Memilih strategi lengket untuk penyeimbang beban Anda

AWS Panduan Preskriptif

AWS Panduan Preskriptif: Memilih strategi lengket untuk penyeimbang beban Anda

Table of Contents

Memilih strategi lengket untuk penyeimbang beban Anda

Ryan Griffin, Amazon Web Services ()AWS

Juli 2024 (sejarah dokumen)

Stickiness adalah istilah yang digunakan untuk menggambarkan fungsionalitas penyeimbang beban untuk berulang kali merutekan lalu lintas dari klien ke satu tujuan, alih-alih menyeimbangkan lalu lintas di beberapa tujuan. Misalnya, lalu lintas dari klien A dapat terus dirutekan ke server tertentu, sehingga server dapat mempertahankan data status sesi. Jika lalu lintas dari klien A dirutekan ke dua server yang berbeda, setiap server mungkin kehilangan informasi penting yang tersedia untuk server lain.

Oleh karena itu, seringkali diperlukan untuk mempertahankan koneksi klien yang konsisten melalui penyeimbang beban. Ada dua jenis lengket: sesi lengket dan kelengketan kelompok sasaran.

- Sesi lengket — Mempertahankan data sesi lokal dalam instans Amazon Elastic Compute Cloud (Amazon EC2) untuk menyederhanakan arsitektur aplikasi atau meningkatkan kinerja aplikasi, karena instans dapat mempertahankan atau menyimpan informasi status sesi secara lokal. AWS Saat ini menawarkan dua jenis sesi lengket, yang dibahas secara rinci oleh panduan ini: cookie aplikasi dan cookie penyeimbang beban.
- Kelengketan grup target - Dalam penerapan biru/hijau, Anda mungkin memiliki beberapa versi aplikasi yang digunakan, dan Anda mungkin ingin klien terus menggunakan versi aplikasi yang sama selama sesi mereka. Dalam hal ini, Anda dapat menggunakan kelengketan grup target untuk merutekan semua komunikasi dari klien ke grup target yang sama, bukan instance yang sama EC2 .

Anda dapat menggunakan dua strategi lengket ini secara terpisah atau bersama-sama.

Panduan ini menjelaskan berbagai jenis kekakuan penyeimbang beban dan kasus penggunaan yang berlaku, untuk membantu Anda memilih strategi. Panduan ini mencakup AWS CloudFormation template yang menggambarkan setiap strategi.

Kode sampel

Panduan ini menyediakan file.zip terlampir yang mencakup empat AWS CloudFormation templat yang dapat Anda gunakan untuk membangun arsitektur dasar dan mencoba setiap strategi

lengket. Kami menyarankan Anda menerapkan template ini di lingkungan lab untuk menguji setiap pendekatan.

Unduh kode sampel

Unduhan mencakup template ini:

- `basic.yml`- Mengatur Application Load Balancer tanpa lengket.
- `targetgroupstickiness.yml`— Menunjukkan kelengketan berdasarkan kelompok sasaran.
- `stickysessionslb.yml`— Menunjukkan sesi lengket dengan cookie yang dihasilkan penyeimbang beban.
- `stickysessionsapp.yml`— Menunjukkan sesi lengket dengan cookie berbasis aplikasi.

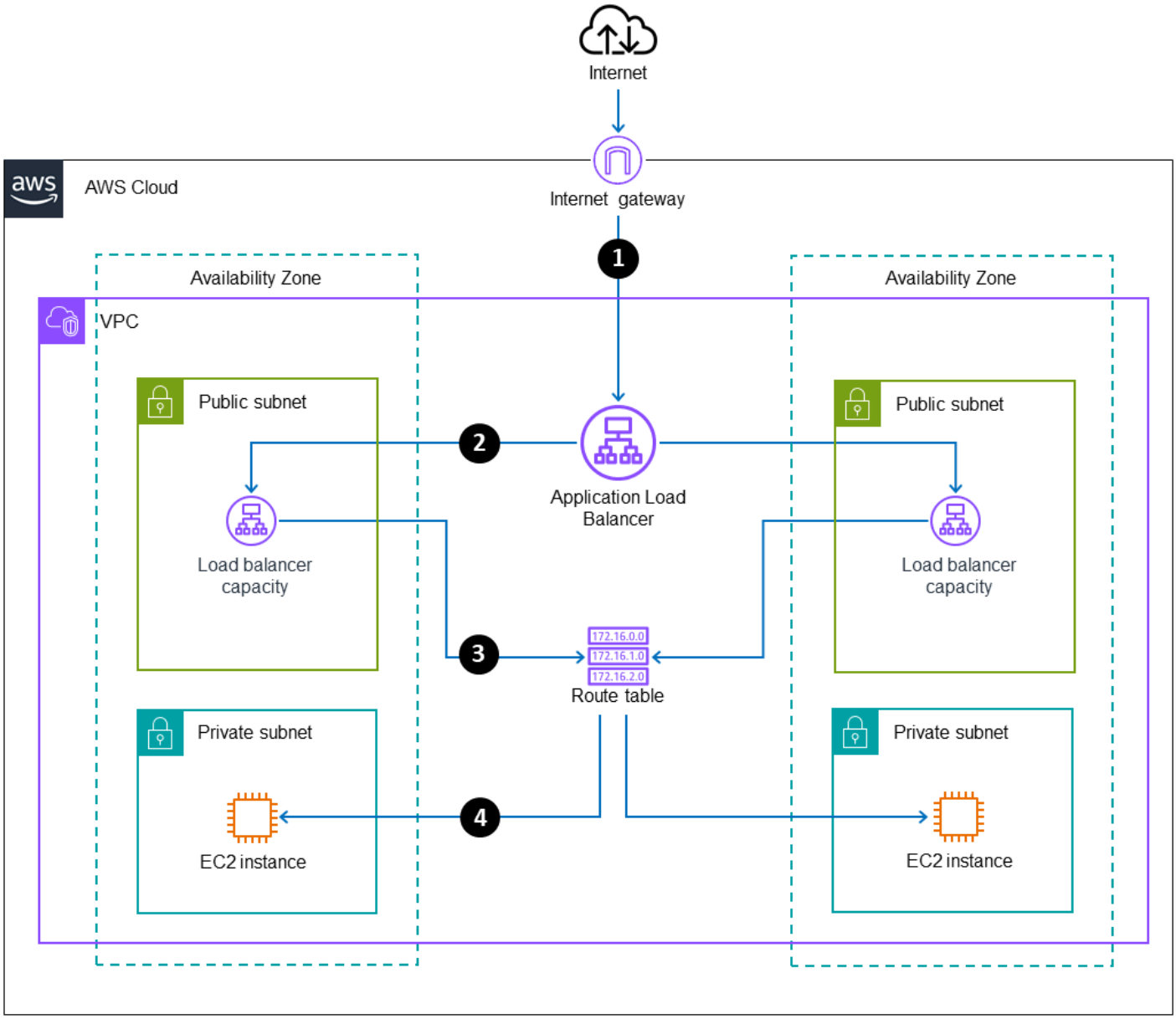
Untuk menyebarkan templat ini, Anda memerlukan AWS akun aktif: dan akses ke CloudFormation konsol. Untuk step-by-steps petunjuk penerapan CloudFormation templat, lihat Membuat tumpukan dalam CloudFormation dokumentasi.

Subnet penyeimbang beban dan perutean

Mari kita mulai dengan ilustrasi bagaimana arus lalu lintas saat Anda mengonfigurasi Application Load Balancer yang menghadap internet, ke instans Amazon Elastic Compute Cloud (Amazon EC2) di subnet pribadi. Arsitektur ini mencerminkan praktik terbaik saat menerapkan Application Load Balancer yang terbuka untuk internet.

Jalur lalu lintas masuk

Diagram berikut menggambarkan subnet virtual private cloud (VPC) dan routing yang terkait dengan arus lalu lintas masuk, dengan lalu lintas kembali dihapus dari diagram untuk kejelasan.

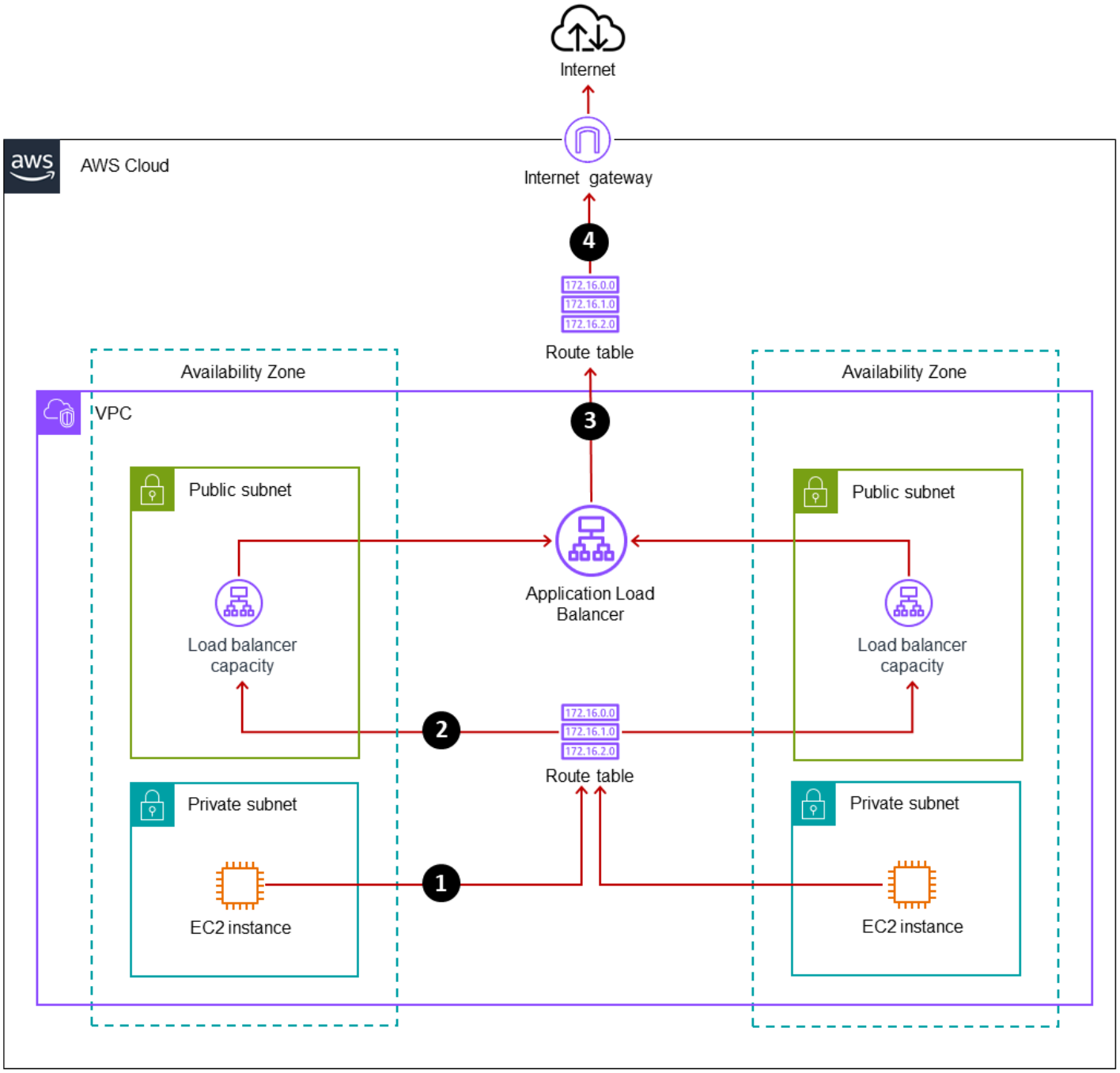


1. Lalu lintas dari internet mengalir ke nama Application Load Balancer DNS.
2. Application Load Balancer dikaitkan dengan dua subnet publik dalam skenario yang diilustrasikan. Layanan Elastic Load Balancing menciptakan kapasitas penyeimbang beban di setiap Availability Zone yang diaktifkan, dan mengirimkan lalu lintas melalui Availability Zone untuk menentukan logika routing yang sesuai. Application Load Balancer menggunakan logika internalnya untuk menentukan kelompok target dan instance mana yang akan mengarahkan lalu lintas ke.
3. Application Load Balancer merutekan permintaan ke instans EC2 melalui node yang terkait dengan subnet publik di Availability Zone yang sama. (Node ini dikonfigurasi, dikelola, dan diskalakan oleh layanan Elastic Load Balancing dan tidak terlihat oleh pengguna.)

4. Tabel rute merutekan lalu lintas secara lokal di dalam VPC, antara subnet publik dan subnet pribadi, dan ke instans EC2.

Kembalikan jalur lalu lintas

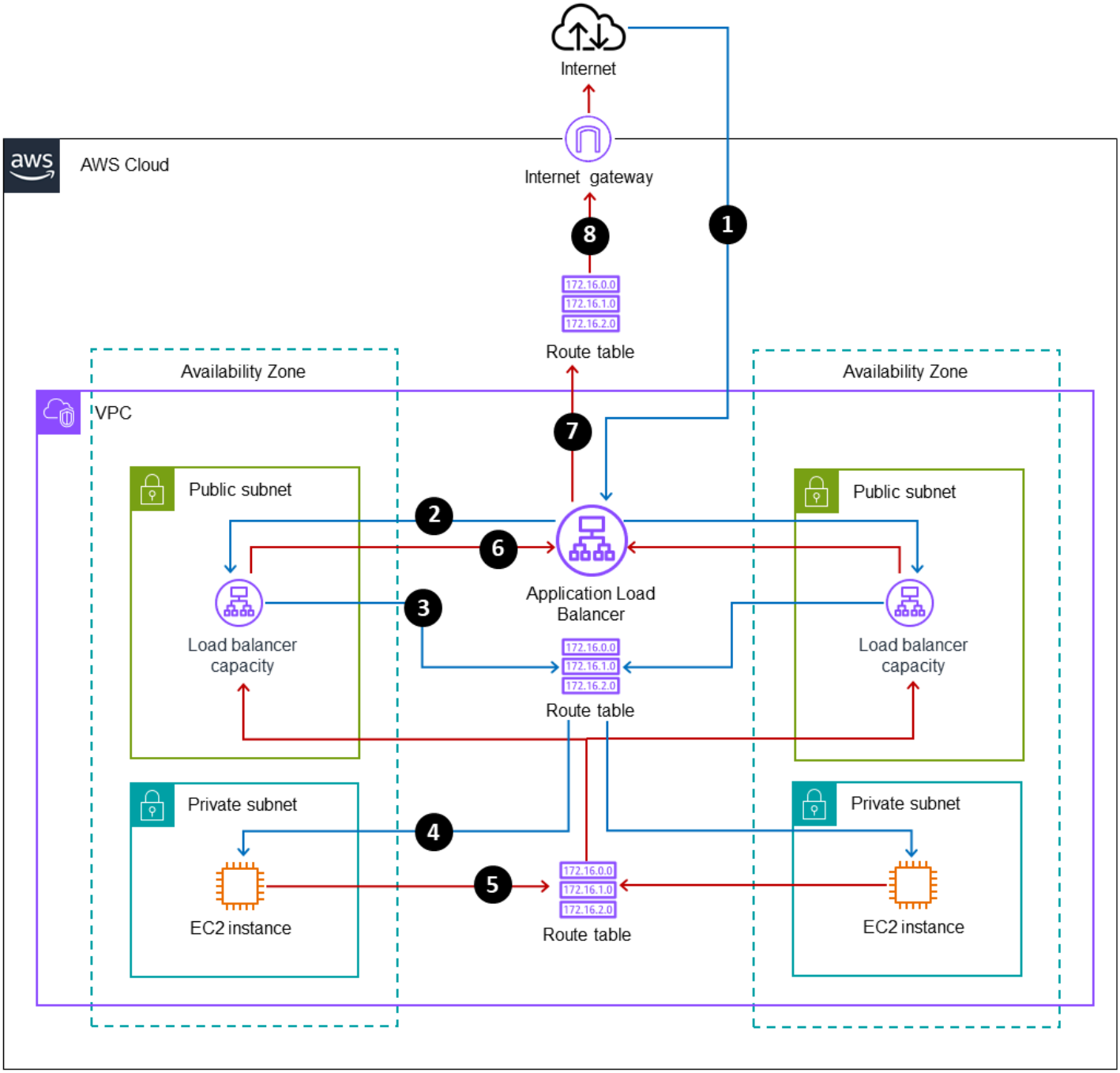
Diagram berikut menggambarkan subnet VPC dan routing yang terkait dengan jalur lalu lintas kembali ke internet, dengan lalu lintas masuk dihapus dari diagram untuk kejelasan.



1. Instans EC2 di subnet pribadi merutekan lalu lintas keluar melalui tabel rute.
2. Tabel rute memiliki rute lokal ke subnet publik. Ini mencapai kapasitas Application Load Balancer yang dimasukkan lalu lintas, di subnet publik yang sesuai, dengan mengikuti jalur kembali cara lalu lintas masuk.
3. Application Load Balancer merutekan lalu lintas keluar melalui antarmuka publiknya.
4. Tabel rute subnet publik memiliki rute default yang menunjuk ke gateway internet, yang mengarahkan lalu lintas kembali ke internet.

Diagram arus lalu lintas lengkap

Diagram berikut menggabungkan arus lalu lintas masuk dan kembali untuk memberikan ilustrasi lengkap perutean penyeimbang beban.



1. Lalu lintas dari internet mengalir ke nama Application Load Balancer DNS.
2. Application Load Balancer dikaitkan dengan dua subnet publik dalam skenario yang diilustrasikan. Layanan Elastic Load Balancing menciptakan kapasitas penyeimbang beban di setiap Availability Zone yang diaktifkan, dan mengirimkan lalu lintas melalui Availability Zone untuk menentukan logika routing yang sesuai. Application Load Balancer menggunakan logika internalnya untuk menentukan kelompok target dan instance mana yang akan mengarahkan lalu lintas ke.

3. Application Load Balancer merutekan permintaan ke instans EC2 melalui node yang terkait dengan subnet publik di Availability Zone yang sama. (Node ini dikonfigurasi, dikelola, dan diskalakan oleh layanan Elastic Load Balancing dan tidak terlihat oleh pengguna.)
4. Tabel rute merutekan lalu lintas secara lokal di dalam VPC, antara subnet publik dan subnet pribadi, dan ke instans EC2.
5. Instans EC2 di subnet pribadi merutekan lalu lintas keluar melalui tabel rute.
6. Tabel rute memiliki rute lokal ke subnet publik. Ini mencapai kapasitas Application Load Balancer yang dimasukkan lalu lintas, di subnet publik yang sesuai, dengan mengikuti jalur kembali cara lalu lintas masuk.
7. Application Load Balancer merutekan lalu lintas keluar melalui antarmuka publiknya.
8. Tabel rute subnet publik memiliki rute default yang menunjuk ke gateway internet, yang mengarahkan lalu lintas kembali ke internet.

Strategi lengket mana yang tepat untuk Anda?

Menjawab pertanyaan-pertanyaan berikut akan membantu Anda menentukan strategi lengket penyeimbang beban Anda.

Apakah Anda menggunakan WebSockets?

- Ya: WebSockets secara inheren lengket, jadi tidak perlu menggunakan strategi apa pun.
- Tidak: Lanjutkan ke pertanyaan berikutnya.

Apakah Anda merencanakan penyebaran biru/hijau?

- Ya: Minimal, gunakan kelengketan kelompok sasaran, tetapi lanjutkan ke pertanyaan berikutnya.
- Tidak: Lanjutkan ke pertanyaan berikutnya.

Apakah Anda memerlukan beberapa atau semua lalu lintas dari klien untuk dialihkan ke EC2 instance yang sama berulang kali?

- Ya: Lanjutkan ke pertanyaan berikutnya.
- Tidak: Anda tidak perlu menggunakan sesi lengket.

Apakah Anda ingin kode aplikasi atau klien Anda mengontrol atribut dan durasi status dengan menggunakan cookie?

- Ya: Gunakan cookie berbasis aplikasi untuk mengaktifkan sesi lengket berbasis aplikasi.
- Tidak: Gunakan cookie yang dihasilkan penyeimbang beban untuk mengaktifkan sesi lengket berbasis durasi.

Application Load Balancer tanpa lengket

Bila Anda menggunakan Application Load Balancer tanpa bentuk lengket, secara default, load balancer menggunakan metode round robin untuk menentukan EC2 instance yang harus mengarahkan lalu lintas.

Template: Gunakan CloudFormation template `basic.yml` (termasuk dalam contoh kode file.zip) untuk mencoba fungsi ini.

> **Note**
>
> Semua CloudFormation template yang disertakan dengan panduan ini menyebarkan VPC kustom, tabel rute, rute, gateway internet, Application Load Balancer, grup target, pendengar, dan instance, untuk mengilustrasikan strategi EC2 lengket penyeimbang beban tertentu.

Kasus penggunaan umum

Gunakan Application Load Balancer tanpa lengket dalam skenario berikut:

- Anda memiliki daftar target untuk mengarahkan lalu lintas ke, tetapi target tidak mempertahankan status sesi.
- Anda menggunakan server web yang tidak mempertahankan status sesi.
- Anda menggunakan server aplikasi yang tidak mempertahankan status sesi.

Langkah-langkah

> **Catatan**
>
> - Gateway NAT dikenakan biaya kecil.
> - Beberapa EC2 instans menggunakan jam tingkat gratis Anda lebih cepat daripada satu EC2 instans.

1. Terapkan CloudFormation template `basic.yml` di lingkungan lab.

2. Tunggu sampai status kesehatan kelompok sasaran Anda berubah dari awal menjadi sehat.
3. Arahkan ke URL Application Load Balancer di browser web, menggunakan HTTP (TCP/80).

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

   Halaman web menampilkan Instance 1 - TG1 atau Instance 2 - TG1.
4. Refresh halaman beberapa kali.

Hasil yang diharapkan

Instance yang memuat halaman web (Instance 1 atau Instance 2) harus berubah setiap saat, seperti yang tercermin dalam teks halaman. Logika penyeimbang beban mengelola target terakhir di beberapa node internal, yang dapat menyebabkan penundaan sinkronisasi, jadi ada kemungkinan Anda akan diarahkan ke target yang sama.

Cara kerjanya

- Dalam contoh ini, dua EC2 instance ditugaskan ke satu grup target. EC2Instance memiliki server web Apache (`httpd`) diinstal, dan teks `index.html` halaman pada setiap EC2 instance di-hardcode untuk mengidentifikasi instance itu.
- Application Load Balancer menjalankan logika round robin internal untuk menentukan EC2 instance mana yang harus menerima lalu lintas.
- Setiap kali Anda memuat ulang halaman web, Application Load Balancer menjalankan logika routingnya, dan halaman menampilkan Instance 1 TG1 - atau Instance 2 -. TG1

Kelengketan kelompok sasaran

Bila Anda menggunakan Application Load Balancer dengan kelengketan grup target:

- Application Load Balancer menggunakan bobot kelompok target untuk menentukan bagaimana menyeimbangkan lalu lintas yang masuk antara kelompok sasaran.
- Secara default, Application Load Balancer menggunakan metode round robin untuk merutekan permintaan ke EC2 instance di grup target tujuan.

Template: Gunakan CloudFormation template `targetgroupstickiness.yml` (termasuk dalam contoh kode file.zip) untuk mencoba kelengketan kelompok target.

Kasus penggunaan umum

Gunakan kelengketan grup target dalam skenario ini:

- Ada beberapa grup target yang ditugaskan ke penyeimbang beban, dan lalu lintas dari klien harus secara konsisten diarahkan ke instance dalam grup target tersebut.
- Penerapan biru/hijau.

Perubahan kode dari basic.yml.

Satu perubahan telah dilakukan pada listener: Kami memodifikasi tindakan default Application Load Balancer untuk menentukan dua grup target `TG1` (`TG2`dan) dengan bobot yang sama, dengan konfigurasi stickiness.

basic.yml

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

targetgroupstickiness.yml

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

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

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

Langkah-langkah

> **Catatan**
>
> - Gateway NAT dikenakan biaya kecil.
> - Beberapa EC2 instans menggunakan jam tingkat gratis Anda lebih cepat daripada satu EC2 instans.

1. Terapkan CloudFormation template `targetgroupstickiness.yml` di lingkungan lab.
2. Tunggu sampai status kesehatan kelompok sasaran Anda berubah dari awal menjadi sehat.
3. Arahkan ke URL Application Load Balancer di browser web, menggunakan HTTP (TCP/80).

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

   Halaman web menampilkan salah satu dari berikut ini: Instance 1 - TG1, Instance 2 - TG1, Instance 3 - TG2, atau Instance 4 - TG2.
4. Refresh halaman beberapa kali.

Hasil yang diharapkan

> **Note**
>
> CloudFormation Template dalam contoh ini mengkonfigurasi lengket untuk bertahan 10 detik.

Contoh yang memuat halaman web harus tetap berada dalam grup target (TG1 atau TG2) dalam durasi 10 detik, seperti yang tercermin dalam teks halaman.

Setelah sekitar 10 detik, kelengketan dilepaskan dan kumpulan instance grup target mungkin berubah.

Cara kerjanya

- Dalam contoh ini, empat EC2 contoh dibagi menjadi dua kelompok target, dengan dua instance per grup target. EC2 Instans memiliki server web Apache (`httpd`) diinstal, dan teks `index.html` halaman pada setiap EC2 instance di-hardcode agar berbeda.
- Application Load Balancer membuat pengikatan untuk sesi pengguna menuju grup target tujuan, dengan waktu kedaluwarsa.
- Saat Anda memuat ulang halaman, Application Load Balancer memeriksa apakah pengikatan ada dan belum kedaluwarsa.
  - Jika pengikatan telah kedaluwarsa atau tidak ada, Application Load Balancer menjalankan logika peruteannya dan menentukan grup target tujuan.
  - Jika pengikatan belum kedaluwarsa, Application Load Balancer merutekan lalu lintas ke grup target yang sama, tetapi tidak harus ke instance yang EC2 sama.

Sesi lengket dengan cookie yang dihasilkan penyeimbang beban

Saat Anda menggunakan Application Load Balancer dengan cookie yang dihasilkan penyeimbang beban:

- Application Load Balancer menggunakan bobot kelompok target untuk menentukan bagaimana menyeimbangkan lalu lintas yang masuk antara kelompok sasaran.
- Secara default, Application Load Balancer menggunakan metode round robin untuk merutekan permintaan ke EC2 instance di grup target tujuan.

  Setelah lalu lintas awalnya diarahkan ke sebuah instance, lalu lintas berikutnya akan menempel pada EC2 instance itu untuk durasi tertentu.

Template: Gunakan CloudFormation template `stickysessionslb.yml` (termasuk dalam kode contoh file.zip) untuk mencoba sesi lengket dengan cookie yang dihasilkan penyeimbang beban.

Kasus penggunaan umum

Gunakan sesi lengket dengan cookie yang dihasilkan penyeimbang beban dalam skenario ini:

- Server web PHP
- Server yang menyimpan data sesi sementara seperti log, keranjang belanja, atau percakapan obrolan

Perubahan kode dari basic.yml.

Perubahan kode yang relevan ada dalam konfigurasi grup target, untuk mengatur jenis lengket `lb_cookie` dan durasinya menjadi 10 detik.

basic.yml	**stickysessionslb.yml**
`TG1:` `  Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'`	`TG1:` `  Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'`

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

```
    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: lb_cookie
        - Key: stickiness.lb_cook
ie.duration_seconds
Value: 10
```

Langkah-langkah

> **Catatan**
>
> - Gateway NAT dikenakan biaya kecil.
> - Beberapa EC2 instans akan menghabiskan jam tingkat gratis Anda lebih cepat daripada satu EC2 instans.

1. Terapkan CloudFormation template `stickysessionslb.yml` di lingkungan lab.
2. Tunggu sampai status kesehatan kelompok sasaran Anda berubah dari awal menjadi sehat.
3. Arahkan ke URL Application Load Balancer di browser web, menggunakan HTTP (TCP/80).

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

   Halaman web menampilkan salah satu dari berikut ini: Instance 1 - TG1, Instance 2 - TG1, Instance 3 - TG2, atau Instance 4 - TG1.
4. Refresh halaman beberapa kali.

Hasil yang diharapkan

> **Note**
>
> CloudFormation Template dalam contoh ini mengkonfigurasi lengket untuk bertahan 10 detik.

Contoh yang memuat halaman web harus tetap sama dalam durasi 10 detik, seperti yang tercermin dalam teks halaman. Setelah sekitar 10 detik, kekakuan dilepaskan dan instance tujuan mungkin berubah.

Cara kerjanya

- Dalam contoh ini, dua EC2 contoh hadir dalam satu kelompok target. EC2 Instans memiliki server web Apache (`httpd`) diinstal, dan teks `index.html` halaman pada setiap EC2 instance di-hardcode agar berbeda.
- Application Load Balancer membuat pengikatan untuk sesi pengguna, yang mengikat ke arah tujuan, dengan waktu kedaluwarsa.
- Saat Anda memuat ulang halaman, Application Load Balancer memeriksa apakah pengikatan ada dan belum kedaluwarsa.
  - Jika pengikatan telah kedaluwarsa atau tidak ada, Application Load Balancer menjalankan logika routing dan menentukan instance tujuan.
  - Jika pengikatan belum kedaluwarsa, Application Load Balancer merutekan lalu lintas ke instance tujuan yang sama.

Sesi lengket dengan cookie berbasis aplikasi

Saat Anda menggunakan Application Load Balancer dengan cookie berbasis aplikasi:

- Application Load Balancer menggunakan bobot kelompok target untuk menentukan bagaimana menyeimbangkan lalu lintas yang masuk antara kelompok sasaran.
- Secara default, Application Load Balancer menggunakan metode round robin untuk merutekan permintaan ke EC2 instance di grup target tujuan.
- Setelah lalu lintas awalnya dialihkan ke sebuah EC2 instance, respons aplikasi EC2 instance harus berisi cookie aplikasi khusus, yang dikirim kembali ke klien bersama dengan cookie Application Load Balancer otomatis.
- Lalu lintas berikutnya akan menempel pada EC2 instance jika klien mengirim kembali cookie aplikasi dan cookie Application Load Balancer.
- Cookie berbasis aplikasi kedaluwarsa setelah tidak digunakan selama durasi yang dikonfigurasi.

Template: Gunakan CloudFormation template `stickysessionsapp.yml` (termasuk dalam kode contoh file.zip) untuk mencoba sesi lengket dengan cookie berbasis aplikasi.

Kasus penggunaan umum

Gunakan sesi lengket dengan cookie yang dihasilkan aplikasi saat Anda menginginkan kontrol tambahan dalam skenario ini:

- Server web PHP
- Server yang menyimpan data sesi sementara seperti log, keranjang belanja, atau percakapan obrolan

Perubahan kode dari basic.yml.

Satu-satunya perubahan kode ada di konfigurasi grup target. Kami menambahkan konfigurasi stickiness ke Application Load Balancer dan atribut grup target. Durasi cookie aplikasi ditentukan, dan grup target mengaktifkan kekakuan cookie aplikasi.

basic.yml	**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
```

```
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
```

Langkah-langkah

> **Catatan**
>
> - Gateway NAT dikenakan biaya kecil.
> - Beberapa EC2 instans akan menghabiskan jam tingkat gratis Anda lebih cepat daripada satu EC2 instans.

1. Terapkan CloudFormation template `stickysessionslb.yml` di lingkungan lab.
2. Tunggu sampai status kesehatan kelompok sasaran Anda berubah dari awal menjadi sehat.
3. Arahkan ke URL Application Load Balancer di browser web, menggunakan HTTP (TCP/80).

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

Halaman web menampilkan salah satu dari berikut ini: Instance 1 - TG1, Instance 2 - TG1, Instance 3 - TG2, atau Instance 4 - TG2.

4. Refresh halaman beberapa kali.

Hasil yang diharapkan

> **Note**
>
> CloudFormation Template dalam contoh ini telah mengonfigurasi kelengketan untuk bertahan 10 detik. Konfigurasi durasi lengket yang valid adalah antara 1 detik dan 1 minggu.

Contoh yang memuat halaman web harus tetap sama, seperti yang ditunjukkan oleh teks halaman.

Durasi lengket tidak disegarkan, tetapi didasarkan pada kedaluwarsa yang dikonfigurasi dalam Application Load Balancer untuk cookie aplikasi yang dihasilkan oleh instance. EC2

Contoh 1: Tunggu 5 detik untuk me-refresh halaman. Contoh yang sama akan dimuat dan kekakuan akan disegarkan selama 10 detik lagi.

Contoh 2: Tunggu lebih dari 10 detik untuk memuat ulang halaman. Cookie aplikasi kedaluwarsa, dan Anda diarahkan ke instance yang berbeda EC2 . Instans baru ini menghasilkan cookie aplikasi lain dengan durasi 10 detik.

Cara kerjanya

- Dalam contoh ini EC2 instance memiliki server web Apache (`httpd`) diinstal. `httpd.conf`File dikonfigurasi untuk mengembalikan `Set-Cookie` nilai statis kembali ke klien (browser web Anda). `Set-Cookie`Nilainya di-hardcode menjadi. `TESTCOOKIE=<somevalue>`
- Buka opsi Inspect Element browser Anda, pilih tab Network, lalu pilih metode Get, yang memuat halaman. Anda akan melihat tab Cookies.
- Browser adalah aplikasi klien yang secara otomatis dikonfigurasi untuk mengembalikan pembaruan berikutnya ke server dengan cookie yang diterimanya dalam `Set-Cookie` respons server.
- Ketika Anda memuat ulang halaman, cookie yang diterima dalam pemuatan halaman awal secara otomatis dikirim kembali ke Application Load Balancer.

- Jika cookie telah kedaluwarsa (yaitu, 10 detik telah berlalu sejak Anda melakukan panggilan terakhir), Application Load Balancer menggunakan logika baru untuk menentukan instance EC2 mana yang akan mengarahkan lalu lintas.
- Jika cookie belum kedaluwarsa, Application Load Balancer merutekan lalu lintas ke instance yang sama. EC2

Sumber daya

- Sesi lengket untuk Application Load Balancer Anda (dokumentasi Elastic Load Balancing)

Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Bagian yang diperbarui	Memperbarui subnet penyeimbang beban dan bagian perutean dengan diagram dan klarifikasi baru.	Juli 5, 2024
Pembaruan dan koreksi	Memperbarui kode, diagram, dan contoh.	31 Mei 2023
Bagian yang diperbarui	Memperbarui bagian Cara kerjanya dalam kelengketan grup Target dan sesi Sticky dengan cookie yang dihasilkan penyeimbang beban untuk memberikan informasi yang lebih akurat dan terperinci.	18 Juli 2022
—	Publikasi awal	5 Agustus 2021

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.