum Ausführen Ihrer Anwendung
- Elastic Load Balancing für den Lastenausgleich des Datenverkehrs über Ihre laufenden Instances
- Auto Scaling zum automatischen Hinzufügen und Beenden von Instances gemäß der festgelegten Richtlinien
- Amazon CloudWatch zum Überwachen Ihrer Instances und zur Benachrichtigung, wenn die festgelegten Grenzwerte überschritten werden

Wenn Sie einen tieferen Einblick in AWS-Services gewonnen haben und genauer wissen, wie Sie sie verwenden möchten, können Sie auch einen einfacheren Weg zur Bereitstellung Ihrer Anwendung finden. AWS CloudFormation unterstützt Sie bei der Bereitstellung von Ressourcen in AWS und Sie müssen die Reihenfolge, in der die AWS-Services bereitgestellt werden, und die erforderlichen Feinheiten, damit die Abhängigkeiten funktionieren, nicht selbst herausfinden. Wenn Sie lernen möchten, wie Sie Beispielvorlagen mit den in diesem Tutorial verwendeten Services erstellen, rufen Sie Auto Scaling-Gruppe mit LoadBalancer, Auto Scaling-Richtlinien und CloudWatch-Alarme im *AWS CloudFormation User Guide* auf.

Wenn Sie die AWS-Ressourcen nicht mehr verwenden, beenden Sie sie, damit die Nutzung nicht in weiter in Rechnung gestellt wird. Fahren Sie mit Schritt 11: Bereinigen (p. 48) fort.

Schritt 11: Bereinigen

Abstract

Nachdem Sie Ihre Webanwendung bereitgestellt haben, müssen Sie Ihre Ressourcen bereinigen und Ihre Instances beenden, um zusätzliche Gebühren zu vermeiden.

Topics

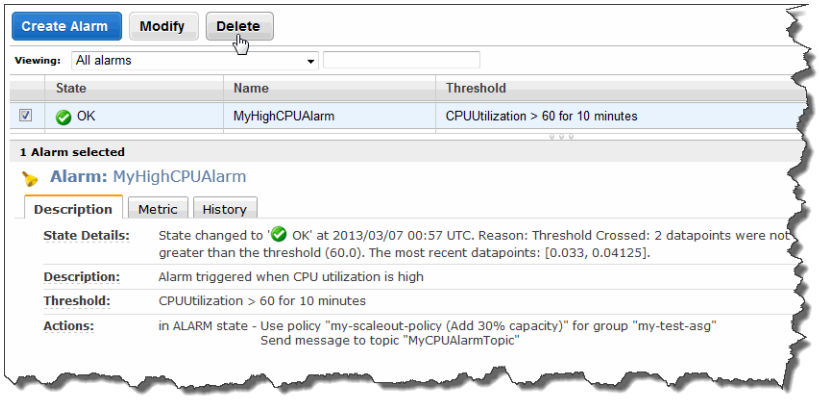
Herzlichen Glückwunsch! Sie haben soeben Ihre Webanwendung bereitgestellt. Um weitere Gebühren zu vermeiden, beenden Sie Ihre Umgebungen und bereinigen Sie Ihre Ressourcen.

Löschen eines CloudWatch-Alarms

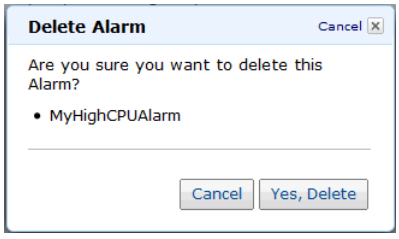
Wenn Sie entscheiden, dass Sie den Alarm nicht mehr benötigen, können Sie ihn löschen.

Löschen Sie den Alarm wie folgt:

1. Öffnen Sie die Amazon CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Klicken Sie im linken Navigationsbereich auf Alarms.
3. Aktivieren Sie das Kontrollkästchen neben dem zu löschenden Alarm und klicken Sie dann auf Delete.



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



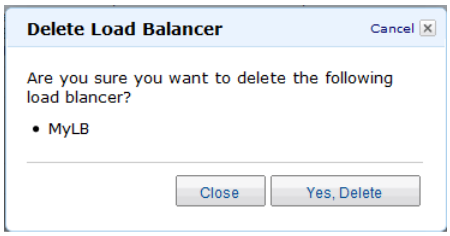
Löschen des Elastic Load Balancers

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 im linken Navigationsbereich auf Load Balancers.
3. Aktivieren Sie das Kontrollkästchen neben dem zu löschenden Load Balancer und klicken Sie dann auf Delete.

4. 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.

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
```

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-xxxxxxxx   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
```

Beenden Ihrer Instance

Abstract

Um Kosten für Ihre AWS-Services möglichst niedrig zu halten, beenden Sie nicht mehr benötigte Amazon EC2-Instances, auch wenn sie inaktiv sind.

Sobald Ihre Instanz startet, berechnet Ihnen AWS jede Stunde oder angebrochene Stunde, in der die Instanz ausgeführt wird, auch wenn sie inaktiv ist. Sie können die Instance beenden, sodass Ihnen keine Kosten mehr dafür entstehen. Da diese Instance kein Teil Ihrer Auto Scaling-Gruppe ist, müssen Sie sie manuell beenden.

Beenden einer Instanz

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Klicken Sie im linken Navigationsbereich auf Instances.
3. Klicken Sie mit der rechten Maustaste auf die Instance und klicken Sie dann auf Terminate.
4. Wenn Sie zur Bestätigung aufgefordert werden, klicken Sie auf Yes, Terminate. Sobald sich der Status der Instance zu shutting down oder terminated ändert, fallen für diese Instanz keine Gebühren mehr an.

Löschen eines Schlüsselpaars

Abstract

Das Löschen eines Schlüsselpaars ist optional, da Ihnen die Beibehaltung eines Schlüsselpaars nicht berechnet wird. Sie können das Schlüsselpaar zu einem späteren Zeitpunkt wiederverwenden.

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

Löschen Sie die Schlüsselpaare wie folgt:

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Klicken Sie im linken Navigationsbereich auf Key Pairs.
3. Aktivieren Sie das Kontrollkästchen neben dem zu löschenden Schlüsselpaar und klicken Sie dann auf Delete.
4. 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 im linken Navigationsbereich auf Security Groups.
3. Wählen Sie im Bereich "Details" unter Security Groups eine Sicherheitsgruppe aus, die Sie löschen möchten, und klicken Sie dann auf Delete.
4. Klicken Sie auf Yes, Delete.

Preise

Topics

AWS – Einfacher Monatsrechner schätzt Ihre monatliche Rechnung. Er bietet eine Kostenaufschlüsselung pro Service sowie eine Schätzung der monatlichen Gesamtkosten. Sie können den Rechner 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.

Note

Die AWS-Preise, die Sie in dieser Dokumentation sehen, sind zum Zeitpunkt der Veröffentlichung aktuell. Eine Übersicht mit aktuellen Preisinformationen finden Sie unter AWS Service Pricing Overview. Weitere Informationen zur Preisgestaltung bei AWS erhalten Sie unter How AWS Pricing Works.

Amazon EC2-Kostenaufschlüsselung

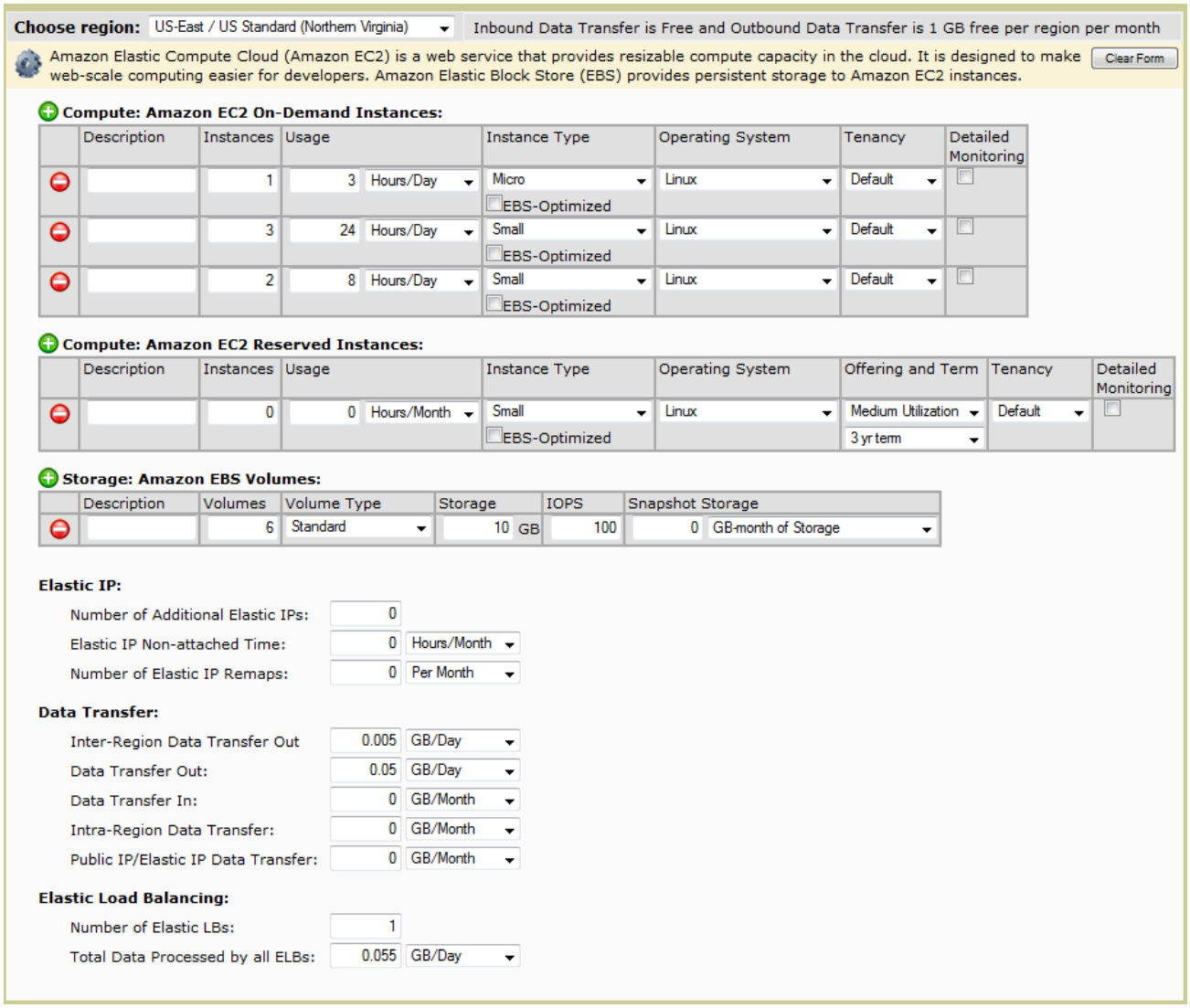
Abstract

Preise und Kostenaufschlüsselung für das Hosten von Web-Anwendungen sowie Empfehlungen zu Instances, die für Amazon EC2 erforderlich sind.

In der folgenden Tabelle sind die Merkmale für Amazon EC2 angegeben, die wir für die Architektur dieser Web-Anwendung ermittelt haben. In diesem Beispiel gehen wir davon aus, dass Sie eine Produktionsumgebung mit drei bis sechs Instances benötigen. Drei Instances sind fortlaufend aktiv, zwei weitere Instances sind für hohes Datenverkehrsaufkommen zu Spitzenzeiten erforderlich, und eine Instance führt nächtliche Sicherungen aus.

Merkmal	Metrik	Beschreibung
Server-Betriebszeit	3 Instances werden 24 Stunden/Tag ausgeführt 2 Instances werden 8 Stunden/Tag ausgeführt 1 Instance wird 3 Stunden/Tag ausgeführt	Ausgehend von durchschnittlichen 30,5 Tagen in einem Monat werden die Vollzeit-Instances 732 Stunden/Monat, die Instances für Spitzenzeiten mit hohem Datenverkehrsaufkommen 244 Stunden/Monat und die Instance für nächtliche Sicherung 91,5 Stunden/Monat ausgeführt.
Computermerkmale	1 T1 Micro-Instance 5 M1 Small-Instances	Micro: 613 MB Speicher, bis zu 2 EC2-Recheneinheiten (für kurze periodische Bursts), nur Elastic Block Store (EBS)-Speicher, 32-Bit- oder 64-Bit-Plattform 1,7 GB Arbeitsspeicher, 1 EC2 Compute Unit (1 virtueller Kern mit 1 EC2-Recheneinheit), 160 GB lokaler Instance-Speicher, 32-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 E/A\Sek.	Das AMI ist EBS-gestützt. Das Volume verfügt über 10 GB bereitgestellten Speicher und unterstützt 100 E/A-Anforderungen pro Sekunde.
Datenübertragung	Eingehende Daten: 0,005 GB/Tag Ausgehende Daten: 0,05 GB/Tag	Es gibt ungefähr 1 000 Zugriffe pro Tag. Jede Antwort umfasst ungefähr 50 KB und jede Anforderung ungefähr 5 KB.
Skalierung der Instances	Zwischen 3 und 6 Instances	Drei Instances sind fortlaufend aktiv, zwei weitere Instances sind für hohes Datenverkehrsaufkommen zu Spitzenzeiten erforderlich, und eine Instance führt nächtliche Sicherungen aus.
Elastic Load Balancing	Nutzung in Stunden: 732 Stunden/Monat Verarbeitete Daten: 1 525 GB/Monat	Elastic Load Balancing wird 24 Stunden/Tag, 7 Tage/Woche verwendet Elastic Load Balancing verarbeitet insgesamt 0,055 GB/Tag (eingehende und ausgehende Daten)

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



Die monatlichen Gesamtkosten ergeben sich aus der Summe der Kosten für die aktiven Instances, die Amazon Elastic Block Store-Volumes, die E/A-Anforderungen, den Elastic Load Balancer und die von den Elastic Load Balancern verarbeiteten Daten. Da wir eine grundlegende Überwachung sowie nur eine Metrik und einen Alarm für die Amazon EC2-Instances verwendet haben, fallen keine zusätzlichen Gebühren für die Amazon CloudWatch-Überwachung an.

Variable	Formel	Berechnung
Instance-Kosten	Instance-Kosten pro Stunde	0,060 USD
	Anzahl der Instances	3
	x Server-Stunden	x 732
	----------	-------
		$131.76

Variable	Formel	Berechnung
Instance-Kosten	Instance-Kosten pro Stunde Anzahl der Instances x Server-Stunden ----------	0,060 USD 2 x 244 ------- $29.28
Instance-Kosten	Instance-Kosten pro Stunde Anzahl der Instances x Server-Stunden ----------	$0.02 1 x 91,5 ------- $1.83
Zusätzlicher Speicher	Speichergebühr x Speichermenge (GB) + (E/A-Anforderungsrate x Sekunden pro Monat x Anforderungsrate (pro 1M-Anforderungen)) x Anzahl der Volumes _____	$0,10 x 10 + (100 x ~2,6M x 0,10 USD)/1M x 6 ---------- $164.11
Elastic Load Balancing	Genutzte Stunden x stündliche Gebühr + (verarbeitete Daten (GB) x Verarbeitungsgebühr) ----------	732 x $0,025 + 1,6775 x $0,008 ---------- $18.31
Gesamtkosten pro Monat		$345.29

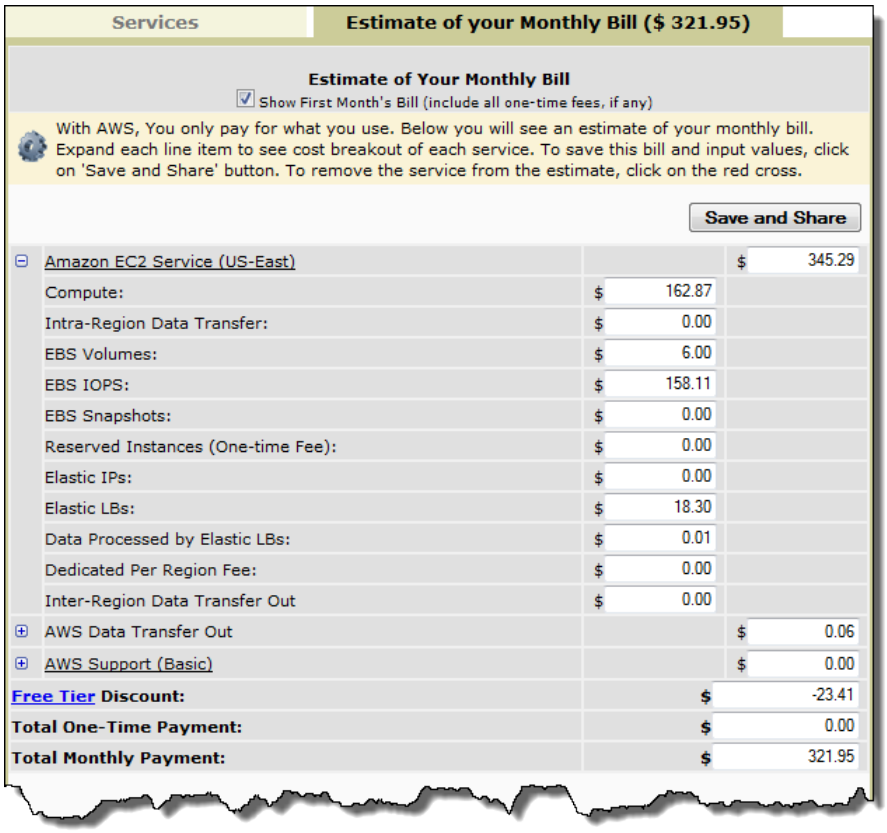
Summe aller Kosten

Um die Gesamtkosten für dieses Beispiel zu berechnen, fügen wir die Kosten der Amazon EC2-Instances und der ausgehenden Datenübertragungen von AWS hinzu und ziehen dann sämtliche Rabatte ab, die im kostenlosen Nutzungskontingent von AWS gewährt werden. Weitere Informationen über das kostenlose Nutzungskontingent und wie Sie herausfinden, ob Sie dafür berechtigt sind, finden Sie unter Getting Started with AWS Free Usage Tier.

Die gesamten ausgehenden Datenübertragungen von AWS setzen sich aus den aggregierten ausgehenden Datenübertragungen aller Amazon EC2-Instances zusammen. Für Amazon EC2 sind dies 0,05 GB pro Tag, was ungefähr 1,525 GB pro Monat ausmacht. 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) + (ausgehende Daten (GB) x Gebühr für ausgehende Daten) ----------	0,1525 x 0,00 USD + (0,525) x 0,12 USD ---------- Danach 0,06 USD

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



Laut der Berechnungen des Rechners belaufen sich die Gesamtkosten für Amazon EC2 auf $321,95.

Weitere Möglichkeiten zur Kosteneinsparung

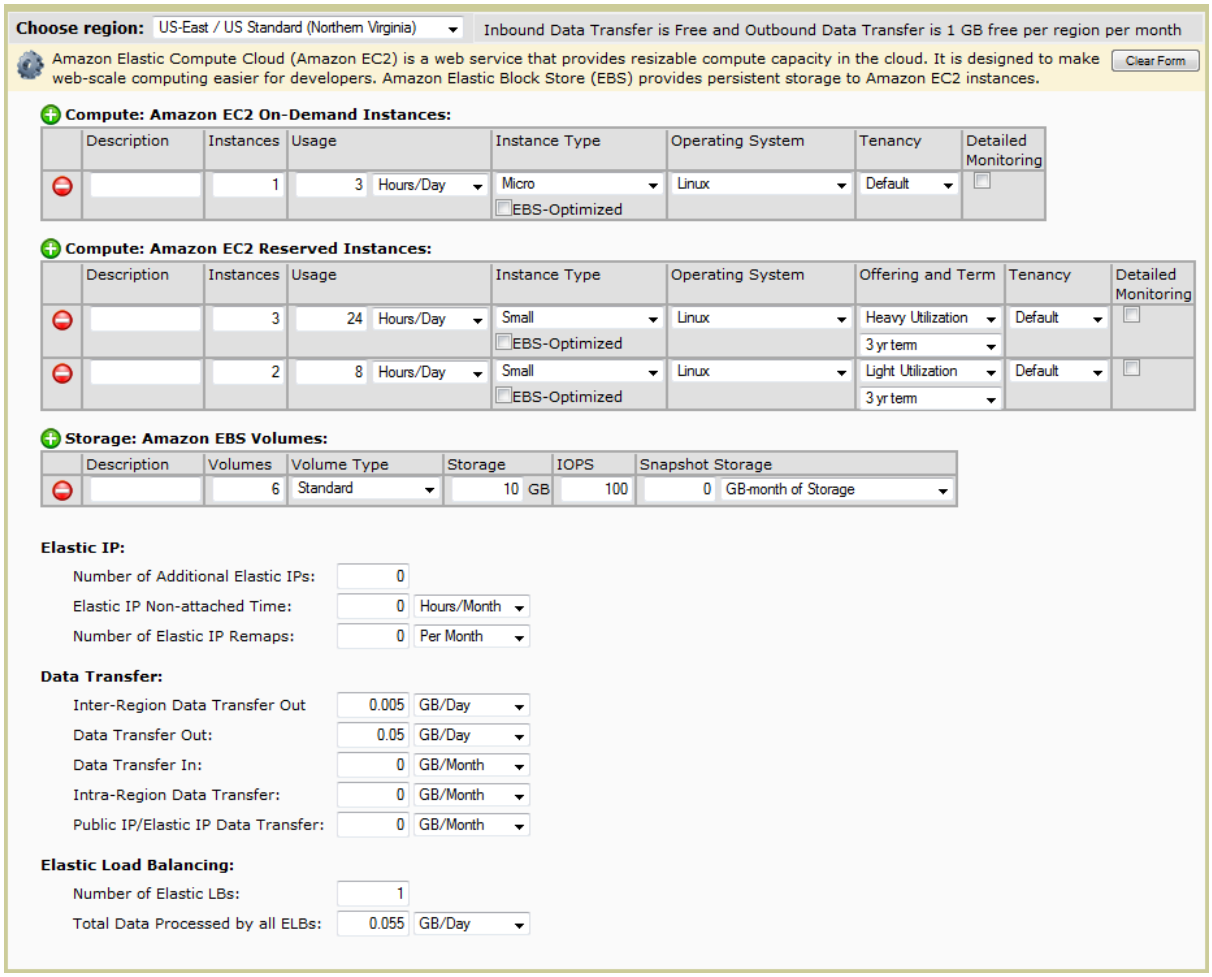
Im besprochenen Beispiel zur Bereitstellung haben wir On-Demand-Instances für alle sechs Instances verwendet. Bei On-Demand-Instances werden nur vom Start bis zum Beenden einer Instance Gebühren verrechnet. Wenn Sie planen, Ihre Instances über einen längeren Zeitraum auszuführen, können Sie Geld sparen, indem Sie sie reservieren.

Sie erhalten Reserved Instances, indem Sie eine geringe, einmalige Zahlung für jede Instance leisten, die Sie reservieren möchten. Dafür wird Ihnen ein beträchtlicher Rabatt auf die stündliche nutzungsabhängige Gebühr gewährt. Wenn Sie ungefähr wissen, wie intensiv Sie Ihre laufenden Amazon EC2-Instances nutzen werden, können Sie sogar noch mehr Geld sparen, indem Sie aus den Optionen zur

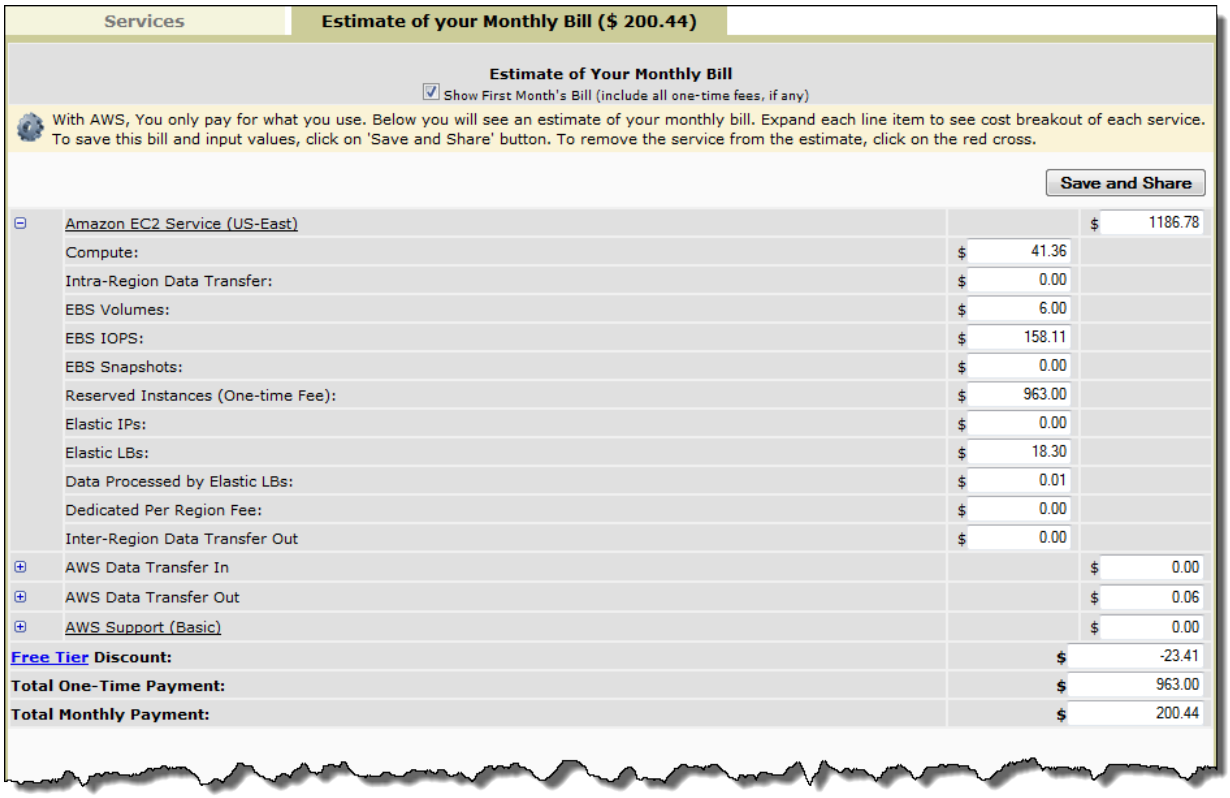
niedrigen, mittleren und hohen Auslastung der Reserved Instances auswählen. Bei hoher Auslastung zahlen Sie vorab eine höhere Gebühr, die stündliche Nutzungsgebühr fällt jedoch geringer als bei mittlerer und niedriger Auslastung der Reserved Instances aus. Für eine niedrige Auslastung wird die niedrigste Vorabgebühr verrechnet, die stündliche Gebühr ist jedoch höher als für die mittlere und hohe Auslastung der Instances. Im vorherigen Beispiel werden drei der Instances fortlaufend ausgeführt. Dies ist ein ideales Beispiel für eine starke Auslastung von Reserved Instances. Zwei Instances laufen nur während des Hauptdatenverkehrs, also zirka ein Drittel der Zeit. Diese Instances eignen sich ideal für Reserved Instances mit niedriger Auslastung. Da die Instance, die nächtliche Sicherungen durchführt, nur wenige Stunden am Tag läuft, können Sie sie als On-Demand-Instance ausführen.

Reserved Instances können für eine einjährige oder dreijährige Laufzeit erworben werden. Die dreijährige Laufzeit bietet weitere Ersparnisse im Vergleich zur einjährigen Laufzeit. Weitere Informationen über Reserved Instances erhalten Sie unter Amazon EC2-Reserved Instances. In der folgenden Tabelle finden Sie den Kostenvergleich von On-Demand- und Reserved Instances über einen Zeitraum von drei Jahren.

Aktualisieren wir den Rechner unter Verwendung derselben Merkmale und Metriken wie im obigen Beispiel, sodass eine hohe und niedrige Auslastung wie im folgenden Diagramm ausgegeben wird.



Die monatlichen Gesamtkosten werden wie im vorherigen Beispiel berechnet, mit der Ausnahme, dass eine zusätzliche Einmalgebühr für Reserved Instances enthalten ist. Im folgenden Diagramm sind die Gesamtkosten aufgeführt.



Services | Estimate of your Monthly Bill ($ 200.44)

Estimate of Your Monthly Bill

Show First Month's Bill (include all one-time fees, if any)

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.

Save and Share

⊖	Amazon EC2 Service (US-East)		$ 1186.78
	Compute:	$ 41.36	
	Intra-Region Data Transfer:	$ 0.00	
	EBS Volumes:	$ 6.00	
	EBS IOPS:	$ 158.11	
	EBS Snapshots:	$ 0.00	
	Reserved Instances (One-time Fee):	$ 963.00	
	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	
⊕	AWS Data Transfer In		$ 0.00
⊕	AWS Data Transfer Out		$ 0.06
⊕	AWS Support (Basic)		$ 0.00
Free Tier Discount:			$ -23.41
Total One-Time Payment:			$ 963.00
Total Monthly Payment:			$ 200.44

In der folgenden Tabelle werden die Gesamtkosten für die gemischte Verwendung von Reserved Instances mit hoher und niedriger Auslastung mit den Gesamtkosten für On-Demand-Instances verglichen.

Instanz	Monatliche Kosten	Einmalige Gebühr	Gesamtkosten (3 Jahre)
6 On-Demand-Instances	$345.29	–	$12430.44
1 On-Demand-Instance 3 Reserved Instances mit hoher Auslastung 2 Reserved Instances mit geringer Auslastung	$200.44	$963.00	$8178.84

Wie aus der Tabelle hervorgeht, beträgt die Kosteneinsparung aufgrund der gemischten Verwendung von Reserved Instances mit hoher und niedriger Auslastung in diesem Beispiel ungefähr 30 %. Weitere Informationen zur AWS-Preisgestaltung finden Sie im Whitepaper http://media.amazonwebservices.com/AWS_Pricing_Overview.pdf.

Sie können ebenfalls Geld sparen, indem Sie Spot-Instances verwenden. Spot-Instances sind nicht genutzte Amazon EC2-Kapazitäten, auf die Sie bieten. Für Instances gilt der Spot Price, der von Amazon EC2 festgelegt wird, und der abhängig von Angebot und Nachfrage für Spot-Instance-Kapazitäten regelmäßigen Schwankungen unterliegt. Wenn Ihr Höchstgebot den aktuellen Spot Price übersteigt, wird diesem stattgegeben, und Ihre Instances werden so lange ausgeführt, bis Sie sie entweder beenden, oder bis der Spot Price Ihr Höchstgebot übersteigt. Weitere Informationen über Spot-Instances finden Sie unter http://aws.amazon.com/ec2/spot-instances.

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 *ausschließlich* 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 bezieht sich auf die Freigabe von Getting Started with AWS-Computing-Grundlagen für Linux. Letzte Aktualisierung: September 06, 2014.

Änderung	Beschreibung	Veröffentlichungsdatum
Neuer Inhalt	Neues Dokument erstellt S	29. Februar 2012
Neuer Abschnitt hinzugefügt	Neuer Abschnitt über die Herstellung einer Verbindung mit Amazon EC2 über den MindTerm-Client hinzugefügt	8. März 2012