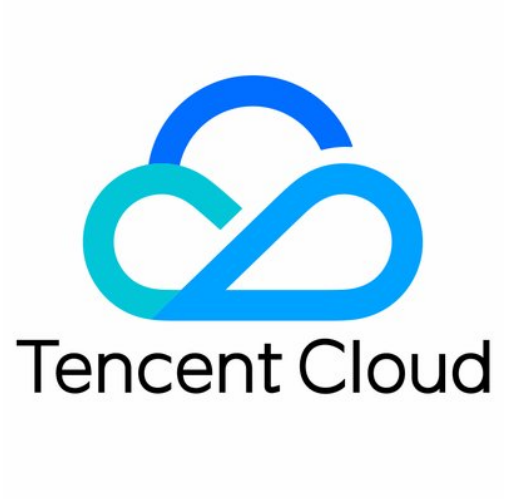
Cloud Load Balancer

Panduan OPS

Dokumen produk



Direktori dokumen

Panduan OPS
Solusi untuk Waktu Tunggu Klien yang Terlalu Lama

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

Latar belakang

Ketika menjalankan pengujian tekanan di CLB, koneksi Anda bisa gagal akibat terlalu banyak TIME-WAIT klien (semua port ditempati dalam waktu singkat).Berikut adalah alasan dan solusinya:

Deskripsi parameter Linux

tcp_timestamps: untuk mengaktifkan cap waktu TCPCap waktu dinegosiasikan dalam handshake tiga arah TCP.Tanpa dukungan kedua belah pihak, parameter ini tidak akan digunakan dalam koneksi ini.
tcp_tw_recycle: untuk mengaktifkan penggunaan ulang kondisi TIME-WAIT TCP.
tcp_tw_reuse: ketika diaktifkan, koneksi dalam kondisi TIME_WAIT yang melampaui 1 detik dapat digunakan ulang secara langsung.

Analisis Sebab

Klien memiliki TIME_WAIT terlalu banyak karena menutup koneksi secara proaktif.Ketika klien menutup koneksi, koneksi akan masuk ke dalam kondisi TIME_WAIT dan dapat digunakan kembali setelah 60 detik secara default.Dalam kasus ini, Anda dapat mengaktifkan parameter `tcp_tw_recycle` dan `tcp_tw_reuse` untuk memudahkan penggunaan ulang koneksi dalam kondisi TIME_WAIT.
Jika `tcp_timestamps` saat ini dinonaktifkan di CLB, parameter `tcp_tw_recycle` dan `tcp_tw_reuse` yang diaktifkan oleh klien tidak akan berlaku efektif, dan koneksi dalam kondisi TIME_WAIT tidak dapat digunakan kembali dengan cepat.Poin-poin berikut menjelaskan beberapa parameter Linux dan alasan mengapa `tcp_timestamps` tidak dapat diaktifkan di CLB:
1. tcp_tw_recycled dan tcp_tw_reuse hanya berlaku efektif ketika tcp_timestamps diaktifkan.
2. tcp_timestamps dan tcp_tw_recycle tidak dapat diaktifkan secara bersamaan karena klien jaringan publik gagal mengakses server melalui gateway NAT.Penyebabnya adalah sebagai berikut:
Jika tcp_tw_recycle dan tcp_timestamps sama-sama diaktifkan, cap waktu dalam permintaan koneksi soket dari IP sumber yang sama (server yang sama) pasti bertambah dalam 60 detik.Dengan mengambil kernel 2.6.32 sebagai

contoh, detailnya adalah sebagai berikut:

```
if (tmp_opt.saw_tstamp &&
    tcp_death_row.sysctl_tw_recycle &&
    (dst = inet_csk_route_req(sk, req)) ! = NULL &&
    (peer = rt_get_peer((struct rtable *)dst)) ! = NULL &&
    peer- >v4daddr == saddr) {
        if (get_seconds() < peer- >tcp_ts_stamp + TCP_PAWS_MSL &&
            (s32)(peer- >tcp_ts -  req- >ts_recent) >
                                TCP_PAWS_WINDOW) {
                NET_INC_STATS_BH(sock_net(sk), LINUX_MIB_PAWSPASSIVEREJEC
                goto ↓drop_and_release;
        }
}
```

Keterangan：

tmp_opt.saw_tstamp: soket ini mendukung tcp_timestamp

sysctl_tw_recycle: tcp_tw_recycle telah diaktifkan untuk server ini

TCP_PAWS_MSL: 60s; komunikasi TCP terakhir dari IP sumber berlangsung dalam 60 detik

TCP_PAWS_WINDOW: 1; cap waktu komunikasi TCP terakhir dari IP sumber lebih besar daripada cap waktu komunikasi TCP ini

3. Di CLB (Lapisan-7), tcp_timestamps dinonaktifkan karena klien jaringan publik gagal mengakses server melalui gateway NAT, seperti yang terlihat dalam contoh berikut:

a.Kuintupel masih dalam kondisi TIME_WAIT.Dalam kebijakan alokasi port dari gateway NAT, kuintupel yang sama digunakan kembali dua kali masa pakai segmen maksimum (2MSL), dan paket SYN dikirim.

b.Ketika tcp_timestamps diaktifkan dan dua kondisi berikut terpenuh, paket SYN akan anjlok (karena opsi cap waktu diaktifkan, dan paket dianggap lama).

i.Cap waktu terakhir kali > Cap waktu kali ini

ii.Paket diterima dalam 24 hari (bidang cap waktu adalah 32-bit dan cap waktu diperbarui satu kali per 1 millidetik secara default di dalam Linux. Cap waktu akan membungkus setelah 24 hari).

Catatan:Masalah ini lebih jelas di perangkat seluler karena klien sama-sama memiliki IP jaringan publik terbatas di bawah gateway NAT dari ISP dan kuintupel dapat digunakan kembali di 2MSL.Cap waktu yang dikirim dari klien berbeda mungkin tidak bertambah.

Dengan mengambil kernel 2.6.32 sebagai contoh, detailnya adalah sebagai berikut:

```
static inline int tcp_paws_check(const struct tcp_options_received *rx_opt,
                                 int paws_win)
{
      if ((s32)(rx_opt->ts_recent - rx_opt->rcv_tsval) <= paws_win)
            return 1;
      if (unlikely(get_seconds() >= rx_opt->ts_recent_stamp + TCP_PAWS_24DAYS))
            return 1;

      return 0;
}
```

Keterangan：

rx_opt->ts_recent: cap waktu terakhir kali

rx_opt->rcv_tsval: cap waktu yang diterima pada waktu ini

get_seconds(): waktu saat ini

rx_opt->ts_recent_stamp: waktu ketika paket sebelumnya diterima

Solusi

Jika klien memiliki terlalu banyak TIME_WAIT, lihat solusi di bawah:

1. **HTTP menggunakan koneksi tidak persisten (Koneksi: tutup).Dalam kasus ini, CLB secara proaktif menutup koneksi, dan klien tidak akan menghasilkan TIME_WAIT.**
2. **Jika skenarionya membutuhkan koneksi persisten, aktifkan opsi SO_LINGER pada soket dan gunakan RST untuk menutup koneksi guna menghindari kondisi TIME_WAIT dan mencapai penggunaan ulang port secara cepat.**

Uji Performa Layanan HTTPS Penyeimbang Beban

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

1.Kemampuan HTTPS dari CLB

Dengan mengoptimalkan tumpukan protokol dan server, Tencent Cloud CLB meningkatkan kinerja HTTPS secara signifikan.Selain itu, Tencent Cloud sangat menghemat biaya sertifikat melalui kerja sama internasional.Tencent CLB dapat menguntungkan bisnis Anda untuk hal-hal berikut:

1.Penggunaan HTTPS tidak memengaruhi kecepatan akses klien.

2.Kinerja enkripsi dan dekripsi SSL dari satu server dalam sebuah kluster mendukung 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, meningkatkan kemampuan layanan secara tajam selama jam-jam kerja bisnis dan puncak lalu lintas, serta memperkuat kemampuan tangkal serangan.

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 protokol dengan versi berbeda, seperti HTTP/2, SPDY, SSL 3.0, dan TLS 1.2.

4.Aplikasi, pemantauan, dan penggantian sertifikat SSL satu atap didukung.Tencent Cloud bekerja sama dengan dua otoritas sertifikat internasional, Comodo dan Symantec, untuk menyederhanakan proses aplikasi sertifikat dan menurunkan biaya.

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.Tujuan Pengujian

Layanan HTTPS memiliki keunggulan seperti autentikasi identitas, enkripsi informasi, dan verifikasi integritas.Akan tetapi, penggunaan protokol SSL untuk menerapkan komunikasi yang aman mengakibatkan hilangnya kinerja tertentu, termasuk meningkatnya latensi dan konsumsi sumber daya CPU oleh enkripsi dan dekripsi.Dokumen ini mencakup data pengujian kinerja ekstrem dari layanan HTTPS Tencent Cloud selama enkripsi dan dekripsi SSL.Anda dapat membandingkannya dengan data kinerja HTTPS tradisional.

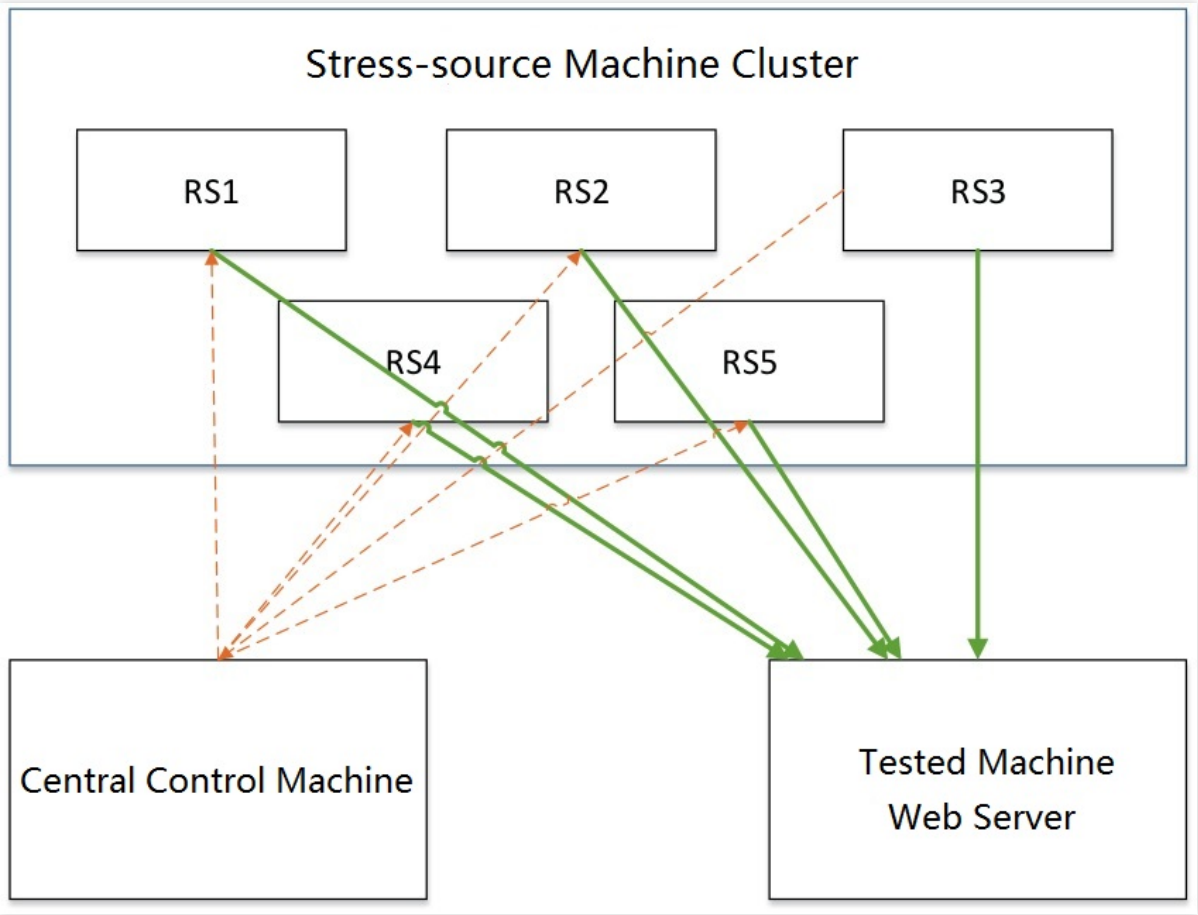
3.Lingkungan Pengujian

Alat pengujian tekanan: wrk 4.0.2

Lingkungan layanan penopang Tencent Cloud:Nginx 1.1.6–1.9.9 + OpenSSL 1.0.2h
Informasi tentang OS untuk menginstal Nginx:Linux TENCENT64.site 3.10.94-1-tlinux2-0036.tl2 # 1 SMP Kam Jan 21 03:40:59 CST 2016 x86_64 x86_64 x86_64 GNU/Linux
OS untuk server tekanan lainnya:Linux TENCENT64.site 2.6.32.43-tlinux-1.0.17-default #1 SMP Sel Nov 17 18:03:12 CST 2015 x86_64 x86_64 x86_64 GNU/Linux

4.Skema pengujian kluster WebServer

Tekanan dari satu server tidak cukup untuk menguji batas kinerja layanan HTTPS dari Tencent Cloud.Beberapa server tekanan diperlukan.Pengujian meliputi tiga bagian:
1.Kluster pengujian tekanan.Kluster pengujian ini digunakan untuk mendistribusikan tekanan HTTP/HTTPS dan memberikan output hasil pengujian tekanan dari satu server tekanan.
2.Server kontrol pusat, yang mengontrol permulaan dan akhir kluster pengujian tekanan, mendapatkan data pengujian dari setiap server tekanan lalu mengagregat dan memberikan output data.
3.Server web, berupa instance CVM yang menghosting layanan HTTPS dari Tencent Cloud.Ketika kinerja WebServer diuji, sebuah halaman dapat dikembalikan langsung tanpa koneksi upstream.
Koneksinya adalah sebagai berikut:



5.Data pengujian kinerja dari WebServer HTTPS

Jenis Koneksi	Cache Sesi	Ukuran Paket (dalam byte)	Paket Enkripsi	Kinerja (QPS)
Persisten	Aktif	230	ECDHE-RSA-AES128-GCM-SHA256	296241
Tidak Persisten	Nonaktif	230	ECDHE-RSA-AES128-GCM-SHA256	65630

6.Kesimpulan pengujian kemampuan HTTPS CLB

Menurut tabel di atas, layanan HTTPS dari Tencent Cloud mendukung enkripsi dan dekripsi SSL.Layanannya memiliki beberapa kluster server di backend, dan satu server dalam sebuah kluster dapat meraih kinerja hingga 65.000 QPS selama handshake penuh dan sekitar 300.000 QPS selama koneksi tanpa putus.
Dalam kondisi normal, protokol HTTPS menambahkan sedikitnya satu proses handshake penuh ketika menggunakan protokol SSL, dan latensi meningkat sebesar 2 * waktu round-trip (RTT).Selain itu, enkripsi simetris/asimetris SSL

mengonsumsi sumber daya CPU dalam jumlah besar.Kemampuan dekripsi RSA merupakan penghalang utama bagi akses berbasis HTTPS.

Dengan layanan HTTPS Tencent Cloud, Anda tidak perlu men-deploy layanan tambahan untuk enkripsi dan dekripsi SSL.Tanpa beban biaya tambahan, layanan ini memudahkan Anda menikmati kemampuan hosting bisnis dan tangkal serangan yang andal.

Pertanyaan Umum Seputar Pengujian Tekanan

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

Berdasarkan pengalaman pelanggan dalam pengujian tekanan, dokumen ini merangkum permasalahan umum kinerja dalam pengujian tekanan serta memberikan solusi dan saran penanggulangan masalah.

Pertanyaan Umum Seputar Pengujian Tekanan

1.Akses jaringan publik tidak diaktifkan di server asli

Jika akses jaringan publik tidak diaktifkan ketika Anda membeli CVM, penerusan bisa gagal ketika CLB jaringan publik dipasang di instance CVM.

2.Bandwidth dari server asli tidak cukup

Jika bandwidth-nya rendah, server asli tidak dapat mengembalikan paket ke CLB ketika ambang batas terlampaui.CLB akan mengembalikan kesalahan 504 atau 502 kepada klien.

3.Port klien tidak cukup

Jika jumlah klien terlalu kecil atau rentang port klien terlalu sempit, port klien menjadi tidak cukup dan koneksi akan gagal dibuat.Selain itu, jika nilai `keep_alive` lebih besar daripada 0 ketika koneksi persisten sudah jadi, koneksi akan menggunakan port secara permanen sehingga mengurangi jumlah port klien yang tersedia.

4.Aplikasi yang menjadi andalan server asli mengalami masalah kinerja

Setelah permintaan sampai di server asli melalui CLB, muatan di server asli menjadi normal.Meskipun demikian, karena aplikasi di server asli juga mengandalkan aplikasi lain seperti database, masalah kinerja di dalam database juga dapat memengaruhi kinerja pengujian tekanan.

5.Server asli tidak sehat

Status kesehatan server asli bisa diabaikan dalam pengujian tekanan.Jika server asli mengalami kegagalan pemeriksaan kesehatan atau status pemeriksaan kesehatan tidak stabil (kadang-kadang bagus, kadang-kadang buruk dengan perubahan cepat), kinerja pengujian tekanan bisa buruk.

6.Persistensi sesi yang diaktifkan untuk CLB mengakibatkan distribusi lalu lintas yang tidak merata di antara server asli

Setelah persistensi sesi diaktifkan untuk CLB, permintaan bisa didistribusikan kepada server asli tetap.Distribusi lalu lintas menjadi tidak merata sehingga memengaruhi kinerja pengujian tekanan.Anda sebaiknya menonaktifkan persistensi sesi selama pengujian tekanan.

Saran untuk pengujian tekanan

Keterangan：
Konfigurasi berikut hanya digunakan untuk pengujian tekanan CLB.Anda tidak harus memilikinya di lingkungan produksi Anda.
Anda sebaiknya menggunakan koneksi tidak persisten ketika tekanan menguji kemampuan meneruskan dari CLB.
Selain verifikasi terhadap fitur persistensi sesi, pengujian tekanan biasanya didesain untuk memverifikasi kemampuan meneruskan dari CLB.Oleh karena itu, koneksi tidak persisten dapat digunakan untuk menguji kemampuan pemrosesan CLB dan server asli.
Anda sebaiknya menggunakan koneksi persisten untuk melakukan pengujian tekanan terhadap throughput CLB, seperti batas atas bandwidth dan layanan koneksi persisten.
Anda sebaiknya menyesuaikan periode waktu habis alat pengujian tekanan ke nilai yang kecil.Jika tidak, rata-rata waktu respons akan meningkat ketika periode waktu habis juga meningkat sehingga membuat Anda tidak dapat dengan cepat menilai tercapai tidaknya level tekanan.
Anda sebaiknya menggunakan situs web statis yang disediakan oleh server asli untuk pengujian tekanan guna menghindari kerugian akibat logika aplikasi, seperti I/O dan DB.
Nonaktifkan persistensi sesi untuk pendengar.Jika tidak, tekanan akan terkonsentrasi pada server asli tertentu.Jika kinerja tekanan tidak memuaskan, Anda dapat menetapkan merata tidaknya distribusi lalu lintas dengan memeriksa data pemantauan server asli di CLB.
Nonaktifkan pemeriksaan kesehatan untuk pendengar agar dapat mengurangi permintaan akses ke server asli yang muncul selama pemeriksaan kesehatan.
Gunakan beberapa klien (> 5) untuk pengujian tekanan.IP sumber yang tersebar dapat meniru kondisi online aktual dengan lebih baik.

Izin Pengoperasian Sertifikat CLB

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

Skenario Pengoperasian

Sejak 23 Maret 2020, semua pengoperasian sertifikat CLB telah terhubung dengan Cloud Access Management (CAM) untuk autentikasi.Oleh karena itu, ketika akun subpengguna menjalankan pengoperasian sertifikat CLB, padahal "Anda tidak diotorisasi untuk pengoperasian ini.Silakan hubungi developer Anda."ditampilkan, Anda dapat memberikan izin sertifikat kepada akun subpengguna sesuai petunjuk berikut.

Prasyarat

Akun yang sudah masuk harus berupa akun root atau akun subpengguna dengan izin CAM (yaitu, berkaitan dengan kebijakan `QcloudCamFullAccess`).
Keterangan：
Untuk memeriksa apakah akun subpengguna memiliki izin CAM atau tidak, buka Daftar Pengguna di Konsol CAM, masuk ke halaman detail subpengguna dan periksa apakah kebijakan `QcloudCamFullAccess` telah berkaitan atau belum.
Jika kebijakan `QcloudCamFullAccess` sudah terkait, tetapi "Tidak ada izin API (message:GetReceiversOnAllType).Silakan hubungi developer Anda."ditampilkan ketika subpengguna menjalankan pengoperasian sertifikat, silakan diabaikan dan lanjutkan saja.

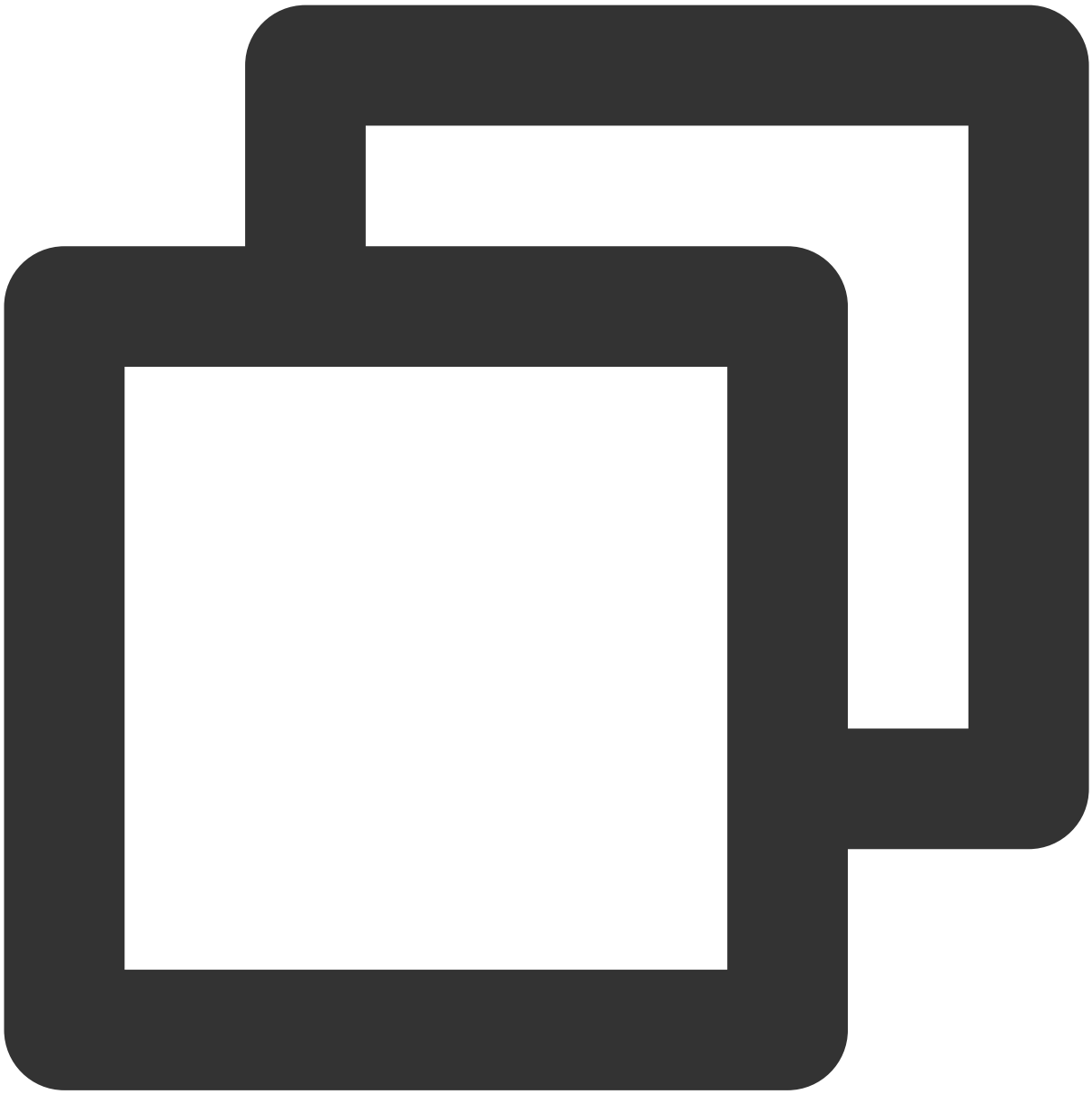
Petunjuk

Silakan beri izin sertifikat dengan metode berikut:

Metode 1.Kaitkan dengan kebijakan khusus

1. Masuk ke Konsol CAM.
2. Di bilah sisi kiri, klik **Policies** (Kebijakan).
3. Klik **Create Custom Policy** (Buat Kebijakan Khusus) dan pilih **Create by Policy Syntax** (Buat berdasarkan Sintaksis Kebijakan) di kotak pop-up.
4. Di halaman "Select Template Policy" (Pilih Kebijakan Templat), pilih **Blank Template** (Templat Kosong) lalu klik **Next** (Selanjutnya).

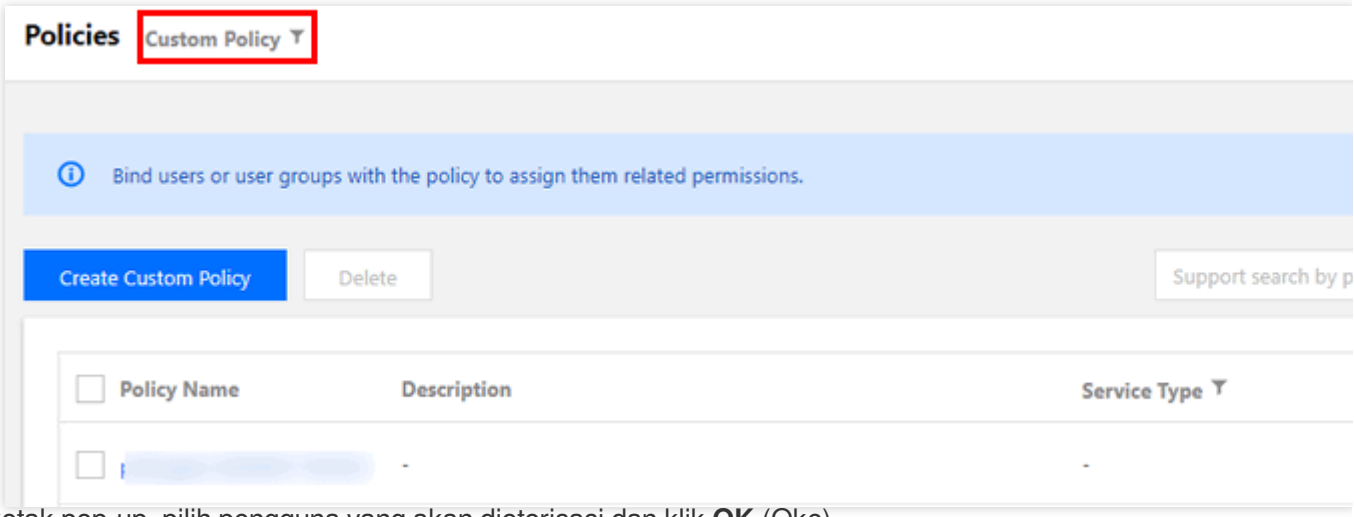
5. Di halaman "Edit Policy" (Edit Kebijakan), masukkan nama kebijakan dan masukkan konten kebijakan berikut ke dalam kotak input "Edit Policy Content" (Edit Konten Kebijakan):



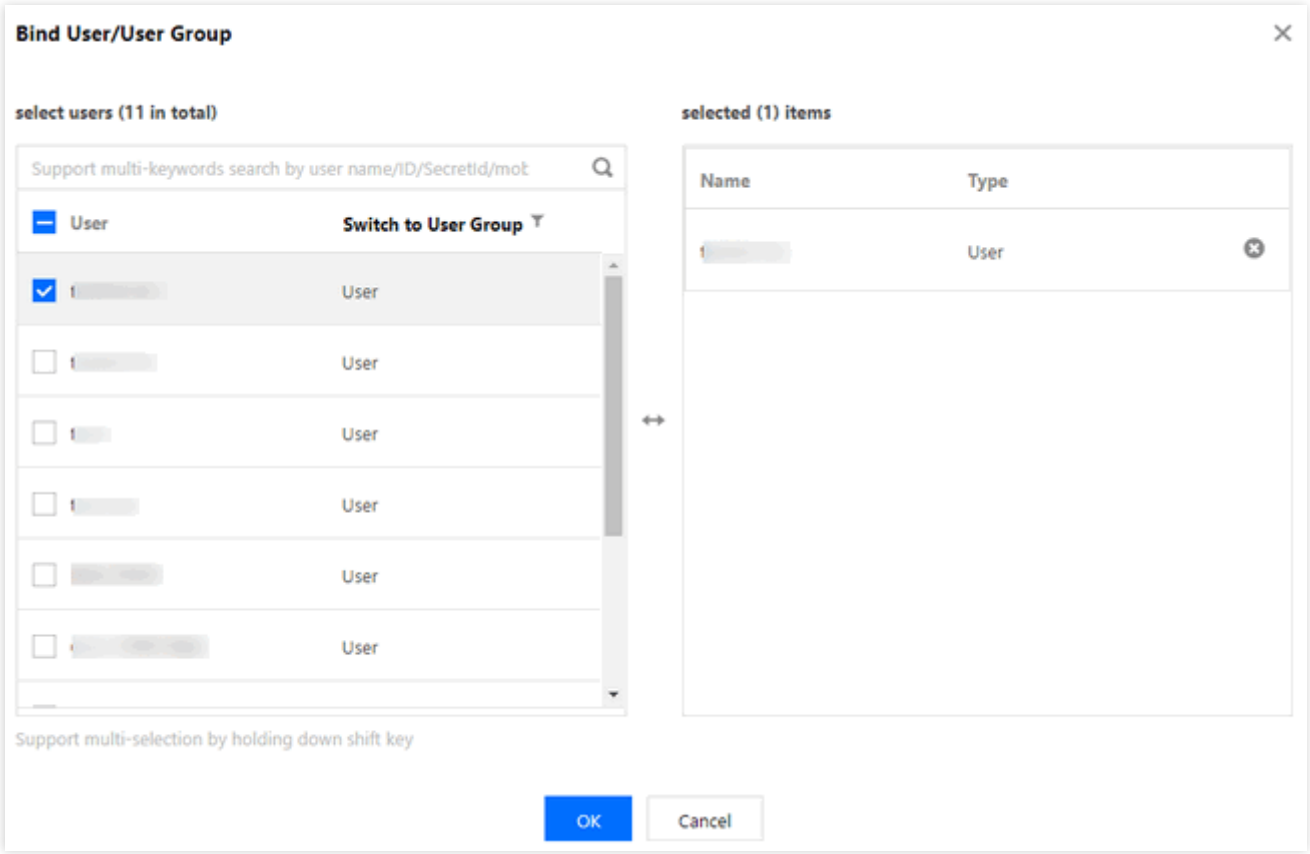
```
   {
"version":"2.0",
"statement": [
        {
"action": "name/ssl:*",
"resource": "qcs::ssl:::*",
"effect": "allow"
        }
```

```
        ]
    }
```

6. Lalu klik **Done** (Selesai) untuk kembali ke halaman daftar "Policy" (Kebijakan).
7. Di bagian atas halaman daftar "Policy" (Kebijakan), pilih **Custom Policy** (Kebijakan Khusus), cari baris kebijakan yang barusan Anda buat dalam daftar, dan klik **Associate User/Group** (Kaitkan Pengguna/Grup) di kolom "Operation" (Pengoperasian).



8. Di kotak pop-up, pilih pengguna yang akan diotorisasi dan klik **OK** (Oke).



Metode 2.Kaitkan kebijakan praatur

1. Masuk ke Konsol CAM.
2. Di bilah sisi kiri, pilih **User** > **User List** (Pengguna > Daftar Pengguna) untuk masuk ke halaman "User List" (Daftar Pengguna).
3. Di baris subpengguna yang akan diotorisasi, klik **Authorize** (Beri Otorisasi) di bilah "Operation" (Pengoperasian).
4. Di kotak pop-up, pilih `QcloudSSLFullAccess` atau `QcloudSSLReadOnlyAccess` dan klik **OK** (Oke).