Wählen Sie eine Stickiness-Strategie für Ihren Load Balancer

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Wählen Sie eine Stickiness-Strategie für Ihren Load Balancer

Table of Contents

Wählen Sie eine Stickiness-Strategie für Ihren Load Balancer

Ryan Griffin, Amazon Web Services ()AWS

Juli 2024 (Verlauf der Dokumente)

Ein Begriff, der verwendet wird, um die Funktionalität eines Load Balancers zu beschreiben, um den Verkehr wiederholt von einem Client an ein einzelnes Ziel weiterzuleiten, anstatt den Verkehr auf mehrere Ziele zu verteilen. Beispielsweise kann der Datenverkehr von Client A kontinuierlich an einen bestimmten Server weitergeleitet werden, sodass der Server die Sitzungsstatusdaten verwalten kann. Wenn der Datenverkehr von Client A an zwei verschiedene Server weitergeleitet wird, fehlen möglicherweise auf jedem Server wichtige Informationen, die für den anderen Server verfügbar sind.

Daher ist es häufig erforderlich, eine konsistente Client-Verbindung über einen Load Balancer aufrechtzuerhalten. Es gibt zwei Arten von Klebrigkeit: Sticky Sessions und Stickiness bei Zielgruppen.

- Sticky Sessions — Verwaltung lokaler Sitzungsdaten in einer Amazon Elastic Compute Cloud (Amazon EC2) -Instance, um die Anwendungsarchitektur zu vereinfachen oder die Anwendungsleistung zu verbessern, da die Instance die Sitzungsstatusinformationen lokal verwalten oder zwischenspeichern kann. AWS bietet derzeit zwei Arten von Sticky Sessions an, auf die in diesem Leitfaden ausführlich eingegangen wird: Anwendungscookies und Load Balancer-Cookies.
- Dauerhaftigkeit bei der Zielgruppe — In blauen/grünen Bereitstellungen haben Sie möglicherweise mehrere Versionen einer Anwendung bereitgestellt, und Sie möchten möglicherweise, dass der Client während seiner Sitzung weiterhin dieselbe Version der Anwendung verwendet. In diesem Fall können Sie Zielgruppenbindung verwenden, um die gesamte Kommunikation vom Client an dieselbe Zielgruppe statt an dieselbe Instanz weiterzuleiten. EC2

Sie können diese beiden Stickiness-Strategien getrennt oder zusammen verwenden.

In diesem Leitfaden werden verschiedene Arten von Load Balancer-Engpässen und entsprechende Anwendungsfälle beschrieben, um Ihnen bei der Auswahl einer Strategie zu helfen. Der Leitfaden enthält AWS CloudFormation Vorlagen, die die einzelnen Strategien veranschaulichen.

Beispiel-Code

Dieses Handbuch enthält eine angehängte ZIP-Datei mit vier AWS CloudFormation Vorlagen, die Sie einsetzen können, um eine grundlegende Architektur zu erstellen und die einzelnen Stickiness-Strategien auszuprobieren. Wir empfehlen, dass Sie diese Vorlagen in einer Testumgebung bereitstellen, um die einzelnen Ansätze zu testen.

Laden Sie den Beispielcode herunter

Der Download beinhaltet die folgenden Vorlagen:

- `basic.yml`— Richtet einen Application Load Balancer ohne Klebrigkeit ein.
- `targetgroupstickiness.yml`— Demonstriert Engagiertheit auf der Grundlage von Zielgruppen.
- `stickysessionslb.yml`— Demonstriert Sticky-Sessions mit vom Load Balancer generierten Cookies.
- `stickysessionsapp.yml`— Demonstriert Sticky-Sitzungen mit anwendungsbasierten Cookies.

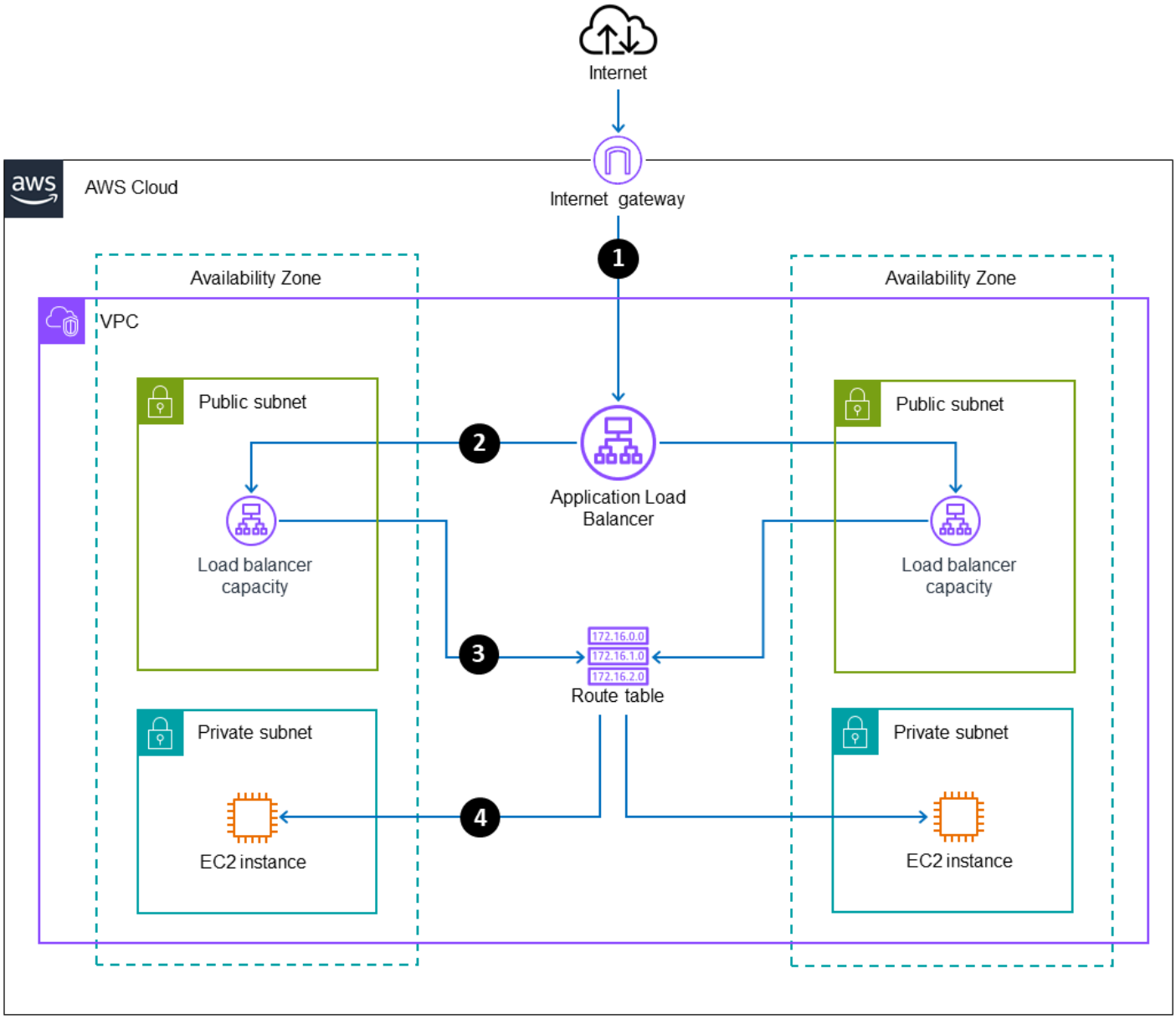
Um diese Vorlagen bereitzustellen, benötigen Sie ein aktives:AWS -Konto und Zugriff auf die CloudFormation Konsole. step-by-stepsAnweisungen zum Bereitstellen einer CloudFormation Vorlage finden Sie in der CloudFormation Dokumentation unter Erstellen eines Stacks.

Load Balancer-Subnetze und Routing

Beginnen wir mit einer Veranschaulichung, wie der Datenverkehr bei der Konfiguration eines Application Load Balancer, der mit dem Internet verbunden ist, zu Amazon Elastic Compute Cloud (Amazon EC2) -Instances in einem privaten Subnetz fließt. Diese Architektur spiegelt bewährte Methoden bei der Bereitstellung eines Application Load Balancer wider, der für das Internet offen ist.

Pfad des eingehenden Datenverkehrs

Das folgende Diagramm zeigt die Virtual Private Cloud (VPC) -Subnetze und das Routing, die mit dem eingehenden Verkehrsfluss verknüpft sind, wobei der Rückverkehr aus Gründen der Übersichtlichkeit aus dem Diagramm entfernt wurde.



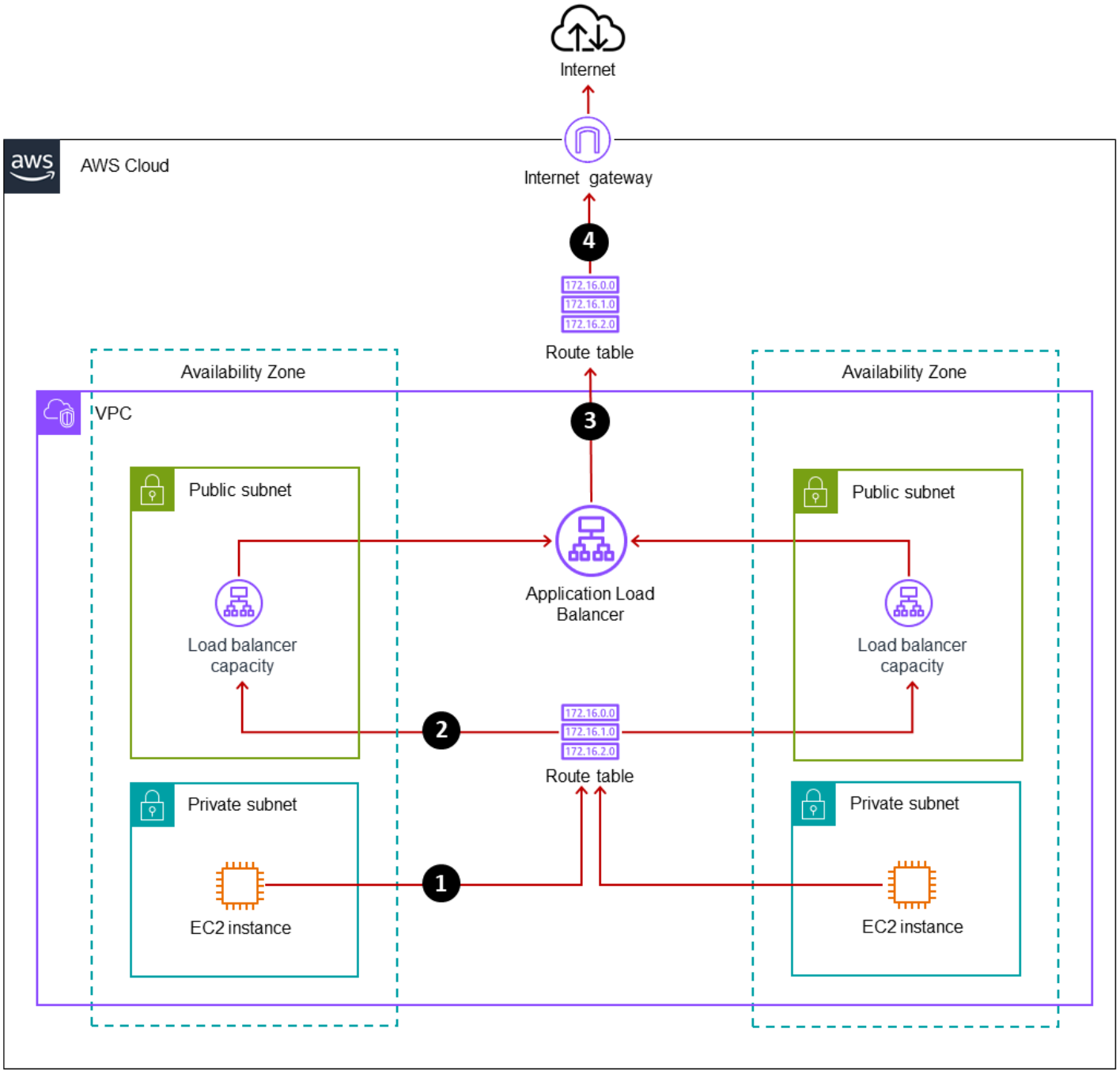
1. Der Datenverkehr aus dem Internet fließt in den DNS-Namen des Application Load Balancer.
2. Der Application Load Balancer ist in dem abgebildeten Szenario mit zwei öffentlichen Subnetzen verknüpft. Der Elastic Load Balancing Service erstellt Load Balancer-Kapazität in jeder aktivierten Availability Zone und leitet Traffic durch die Availability Zone, um die passende Routing-Logik zu ermitteln. Der Application Load Balancer verwendet seine interne Logik, um zu bestimmen, an welche Zielgruppe und Instanz der Traffic weitergeleitet werden soll.
3. Der Application Load Balancer leitet die Anfrage über einen Knoten, der dem öffentlichen Subnetz in derselben Availability Zone zugeordnet ist, an die EC2-Instance weiter. (Diese Knoten werden

vom Elastic Load Balancing Service konfiguriert, verwaltet und skaliert und sind für Benutzer nicht sichtbar.)

4. Die Routentabelle leitet den Verkehr lokal innerhalb der VPC, zwischen dem öffentlichen Subnetz und dem privaten Subnetz sowie zur EC2-Instance weiter.

Verkehrspfad zurückgeben

Das folgende Diagramm zeigt die VPC-Subnetze und das Routing, die dem Datenverkehrspfad zurück ins Internet zugeordnet sind, wobei der eingehende Verkehr aus Gründen der Übersichtlichkeit aus dem Diagramm entfernt wurde.

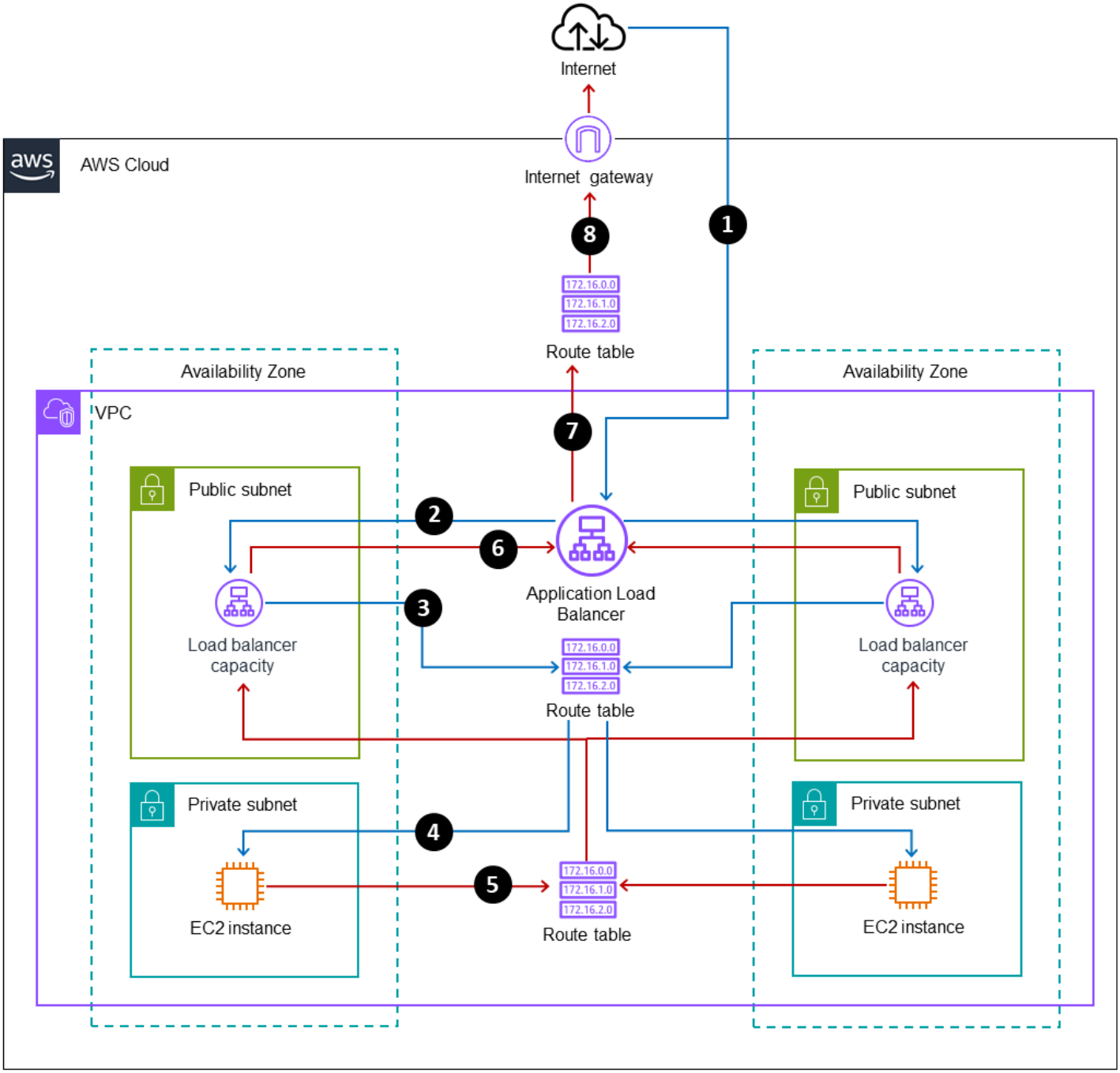


1. Die EC2-Instance im privaten Subnetz leitet den ausgehenden Datenverkehr durch die Routing-Tabelle weiter.
2. Die Routentabelle hat eine lokale Route zum öffentlichen Subnetz. Es erreicht die Kapazität des Application Load Balancer, über die der Datenverkehr eingegeben wurde, im entsprechenden öffentlichen Subnetz, indem es dem Pfad zurück folgt, auf dem der eingegebene Verkehr eingegeben wurde.

3. Der Application Load Balancer leitet den Datenverkehr über seine öffentliche Schnittstelle weiter.
4. Die Routing-Tabelle des öffentlichen Subnetzes hat eine Standardroute, die auf ein Internet-Gateway verweist, das den Verkehr zurück ins Internet leitet.

Vollständiges Verkehrsflussdiagramm

Das folgende Diagramm kombiniert die eingehenden und zurückgehenden Verkehrsflüsse, um eine vollständige Darstellung des Load Balancer-Routing zu bieten.



1. Der Datenverkehr aus dem Internet fließt in den DNS-Namen des Application Load Balancer.
2. Der Application Load Balancer ist in dem abgebildeten Szenario mit zwei öffentlichen Subnetzen verknüpft. Der Elastic Load Balancing Service erstellt Load Balancer-Kapazität in jeder aktivierten Availability Zone und leitet Traffic durch die Availability Zone, um die passende Routing-Logik zu ermitteln. Der Application Load Balancer verwendet seine interne Logik, um zu bestimmen, an welche Zielgruppe und Instanz der Traffic weitergeleitet werden soll.

3. Der Application Load Balancer leitet die Anfrage über einen Knoten, der dem öffentlichen Subnetz in derselben Availability Zone zugeordnet ist, an die EC2-Instance weiter. (Diese Knoten werden vom Elastic Load Balancing Service konfiguriert, verwaltet und skaliert und sind für Benutzer nicht sichtbar.)
4. Die Routentabelle leitet den Verkehr lokal innerhalb der VPC, zwischen dem öffentlichen Subnetz und dem privaten Subnetz sowie zur EC2-Instance weiter.
5. Die EC2-Instance im privaten Subnetz leitet den ausgehenden Datenverkehr über die Routentabelle weiter.
6. Die Routentabelle hat eine lokale Route zum öffentlichen Subnetz. Es erreicht die Kapazität des Application Load Balancer, über die der Datenverkehr eingegeben wurde, im entsprechenden öffentlichen Subnetz, indem es dem Pfad zurück folgt, auf dem der eingegebene Verkehr eingegeben wurde.
7. Der Application Load Balancer leitet den Datenverkehr über seine öffentliche Schnittstelle weiter.
8. Die Routing-Tabelle des öffentlichen Subnetzes hat eine Standardroute, die auf ein Internet-Gateway verweist, das den Verkehr zurück ins Internet leitet.

Welche Stickiness-Strategie ist die richtige für Sie?

Die Beantwortung der folgenden Fragen hilft Ihnen bei der Festlegung Ihrer Stabilitätsstrategie für den Load Balancer.

Verwenden Sie? WebSockets

- Ja: WebSockets sind von Natur aus klebrig, sodass Sie keine Strategie anwenden müssen.
- Nein: Fahren Sie mit der nächsten Frage fort.

Planen Sie eine blaue/grüne Implementierung?

- Ja: Verwenden Sie mindestens Zielgruppenbindung, fahren Sie aber mit der nächsten Frage fort.
- Nein: Fahren Sie mit der nächsten Frage fort.

Müssen Sie einen Teil oder den gesamten Datenverkehr von einem Client wiederholt an dieselbe EC2 Instance weiterleiten?

- Ja: Fahren Sie mit der nächsten Frage fort.
- Nein: Sie müssen keine Sticky-Sessions verwenden.

Möchten Sie, dass Ihr Anwendungscode oder Client die Statusattribute und die Dauer mithilfe von Cookies steuert?

- Ja: Verwenden Sie anwendungsbasierte Cookies, um anwendungsbasierte Sticky Sessions zu aktivieren.
- Nein: Verwenden Sie vom Load Balancer generierte Cookies, um dauerabhängige Sticky-Sitzungen zu ermöglichen.

Application Load Balancer ohne Klebrigkeit

Wenn Sie einen Application Load Balancer ohne jegliche Form von Stickiness verwenden, verwendet der Load Balancer standardmäßig die Round-Robin-Methode, um die EC2 Instance zu bestimmen, an die er den Datenverkehr weiterleiten soll.

Vorlage: Verwenden Sie die CloudFormation Vorlage `basic.yml` (in der ZIP-Datei mit dem Beispielcode enthalten), um diese Funktionalität auszuprobieren.

> **Note**
>
> Alle in diesem Handbuch enthaltenen CloudFormation Vorlagen stellen eine benutzerdefinierte VPC, Routing-Tabellen, Routen, ein Internet-Gateway, einen Application Load Balancer, Zielgruppen, Listener und EC2 Instances bereit, um eine bestimmte Load Balancer-Stickiness-Strategie zu veranschaulichen.

Häufige Anwendungsfälle

Verwenden Sie in diesen Szenarien einen Application Load Balancer ohne Klebrigkeit:

- Sie haben eine Liste von Zielen, an die der Datenverkehr weitergeleitet werden soll, aber die Ziele behalten den Sitzungsstatus nicht bei.
- Sie verwenden Webserver, die den Sitzungsstatus nicht beibehalten.
- Sie verwenden Anwendungsserver, die den Sitzungsstatus nicht beibehalten.

Schritte

> **Hinweise**
>
> - Für NAT-Gateways fallen geringe Kosten an.
> - Mehrere EC2 Instances verbrauchen Ihr kostenloses Kontingent schneller als eine einzelne EC2 Instanz.

1. Stellen Sie die CloudFormation Vorlage `basic.yml` in einer Laborumgebung bereit.

2. Warten Sie, bis sich der Integritätsstatus Ihrer Zielgruppeninstanzen von anfänglich auf fehlerfrei ändert.
3. Navigieren Sie in einem Webbrowser mit HTTP (TCP/80) zur Application Load Balancer Balancer-URL.

   Zum Beispiel: `http://alb-123456789.us-east-1.elb.amazonaws.com/`

   Auf der Webseite wird Instanz 1 — TG1 oder Instanz 2 — angezeigt. TG1
4. Aktualisieren Sie die Seite mehrmals.

Erwartete Ergebnisse

Die Instanz, die die Webseite lädt (Instanz 1 oder Instanz 2), sollte sich jedes Mal ändern, was sich im Seitentext widerspiegelt. Die Load Balancer-Logik verwaltet das letzte Ziel über mehrere interne Knoten hinweg, was zu einer Synchronisationsverzögerung führen kann, sodass die Möglichkeit besteht, dass Sie an dasselbe Ziel weitergeleitet werden.

Funktionsweise

- In diesem Beispiel werden zwei EC2 Instanzen einer einzigen Zielgruppe zugewiesen. Auf den EC2 Instanzen ist ein Apache-Webserver (`httpd`) installiert, und der `index.html` Seitentext auf jeder EC2 Instanz ist fest codiert, um diese Instanz zu identifizieren.
- Der Application Load Balancer führt seine interne Round-Robin-Logik aus, um zu bestimmen, welche EC2 Instance den Datenverkehr empfangen soll.
- Jedes Mal, wenn Sie die Webseite neu laden, führt der Application Load Balancer seine Routing-Logik aus, und auf der Seite wird Instanz 1 — TG1 oder Instanz 2 — angezeigt. TG1

Klebrigkeit bei der Zielgruppe

Wenn Sie einen Application Load Balancer mit Zielgruppenbindung verwenden:

- Der Application Load Balancer bestimmt anhand der Zielgruppengewichtung, wie der eingehende Verkehr zwischen den Zielgruppen verteilt werden soll.
- Standardmäßig verwendet der Application Load Balancer die Round-Robin-Methode, um Anfragen an die EC2 Instances in der Zielzielgruppe weiterzuleiten.

Vorlage: Verwenden Sie die CloudFormation Vorlage `targetgroupstickiness.yml` (in der Beispielcode-.zip-Datei enthalten), um die Zielgruppenbindung auszuprobieren.

Häufige Anwendungsfälle

Verwenden Sie Zielgruppen-Klebrigkeit in diesen Szenarien:

- Dem Load Balancer sind mehrere Zielgruppen zugewiesen, und der Traffic von einem Client sollte konsistent an Instanzen innerhalb dieser Zielgruppe weitergeleitet werden.
- Blaue/grüne Bereitstellungen.

Codeänderungen von basic.yml

Eine einzige Änderung wurde am Listener vorgenommen: Wir haben die Standardaktionen des Application Load Balancer so geändert, dass sie zwei Zielgruppen (`TG1`und`TG2`) mit gleichem Gewicht und einer Stickiness-Konfiguration angeben.

basic.yml

```
Listener1:
  Type: 'AWS::ElasticLoadBalancingV
2::Listener'
  Properties:
    LoadBalancerArn: !Ref ALB
    Protocol: HTTP
    Port: 80
```

targetgroupstickiness.yml

```
Listener1:
  Type: 'AWS::ElasticLoadBalancingV
2::Listener'
  Properties:
    LoadBalancerArn: !Ref ALB
    Protocol: HTTP
    Port: 80
```

```
    DefaultActions:
      - TargetGroupArn: !Ref TG1
        Type: forward
```

```
    DefaultActions:
      - ForwardConfig:
          TargetGroups:
            - TargetGroupArn: !Ref TG1
              Weight: 1
            - TargetGroupArn: !Ref TG2
              Weight: 1
          TargetGroupStickinessConfig
:
              DurationSeconds: 10
              Enabled: true
        Type: forward
```

Schritte

> **Hinweise**
>
> - Für NAT-Gateways fallen geringe Kosten an.
> - Mehrere EC2 Instances verbrauchen Ihr kostenloses Kontingent schneller als eine einzelne EC2 Instanz.

1. Stellen Sie die CloudFormation Vorlage `targetgroupstickiness.yml` in einer Laborumgebung bereit.
2. Warten Sie, bis sich der Integritätsstatus Ihrer Zielgruppeninstanzen von anfänglich auf fehlerfrei ändert.
3. Navigieren Sie in einem Webbrowser mit HTTP (TCP/80) zur Application Load Balancer Balancer-URL.

   Zum Beispiel: `http://alb-123456789.us-east-1.elb.amazonaws.com/`

   Auf der Webseite wird eine der folgenden Optionen angezeigt: Instanz 1 - TG1, Instanz 2 - TG1, Instanz 3 - oder Instanz 4 - TG2. TG2
4. Aktualisieren Sie die Seite mehrmals.

Erwartete Ergebnisse

> **Note**
>
> Die CloudFormation Vorlage in diesem Beispiel konfiguriert die Klebrigkeit so, dass sie 10 Sekunden anhält.

Die Instanzen, die die Webseite laden, sollten innerhalb der Dauer von 10 Sekunden innerhalb der Zielgruppe (TG1 oder TG2) bleiben, was sich im Seitentext widerspiegelt.

Nach etwa 10 Sekunden wird die Klebrigkeit aufgehoben und die für die Zielgruppe eingestellte Instanzgruppe kann sich ändern.

Funktionsweise

- In diesem Beispiel sind vier EC2 Instanzen auf zwei Zielgruppen aufgeteilt, mit zwei Instanzen pro Zielgruppe. Auf den EC2 Instanzen ist ein Apache-Webserver (`httpd`) installiert, und der `index.html` Seitentext auf jeder EC2 Instanz ist fest codiert, sodass er eindeutig ist.
- Der Application Load Balancer erstellt eine Bindung für die Sitzung des Benutzers an die Zielzielgruppe mit einer Ablaufzeit.
- Wenn Sie die Seite neu laden, prüft der Application Load Balancer, ob die Bindung existiert und nicht abgelaufen ist.
  - Wenn die Bindung abgelaufen ist oder nicht existiert, führt der Application Load Balancer seine Routing-Logik aus und bestimmt die Zielzielgruppe.
  - Wenn die Bindung nicht abgelaufen ist, leitet der Application Load Balancer den Traffic an dieselbe Zielgruppe weiter, aber nicht unbedingt an dieselbe EC2 Instanz.

Sticky Sessions mit vom Load Balancer generierten Cookies

Wenn Sie einen Application Load Balancer mit einem vom Load Balancer generierten Cookie verwenden:

- Der Application Load Balancer bestimmt anhand der Zielgruppengewichtung, wie der eingehende Verkehr zwischen den Zielgruppen verteilt werden soll.
- Standardmäßig verwendet der Application Load Balancer die Round-Robin-Methode, um Anfragen an die EC2 Instances in der Zielzielgruppe weiterzuleiten.

  Nachdem der Datenverkehr anfänglich an eine Instance weitergeleitet wurde, bleibt der nachfolgende Traffic für eine bestimmte Dauer bei dieser EC2 Instance.

Vorlage: Verwenden Sie die CloudFormation Vorlage `stickysessionslb.yml` (in der ZIP-Datei mit dem Beispielcode enthalten), um Sticky-Sessions mit vom Load Balancer generierten Cookies auszuprobieren.

Häufige Anwendungsfälle

Verwenden Sie in diesen Szenarien Sticky-Sitzungen mit vom Load Balancer generierten Cookies:

- PHP-Webserver
- Server, die temporäre Sitzungsdaten wie Protokolle, Einkaufswagen oder Chat-Konversationen verwalten

Änderungen am Code von basic.yml

Die relevanten Codeänderungen sind in der Zielgruppenkonfiguration enthalten, um den Stickiness-Typ auf `lb_cookie` und die Dauer auf 10 Sekunden einzustellen.

basic.yml

```
TG1:
   Type: 'AWS::ElasticLoadBalancingV
2::TargetGroup'
   Properties:
```

stickysessionslb.yml

```
TG1:
   Type: 'AWS::ElasticLoadBalancingV
2::TargetGroup'
   Properties:
```

```
      Name: TG1
      Protocol: HTTP
      Port: 80
      TargetType: instance
      Targets:
        - Id: !Ref Instance1
        - Id: !Ref Instance2
VpcId: !Ref CustomVPC
```

```
      Name: TG1
      Protocol: HTTP
      Port: 80
      TargetType: instance
      Targets:
        - Id: !Ref Instance1
        - Id: !Ref Instance2
      VpcId: !Ref CustomVPC
      TargetGroupAttributes:
        - Key: stickiness.enabled
          Value: true
        - Key: stickiness.type
          Value: lb_cookie
        - Key: stickiness.lb_cook
ie.duration_seconds
Value: 10
```

Schritte

> **Hinweise**
>
> - Für NAT-Gateways fallen geringe Kosten an.
> - Durch mehrere EC2 Instances wird Ihr kostenloses Kontingent schneller aufgebraucht als durch eine einzelne EC2 Instanz.

1. Stellen Sie die CloudFormation Vorlage `stickysessionslb.yml` in einer Laborumgebung bereit.
2. Warten Sie, bis sich der Integritätsstatus Ihrer Zielgruppeninstanzen von anfänglich auf fehlerfrei ändert.
3. Navigieren Sie in einem Webbrowser mit HTTP (TCP/80) zur Application Load Balancer Balancer-URL.

   Zum Beispiel: `http://alb-123456789.us-east-1.elb.amazonaws.com/`

   Auf der Webseite wird eine der folgenden Optionen angezeigt: Instanz 1 - TG1, Instanz 2 - TG1, Instanz 3 - oder Instanz 4 - TG2. TG1
4. Aktualisieren Sie die Seite mehrmals.

Erwartete Ergebnisse

> **Note**
>
> Die CloudFormation Vorlage in diesem Beispiel konfiguriert die Klebrigkeit so, dass sie 10 Sekunden anhält.

Die Instanz, die die Webseite lädt, sollte innerhalb der Dauer von 10 Sekunden unverändert bleiben, was sich im Seitentext widerspiegelt. Nach etwa 10 Sekunden wird die Klebrigkeit aufgehoben und die Zielinstanz kann sich ändern.

Funktionsweise

- In diesem Beispiel sind zwei EC2 Instanzen in einer Zielgruppe vorhanden. Auf den EC2 Instanzen ist ein Apache-Webserver (`httpd`) installiert, und der `index.html` Seitentext auf jeder EC2 Instanz ist fest codiert, sodass er eindeutig ist.
- Der Application Load Balancer erstellt eine Bindung für die Sitzung des Benutzers, die an das Ziel gebunden wird, mit einer Ablaufzeit.
- Wenn Sie die Seite neu laden, prüft der Application Load Balancer, ob die Bindung existiert und nicht abgelaufen ist.
  - Wenn die Bindung abgelaufen ist oder nicht existiert, führt der Application Load Balancer seine Routing-Logik aus und bestimmt die Zielinstanz.
  - Wenn die Bindung nicht abgelaufen ist, leitet der Application Load Balancer den Datenverkehr an dieselbe Zielinstanz weiter.

Sticky Sessions mit anwendungsbasierten Cookies

Wenn Sie einen Application Load Balancer mit einem anwendungsbasierten Cookie verwenden:

- Der Application Load Balancer bestimmt anhand der Zielgruppengewichtung, wie der eingehende Verkehr zwischen den Zielgruppen verteilt werden soll.
- Standardmäßig verwendet der Application Load Balancer die Round-Robin-Methode, um Anfragen an die EC2 Instances in der Zielzielgruppe weiterzuleiten.
- Nachdem der Datenverkehr zunächst an eine EC2 Instance weitergeleitet wurde, sollte die Antwort der EC2 Instance-Anwendung ein benutzerdefiniertes Anwendungscookie enthalten, das zusammen mit einem automatisierten Application Load Balancer Balancer-Cookie an den Client zurückgesendet wird.
- Der nachfolgende Datenverkehr bleibt an der EC2 Instanz hängen, wenn der Client das Anwendungscookie und das Application Load Balancer Balancer-Cookie zurücksendet.
- Das anwendungsbasierte Cookie läuft ab, wenn es für die konfigurierte Dauer nicht verwendet wird.

Vorlage: Verwenden Sie die CloudFormation Vorlage `stickysessionsapp.yml` (in der Beispielcode-.zip-Datei enthalten), um Sticky-Sessions mit anwendungsbasierten Cookies auszuprobieren.

Häufige Anwendungsfälle

Verwenden Sie Sticky-Sitzungen mit anwendungsgenerierten Cookies, wenn Sie in diesen Szenarien zusätzliche Kontrolle wünschen:

- PHP-Webserver
- Server, auf denen temporäre Sitzungsdaten wie Protokolle, Einkaufswagen oder Chat-Konversationen gespeichert werden

Änderungen am Code von basic.yml

Die einzige Codeänderung betrifft die Zielgruppenkonfiguration. Wir haben dem Application Load Balancer und den Zielgruppenattributen eine Stickiness-Konfiguration hinzugefügt. Die Dauer der Anwendungs-Cookies ist angegeben, und für die Zielgruppe ist die Anwendungs-Cookie-Stickiness aktiviert.

basic.yml

```
TG1:
   Type: 'AWS::ElasticLoadBalancingV
2::TargetGroup'
   Properties:
     Name: TG1
     Protocol: HTTP
     Port: 80
     TargetType: instance
     Targets:
       - Id: !Ref Instance1
       - Id: !Ref Instance2
VpcId: !Ref CustomVPC
```

stickysessionsapp.yml

```
TG1:
  Type: 'AWS::ElasticLoadBalancingV
2::TargetGroup'
  Properties:
    Name: TG1
    Protocol: HTTP
    Port: 80
    TargetType: instance
    Targets:
      - Id: !Ref Instance1
      - Id: !Ref Instance2
    VpcId: !Ref CustomVPC
    TargetGroupAttributes:
      - Key: stickiness.enabled
        Value: true
      - Key: stickiness.type
        Value: app_cookie
      - Key: stickiness.app_coo
kie.duration_seconds
        Value: 10
      - Key: stickiness.app_coo
kie.cookie_name
        Value: TESTCOOKIE
```

Schritte

Hinweise

- Für NAT-Gateways fallen geringe Kosten an.
- Durch mehrere EC2 Instances wird Ihr kostenloses Kontingent schneller aufgebraucht als durch eine einzelne EC2 Instanz.

1. Stellen Sie die CloudFormation Vorlage `stickysessionslb.yml` in einer Laborumgebung bereit.

2. Warten Sie, bis sich der Integritätsstatus Ihrer Zielgruppeninstanzen von anfänglich auf fehlerfrei ändert.
3. Navigieren Sie in einem Webbrowser mit HTTP (TCP/80) zur Application Load Balancer Balancer-URL.

   Zum Beispiel: `http://alb-123456789.us-east-1.elb.amazonaws.com/`

   Auf der Webseite wird eine der folgenden Optionen angezeigt: Instanz 1 - TG1, Instanz 2 - TG1, Instanz 3 - oder Instanz 4 - TG2. TG2
4. Aktualisieren Sie die Seite mehrmals.

Erwartete Ergebnisse

> **Note**
>
> Die CloudFormation Vorlage in diesem Beispiel hat die Klebrigkeit so konfiguriert, dass sie 10 Sekunden anhält. Die gültige Konfiguration für die Dauer der Klebrigkeit liegt zwischen 1 Sekunde und 1 Woche.

Die Instanz, die die Webseite lädt, sollte dieselbe bleiben, wie im Seitentext angegeben.

Die Dauer der Sperrzeit wird nicht aktualisiert, sondern basiert auf dem Ablauf, der im Application Load Balancer für das von der EC2 Instanz generierte Anwendungscookie konfiguriert wurde.

Beispiel 1: Warten Sie 5 Sekunden, um die Seite zu aktualisieren. Dieselbe Instanz wird geladen und die Stickiness wird für weitere 10 Sekunden aktualisiert.

Beispiel 2: Warten Sie länger als 10 Sekunden, um die Seite neu zu laden. Das Anwendungscookie läuft ab und Sie werden zu einer anderen EC2 Instanz weitergeleitet. Diese neue Instanz generiert ein weiteres Anwendungscookie mit einer Dauer von 10 Sekunden.

Funktionsweise

- In diesem Beispiel ist auf den EC2 Instanzen ein Apache-Webserver (`httpd`) installiert. Die `httpd.conf` Datei ist so konfiguriert, dass sie einen statischen `Set-Cookie` Wert an den Client (Ihren Webbrowser) zurückgibt. Der `Set-Cookie` Wert ist fest codiert als `TESTCOOKIE=<somevalue>`

- Öffnen Sie in Ihrem Browser die Option „Element untersuchen“, wählen Sie die Registerkarte „Netzwerk“ und anschließend die Methode „Abrufen“, wodurch die Seite geladen wird. Sie werden eine Registerkarte „Cookies“ sehen.
- Der Browser ist eine Client-Anwendung, die automatisch so konfiguriert ist, dass sie alle nachfolgenden Updates mit den Cookies, die sie in der `Set-Cookie` Antwort des Servers empfängt, an den Server zurücksendet.
- Wenn Sie die Seite neu laden, werden die beim ersten Laden der Seite empfangenen Cookies automatisch an den Application Load Balancer zurückgesendet.
  - Wenn das Cookie abgelaufen ist (d. h. 10 Sekunden seit dem letzten Aufruf vergangen sind), verwendet der Application Load Balancer eine neue Logik, um zu bestimmen, an welche EC2 Instanz der Datenverkehr weitergeleitet werden soll.
  - Wenn das Cookie nicht abgelaufen ist, leitet der Application Load Balancer den Datenverkehr an dieselbe EC2 Instanz weiter.

Ressourcen

- Sticky Sessions für Ihren Application Load Balancer (Elastic Load Balancing Balancing-Dokumentation)

Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen in diesem Leitfaden beschrieben. Um Benachrichtigungen über zukünftige Aktualisierungen zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Der Abschnitt wurde aktualisiert	Der Abschnitt Subnetze und Routing für den Load Balancer wurde mit neuen Diagrammen und Erläuterungen aktualisiert.	5. Juli 2024
Aktualisierungen und Korrekturen	Der Code, die Diagramme und die Beispiele wurden aktualisiert.	31. Mai 2023
Abschnitte wurden aktualisiert	Die Abschnitte „So funktioniert's“ in Zielgruppen-Stickiness und Sticky-Sessions mit vom Load Balancer generierten Cookies wurden aktualisiert, um genauere und detailliertere Informationen bereitzustellen.	18. Juli 2022
—	Erste Veröffentlichung	05. August 2021

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.