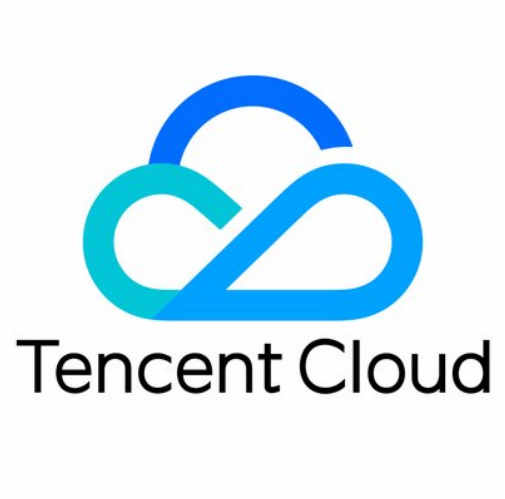
Cloud Load Balancer

Praktik Terbaik

Dokumen produk



Direktori dokumen

Praktik Terbaik
Konfigurasi Penerusan HTTPS

Waktu update terbaru：2024-01-04 20:56:41

1.CLB Capability Description (Deskripsi Kemampuan CLB)

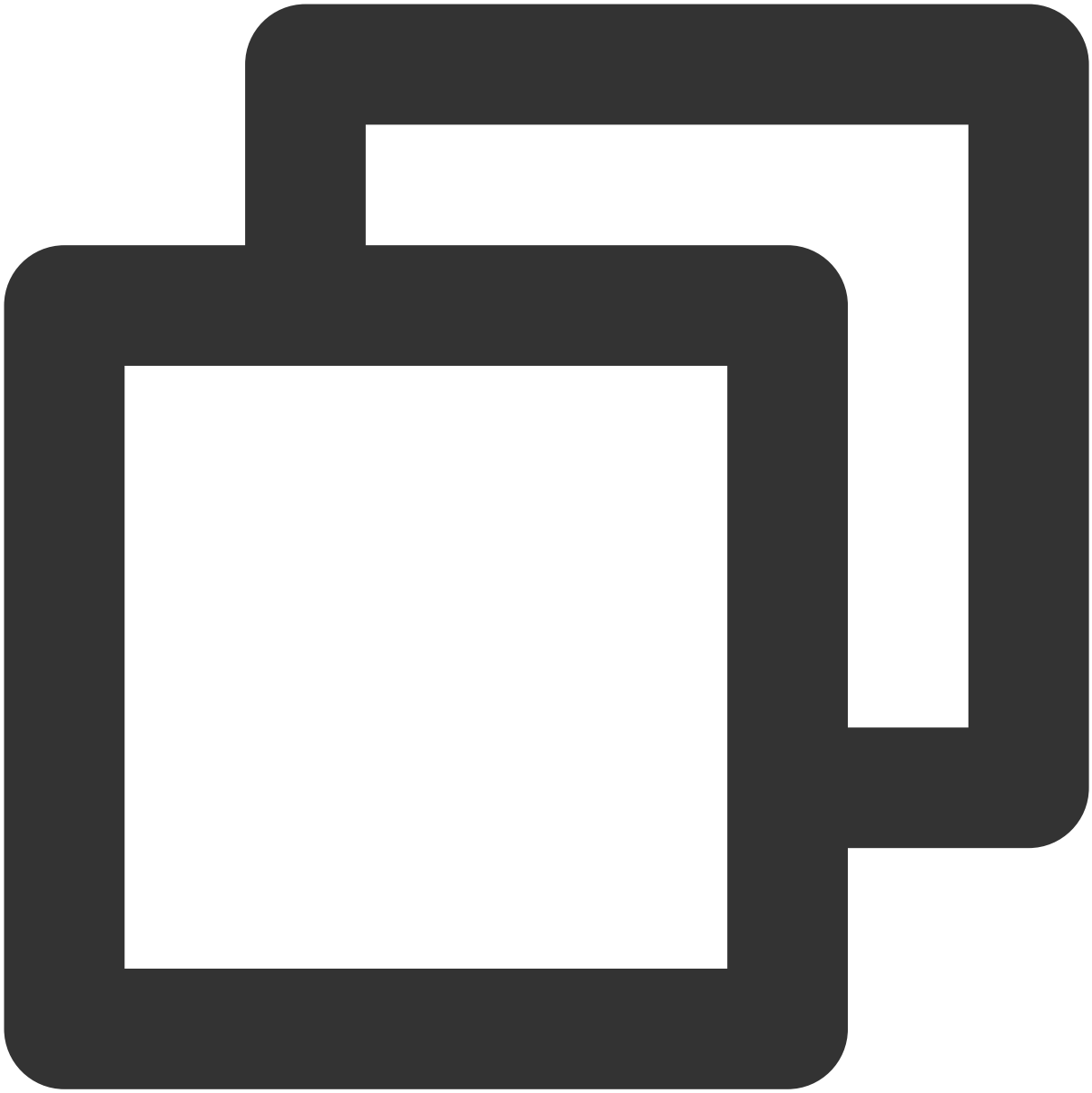
Dengan mengoptimalkan tumpukan protokol dan server secara mendalam, Tencent Cloud CLB mencapai peningkatan yang besar dalam kinerja HTTPS.Sementara itu, Tencent Cloud sangat menghemat biaya sertifikat melalui kerja sama dengan otoritas sertifikat internasional.CLB dapat menghadirkan manfaat besar bagi bisnis Anda dalam aspek-aspek berikut:
1.Penggunaan HTTPS tidak memengaruhi kecepatan akses klien.
2.Kinerja enkripsi dan dekripsi SSL dari satu server dalam sebuah kluster dapat mempertahankan handshake penuh hingga 65.000 koneksi per detik (CPS), yang sekurang-kurangnya 3,5 kali lebih tinggi daripada CPU berkinerja tinggi.Keunggulan ini mengurangi biaya server, sangat meningkatkan kemampuan layanan selama periode puncak bisnis dan lonjakan lalu lintas, serta memperkuat kemampuan tangkal serangan berbasis komputasi.
3.Offloading dan konversi beberapa protokol didukung, yang mengurangi tekanan bisnis dalam beradaptasi dengan beragam protokol klien.Backend bisnis hanya perlu mendukung HTTP/1.1 untuk menggunakan berbagai macam protokol, seperti HTTP/2, SPDY, SSL 3.0, dan TLS 1.2.
4.Layanan permohonan, pemantauan, dan penggantian sertifikat SSL satu atap disediakan.Tencent Cloud bekerja sama dengan Comodo dan SecureSite, dua otoritas sertifikat internasional, untuk menyederhanakan proses permohonan sertifikat dan mengurangi biaya permohonan tersebut.
5.Tersedia Fitur Anti-CC dan WAF untuk menghadang serangan lapisan-aplikasi, seperti serangan HTTP lambat, serangan dengan target frekuensi tinggi, injeksi SQL, dan trojan situs web.

2.HTTP and HTTPS Header Identifiers (Pengidentifikasi Header HTTP dan HTTPS)

CLB bertindak sebagai proksi untuk HTTPS.Baik permintaan HTTP maupun HTTPS akan menjadi permintaan HTTP ketika diteruskan ke instance CVM backend oleh CLB.Dengan demikian, Anda tidak dapat membedakan mana permintaan frontend yang dalam HTTP atau HTTPS.
CLB menanamkan `X-Client-Proto` ke dalam header saat meneruskan permintaan ke server asli:
X-Client-Proto: http (permintaan HTTP di frontend)
X-Client-Proto: https (permintaan HTTPS di frontend)

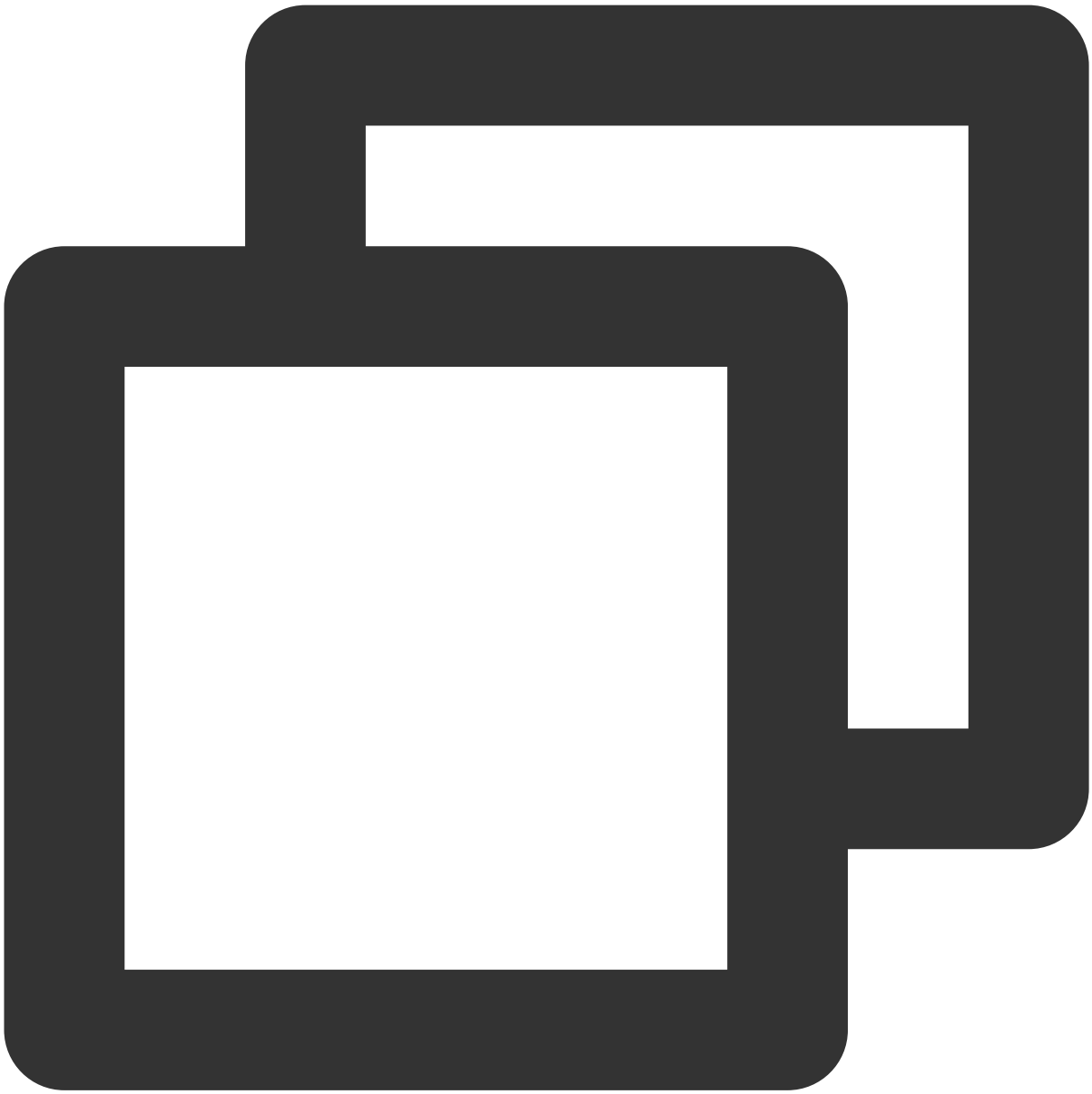
3.Memulai

Anggaplah bahwa Anda perlu mengonfigurasi situs web `https://example.com`, agar pengguna akhir dapat mengunjunginya secara aman melalui HTTPS ketika pengguna tersebut memasukkan `www.example.com` langsung di peramban.
Dalam hal ini, permintaan untuk mengakses `www.example.com` yang dimasukkan oleh pengguna akhir akan diteruskan seperti di bawah ini:
1. Permintaan ditransfer melalui HTTP dan mengakses port 80 pendengar CLB melalui VIP.Setelah itu, permintaan diteruskan ke port 8080 server asli.
2. Dengan konfigurasi penulisan ulang di Nginx di server asli, permintaan melewati port 8080 dan ditulis ulang di halaman `https://example.com`.
3. Setelah itu, peramban akan mengirimkan kembali permintaan `https://example.com` ke situs HTTPS yang sesuai.Permintaan tersebut mengakses port 443 pendengar CLB melalui VIP, kemudian diteruskan ke port 80 server asli.
Dengan demikian, proses penerusan permintaan telah selesai.
Operasi ini menuliskan ulang permintaan HTTP pengguna peramban menjadi permintaan HTTPS yang lebih aman dan tidak terlihat oleh pengguna.Untuk menerapkan operasi penerusan permintaan di atas, Anda dapat mengonfigurasi server asli sebagai berikut:



```
server {
listen 8080;
server_name example.qcloud.com;
location / {
#! customized_conf_begin;
client_max_body_size 200m;
rewrite ^/.(.*) https://$host/$1 redirect;
}
}
```

Atau, dalam versi baru Nginx, alihkan halaman HTTP Nginx ke halaman HTTPS menggunakan metode pengalihan 301 yang direkomendasikan:



```
server {
listen  80;
server_name    example.qcloud.com;
return  301 https://$server_name$request_uri;
}
server {
listen  443 ssl;
server_name    example.qcloud.com;
```

```
        [....]
}
```

Menerapkan HA di Beberapa AZ

Waktu update terbaru：2024-01-04 20:56:41

Menerapkan HA di Beberapa AZ

Instance CLB mendukung pemulihan bencana di seluruh zona ketersediaan.Misalnya, beberapa kluster dapat di-deploy di dua zona ketersediaan di wilayah yang sama:Zona 1 Hong Kong (Tiongkok) dan Zona 2 Hong Kong (Tiongkok).Hal ini mendukung tercapainya pemulihan bencana seluruh zona ketersediaan di wilayah yang sama.Dengan fitur ini, instance CLB dapat meneruskan lalu lintas akses frontend ke zona ketersediaan lainnya di wilayah yang sama dalam 10 detik apabila seluruh zona ketersediaannya gagal sehingga memulihkan kemampuan layanan.

Pertanyaan Umum dan Kasus Penggunaan

Pertanyaan 1:Apabila instance CLB `test1` dikonfigurasi untuk Zona 1 dan Zona 2 Hong Kong (Tiongkok), apa kebijakan untuk lalu lintas jaringan publik masuk klien?

Zona 1 dan Zona 2 Hong Kong (Tiongkok) mempunyai sepasang kumpulan sumber daya IP, yang dapat dianggap sebagai sumber daya IP dengan kemampuan penyeimbangan beban yang setara.Developer tidak perlu menentukan antara kluster master dan kluster slave.Ketika developer membeli instance CLB dan mengikatnya ke CVM, dua set peraturan akan dibuat dan dituliskan ke dalam dua kluster tersebut sehingga memberikan ketersediaan yang tinggi.

Pertanyaan 2:Misalkan instance CLB test1 dikonfigurasikan untuk Zona 1 dan 2 Hong Kong (Tiongkok), dan diikat ke 100 server asli di setiap zona ketersediaan.Selama operasi bisnis, 1 juta koneksi persisten HTTP (dengan tetap mengaktifkan koneksi TCP) akan tersambung di setiap zona.Apabila seluruh kluster CLB di Zona 1 gagal dan menjadi tidak tersedia, apa yang akan terjadi pada bisnis?

Ketika instance CLB di Zona 1 Hong Kong (Tiongkok) gagal, semua koneksi persistent pada saat itu akan ditutup, sementara koneksi non-persisten tidak akan terpengaruh.Arsitektur pemulihan bencana akan secara otomatis mengikat 100 server di setiap zona ke instance CLB di Zona 2 Hong Kong (Tiongkok) dalam 10 detik sehingga segera memulihkan kemampuan bisnis tanpa memerlukan intervensi manual.

Pertanyaan 3:Jenis CLB mana yang kompatibel dengan pemulihan bencana multi-AZ?Apakah dikenakan biaya tambahan?

Pemulihan bencana multi-AZ tersedia secara gratis.Fitur ini tersedia untuk instance CLB jaringan publik dan jaringan pribadi, kecuali instance CLB jaringan pribadi yang dibuat di Guangzhou sebelum 29 April 2020, di Shanghai sebelum 19 Desember 2019, dan di Beijing sebelum 18 Desember 2019.

Algoritme dan Konfigurasi Bobot

Waktu update terbaru：2024-01-04 20:56:41

Analisis Perbandingan Algoritme CLB

Penjadwalan round-robin tertimbang

How it works (Cara kerjanya)

Penjadwalan round-robin digunakan untuk menjadwalkan permintaan ke berbagai server berdasarkan polling, yaitu, sistem menjalankan `i = (i + 1) mod n` untuk memilih server i di setiap penjadwalan.Penjadwalan round-robin tertimbang dapat mengatasi ketidakseimbangan kinerja berbagai server.Algoritme ini memakai bobot untuk mewakili performa pemrosesan server dan menjadwalkan permintaan ke berbagai server berdasarkan bobot melalui polling.Server dengan bobot lebih tinggi akan lebih cepat mendapatkan koneksi, tetapi perlu memproses lebih banyak koneksi daripada server dengan bobot lebih rendah.Server dengan bobot yang sama memproses jumlah koneksi yang sama.

Advantages (Keuntungan)

Algoritme ini memberikan kemudahan dan kepraktisan tinggi.Algoritme ini merupakan algoritme penjadwalan stateless yang tidak mencatat status semua koneksi.

Disadvantages (Kekurangan)

Penjadwalan round-robin tertimbang ini cukup sederhana, tetapi tidak cocok untuk skenario dengan waktu layanan untuk suatu permintaan yang berubah secara signifikan, atau setiap permintaan membutuhkan jumlah waktu yang berbeda.Dalam skenario semacam ini, penjadwalan round-robin dapat menyebabkan ketidakseimbangan distribusi beban di antara server.

Use Cases (Kasus Penggunaan)

Algoritme ini cocok untuk skenario saat setiap permintaan pada dasarnya membutuhkan jumlah waktu yang sama di backend dengan performa pemuatan terbaik.Algoritme ini biasanya digunakan di layanan koneksi non-persisten seperti layanan HTTP.

Recommendations (Rekomendasi)

Jika Anda tahu bahwa setiap permintaan pada dasarnya membutuhkan jumlah waktu yang sama di backend (contohnya, permintaan yang diproses server asli berjenis sama atau mirip), sebaiknya Anda menggunakan penjadwalan round-robin tertimbang.Jika perbedaan waktu antara setiap permintaan kecil, sebaiknya Anda menggunakan algoritme ini karena memiliki tingkat konsumsi rendah, efisiensi tinggi, dan tidak membutuhkan traversal.

Penjadwalan koneksi terkecil tertimbang

How it works (Cara kerjanya)

Pada situasi sebenarnya, waktu yang dihabiskan setiap permintaan klien di server bisa sangat bervariasi.Jika round-

robin sederhana atau algoritme penyeimbangan beban acak digunakan seiring bertambah panjangnya waktu pengerjaan, jumlah proses koneksi di setiap server bisa sangat bervariasi dan penyeimbangan beban bisa saja tidak tercapai.Berbeda dengan penjadwalan round-robin, penjadwalan koneksi terkecil adalah algoritme penjadwalan dinamis yang memperkirakan beban server berdasarkan jumlah koneksi aktifnya.Penjadwal mencatat jumlah koneksi yang saat ini tersambung di setiap server.Jika ada permintaan yang dijadwalkan ke satu server, jumlah koneksinya bertambah 1.Jika koneksi terhenti atau habis waktu, jumlah koneksinya berkurang 1.Algoritme penjadwalan koneksi terkecil tertimbang didasarkan pada dan menyempurnakan penjadwalan koneksi terkecil.Bobot yang beragam dialokasikan ke server berdasarkan performa pemrosesannya.Dengan demikian, server akan menerima jumlah permintaan yang sesuai dengan bobotnya.

1.Misalkan bobot satu server asli adalah wi (i=1…n) dan jumlah koneksinya saat ini adalah ci (i=1…n).Server asli dengan nilai ci/wi terkecil akan menjadi server berikutnya yang menerima permintaan baru.
2.Server asli dengan nilai ci/wi yang sama akan dijadwalkan berdasarkan round-robin tertimbang.
Advantages (Keuntungan)
Algoritme ini cocok untuk permintaan yang membutuhkan waktu pemrosesan lama, seperti FTP.
Disadvantages (Kekurangan)
Karena adanya pembatasan API, koneksi terkecil dan persistensi sesi tidak dapat diaktifkan secara bersamaan.
Use Cases (Kasus Penggunaan)
Algoritme ini cocok untuk skenario ketika waktu yang digunakan oleh setiap permintaan di backend sangat bervariasi.Algoritme ini biasa digunakan di layanan koneksi persistensi.
Recommendations (Rekomendasi)
Jika Anda perlu memproses beberapa permintaan dan waktu layanannya di backend sangat bervariasi (misalnya, 3 milidetik dan 3 detik), sebaiknya Anda menggunakan penjadwalan koneksi terkecil tertimbang untuk mencapai keseimbangan beban.

Penjadwalan hashing sumber (ip_hash)

How it works (Cara kerjanya)
Penjadwalan hashing sumber menggunakan alamat IP sumber permintaan sebagai kunci hash dan menemukan server yang sesuai dari tabel hash yang ditetapkan secara statis.Permintaan akan dikirimkan ke server ini jika tersedia dan tidak kelebihan beban; jika tidak, nol akan dikembalikan.
Advantages (Keuntungan)
ip_hash dapat mencapai persistensi sesi tertentu dengan mengingat IP sumber dan memetakan permintaan dari klien ke server asli yang sama melalui tabel hash.Apabila persistensi sesi tidak didukung, ip_hash dapat digunakan untuk penjadwalan.
Recommendations (Rekomendasi)
Algoritme ini menghitung nilai hash alamat sumber sebuah permintaan dan mendistribusikan permintaan tersebut ke server asli yang sesuai berdasarkan bobotnya.Ini memungkinkan pendistribusian permintaan dari IP klien yang sama

ke server yang sama.Algoritme ini cocok untuk penyeimbangan beban melalui protokol TCP yang tidak mendukung cookie.

Memilih Algoritme Penyeimbangan Beban dan Mengonfigurasi Bobot

Dalam fitur CLB yang akan datang, **penerusan lapisan 7 akan mendukung metode penyeimbangan koneksi terkecil.** Kami menyediakan beberapa contoh untuk referensi Anda tentang cara memilih algoritme penyeimbangan beban dan mengonfigurasi bobot agar Anda dapat memastikan bahwa kluster server asli dapat menjalankan bisnis dalam berbagai skenario dengan stabil.

Skenario 1

Misalkan ada 3 server asli dengan konfigurasi yang sama (CPU dan memori) dan Anda mengonfigurasi semua bobotnya menjadi 10.Misalkan 100 koneksi TCP telah dibuat antara setiap server asli dan klien.Jika sebuah server asli baru ditambahkan, sebaiknya Anda menggunakan algoritme penjadwalan koneksi terkecil, yang bisa dengan cepat menambah beban server ke-4 dan mengurangi tekanan pada 3 server lainnya.

Skenario 2

Misalkan Anda menggunakan layanan Tencent Cloud untuk pertama kalinya.Situs web Anda baru saja dibuat dan memiliki beban rendah.Sebaiknya Anda membeli server asli dengan konfigurasi yang sama, karena semuanya merupakan server layer akses.Dalam skenario ini, Anda bisa mengonfigurasi bobot semua server asli ke 10 dan menggunakan algoritme penjadwalan round-robin tertimbang untuk mendistribusikan lalu lintas.

Skenario 3

Misalkan Anda memiliki 5 server asli untuk menjalankan permintaan akses sederhana ke situs web statis, dan rasio daya komputasinya (dihitung berdasarkan CPU dan memori) adalah 9:3:3:3:1.Dalam skenario ini, Anda dapat mengonfigurasi bobot tiap-tiap server itu menjadi 90, 30, 30, 30, dan 10.Karena sebagian besar permintaan akses ke situs web statis merupakan jenis koneksi non-persisten, Anda dapat menggunakan algoritme penjadwalan round-robin tertimbang agar CLB bisa mengalokasikan permintaan sesuai rasio performa server aslinya.

Skenario 4

Misalkan Anda memiliki 10 server asli untuk menjalankan permintaan akses web dalam jumlah yang masif, dan Anda tidak ingin membeli server lagi.Salah satu server tersebut sering dimulai ulang karena kelebihan beban.Dalam skenario ini, sebaiknya Anda mengonfigurasi bobot server yang sudah ada tersebut sesuai dengan performanya.Server dengan beban lebih tinggi sebaiknya memiliki bobot lebih rendah.Selain itu, Anda dapat menggunakan algoritme penjadwalan koneksi terkecil untuk mengalokasikan permintaan ke server asli dengan koneksi aktif yang lebih sedikit agar server tidak kelebihan beban.

Skenario 5

Misalkan Anda memiliki 3 server asli untuk memproses koneksi persisten, dan rasio daya komputasinya (dihitung dari CPU dan memori) adalah 3:1:1.Server dengan performa terbaik memproses lebih banyak permintaan, tetapi pastikan server itu tidak sampai kelebihan beban.Sebaiknya Anda mengalokasikan permintaan baru ke server yang sedang

tidak digunakan.Dalam skenario ini, Anda dapat menggunakan algoritme penjadwalan koneksi terkecil dan mengurangi bobot server yang penuh dengan tepat. Ini dilakukan agar CLB dapat mengalokasikan permintaan ke server dengan jumlah koneksi aktif yang lebih sedikit sehingga tercipta keseimbangan beban.

Skenario 6

Misalkan Anda ingin permintaan-permintaan berikutnya dari klien untuk dialokasikan ke server yang sama.Penjadwalan round-robin tertimbang atau koneksi terkecil tertimbang tidak menjamin bahwa permintaan dari klien yang sama akan dialokasikan ke server yang sama.Untuk memenuhi persyaratan server aplikasi pilihan Anda dan menjaga "kelekatan" (atau "kelanjutan") sesi klien, sebaiknya Anda menggunakan ip_hash untuk mendistribusikan lalu lintasnya.Algoritme ini memastikan bahwa semua permintaan dari klien yang sama akan didistribusikan ke server asli yang sama, kecuali jika jumlah servernya berubah atau server menjadi tidak tersedia.

Mengonfigurasi Perlindungan WAF untuk nama domain pendengar CLB

Waktu update terbaru：2024-01-04 20:56:41

Ketika mengikat nama domain dengan pendengar CLB, CLB Web Application Firewall (WAF) dapat mendeteksi dan memblokir lalu lintas HTTP atau HTTPS yang melewati pendengar CLB.Dokumen ini memperkenalkan cara menggunakan CLB WAF untuk menerapkan perlindungan keamanan Web untuk nama domain yang ditambahkan ke CLB.

Prasyarat

Saat ini CLB WAF tersedia dalam versi beta. Jika Anda ingin mencobanya, silakan Ajukan Permohonan.
Anda telah berhasil membuat pendengar HTTP atau HTTPS, dan nama domainnya dapat diakses.Untuk informasi selengkapnya, silakan lihat Memulai CLB.
Anda telah berhasil membeli layanan CLB WAF.Untuk informasi selengkapnya, silakan lihat Panduan Pembelian.

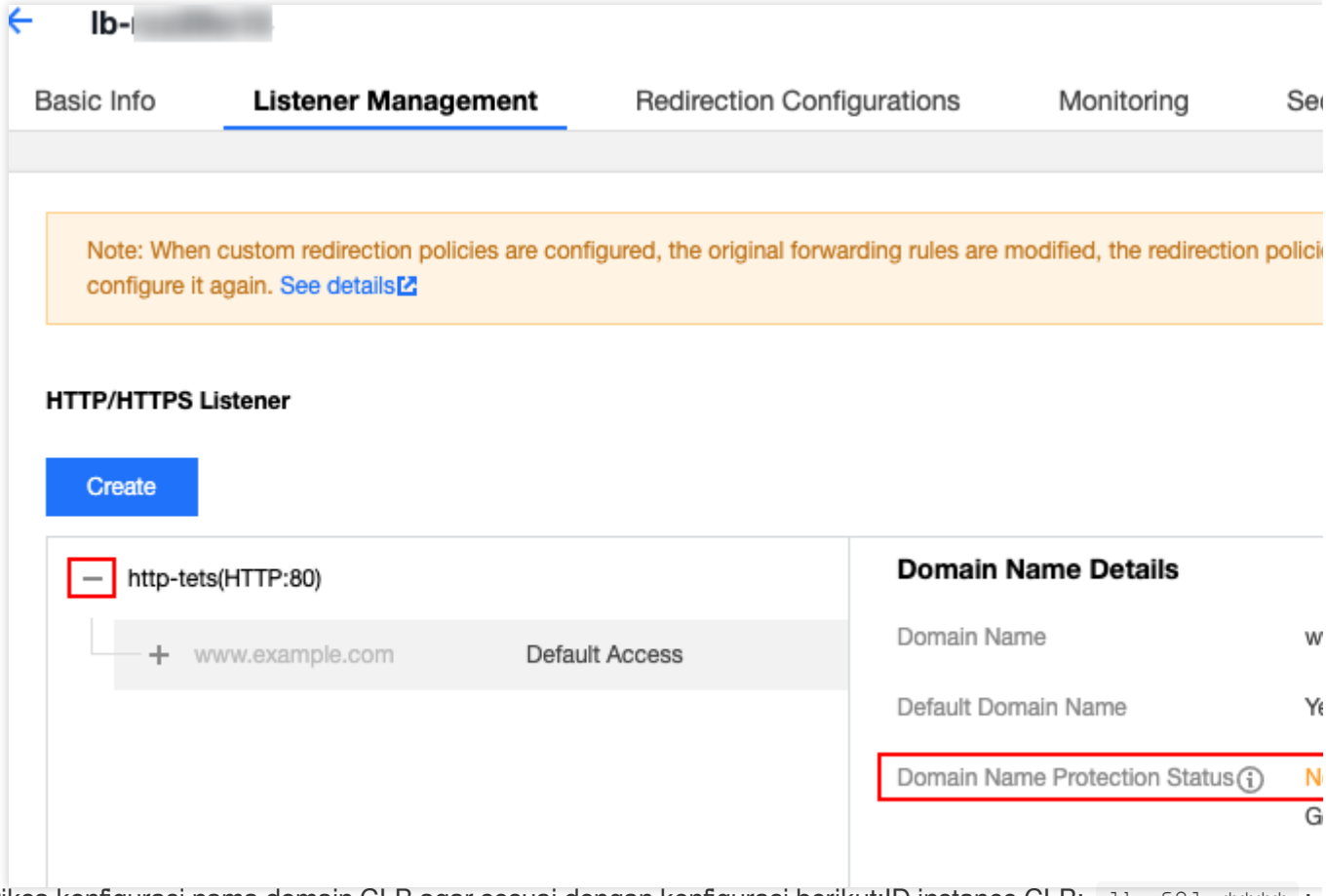
Batas

Saat ini, hanya instance IPv4 CLB yang mendukung perlindungan CLB WAF, fitur ini tidak tersedia untuk IPv6 dan NAT64.

Petunjuk

Langkah 1:Konfirmasi konfigurasi nama domain CLB

Dokum
en ini m
enggunakan nama domain `www.example.com` sebagai contoh.
1. Masuk ke Konsol CLB, klik **CLB Instance List** (Daftar Instance CLB) di bilah sisi kiri untuk masuk ke halaman **Instance Management** (Pengelolaan Instance).
2. Pilih wilayah instance, kemudian klik **Configure Listener** (Konfigurasi Pendengar) di sebelah kanan instance target.
3. Pilih tab **Listener Management** (Manajemen Pendengar), di bagian **HTTP/HTTPS Listener** (Pendengar HTTP/HTTPS), klik tanda **+** di sebelah kiri pendengar target untuk melihat detail nama domain.

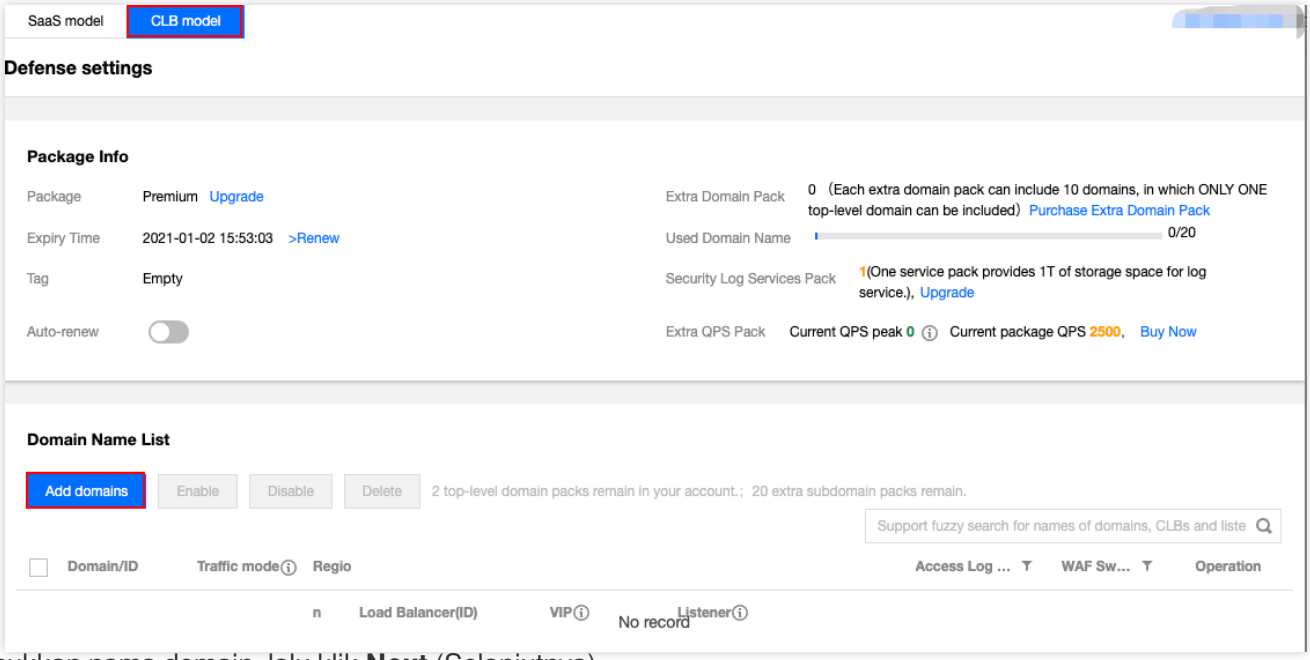


4. Periksa konfigurasi nama domain CLB agar sesuai dengan konfigurasi berikut:ID instance CLB: `lb-f8lm****` ; nama pendengar: `http-test` ; nama domain: `www.example.com` ; status perlindungan nama domain:**Not Enabled** (Tidak Aktif) (ID, nama, dan nama domain akan menyesuaikan dengan keadaan yang sebenarnya).

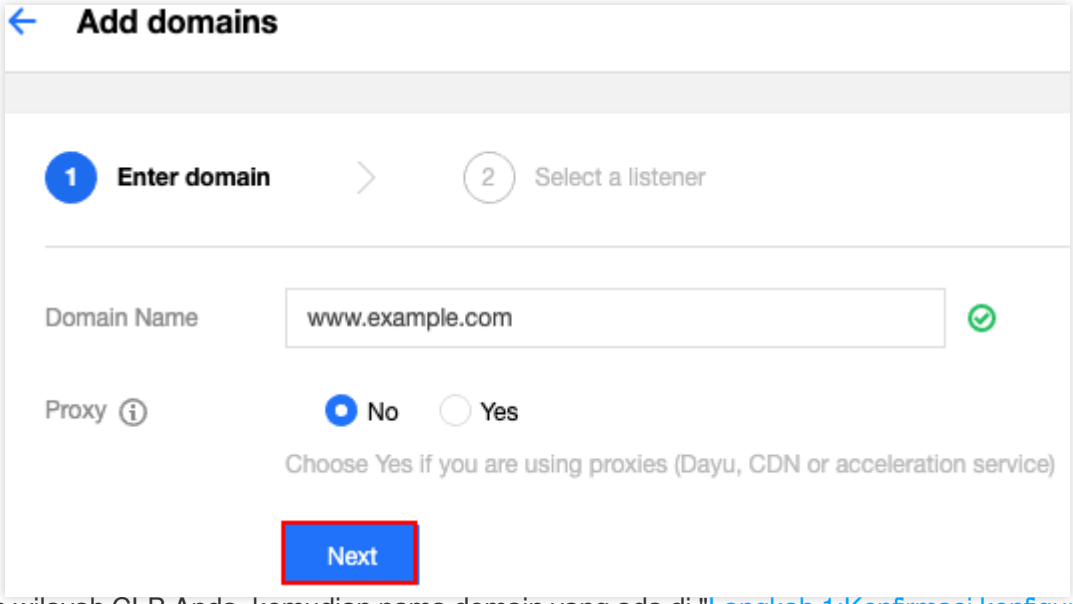
Langkah 2:Tambahkan nama domain di konsol WAF dan ikat nama domain tersebut ke instance CLB

Untuk menerapkan perlindungan ke nama domain dengan layanan CLB WAF, Anda perlu menambahkan nama domain pendengar CLB di WAF dan mengikatnya dengan pendengar CLB.

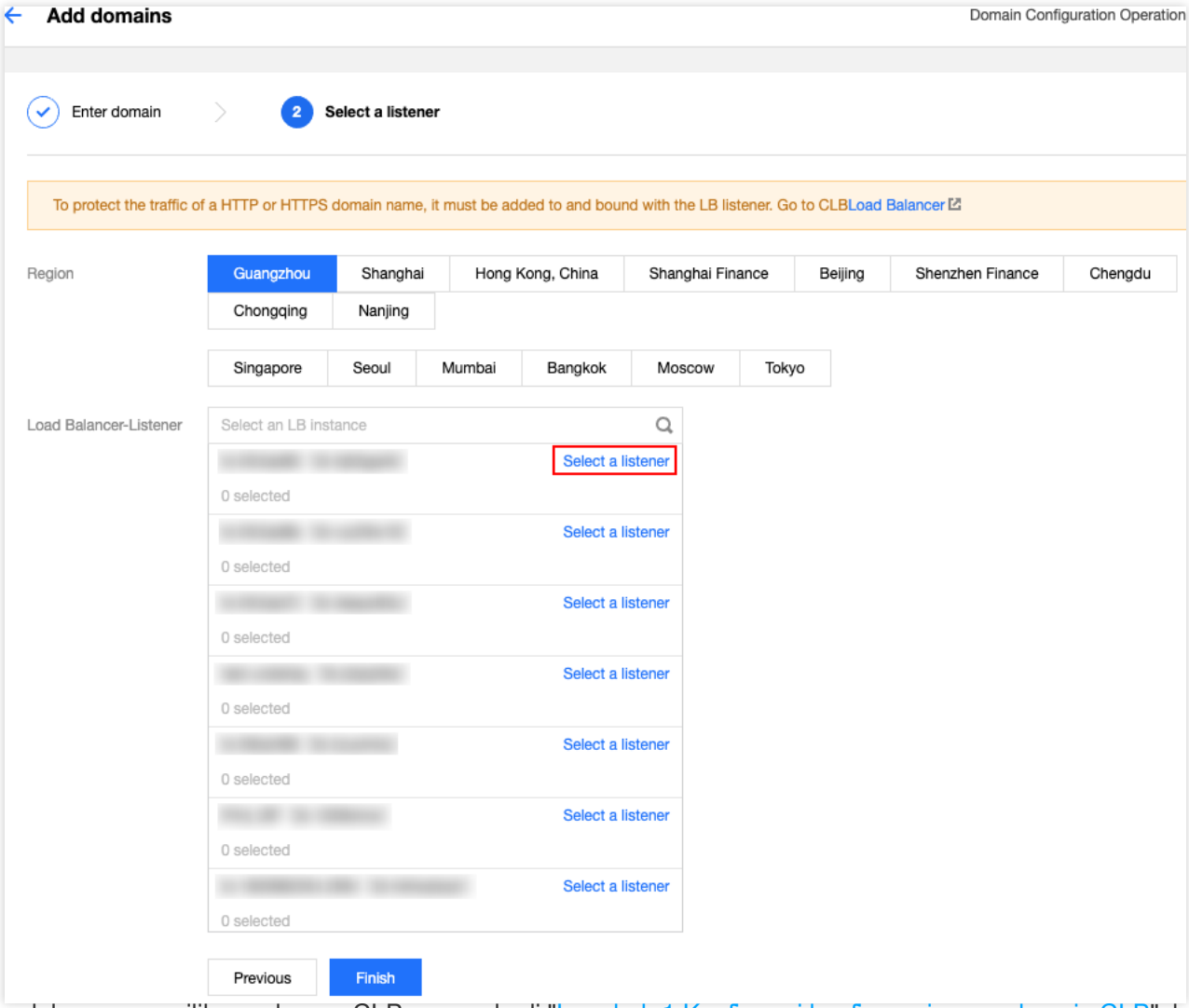
1. Masuk ke Konsol WAF, kemudian pilih **Web Application Firewall** -> **Defense Settings** (Firewall Aplikasi Web -> Pengaturan Pertahanan) di bilah sisi kiri.
2. Pilih tab **CLB**.
3. Klik **Add Domains** (Tambahkan Domain).



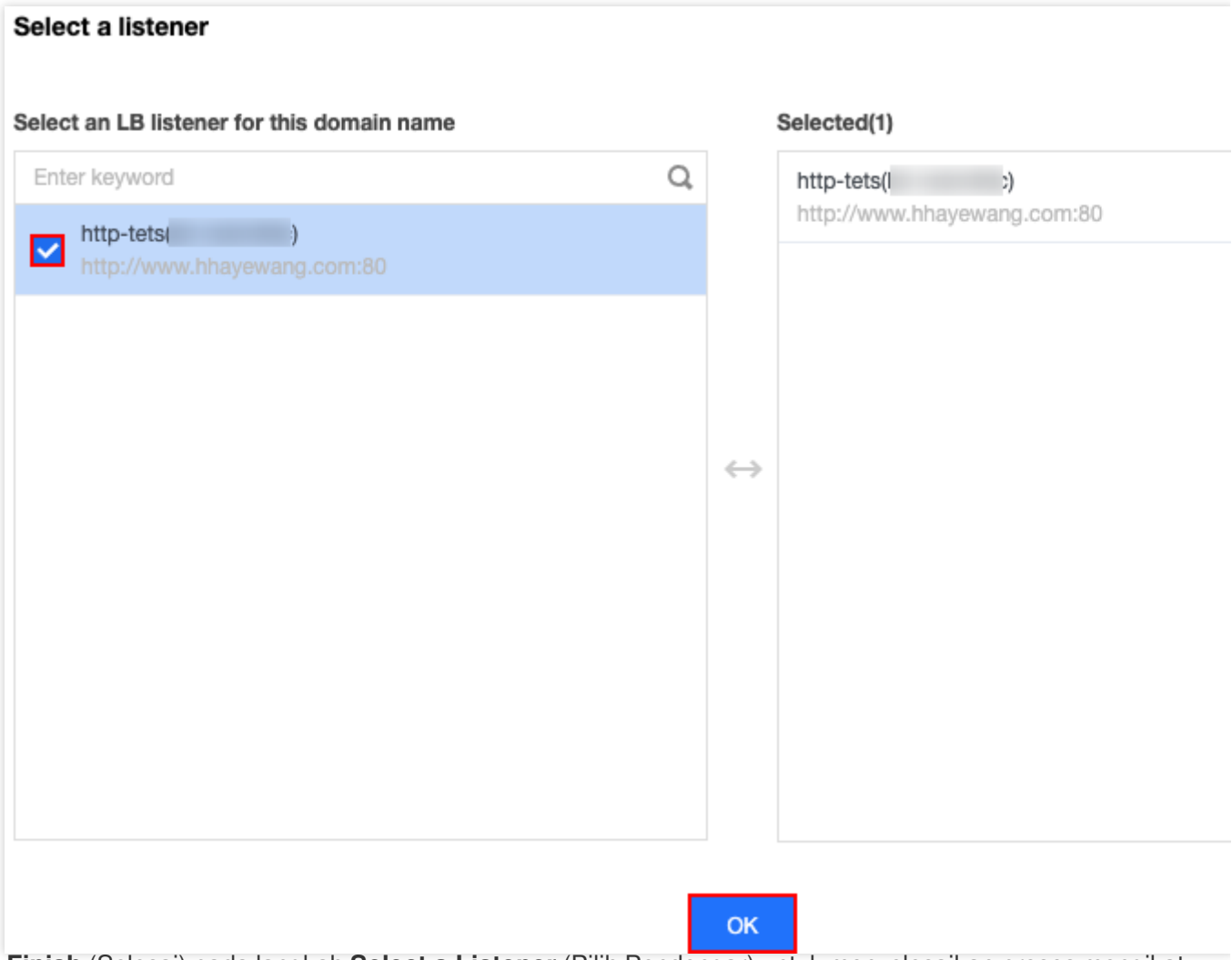
4. Masukkan nama domain, lalu klik **Next** (Selanjutnya).



5. Pilih wilayah CLB Anda, kemudian nama domain yang ada di "Langkah 1:Konfirmasi konfigurasi nama domain CLB", lalu klik **Select a Listener** (Pilih Pendengar).



6. Di jendela pop-up, pilih pendengar CLB yang ada di "Langkah 1:Konfirmasi konfigurasi nama domain CLB", lalu klik **OK**.



7. Klik **Finish** (Selesai) pada langkah **Select a Listener** (Pilih Pendengar) untuk menyelesaikan proses mengikat nama domain dengan pendengar CLB di WAF.
8. Kembalilah ke halaman **Domain List** (Daftar Domain), dan periksa nama domain, wilayah, ID instance CLB ikatan, pendengar, dan informasi lainnya.

Langkah 3:Verifikasi hasilnya

1. Ikuti petunjuk di "Langkah 1:Konfirmasi konfigurasi nama domain CLB" untuk memeriksa nama domain.Perlindungan nama domain akan berfungsi apabila perlindungan nama domainnya **Enabled** (Diaktifkan) dan mode lalu lintasnya adalah **Mirror** (Cermin).
Apabila Anda belum mengonfigurasi resolusi DNS untuk nama domain Anda, silakan lihat Langkah 2.Melakukan Pengujian Lokal untuk memastikan bahwa perlindungan WAF telah diaktifkan.
Apabila Anda sudah mengonfigurasi resolusi DNS untuk nama domain Anda, silakan ikuti petunjuk di bawah ini untuk memastikan bahwa perlindungan WAF telah diaktifkan.
2. Buka `http://www.example.com/?test=alert(123)` lewat peramban.
3. Masuk ke Konsol WAF, kemudian pilih **Log Services** -> **Attack Log** (Layanan Log -> Log Serangan) di bilah sisi kiri.

4. Pilih tab **Log Search** (Pencarian Log), pilih nama domain yang telah diberi perlindungan `www.example.com`, lalu klik **Search** (Cari).Perlindungan WAF untuk nama domain yang dikonfigurasi di CLB akan efektif apabila terdapat log **XSS Attack** (Serangan XSS) di daftar log.