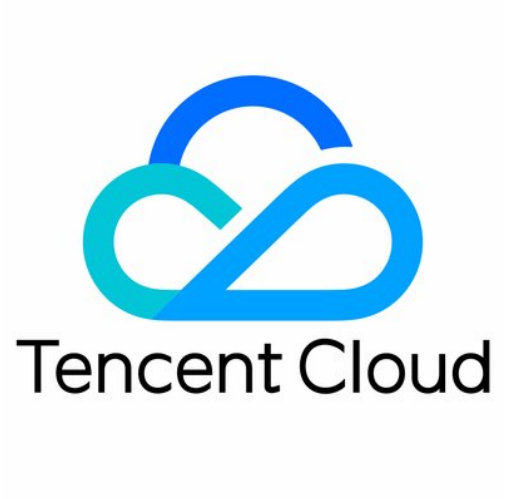
Cloud Load Balancer

操作ガイド

製品ドキュメント



カタログ：

操作ガイド
CLBインスタンス
ドメイン名型CLBアップグレードガイド

最終更新日：：2023-04-26 11:28:15

既存のパブリックネットワークCLBインスタンスをドメイン名型CLBインスタンスにアップグレードできます。アップグレード後CLBはドメイン名の方式でサービスを提供し、製品コンソールはVIP情報を表示しなくなります。業務リクエストの増加に伴い、VIPは業務リクエストに応じて動的に変化します。

アップグレード前後のCLBサービスの比較

比較項目	アップグレード後	アップグレード前
SLA	99.99%	99.95%
ドメイン名をサポートしているかどうか	はい	いいえ
VIPの自動拡張をサポートしているかどうか	サポート	サポートしません。
VIPが変化するかどうか	業務リクエストの増加に伴い、VIPは業務リクエストに応じて動的に変化し、コンソールはVIPアドレスを表示しなくなります	VIP固定
ヘルスチェックソースIP	デフォルトは100.64.0.0/10 ネットワークセグメント。アドレスの競合を効果的に回避	デフォルトのCLBインスタンスはVIP。100.64.0.0/10ネットワークセグメントを選択可能

制限事項

基幹ネットワーク内のインスタンスはアップグレードをサポートしていません。まずマイグレーションを完了してください。詳細はマイグレーションガイド をご参照ください。

従来型CLBはアップグレードをサポートしていません。まずCLBインスタンスにアップグレードしてください。詳細は従来型インスタンスのアップグレードをご参照ください。

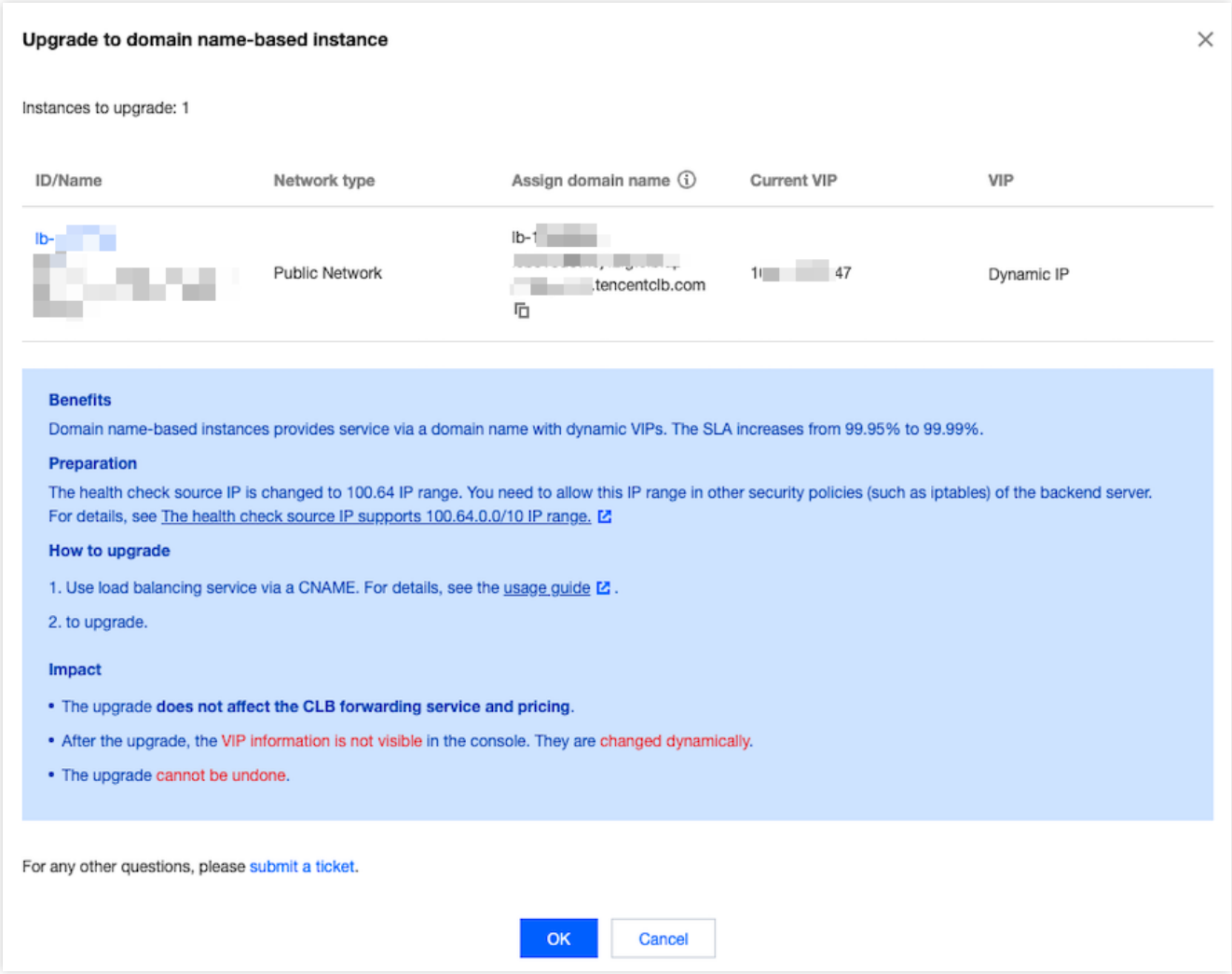
前提条件

1. クライアントの外部へのアクセスにCNAMEドメイン名解決を使用する方式を提供。
2. ヘルスチェックソースIPを100.64.0.0/10ネットワークセグメントに修正します。詳細はヘルスチェックソースIPの100.64.0.0/10ネットワークセグメントのサポートをご参照ください。

操作手順

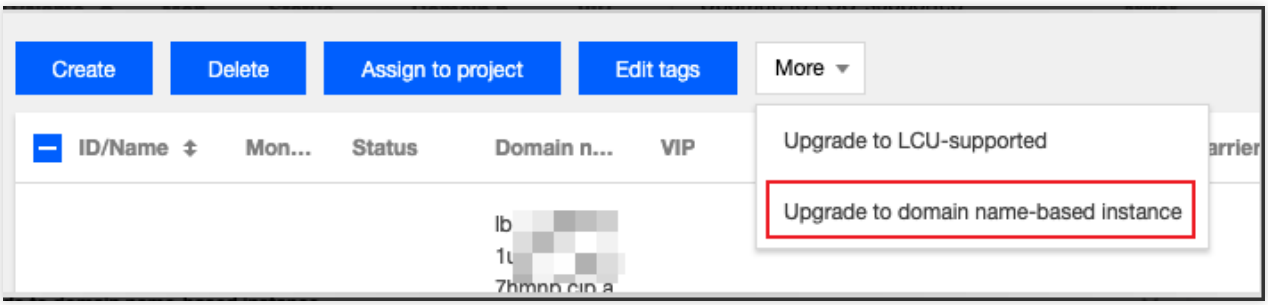
方法1：指定インスタンスのアップグレード

1. CLBコンソールにログインします。
2. **インスタンス管理**ページの左上隅でリージョンを選択し、インスタンスリストで目的のインスタンスを見つけ、右側操作バーの**その他 > ドメイン名型インスタンスにアップグレード**をクリックします。
3. **ドメイン名型インスタンスにアップグレード**ポップアップウィンドウで**OK**をクリックします。

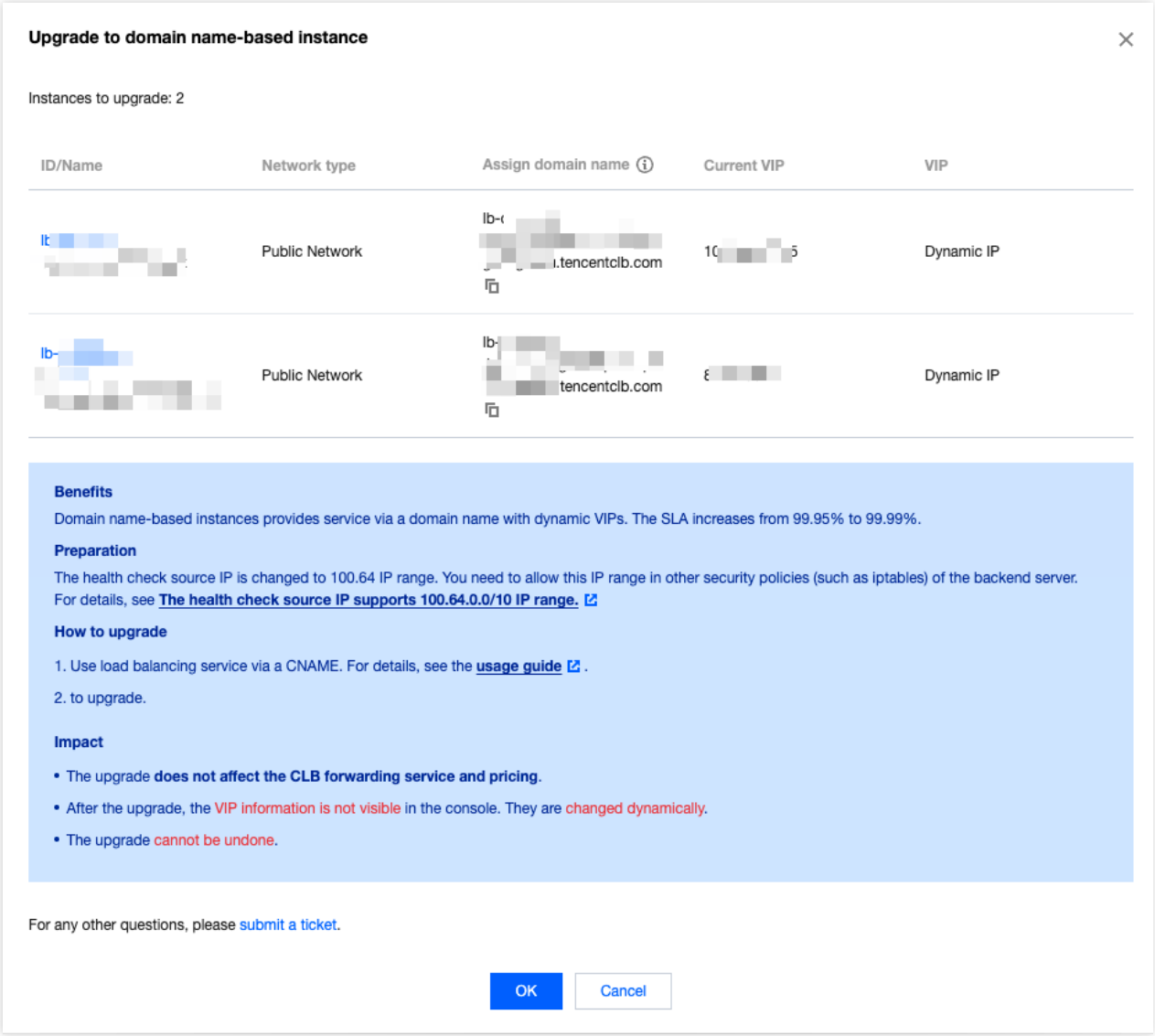


方式2：バッチインスタンスのアップグレード

1. CLBコンソールにログインします。
2. **インスタンス管理**ページの左上隅でリージョンを選択し、インスタンスリストでアップグレードしていないCLBインスタンスにチェックを入れます。
3. インスタンスリストの上で、**その他の操作** > **ドメイン名型インスタンスにアップグレード**を選択します。



4. **ドメイン名型インスタンスにアップグレード**ポップアップウィンドウで**OK**をクリックします。



CLBインスタンスの作成

最終更新日：：2024-01-04 17:48:23

Tencent Cloudでは公式サイト購入およびAPI購入という2種類のCLB購入方法を提供しています。ここではその2種類の購入方法をご紹介します。

公式Webサイトから購入

すべてのユーザーはTencent Cloud公式サイトからCLBをご購入いただけます。Tencent Cloudアカウントには標準アカウントタイプと従来型アカウントタイプがあり、2020年6月17日0時以降に登録したアカウントはすべて標準アカウントタイプとなります。この時点より前に登録したユーザーは、コンソールでアカウントタイプを確認してください。具体的な操作については、アカウントタイプの判断をご参照ください。

1. Tencent CloudのCLB購入ページにログインします。
2. 必要に応じて次のCLB関連設定を選択します。

標準アカウントタイプ

パラメータ	説明
課金モデル	従量課金モデルをサポートしています。
リージョン	所属リージョンを選択します。CLBのサポートリージョンの詳細については、リージョンリストをご参照ください。
インスタンスタイプ	CLBインスタンスタイプのみをサポートしています。従来型CLBは2021年10月20日をもって販売終了となります。詳細については、従来型CLB販売終了のお知らせをご参照ください。
ネットワークタイプ	ネットワークタイプにはパブリックネットワークとプライベートネットワークの2種類があります。詳細については、ネットワークタイプをご参照ください。 パブリックネットワーク：CLBを使用してパブリックネットワークからのリクエストを振り分けます。 プライベートネットワーク：CLBを使用して、Tencent Cloudプライベートネットワークのリクエストを振り分けます。プライベートネットワークは以下のElastic IP、IPバージョン、キャリアタイプ、インスタンス仕様、ネットワーク課金モデル、帯域幅上限の設定をサポートしていないため、これらの設定項目はデフォルトでは表示されません。 ネットワークタイプのサポート状況は課金モデルによって異なります。 従量課金モデルでは、パブリックネットワークとプライベートネットワークという2種類のネットワークタイプをサポートしています。
Elastic IP	Elastic IPを選択しない場合、Tencent CloudはパブリックCLBを割り当てます。パブリックIPを

	変更することはできません。 Elastic IPを選択する場合、Tencent CloudはElastic IPとプライベートネットワークCLBを1つずつ割り当てます。機能はパブリックCLBに類似しています。（従量課金モデルの場合、パブリックCLBのみがElastic IPの選択をサポートします） この機能はベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。使用制限については、使用制限をご参照ください。
IPバージョン	CLBのIPバージョンは、IPv4、IPv6、IPv6 NAT64から選択できます。従量課金モデルは、IPv6バージョンのみをサポートしています。その他の制限事項については、IPバージョンをご参照ください。IPv6バージョンのCLBは現在、ベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。
所属ネットワーク	CLBがサポートする所属ネットワークはClassic networkとVPCです。 基幹ネットワークとはTencent Cloud上のすべてのユーザーのための公共のネットワークリソースプールです。すべてのCVMのプライベートIPアドレスはTencent Cloudが一元的に割り当てており、ネットワークセグメントの区分、IPアドレスをカスタマイズすることはできません。 VPCとは、ユーザーがTencent Cloud上に構築した、論理的に隔離されているネットワークスペースです。VPC内では、ユーザーはネットワークセグメントの区分、IPアドレスおよびルーティングポリシーを自由に定義できます。 両者を比べた場合、プライベートネットワークは基幹ネットワークよりもネットワークのカスタマイズが必要なシナリオに適しています。基幹ネットワークの全製品は、2022年12月31日にオフラインとなります。詳細については、基幹ネットワークオフラインのお知らせをご参照ください。VPCを選択されることをお勧めします。
キャリアタイプ	キャリアタイプには、BGP（複数回線）、チャイナモバイル、チャイナテレコム、チャイナユニコムがあります。 従量課金モデルでは、以上4種類の選択肢をサポートしています。現在は広州、上海、南京、済南、杭州、福州、北京、石家荘、武漢、長沙、成都、重慶リージョンのみで静的単一IP回線タイプをサポートしています。その他のリージョンのサポート状況は、コンソールページでご確認ください。体験をご希望の場合はビジネスマネージャーにご連絡の上、お申し込みください。承認後、購入ページでチャイナモバイル、チャイナユニコムまたはチャイナテレコムのキャリアタイプを選択できるようになります。
マスター/スレーブアベイラビリティーゾーン	マスターアベイラビリティーゾーンとは現在トラフィックを担っているアベイラビリティーゾーンです。スレーブアベイラビリティーゾーンはマスターアベイラビリティーゾーンが使用できない場合に使用します。現在は広州、上海、南京、北京、中国香港、ソウルリージョンのIPv4バージョンのCLBのみマスター/スレーブアベイラビリティーゾーンをサポートしています。
インスタンス仕様	共有タイプインスタンスとLCUタイプインスタンスをサポートしています。 共有タイプインスタンスでは複数のインスタンスがリソースを共有し、単一のインスタンスで同時接続数最大5万、1秒あたりの新規接続数5000、1秒あたりの照会数(QPS)5000をサポートしています。

	LCUタイプインスタンスはパフォーマンスを保証するもので、互いにリソースを奪い合う共有タイプインスタンスはなく、転送パフォーマンスは他のインスタンスの影響を受けません。単一のインスタンスで同時接続数最大100万、新規接続数10万、1秒あたりの照会数5万をサポートします。
ネットワーク課金方式	ネットワーク課金モデルには、帯域幅課金（月額帯域幅）、帯域幅課金（時間単位帯域幅）、トラフィック課金、共有帯域幅パッケージがあります。 従量課金のインスタンス課金モデルは、帯域幅課金（1時間あたりの帯域幅）、使用トラフィック課金、共有帯域幅パッケージという3つのネットワーク課金モデルをサポートしています。現在、共有帯域幅パッケージはベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。
帯域幅の上限	1-1024Mbps。
プロジェクト	所属プロジェクトを選択してください。
タグ	タグキーとタグ値を選択するか、もしくはタグの新規作成を選択することもできます。詳細については、タグの作成をご参照ください。
インスタンス名	60文字まで入力でき、アルファベット、数字、中国語、「-」、「_」、「.」を使用できます。入力しない場合はデフォルトで自動生成されます。

従来型アカウントタイプ

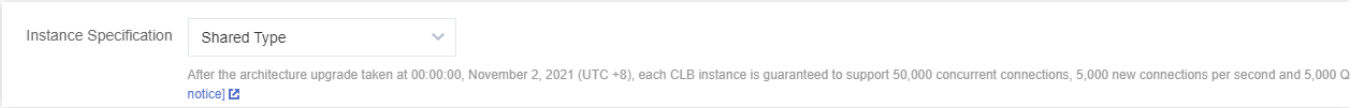
パラメータ	説明
課金モデル	従量課金モデルのみサポートしています。
リージョン	所属リージョンを選択します。CLBのサポートリージョンの詳細については、リージョンリストをご参照ください。
インスタンスタイプ	CLBインスタンスタイプのみをサポートしています。従来型CLBは2021年10月20日をもって販売終了となります。詳細については、従来型CLB販売終了のお知らせをご参照ください。
ネットワークタイプ	ネットワークタイプにはパブリックネットワークとプライベートネットワークの2種類があります。詳細については、ネットワークタイプをご参照ください。 パブリックネットワーク：CLBを使用してパブリックネットワークからのリクエストを振り分けます。 プライベートネットワーク：CLBを使用して、Tencent Cloudプライベートネットワークのリクエストを振り分けます。プライベートネットワークは以下のIPバージョン、キャリアタイプ、イ

	ンスタンス仕様の設定をサポートしていないため、これらの設定項目はデフォルトでは表示されません。
IPバージョン	CLBのIPバージョンは、IPv4、IPv6、IPv6 NAT64から選択できます。使用制限の詳細については、IPバージョンをご参照ください。IPv6バージョンのCLBは現在、ベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。
所属ネットワーク	CLBがサポートする所属ネットワークはClassic networkとVPCです。 基幹ネットワークとはTencent Cloud上のすべてのユーザーのための公共のネットワークリソースプールです。すべてのCVMのプライベートIPアドレスはTencent Cloudが一元的に割り当てており、ネットワークセグメントの区分、IPアドレスをカスタマイズすることはできません。 VPCとは、ユーザーがTencent Cloud上に構築した、論理的に隔離されているネットワークスペースです。VPC内では、ユーザーはネットワークセグメントの区分、IPアドレスおよびルーティングポリシーを自由に定義できます。 両者を比べた場合、プライベートネットワークは基幹ネットワークよりもネットワークのカスタマイズが必要なシナリオに適しています。基幹ネットワークの全製品は、2022年12月31日にオフラインとなります。詳細については、基幹ネットワークオフラインのお知らせをご参照ください。VPCを選択されることをお勧めします。
キャリアタイプ	キャリアタイプには、BGP（複数回線）、チャイナモバイル、チャイナテレコム、チャイナユニコムがあります。 現在は広州、上海、南京、済南、杭州、福州、北京、石家荘、武漢、長沙、成都、重慶リージョンのみで静的単一IP回線タイプをサポートしています。この機能は現在、ベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。その他のリージョンのサポート状況は、コンソールページでご確認ください。体験をご希望の場合はビジネスマネージャーにご連絡の上、お申し込みください。承認後、購入ページでチャイナモバイル、チャイナユニコムまたはチャイナテレコムのキャリアタイプを選択できるようになります。
インスタンス仕様	共有タイプインスタンスとLCUタイプインスタンスをサポートしています。 共有タイプインスタンスでは複数のインスタンスがリソースを共有し、単一のインスタンスで同時接続数最大5万、1秒あたりの新規接続数5000、1秒あたりの照会数(QPS)5000をサポートしています。 LCUタイプインスタンスはパフォーマンスを保証するもので、互いにリソースを奪い合う共有タイプインスタンスはなく、転送パフォーマンスは他のインスタンスの影響を受けません。単一のインスタンスで同時接続数最大100万、新規接続数10万、1秒あたりの照会数5万をサポートします。
プロジェクト	所属プロジェクトを選択してください。
タグ	タグキーとタグ値を選択するか、もしくはタグの新規作成を選択することもできます。詳細については、タグの作成をご参照ください。
インスタンス名	60文字まで入力でき、アルファベット、数字、中国語、「-」、「_」、「.」を使用できます。入力しない場合はデフォルトで自動生成されます。

3. 上記の設定が完了した後、購入数と料金をご確認の上、**今すぐ購入**をクリックします。
従量課金モデル：ポップアップした「確認」ダイアログボックスで**OK**をクリックします。
4. 購入に成功すると、CLBサービスがすぐにアクティブになり、CLBの設定を行って使用することができます。

共有タイプインスタンスの購入方法

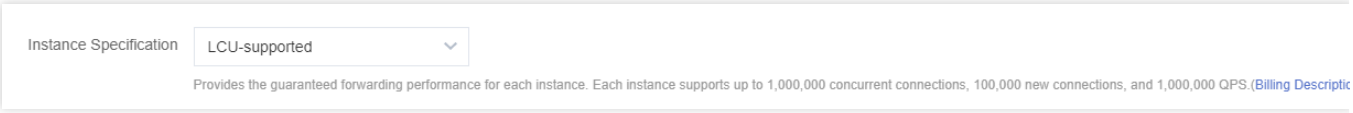
1. Tencent CloudのCLB購入ページにログインします。
2. 上記の公式サイト購入の操作手順を参照し、必要に応じて共有タイプCLBインスタンスの関連設定を選択し、「インスタンス仕様」で**共有タイプ**を選択します。



3. 引き続き、上記の公式サイト購入の操作手順を参照し、その後の操作を完了します。

LCUタイプインスタンスの購入方法

1. Tencent CloudのCLB購入ページにログインします。
2. 上記の公式サイト購入の操作手順を参照し、必要に応じてLCUタイプCLBインスタンスの関連設定を選択し、「インスタンス仕様」で**LCUタイプ**を選択します。



3. 引き続き、上記の公式サイト購入の操作手順を参照し、その後の操作を完了します。

API を介してインスタンスの購入

APIによるCLBの購入を希望するユーザーは、CLB API - CLBインスタンスの購入をご参照ください。

後続の操作

CLBにリスナーを作成したい場合は、CLBリスナーをご参照ください。
CLBのリスナーにバックエンドサービスをバインドしたい場合は、バックエンドサーバーをご参照ください。

関連ドキュメント

製品属性の選択

IPv6 CLBインスタンスの作成

最終更新日：：2024-01-04 17:48:23

説明：
IPv6 CLBはベータ版テスト中です。ご利用を希望される場合は、チケット申請を提出してください。
IPv6 CLBは現在、広州、上海、南京、北京、成都、重慶、中国香港、シンガポール、バージニアといったリージョンでのみサポートしています。
IPv6 CLBは従来型CLBをサポートしていません。
IPv6 CLBは、クライアントIPv6ソースアドレスの取得をサポートしています。レイヤー4のIPv6 CLBは、クライアントのIPv6ソースアドレスの直接取得をサポートしています。レイヤー7のIPv6 CLBは、HTTPのX-Forwarded-Forヘッダーフィールドを介したクライアントのIPv6ソースアドレスの取得をサポートしています。
現在、IPv6 CLBは純粋なパブリックCloud Load Balancerであり、同じVPCのクライアントがプライベートネットワークを通じてIPv6 CLBにアクセスすることはできません。
インターネットのIPv6ネットワークマクロ環境は構築の初期段階にあるため、ネットワークにアクセスできない状態が発生した場合は、チケットを提出してフィードバックしてください。また、ベータ版テスト期間中は、SLA保証は提供していません。

概要

IPv6 CLBは、IPv6シングルスタック技術をベースとして実装されたCLBで、IPv4 CLBと連携してIPv6/IPv4のデュアルスタック通信を実現します。IPv6 CLBはCloud Virtual Machine(CVM)のIPv6アドレスにバインドされ、IPv6 VIPアドレスを外部に提供します。

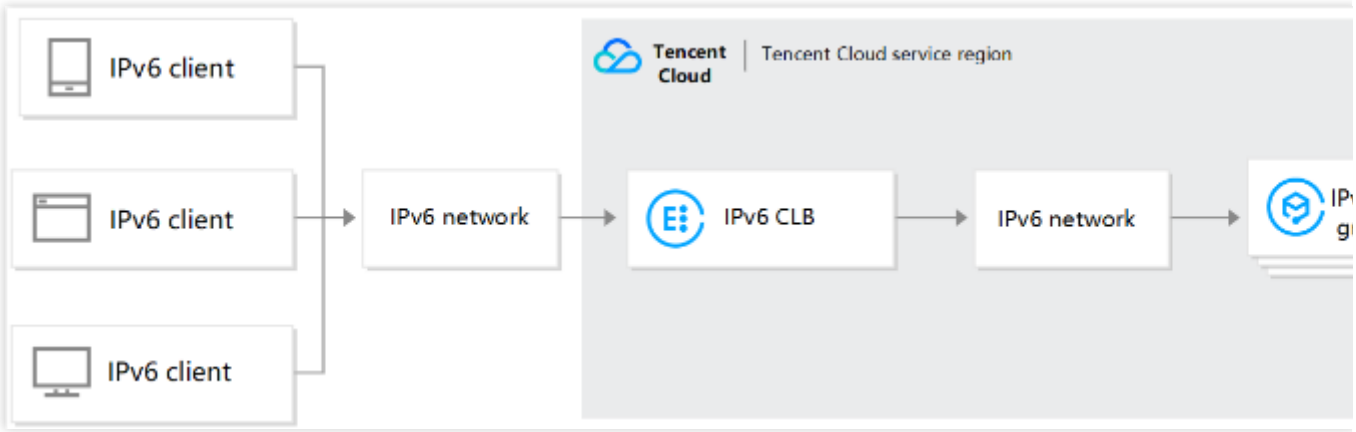
IPv6 CLBのメリット

Tencent Cloud IPv6 CLBは、業務のIPv6への高速アクセスを支援する上で、次のようなメリットを提供します。
クイックアクセス：IPv6に秒速で接続でき、購入後すぐに使用できます。
使いやすさ：IPv6 CLBは旧IPv4 CLBのトラフィック操作フローとの間に互換性があり、学習コストがかからないため、使用のハードルが低くなっています。
エンドツーエンドIPv6通信：IPv6 CLBとCVM間でIPv6を介した通信を行うことによって、CVM上にデプロイされたアプリケーションが速やかにIPv6変換を行い、エンドツーエンドのIPv6通信を実現します。

IPv6 CLBアーキテクチャ

CLBは、IPv6 CLB（以下、IPv6 CLBとも呼びます）インスタンスの作成をサポートしています。Tencent Cloudは、インスタンスにIPv6パブリックアドレス（すなわちIPv6バージョンのVIP）を割り当てます。このVIPは、IPv6クライアントからのリクエストをバックエンドのIPv6 CVMに転送します。

IPv6 CLBインスタンスは、IPv6パブリックネットワークユーザーにすばやくアクセスできるだけでなく、IPv6プロトコルを介してバックエンドCVMと通信することもできます。これによって、クラウド上のアプリケーションはIPv6をすばやく変換し、エンドツーエンドのIPv6通信を実現できます。
IPv6 CLBのアーキテクチャは、下図に示すとおりです。



ステップ1：IPv6 CLBインスタンスの作成

1. Tencent Cloud公式サイトにログインし、CLB購入ページに進みます。
2. 必要に応じて次のCLB関連設定を選択します。

標準アカウントタイプ

パラメータ	説明
課金モデル	従量課金モデルをサポートしています。IPv6バージョンのサポートは従量課金モデルのみとなります。その他の制限の状況については、IPバージョンをご参照ください。
リージョン	所属リージョンを選択します。CLBのサポートリージョンの詳細については、リージョンリストをご参照ください。
インスタンスタイプ	CLBインスタンスタイプのみをサポートしています。従来型CLBは2021年10月20日をもって販売終了となります。詳細については、従来型CLB販売終了のお知らせをご参照ください。
ネットワークタイプ	ネットワークタイプには、パブリックネットワークとプライベートネットワークという2種類があります。詳細については、ネットワークタイプをご参照ください。IPv6 CLBは、パブリックネットワークタイプを選択する必要があります。
Elastic IP	Elastic IPは選択しないでください。
IPバージョン	IPv6バージョンを選択します。
所属ネッ	所属するネットワークを選択する場合、取得済みのVirtual Private Cloud(VPC)とサブネットを

トワーク	選択してください。既存のネットワークが適切でない場合は、VPCの新規作成またはサブネットの新規作成を行うことができます。
キャリアタイプ	キャリアタイプはBGP（複数回線）です。
インスタンス仕様	共有タイプインスタンスとLCUタイプインスタンスをサポートしています。 共有タイプインスタンスでは複数のインスタンスがリソースを共有し、単一のインスタンスでは保証された性能指標を提供しません。デフォルトでは、すべてのインスタンスが共有タイプインスタンスとなります。 LCUタイプインスタンスはパフォーマンスを保証するもので、互いにリソースを奪い合う共有タイプインスタンスはなく、転送パフォーマンスは他のインスタンスの影響を受けません。単一のインスタンスで同時接続数最大100万、新規接続数10万、1秒あたりの照会数5万をサポートできます。
デュアルスタックハイブリッドバインド	有効化した場合、このCLBインスタンスのレイヤー7リスナーは、IPv4とIPv6のバックエンドサーバーをバインドできます。レイヤー4リスナーはハイブリッドバインドをサポートしていませんので、バインドできるのはIPv6のバックエンドサーバーのみとなります。
ネットワーク課金方式	ネットワーク課金モデルには、トラフィック課金、共有帯域幅パッケージがあります。
帯域幅の上限	1-2048Mbps。
プロジェクト	所属プロジェクトを選択してください。
タグ	タグキーとタグ値を選択するか、もしくはタグの新規作成を選択することもできます。詳細については、タグの作成をご参照ください。
インスタンス名	60文字まで入力でき、アルファベット、数字、中国語、「-」、「_」、「.」を使用できます。入力しない場合はデフォルトで自動生成されます。

従来型アカウントタイプ

パラメータ	説明
課金モデル	従量課金モデルのみサポートしています。
リージョ	所属リージョンを選択します。CLBのサポートリージョンの詳細については、リージョンリス

ン	トをご参照ください。
インスタンスタイプ	CLBインスタンスタイプのみをサポートしています。従来型CLBは2021年10月20日をもって販売終了となります。詳細については、従来型CLB販売終了のお知らせをご参照ください。
ネットワークタイプ	ネットワークタイプにはパブリックネットワークとプライベートネットワークという2種類があります。詳細については、ネットワークタイプをご参照ください。 パブリックネットワーク：CLBを使用してパブリックネットワークからのリクエストを振り分けます。 プライベートネットワーク：CLBを使用して、Tencent Cloudプライベートネットワークのリクエストを振り分けます。プライベートネットワークは以下のIPバージョン、キャリアタイプ、インスタンス仕様の設定をサポートしていないため、これらの設定項目はデフォルトでは表示されません。
IPバージョン	IPv6バージョンを選択します。使用制限の詳細については、IPバージョンをご参照ください。
所属ネットワーク	CLBがサポートする所属ネットワークはClassic networkとVPCです。 基幹ネットワークとはTencent Cloud上のすべてのユーザーのための公共のネットワークリソースプールです。すべてのCVMのプライベートIPアドレスはTencent Cloudが一元的に割り当てており、ネットワークセグメントの区分、IPアドレスをカスタマイズすることはできません。 VPCとは、ユーザーがTencent Cloud上に構築した、論理的に隔離されているネットワークスペースです。VPC内では、ユーザーはネットワークセグメントの区分、IPアドレスおよびルーティングポリシーを自由に定義できます。 両者を比べた場合、プライベートネットワークは基幹ネットワークよりもネットワークのカスタマイズが必要なシナリオに適しています。基幹ネットワークの全製品は、2022年12月31日にオフラインとなります。詳細については、基幹ネットワークオフラインのお知らせをご参照ください。VPCを選択されることをお勧めします。
キャリアタイプ	キャリアタイプはBGP（複数回線）です。
インスタンス仕様	共有タイプインスタンスとLCUタイプインスタンスをサポートしています。 共有タイプインスタンスでは複数のインスタンスがリソースを共有し、単一のインスタンスでは保証された性能指標を提供しません。デフォルトでは、すべてのインスタンスが共有タイプインスタンスとなります。 LCUタイプインスタンスはパフォーマンスを保証するもので、互いにリソースを奪い合う共有タイプインスタンスはなく、転送パフォーマンスは他のインスタンスの影響を受けません。単一のインスタンスで同時接続数最大100万、新規接続数10万、1秒あたりの照会数5万をサポートできます。
ネットワーク課金方式	ネットワークの課金モデルは共有帯域幅パッケージです。
帯域幅の	1-1024Mbps。

上限	
プロジェクト	所属プロジェクトを選択してください。
タグ	タグキーとタグ値を選択するか、もしくはタグの新規作成を選択することもできます。詳細については、タグの作成をご参照ください。
インスタンス名	60文字まで入力でき、アルファベット、数字、中国語、「-」、「_」、「.」を使用できます。入力しない場合はデフォルトで自動生成されます。

3. 購入ページで各項目の設定を選択し、**今すぐ購入**をクリックします。「CLBオーダーの確認」ポップアップウィンドウで、**オーダーの確認**をクリックします。CLBインスタンスリストページに戻ると、IPv6 CLBが購入済みになっていることを確認できます。

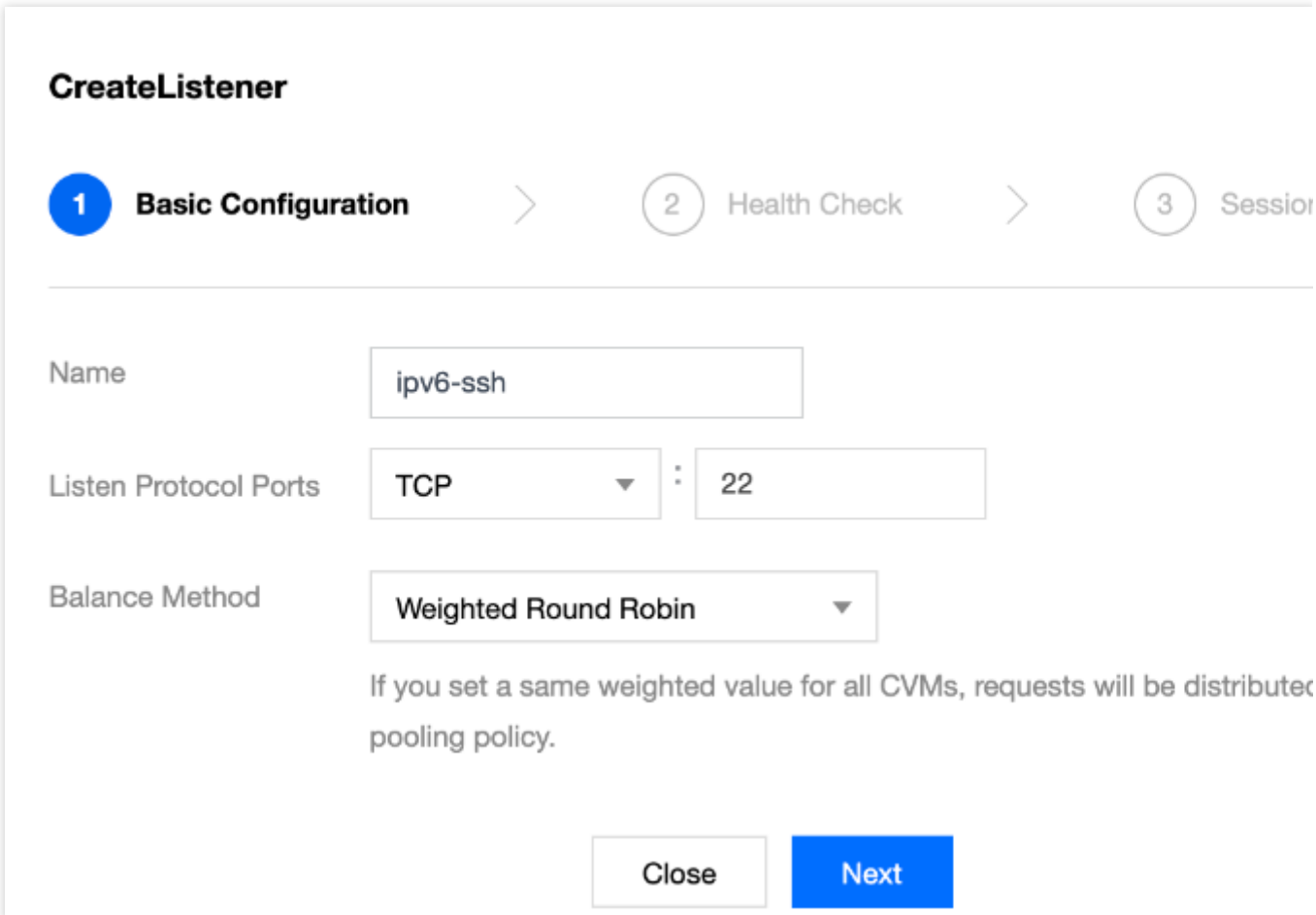
ステップ2：IPv6 CLBリスナーの作成

1. CLBコンソールにログインし、IPv6 CLBのインスタンスIDをクリックして、詳細ページに進みます。
2. **リスナー管理**タブを選択し、**新規作成**をクリックしてTCPリスナーの作成などを行います。

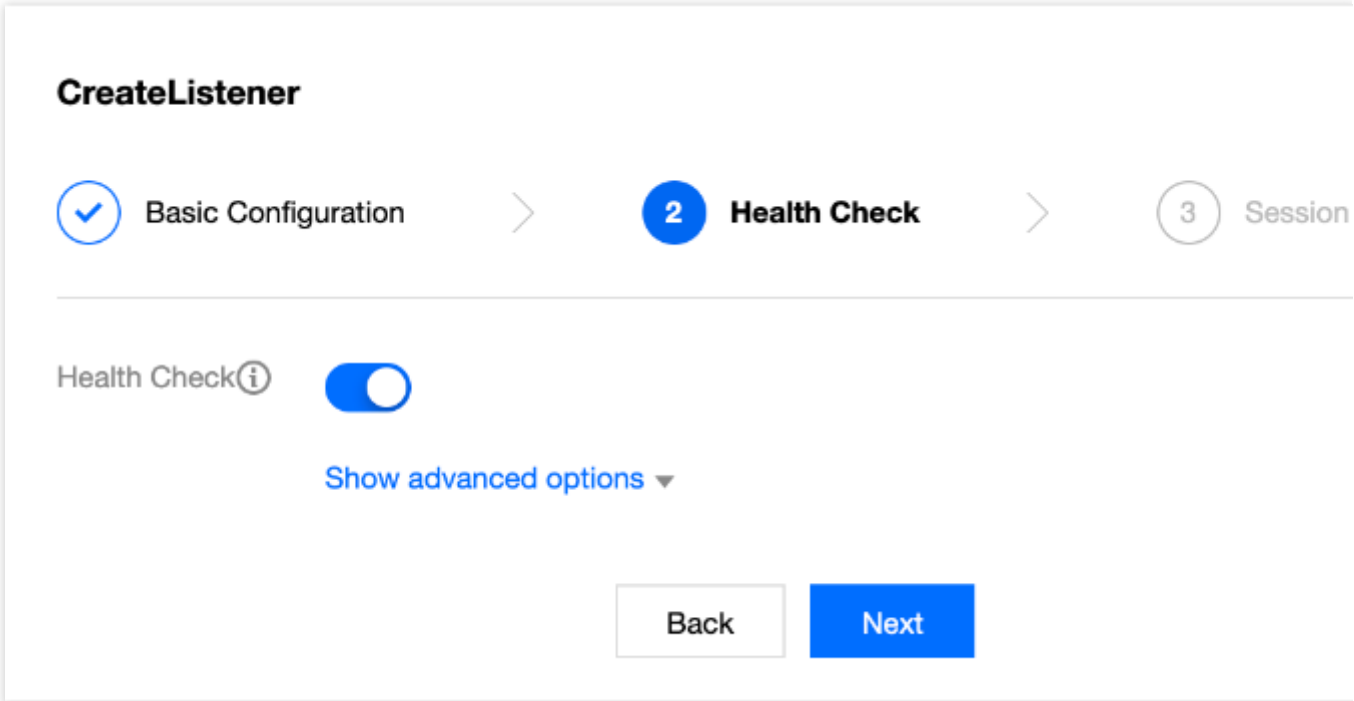
説明：

レイヤー4のIPv6 CLBリスナー(TCP/UDP/TCP SSL)とレイヤー7のIPv6 CLBリスナー(HTTP/HTTPS)の作成をサポートしています。詳細については、CLBリスナーの概要をご参照ください。

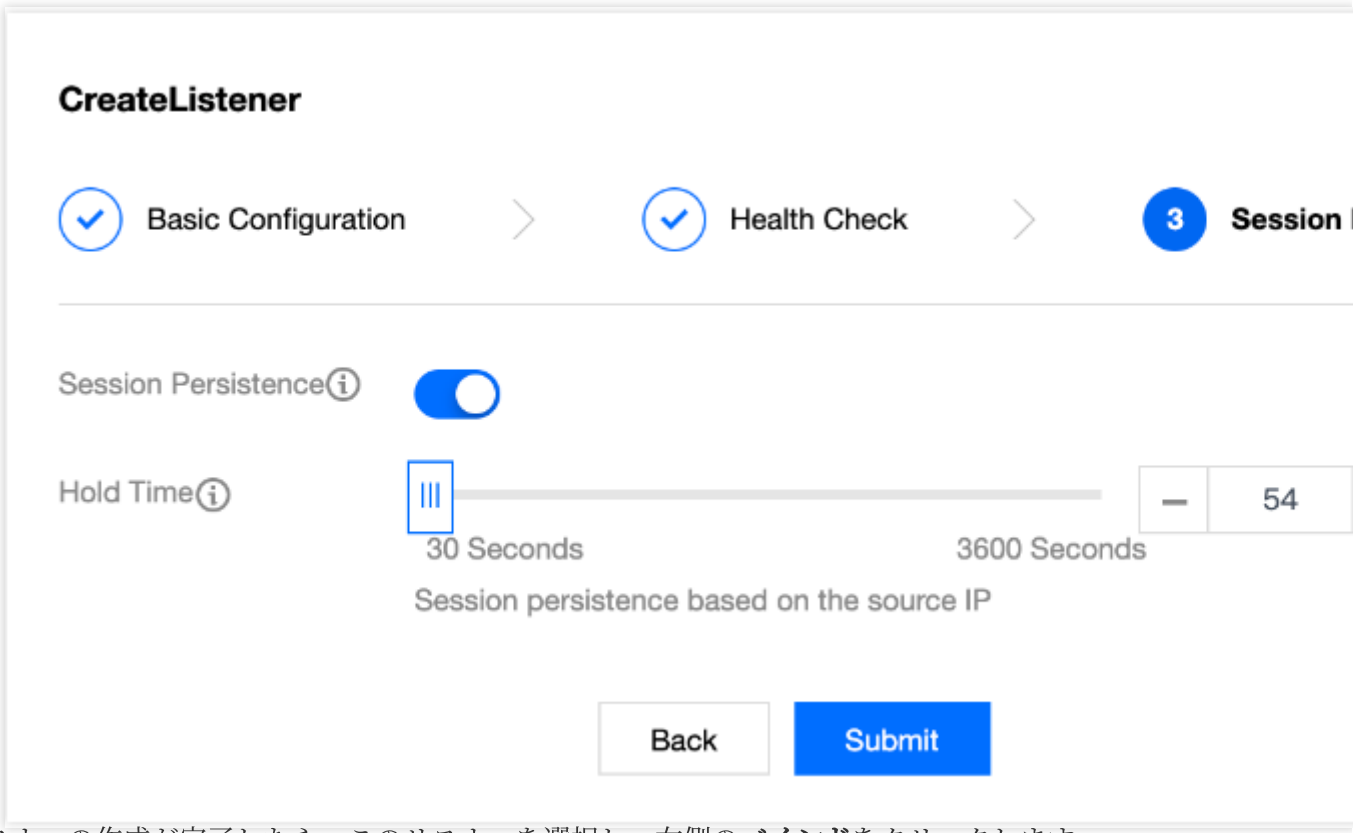
3. 「基本設定」で、名前、リスニングプロトコルポート、バランシング方式を設定し、**次へ**をクリックします。



4. ヘルスチェックを設定し、**次へ**をクリックします。



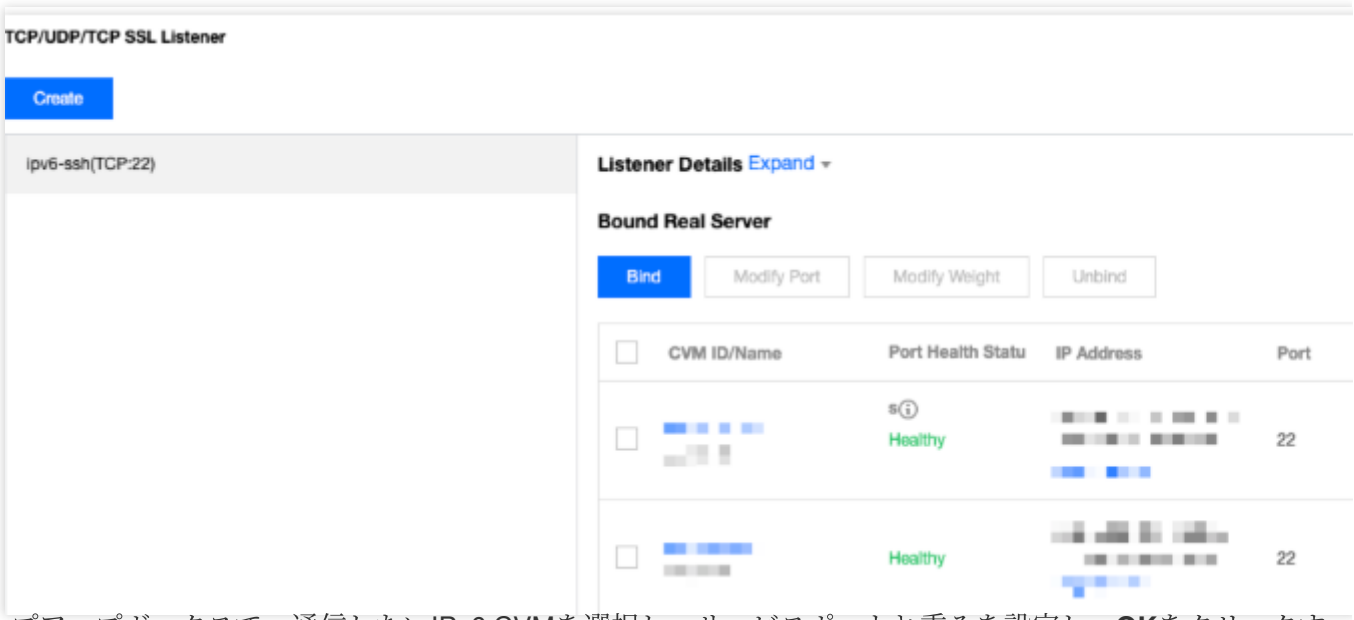
5. セッション維持を設定し、**送信**をクリックします。



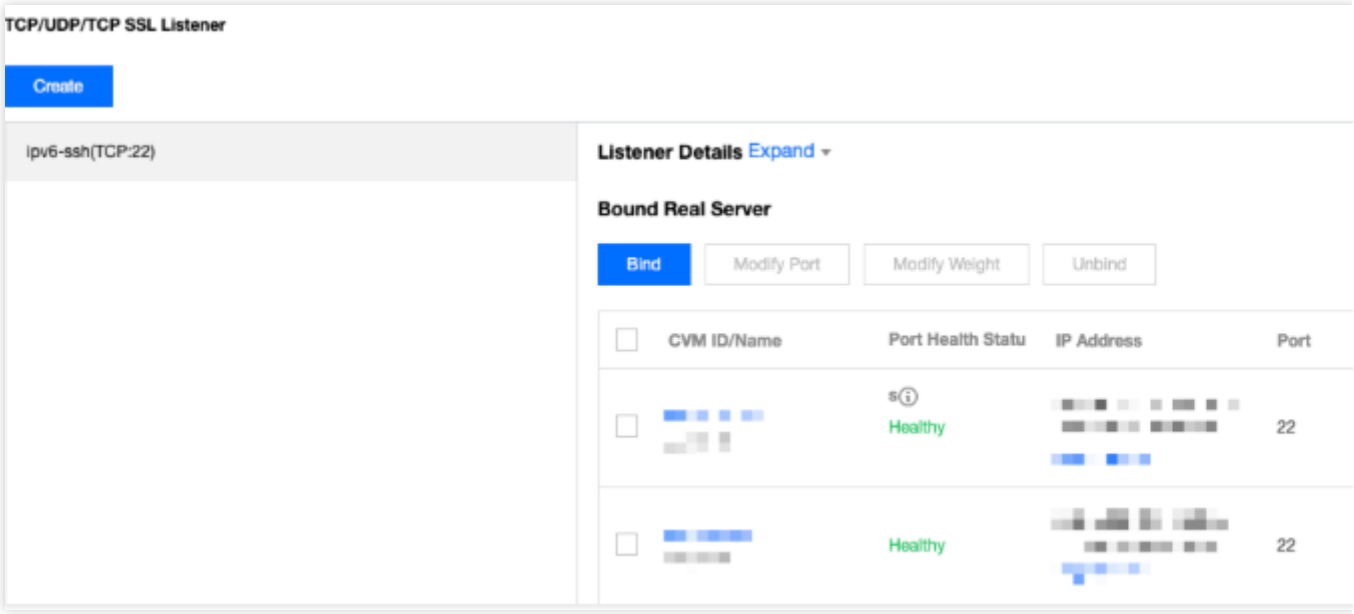
6. リスナーの作成が完了したら、このリスナーを選択し、右側の**バインド**をクリックします。

説明：

CVMをバインドする前に、このCVMがIPv6アドレスを取得していることを確認してください。



7. ポップアップボックスで、通信したいIPv6 CVMを選択し、サービスポートと重みを設定し、**OK**をクリックすれば完了です。



更なる操作

IPv6 CLBでのIPv6とIPv4バックエンドサービスのハイブリッドバインド

デュアルスタックハイブリッドバインドを有効化すると、IPv6 CLBのレイヤー7リスナーは、IPv6とIPv4のバックエンドCVMを同時にバインドでき、XFFからソースIPの取得をサポートします。IPv6 CLBのレイヤー4リスナーはハイブリッドバインドをサポートしておらず、バインドできるのはIPv6のバックエンドサーバーのみとなります。

1. デュアルスタックハイブリッドバインドを有効化します。
購入ページでIPv6 CLBを購入する際に、デュアルスタックハイブリッドバインドを有効化します。
IPv6 CLBインスタンス詳細ページで、デュアルスタックハイブリッドバインドを有効化します。
2. レイヤー7 HTTPまたはHTTPSリスナーを作成します。
3. IPv6またはIPv4タイプのバックエンドサービスのバインドを選択します。

IPv6 NAT64 CLBインスタンスの作成

最終更新日：：2024-01-04 17:48:23

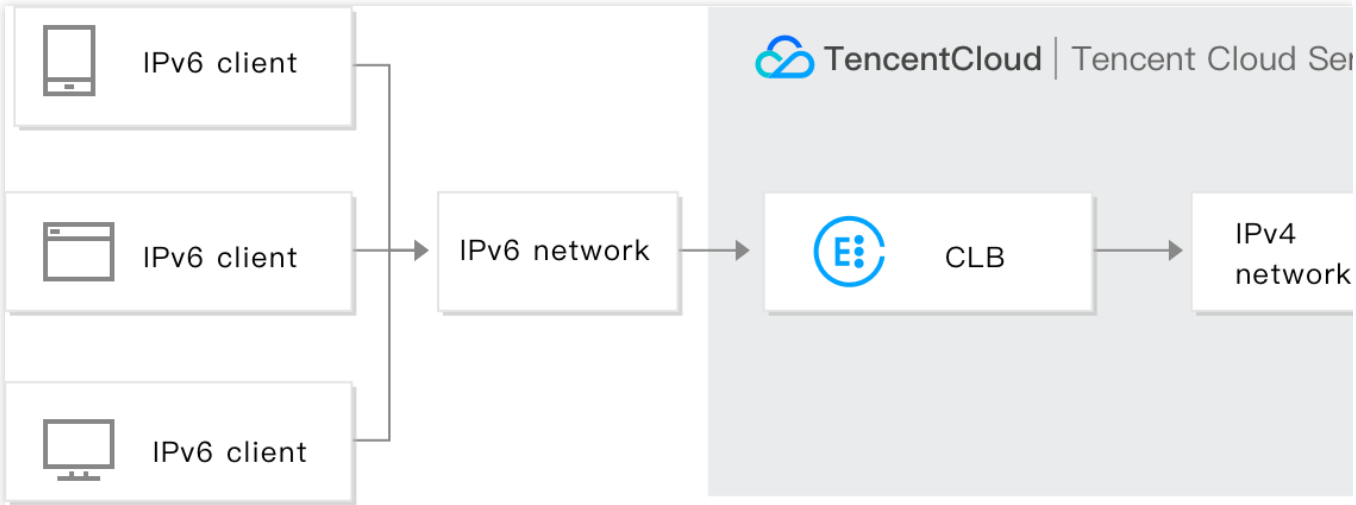
説明：
IPv6 NAT64 CLBは北京、上海、広州の3リージョンのみサポートしています。
IPv6 NAT64 CLBは、従来型CLBをサポートしていません。
インターネットのIPv6ネットワークマクロ環境は構築の初期段階のため、SLA保障を提供していません。ネットワークにアクセスできない状態が発生した場合は、チケットを提出して、フィードバックしてください。
CLBはIPv6 NAT64 CLBインスタンスの作成をサポートしています。Tencent CloudはインスタンスにIPv6パブリックアドレス（すなわち、IPv6版のVIP）を割り当てます。このVIPはIPv6クライアントからのリクエストをバックエンドのIPv4 CVMに転送します。

IPv6 NAT64 CLBとは何ですか

IPv6 NAT64 CLBはNAT64 IPv6移行技術をベースにして実現したロードバランサです。IPv6 NAT64 CLBによって、バックエンドCVMはIPv6用の修正を何も行うことなく、スピーディーにIPv6ユーザーからのアクセスに対応できます。

IPv6 NAT64 CLBのアーキテクチャ

IPv6 NAT64 CLBのアーキテクチャは、下図のとおりです。



IPv6ネットワークからIPv6 NAT64 CLBにアクセスする場合、CLBはIPv6アドレスをIPv4アドレスにスムーズに変換し、既存のサービスに適用させることで、IPv6の修正をスピーディーに実現します。

IPv6 NAT64 CLBのメリット

Tencent Cloud IPv6 NAT64 CLBは業務をスピーディーにIPv6に接続させる際に、次のようなメリットを発揮します。

クイックアクセス：秒レベルでIPv6に接続でき、購入後すぐに使用できます。

業務のスムーズな移行：業務上修正が必要なのはクライアントのみで、バックエンドサービスの修正は必要なく、スムーズにIPv6に接続できます。IPv6 NAT64 CLBはIPv6クライアントからのアクセスをサポートし、IPv6メッセージのIPv4メッセージへの変換も行います。バックエンドCVM上のアプリケーションはIPv6であることを感知せず、従来の形式でデプロイを行うことができます。

使いやすさ：IPv6 NAT64 CLBは旧IPv4 CLBの操作フローとの間に互換性があり、学習コストがかからないため、使用のハードルが低くなっています。

操作ガイド

IPv6 NAT64 CLBの作成

1. Tencent Cloud公式サイトにログインし、CLB購入ページに進みます。
2. 次のパラメータを正しく選択してください。
課金モデル：従量課金モデルをサポートしています。
リージョン：北京、上海、広州の3リージョンのみサポートしています。
インスタンスタイプ：CLBです。
ネットワークタイプ：パブリックネットワークです。
IPバージョン：IPv6 NAT64です。
所属ネットワーク：VPCです。
その他の設定は一般的なインスタンスの設定と同様です。
3. 購入ページで各項目の設定を選択し、今すぐ購入をクリックします。CLBインスタンスリストページに戻ると、IPv6 NAT64が購入済みになっていることを確認できます。

IPv6 NAT64 CLBの使用

CLBコンソールにログインし、インスタンスIDをクリックして詳細ページに進み、「リスナー管理」ページで、リスナー、転送ルール、CVMのバインドを設定します。詳細については、 CLBクイックスタートをご参照ください。

関連ドキュメント

ハイブリッドクラウドのデプロイシーンでのTOAによるクライアントリアルIPの取得

CLBセキュリティグループの設定

最終更新日：：2024-01-04 17:41:37

Cloud Load Balancer（CLB）を作成すると、CLBのセキュリティグループを設定してパブリックネットワークのトラフィックを分離することができるようになります。ここではさまざまな方式のCLBセキュリティグループの設定方法についてご説明します。

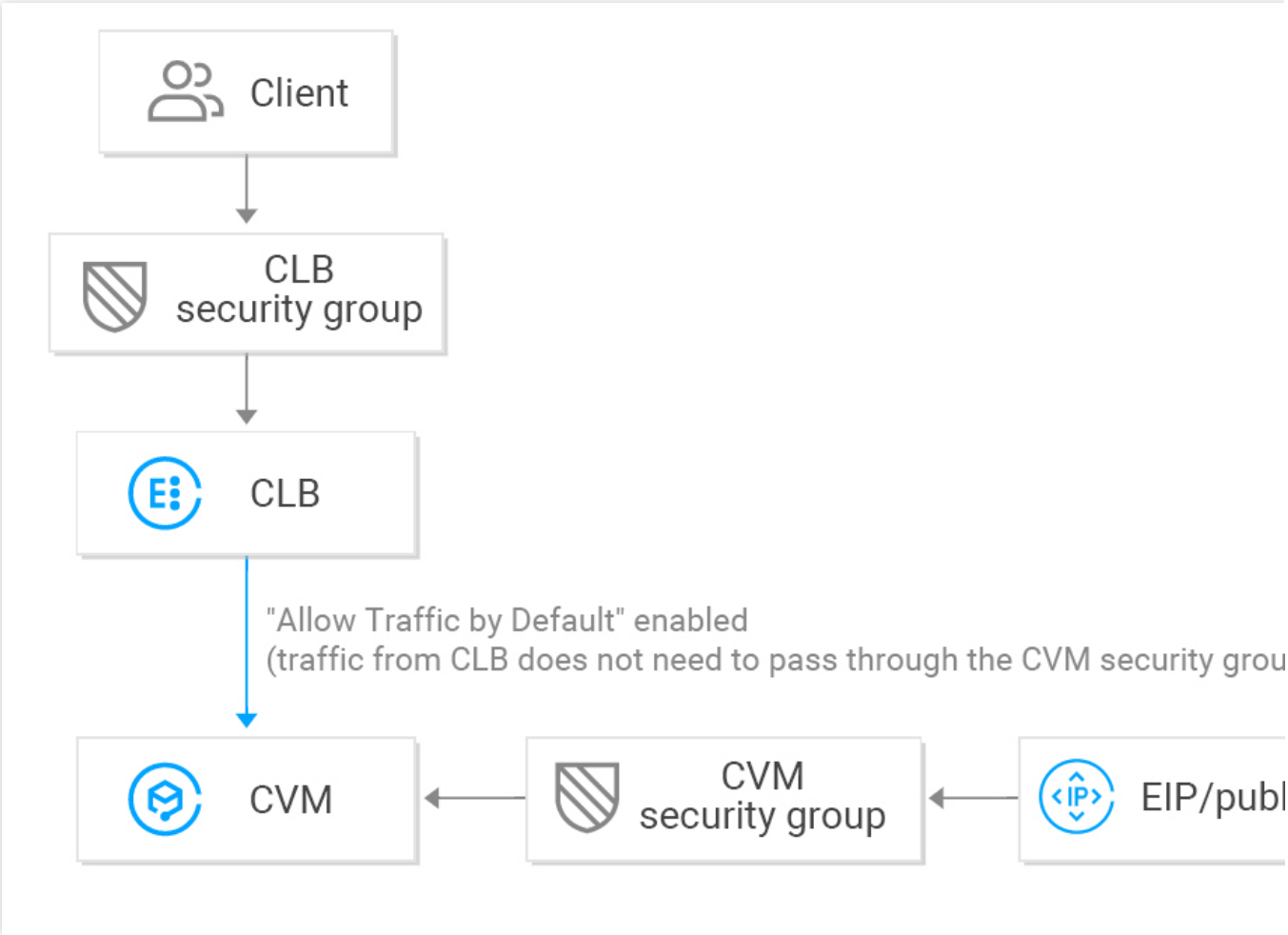
使用制限

各CLBにつき、最大5つまでのセキュリティグループをバインドできます。
CLBの各セキュリティグループのルール数は最大512個です。
基幹ネットワークのプライベートネットワークCLBはセキュリティグループのバインドをサポートしていません。プライベートネットワークCLBにAnycast EIPをバインドした場合、プライベートネットワークCLBにバインドしたセキュリティグループは有効になりません。
基幹ネットワークのCLBは、セキュリティグループのデフォルト許可機能をサポートしていません。

背景情報

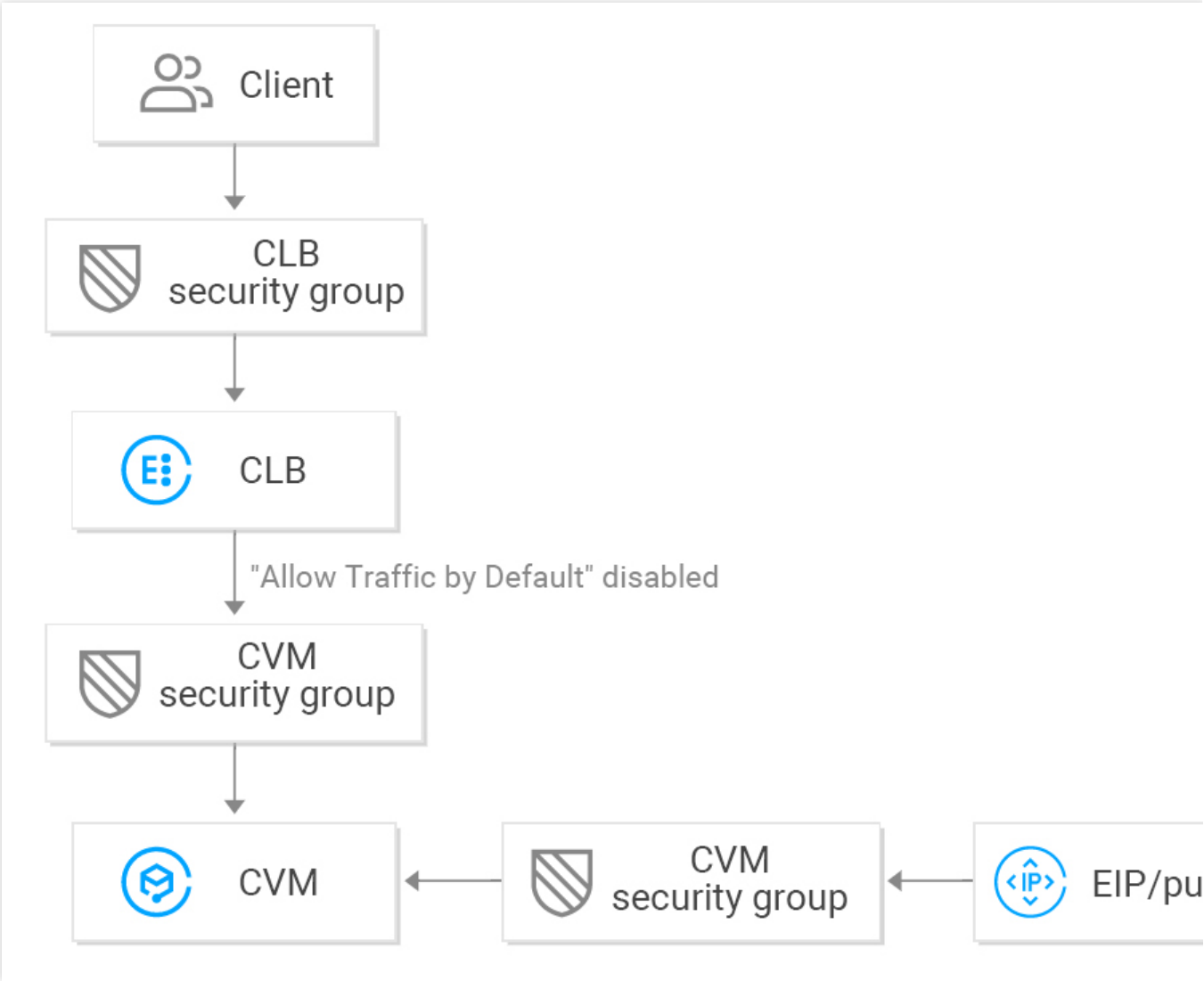
セキュリティグループとは一種の仮想ファイアウォールであり、ステートフルなデータパケットフィルタリング機能を有し、インスタンスレベルでのアウトバウンドおよびインバウンドトラフィックを制御します。詳細については、セキュリティグループの概要をご参照ください。
CLBセキュリティグループはCLBインスタンスにバインドするセキュリティグループであり、CVMセキュリティグループはCVMにバインドするセキュリティグループです。両者は制限の対象が異なります。CLBのセキュリティグループの設定には主に次の2種類の方式があります。
セキュリティグループのデフォルト許可を有効にする
セキュリティグループのデフォルト許可を無効にする
説明：
デフォルトの状態では、IPv4 CLB、NAT64セキュリティグループのデフォルト許可は無効になっています。コンソールで有効化/無効化を行うことができます。
デフォルトの状態では、IPv6 CLBセキュリティグループのデフォルト許可は有効になっており、無効化はできません。

セキュリティグループのデフォルト許可を有効にする



セキュリティグループのデフォルト許可を有効にすると、次のようになります。
固定のClient IPからのアクセスを指定したい場合、CLBセキュリティグループはClient IPとリスニングポートを許可する必要があり、バックエンドCVMのセキュリティグループはClient IPとサービスポートを許可する必要はありません。CLBからのアクセストラフィックはCLBのセキュリティグループのみを通過させればよく、バックエンドCVMはCLBからのトラフィックをデフォルトで許可します。バックエンドCVMはポートを外部に公開する必要はありません。
パブリックIP（一般的なパブリックIPとEIPを含む）からのトラフィックは、CVMのセキュリティグループを通過する必要があります。
CLBインスタンスにセキュリティグループを設定しない場合は、すべてのトラフィックが許可されます。CLBインスタンスのVIP上では、リスナーを設定したポートのみがアクセス可能なため、リスニングポートはすべてのIPのトラフィックを許可します。
あるClient IPからのトラフィックを拒否したい場合は、CLBのセキュリティグループでアクセスを拒否する必要があります。あるIPからのアクセスをCVMのセキュリティグループで拒否しても、CLBからのトラフィックに対しては有効にならず、パブリックIP（一般的なパブリックIPとEIPを含む）からのトラフィックに対してのみ有効になります。

セキュリティグループのデフォルト許可を無効にする



セキュリティグループのデフォルト許可を無効にすると、次のようになります。
固定のClient IPからのアクセスを指定したい場合、CLBセキュリティグループはClient IPとリスニングポートを許可する必要があり、バックエンドCVMのセキュリティグループもClient IPとサービスポートを許可する必要があります。すなわち、CLBを通過する業務トラフィックはCLBセキュリティグループとCVMセキュリティグループによる二重のチェックを受けることになります。
パブリックIP（一般的なパブリックIPとEIPを含む）からのトラフィックは、CVMのセキュリティグループを通過する必要があります。
CLBインスタンスにセキュリティグループを設定しない場合は、CVMセキュリティグループを通過したトラフィックのみを許可します。
あるClient IPからのトラフィックを拒否したい場合は、CLBかCVMのいずれかのセキュリティグループでアクセスを拒否することができます。
セキュリティグループのデフォルト許可を無効にしている場合は、ヘルスチェック機能を保障するため、CVMセキュリティグループに次の設定を行う必要があります。

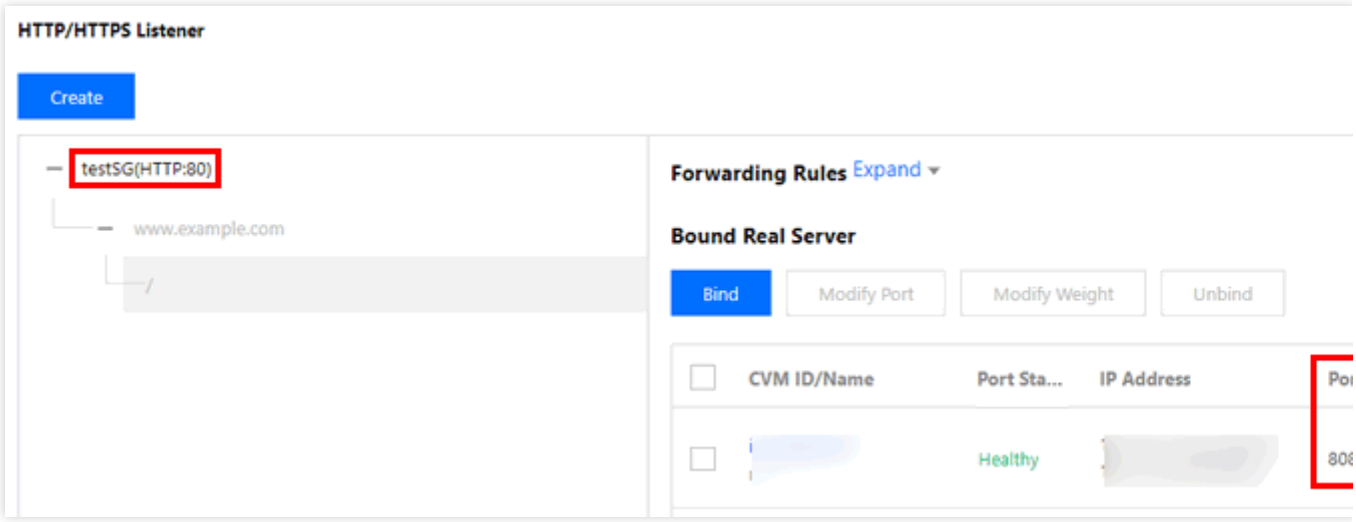
1. パブリックネットワークCLBの設定
バックエンドCVMのセキュリティグループでCLBのVIPを許可する必要がある場合、CLBはVIPを使用してバックエンドCVMのヘルスステータスをチェックします。
2. プライベートネットワークCLBの設定
プライベートネットワークCLB（旧「アプリケーション型プライベートネットワークCLB」）については、CLBがVPCネットワークにある場合、 バックエンドCVMのセキュリティグループ上でCLBのVIP（ヘルスチェック用）を開放する必要があります。CLBが基幹ネットワークにある場合は、バックエンドCVMのセキュリティグループ上で設定を行う必要はなく、ヘルスチェックIPがデフォルトで開放されています。

操作手順

パブリックネットワークCLBのセキュリティグループ設定の例を次に示します。CLBではあらかじめ80番ポートからのインバウンド業務トラフィックのみを許可し、CVMの8080番ポートからサービスを提供するようにし、なおかつClient IPは制限せず、任意のIPからのアクセスをサポートしています。
ご注意：
この例ではパブリックネットワークCLBを使用しており、バックエンドCVMのセキュリティグループでCLBのVIPを許可してヘルスチェックを行う必要があります。現在の `0.0.0.0/0` は任意のIPを意味し、CLBのVIPも含まれます。

ステップ1：CLBおよびリスナーの作成とCVMのバインド

詳細については、CLBクイックスタートをご参照ください。今回はHTTP:80リスナーを作成し、バックエンドCVMにバインドします。バックエンドCVMのサービスポートは8080です。



ステップ2：CLBセキュリティグループの設定

1. CLBセキュリティグループルールの設定 セキュリティグループコンソール 上でセキュリティグループルールを設定します。インバウンドルールですべてのIP（すなわち `0.0.0.0/0` ）の80番ポートを許可し、その他のポートからのトラフィックを拒否します。

説明：

セキュリティグループルールは上から下の順にフィルタリングされて有効になるため、以前に設定した許可ルールが適用されると、その他のルールはデフォルトで拒否されますので、設定の順序に注意してください。詳細については、セキュリティグループルールの説明 をご参照ください。

セキュリティグループにはインバウンドルールとアウトバウンドルールがあり、上記の設定によって制限されるのはインバウンドトラフィックです。このため、設定はすべて**インバウンドルール**の設定であり、アウトバウンドルールは特に設定する必要はありません。

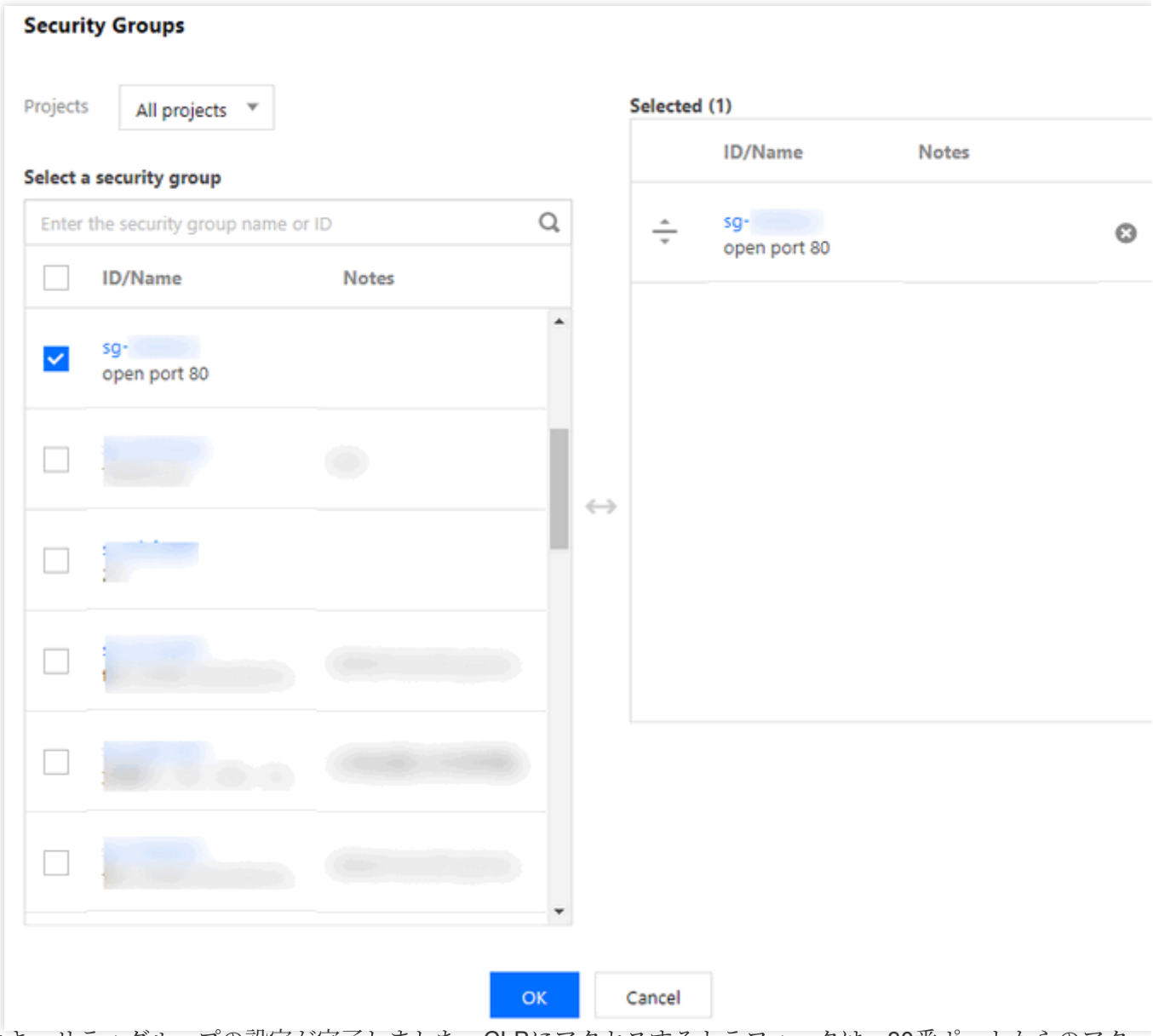
2. セキュリティグループのCLBへのバインド

2.1 CLBコンソールにログインします。

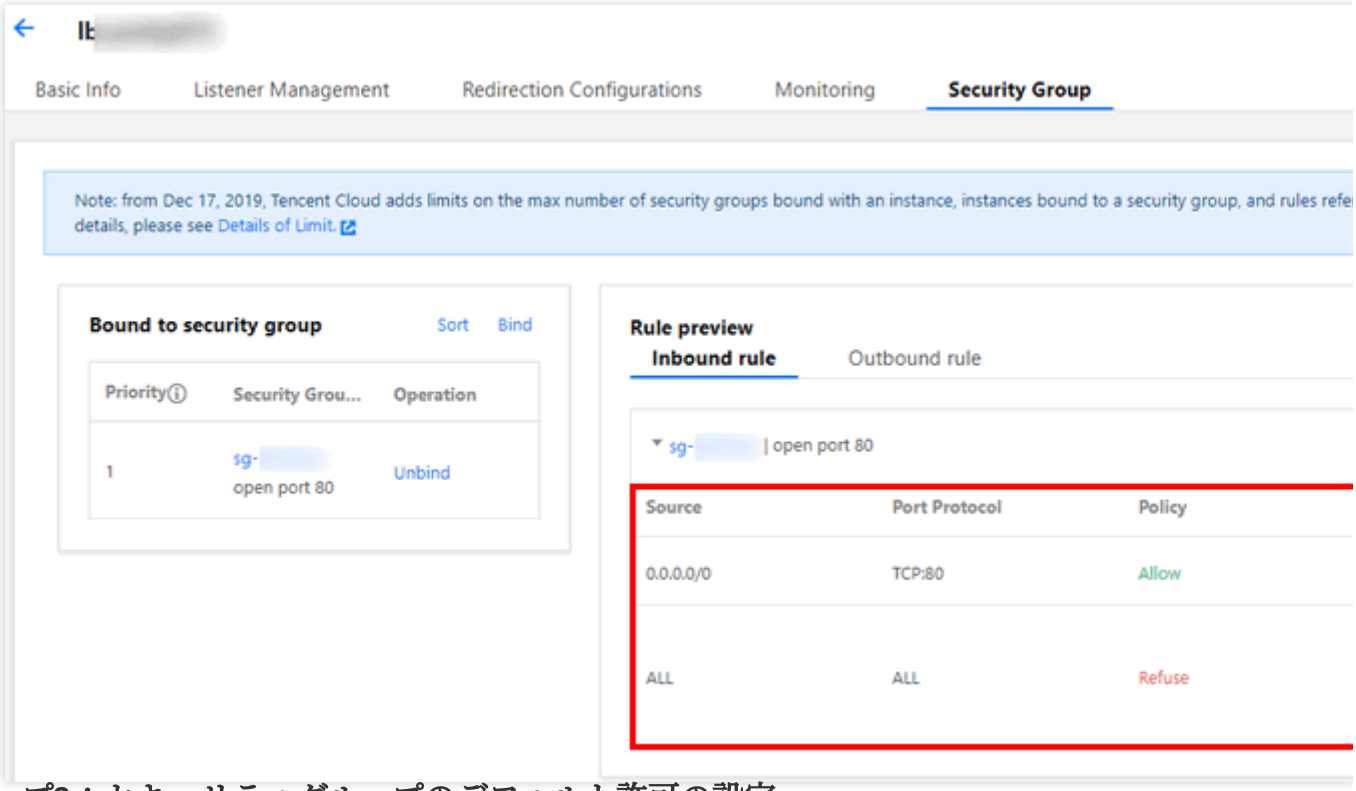
2.2 「インスタンス管理」ページで目的のCLBインスタンスを見つけ、インスタンスIDをクリックします。

2.3 インスタンス詳細ページで、【セキュリティグループ】タブをクリックし、「バインド済みのセキュリティグループ」モジュールで【バインド】をクリックします。

2.4 ポップアップした「セキュリティグループの設定」ウィンドウで、CLB上にバインドするセキュリティグループを選択し、【OK】をクリックします。



CLBセキュリティグループの設定が完了しました。CLBにアクセスするトラフィックは、80番ポートからのアクセスのみ許可されます。



ステップ3：セキュリティグループのデフォルト許可の設定

セキュリティグループのデフォルト許可は有効か無効かを選択することができます。それぞれを選択した場合の設定は次のようになります。

方法1：セキュリティグループのデフォルト許可を有効にした場合、バックエンドCVMはポートを外部に公開する必要はありません。

説明：

基幹ネットワークのCLBは、セキュリティグループのデフォルト許可機能をサポートしていません。

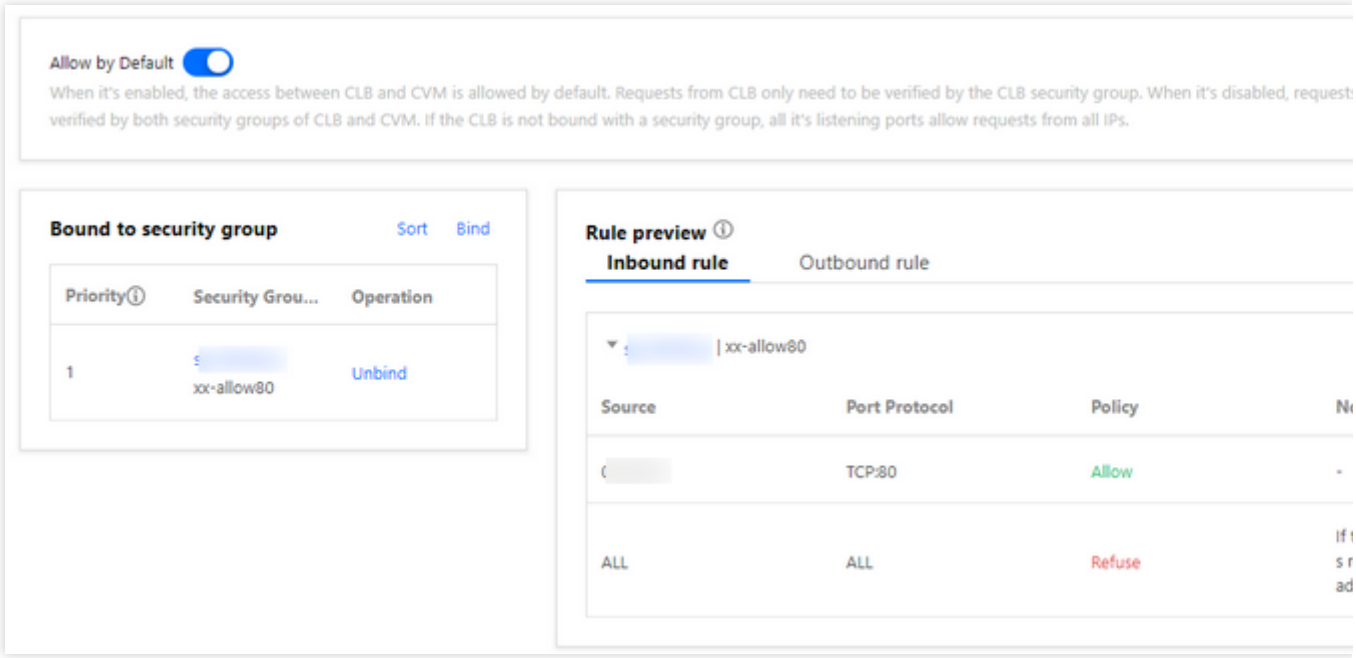
方法2：セキュリティグループのデフォルト許可を無効にした場合は、CVMのセキュリティグループでもClient IPを開放する必要があります（この例では0.0.0.0/0）。

方法1：セキュリティグループのデフォルト許可を有効にする

1. CLBコンソールにログインします。
2. 「インスタンス管理」ページで目的のCLBインスタンスを見つけ、インスタンスIDをクリックします。
3. インスタンス詳細ページで【セキュリティグループ】タブをクリックします。
4. 「セキュリティグループ」ページで

をクリックし、デフォルト許可を有効化します。

5. デフォルト許可機能を有効化すると、次の**ルールプレビュー**にあるセキュリティグループルールのみを検証します。



方法2：セキュリティグループのデフォルト許可を無効にする

デフォルト許可を無効化する場合は、CVMのセキュリティグループ上でもClient IPを許可する必要があります。CLBを介してCVMにアクセスする業務トラフィックは、CLBの80番ポートからのインバウンドのみを許可し、サービスはCVMの8080番ポートから提供されます。

説明：

あるClient IPからのトラフィックを許可するには、CLBとCVMの両方のセキュリティグループで許可する必要があります。CLBにセキュリティグループを設定していない場合は、CVM上のセキュリティグループの許可だけが必要です。

1. CVMセキュリティグループルールの設定

バックエンドCVMへのアクセストラフィックについて、CVMセキュリティグループを設定することで、サービスポートからのアクセスのみを許可するよう制限します。

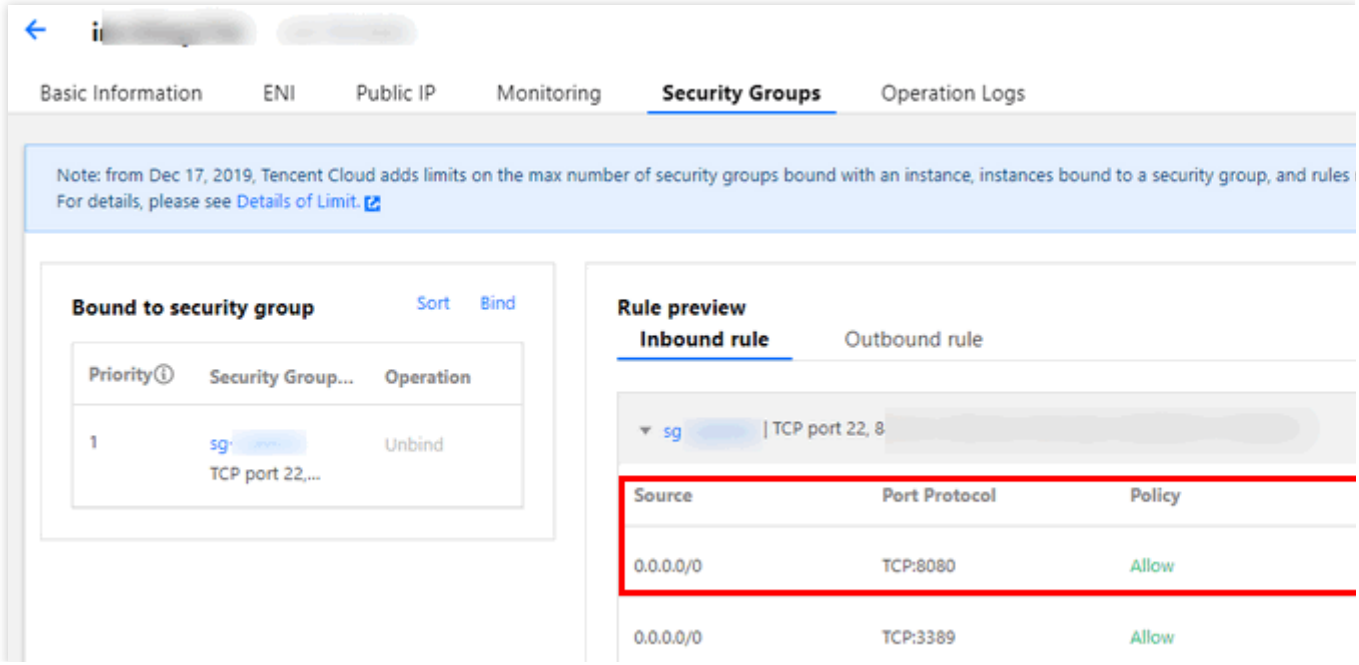
セキュリティグループコンソール上でセキュリティグループポリシーを設定し、インバウンドルールですべてのIPの8080番ポートを許可します。リモートログインホストおよびPingサービスを保障するため、セキュリティグループでは22、3389およびICMPサービスを許可する必要があります。

2. セキュリティグループのCVMへのバインド

2.1 CVMコンソールで、CLBにバインドされたCVMのIDをクリックし、詳細ページに進みます。

2.2 【セキュリティグループ】タブを選択し、「バインド済みのセキュリティグループ」モジュールで【バインド】をクリックします。

2.3 ポップアップした「セキュリティグループの設定」ウィンドウで、CVM上にバインドするセキュリティグループを選択し、【OK】をクリックします。



プライベートネットワークCLBインスタンスのEIPバインド

最終更新日：：2024-01-04 17:48:23

プライベートネットワークCLBは、Tencent Cloudプライベートネットワークのリクエストを配信するのに用います。パブリックIPがない場合は、パブリックネットワークとの相互接続ができません。プライベートネットワークCLBを利用してパブリックネットワークと相互接続をする場合は、プライベートネットワークCLBのEIPバインドを選択し、EIPからパブリックネットワークにアクセスします。
説明：
プライベートネットワークCLBのEIPバインド機能はベータ版テスト段階です。利用される場合はチケットを提出してください。

使用制限

リージョン制限
済南、福州、石家庄、武漢、長沙リージョンにはプライベートネットワークCLBがないので、この機能をサポートしていません。
製品属性の制限
標準アカウントタイプのみをサポートし、従来型アカウントタイプはサポートしていません。
CLBインスタンスタイプのみをサポートし、従来型CLBはサポートしていません。
VPCのプライベートネットワークCLBのみをサポートし、基幹ネットワークのプライベートネットワークCLBはサポートしていません。
機能制限
現在プライベートネットワークCLBはポート側をサポートしていません。
プライベートネットワークCLBは、同一リージョンで、かつその他のリソースにバインドされていないEIPのみバインドが可能です。
各プライベートネットワークCLBは、それぞれ1つのEIPとのみ相互バインドが可能です。
プライベートネットワークCLBはEIPをバインド後、その機能はパブリックネットワークCLBに類似したものとなります。ただしパブリックネットワークCLBでは、プライベートネットワークCLBとEIPに分けることはできません。
セキュリティグループの制限
プライベートネットワークCLBはEIPをバインド後、CLBのセキュリティグループによってEIPからのトラフィックを無効化し、プライベートネットワークCLBからのトラフィックを有効化します。
プライベートネットワークCLBがEIPをバインドし、セキュリティグループのデフォルト許可を有効化した後に、バックエンドCVMのセキュリティグループがEIPとプライベートネットワークCLBからのトラフィックをデフォル

ト許可します。つまり、バックエンドCVMのセキュリティグループは2者のトラフィックをどちらも無効化します。このタイプのシーンでは、セキュリティグループのデフォルト許可を無効化しておくことをお勧めします。

操作手順

方法1：CLB購入時には、EIPを選択

1. し、Tencent CloudのCLB購入ページにログインします。
2. 必要に応じて、次のCLB関連の設定を選択します。残りの設定の詳しい内容については、購入方法をご参照ください。

パラメータ	説明
課金モデル	「従量課金」モードを選択します。
リージョン	所属リージョンを選択します。CLBのサポートリージョンの詳細については、リージョンリストをご参照ください。
インスタンスタイプ	CLBインスタンスタイプのみサポートしています。
ネットワークタイプ	ネットワークタイプは、「パブリックネットワーク」を選択します。
Elastic IP	EIPを選択すると、Tencent CloudはEIPとプライベートネットワークCLBをそれぞれ1つずつユーザーにアサインします。EIPがサポートするタイプは、標準IP、アクセラレーションIP、静的単一IPです。

方法2：プライベートネットワークCLBでEIPをバインド

1. し、CLBコンソールにログインの上、左側ナビゲーションバーのインスタンス管理をクリックします。
2. 「インスタンス管理」ページの左上隅でリージョンを選択し、インスタンスリストからターゲットのプライベートネットワークCLBインスタンスを選択します。右側の「操作」列で**その他 > EIPをバインド**を選びます。
3. ポップアップした「EIPのバインド」ダイアログボックスで、バインドしたいEIPを選択し、クリックで送信するとプライベートネットワークCLBがEIPをバインドします。

説明：

アクセラレーションIPと静的単一IPは、現在ベータ版テスト段階です。利用される場合はチケットを提出してください。

4. （オプション）インスタンスリストからターゲットのプライベートネットワークCLBインスタンスを選択します。右側の「操作」列で、その他 > EIPバインド解除を選択すると、プライベートネットワークCLBが解除されます。

関連ドキュメント

EIP APIドキュメントのバインド
購入方法
製品属性の選択

CLBインスタンスの起動と停止

最終更新日：：2024-01-04 17:48:23

インスタンスは起動または停止することができます。インスタンスを停止すると、それ以降はトラフィックの受信と転送は行われず、ヘルスチェックも行われません。また、Pingは無効になります。

説明：

この機能はベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。

ユースケース

大量のCLBインスタンスを設定していて、いくつかのインスタンスは業務上現時点では使用しないが、削除することもできない場合は、インスタンスの停止を選択することができます。

インスタンスを停止すると、リスナーもすべて停止し、インスタンスはそれ以降トラフィックの受信と転送を行いません。

インスタンスを起動すると、リスナーもすべて起動し、インスタンスはトラフィックを正常に受信および転送します。

リスナーを停止すると、リスナーはそれ以降トラフィックの受信と転送を行いません。すべてのリスナーを停止すると、インスタンス全体が停止します。

リスナーを起動すると、リスナーはトラフィックを正常に受信および転送します。すべてのリスナーを起動すると、インスタンス全体が起動します。

インスタンスの停止後、任意のリスナーを起動すると、インスタンスは起動状態に切り替わります。それ以外のリスナーは停止状態を維持しますが、インスタンスと起動しているリスナーはトラフィックを正常に受信および転送します。

制限事項

従来型CLBタイプはサポートしていません。

VPCネットワークのみサポートし、基幹ネットワークではサポートしていません。

TLS 1.3以下のプロトコルバージョンはサポートしていません。

前提条件

CLBインスタンスの作成を完了していること。

リスナーの作成を完了していること。

操作手順

1. CLBコンソールにログインします。
2. **インスタンス管理**ページの左上隅でリージョンを選択し、インスタンスリストで目的のインスタンスを見つけ、右側操作バーの**その他 > 起動**または**その他 > 停止**をクリックします。
3. （オプション）**リスナー管理**タブで目的のリスナーを見つけ、**リスナーの起動**または**リスナーの停止**をクリックします。

CLBクローンインスタンス

最終更新日：：2024-01-04 17:48:23

CLBはクローンインスタンス機能を提供します。ワンクリックで迅速にCLBを含むインスタンス属性、リスナー、セキュリティグループおよびログなどの既存のインスタンスの設定をコピーすることができます。

説明：

現在クローンインスタンス機能はベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。

制限事項

インスタンス属性のディメンション制限

クローン従量課金インスタンスのみサポートしています。

クローンはインスタンス課金項目が関連付けられていないCLBはサポートしていません。

従来型CLBインスタンスおよび高セキュリティCLBのクローン作成はサポートしていません。

基幹ネットワークタイプのクローンインスタンスはサポートしていません。

IPv6、IPv6 NAT64バージョンおよびミックスバインドのインスタンスはサポートしていません。

カスタム設定、リダイレクト設定、セキュリティグループデフォルト許可の有効化/無効化の設定はクローンされないため、再設定が必要です。

クローン操作を行う前に、インスタンス上に期限切れの証明書を使用していないことを確認してください。そうしない場合、クローンが失敗します。

リスナーディメンション制限

QUICタイプおよびポートセグメントのクローンリスナーのインスタンスはサポートしていません。

リスナーがTCP_SSLのプライベートネットワーク型CLBのインスタンスはサポートしていません。

レイヤー7リスナーのクローンは転送ルールがないインスタンスをサポートしていません。

バックエンドサービスのディメンション制限

バインドするバックエンドサービスタイプが目標グループおよびServerless Cloud Function（SCF）のクローンインスタンスはサポートしていません。

コンソールを介したクローンインスタンス

1. CLBコンソールにログインし、左側ナビゲーションバーの**インスタンス管理**をクリックします。
2. 「インスタンス管理」ページの左上隅でリージョンを選択し、インスタンスリストでクローン待ちのインスタンスを見つけ、右側の**操作**列の**その他 > クローン**をクリックします。

3. ポップアップした**クローンCLB**ダイアログボックスで、クローンインスタンスの名称を入力し、**OK**をクリックします。

APIによるクローンインスタンス

APIインターフェースのCLBクローンインスタンスをご参照ください。

CLBインスタンスのエクスポート

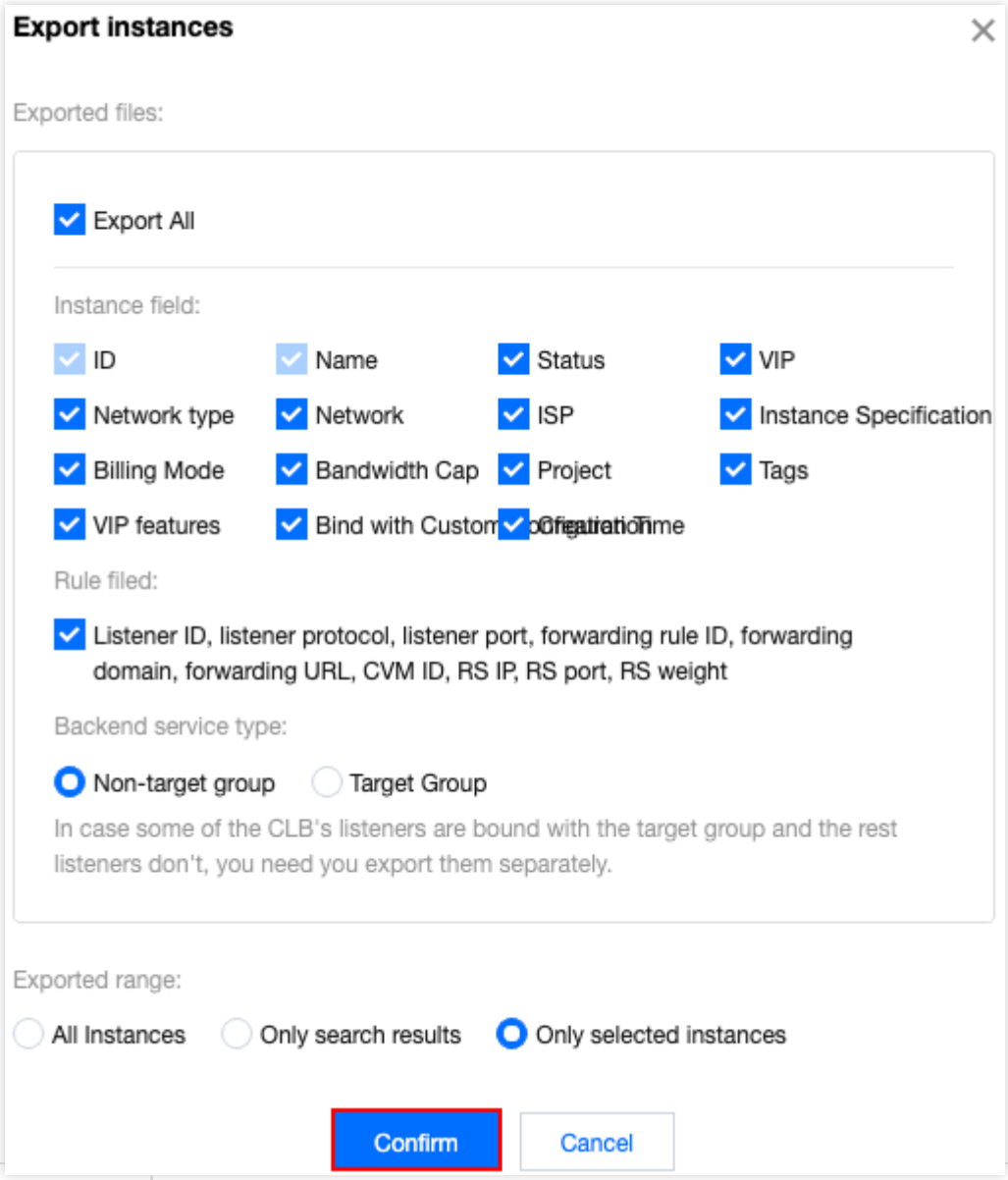
最終更新日：：2024-01-04 17:48:23

あるリージョンのCLBインスタンスリストをコンソールにエクスポートすることができます。またエクスポートするフィールドをカスタマイズし、インスタンスリソースの設定および使用状況の分析に役立てることもできます。

操作手順

1. CLBコンソールにログインし、「インスタンス管理」ページの左上隅で所在リージョンを選択します。
2. インスタンスリストで目的のインスタンスにチェックを入れ、右上隅の

アイコンをクリックします。
3. ポップアップした「インスタンスのエクスポート」ダイアログボックスで、エクスポートするフィールドおよびエクスポート範囲を選択し、**確定**をクリックしてインスタンスリストをローカルにダウンロードします。



パラメータ	説明
エクスポートフィールド	エクスポート可能なフィールドには次が含まれます。 インスタンスフィールド ルールフィールド このうち、ルールフィールドの「RSヘルスステータス」は、エクスポート範囲が「選択したインスタンスのみ」であり、かつルールフィールドにチェックを入れている場合のみステータスを見ることができ、そうでない場合は見ることができません。
エクスポート範囲	エクスポート範囲には次が含まれます。 全インスタンス 検索結果のみ 選択したインスタンスのみ ここで、どのインスタンスにもチェックを入れていない場合、「選択したインスタンスのみ」はグレーアウト状態となり、選択できません。

CLBインスタンスのアップグレード

最終更新日：：2024-01-04 17:48:23

CLBのインスタンス仕様は共有タイプのインスタンスおよびLCUタイプのインスタンスをサポートしています。デフォルトではすべてのインスタンスは共有インスタンスです。共有タイプのインスタンスはLCUタイプのインスタンスにアップグレードすることができます。

アップグレードのメリット

共有CLBインスタンスは同時接続数5万、1秒あたりの新規接続数5000、1秒あたりの照会数（QPS）5000のパフォーマンス保障機能を提供します。共有CLBインスタンスはパフォーマンスの保障範囲内で転送パフォーマンスを単独で利用し、保障範囲を超える部分ではクラスターリソースを共有しますが、パフォーマンスの占有が存在する可能性があります。
パフォーマンス キャパシティ インスタンスにアップグレードすると、単一インスタンスで最大 1,000 万の同時接続、1 秒あたり 100 万の新規接続、および 1 秒あたり 300,000 のクエリ (QPS) をサポートできます。

アップグレードの影響

速度制限関連
アップグレード時、イントラネット パフォーマンス キャパシティ インスタンスは、対応する仕様の帯域幅の上限にデフォルト設定されます。アップグレード後にコンソールで仕様を調整できます。パブリック ネットワーク パフォーマンス キャパシティ インスタンスのデフォルトの帯域幅は、アップグレード前と同じです。アップグレード後にコンソールで帯域幅を調整できます。
アップグレード後はインスタンス仕様に応じて速度制限がかかり、インスタンス仕様の上限を超えると速度制限やパケットロスが発生します。性能および容量の速度制限指標については、以下の監視指標を参照してください。詳細については監視指標の説明をご参照ください。
ClientConcurConn（クライアントからLBへの同時接続数）
ClientNewConn（クライアントからLBへの新規接続数）
TotalReq（1秒あたりのリクエスト数）
ClientOuttraffic（クライアントからLBへのアウトバウンド帯域幅）
ClientIntraffic（クライアントからLBへのインバウンド帯域幅）
アップグレード後に実際のパフォーマンス消費がインスタンスのパフォーマンス速度制限値を超過しなければ、既存の接続への影響はありません。
課金関連
アップグレードの前後で課金モデルに変更はありません。

アップグレード後は、実際に消費されるパフォーマンスに応じて、1時間単位でロードバランサキャパシティユニット（LCU）料金が発生します。詳細については、ロードバランサキャパシティユニット（LCU）課金説明をご参照ください。

ネットワーク接続関連

アップグレード中にネットワークが中断されることはありません。アップグレードにかかる時間は1分以内です。

ダウングレード関連

アップグレード後に共有インスタンスに戻すことはできません。

アップグレードの制限

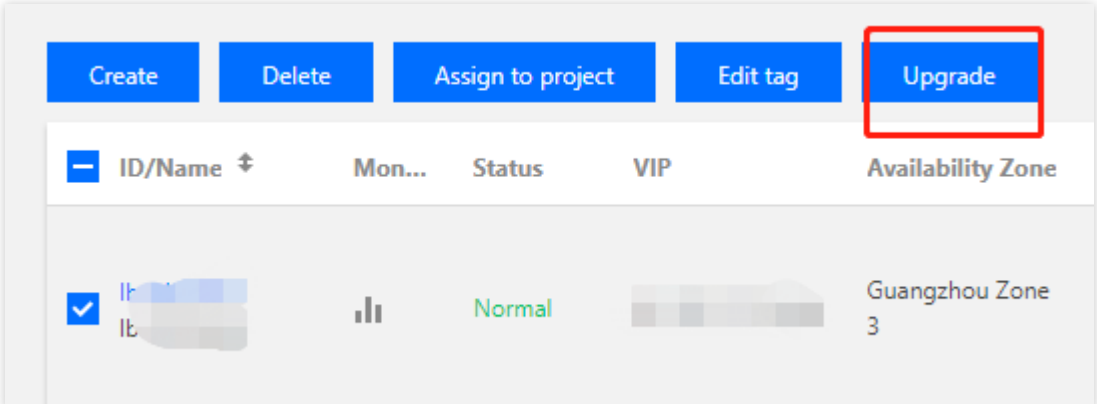
現在、LCUタイプインスタンスはベータ版テスト中です。ご利用を希望される場合は、チケット申請を提出してください。

複数の従量課金タイプの共有タイプインスタンスの一括アップグレードをサポートしています。

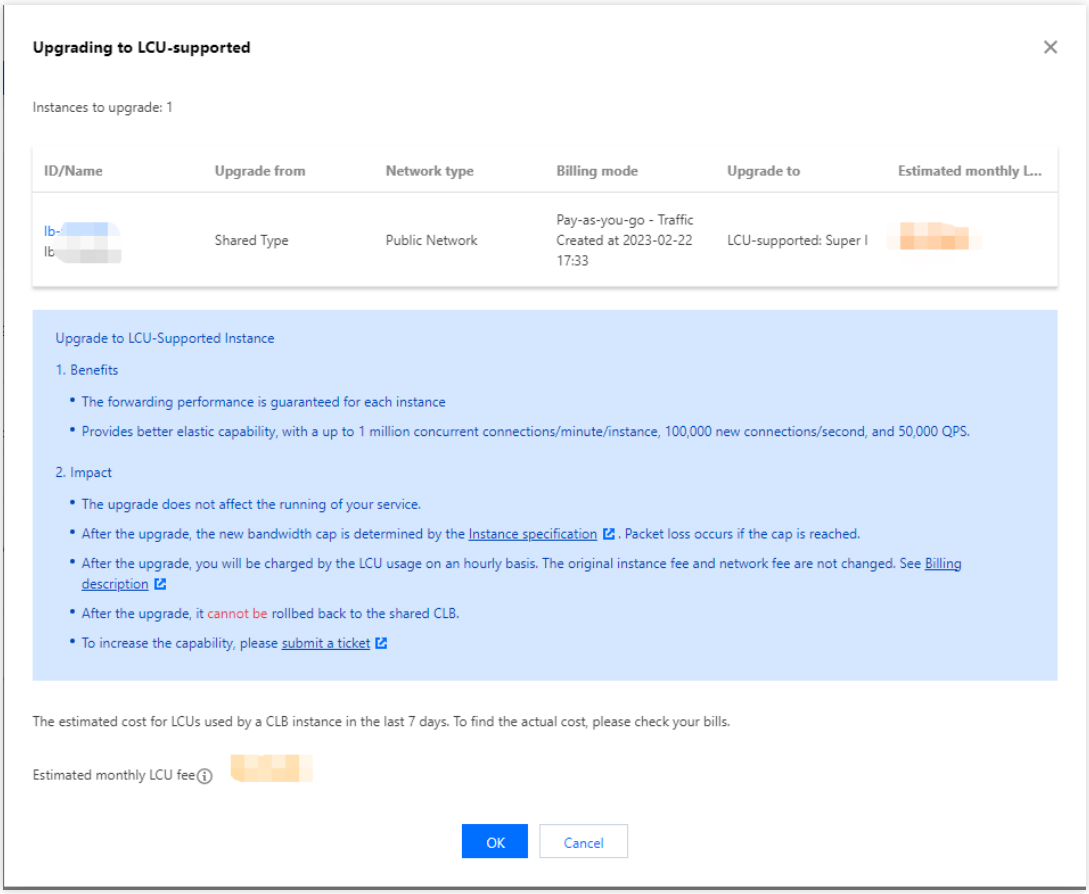
従来型のCLBインスタンスのLCUタイプインスタンスへのアップグレードはサポートしていません。

アップグレード方式

1. CLBコンソールにログインし、左側ナビゲーションバーの**インスタンス管理**をクリックします。
2. CLBのインスタンスリストで、アップグレード対象の共有タイプインスタンスにチェックを入れ、インスタンスリストの上にある**アップグレード**をクリックします。



3. ポップアップした「インスタンスのアップグレード」ダイアログボックスで、**OK**をクリックします。



関連ドキュメント

ロードバランサキャパシティユニット（LCU）課金説明

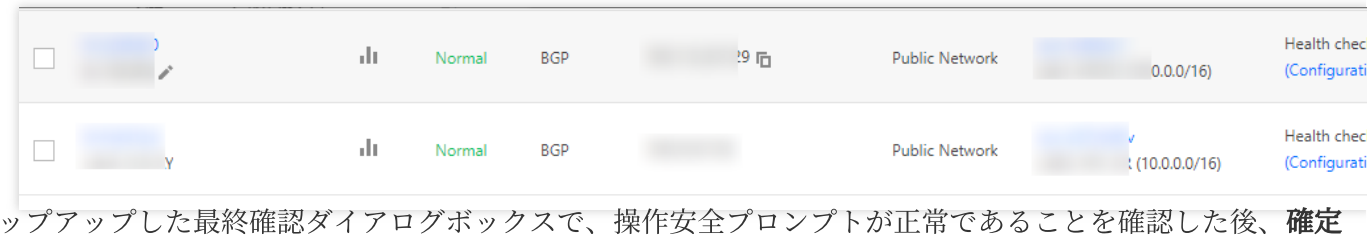
CLBインスタンスを削除

最終更新日：：2024-01-04 17:48:23

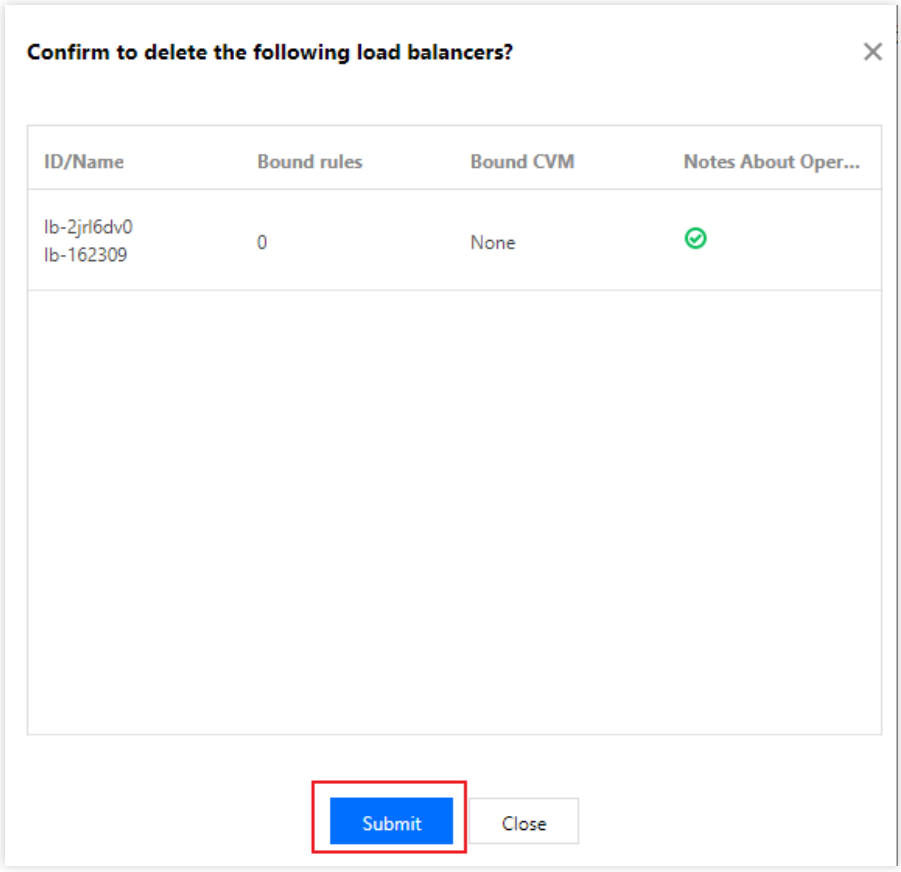
CLBインスタンスにトラフィックがすでになく、使用を継続する必要がないことを確認した場合は、CLBコンソールまたはAPIによってインスタンスを削除することができます。
削除したインスタンスは完全に破棄され、復元はできません。インスタンスを削除する前にすべてのバックエンドサーバーのバインドを解除し、一定期間様子を見てから削除操作を行うことを強く推奨します。

コンソールでCLBインスタンスを削除する

1. CLBコンソールにログインします。
2. 削除したいCLBインスタンスを見つけ、一番右側の操作バーの下にある**その他 > 削除**をクリックします。



3. ポップアップした最終確認ダイアログボックスで、操作安全プロンプトが正常であることを確認した後、**確定**をクリックすると削除できます。
最後にダイアログボックスが下図のように表示されていることを確認します。バインドルール数が「**0**」、バインドするバックエンドサーバーが「**なし**」、操作のセキュリティメッセージが「**グリーン**」であることを確認してから、さらに削除操作を行うことをお勧めします。



APIでCLBインスタンスを削除する

詳しい手順については、CLBの削除をご参照ください。

インスタンス削除の保護を設定

最終更新日：：2024-01-04 17:48:23

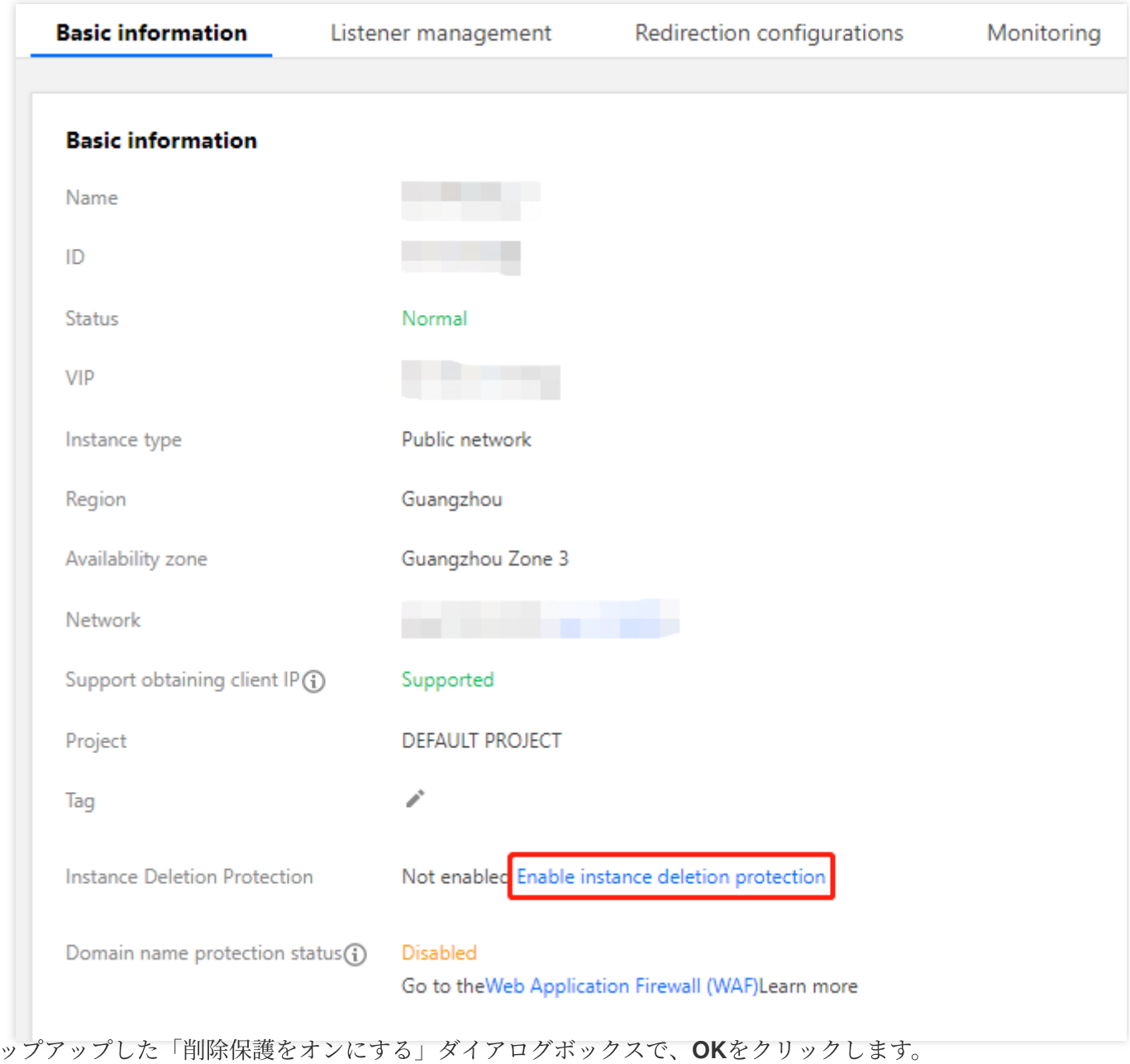
削除保護機能を有効化すると、誤って削除したことによるインスタンスの解放を防ぐことができます。

制限事項

CLBインスタンスが料金滞納でサービスを停止した場合、削除防止機能が有効になっていても受動的に解放されます。

操作手順

1. CLBコンソールにログインし、**インスタンス管理**ページの左上隅で、所在リージョンを選択します。
2. インスタンスリストで対象のインスタンスIDをクリックします。
3. インスタンスの基本情報ページで、**削除保護の有効化**をクリックします。



4. ポップアップした「削除保護をオンにする」ダイアログボックスで、**OK**をクリックします。

説明：

インスタンス削除保護機能を有効化すると、コンソールまたはAPIを呼び出してこのインスタンスを削除することができなくなります。インスタンスを削除したい場合は、インスタンスの基本情報ページで**削除保護をオフにする**をクリックしてから削除する必要があります。

関連ドキュメント

CLBインスタンスの削除。

インスタンスのパブリックネットワーク設定の調整

最終更新日：：2024-01-04 17:43:29

パブリックネットワークタイプのCLBでは必要に応じてパブリックネットワークの帯域幅または課金モデルの調整を行うことができます。この調整は即時有効になります。

制限事項

IPv4バージョンのCLB：ネットワーク設定の調整は標準アカウントタイプのみサポートしており、従来型アカウントタイプではサポートしていません。
IPv6バージョンのCLB：ネットワーク設定の調整は標準アカウントタイプと従来型アカウントタイプの両方でサポートしています。
アカウントタイプが確実ではない場合は、アカウントタイプの判断をご参照ください。

帯域幅の上限

インスタンス課金モデル	ネットワーク課金モデル	設定可能な帯域幅の上限範囲（Mbps）
従量課金	帯域幅課金（1時間単位帯域幅）	0 - 2048（2048を含む）
	トラフィック課金	
	共有帯域幅パッケージ	

説明：
上限の引き上げをご希望の場合は、チケット申請を提出するか、またはビジネスマネージャーまでご連絡ください。

帯域幅の調整

1. CLBコンソールにログインします。
2. 「インスタンス管理」ページで所在リージョンを選択し、目的のパブリックネットワークCLBインスタンスを見つけ、右側の「操作」バーで【その他】>【帯域幅の調整】を選択します。

3. ポップアップした「帯域幅の調整」ダイアログボックスで、目的の帯域幅上限値を設定し、【送信】をクリックします。

課金モデルの変更

1. CLBコンソールにログインします。
2. 「インスタンス管理」ページで所在リージョンを選択し、目的のパブリックネットワークCLBインスタンスを見つけ、右側の「操作」バーで【その他】をクリックし、ネットワーク課金モデルの調整を選択します。調整についての説明は次のとおりです。

インスタンス課金モデル	ネットワーク課金モデル	ネットワーク課金モデルの調整
従量課金	帯域幅課金（1時間単位帯域幅）	共有帯域幅パッケージの追加をサポート：インスタンス課金に変更はなく、ネットワーク課金を共有帯域幅パッケージ課金に切り替えます。切り替えは各CLBインスタンスにつき、1回のみ可能です。
	トラフィック課金	サブスクリプションへの切り替えをサポート：インスタンス課金をサブスクリプションに、ネットワーク課金を帯域幅課金（月額帯域幅）に切り替えます。切り替えは各CLBインスタンスにつき、1回のみ可能です。 共有帯域幅パッケージの追加をサポート：インスタンス課金に変更はなく、ネットワーク課金を共有帯域幅パッケージ課金に切り替えます。切り替え回数に制限はありません。
	共有帯域幅パッケージ	共有帯域幅パッケージの終了をサポート：インスタンス課金に変更はなく、ネットワーク課金をトラフィック課金に切り替えます。切り替え回数に制限はありません。

3. ポップアップしたダイアログボックスで、【送信】をクリックします。

CLBリスナー
CLBリスナーの概要

最終更新日：：2024-01-04 18:36:26

CLBインスタンスを作成後、インスタンスにリスナーを設定する必要があります。リスナーはCLBインスタンス上のリクエストを監視し、バランシングポリシーに基づいてトラフィックをバックエンドサーバーに振り分ける役割を担います。
CLBリスナーには次の設定が必要です。
1. リスニングプロトコルおよびリスニングポート。CLBのリスニングポートはフロントエンドポートとも呼ばれ、リクエストを受信してバックエンドサーバーに転送するためのポートとして用いられます。
2. リスニングポリシー。バランシングポリシー、セッション維持 など。
3. ヘルスチェック ポリシーです。
4. バインドするバックエンドサービス。バックエンドサーバーのIPおよびポートを設定する必要があります。サービスポートはバックエンドポートとも呼ばれ、バックエンドサービスがリクエストを受信するためのポートとして用いられます。

サポートするプロトコルタイプ

CLBリスナーはCLBインスタンス上のレイヤー4およびレイヤー7リクエストを監視し、これらのリクエストをバックエンドサーバーに振り分けることができ、その後バックエンドサーバーがリクエストを処理します。レイヤー4およびレイヤー7CLBの主な違いは、ユーザーのリクエストに対しロードバランシングを行う際に、トラフィックの転送をレイヤー4プロトコルとレイヤー7プロトコルのどちらに基づいて行うかという点にあります。例えば、TCP、UDPなどのレイヤー4プロトコルリクエストに対してはレイヤー4CLBが、HTTP、HTTPSなどのレイヤー7プロトコルリクエストに対してはレイヤー7CLBがそれぞれ用いられます。
レイヤー4プロトコル：トランスポート層プロトコルです。主にVIP + Portによってリクエストを受信し、トラフィックをバックエンドサーバーに分配します。
レイヤー7プロトコル：アプリケーション層プロトコルです。URL、HTTPヘッダーなどのアプリケーション層情報に基づいてトラフィックを振り分けます。
レイヤー4リスナーを使用する（レイヤー4プロトコルを使用して転送する）場合、CLBインスタンスはリスニングポート上にバックエンドインスタンスとのバックエンドインスタンス接続を確立し、リクエストをバックエンドサーバーに直接転送します。このプロセスではデータパケットの変更が何も行われないため（パススルーモード）、転送効率が非常に高くなります。
Tencent Cloud CLBは次のプロトコルによるリクエスト転送をサポートしています。
TCP（トランスポート層）
UDP（トランスポート層）

TCP SSL（トランスポート層）
QUIC（トランスポート層）
HTTP（アプリケーション層）
HTTPS（アプリケーション層）

説明：

TCP SSLリスナーは現在パブリックネットワークCLBのみサポートしています（プライベートネットワークはサポートしていません。従来型CLBはサポートしていません。

プロトコルカテゴリー	プロトコル	説明	ユースケース
レイヤー4プロトコル	TCP	接続指向で、信頼性の高いトランスポート層プロトコル 伝送する移行元および端末はまず3ウェイハンドシェイクで接続を確立し、さらにデータを伝送する必要があります クライアントIP（ソースIP）に基づくセッション維持をサポートしています ネットワーク層でクライアントIPを見ることができます サーバーは直接クライアントIPを取得できます	信頼性およびデータの正確性に対する要件が高く、ファイル伝送、メール送受信、リモートログインなど、伝送速度に対する要件が比較的低いシーンに適しています。詳細についてはTCPリスナーの設定をご参照ください。
	UDP	接続がないトランスポート層プロトコル 伝送する移行元と端末は接続を確立せず、接続状態を維持する必要はありません 各UDP接続はいずれもポイントツーポイントのみ可能です 1対1、1対多、多対1および多対多の相互通信をサポートしています クライアントIP（ソースIP）に基づくセッション維持をサポートしています サーバーは直接クライアントIPを取得できます	インスタントメッセージ、オンラインビデオなど、伝送効率に対する要件が高く、正確性に対する要件が比較的低いシーンに適しています。詳細については、UDPリスナーの設定をご参照ください。
	TCP SSL	安全なTCP TCP SSLリスナーは証明書の設定をサポートし、承認されていないアクセスを阻止します 一元的な証明書管理サービス、CLBによって復号操作を完了します。 単方向認証および双方向認証をサポートしています	TCPプロトコルの下でのセキュリティ要件が非常に高いシーンに適しています。TCPベースのカスタムプロトコルをサポートしています。詳細については、TCP SSLリスナーの設定をご参照ください。

		サーバーは直接クライアントIPを取得できます	
	QUIC	UDPのマルチパス通信プロトコルをベースにしています。 UDP上でデータの信頼性の高い伝送、セキュリティおよびHTTP2を実現し、TCP + TLS + HTTP2と同等の効果を有します。 QIUC接続では、IPやポートにどのような変化があっても接続の中断や再接続が起こらないため、シームレスなコネクションマイグレーションが実現できます。	オーディオビデオ業務、ゲーム業務などでネットワークに変化が生じる場合、例えば4GネットワークとWi-Fiネットワークを頻繁に切り替える場合など、中断せずスムーズに接続を移行したいシーンに適しています。詳細については、QUICリスナーの設定をご参照ください。
レイヤー7プロトコル	HTTP。	アプリケーション層プロトコル リクエストドメイン名およびURLに基づく転送をサポートしています Cookieに基づくセッション維持をサポートしています	WebアプリケーションやAppサービスなど、リクエストの内容を認識する必要があるアプリケーションに適しています。詳細については、HTTPリスナーの設定をご参照ください。
	HTTPS	暗号化されたアプリケーション層プロトコル リクエストドメイン名およびURLに基づく転送をサポートしています Cookieに基づくセッション維持をサポートしています 一元的な証明書管理サービス、CLBによって復号操作を完了します。 単方向認証および双方向認証をサポートしています	暗号化通信が必要なHTTPアプリケーションに適しています。詳細については、HTTPSリスナーの設定をご参照ください。

ポートの設定

ポートタイプ	説明	制限
リスニングポート（フロントエンドポート）	リスニングポートとは、CLBがリクエストを受信してバックエンドサーバーにリクエストを転送するためのポートです。1～65535番ポートにCLBを設定することができます。例えば、21（FTP）、25（SMTP）、80（HTTP）、443（HTTPS）などです。	同一のCLBインスタンス内ではUDPクラスのプロトコルはTCPクラスのプロトコルのリスニングポートと重複することができます。例えば、リスナーのTCP:80とリスナーのUDP:80であれば、同時に作成することができます。 同一クラスのプロトコル下ではリスニングポートは重複できません。TCP/TCP

		SSL/HTTP/HTTPSはTCPクラスに属しています。例えば、同時にリスナーTCP:80およびリスナー HTTP:80を作成することはできません。
サービスポート（バックエンドポート）	サービスポートとはバックエンドサーバーがサービスを提供するためのポートであり、CLBからのトラフィックを受信して処理します。1つのCLBインスタンスにおいて、同一のCLBリスニングポートが複数のバックエンドサーバーの複数のポートにトラフィックを転送することができます。	同一のCLBインスタンス内では異なるリスニングプロトコルのサービスポートは重複することができます。例えば、リスナーのHTTP:80とリスナーのHTTPS:443は、同じバックエンドサーバーの同じポートを同時にバインドすることができます。 同一リスニングプロトコルでは、同一のバックエンドサービスポートは1つのリスナーのみによってバインドされます。すなわち4つ組（VIP、リスニングプロトコル、バックエンドサービスのプライベートIP、バックエンドサービスポート）は一意である必要があります。

関連ドキュメント

使用上の制約

TCPリスナーの設定

最終更新日：：2024-01-04 18:36:26

CLBインスタンスにTCPリスナーを追加して、クライアントからのTCPプロトコルリクエストを転送することができます。TCPプロトコルは、信頼性およびデータの正確性に対する要件が高く、伝送速度に対する要件が比較的低いシーン（ファイル伝送、メール送受信、リモートログインなど）に適しています。TCPリスナーにバインドしたバックエンドサーバーはクライアントのリアルIPを直接取得することができます。

前提条件

CLBインスタンスの作成が必要です。

操作手順

ステップ1：リスナーの設定

1. CLBコンソールにログインし、左側ナビゲーションバーの**インスタンス管理**をクリックします。
2. CLBインスタンスリストページの左上隅でリージョンを選択し、インスタンスリスト右側の操作列で**リスナーの設定**をクリックします。

3. TCP/UDP/TCP SSL/QUIC リスナーで**新規作成**をクリックし、ポップアップした「リスナーの作成」ダイアログボックスでTCPリスナーの設定を行います。

3.1 **基本設定**

リスナーの基本設定	説明	事例
名前	リスナーの名称です。	test-tcp-80
リスニングプロトコルポート	リスニングプロトコル：この例ではTCPを選択します。 リスニングポート：リクエストを受信してバックエンドサーバーにリクエストを転送するために使用するポートで、ポート範囲は1～65535です。 同一CLBインスタンス内で、リスニングポートは重複できません。	TCP:80

バランシング方式	TCPリスナーでは、CLBは重み付けラウンドロビン（WRR）および重み付け最小接続（WLC）の2種類のスケジューリングアルゴリズムをサポートしています 重み付けラウンドロビンアルゴリズム：バックエンドサーバーの重みに基づき、順番にリクエストを異なるサーバーに配信します。重み付けラウンドロビンアルゴリズムは新規接続数に基づいてスケジューリングし、重みの高いサーバーがラウンドロビンされる回数（確率）が高くなるほど、同じ重みのサーバーは同じ数の接続数を処理します。 重み付け最小接続：サーバーの現在アクティブな接続数に基づいてサーバーの負荷状況を推定します。重み付け最小接続はサーバー負荷および重みに基づいて総合的にスケジューリングし、重み値が同じ場合、現在の接続数が少ないバックエンドサーバーほどラウンドロビンされる回数（確率）も高くなります。 **説明**：重み付け最小接続のバランシング方式を選択した場合、リスナーはセッション維持機能の有効化をサポートしません。	重み付けラウンドロビン
双方向RST	チェックを入れると、対応する操作によって両側（クライアントとサーバー）に対しRSTレポートを送信して接続を終了します。チェックを入れない場合は双方向RSTを送信せず、タイムアウトするまで長時間接続します。	チェックを入れる

3.2 ヘルスチェック

ヘルスチェックの詳細については、TCPヘルスチェックをご参照ください。

3.3 セッション維持

セッション維持の設定	説明	事例
セッション維持の有効化/無効化	セッションの維持を有効化すると、CLBリスナーは同一クライアントからのアクセスリクエストを同一のバックエンドサーバーに配信します。 TCPプロトコルはクライアントIPアドレスのセッションの維持に基づき、同一IPアドレスからのアクセスリクエストを同一のバックエンドサーバーに転送します。 重み付けラウンドロビンスケジューリングはセッションの維持をサポートします。 重み付け最小接続スケジューリングはセッション維持機能の有効化をサポートしていません。	オン
セッションの維持時間	セッションの維持時間 維持時間を超え、接続中に新たなリクエストがない場合は、自動的にセッションの維持が切断されます。 設定可能範囲は30～3600秒です。	30s

ステップ2：バックエンドサーバーのバインド

1.「リスナー管理」ページで、上記の `TCP:80` リスナーなどの、先ほど作成したリスナーをクリックすると、リスナーの右側にバインド済みのバックエンドサービスが表示されます。

2. **バインド**をクリックし、バインドしたいバックエンドサーバーをポップアップボックスから選択し、サービスポートと重みを設定します。

説明

デフォルトポート機能：先に「デフォルトポート」を入力してからバックエンドサーバーを選択すると、それぞれのバックエンドサーバーのポートがすべてデフォルトポートとなります。

ステップ3：セキュリティグループ（オプション）

CLBのセキュリティグループを設定して、パブリックネットワークトラフィックの分離を行うことができます。詳細については、CLBセキュリティグループの設定をご参照ください。

ステップ4：リスナーの変更/削除（オプション）

作成したリスナーを変更または削除したい場合、「リスナー管理」ページで、作成したリスナーをクリックし、

アイコンをクリックして変更または

アイコンをクリックして削除してください。

UDPリスナーの設定

最終更新日：：2024-01-04 18:36:26

CLBインスタンスにUDPリスナーを追加して、クライアントからのUDPプロトコルリクエストを転送することができます。UDPプロトコルは、伝送効率に対する要件が高く、正確性に対する要件が比較的低いシーン（インスタントメッセージ、オンラインビデオなど）に適しています。UDPプロトコルのリスナーでは、バックエンドサーバーはクライアントのリアルIPを直接取得することができます。

制限事項

UDPリスナーの4789番ポートはシステムによって予約されているポートであり、現時点では外部に開放されていません。

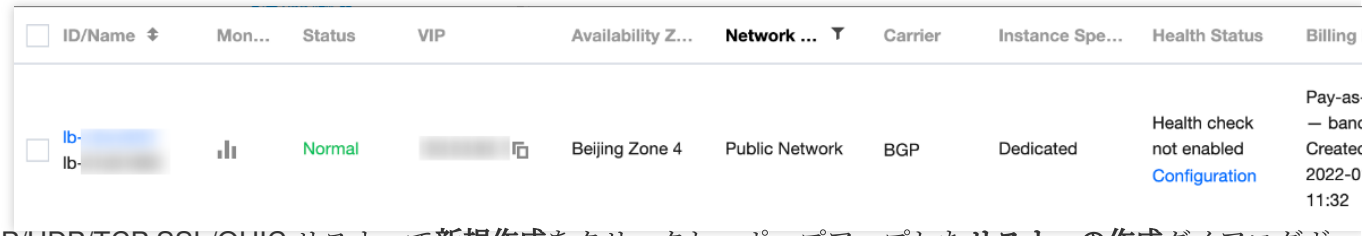
前提条件

CLBインスタンスの作成が必要です。

操作手順

ステップ1：リスナーの設定

1. CLBコンソールにログインし、左側ナビゲーションバーの**インスタンス管理**をクリックします。
2. CLBインスタンスリストページの左上隅でリージョンを選択し、インスタンスリスト右側の操作列で**リスナーの設定**をクリックします。



3. TCP/UDP/TCP SSL/QUIC リスナーで**新規作成**をクリックし、ポップアップした**リスナーの作成**ダイアログボックスでUDPリスナーの設定を行います。

3.1 基本設定

リスナーの基本設定	説明	事例

名前	リスナーの名称です。	test-udp-8000
リスニングプロトコルポート	リスニングプロトコル：この例ではUDPを選択します。 リスニングポート：リクエストを受信してバックエンドサーバーにリクエストを転送するために使用するポートで、ポート範囲は1～65535です。このうち、4789ポートはシステムがポートを保持し、外部に開放されていません。 同一CLBインスタンス内で、リスニングポートは重複できません。	UDP:8000
バランシング方式	UDPリスナー内では、CLBは重み付けラウンドロビン（WRR）および重み付け最小接続（WLC）の2種類のスケジューリングアルゴリズムをサポートしています。 重み付けラウンドロビンアルゴリズム：バックエンドサーバーの重みに基づき、順番にリクエストを異なるサーバーに配信します。重み付けラウンドロビンアルゴリズムは新規接続数に基づいてスケジューリングし、重みの高いサーバーがラウンドロビンされる回数（確率）が高くなるほど、同じ重みのサーバーは同じ数の接続数を処理します。 重み付け最小接続：サーバーの現在アクティブな接続数に基づいてサーバーの負荷状況を推定します。重み付け最小接続はサーバー負荷および重みに基づいて総合的にスケジューリングし、重み値が同じ場合、現在の接続数が少ないバックエンドサーバーほどラウンドロビンされる回数（確率）も高くなります。 **説明**：重み付け最小接続のバランシング方式を選択した場合、リスナーはセッション維持機能の有効化をサポートしません。	重み付けラウンドロビン
QUIC IDによるスケジューリング	有効化すると、CLBはQUIC IDによってスケジューリングされ、同一のQUIC Connection IDは同一のバックエンドサーバーにスケジューリングされます。クライアントリクエストにQUIC Connection IDが含まれない場合は一般的な重み付けラウンドロビンにダウングレードされ、4つ組（ソースIP+ターゲットIP+ソースポート+ターゲットポート）によってスケジューリングされます。	オン

3.2 ヘルスチェック

ヘルスチェックの詳細については、UDPヘルスチェックをご参照ください。

3.3 セッション維持

セッション維持の設定	説明	事例
セッション維持の有効化/無効化	セッションの維持を有効化すると、CLBリスナーは同一クライアントからのアクセスリクエストを同一のバックエンドサーバーに配信します。 TCPプロトコルはクライアントIPアドレスのセッションの維持に基づき、同一IPアドレスからのアクセスリクエストを同一のバックエンドサーバーに転送します。	オン

	重み付けラウンドロビンスケジューリングはセッションの維持をサポートします。重み付け最小接続スケジューリングはセッション維持機能の有効化をサポートしていません。	
セッションの維持時間	セッションの維持時間 維持時間を超え、接続中に新たなリクエストがない場合は、自動的にセッションの維持が切断されます。 設定可能範囲は30～3600秒です。	30s

ステップ2：バックエンドサーバーのバインド

1. **リスナー管理**ページで、上記の `UDP:8000` リスナーなどの、先ほど作成したリスナーをクリックすると、リスナーの右側にバインド済みのバックエンドサービスが表示されます。
2. **バインド**をクリックし、バインドしたいバックエンドサーバーをポップアップボックスから選択し、サービスポートと重みを設定します。

説明：
デフォルトポート機能：先に「デフォルトポート」を入力してからバックエンドサーバーを選択すると、それぞれのバックエンドサーバーのポートがすべてデフォルトポートとなります。

ステップ3：セキュリティグループ（オプション）

CLBのセキュリティグループを設定して、パブリックネットワークトラフィックの分離を行うことができます。詳細については、CLBセキュリティグループの設定をご参照ください。

ステップ4：リスナーの変更/削除（オプション）

作成したリスナーを変更または削除したい場合、「リスナー管理」ページで、作成したリスナーをクリックし、

アイコンをクリックして変更または

アイコンをクリックして削除してください。

TCP SSLリスナーの設定

最終更新日：：2024-01-04 18:36:26

CLBインスタンスにTCP SSLリスナーを追加して、クライアントからの暗号化されたTCPプロトコルリクエストを転送することができます。TCP SSLプロトコルは、超ハイパフォーマンスかつ大規模なTLSオフロードのシーンに適しています。TCP SSLプロトコルのリスナーでは、バックエンドサーバーがクライアントのリアルIPを直接取得することができます。

説明：

TCP SSLリスナーは現在CLBインスタンスタイプのみサポートしています。従来型CLBはサポートしていません。

ユースケース

TCP SSLはTCPプロトコルでセキュリティの要求が非常に高いシナリオに適用されます。
TCP SSLリスナーは証明書の設定をサポートし、承認されていないアクセスを阻止します。
一元的な証明書管理サービスをサポートし、CLBによって復号操作を完了します。
単方向認証および双方向認証をサポートしています。
サーバーは直接クライアントIPを取得できます。

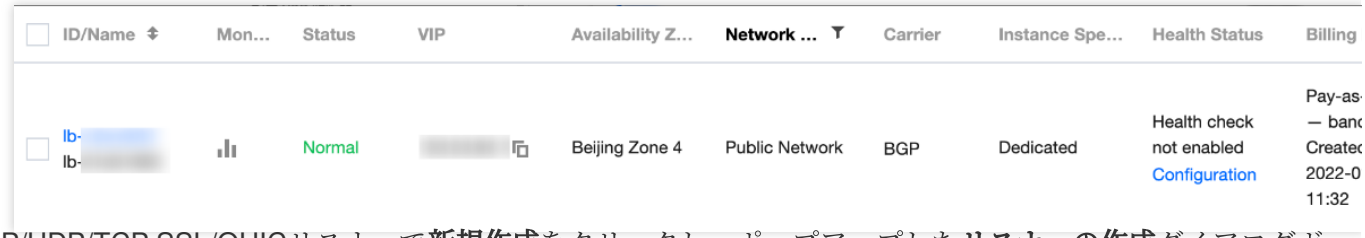
前提条件

CLBインスタンスの作成が必要です。

操作手順

ステップ1：リスナーの設定

1. CLBコンソールにログインし、左側ナビゲーションバーの**インスタンス管理**をクリックします。
2. CLBインスタンスリストページの左上隅でリージョンを選択し、インスタンスリスト右側の操作列で**リスナーの設定**をクリックします。



3. TCP/UDP/TCP SSL/QUICリスナーで**新規作成**をクリックし、ポップアップした**リスナーの作成**ダイアログボックスでTCP SSLリスナーの設定を行います。

3.1 基本設定

リスナーの基本設定	説明	事例
名前	リスナーの名称です。	test-tcpssl-9000
リスニングプロトコルポート	リスニングプロトコル：この例ではTCP SSLを選択します。 リスニングポート：リクエストを受信してバックエンドサーバーにリクエストを転送するために使用するポートで、ポート範囲は1～65535です。 同一CLBインスタンス内で、リスニングポートは重複できません。	TCP SSL:9000
SSL解析方式	単方向認証および双方向認証をサポートしています。	単方向認証
サーバー証明書	SSL証明書プラットフォームにすでにある証明書を選択するか、または証明書をアップロードできます。	既存証明書を選択
バランシング方式	TCP SSLリスナーでは、CLBは重み付けラウンドロビン（WRR）および重み付け最小接続（WLC）の2種類のスケジューリングアルゴリズムをサポートしています。 重み付けラウンドロビンアルゴリズム：バックエンドサーバーの重みに基づき、順番にリクエストを異なるサーバーに配信します。重み付けラウンドロビンアルゴリズムは新規接続数に基づいてスケジューリングし、重みの高いサーバーがラウンドロビンされる回数（確率）が高くなるほど、同じ重みのサーバーは同じ数の接続数を処理します。 重み付け最小接続：サーバーの現在アクティブな接続数に基づいてサーバーの負荷状況を推定します。重み付け最小接続はサーバー負荷および重みに基づいて総合的にスケジューリングし、重み値が同じ場合、現在の接続数が少ないバックエンドサーバーほどラウンドロビンされる回数（確率）も高くなります。	重み付けラウンドロビン

3.2 ヘルスチェック

ヘルスチェックの詳細については、TCP SSLヘルスチェックをご参照ください。

3.3 セッションの維持（現時点ではサポートしていません）

TCP SSLリスナーは、現時点ではセッション維持をサポートしていません。

ステップ2：バックエンドサーバーのバインド

1. **リスナー管理**ページで、上記の `TCP SSL:9000` リスナーなどの、先ほど作成したリスナーをクリックすると、リスナーの右側にバインド済みのバックエンドサービスが表示されます。
2. **バインド**をクリックし、バインドしたいバックエンドサーバーをポップアップボックスから選択し、サービスポートと重みを設定します。

説明：
デフォルトポート機能：先に「デフォルトポート」を入力してからバックエンドサーバーを選択すると、それぞれのバックエンドサーバーのポートがすべてデフォルトポートとなります。

ステップ3：セキュリティグループ（オプション）

CLBのセキュリティグループを設定して、パブリックネットワークトラフィックの分離を行うことができます。詳細については、CLBセキュリティグループの設定をご参照ください。

ステップ4：リスナーの変更/削除（オプション）

作成したリスナーを変更または削除したい場合、「リスナー管理」ページで、作成したリスナーをクリックし、

アイコンをクリックして変更または

アイコンをクリックして削除してください。

QUICリスナーを設定する

最終更新日：：2024-01-04 18:36:26

CLBインスタンスにQUICリスナーを追加すると、クライアントからの暗号化されたQUICプロトコルリクエストを転送できます。QUICプロトコルのリスナーによって、バックエンドサーバーはクライアントのリアルIPを直接取得できます。
QUIC（Quick UDP Internet Connection）は高速UDPインターネット接続とも呼ばれ、Googleが提唱する、UDPを使用してマルチパス通信を行うプロトコルです。現在広く用いられているTCP+TLS+HTTP2プロトコルと比較して、QUICには次のようなメリットがあります。
接続確立時間が短縮されます。
輻輳制御が改善されます。
多重化によってHOLブロッキングを解消します。
コネクションのマイグレーションが可能です。

シナリオ

QUICリスナーは接続移行をサポートしており、4GネットワークとWi-Fiネットワークの頻繁な切り替えなど、ネットワークに変化が生じても、接続を中断することなくスムーズに移行することができます。オーディオビデオサービス、ゲームサービスなどに適しています。

制限事項

QUICリスナーはCLBインスタンスのみでサポートされており、従来型のCLBではサポートされていません。
QUICリスナーはVPCネットワークタイプのCLBインスタンスのみでサポートされており、基幹ネットワークタイプではサポートされていません。
QUICリスナーはIPv4、IPv6 NAT64バージョンのCLBインスタンスのみでサポートしています。IPv6バージョンではサポートしていません。

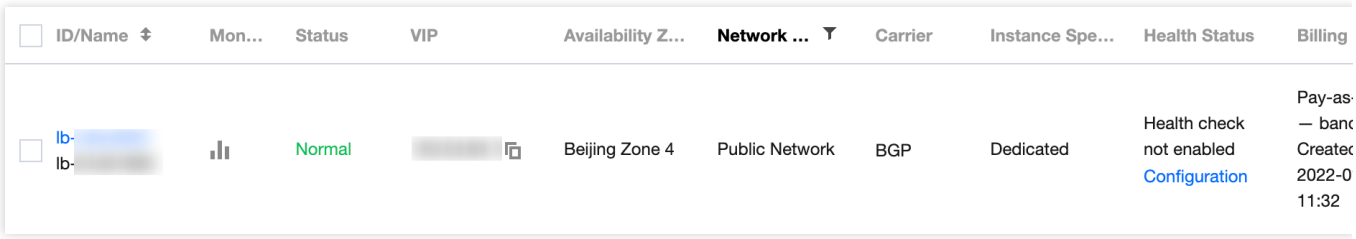
前提条件

CLBインスタンスの作成が必要です。

操作手順

ステップ1：リスナーの設定

1. CLBコンソールにログインし、左側のナビゲーションバーでインスタンス管理をクリックします。
2. CLBインスタンスリストページの左上隅でリージョンを選択し、インスタンスリスト右側の操作列でリスナーの設定をクリックします。



3. TCP/UDP/TCP SSL/QUICリスナーで新規作成をクリックし、ポップアップしたリスナーの作成ダイアログボックスでQUICリスナーの設定を行います

3.1 基本設定

リスナーの基本設定	説明	事例
名称	リスナーの名称です。	test-quic-443
リスニングプロトコルポート	リスニングプロトコル：この例ではQUICを選択します。QUICを選択すると、CLBはクライアントからQUICリクエストを受信して、CLBとバックエンドサーバーの間でTCPプロトコルを使用することができます。 リスニングポート：リクエストを受信してバックエンドサーバーにリクエストを転送するために用いるポートです。ポートの範囲は1～65535です。 同一のCLBインスタンス内では、リスニングポートを重複してはなりません。	QUIC:443
SSL解析方式	単方向認証および双方向認証をサポートしています。	単方向認証
サーバー証明書	SSL証明書プラットフォームにすでにある証明書を選択するか、または証明書をアップロードできます。	既存の証明書の選択
バランシング方式	QUICリスナーにおいて、CLBは重み付けラウンドロビン（WRR）と重み付け最小接続（WLC）という2種類のスケジューリングアルゴリズムをサポートしています。 重み付けラウンドロビンアルゴリズム：バックエンドサーバーの重み付けに基づき、リクエストを順序に従ってそれぞれのサーバーに振り分けます。重み付けラウンドロビンアルゴリズムでは新規接続数に基づいてスケジューリングを行います。重みが高いサーバーへの問い合わせ回数が多く（確率が高く）なると、同じ重みのサーバーはそれに応じて同等の接続数を処理します。	重み付けラウンドロビン

	重み付け最小接続：サーバーの現在のアクティブ接続数に応じてサーバーの負荷状態を予測します。重み付け最小接続はサーバーの負荷と重みを総合的にスケジューリングし、重みが同じ場合は、その時点での接続数が少ないバックエンドサーバーへの問い合わせ回数を多く（確率を高く）します。	

3.2 **ヘルスチェック**

ヘルスチェックの詳細については、TCP SSLヘルスチェックをご参照ください。

3.3 **セッション維持**

QUICリスナーは、現時点ではセッション維持をサポートしていません。

ステップ2：バックエンドサーバーのバインド

1. リスナー管理ページで、上記のQUIC:443リスナーなどの、先ほど作成したリスナーをクリックすると、リスナーの右側にバインド済みのバックエンドサービスが表示されます。
2. バインドをクリックし、バインドしたいバックエンドサーバーをポップアップボックスから選択し、サービスポートと重みを設定します。

説明：

デフォルトポート機能：先にデフォルトポートを入力してからバックエンドサーバーを選択すると、それぞれのバックエンドサーバーのポートがすべてデフォルトポートとなります。

手順3：セキュリティグループの設定

CLBのセキュリティグループを設定して、パブリックネットワークトラフィックの分離を行う必要があります。詳細については、CLBセキュリティグループの設定をご参照ください。

ステップ4：リスナーの変更/削除（オプション）

作成済みのリスナーを変更または削除したい場合は、「リスナー管理」ページで作成済みのリスナーをクリックし、

のアイコンをクリックして変更するか、

のアイコンをクリックして削除してください。

関連ドキュメント

CLBによるQUICプロトコルのサポート

HTTPリスナーの設定

最終更新日：：2024-01-04 18:36:26

CLBインスタンスにHTTPリスナーを追加し、クライアントからのHTTPプロトコルリクエストを転送することができます。HTTPプロトコルはWebアプリケーションやAppサービスなど、リクエストの内容を認識する必要があるアプリケーションに適しています。

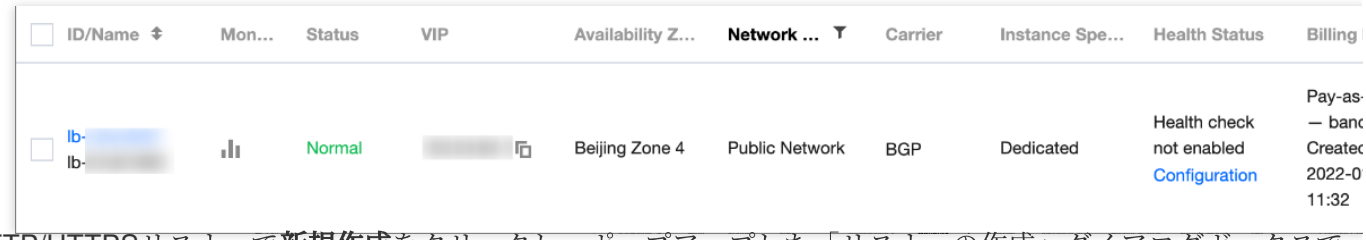
前提条件

CLBインスタンスの作成が必要です。

操作手順

ステップ1：リスナーの設定

1. CLBコンソールにログインし、左側ナビゲーションバーの**インスタンス管理**をクリックします。
2. CLBインスタンスリストページの左上隅でリージョンを選択し、インスタンスリスト右側の操作列で**リスナーの設定**をクリックします。



3. HTTP/HTTPSリスナーで**新規作成**をクリックし、ポップアップした「リスナーの作成」ダイアログボックスでHTTPリスナーの設定を行います。

3.1 リスナーの作成

リスナーの基本設定	説明	事例
名前	リスナーの名称です。	test-http-80
リスニングプロトコルポート	リスニングプロトコル：この例ではHTTPを選択します。 リスニングポート：リクエストを受信してバックエンドサーバーにリクエストを転送するために使用するポートで、ポート範囲は1～65535です。 同一CLBインスタンス内で、リスニングポートは重複できません。	HTTP:80
長時間接	有効化すると、CLBとバックエンドサービス間で長時間接続を使用し、CLBは	未有効化

続の有効化	ソースIPをパススルーしなくなりますので、XFFからソースIPを取得してください。正常な転送を保証するため、CLB上でセキュリティグループを有効化してデフォルトで許可するか、またはCVMのセキュリティグループで100.127.0.0/16を許可してください。 **説明**：有効化すると、CLBとバックエンドサーバーの接続数の範囲はリクエストの[QPS、QPS*60]の間で変動し、具体的な数値は接続再利用率によって決まります。バックエンドサービスが接続数の上限に制限を設けている場合、有効化は慎重に行うことをお勧めします。この機能は現在ベータ版テスト段階です。ご利用を希望される場合は、チケット申請を行ってください。	

3.2 転送ルールの作成

転送ルールの基本設定	説明	事例
ドメイン名	転送ドメイン名： 長さ制限：1～80文字です。 _ で始めることはできません。 正確なドメイン名およびワイルドカードのドメイン名をサポートしています。 正規表現をサポートしています。 具体的な設定ルールです。詳細については、転送ドメイン名設定ルールをご参照ください。	www.example.com
デフォルトドメイン名	リスナーのすべてのドメイン名がマッチングに成功しなかった場合、システムはリクエストにデフォルトのアクセスドメイン名を指定し、デフォルトアクセスを制御可能にします。 1つのリスナーに設定できるデフォルトドメイン名は1つのみです。	デフォルトで有効
URLパス	転送URLパス： 長さ制限：1～200文字です。 正規表現をサポートしています。 具体的な設定ルールです。詳細については、転送URLパス設定ルールをご参照ください。	/index
バランシング方式	HTTPリスナーでは、CLBは重み付けラウンドロビン（WRR）、重み付け最小接続（WLC）およびIP Hashの3種類のスケジューリングアルゴリズムをサポートしています。 重み付けラウンドロビンアルゴリズム：バックエンドサーバーの重みに基づき、順番にリクエストを異なるサーバーに配信します。重み付けラウンドロビンアルゴリズムは新規接続数に基づいてスケジューリングし、重みの高いサーバーがラウンドロビンされる回数（確率）が高くなるほど、同じ重みのサーバーは同じ数の接続数を処理します。 重み付け最小接続：サーバーの現在アクティブな接続数に基づいてサーバーの負荷状況を推定します。重み付け最小接続はサーバー負荷	重み付けラウンドロビン

	および重みに基づいて総合的にスケジューリングし、重み値が同じ場合、現在の接続数が少ないバックエンドサーバーほどラウンドロビンされる回数（確率）も高くなります。 IP Hash：リクエストのソースIPアドレスに応じて、ハッシュキー（Hash Key）を使用し、静的に割り当てられたハッシュテーブルから対応するサーバーを見つけます。そのサーバーが使用可能であり、かつオーバーロード状態ではない場合はリクエストがそのサーバーに送信され、そうではない場合は空が返されます。	
クライアントIPを取得	デフォルトで有効	すでにオンです
Gzip圧縮	デフォルトで有効	すでにオンです

3.3 ヘルスチェック

ヘルスチェックの詳細については、 HTTPヘルスチェックをご参照ください。

3.4 セッション維持

セッション維持の設定	説明	事例
セッション維持の有効化/無効化	セッションの維持を有効化すると、CLBリスナーは同一クライアントからのアクセスリクエストを同一のバックエンドサーバーに配信します。 TCPプロトコルはクライアントIPアドレスのセッションの維持に基づき、同一IPアドレスからのアクセスリクエストを同一のバックエンドサーバーに転送します。 重み付けラウンドロビンスケジューリングはセッションの維持をサポートします。重み付け最小接続スケジューリングはセッション維持機能の有効化をサポートしていません。	オン
セッションの維持時間	維持時間を超え、接続中に新たなリクエストがない場合は、自動的にセッションの維持が切断されます。 設定可能範囲は30～3600秒です。	30s

ステップ2：バックエンドサーバーのバインド

1. 「リスナー管理」ページで、上記の `HTTP:80` リスナーなどの、先ほど作成したリスナーをクリックし、左側の**+**アイコンをクリックしてドメイン名およびURLパスを表示します。具体的なURLパスを選択すると、リスナーの右側にそのパスにバインド済みのバックエンドサービスが表示されます。
2. **バインド**をクリックし、バインドしたいバックエンドサーバーをポップアップボックスから選択し、サービスポートと重みを設定します。

説明：

デフォルトポート機能：先に「デフォルトポート」を入力してからバックエンドサーバーを選択すると、それぞれのバックエンドサーバーのポートがすべてデフォルトポートとなります。

ステップ3：セキュリティグループ（オプション）

CLBのセキュリティグループを設定して、パブリックネットワークトラフィックの分離を行うことができます。詳細については、CLBセキュリティグループの設定をご参照ください。

ステップ4：リスナーの変更/削除（オプション）

作成したリスナーを変更または削除したい場合、「リスナー管理」ページで、作成したリスナーをクリックし、

アイコンをクリックして変更または

アイコンをクリックして削除してください。

HTTPSリスナーの設定

最終更新日：：2024-01-04 18:36:26

CLBインスタンスにHTTPSリスナーを追加し、クライアントからのHTTPSプロトコルリクエストを転送することができます。HTTPSプロトコルは暗号化伝送を必要とするHTTPアプリケーションに適しています。

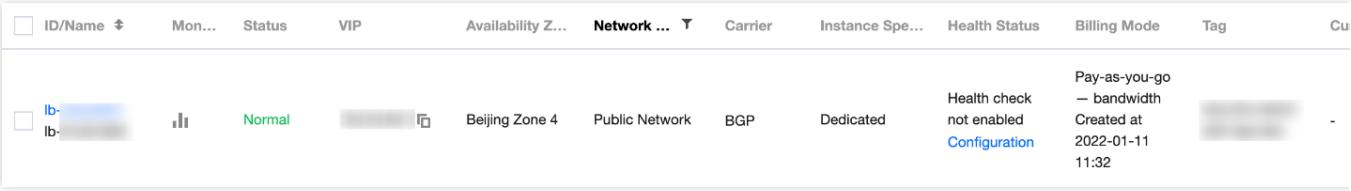
前提条件

CLBインスタンスの作成が必要です。

操作手順

ステップ1：リスナーの設定

1. CLBコンソールにログインし、左側ナビゲーションバーの**インスタンス管理**をクリックします。
2. CLBインスタンスリストページの左上隅でリージョンを選択し、インスタンスリスト右側の操作列で**リスナーの設定**をクリックします。



3. HTTP/HTTPSリスナーで**新規作成**をクリックし、ポップアップした「リスナーの作成」ダイアログボックスでHTTPSリスナーの設定を行います。

3.1 リスナーの作成

リスナーの基本設定	説明	事例
名前	リスナーの名称です。	test-https-443
リスニングプロトコルポート	リスニングプロトコル：この例ではHTTPSを選択します。 リスニングポート：リクエストを受信してバックエンドサーバーにリクエストを転送するために使用するポートで、ポート範囲は1～65535です。 同一CLBインスタンス内で、リスニングポートは重複できません。	HTTPS:443
長時間接続の有効化	有効化すると、CLBとバックエンドサービス間で長時間接続を使用し、CLBはソースIPをパススルーしなくなりますので、XFFからソースIPを取	未有効化

	得してください。正常な転送を保証するため、CLB上でセキュリティグループを有効化してデフォルトで許可するか、またはCVMのセキュリティグループで100.127.0.0/16を許可してください。 **説明**：有効化すると、CLBとバックエンドサーバーの接続数の範囲はリクエストの[QPS、QPS*60]の間で変動し、具体的な数値は接続再利用率によって決まります。バックエンドサービスが接続数の上限に制限を設けている場合、有効化は慎重に行うことをお勧めします。この機能は現在ベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。	
back-to-originの有効化	SNIの有効化は、1つのリスナーの下でドメイン名ごとに異なる証明書を設定できることを意味します。SNIを有効化しないことは、このリスナーでは複数のドメイン名に同一の証明書を使用することを意味します。	未有効化
SSL解析方式	単方向認証および双方向認証をサポートしています。ロードバランサがSSLの暗号化と復号のオーバーヘッドを代行し、アクセスの安全性を保証します。	
サーバー証明書	SSL証明書プラットフォーム にすでにある証明書を選択するか、証明書を新規作成してアップロードできます。サーバー証明書は2つの証明書の設定をサポートしています。すなわち2種類の異なるタイプの暗号化アルゴリズムの証明書です。 **説明**：2つの証明書の設定は、CLBのみサポートしており、従来型CLBはサポートしていません。かつ2つの証明書の設定後は、QUIC機能の有効化をサポートしていません。	既存のものを選択します
CA証明書	SSL証明書プラットフォームにすでにある証明書を選択するか、証明書を新規作成してアップロードできます。	既存のものを選択します

3.2 転送ルールの作成

転送ルールの基本設定	説明	事例
ドメイン名	転送ドメイン名： 長さ制限：1～80文字です。 `_` で始めることはできません。 正確なドメイン名およびワイルドカードのドメイン名をサポートしています。 正規表現をサポートしています。 具体的な設定ルールです。詳細については、転送ドメイン名設定ルールをご参照ください。	www.example.com
デフォルトドメイン名	リスナーのすべてのドメイン名がマッチングに成功しなかった場合、システムはリクエストにデフォルトのアクセスドメイン名を指	オン

	定し、デフォルトアクセスを制御可能にします。 1つのリスナーの下に設定できるデフォルトドメイン名は1つだけです。	
HTTP 2.0	HTTP2.0を有効化すると、CLBはHTTP2.0のリクエストを受信できるようになります。クライアントがCLBをリクエストする際にどのHTTPバージョンを使用していても、CLBがバックエンドサーバーにアクセスする際のHTTPバージョンは常にHTTP 1.1となります。	オン
URLパス	転送URLパス： 長さ制限：1～200文字です。 正規表現をサポートしています。 具体的な設定ルールです。詳細については、転送URLパス設定ルールをご参照ください。	/index
バランシング方式	HTTPSリスナーでは、CLBは重み付けラウンドロビン（WRR）、重み付け最小接続（WLC）およびIP Hashの3種類のスケジューリングアルゴリズムをサポートしています。 重み付けラウンドロビンアルゴリズム：バックエンドサーバーの重みに基づき、順番にリクエストを異なるサーバーに配信します。重み付けラウンドロビンアルゴリズムは**新規接続数**に基づいてスケジューリングし、重みの高いサーバーがラウンドロビンされる回数（確率）が高くなるほど、同じ重みのサーバーは同じ数の接続数を処理します。 重み付け最小接続：サーバーの現在アクティブな接続数に基づいてサーバーの負荷状況を推定します。重み付け最小接続はサーバー負荷および重みに基づいて総合的にスケジューリングし、重み値が同じ場合、現在の接続数が少ないバックエンドサーバーほどラウンドロビンされる回数（確率）も高くなります。 IP Hash：リクエストのソースIPアドレスに応じて、ハッシュキー（Hash Key）を使用し、静的に割り当てられたハッシュテーブルから対応するサーバーを見つけます。そのサーバーが使用可能であり、かつオーバーロード状態ではない場合はリクエストがそのサーバーに送信され、そうではない場合は空が返されます。	重み付けラウンドロビン
バックエンドプロトコル	バックエンドプロトコルとは、CLBとバックエンドサービスとの間のプロトコルのことです。 バックエンドプロトコルとしてHTTPを選択した場合、バックエンドサービスはHTTPサービスをデプロイする必要があります。 バックエンドプロトコルとしてHTTPを選択した場合、バックエンドサービスはHTTPサービスをデプロイする必要があり、HTTPSサービスの暗号化/復号により、バックエンドサービスのリソース消費量がより多くなります。 バックエンドプロトコルとしてgRPCを選択した場合、バックエンドサービスはgRPCサービスをデプロイする必要があります。HTTP2.0	HTTP。

	が有効でQUICが無効になっている場合にのみ、バックエンドの転送プロトコルとしてgRPCの選択がサポートされます。	
クライアントIPを取得	デフォルトで有効	すでにオンです
Gzip圧縮	デフォルトで有効	すでにオンです

3.3 ヘルスチェック

ヘルスチェックの詳細については、HTTPSヘルスチェックをご参照ください。

3.4 セッション維持

セッション維持の設定	説明	事例
セッション維持の有効化/無効化	セッションの維持を有効化すると、CLBリスナーは同一クライアントからのアクセスリクエストを同一のバックエンドサーバーに配信します。 TCPプロトコルはクライアントIPアドレスのセッションの維持に基づき、同一IPアドレスからのアクセスリクエストを同一のバックエンドサーバーに転送します。 重み付けラウンドロビンスケジューリングはセッションの維持をサポートします。重み付け最小接続スケジューリングはセッション維持機能の有効化をサポートしていません。	オン
セッションの維持時間	維持時間を超え、接続中に新たなリクエストがない場合は、自動的にセッションの維持が切断されます。 設定可能範囲は30～3600秒です。	30s

ステップ2：バックエンドサーバーのバインド

1. 「リスナー管理」ページで、上記の `HTTPS:443` リスナーなどの、先ほど作成したリスナーをクリックし、左側の**+**をクリックしてドメイン名およびURLパスを表示します。具体的なURLパスを選択すると、リスナーの右側にそのパスにバインド済みのバックエンドサービスが表示されます。

2. **バインド**をクリックし、バインドしたいバックエンドサーバーをポップアップボックスから選択し、サービスポートと重みを設定します。

説明：

デフォルトポート機能：先に「デフォルトポート」を入力してからバックエンドサーバーを選択すると、それぞれのバックエンドサーバーのポートがすべてデフォルトポートとなります。

ステップ3：セキュリティグループ（オプション）

CLBのセキュリティグループを設定して、パブリックネットワークトラフィックの分離を行うことができます。詳細については、CLBセキュリティグループの設定をご参照ください。

ステップ4：リスナーの変更/削除（オプション）

作成したリスナーを変更または削除したい場合、「リスナー管理」ページで、作成したリスナーをクリックし、


アイコンをクリックして変更または


アイコンをクリックして削除してください。

バランシング方式

最終更新日：：2024-01-04 18:36:26

バランシング方式とは、CLBがバックエンドサーバーにトラフィックを分配する際のアルゴリズムであり、バランシング方式の違いによって異なる負荷分散効果を得ることができます。

重み付けラウンドロビンアルゴリズム

重み付けラウンドロビンアルゴリズム（Weighted Round-Robin Scheduling）は、ポーリングの方式によって、リクエストを順に異なるサーバーにスケジューリングするものです。重み付けラウンドロビンスケジューリングアルゴリズムでは、サーバー間でパフォーマンスに違いがある状態を解決することができます。サーバーの処理パフォーマンスをそれに応じた重みの値で表し、重みの高低とポーリングの方式によってリクエストを各サーバーに分配します。重み付けラウンドロビンアルゴリズムでは新規接続数に基づいてスケジューリングを行います。重みが高いサーバーは先に接続を確立することができ、重みが高いほどポーリングの回数が多く（確率が高く）なります。同じ重みのサーバーは同等の接続数を処理します。
メリット：シンプルで実用的であり、現時点のすべての接続ステータスを記録する必要がない、ステートレスなスケジューリングです。
デメリット：相対的にシンプルなため、リクエストのサービス時間の変化が大きい場合や、各リクエストの消費時間が一致しない場合に、サーバー間の負荷がアンバランスになりやすいです。
適用ケース：各リクエストがバックエンドを占有する時間が基本的に同じ場合に、負荷の状態が最適になります。HTTPなどの短時間接続サービスによく用いられます。
ユーザーへの推奨事項：各リクエストのバックエンド占有時間が基本的に同じであり、バックエンドサーバーが処理するリクエストタイプが同一または類似している場合は、重み付けラウンドロビン方式を選択することをお勧めします。リクエスト時間の差があまりない場合も重み付けラウンドロビン方式を使用することをお勧めします。この実現方式は低消費かつトラバーサルの必要がなく、高効率なためです。

重み付け最小接続アルゴリズム

実際の状況では、クライアントのサービスリクエストがサーバーにとどまる時間には比較的大きな差異があります。シンプルなポーリングやランダムなバランシングアルゴリズムを用いた場合、動作時間が長くなるに従って、各サーバー上の接続プロセス数に大きな違いが生じるようになり、負荷分散の真の効果が得られなくなる可能性があります。
最小接続スケジューリングは一種の動的スケジューリングアルゴリズムであり、ラウンドロビンスケジューリングアルゴリズムと異なり、サーバーのその時点でアクティブな接続数によってサーバーの負荷状況を推測します。スケジューラーが各サーバーの確立した接続数を記録する必要があり、あるサーバーにリクエストがスケジュー

リングされると接続数に1をプラスし、接続が中断またはタイムアウトになると、接続数から1をマイナスします。
重み付け最小接続アルゴリズム（Weighted Least-Connection Scheduling）は最小接続スケジューリングアルゴリズムをベースに、サーバーの処理能力に応じて各サーバーに異なる重みを割り当て、各サーバーがその重みに応じた数のサービスリクエストを受け付けることができるようにするもので、最小接続スケジューリングアルゴリズムをベースに改善を加えたものです。
説明：
仮に各バックエンドサーバーの重みを順にwiとし、現在の接続数を順にciとした場合、ci/wiを順に計算し、値が最小のバックエンドサーバーインスタンスを、次に割り当てるインスタンスにします。ci/wiが同一のバックエンドサーバーインスタンスが存在する場合は、さらに重み付けラウンドロビン方式でスケジューリングを行います。
メリット：このアルゴリズムは、FTPなどのアプリケーションのような、処理時間の長いリクエストサービスに適しています。
デメリット：ポートの制限により、現在最小接続とセッション維持機能を同時に有効にすることはできません。
適用ケース：各リクエストがバックエンドを占有する時間の差が比較的大きいケースです。長時間接続サービスによく用いられます。
ユーザーへの推奨事項：ユーザーがさまざまなリクエストを処理する必要があり、かつリクエストがバックエンドを占有する時間の差が比較的大きい場合（例えば3ミリ秒と3秒のように、単位レベルの違いがある場合など）では、重み付け最小接続アルゴリズムを使用して負荷分散を実現することをお勧めします。

ソースIPハッシュスケジューリングアルゴリズム

ソースIPハッシュスケジューリングアルゴリズム（ip_hash）ではリクエストのソースIPアドレスに応じて、ハッシュキー（Hash Key）を使用し、静的に割り当てられたハッシュテーブルから対応するサーバーを見つけます。そのサーバーが使用可能であり、かつオーバーロード状態ではない場合はリクエストがそのサーバーに送信され、そうではない場合は空が返されます。
メリット：あるクライアントのリクエストを、ハッシュテーブルによって同一のバックエンドサーバー上に一貫してマッピングできるため、セッション維持がサポートされていないシーンでも、ip_hashを使用してシンプルなセッション維持を実現することができます。
ユーザーへの推奨事項：リクエストのソースアドレスに対しハッシュ計算を行い、設定したバックエンドサーバーの重みに応じて、リクエストをマッチしたサーバーに転送することで、同一のクライアントIPのリクエストが常に特定のサーバーに転送されるようになります。この方式はCookie機能のないプロトコルに適しています。

バランシングアルゴリズムの選択と重みの設定

ユーザーのバックエンドサーバークラスターがさまざまなシナリオの下で安定して業務を担うことができるよう、CLBの選択と重みの設定について、シナリオごとの例を参考までにご紹介します。

シナリオ1：
1.1 同一の設定（CPU/メモリ）のバックエンドサーバーが3台あるとします。パフォーマンスが同じであるため、バックエンドサーバーの重みはすべて10に設定することができます。
1.2 現在、各バックエンドサーバーとクライアントの間に100のTCP接続を確立しており、さらにバックエンドサーバーを1台増設します。
1.3 このシナリオでは、最小接続のバランシング方式を使用して速やかに4台目のバックエンドサーバーの負荷を増大させ、他の3台のバックエンドサーバーの負荷を低減することをお勧めします。
シナリオ2：
1.1 クラウドサービスを初めて使用する場合で、なおかつウェブサイト構築からあまり時間が経っておらず、サイトの負荷が比較的小さい場合は、同一の設定のバックエンドサーバーの購入をお勧めします。この場合、バックエンドサーバーはすべて同一のアクセス層サーバーとなります。
1.2 このシナリオでは、バックエンドサーバーの重みをすべてデフォルト値の10に設定し、重み付けラウンドロビンのバランシング方式によってトラフィックを振り分けることができます。
シナリオ3：
1.1 単純な静的ウェブサイトへのアクセスを担う5台のサーバーがあり、なおかつ5台のサーバーのコンピューティング能力の比率が9：3：3：3：1（CPU、メモリ換算）であるとします。
1.2 このシナリオでは、バックエンドサーバーの重みの割合を、順に90、30、30、30、10に設定することができます。静的ウェブサイトへのアクセスの大半は短時間接続のリクエストであるため、重み付けラウンドロビンのバランシング方式を使用して、バックエンドサーバーのパフォーマンス比率に従ってCLBインスタンスにリクエストを分配させることができます。
シナリオ4：
1.1 大量のWebアクセスリクエストの処理を担う10台のバックエンドサーバーがあり、なおかつバックエンドサーバー増設のための追加支出を望まないものの、あるバックエンドサーバーはオーバーロードのために頻繁に再起動が発生しているとします。
1.2 このシナリオでは、バックエンドサーバーのパフォーマンスに応じた重みを設定し、オーバーロードになっているバックエンドサーバーに低い重みを設定することをお勧めします。また、最小接続のロードバランシング方式を用いて、リクエストをアクティブ接続数が比較的少ないバックエンドサーバーに分配することで、問題のバックエンドサーバーのオーバーロードを解決することもできます。

シナリオ5：
1.1 いくつかの長時間接続リクエストの処理に用いる3台のバックエンドサーバーがあり、なおかつ3台のサーバーのコンピューティング能力の比率が3：1：1（CPU、メモリ換算）であるとします。
1.2 この場合、パフォーマンスが最も良好なサーバーが多くのリクエストを処理しますが、このサーバーがオーバーロードにならないように、新しいリクエストをアイドル状態のサーバーに分配したいとします。
1.3 このシナリオでは、最小接続のバランシング方式を使用し、かつビジーなサーバーの重みを適宜低下させることで、CLBがリクエストをアクティブ接続数の比較的少ないバックエンドサーバーに分配し、負荷分散を実現できるようにすることが可能です。
シナリオ6：

1.1 後続のクライアントリクエストを同一のサーバー上に分配したいとします。この場合、重み付けラウンドロビンまたは重み付け最小接続の方式を用いると、同一のクライアントからのリクエストを固定のサーバーに転送することが保証されません。
1.2 特定のアプリケーションサーバーのニーズに合わせるため、クライアントのセッションの「粘着性」あるいは「継続性」を保証します。このシナリオでは、ip_hashのバランシング方式を用いてトラフィックを振り分け、同一のクライアントからのリクエストが常に同一のバックエンドサーバーに振り分けられるようにすることができます（サーバー数に変化があった場合またはこのサーバーが使用不能になった場合を除きます）。

セッションの維持

最終更新日：：2024-01-04 18:36:26

セッション維持は、同一のIPからのリクエストが同一のバックエンドサーバーに転送されることを可能にする機能です。デフォルトでは、CLBは各リクエストをそれぞれ異なるバックエンドサーバーインスタンスにルーティングしますが、セッション維持機能を使用することで、特定のユーザーからのリクエストを同一のバックエンドサーバーインスタンス上にルーティングすることが可能になります。こうすることで、セッションを維持する必要があるアプリケーション（ショッピングカートなど）を正しく動作させることができます。

レイヤー4セッション維持

レイヤー4プロトコル（TCP/UDP）はソースIPベースのセッション維持機能をサポートしています。セッション維持時間は30～3600秒の間の任意の整数値を設定でき、この時間閾値を超過すると、セッション中に新しいリクエストがなければセッション維持状態が中断されます。セッション維持とバランシング方式の関連は次のとおりです。
バランシング方式が「重み付けラウンドロビン」の場合は、バックエンドサーバーの重みに応じてリクエストが振り分けられ、ソースIPベースのセッション維持がサポートされます。
バランシング方式が「重み付け最小接続」の場合は、サーバーの負荷と重みに応じて総合的にスケジューリングされ、セッション維持はサポートされません。

レイヤー7セッション維持

レイヤー7プロトコル（HTTP/HTTPS）はCookie挿入ベースのセッション維持機能（ロードバランサがクライアントにCookieを埋め込む）をサポートしています。セッション維持時間は30～3600秒の間で設定できます。セッション維持とバランシング方式の関連は次のとおりです。
バランシング方式が「重み付けラウンドロビン」の場合は、バックエンドサーバーの重みに応じてリクエストが振り分けられ、Cookie挿入ベースのセッション維持がサポートされます。
バランシング方式が「重み付け最小接続」の場合は、サーバーの負荷と重みに応じて総合的にスケジューリングされ、セッション維持はサポートされません。
バランシング方式が「IP Hash」の場合は、ソースIPベースのセッション維持がサポートされ、Cookie挿入ベースのセッション維持はサポートされません。

接続タイムアウト時間

現在、HTTP接続タイムアウト時間（keepalive_timeout）はデフォルトで75秒です。調整をご希望の場合はカスタム設定をアクティブ化してください。この時間閾値を超過すると、セッション中にデータ通信が行われなければ接続が切断されます。
現在、TCP接続のタイムアウト時間は調整できず、デフォルトで900秒となっています。この時間閾値を超過すると、セッション中にデータ通信が行われなければ接続が切断されます。

セッション維持の設定

1. CLBコンソールにログインし、セッション維持の設定を行いたいCLBインスタンスIDをクリックしてCLB詳細ページに進みます。
2. **リスナー管理**タブを選択します。
3. セッション維持の設定を行いたいCLBリスナーの後方の**変更**をクリックします。
4. セッション維持機能を有効にするかどうかを選択し、ボタンをクリックして有効化し、維持時間を入力して**OK**をクリックします。

長時間接続とセッション維持の関係

シナリオ1：HTTPレイヤー7業務

Clientからのアクセスが HTTP/1.1プロトコルであり、ヘッダー情報にConnection:keep-aliveが設定されているとします。CLBを介してさらにバックエンドサーバーにアクセスし、このときセッション維持を有効にしていなかった場合、次のアクセスの際に同一のサーバーにアクセスすることはできますか。
回答：できません。
まず、HTTP keep-aliveとはTCP接続が送信後も有効な状態を維持することで、ブラウザが引き続き同一の接続によってリクエストを送信できることを指します。接続を維持することで、各リクエストが新たに接続を確立するのにかかる時間が節約でき、帯域幅の節約にもなります。CLBクラスターのデフォルトのタイムアウト時間は75秒です（75秒以内に新しいリクエストが更新されなかった場合、デフォルトでTCP接続を切断します）。
HTTP keep-aliveはClient側がCLBとの間で確立するものであり、このときCookieによるセッション維持が有効になっていなければ、次のアクセスの際、CLBはラウンドロビンポリシーに基づいて1台のバックエンドサーバーをランダムに選択し、それまでの長時間接続は無効になります。
このため、セッション維持を有効化しておくことをお勧めします。
Cookieによるセッション維持時間を1000秒に設定した場合、Client側は再度リクエストを送信します。前回のリクエストから75秒以上経過しているため、TCPの接続を再度確立する必要があります。アプリケーション層はCookieを判断し、同一のバックエンドサーバーを見つけるため、Clientがアクセスするサーバーは前回アクセスしたものと同じになります。

シナリオ2：TCPレイヤー4業務

Clientがアクセスを開始し、トランスポート層プロトコルがTCPであり、長時間接続を有効にしているとします。ただし、ソースIPベースのセッション維持は有効にしていません。この場合、次のアクセスの際に、同一のClientが同一のマシンにアクセスすることはできますか。

回答：場合によります。

まず、レイヤー4の実現メカニズムにより、TCPが長時間接続を有効にしている場合、この長時間接続が切断されなければ、連続した2回のアクセスはどちらも同じ接続となり、同一のマシンにアクセスすることができます。2回目のアクセスの際に、最初の接続が他の原因（ネットワークの再起動、接続タイムアウト）によってリリースされた場合、2回目のアクセスは他のバックエンドサーバーにスケジューリングされる可能性があります。また、長時間接続はデフォルトのグローバルタイムアウト時間が900秒であり、新しいリクエストがなければリリースされます。

長時間接続の有効化する方法についてはHTTPリスナーの設定およびHTTPSリスナーの設定をご参照ください。

レイヤー7リダイレクト設定

最終更新日：：2024-01-04 18:36:26

CLBはレイヤー7リダイレクトをサポートしています。この機能はユーザーのレイヤー7HTTP/HTTPSリスナー上でのリダイレクト設定をサポートします。

説明：

セッション維持：クライアントが `example.com/bbs/test/123.html` にアクセスし、かつバックエンドCVMがセッション維持を有効にしている場合、リダイレクトを有効化すると、トラフィックを `example.com/bbs/test/456.html` にリダイレクトした際、元のセッション維持メカニズムは無効になります。

TCP/UDPリダイレクト：現在はIP + Portレベルのリダイレクトはサポートしていません。今後のバージョンで提供される予定です。

リダイレクトの概要

自動リダイレクト

概要

すでに存在する `HTTPS:443` リスナーに、システムが自動的にHTTPリスナーを作成して転送を行います。デフォルトでは80番ポートを使用します。作成に成功すると、 `HTTP:80` アドレスから `HTTPS:443` アドレスに自動的にリダイレクトしてアクセスすることができます。

ユースケース

強制HTTPSリダイレクト、すなわちHTTPからHTTPSへの強制転送です。PC、スマホブラウザなどがHTTPリクエストによってWebサービスにアクセスしようとする場合、CLBはすべての `HTTP:80` リクエストを `HTTPS:443` にリダイレクトして転送を行います。

ソリューションの優位性

設定は一度のみ：1つのドメイン名、一度の設定で強制HTTPSリダイレクトが完了します。

アップデートに便利：HTTPSサービスのURLに増減があった場合も、コンソールでこの機能を再度使用して更新するだけで済みます。

手動リダイレクト

概要

1対1リダイレクトの設定が可能です。例えばあるCLBインスタンスで、 `リスナー1/ドメイン名1/URL1` を `リスナー2/ドメイン名2/URL2` にリダイレクトするよう設定できます。

説明：

ドメイン名にすでに自動リダイレクトを設定したことがある場合は、手動リダイレクトを設定することはできません。

ユースケース
単一パスのリダイレクトです。例えばWeb業務を一時的にオフラインにする必要がある場合（ECでの完売、ページメンテナンス、更新・アップグレードなど）、従来のページを新しいページにリダイレクトする必要があります。リダイレクトを行わなければ、ユーザーのお気に入りや検索エンジンデータベース内の古いアドレスにアクセスすると `404/503` エラーページが表示されるだけになり、ユーザー体験を低下させ、アクセス数が無駄に失われることになります。

自動リダイレクト

Tencent Cloud CLBはワンクリックでのHTTPからHTTPSへの強制リダイレクトをサポートしています。
開発者がウェブサイト `https://www.example.com` を設定したいとします。開発者は、ユーザーがブラウザにURLを入力する際、それがHTTPリクエスト（ `http://www.example.com` ）かHTTPSリクエスト（ `https://www.example.com` ）のどちらであっても、HTTPSプロトコルによってセキュアにアクセスできるようにしたいと考えています。

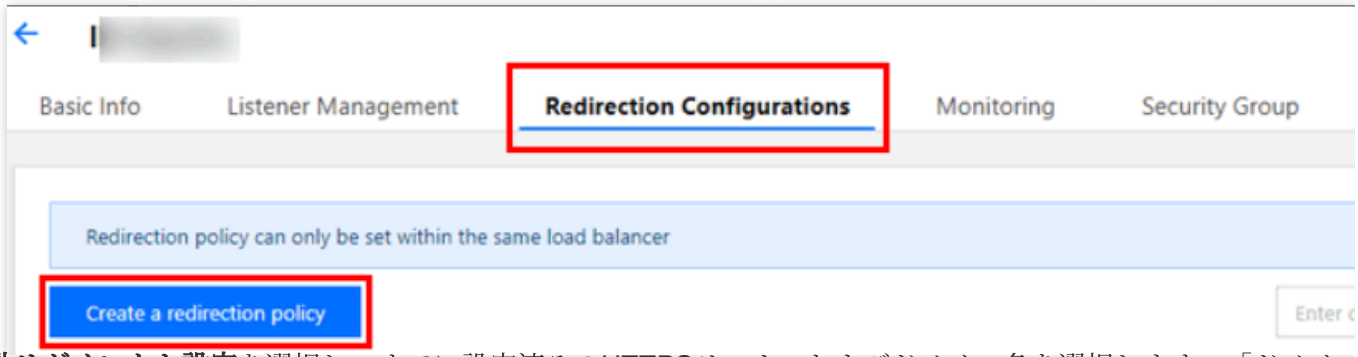
前提条件

`HTTPS:443` リスナーが設定済みであること。

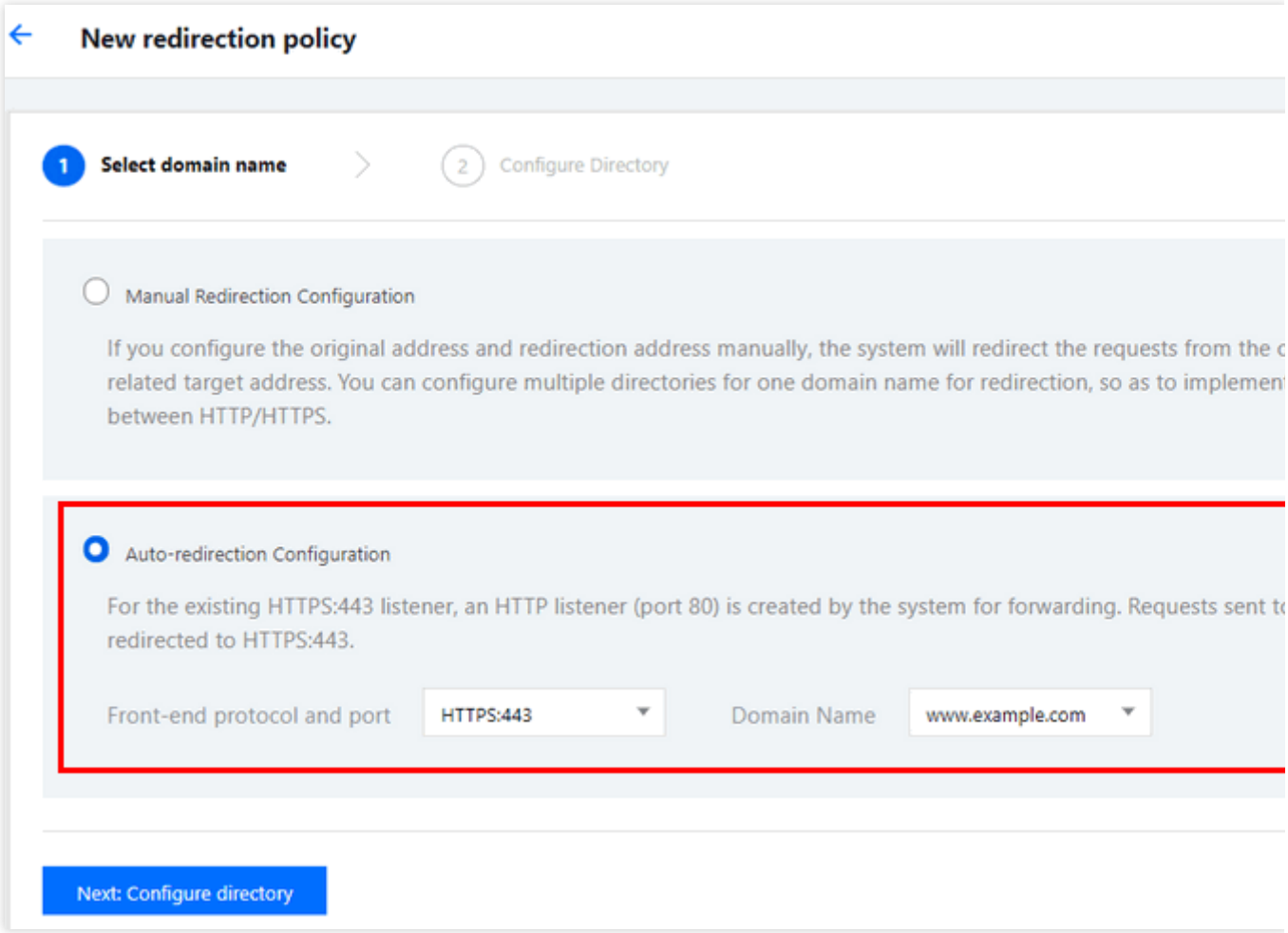
操作手順

1. Tencent Cloud CLBコンソールにログインし、CLBのHTTPSリスナーの設定を完了し、 `https://example.com` のWeb環境を構築してください。詳細については、HTTPSリスナーの設定をご参照ください。
2. HTTPSリスナー設定完了後の結果は下図のとおりです。

3. CLBインスタンス詳細の「リダイレクト設定」タブで、**リダイレクト設定の新規作成**をクリックします。



4. **自動リダイレクト設定**を選択し、すでに設定済みのHTTPSリスナーおよびドメイン名を選択します。「ドメイン名の設定」でリダイレクトステータスコードを選び、**送信**をクリックすれば設定が完了します。

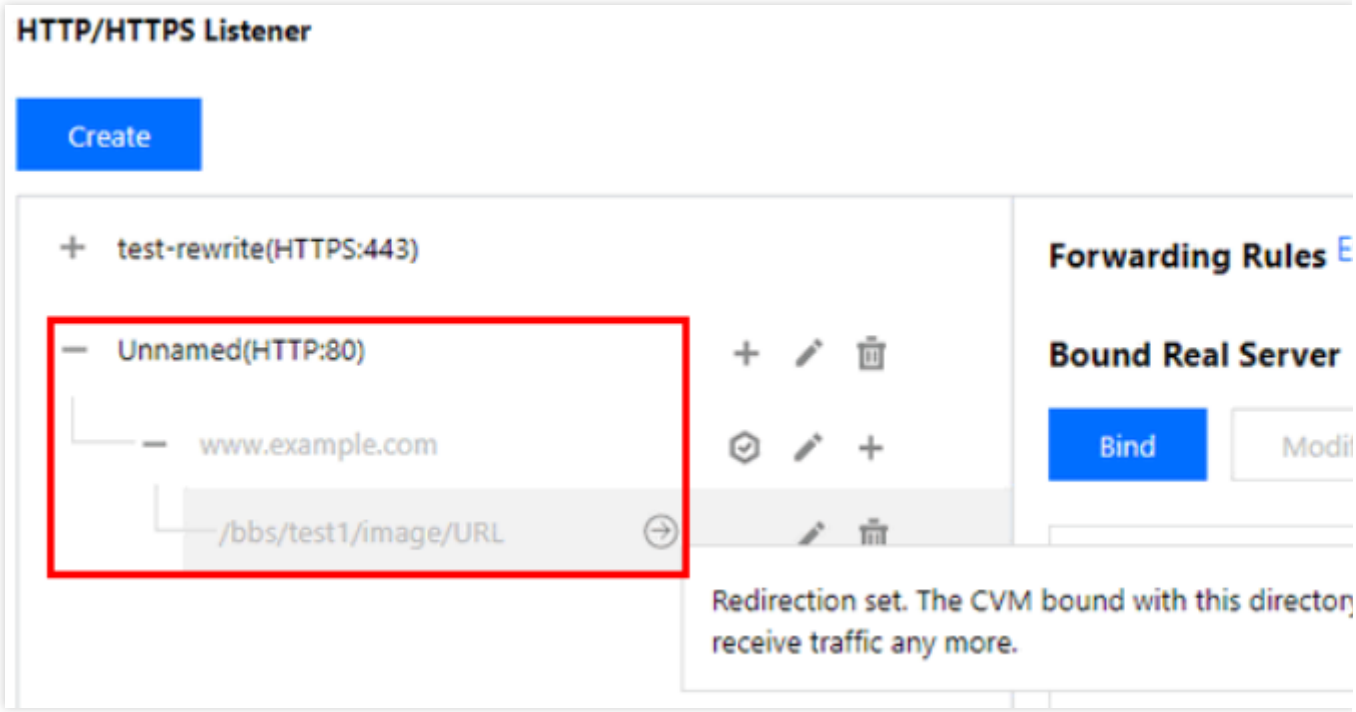


説明：

リダイレクトの中の「ドメイン名の設定」機能は、現在ベータ版テスト段階です。利用される場合はチケットを提出してください。

ステータスコード301（Moved Permanently）、302（Move Temporarily）、307（Temporary Redirect）についての詳しい内容は、 HTTP / 1.1標準（RFC 7231）をご参照ください。

5. リダイレクト設定が完了すると、結果は下図のようになります。 `HTTPS:443` リスナーに `HTTP:80` リスナーが自動的に設定され、なおかつHTTPのトラフィックがすべて自動的にHTTPSにリダイレクトされるようになっていることが確認できます。



手動リダイレクト

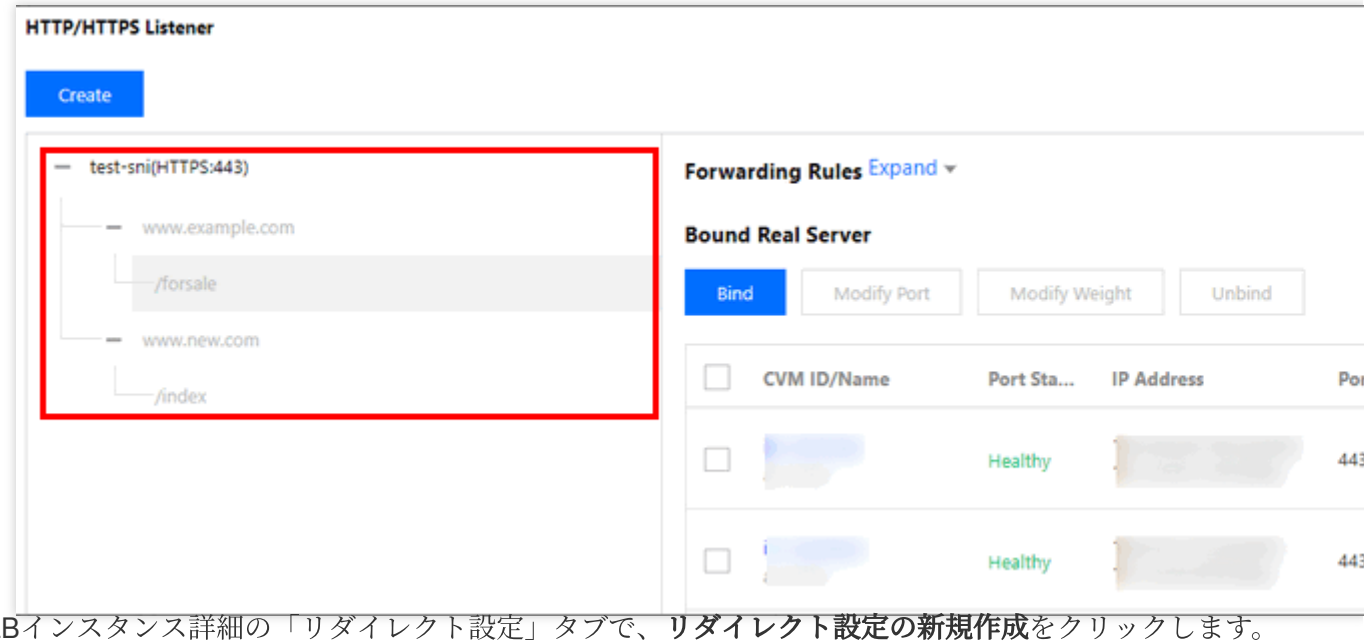
Tencent Cloud CLBは1対1リダイレクトの設定をサポートしています。
例えば、業務でforsaleページを使用してキャンペーン運営を行い、現在のキャンペーンが終了すればキャンペーンページ `https://www.example.com/forsale` を新たなホームページ `https://www.new.com/index` にリダイレクトする必要がある場合などです。

前提条件

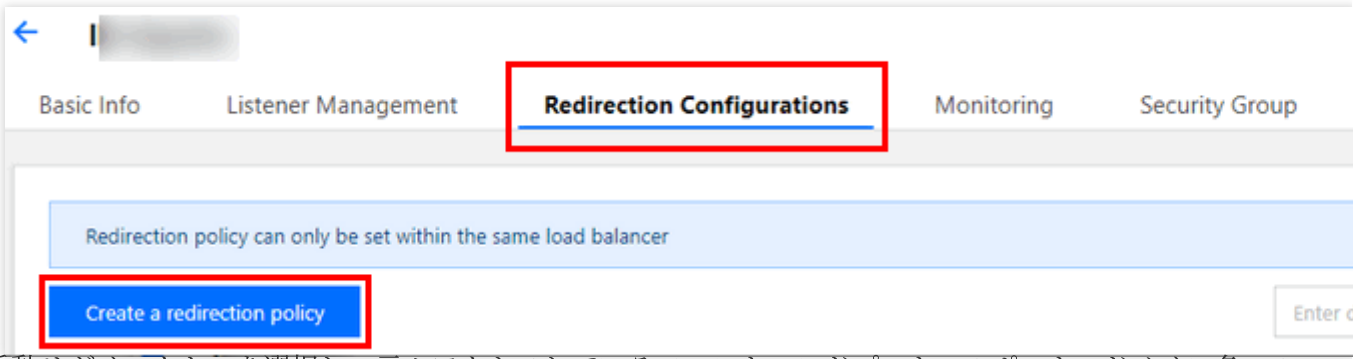
HTTPSリスナーが設定済みであること。
転送ドメイン名 `https://www.example.com/forsale` が設定済みであること。
転送ドメイン名およびパス `https://www.new.com/index` が設定済みであること。

操作手順

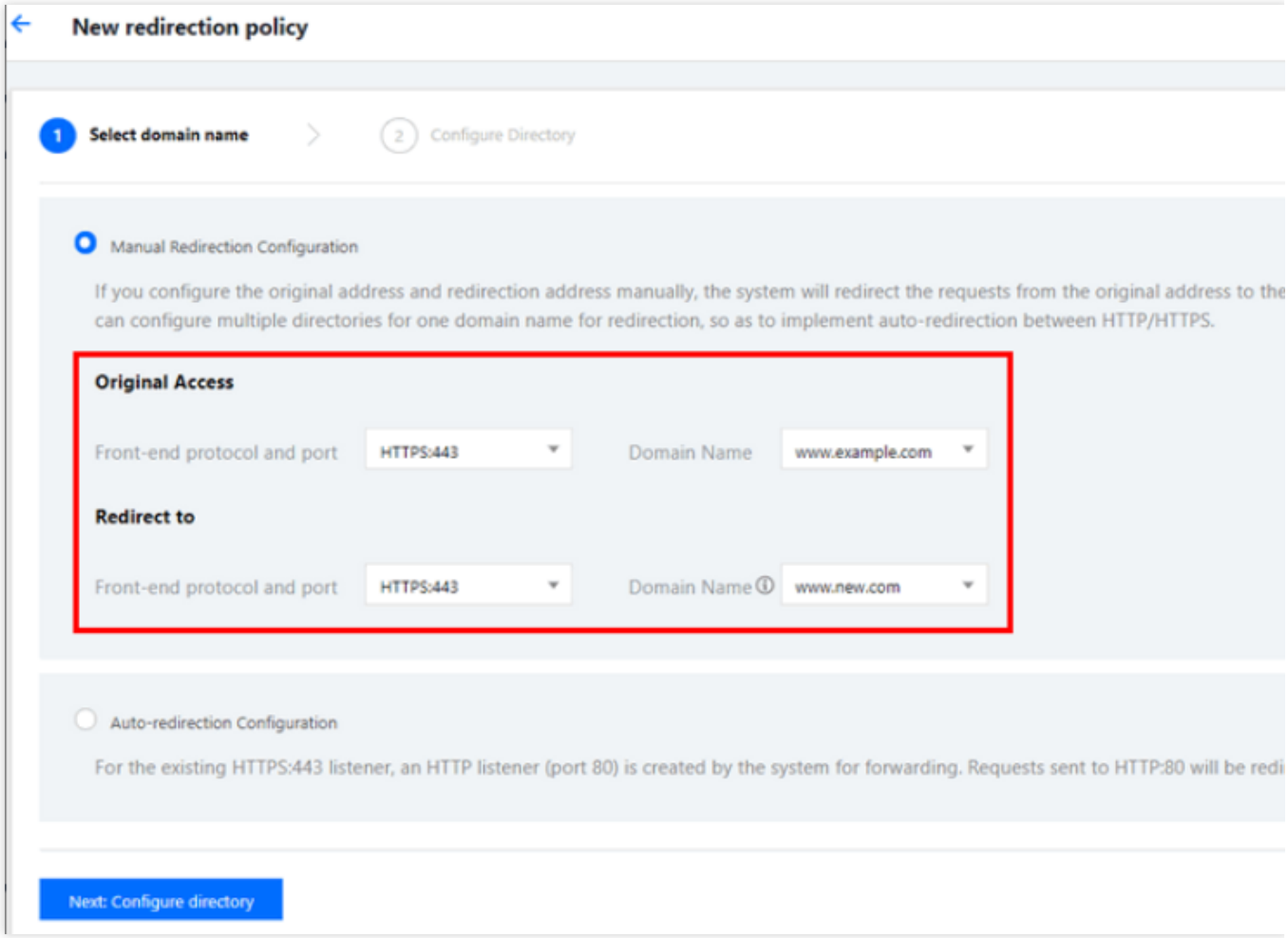
1. Tencent Cloud CLBコンソールにログインし、CLBのHTTPSリスナーの設定を完了し、 `https://example.com` のWeb環境を構築してください。詳細については、HTTPSリスナーの設定をご参照ください。
2. HTTPS設定完了後の結果は下図のとおりです。



3. CLBインスタンス詳細の「リダイレクト設定」タブで、**リダイレクト設定の新規作成**をクリックします。



4. 「手動リダイレクト」を選択し、元々アクセスしているフロントエンドプロトコルポート、ドメイン名、ルートを選びます。それから、リダイレクト後のフロントエンドプロトコルポート、ドメイン名、ルートを選び、「ドメイン名の設定」でリダイレクトのステータスコードを選択します。URLを保留するか、保留しないかを選択し、「送信」をクリックすると設定が完了します。

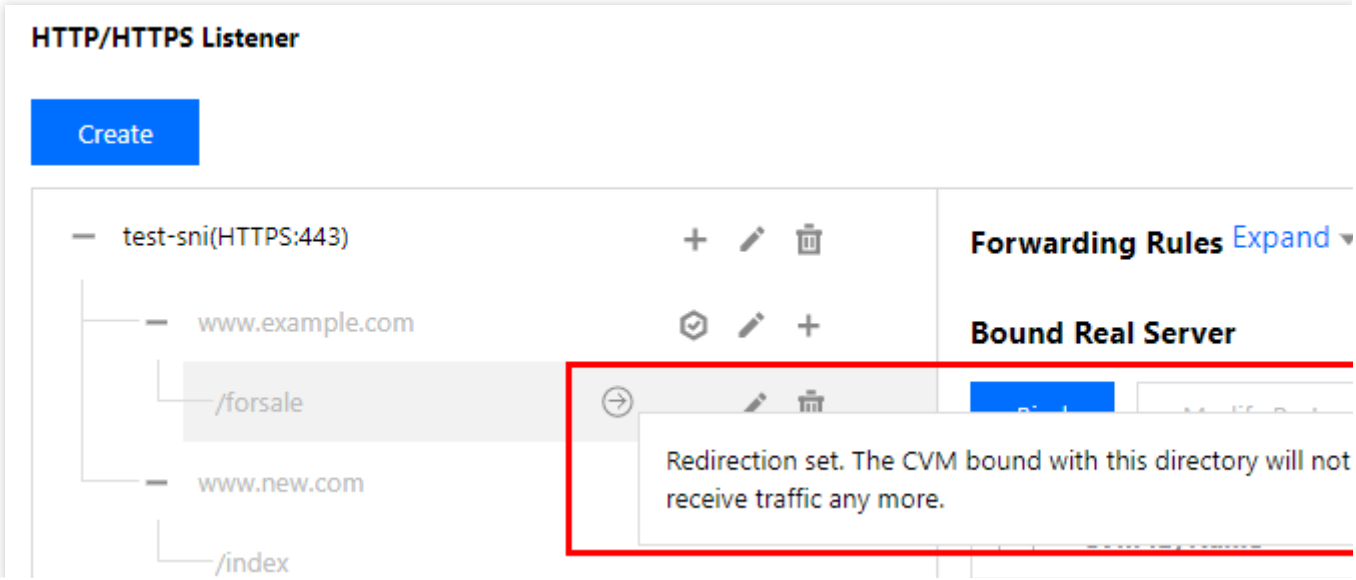


説明：

リダイレクトの中の「ドメイン名の設定」機能は、現在ベータ版テスト段階です。利用される場合はチケットを提出してください。

ステータスコード301（Moved Permanently）、302（Move Temporarily）、307（Temporary Redirect）についての詳しい内容は、 HTTP / 1.1標準（RFC 7231）をご参照ください。

5. リダイレクト設定が完了すると、結果は下図のようになります。 `HTTPS:443` リスナーで、 `https://www.example.com/forsale` が `https://www.new.com/index` にリダイレクトされるようになっていることが確認できます。



レイヤー7カスタム設定

最終更新日：：2024-01-29 15:55:11

CLBはカスタム設定機能をサポートしており、client_max_body_size、ssl_protocolsなどの個別のCLBインスタンスの設定パラメータをユーザーが設定でき、カスタム設定のニーズを満たすことができます。

説明：

カスタム設定の個数は各リージョンにつき200までとなります。

カスタム設定の長さは64kまでとなります。

現在は1つのインスタンスにバインドできるカスタム設定は1つだけです。

カスタム設定は、CLB（旧「アプリケーション型CLB」）のレイヤー7HTTP/HTTPSリスナーについてのみ有効です。

CLBカスタム設定パラメータの説明

現在CLBのカスタム設定では次のフィールドをサポートしています。

フィールド設定	デフォルト値/推奨値	パラメータ範囲	説明
ssl_protocols	デフォルト値：TLSv1、TLSv1.1、TLSv1.2 推奨値：TLSv1.2、TLSv1.3	TLSv1 TLSv1.1 TLSv1.2 TLSv1.3	TLSプロトコルのバージョンを使用しました。
ssl_ciphers	ssl_ciphersデフォルト値	ssl_ciphersパラメータ範囲	暗号スイートです。
client_header_timeout	60s	[30-120]s	Clientリクエストヘッダーのタイムアウト時間を取得し、タイムアウトになると408を返します。
client_header_buffer_size	4k	[1-256]k	Clientリクエストヘッダーを格納するためのデフォルトのBufferサイズです。
client_body_timeout	60s	[30-120]s	ClientリクエストBodyを取得する際のタイムアウト時間です。Body全体の取

			得にかかる持続時間ではなく、一定時間データ伝送のないアイドル状態となった場合のタイムアウト時間を指します。タイムアウトになると408を返します。
client_max_body_size	60M	[1-10240]M	デフォルトの設定範囲は1M-256Mで、直接設定できます。 最大で10240M、つまり10Gをサポートしています。client_max_body_sizeの設定範囲が256Mより大きい場合、proxy_request_bufferingの値をoffに設定する必要があります。
keepalive_timeout	75s	[0-900]s	Client-Serverの長時間接続維持時間です。0に設定すると、長時間接続が無効になります。900sより長く設定したい場合は、チケット申請を提出してください。最大3600sまで設定可能です。
add_header	ユーザーカスタムの追加	-	特定のヘッダーフィールドをクライアントに返します。形式はadd_header xxx yyyです。 例えばクロスドメインのケースでは、`add_header Access-Control-Allow-Methods 'POST, OPTIONS'; add_header Access-Control-Allow-Origin *;` のように設定することができます。
more_set_headers	ユーザーカスタムの追加	-	特定のヘッダーフィールドをクライアントに返します。形式はmore_set_headers "A:B"です。
proxy_connect_timeout	4s	[4-120]s	upstreamバックエンドの接続タイムアウト時間です。
proxy_read_timeout	60s	[30-3600]s	upstreamバックエンドのレスポンスタイムアウト時間を読み取ります。
proxy_send_timeout	60s	[30-3600]s	upstreamバックエンドにリクエストを送信する際のタイムアウト時間です。
server_tokens	on	on,off	onはバージョン情報を表示することを意味します。

			offはバージョン情報を非表示にすることを意味します。
keepalive_requests	100	[1-10000]	Client-Serverの長い接続で送信できるリクエストの最大数です。
proxy_buffer_size	4k	[1-32]k	Serverのレスポンスヘッダーのサイズです。デフォルトではproxy_bufferで設定した単独のバッファサイズとなります。proxy_buffer_sizeを設定する場合は、proxy_buffersも同時に設定する必要があります。
proxy_buffers	8 4k	[3-8] [4-16]k	バッファ数とバッファサイズです。
proxy_request_buffering	off	on,off	onはクライアントリクエストボディをキャッシュすることを意味します。CLBはリクエストをキャッシュし、すべてのリクエストを受信した後、バックエンドCVMにチャンクで転送します。 offはクライアントリクエストボディをキャッシュしないことを意味します。CLBがリクエストを受信すると、すぐにバックエンドCVMに転送します。この際、バックエンドCVMのパフォーマンスに一定のプレッシャーが生じます。
proxy_set_header	X-Real-Port $remote_port	X-Real-Port $remote_port X-clb-lbid $lbid Stgw-request-id $stgw_request_id X-Forwarded-Port $vport X-Method $request_method X-Uri $uri	X-Real-Port $remote_portは、クライアントポートを意味します。 X-clb-lbid $lbidは、CLBインスタンスの識別子であるCLBのLBIDを意味します。 Stgw-request-id $stgw_request_idは、リクエストID（CLB内部で使用）を意味します。 X-Forwarded-Portは、CLBリスナーのポートを意味します。 X-Methodは、クライアントのリクエスト方法を意味します。 X-Uriは、クライアントのリクエストパスURIを意味します。
send_timeout	60s	[1-3600]s	サーバーからクライアントへのデータ伝送する際のタイムアウト時間です。

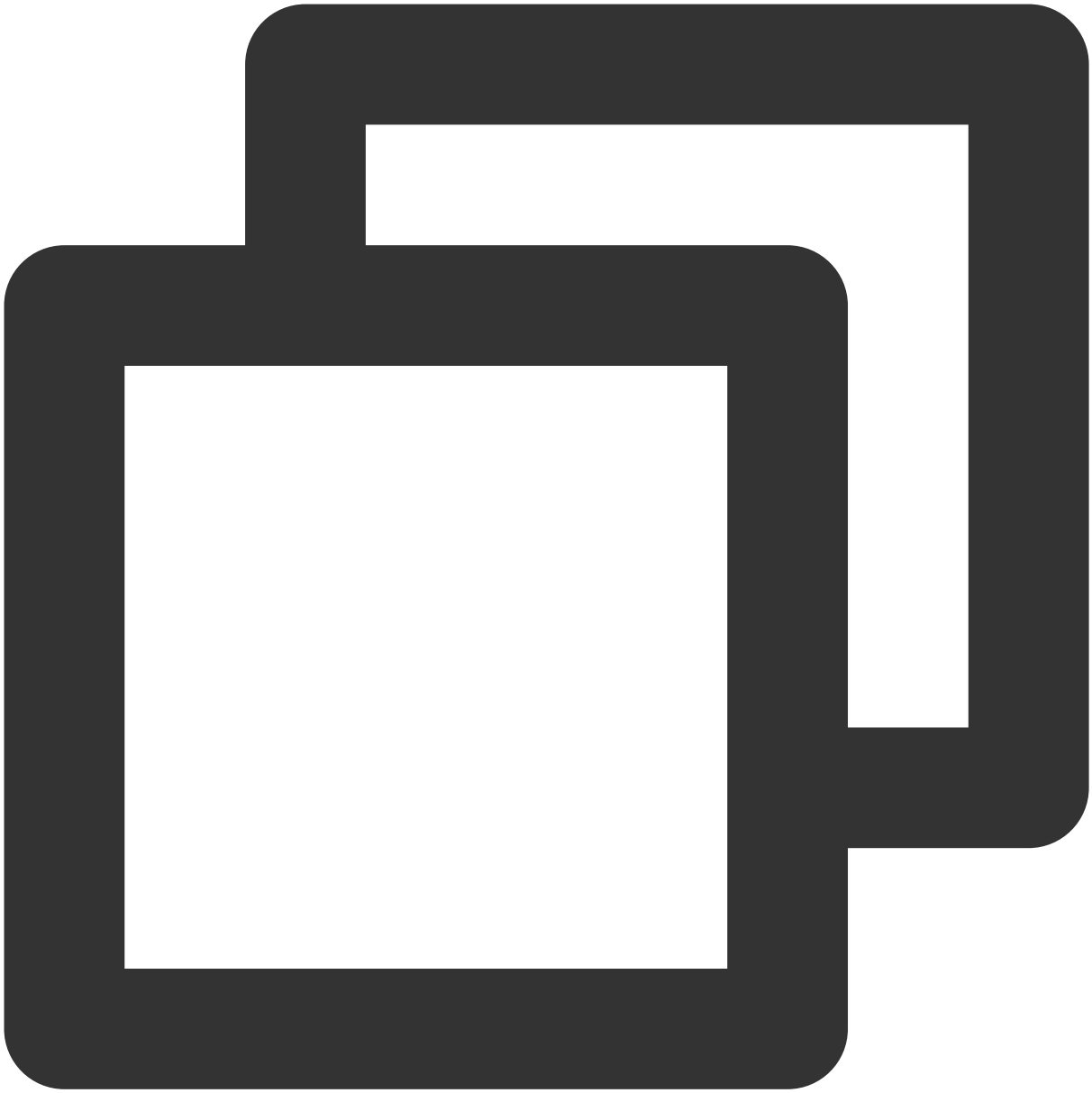
			連続した2回のデータ送信の間隔であり、リクエスト全体の伝送時間ではありません。
ssl_verify_depth	1	[1,10]	クライアント証明書チェーンの検証深度を設定します。
proxy_redirect	http:// https://	http:// https://	アップストリームサーバーが返すレスポンスがリダイレクトやリフレッシュのリクエストである場合（HTTPレスポンスコードが301または302の場合）、proxy_redirectはHTTPヘッダーのLocationまたはRefreshフィールド内のhttpをhttpsに再設定し、安全なリダイレクトを実現します。
ssl_early_data	off	on,off	TLS 1.3 0-RTTを有効化または無効化します。ssl_protocolsフィールドの値にTLSv1.3が含まれる場合のみ、ssl_early_dataをオンにすると有効になります。**ssl_early_dataをオンにするとリプレイアタックを受けるリスクがありますので、慎重に行ってください。**
http2_max_field_size	4k	[1-256]k	HPACK圧縮を行うリクエストヘッダーフィールドの最大サイズ(Size)を制限します。
proxy_intercept_errors	off	on，off	error_pageを設定する時に、事前にproxy_intercept_errorsをonに設定する必要があります。
error_page	-	error_page code [= [response]] uri	特定のエラーコード(Code)が発生した場合、あらかじめ定義したURIを表示することができます。デフォルトのステータスコード(Response)は302です。URIは必ず `/` で始まるパスでなければなりません。error_pageを設定する時に、事前にproxy_intercept_errorsをonに設定する必要があります。
proxy_ignore_client_abort	off	on,off	クライアントがレスポンス結果を待たずにCLBとの接続を自主的に切断する場合、CLBとバックエンドサーバー間

			の接続を中断するかどうかを設定します。

説明：
このうち、proxy_buffer_sizeとproxy_buffersの設定の値は制約条件である、2 * *max（proxy_buffer_size, proxy_buffers.size) ≤（proxy_buffers.num - 1）** proxy_buffers.sizeを満たす必要があります。例えば、proxy_buffer_sizeが24k、proxy_buffersが8 8kの場合、2 * *24k = 48k*、*（8 - 1）* 8k = 56kとなり、このとき48k ≤ 56kであるため、エラーは発生しません。これを満たさない場合はエラーが発生します。

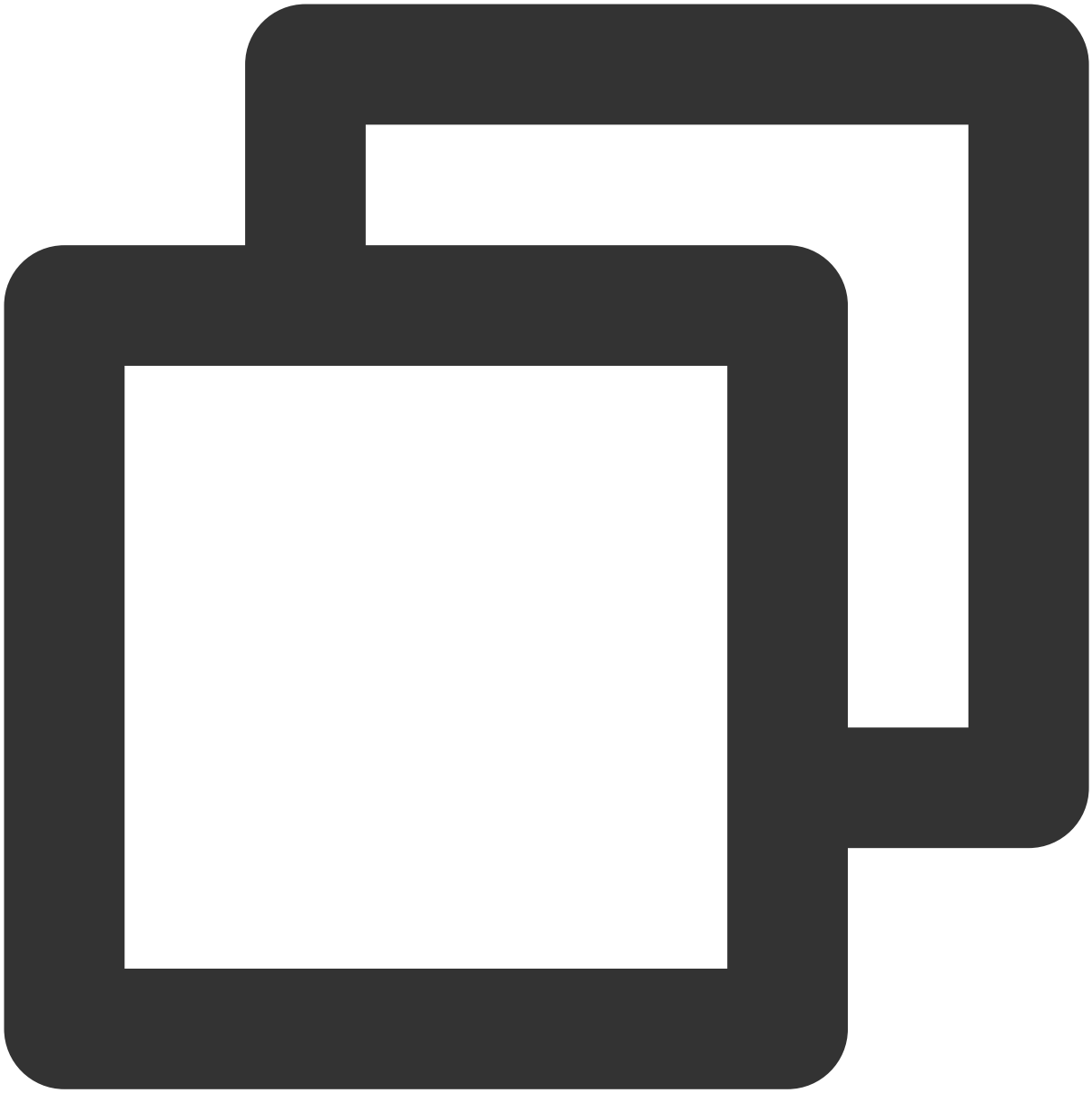
ssl_ciphers設定の説明

ssl_ciphers暗号スイートを設定する際、形式はOpenSSLで使用する形式と一致させる必要があります。アルゴリズムリストは1つまたは複数の `<cipher strings>` とし、複数のアルゴリズムの間は「:」で区切ります。ALLはすべてのアルゴリズムを表し、「!」はこのアルゴリズムが有効になっていないことを表します。「+」はこのアルゴリズムの配置順を最後にすることを表します。
デフォルトで強制的に無効化される暗号化アルゴリズムは、 `!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!DHE` です。
デフォルト値：



```
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA3
```

パラメータ範囲：

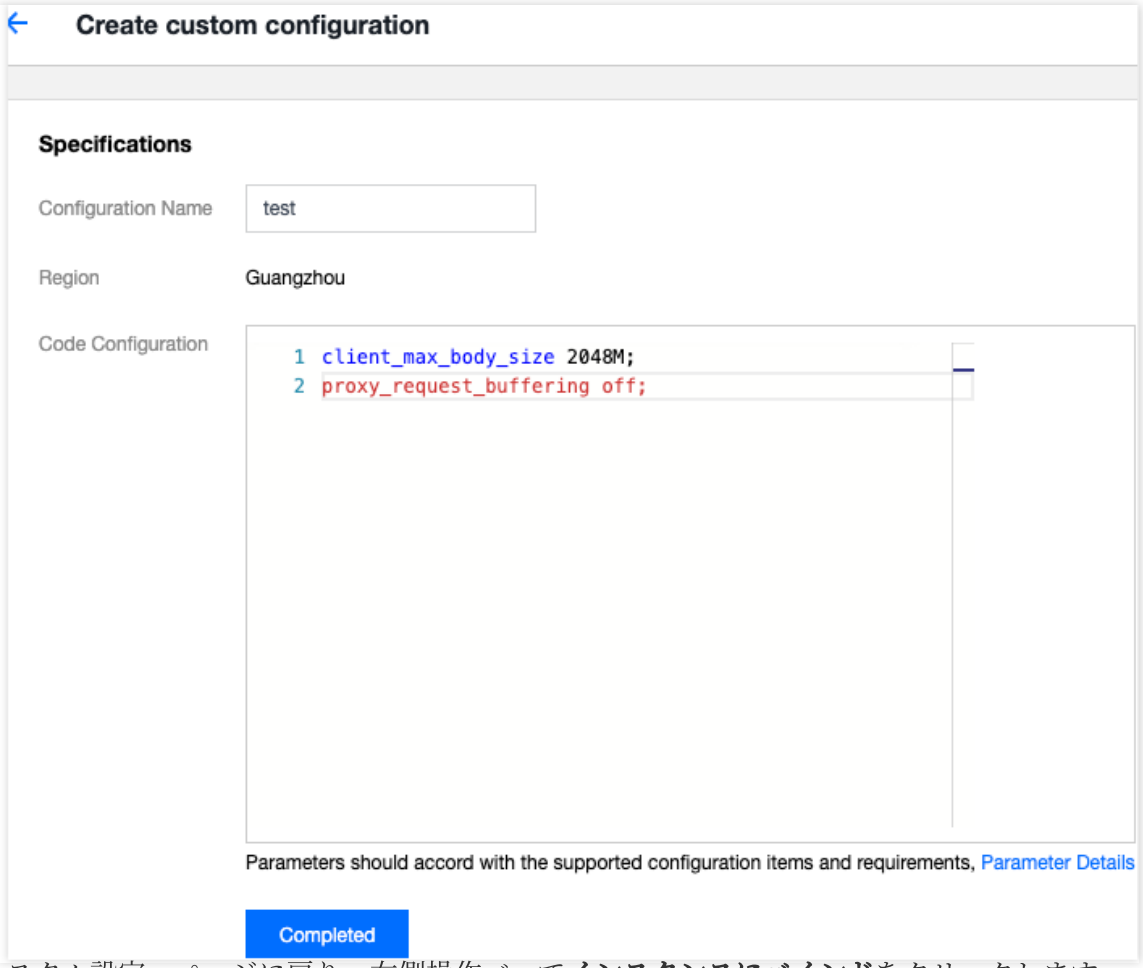


```
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA3
```

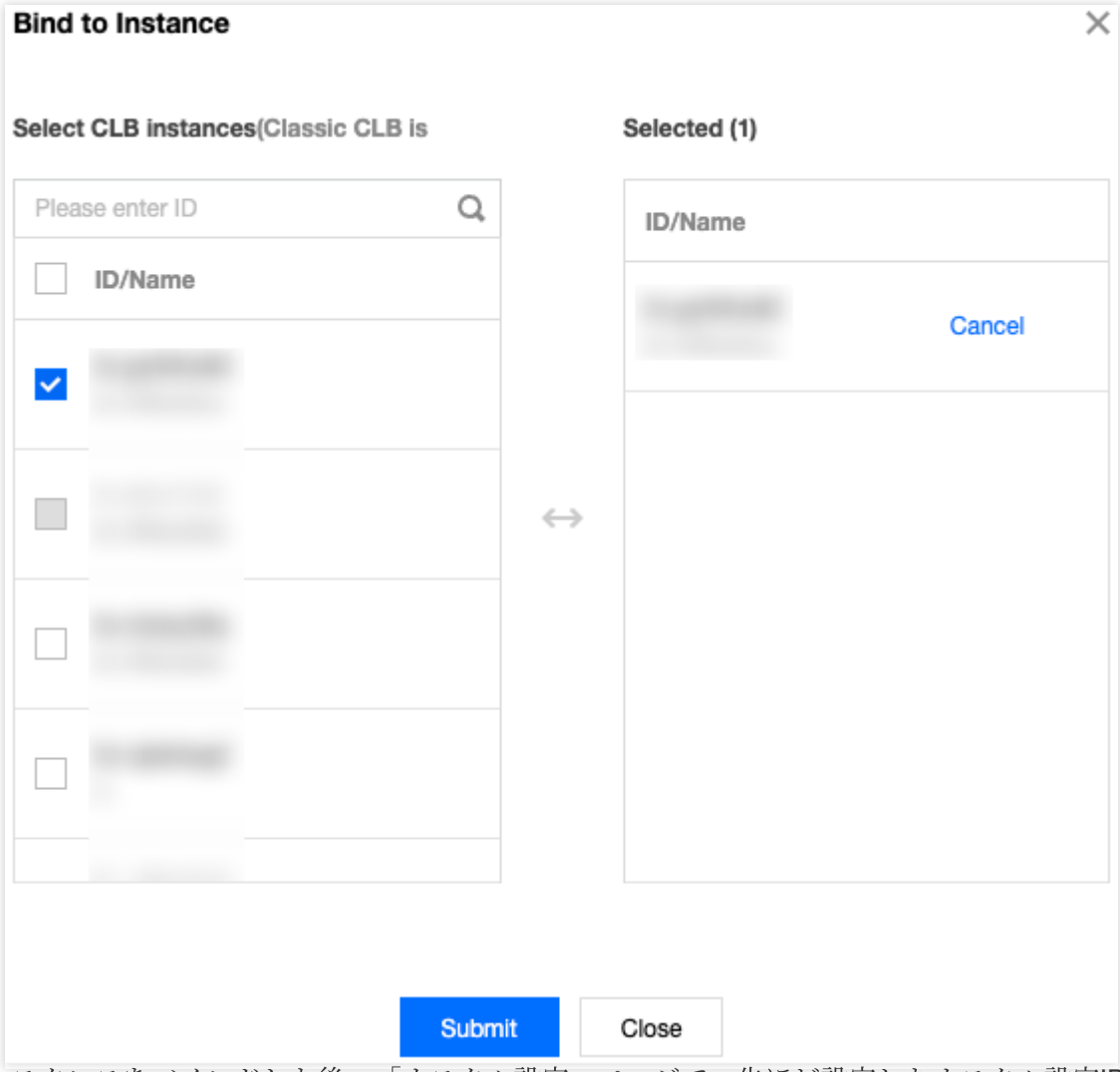
CLBカスタム設定の例

1. CLBコンソールにログインし、左側ナビゲーションバーで**カスタム設定**をクリックします。
2. 「カスタム設定」ページトップでリージョンを選択し、**新規作成**をクリックします。

3. 「カスタム設定の新規作成」ページで設定名とコード設定項目を入力します。コード設定項目の末尾は `;` とします。設定完了後、**完了**をクリックします。



4. 「カスタム設定」ページに戻り、右側操作バーで**インスタンスにバインド**をクリックします。
5. ポップアップした「インスタンスにバインド」ダイアログボックスでバインドしたいCLBインスタンスを選択し、**送信**をクリックします。



6. インスタンスをバインドした後、「カスタム設定」ページで、先ほど設定したカスタム設定IDをクリックして詳細ページに進み、**インスタンスのバインド**タブをクリックすると、先ほどバインドしたCLBインスタンスを確認できます。

7.（オプション）インスタンスをバインドした後、インスタンスのリストページで対応するカスタム設定情報を検索することもできます。

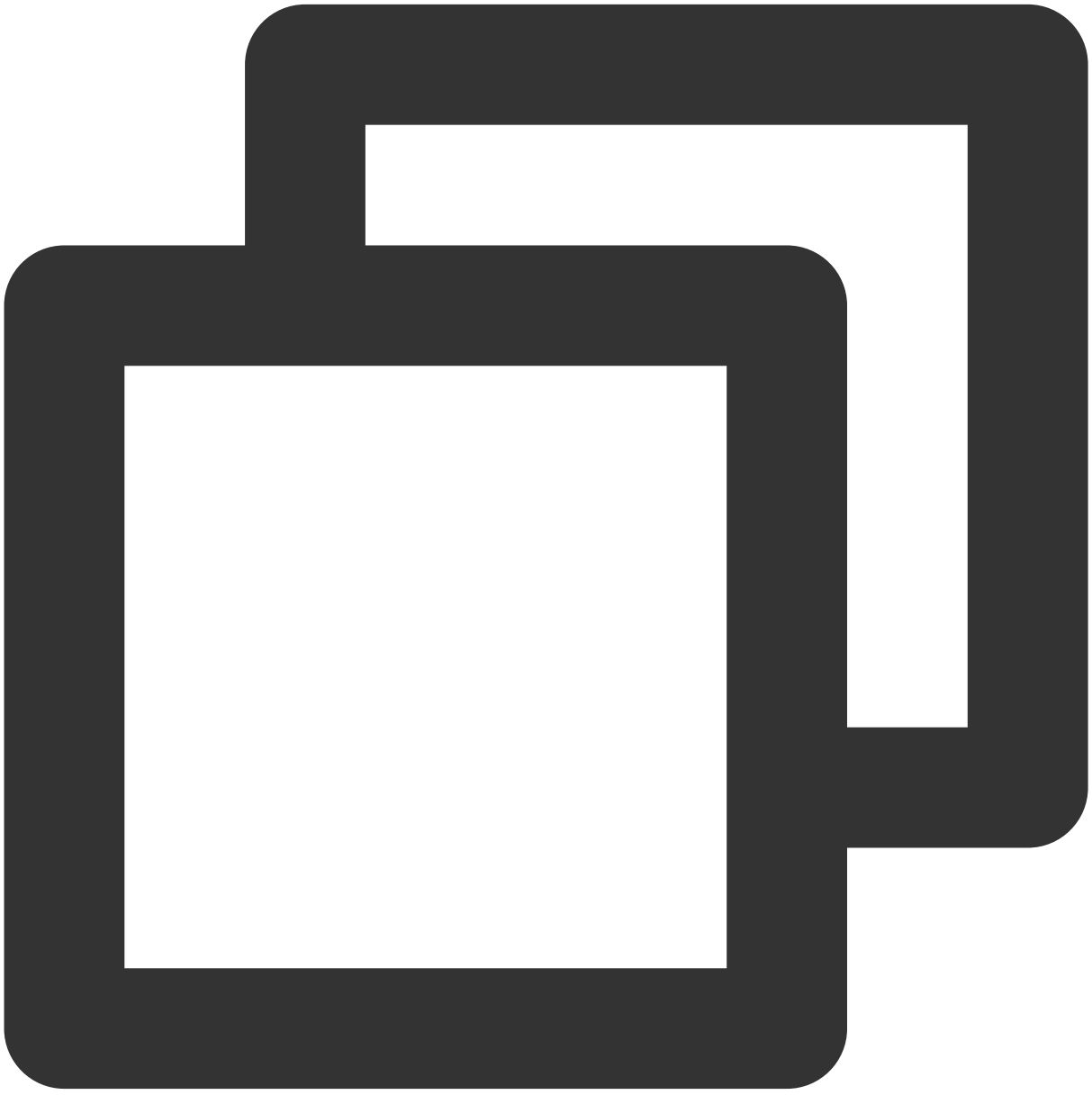
説明：

リストページに「カスタム設定のバインド」列が表示されていない場合は、リストページ右上隅の

アイコンをクリックし、ポップアップした「カスタムリストフィールド」ダイアログボックスで「カスタム設定のバインド」オプションにチェックを入れ、**OK**をクリックすると、リストページに「カスタム設定のバインド」列が表示されます。

ID/Name	Mon...	Status	VIP	Availability Z...	Network ...	Network	Health Status
[redacted]		Normal	[redacted]	Guangzhou Zone 4	Public Network	Basic Network	Health check not enabled (Configuration)

デフォルト設定コードの例：



```
ssl_protocols   TLSv1 TLSv1.1 TLSv1.2;
client_header_timeout   60s;
```

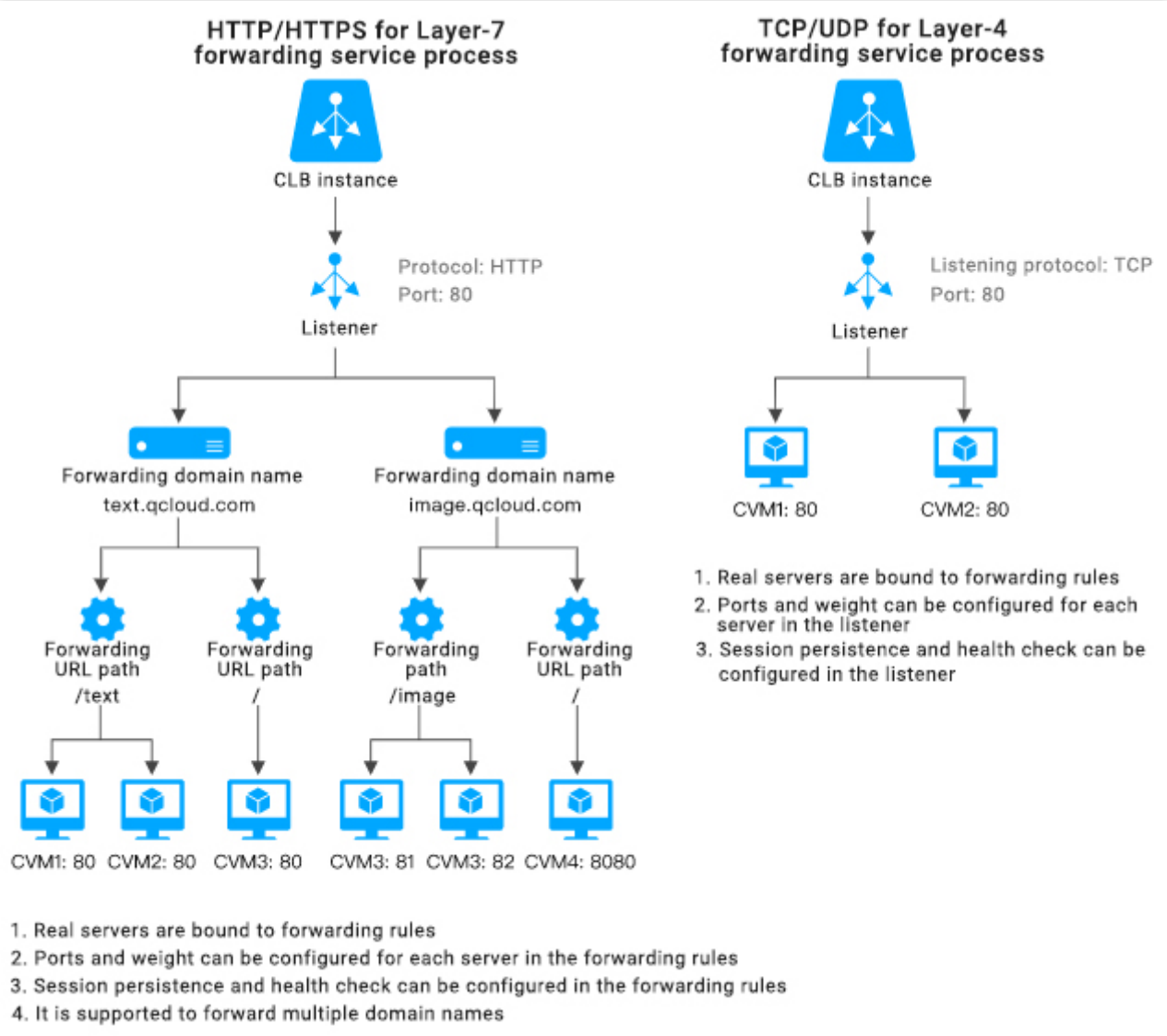
```
client_header_buffer_size   4k;
client_body_timeout    60s;
client_max_body_size   60M;
keepalive_timeout    75s;
add_header     xxx yyy;
more_set_headers      "A:B";
proxy_connect_timeout    4s;
proxy_read_timeout    60s;
proxy_send_timeout    60s;
```

レイヤー7転送ドメイン名およびURLルールの説明

最終更新日：：2024-01-04 18:36:26

業務フローチャート

CLB（旧「アプリケーション型CLB」）のレイヤー7の業務フローおよびレイヤー4の業務フローを次に示します。



CLBのレイヤー7転送HTTP/HTTPSプロトコルを使用する際は、CLBインスタンスのリスナーに新たな転送ルールを作成します。ユーザーは対応するドメイン名を追加することができます。
ユーザーが1つの転送ルールのみを作成した場合、アクセスVIP + URLはそれに応じた転送ルールに対応し、サービスにも正常にアクセスできます。

ユーザーが複数の転送ルールを作成した場合、アクセスVIP + URLはある具体的なドメイン名 + URLへのアクセスを確実に保証できないため、具体的な転送ルールを確実に有効化するには、ユーザーがドメイン名 + URLに直接アクセスする必要があります。つまり、ユーザーが複数の転送ルールを設定した場合、同一のVIPが複数のドメイン名に対応することになり、この場合はVIP + URLによるサービスへのアクセスは推奨されず、具体的なドメイン名 + URLによってサービスにアクセスする必要があります。

レイヤー7転送設定の説明

転送ドメイン名設定ルール

レイヤー7CLBはさまざまなドメイン名およびURLからのリクエストをさまざまなサーバーに転送して処理することができます。1つのレイヤー7リスナーには複数のドメイン名を設定することができ、1つのドメイン名には複数の転送パスを設定することができます。
転送ドメイン名の長さ制限は1～80文字です。
`_` で始めることはできません。
`www.example.com` のような、完全一致ドメイン名をサポートしています。
ワイルドカードドメイン名をサポートしています。現在は `*.example.com` または `www.example.*` の形式のみサポートしています。すなわち `*` を先頭または末尾に置き、かつ1つのドメイン名に使用できる `*` は1回のみとします。
非正規表現の転送ドメイン名については、サポートする文字セットは `a-z` `0-9` `.` `-` `_` となります。
転送ドメイン名は正規表現をサポートしています。正規表現のドメイン名については次のとおりです。
サポートする文字セットは `a-z0-9.-?=~_-+\\^*!$&|()[]` となります。
`~` で開始する必要があり、かつ `~` の使用は1回のみとします。
CLBがサポートする正規表現ドメイン名の例： `~^www\\d+\\.example\\.com$` 。

転送ドメイン名マッチングの説明

転送ドメイン名の共通マッチングポリシー

1. 転送ルールにドメイン名を設定せず、代わりにIPを入力し、転送グループ内に複数のURLを設定する場合、このサービスにはVIP + URLによってアクセスします。
2. 転送ルールに完全なドメイン名を設定し、転送グループ内に複数のURLを設定する場合、サービスにはドメイン名 + URLによってアクセスします。
3. 転送ルールにワイルドカードドメイン名を設定し、転送グループ内に複数のURLを設定する場合は、リクエストにマッチしたドメイン名 + URL によってアクセスします。異なるドメイン名が同一のURLアドレスを指定するようにしたい場合は、この方式を参照して設定することができます。 `example.qcould.com` を例にした形式を次に示します。
完全一致ドメイン名 `example.qcloud.com` は、 `example.qcloud.com` のドメイン名に正確にマッチします。

プレフィックスワイルドカードドメイン名 `*.qcloud.com` は、 `qcloud.com` で終わるドメイン名すべてにマッチします。
サフィックスワイルドカードドメイン名 `example.qcloud.*` は、 `example.qcloud` で始まるドメイン名すべてにマッチします。
正規表現マッチングドメイン名 `~^www\\d+\\.example\\.com$` は、正規表現に基づいてマッチングを行います。
マッチングの優先順位：完全一致ドメイン名>プレフィックスワイルドカードドメイン名>サフィックスワイルドカードドメイン名>正規表現マッチングドメイン名となります。同一の順位のドメイン名に複数のドメイン名が同時にヒットした場合は、マッチング順位の前後を保証できませんので、より正確なドメイン名を使用することで、複数のルールに同時にヒットしないようにすることをお勧めします。
4. 転送ルールにドメイン名を設定し、転送グループ内にあいまい一致のURLを設定します。プレフィックスマッチングを使用し、最後にワイルドカード$を加えて完全なマッチングを行うことができます。
例えば、ユーザーが `gif` 、 `jpg` または `bmp` で終わるあらゆるファイルにマッチしたい場合は、転送グループ `URL ~*.(gif|jpg|bmp)$` を設定することができます。

転送ドメイン名におけるデフォルトドメイン名ポリシー

クライアントリクエストがそのリスナーのどのドメイン名にもマッチしなかった場合、CLBはリクエストをデフォルトドメイン名（Default Server）に転送し、デフォルトルールを制御可能にします。1つのリスナーに設定できるデフォルトドメイン名は1つのみです。
例えば、CLB1の `HTTP:80` リスナーに2つのドメイン名 `www.test1.com` 、 `www.test2.com` を設定し、そのうち `www.test1.com` がデフォルトドメイン名だとします。ユーザーが `www.example.com` にアクセスした場合、どのドメイン名にもマッチしないため、CLBはこのリクエストをデフォルトドメイン名 `www.test1.com` に転送します。
説明：
2020年5月18日より以前は、レイヤー7リスナーにデフォルトドメイン名を設定するかどうかはオプションであり、デフォルトドメイン名を設定するかしないかを選択できました。
レイヤー7リスナーにすでにデフォルトドメイン名を設定している場合、他のルールにマッチしなかったクライアントリクエストはデフォルトドメイン名に転送されます。
レイヤー7リスナーにデフォルトドメイン名を設定していない場合、他のルールにマッチしなかったクライアントリクエストはCLBにロードされた最初のドメイン名に転送されます。ロードの順序はコンソールの設定順序とは一致しない可能性があるため、コンソールで最初に設定されているものと同じとは限りません。
2020年5月18日からは次のようになります。
新たに作成するすべてのレイヤー7リスナーにデフォルトドメイン名の設定が必要となります。レイヤー7リスナーの最初のルールは、デフォルトドメイン名を必ず有効化するものであり、APIを呼び出してレイヤー7ルールを作成する際、CLBは `DefaultServer` フィールドを自動的にtrueに設定します。
すでにデフォルトドメイン名を設定済みのすべてのリスナーは、デフォルトドメイン名を変更または削除する際に新しいデフォルトドメイン名を指定する必要があります。コンソールでの操作の際に新しいドメイン名の指定が

必要です。APIを呼び出して操作を行う際、新しいデフォルトドメイン名を設定しなければ、CLBは残りのドメイン名の中から、作成時間が最も古いものを新たなデフォルトドメイン名として自動的に設定します。
既存の未設定デフォルトドメイン名のルール：業務ニーズに応じてデフォルトドメイン名を直接設定することができます。操作手順は次の「操作4」のとおりです。設定しなければ、Tencent CloudはCLBにロードされた最初のドメイン名をデフォルトドメイン名に設定します。既存リスナーの処理は2020年6月19日までに完了します。
上記のポリシーは2020年5月18日から順次実施されますが、各インスタンスでの発効日には多少の違いが生じる可能性があります。2020年6月20日以降は、転送ドメイン名が空でないすべてのレイヤー7リスナーにはデフォルトドメイン名が存在することになります。
デフォルトドメイン名に関連する操作には次の4つがあります。
操作1：レイヤー7リスナーに最初の転送ルールを設定する際、デフォルトドメイン名は必ず有効な状態にしておかなければなりません。
操作2：現在のデフォルトドメイン名を無効化します。
あるリスナーに複数のドメイン名があり、現在のデフォルトドメイン名を無効化した場合、新しいデフォルトドメイン名を指定する必要があります。
あるリスナーにドメイン名が1つしかなく、かつそのドメイン名がデフォルトドメイン名の場合、デフォルトドメイン名を無効化することはできません。
操作3：デフォルトドメイン名を削除します。
あるリスナーに複数のドメイン名がある状況で、デフォルトドメイン名下のルールを削除する場合は次のようになります。
そのルールがデフォルトドメイン名の最後のルールではない場合は、直接削除することができます。
そのルールがデフォルトドメイン名の最後のルールである場合は、新しいデフォルトドメイン名を設定する必要があります。
あるリスナーにドメイン名が1つしかない場合は、すべてのルールを直接削除することができ、新しいデフォルトドメイン名を設定する必要もありません。

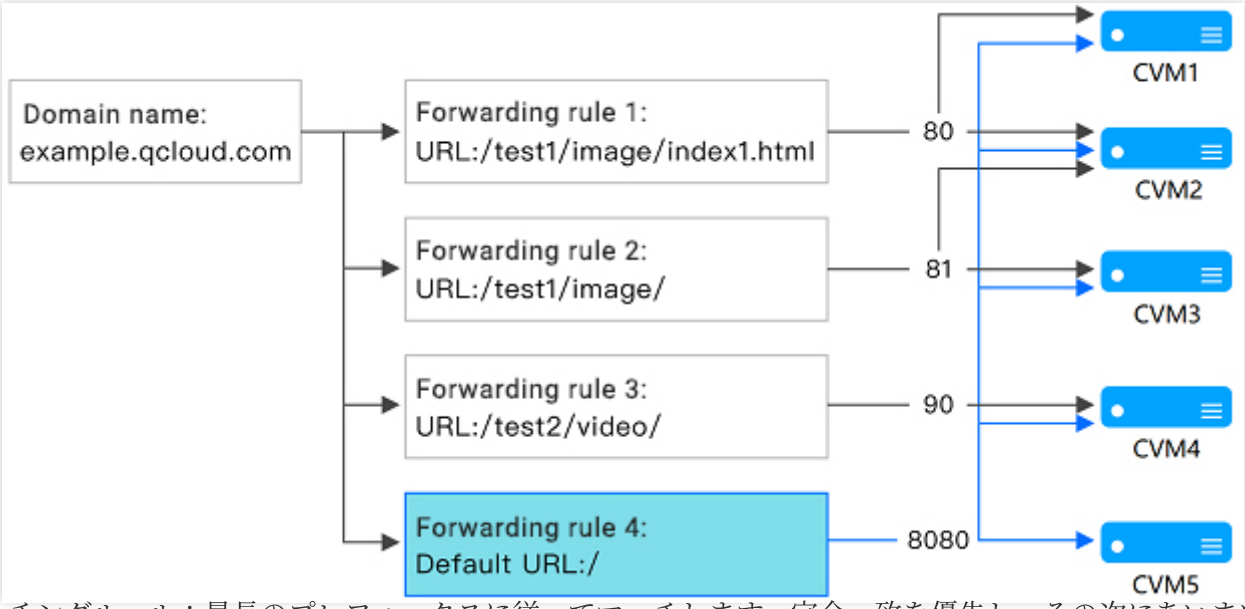
操作4
：デフォルトドメイン名を変更します。デフォルトドメイン名はリスナーリストからすぐに変更できます。

転送URLパス設定ルール

レイヤー7CLBはさまざまなURLからのリクエストをさまざまなサーバーに転送して処理することができます。1つのドメイン名には複数の転送URLパスを設定することができます。
転送URLの長さ制限は1～200文字です。
非正規表現の転送URLは `/` で始まる必要があり、大文字と小文字を区別します。サポートする文字セットは `a-z` `A-Z` `0-9` `.` `-` `_` `/` `=` `?` `:` となります。
転送URLは正規表現をサポートしています。
正規表現のURLは `~` で開始する必要があり、かつ `~` の使用は1回のみとします。
正規表現のURLがサポートする文字セットは `a-z` `A-Z` `0-9` `.` `-` `_` `/` `=` `?` `~` `^` `*` `$` `:` `(` `)` `[` `]` `+` `|` となります。

正規表現のURLの例は `~* .png$` です。
転送URLのマッチングルールは次のとおりです。
`=` で始まる場合は完全一致を表します。
`^~` で始まる場合は、URLがある通常の文字列で始まり、正規表現マッチングではないことを表します。
`~` で始まる場合は、大文字と小文字を区別する正規表現マッチングであることを表します。
`~*` で始まる場合は、大文字と小文字を区別しない正規表現マッチングであることを表します。
`/` は汎用マッチングです。他のマッチングがない場合、あらゆるリクエストがマッチします。

転送URLパスマッチング説明



1. マッチングルール：最長のプレフィックスに従ってマッチします。完全一致を優先し、その次にあいまい一致とします。
例えば、上の図に従って転送ルールおよび転送グループを設定した場合、リクエストは次のように、異なる転送ルールに順番にマッチングされます。
1.1 `example.qloud.com/test1/image/index1.html` は転送ルール1で設定したURLのルールに完全一致し、このリクエストは転送ルール1に関連付けられたバックエンドCVM、すなわち図のCVM1およびCVM2の80番ポートに転送されます。
1.2 `example.qloud.com/test1/image/hello.html` には完全一致がなく、最長のプレフィックスに従って転送ルール2にマッチします。このため、このリクエストは転送ルール2に関連付けられたバックエンドCVM、すなわち図のCVM2およびCVM3の81番ポートに転送されます。
1.3 `example.qloud.com/test2/video/mp4/` には完全一致がなく、最長のプレフィックスに従って転送ルール3にマッチします。このため、このリクエストは転送ルール3に関連付けられたバックエンドCVM、すなわち図のCVM4の90番ポートに転送されます。
1.4 `example.qloud.com/test3/hello/index.html` には完全一致がなく、最長のプレフィックスに従ってルートディレクトリDefault URL： `example.qloud.com/` にマッチします。この場合、Nginxはリクエストを

FastCGI（php）、Tomcat（jsp）のようなバックエンドアプリケーションサーバーに転送し、Nginxはリバースプロキシサーバーとして存在します。
1.5 `example.qloud.com/test2/` には完全一致がなく、最長のプレフィックスに従ってルートディレクトリDefault URL： `example.qloud.com/` にマッチします。
2. ユーザーが設定したURLルールにおいてサービスが正常に実行できない場合、マッチング成功後に他のページへのリダイレクトは行われません。
例えば、クライアントリクエスト `example.qloud.com/test1/image/index1.html` が転送ルール1にマッチしたものの、このとき転送ルール1のバックエンドサーバーの動作に異常があり、404のページが表示された場合、ユーザーがアクセスした際にも404が表示され、他のページにはリダイレクトされません。
3. Default URLを設定し、それがサービスの安定しているページ（静的ページ、トップページなど）を指定するようにした上で、すべてのバックエンドCVMをバインドすることをユーザーにお勧めします。このとき、どのルールもマッチングに成功しなかった場合、システムによってリクエストはDefault URLの存在するページを指定します。それが行われない場合は404の問題が発生する可能性があります。
4. ユーザーがDefault URLを設定せず、なおかつすべての転送ルールがマッチしなかった場合、サービスにアクセスすると404が返されます。
5. レイヤー7URLパス末尾のスラッシュの説明：ユーザーが設定したURLは `/` で終わっているが、クライアントのアクセス時に `/` が含まれなかった場合、このリクエストは `/` で終わるルールにリダイレクトされます（301リダイレクト）。
例えば、 `HTTP:80` リスナー下で設定したドメイン名が `www.test.com` であるとします。
5.1 このドメイン名に設定したURLが `/abc/` である場合：
クライアントが `www.test.com/abc` にアクセスした際は、 `www.test.com/abc/` にリダイレクトされます。
クライアントが `www.test.com/abc/` にアクセスした際は、 `www.test.com/abc/` にマッチします。
5.2 このドメイン名に設定したURLが `/abc` である場合：
クライアントが `www.test.com/abc` にアクセスした際は、 `www.test.com/abc` にマッチします。
クライアントが `www.test.com/abc/` にアクセスした際も、 `www.test.com/abc` にマッチします。

レイヤー7ヘルスチェック設定の説明

ヘルスチェックドメイン名設定ルール

ヘルスチェックドメイン名はレイヤー7CLBがバックエンドサービスのヘルスステータスをチェックするためのドメイン名です。
ヘルスチェックドメイン名の長さ制限は1～80文字です。
ヘルスチェックドメイン名はデフォルトでは転送ドメイン名です。
ヘルスチェックドメイン名は正規表現をサポートしていません。転送ドメイン名がワイルドカードドメイン名の場合は、ある特定のドメイン名（非正規表現）をヘルスチェックドメイン名として指定する必要があります。

ヘルスチェックドメイン名がサポートする文字セットは `a-z` `0-9` `.` `-` `_` であり、例えば `www.example.qcould.com` のようになります。

ヘルスチェックパス設定ルール

ヘルスチェックパスはレイヤー7CLBがバックエンドサービスのヘルスステータスをチェックするためのURLパスです。
ヘルスチェックパスの長さ制限は1～200文字です。
ヘルスチェックパスはデフォルトでは `/` であり、必ず `/` で始めなければなりません。
ヘルスチェックパスは正規表現をサポートしていません。ある特定のURLパス（静的ページ）を指定してヘルスチェックを行うことをお勧めします。
ヘルスチェックパスがサポートする文字セットは `a-z` `A-Z` `0-9` `.` `-` `_` `/` `=` `?` `:` であり、例えば `/index` のようになります。

CLBのQUICプロトコルのサポート

最終更新日：：2024-01-04 18:36:26

QUICプロトコルはAppへのアクセス速度を大幅に向上させることができ、脆弱なネットワーク下や、Wi-Fiと4Gを頻繁に切り替えるシーンなどで、再接続を必要とせずに多重化を実現できます。ここでは、CLBコンソールでQUICプロトコルを設定する方法についてご説明します。

QUICの概要

QUIC（Quick UDP Internet Connection）は高速UDPインターネット接続とも呼ばれ、Googleが提唱する、UDPを使用してマルチパス通信を行うプロトコルです。現在広く用いられているTCP+TLS+HTTP2プロトコルと比較して、QUICには次のようなメリットがあります。
接続確立時間が短縮されます。
輻輳制御が改善されます。
多重化によってHOLブロッキングを解消します。
コネクションのマイグレーションが可能です。
CLBでQUICを有効化すると、クライアントはCLBとの間でQUIC接続を確立することができ、両者のプロトコルがQUIC接続を確立できない場合は自動的にHTTPSまたはHTTP/2にダウングレードされます。ただし、CLBとバックエンドサーバーの間では引き続きHTTP1.xプロトコルが用いられます。

使用制限

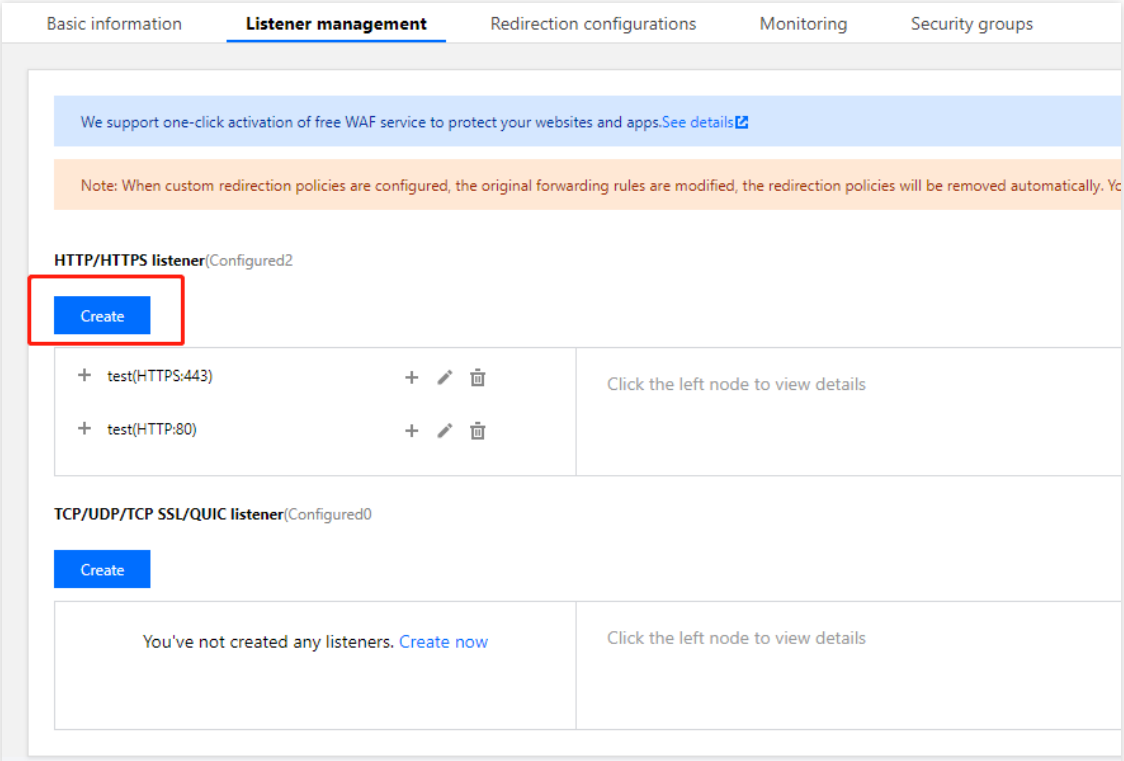
CLBインスタンスタイプのみサポートし、従来型CLBインスタンスはサポートしていません。
IPv4、IPv6 NAT64バージョンのCLBのみサポートしています。IPv6バージョンは現時点ではサポートしていません。
レイヤー7HTTPSリスナーのみ、QUICプロトコルをサポートしています。
現在CLBがサポートしているQUICのバージョンは、Q050、Q046、Q043、h3-29、h3-27です。

操作手順

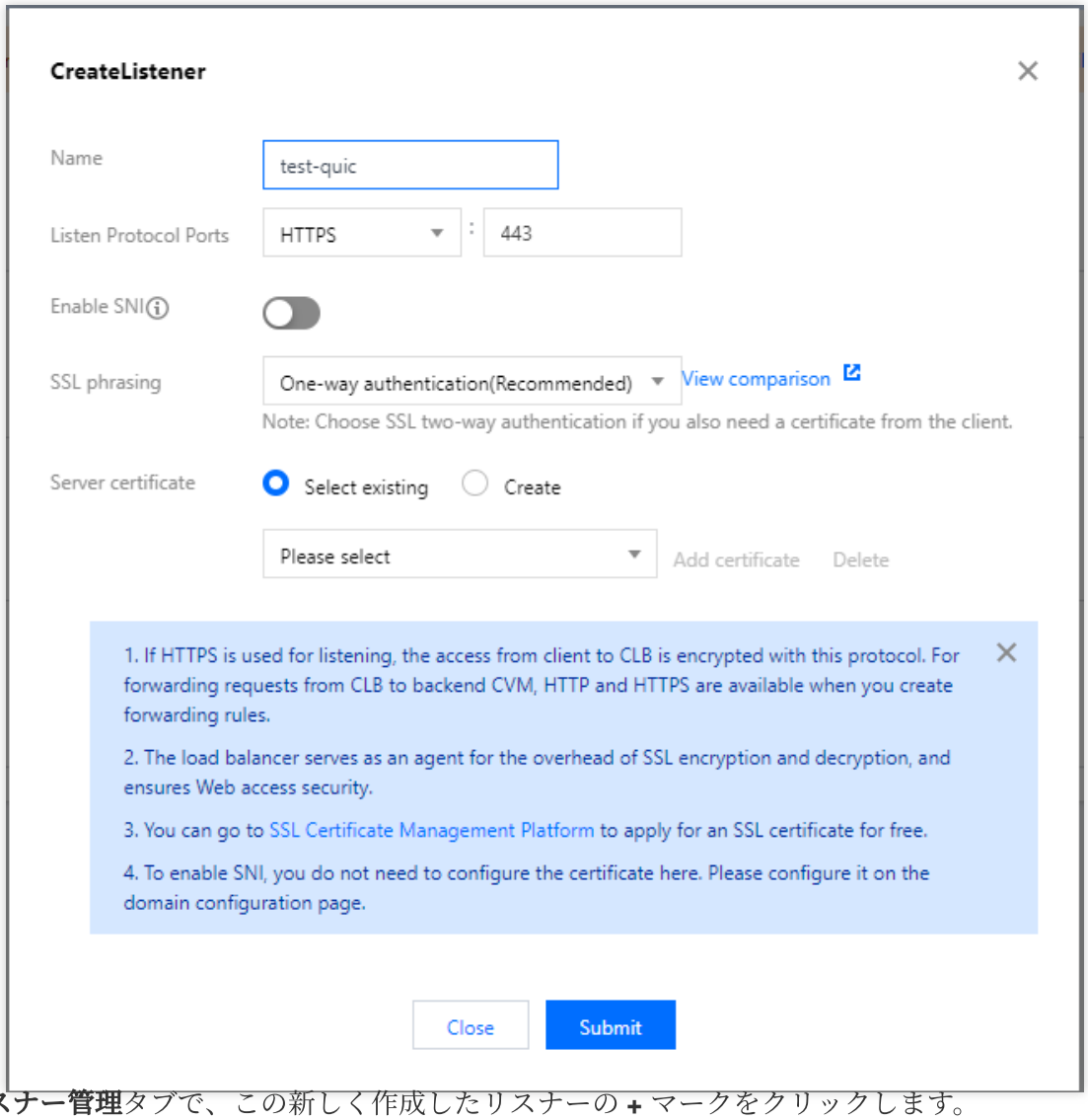
1. 必要に応じてCLBインスタンスを作成します。詳細については、CLBインスタンスの作成をご参照ください。
説明：
CLBインスタンスを作成する際、作成リージョンは「北京」、「上海」または「ムンバイ」を選択し、ネットワークタイプは「パブリックネットワーク」を選択します。
2. CLBコンソールにログインし、左側ナビゲーションバーの**インスタンス管理**をクリックします。

3. 「インスタンス管理」ページで、**CLB**をクリックします。
4. 「CLB」タブで、作成リージョンが「北京」、「上海」または「ムンバイ」のパブリックネットワークCLBインスタンスを見つけ、右側の操作バーで**リスナーの設定**をクリックします。

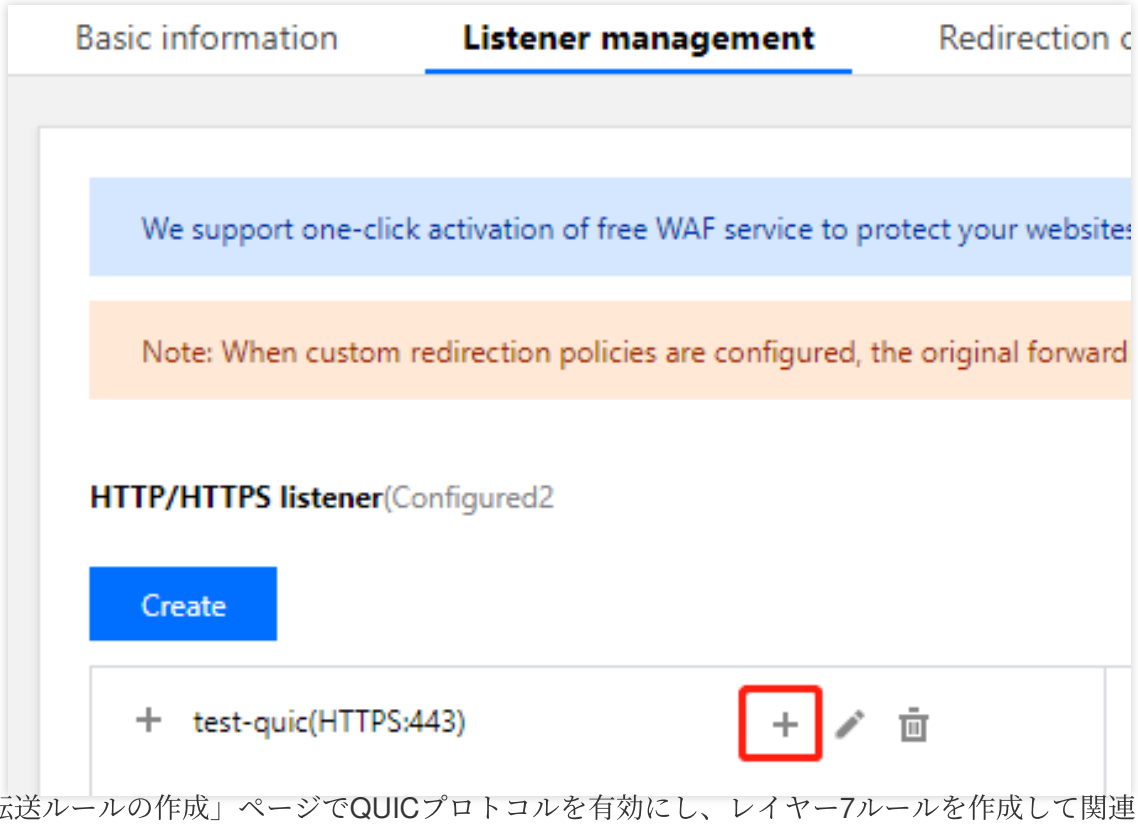
5. 「リスナー管理」ページの「HTTP/HTTPSリスナー」で、**新規作成**をクリックします。



6. 「リスナーの作成」ページでリスニングプロトコルポートをHTTPSに切り替え、必要に応じて入力を完了した後、**送信**をクリックします。



7. **リスナー管理**タブで、この新しく作成したリスナーの **+** マークをクリックします。

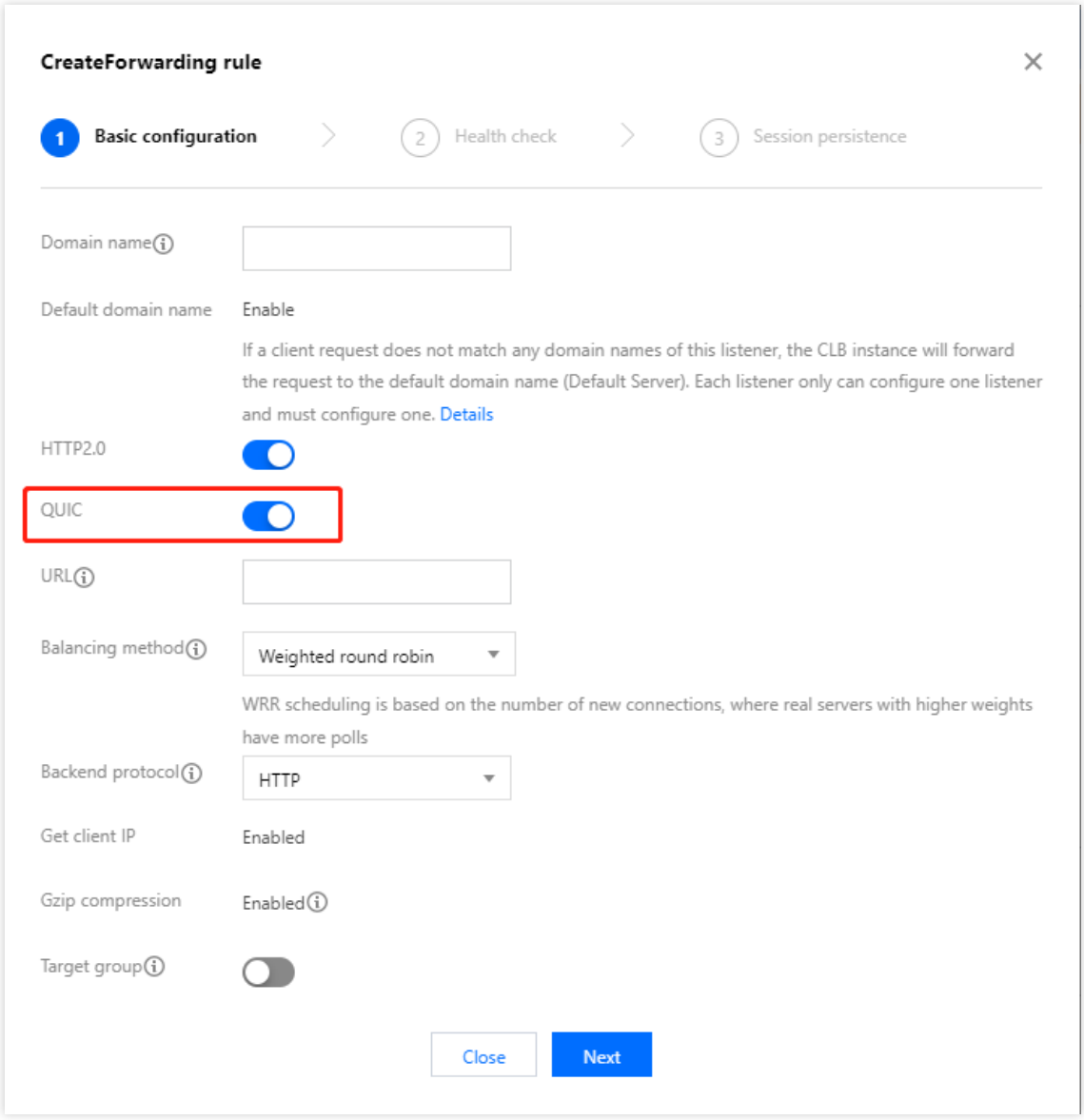


8. 「転送ルールの作成」ページでQUICプロトコルを有効にし、レイヤー7ルールを作成して関連のフィールドに入力した後、**次へ**をクリックすると、基本設定は完了です。

説明：

作成完了後にQUICプロトコルのスイッチステータスを変更する場合は、対応するルールのドメイン名のところで編集してください。

QUICはUDPプロトコルを使用し、CLBのUDPポートを占有します。つまり、HTTPSリスナーでQUICプロトコルを有効化すると、対応するUDPポートおよびTCPポートを自動的に占有します。例えば、HTTPS:443リスナーでQUICプロトコルを有効化すると、このルールによって同時にTCP:443およびUDP:443ポートが占有されるため、TCP:443およびUDP:443リスナーは作成できなくなります。



後続の操作

基本設定の入力完了後、引き続きヘルスチェックおよびセッション維持の関連操作を行うことができます。

CLBのSNIマルチドメイン名証明書のサポート

最終更新日：：2024-01-04 18:36:26

サーバー名表示（Server Name Indication，SNI）とは、サーバーとクライアント間のSSL/TLSを改善するために用いられるもので、1台のサーバーにつき1つの証明書しか使用できない問題を主に解決します。SNIをサポートすることは、サーバーに複数の証明書をバインドできることを意味します。クライアントがSNIを使用するには、サーバーとの間でSSL/TLS接続を確立する前に、接続したいドメイン名を指定する必要があり、サーバーはこのドメイン名に基づいて適切な証明書を返します。

シナリオ

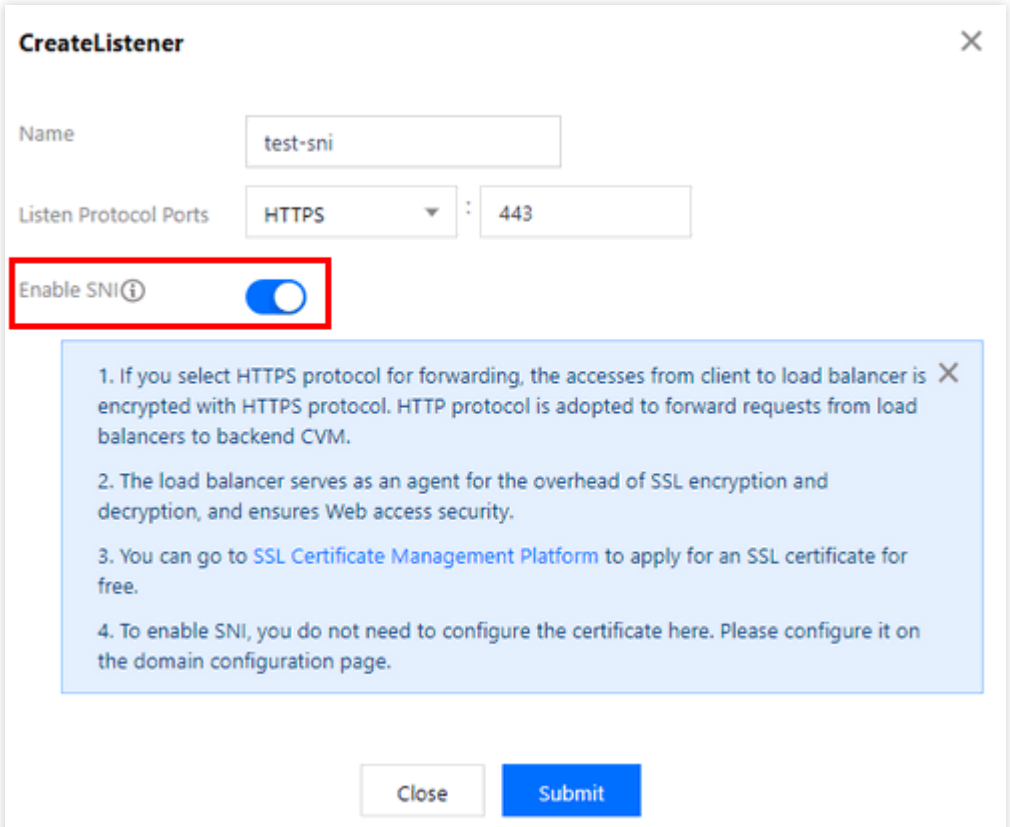
Tencent Cloud CLBのレイヤー7HTTPSリスナーはSNIをサポートしています。つまり、複数の証明書のバインドをサポートし、リスニングルール内のドメイン名ごとに異なる証明書を使用できます。例えば、同一のCLBの `HTTPS:443` リスナーで、 `*.test.com` が証明書1を使用している場合、このドメイン名からのリクエストは1組のサーバーに転送され、証明書2を使用している `*.example.com` からのリクエストは別の1組のサーバーに転送されます。

前提条件

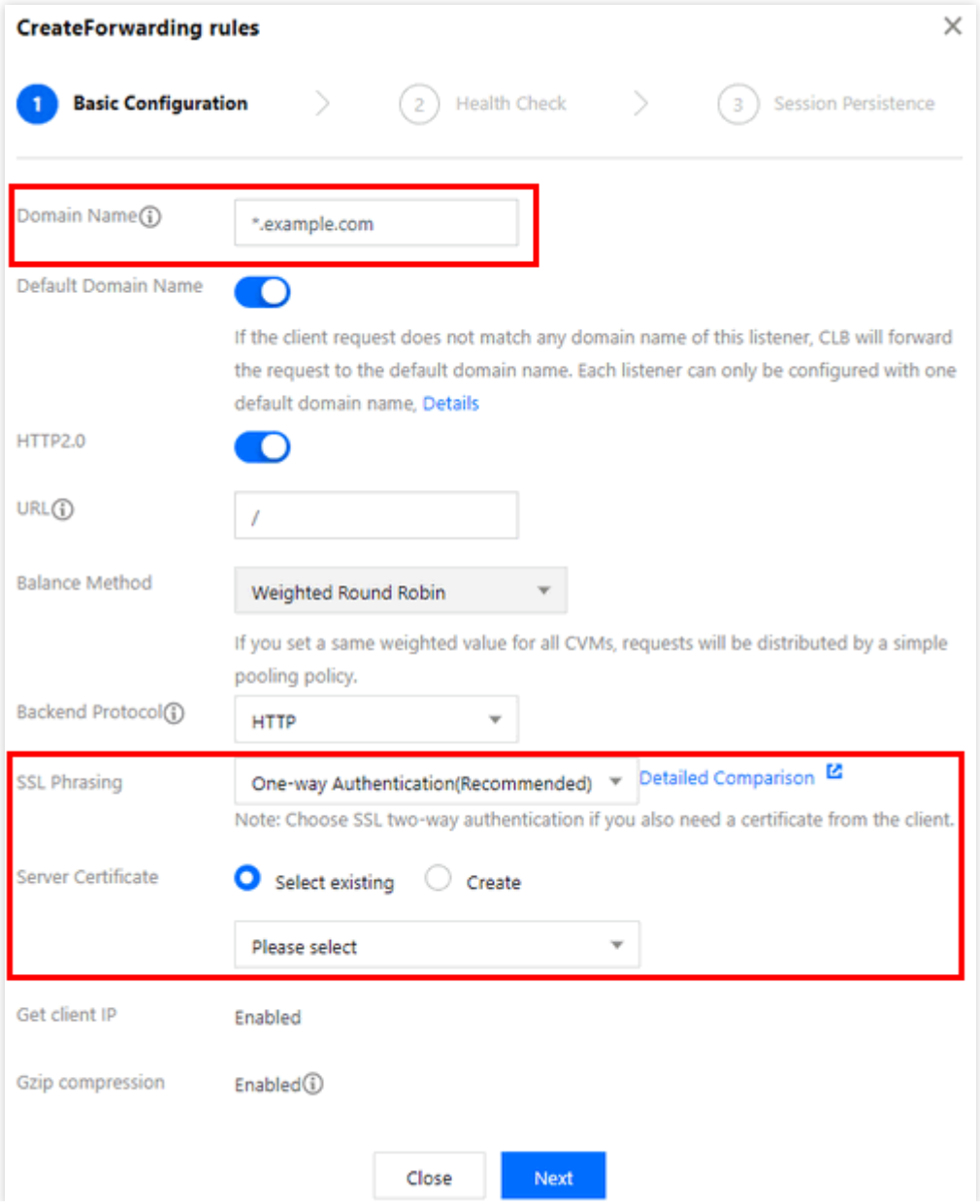
CLBインスタンスの購入をしていることが必要です。
説明：
従来型CLBは、ドメイン名およびURLベースの転送をサポートしていないため、従来型CLBはSNIをサポートしていません。

操作手順

1. CLBコンソールにログインします。
2. リスナーの設定の操作手順を参照してリスナーを設定し、HTTPSリスナーを設定する際にSNIを有効化します。



3. リスナーに転送ルールを追加する際、ドメイン名ごとに異なるサーバー証明書を設定し、**次のステップ**をクリックします。続いてヘルスチェックとセッション維持の設定を完了します。



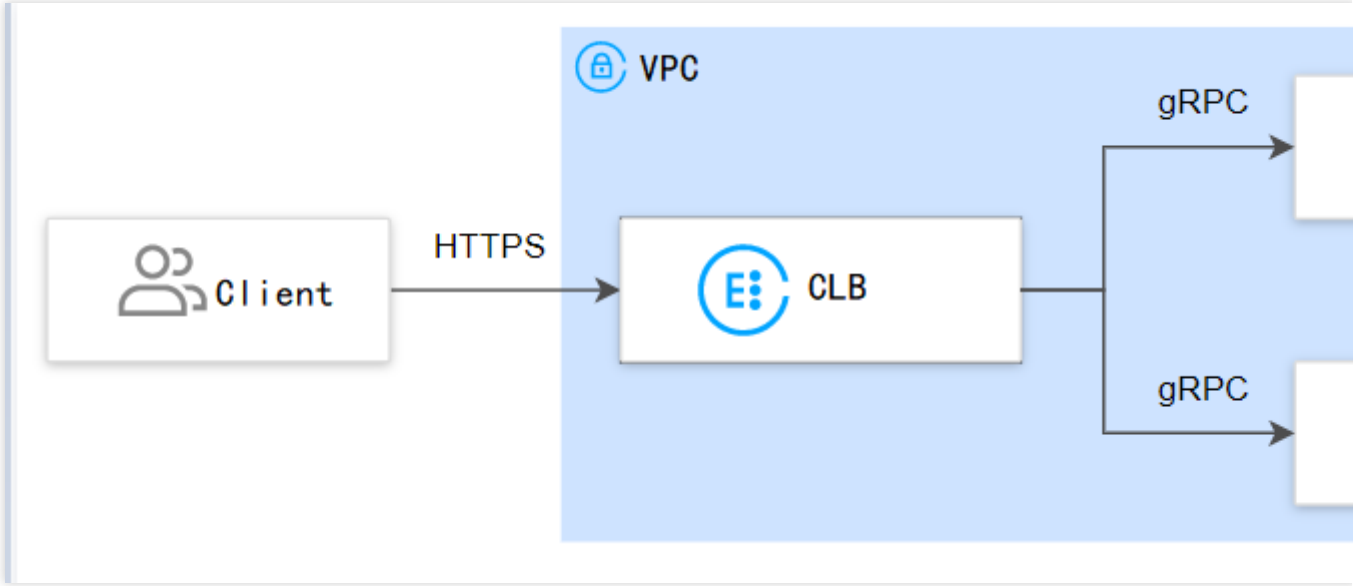
レイヤー7プロトコル　gRPCをサポート

最終更新日：：2024-01-04 18:36:26

gRPCは、Googleが公開したHTTP 2.0トランスポート層プロトコルをベースとする高性能なオープンソースソフトウェアフレームワークで、複数のプログラミング言語をサポートし、ネットワークデバイスを設定かつ管理する方法を提供します。ここでは、HTTPSリスナーのgRPCプロトコルのヘルスチェックを設定することによって、クライアントのgRPCリクエストを、CLBインスタンスを介してバックエンドプロトコルがgRPCであるバックエンドサービスに転送する方法についてご説明します。

シナリオ

クライアントがHTTPSリクエストでプロトコルタイプがgRPCであるバックエンドサービスにアクセスする場合、CLBインスタンスのHTTPSリスナーを介してgRPCプロトコルをサポートすることで実現できます。



前提条件

VPCを作成済みであること。詳細については、VPCの作成をご参照ください。
VPC内でCVMインスタンスを作成し、そのインスタンスにgRPCサービスをデプロイしていること。詳細については、イメージによってインスタンスを作成をご参照ください。
CLBインスタンスを購入済みであること。詳細については、CLBインスタンスの作成をご参照ください。

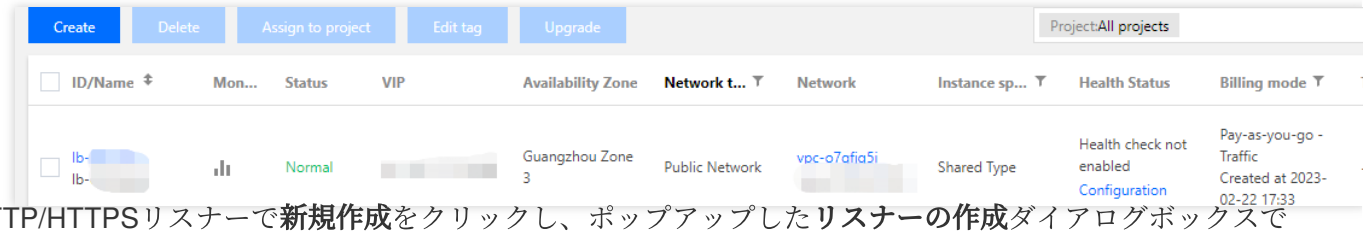
使用制限

CLBタイプのみサポートし、従来型CLBはサポートしていません。
IPv6バージョンのCLBとレイヤー7ハイブリッドバインドを有効化しているIPv6バージョンのCLBはサポートしていません。
VPCネットワークのみサポートし、基幹ネットワークではサポートしていません。
バックエンドサービスではSCFをサポートしていません（SCF target内でgRPCプロトコルをサポートしている必要があります）。

操作手順

ステップ1：リスナーの設定

1. CLBコンソールにログインし、左側ナビゲーションバーの**インスタンス管理**をクリックします。
2. CLBインスタンスリストページの左上隅でリージョンを選択し、インスタンスリスト右側の**操作**列で**リスナーの設定**をクリックします。



3. HTTP/HTTPSリスナーで**新規作成**をクリックし、ポップアップした**リスナーの作成**ダイアログボックスでHTTPSリスナーの設定を行います。

3.1 リスナーの作成

リスナーの基本設定	説明	事例
名前	リスナーの名称です。	test-https-443
リスニングプロトコルポート	リスニングプロトコル：この例ではHTTPSを選択します。 リスニングポート：リクエストを受信してバックエンドサーバーにリクエストを転送するために使用するポートで、ポート範囲は1～65535です。 同一CLBインスタンス内で、リスニングポートは重複できません。	HTTPS:443
長時間接続の有効化	有効化すると、CLBとバックエンドサービス間で長時間接続を使用し、CLBはソースIPをパススルーしなくなりますので、XFFからソースIPを取得してください。正常な転送を保証するため、CLB上でセキュリティグループを有効化してデフォルトで許可するか、またはCVMのセキュリティグループで100.127.0.0/16を許可してください。 **説明**：有効化すると、CLBとバックエンドサーバーの接続数の範囲はリクエストの[QPS、QPS*60]の間で変動し、具体的な数値は接続再利用率によって決まります。バックエンドサービスが接続数の上限に制限を設けている場	未有効化

	合、有効化は慎重に行うことをお勧めします。この機能は現在ベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。	
back-to-originの有効化	SNIの有効化は、1つのリスナーの下でドメイン名ごとに異なる証明書を設定できることを意味します。SNIを有効化しないことは、このリスナーでは複数のドメイン名に同一の証明書を使用することを意味します。	未有効化
SSL解析方式	単方向認証および双方向認証をサポートしています。ロードバランサがSSLの暗号化と復号のオーバーヘッドを代行し、アクセスの安全性を保証します。	単方向認証
サーバー証明書	SSL証明書プラットフォームにすでにある証明書を選択するか、または証明書をアップロードできます。	既存のものを選択します

3.2 転送ルールの作成

転送ルールの基本設定	説明	事例
ドメイン名	転送ドメイン名： 長さ制限：1〜80文字です。 `_` で始めることはできません。 正確なドメイン名およびワイルドカードのドメイン名をサポートしています。 正規表現をサポートしています。 具体的な設定ルールです。詳細については、転送ドメイン名設定ルールをご参照ください。	www.example.com
デフォルトドメイン名	リスナーのすべてのドメイン名がマッチングに成功しなかった場合、システムはリクエストにデフォルトのアクセスドメイン名を指定し、デフォルトアクセスを制御可能にします。 1つのリスナーの下に設定できるデフォルトドメイン名は1つだけです。	オン
HTTP 2.0	HTTP2.0を有効化すると、CLBはHTTP2.0のリクエストを受信できるようになります。クライアントがCLBをリクエストする際にどのHTTPバージョンを使用していても、CLBがバックエンドサーバーにアクセスする際のHTTPバージョンは常にHTTP 1.1となります。	オン
QUIC	QUICを有効化すると、クライアントはCLBとのQUIC接続を確立できるようになります。両者間のネゴシエーションによってQUIC接続が確立できない場合、自動的にHTTPSまたはHTTP/2にダウングレードされますが、CLBとバックエンドサーバーの間ではHTTP1.xプロトコ	オン

	ルが引き続き使用されます。詳細については、CLBがサポートするQUICプロトコルをご参照ください。	
URLパス	転送URLパス： 長さ制限：1～200文字です。 正規表現をサポートしています。 具体的な設定ルールです。詳細については、転送URLパス設定ルールをご参照ください。	/index
バランシング方式	HTTPSリスナーでは、CLBは重み付けラウンドロビン（WRR）、重み付け最小接続（WLC）およびIP Hashの3種類のスケジューリングアルゴリズムをサポートしています。 重み付けラウンドロビンアルゴリズム：バックエンドサーバーの重みに基づき、順番にリクエストを異なるサーバーに配信します。重み付けラウンドロビンアルゴリズムは新規接続数に基づいてスケジューリングし、重みの高いサーバーがラウンドロビンされる回数（確率）が高くなるほど、同じ重みのサーバーは同じ数の接続数を処理します。 重み付け最小接続：サーバーの現在アクティブな接続数に基づいてサーバーの負荷状況を推定します。重み付け最小接続はサーバー負荷および重みに基づいて総合的にスケジューリングし、重み値が同じ場合、現在の接続数が少ないバックエンドサーバーほどラウンドロビンされる回数（確率）も高くなります。 IP Hash：リクエストのソースIPアドレスに応じて、ハッシュキー（Hash Key）を使用し、静的に割り当てられたハッシュテーブルから対応するサーバーを見つけます。そのサーバーが使用可能であり、かつオーバーロード状態ではない場合はリクエストがそのサーバーに送信され、そうではない場合は空が返されます。	重み付けラウンドロビン
バックエンドプロトコル	バックエンドプロトコルとは、CLBとバックエンドサービスとの間のプロトコルのことです。 バックエンドプロトコルとしてHTTPを選択した場合、バックエンドサービスはHTTPサービスをデプロイする必要があります。 バックエンドプロトコルとしてHTTPを選択した場合、バックエンドサービスはHTTPサービスをデプロイする必要があり、HTTPSサービスの暗号化/復号により、バックエンドサービスのリソース消費量がより多くなります。 バックエンドプロトコルとしてgRPCを選択した場合、バックエンドサービスはgRPCサービスをデプロイする必要があります。HTTP2.0が有効でQUICが無効になっている場合にのみ、バックエンドの転送プロトコルとしてgRPCの選択がサポートされます。	gRPC
クライアントIPを取得	デフォルトで有効です。	すでにオンです
Gzip圧縮	デフォルトで有効です。	すでにオンです

3.3 HTTPSヘルスチェックログ

3.4 セッション維持

セッション維持の設定	説明	事例
セッション維持の有効化/無効化	セッションの維持を有効化すると、CLBリスナーは同一クライアントからのアクセスリクエストを同一のバックエンドサーバーに配信します。 TCPプロトコルはクライアントIPアドレスのセッションの維持に基づき、同一IPアドレスからのアクセスリクエストを同一のバックエンドサーバーに転送します。 重み付けラウンドロビンスケジューリングはセッションの維持をサポートします。 重み付け最小接続スケジューリングはセッション維持機能の有効化をサポートしていません。	オン
セッションの維持時間	維持時間を超え、接続中に新たなリクエストがない場合は、自動的にセッションの維持が切断されます。 設定可能範囲は30s～3600sです。	30s

ステップ2：バックエンドCVMのバインド

1. **リスナー管理**ページで、上記の `HTTPS:443` リスナーなどの、先ほど作成したリスナーをクリックし、左側の**+**をクリックしてドメイン名およびURLパスを表示します。具体的なURLパスを選択すると、リスナーの右側にそのパスにバインド済みのバックエンドサービスが表示されます。
2. **バインド**をクリックし、バインドしたいバックエンドサーバーをポップアップボックスから選択し、サービスポートと重みを設定します。

説明：

デフォルトポート機能：先に「デフォルトポート」を入力してからCVMを選択すると、それぞれのCVMのポートがすべてデフォルトポートとなります。

ステップ3：セキュリティグループ（オプション）

CLBのセキュリティグループを設定して、パブリックネットワークトラフィックの分離を行うことができます。詳細については、CLBセキュリティグループの設定をご参照ください。

ステップ4：リスナーの変更/削除（オプション）

作成したリスナーを変更または削除したい場合、**リスナー管理**ページで、作成したリスナーをクリックし、

アイコンをクリックして変更または

アイコンをクリックして削除してください。

バックエンドサーバー
バックエンドCVMの概要

最終更新日：：2024-01-04 18:36:26

バックエンドサーバーはCLBインスタンスを作成した後、CLBにバインドして対応する転送リクエストを処理するサーバーです。CLBリスナーを設定する時に、バックエンドサーバーをバインドし、異なるラウンドロビン方式によって、リクエストをバックエンドサーバーに転送し、バックエンドサーバーによって処理し、アプリケーションの安定的かつ信頼性のある実行を保証する必要があります。

サポートするバックエンドサーバータイプ

CLBがサポートするバックエンドサービスタイプはインスタンスタイプ、IPタイプおよびServerless Cloud Function（SCF）タイプを含みます。このうち、
インスタンスタイプはCloud Virtual Machine（CVM）、Elastic Network Interface（ENI）およびElastic Kubernetes Service（EKS）を含みます。
IPタイプは主にクラウド上のマルチVPCのプライベートIP、およびクラウド下のIDCのプライベートIPをバインドするために使用されます。

注意事項

バックエンドサーバーを追加する際は、次のことをお勧めします。
セッション維持機能を有効化し、CLBに比較的長時間のTCP接続を維持させ、複数のリクエストによる再利用を可能にすることで、Webサーバー上の負荷を減少させてCLBのスループットを向上させることをお勧めします。
バックエンドサービスのセキュリティグループがCLBリスナーポートおよびヘルスチェックポートのインバウンドルールを有することを確実にします。詳細については、バックエンドCVMのセキュリティグループ設定をご参照ください。

関連ドキュメント

バックエンドサーバーの管理
ENIのバインド
コンテナインスタンスのバインド
ハイブリッドクラウドのデプロイ
Serverless Cloud Function（SCF）のバインド

バックエンドサーバーの管理

最終更新日：：2024-01-04 18:36:26

CLBは正常に動作しているバックエンドサーバーインスタンスにリクエストをルーティングします。CLBを初めて使用する際または業務上のニーズに応じてバックエンドサーバーの数を追加または削除したい場合は、このガイドに従って操作することができます。

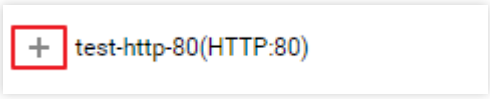
前提条件

CLBインスタンスを作成し、リスナーを設定済みであること。詳細については、CLBクイックスタートをご参照ください。

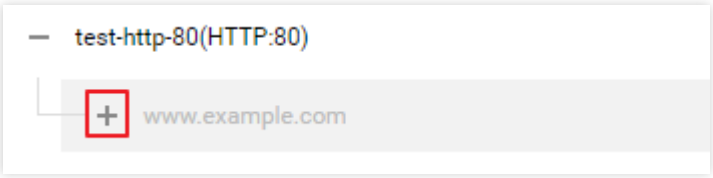
操作手順

CLBバックエンドCVMの追加

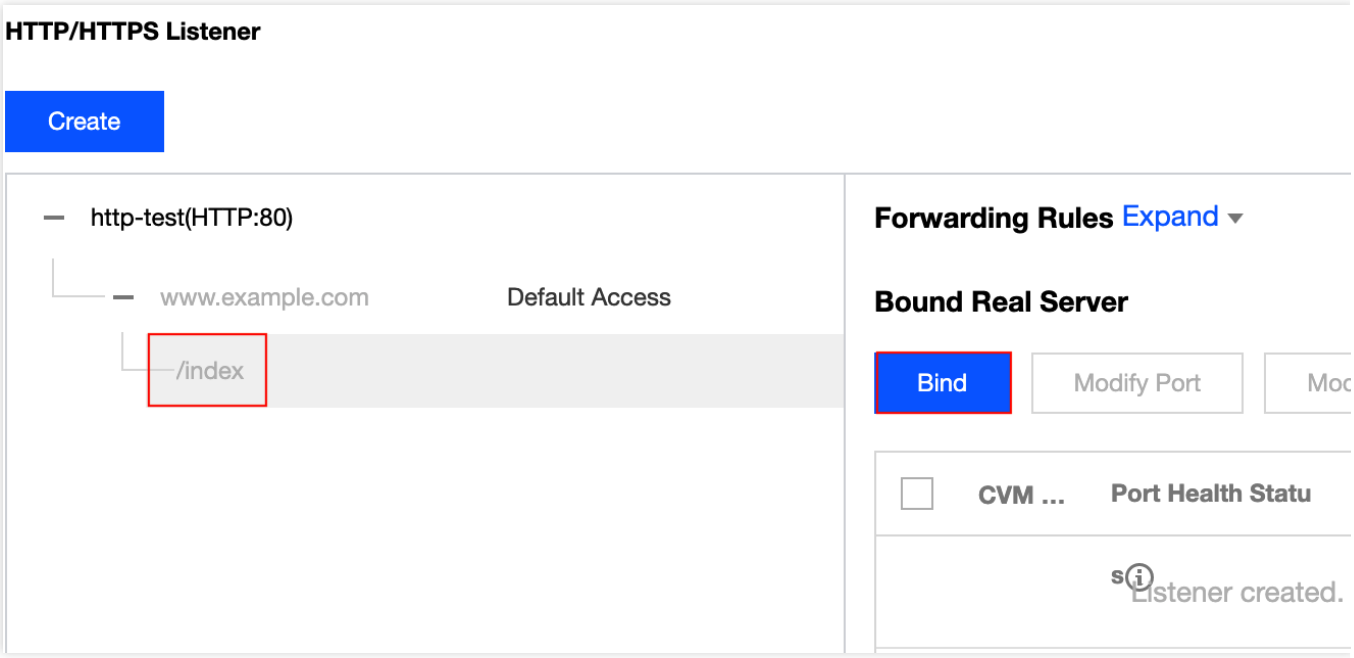
説明：
CLBインスタンスがある自動スケーリンググループに関連付けられている場合、このグループのCVMがCLBのバックエンドCVMに自動的に追加されます。自動スケーリンググループからあるCVMインスタンスが削除されると、このCVMインスタンスはCLBのバックエンドCVMからも自動的に削除されます。
APIを使用してCLBにバックエンドサーバーを追加したい場合は、バックエンドサーバーのCLBへのバインドインターフェースの説明をご参照ください。
アカウントタイプが従来型アカウントタイプであり、かつインスタンスのキャリアタイプがチャイナモバイル、チャイナテレコムまたはチャイナユニコムの場合は、ネットワーク課金モデルがトラフィック課金および共有帯域幅パッケージのCVMに限りバインドできます。アカウントタイプの詳細についてはアカウントタイプの判断を、キャリアタイプの詳細についてはキャリアタイプをそれぞれご参照ください。
1. CLBコンソールにログインします。
2. 「インスタンス管理」ページの「CLB」タブで、目的のCLBインスタンスの右側にある操作列の**リスナーの設定**をクリックします。
3. リスナー設定モジュールで、バックエンドCVMをバインドしたいリスナーを選択します。
HTTP/HTTPSリスナー
3.1.1 HTTP/HTTPSリスナーエリアで、目的のリスナーの左側にある+をクリックします。



3.1.2 表示されたドメイン名の左側にある+をクリックします。



3.1.3 表示されたURLパスを選択し、**バインド**をクリックします。



TCP/UDP/TCP SSLリスナー

TCP/UDP/TCP SSLリスナーモジュールの左側のリストから、バインドしたいバックエンドCVMのリスナーを選択し、**バインド**をクリックします。

4. CLBインスタンスにバックエンドサービスをバインドします。

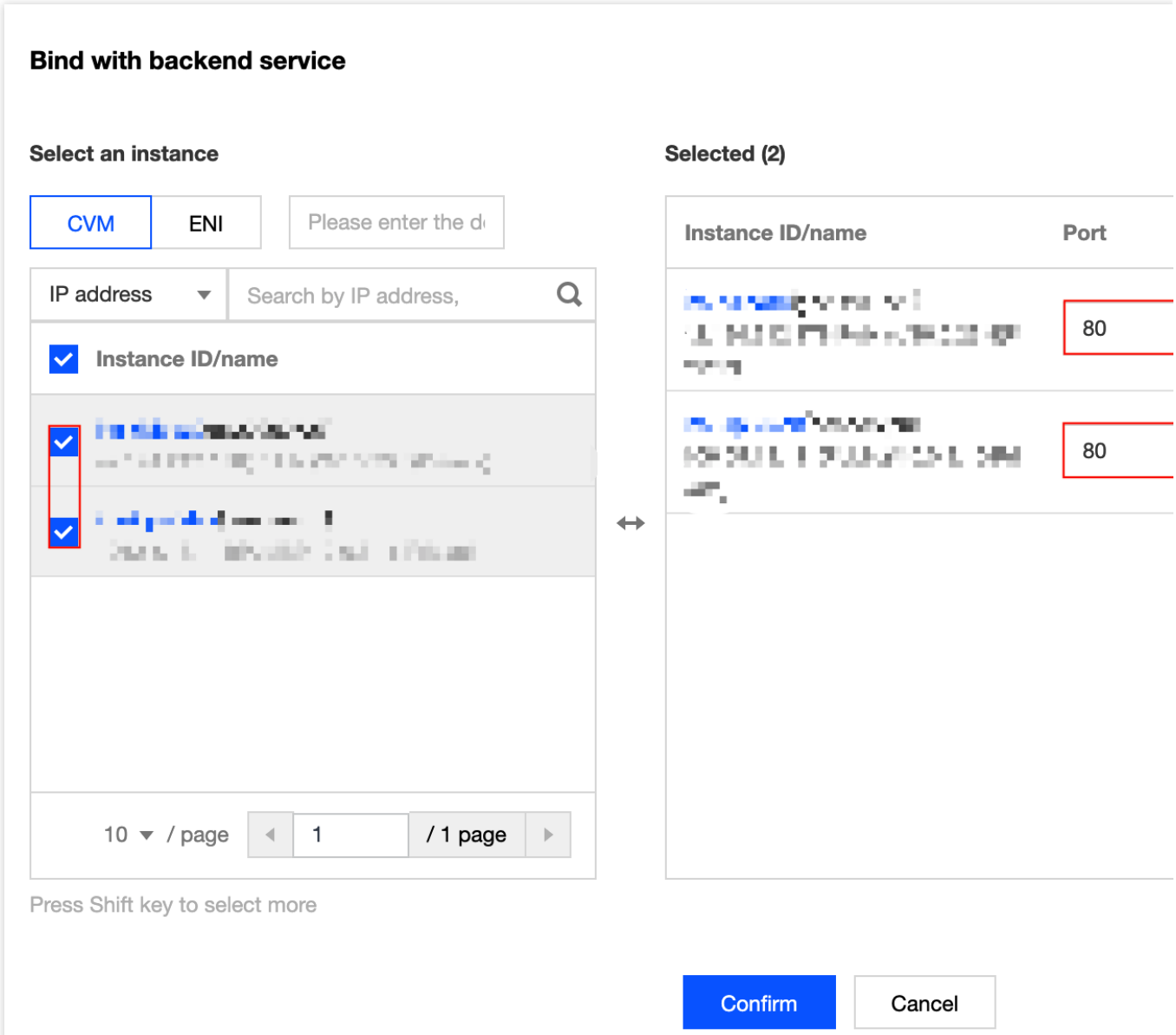
方法1：「バックエンドサービスのバインド」ポップアップボックスで**CVM**をクリックし、関連付けたいCVM（複数選択可）を選択し、関連のCVMの、転送を希望するポートと重みを入力します。詳細については、サーバーの一般的なポートをご参照ください。その後、**OK**をクリックします。

説明：

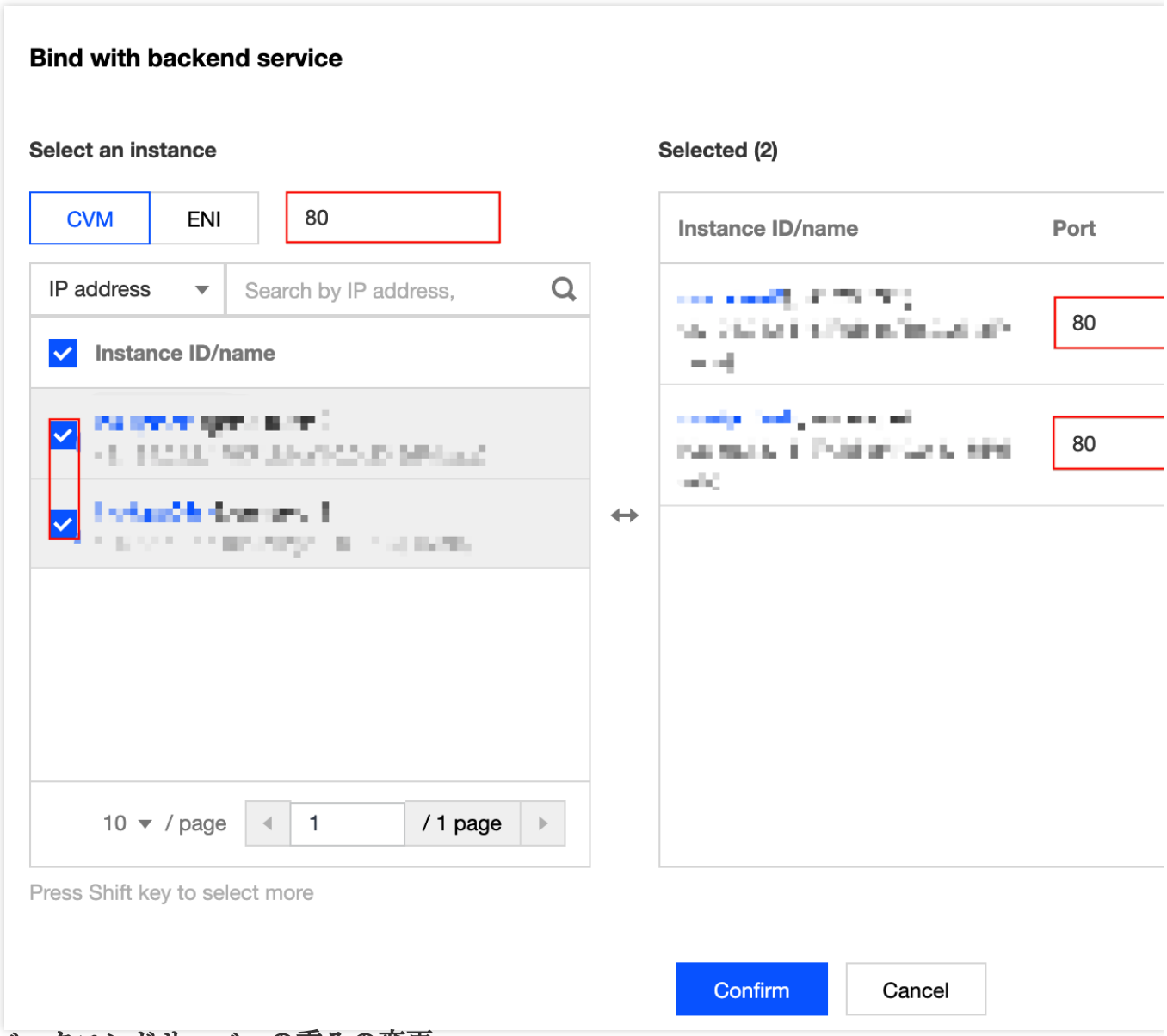
「バックエンドサービスのバインド」ポップアップボックスには、同一のリージョン、同一のネットワーク環境の、隔離されていない、期限切れではない、帯域幅（ピーク値）が0ではない、選択可能なCVMのみが表示されます。

複数のバックエンドサーバーをバインドする場合、CLBはHashアルゴリズムによってトラフィックを転送することで負荷分散の役割を果たします。

重みが高いほど、転送されるリクエストの数は多くなります。デフォルトは10、設定可能範囲は0～100です。 重みを0に設定すると、そのサーバーは新しいリクエストを受信しなくなります。セッション維持を有効にしていると、バックエンドサーバーのリクエストが不均一になる可能性があります。詳細については、バランシングアルゴリズムの選択と重みの設定の例をご参照ください。



方法2：サーバーを一括でバインドし、かつあらかじめ設定したポート値と一致させたい場合は、「バックエンドサービスのバインド」ポップアップボックスで**CVM**をクリックし、デフォルトのポート値を入力し（ポートの選択については**サーバーの一般的なポート**をご参照ください）、関連のサーバーにチェックを入れて重み値を設定し、**OK**をクリックします。



CLBバックエンドサーバーの重みの変更

CVMに転送されるリクエストの相対数はバックエンドサーバーの重みによって決まります。バックエンドCVMをバインドする際に、重みの情報をあらかじめ設定する必要があります。次は「HTTP/HTTPSリスナー」を例に（TCP/UDP/TCP SSLリスナーの変更方法も同様です）、CLBバックエンドサーバーの重みを変更する方法についてご説明します。

説明：

APIを使用してCLBバックエンドサーバーの重みを変更したい場合は、CLBバックエンドサーバーの重みの変更インターフェースの説明をご参照ください。

CLBバックエンドサーバーの重みに関するその他の情報については、CLBのポーリング方式をご参照ください。

1. CLBコンソールにログインします。
2. 「インスタンス管理」ページの「CLB」タブで、目的のCLBインスタンスの右側にある操作列の**リスナーの設定**をクリックします。

3. HTTP/HTTPSリスナーモジュールの左側のリストで、インスタンスとリスナーのルールを表示し、URLパスを選択します。

4. HTTP/HTTPSリスナーモジュールの右側のサーバーリストで、関連のサーバーの重みを変更します。

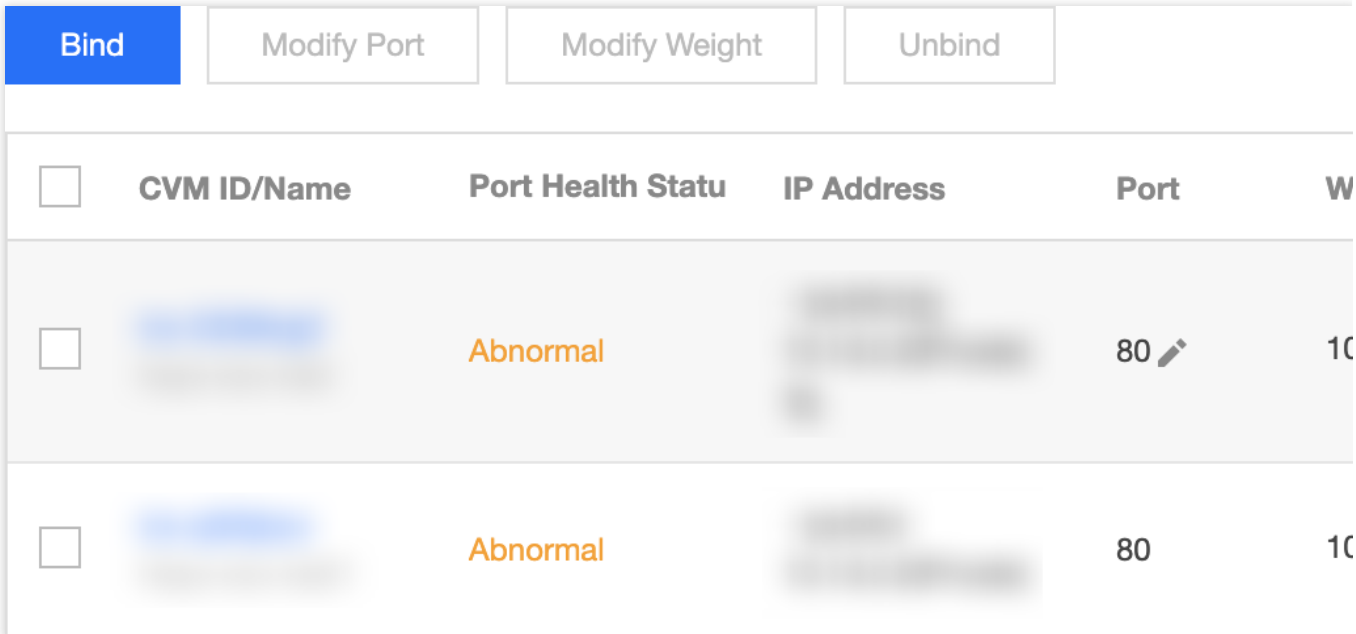
説明：

重みが高いほど、転送されるリクエストの数は多くなります。デフォルトは10、設定可能範囲は0〜100です。 重みを0に設定すると、そのサーバーは新しいリクエストを受信しなくなります。セッション維持を有効にしていると、バックエンドサーバーのリクエストが不均一になる可能性があります。詳細については、バランシングアルゴリズムの選択と重みの設定の例をご参照ください。

方法1：あるサーバーの重みを単独で変更します。

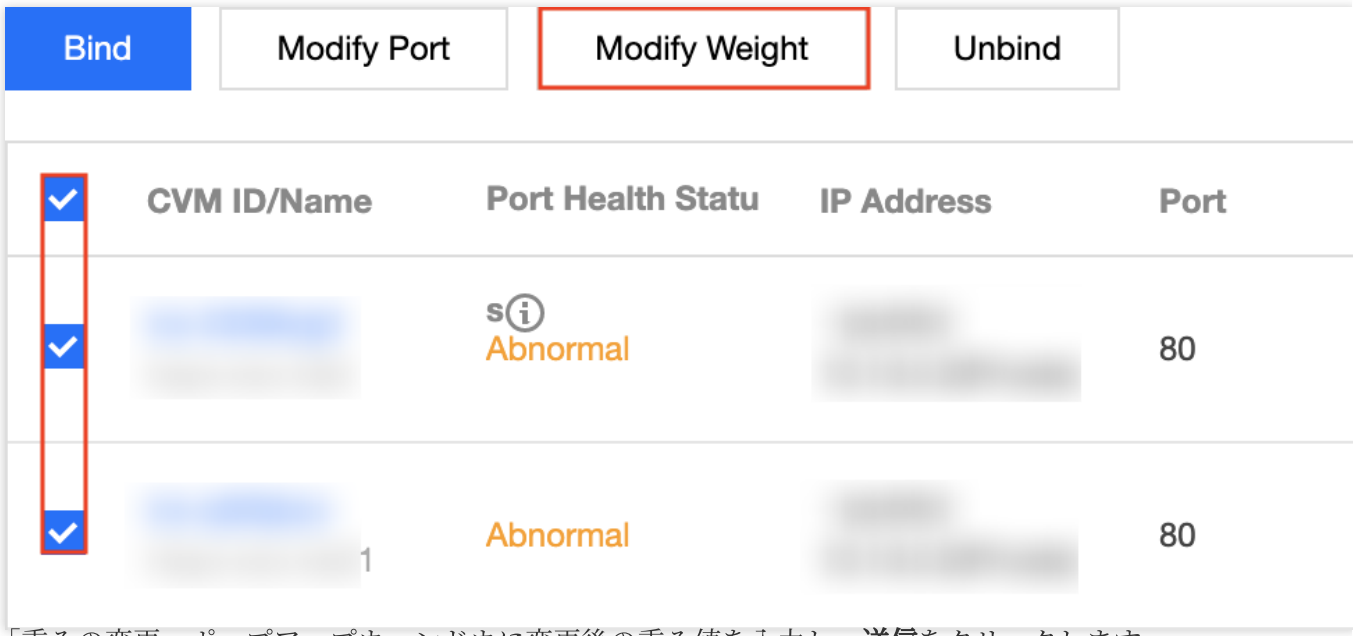
4.1.1 重みを変更したいサーバーを見つけ、対応する重みの上にマウスを合わせ、編集ボタン

をクリックします。



4.1.2 「重みの変更」ポップアップウィンドウに変更後の重み値を入力し、**送信**をクリックします。

方法2：いくつかのサーバーの重みを一括変更します。
説明：
一括変更後のサーバーの重みはすべて同じになります。
4.1.1 サーバーの前にあるチェックボックスをクリックして複数のサーバーを選択し、リストの上方で**重みの変更**をクリックします。



4.1.2 「重みの変更」ポップアップウィンドウに変更後の重み値を入力し、**送信**をクリックします。

CLBバックエンドサーバーポートの変更

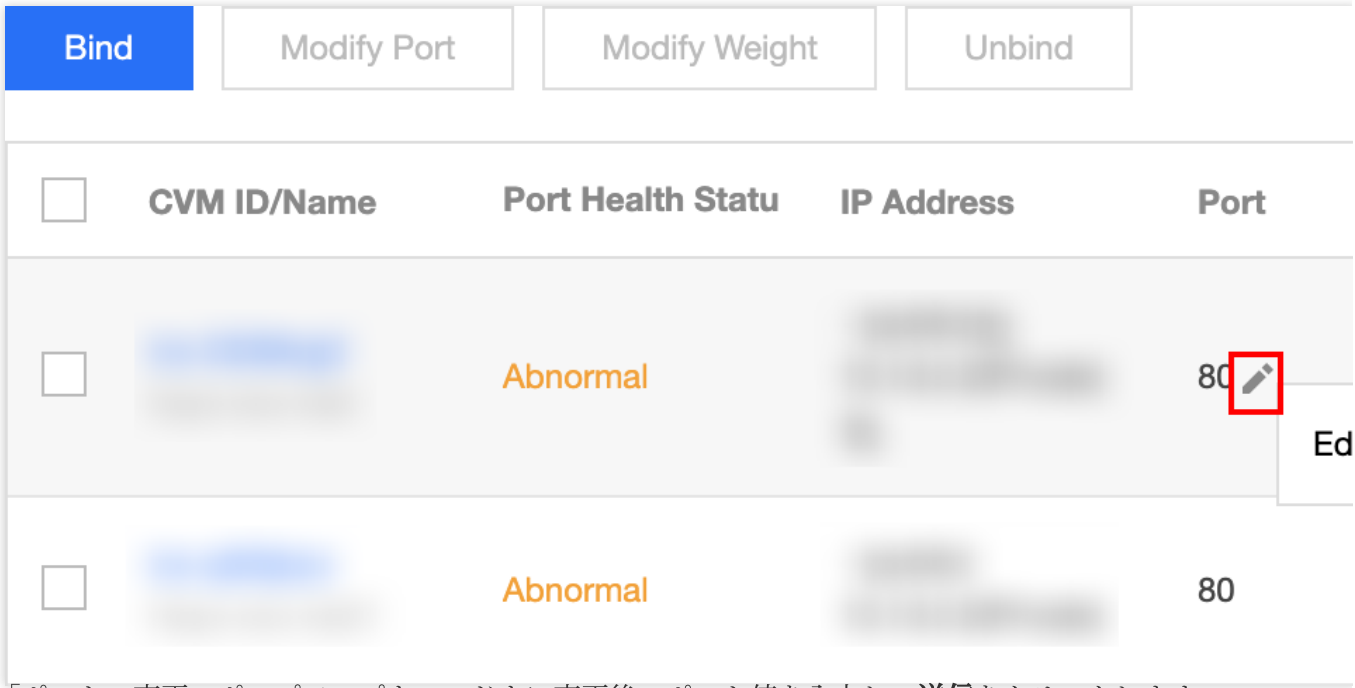
CLBコンソールはバックエンドサーバーポートの変更をサポートしています。次は「HTTP/HTTPSリスナー」を例に（TCP/UDP/TCP SSLリスナーの変更方法も同様です）、CLBバックエンドサーバーのポートを変更する方法についてご説明します。
説明：
APIを使用してCLBバックエンドサーバーポートを変更したい場合は、リスナーにバインドしたバックエンドマシンのポートの変更インターフェースの説明をご参照ください。
1. CLBコンソールにログインします。
2. 「インスタンス管理」ページの「CLB」タブで、目的のCLBインスタンスの右側にある操作列の**リスナーの設定**をクリックします。
3. HTTP/HTTPSリスナーモジュールの左側のリストで、インスタンスとリスナーのルールを表示し、URLパスを選択します。

4. HTTP/HTTPSリスナーモジュール右側のサーバーリストで、関連のサーバーポートを変更します。ポートの選択については、サーバーの一般的なポートをご参照ください。

方法1：あるサーバーのポートを単独で変更します。

4.1.1 ポートを変更したいサーバーを見つけ、対応するポートの上にマウスを合わせ、編集ボタン

をクリックします。



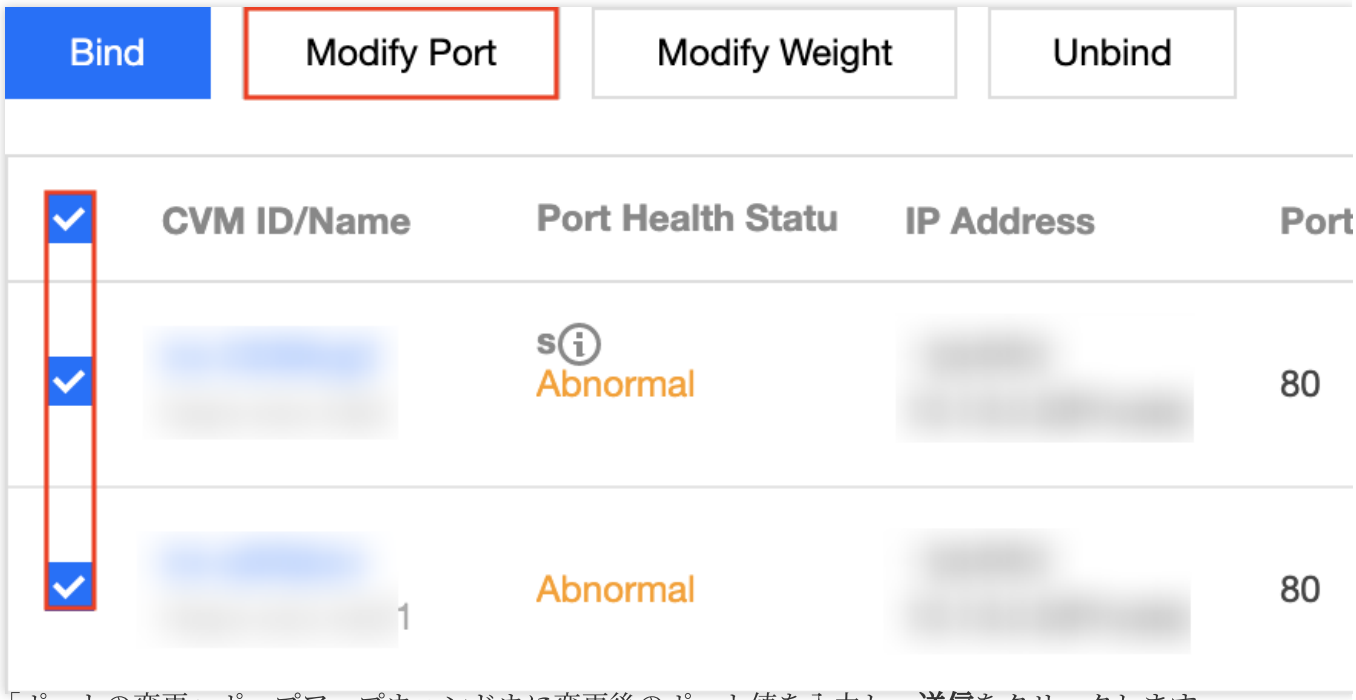
4.1.2 「ポートの変更」ポップアップウィンドウに変更後のポート値を入力し、**送信**をクリックします。

方法2：いくつかのサーバーのポートを一括変更します。

説明：

一括変更後のサーバーポートはすべて同じになります。

4.1.3 サーバーの前にあるチェックボックスをクリックして複数のサーバーを選択し、リストの上方で**ポートの変更**をクリックします。



4.1.4 「ポートの変更」ポップアップウィンドウに変更後のポート値を入力し、**送信**をクリックします。

CLBバックエンドサーバーのバインド解除

CLBコンソールはバインド済みのバックエンドサーバーのバインド解除をサポートしています。次は「HTTP/HTTPSリスナー」を例に（TCP/UDP/TCP SSLリスナーのバインド解除方法も同様です）、バインド済みのCLBバックエンドサーバーのバインドを解除する方法についてご説明します。

説明：

バックエンドサーバーのバインドを解除すると、CLBインスタンスとCVMインスタンスとの関連付けが解除され、CLBからのリクエスト転送はその時点で停止します。

バックエンドサーバーのバインドを解除しても、CVMのライフサイクルには影響はありません。再度バックエンドサーバークラスターに追加することもできます。

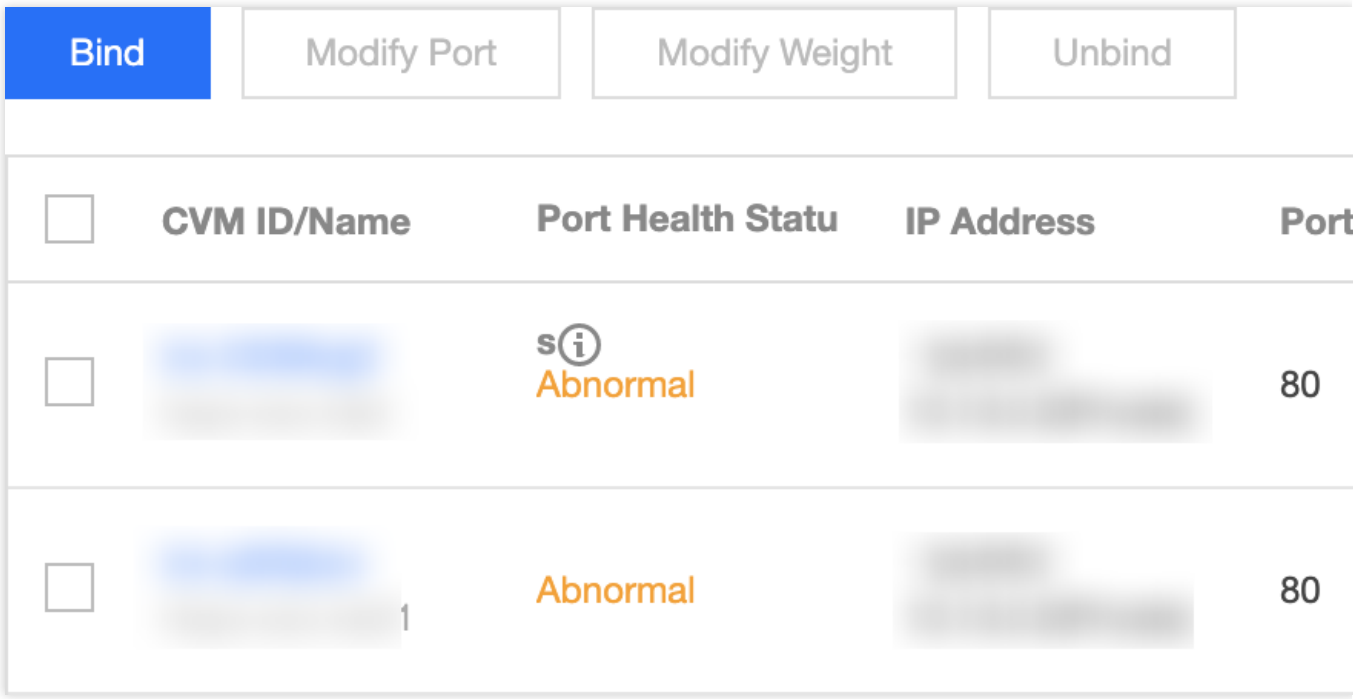
APIを使用してCLBバックエンドサーバーのバインドを解除したい場合は、CLBリスナーとバックエンドサービスのバインド解除インターフェースの説明をご参照ください。

1. CLBコンソールにログインします。
2. 「インスタンス管理」ページの「CLB」タブで、目的のCLBインスタンスの右側にある操作列の**リスナーの設定**をクリックします。
3. HTTP/HTTPSリスナーモジュールの左側のリストで、インスタンスとリスナーのルールを表示し、URLパスを選択します。

4. HTTP/HTTPSリスナーモジュールの右側のサーバーリストで、バインド済みのバックエンドサーバーのバインドを解除します。

方法1：あるサーバーのバインドを単独で解除します。

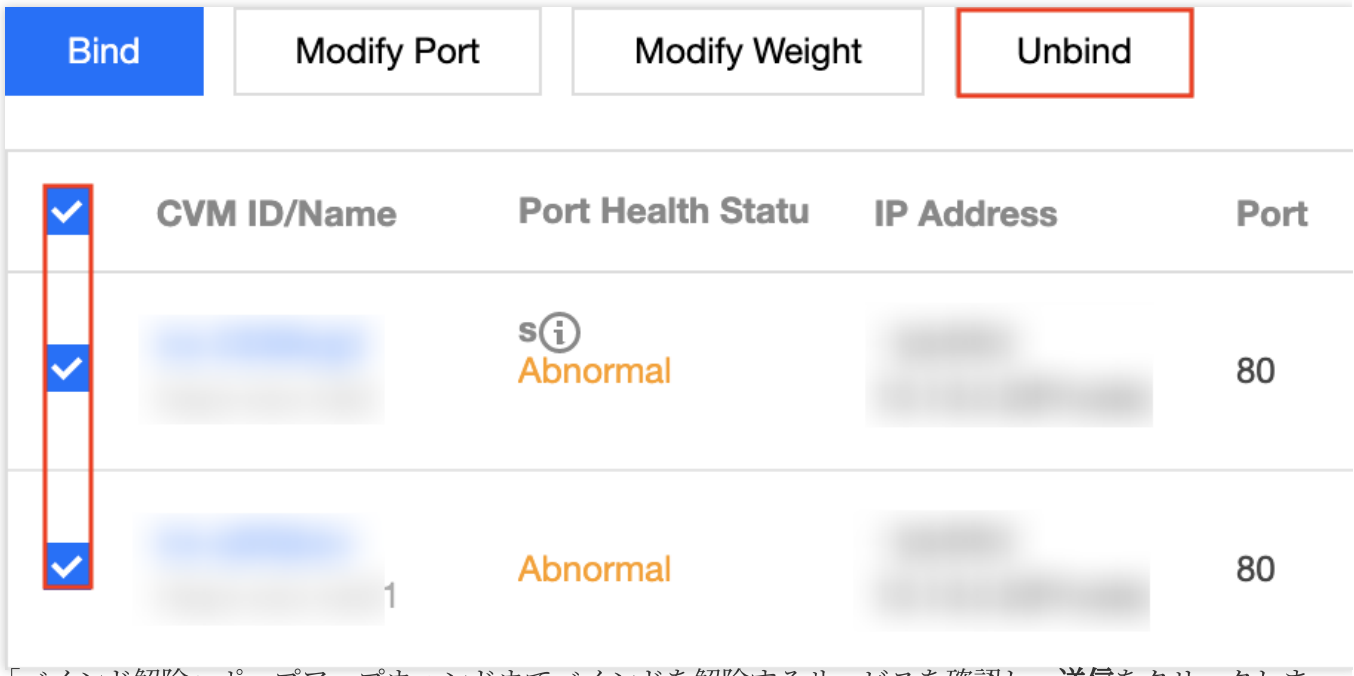
4.1.1 バインドを解除したいサーバーを見つけ、その右側の操作バーで**バインド解除**をクリックします。



4.1.2 「バインド解除」ポップアップウィンドウでバインドを解除するサービスを確認し、**送信**をクリックします。

方法2：いくつかのサーバーのバインドを一括解除します。

4.1.3 サーバーの前にあるチェックボックスをクリックして複数のサーバーを選択し、リストの上方で**バインド解除**をクリックします。



4.1.4 「バインド解除」ポップアップウィンドウでバインドを解除するサービスを確認し、**送信**をクリックします。

ENIのバインド

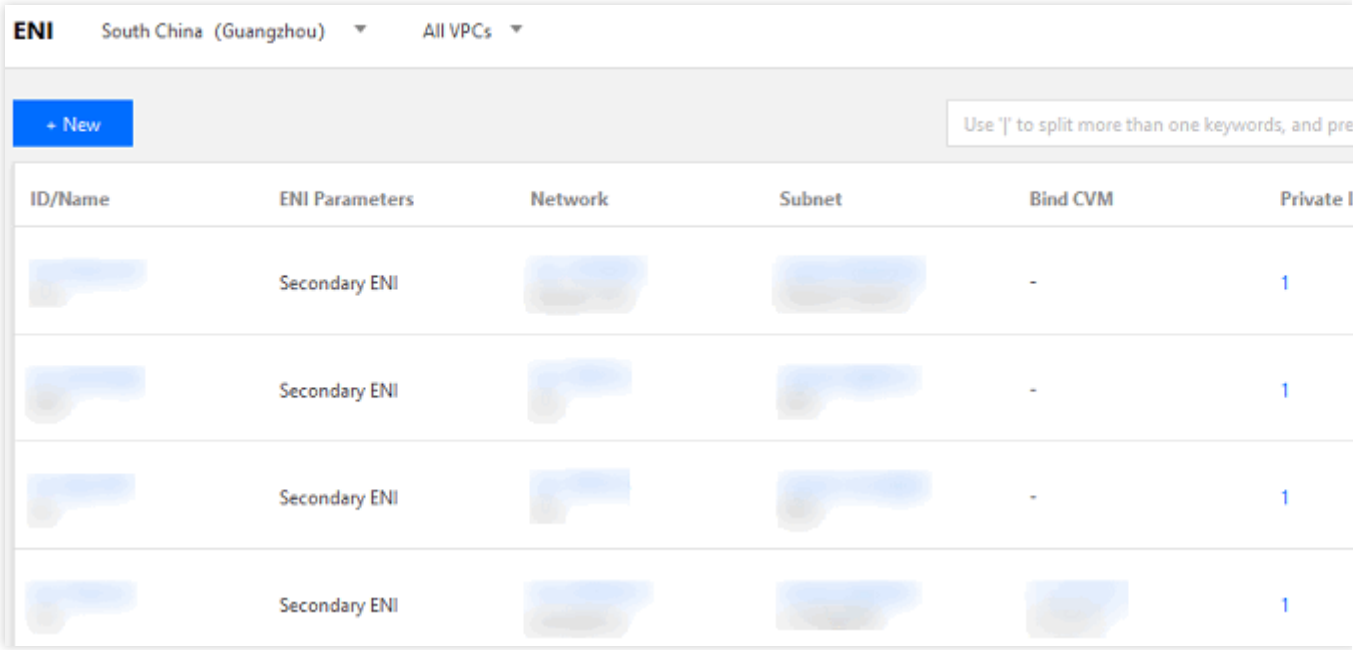
最終更新日：：2024-01-04 18:36:26

Elastic Network Interfaceの概要

Elastic Network Interface（ENI）は、VPC内のCVMインスタンスをバインドすることができる仮想ネットワークカードです。ENIは、同じVPC、アベイラビリティーゾーンにおけるCVM間で、自由に移行することができます。ENIによって、高可用性のクラスターの構築、低コストのフェイルオーバーおよび精細化されたネットワーク管理を実現することができます。
CLBのバックエンドサービスは、CVMおよびENIをサポートしています。すなわち、CLBは、CVMとENIのバインドをサポートしています。CLBとバックエンドサービス間では、プライベートネットワーク通信を使用します。CLBが複数のCVMとENIをバインドした場合、アクセストラフィックはCVMのプライベートIPとENIのプライベートIPに転送されます。

前提条件

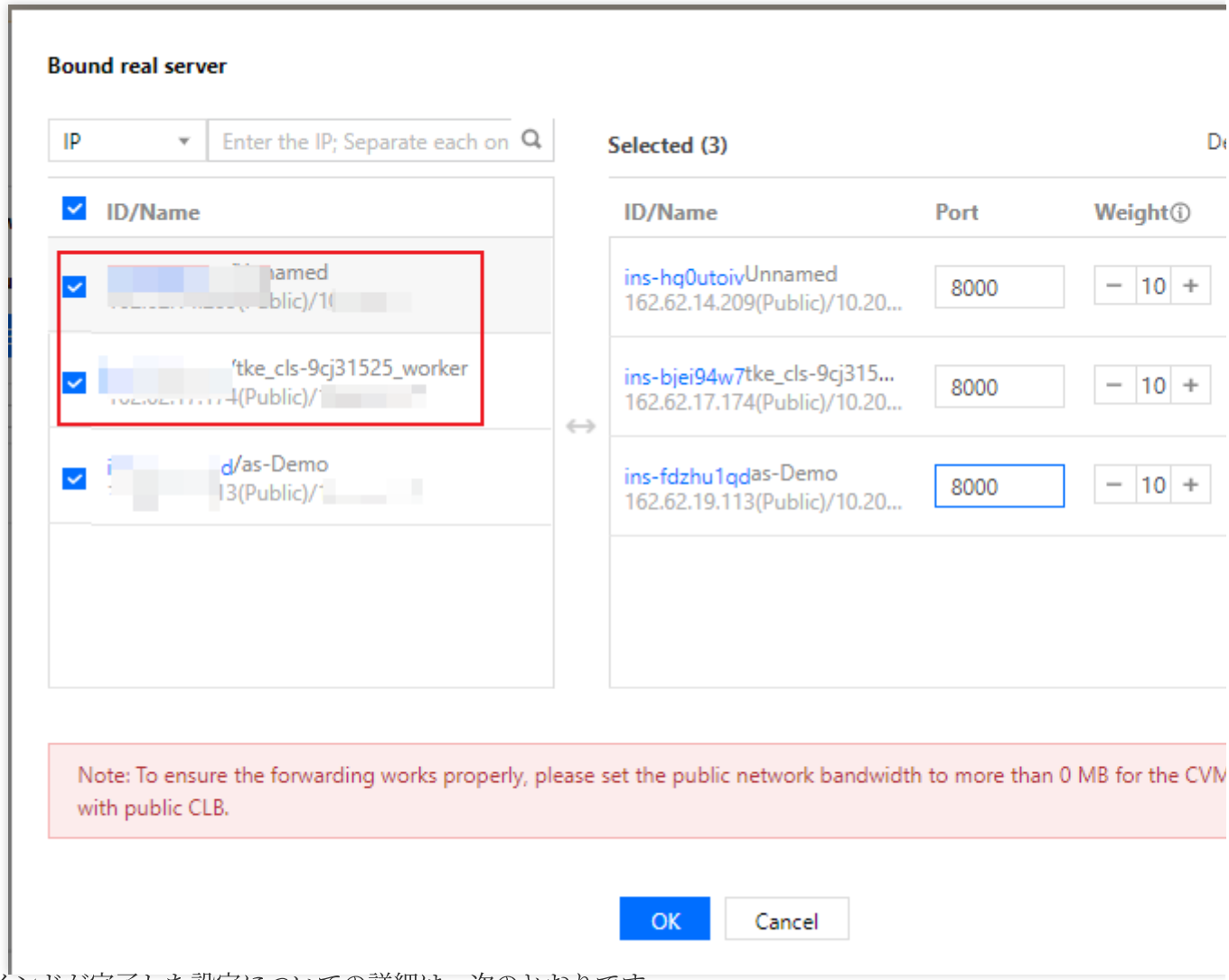
ENIが、あるCVMを先にバインドすることによって、CLBはそのENIをバインドすることができます。CLBは、CLBによるトラフィックの転送のみを行い、ビジネスロジックの実際の処理は行わないため、コンピューティングリソースのCVMインスタンスによってユーザーのリクエストを処理する必要があります。まずはENIコンソールに進み、必要なENIとCVMにバインドを行ってください。



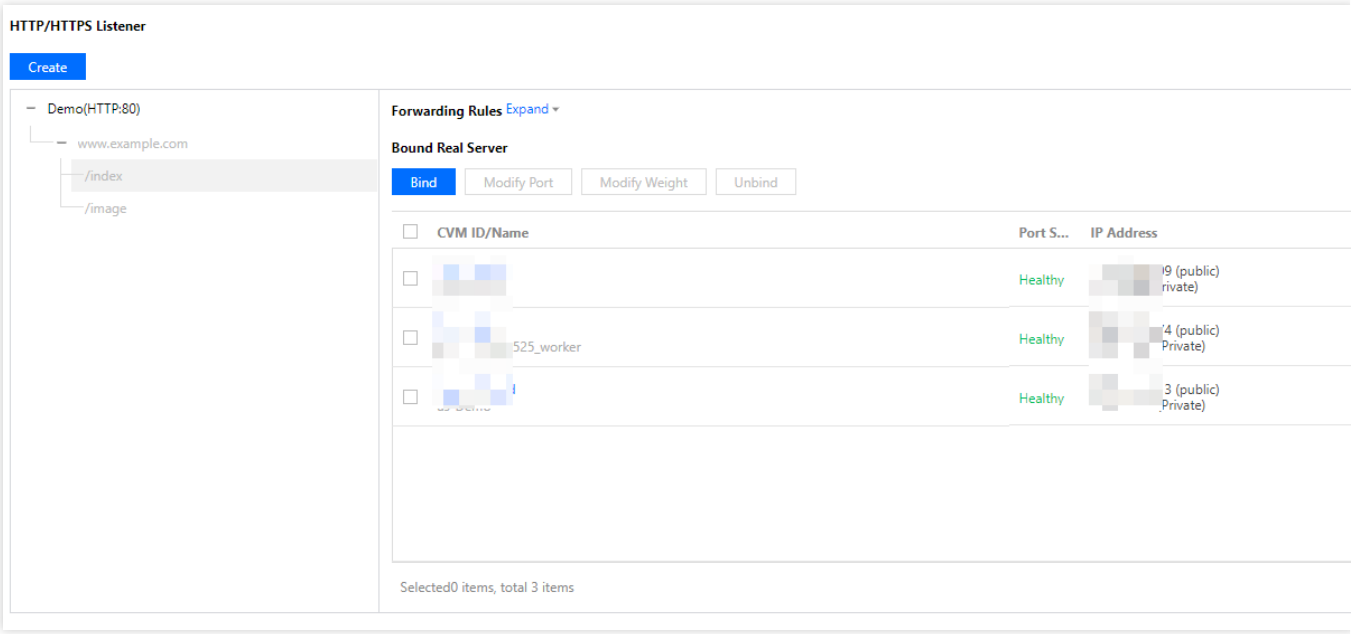
操作手順

1. 先にCLBリスナーの設定をする必要があります。詳細については、CLBリスナーの概要をご参照ください。
2. 作成が完了したリスナーの左側の**+ **をクリックし、ドメイン名およびURLパスを表示します。具体的なURLパスを選択すると、リスナーの右側からバインド済みのバックエンドサービスを確認することができます。

3. **バインド**をクリックすると、ポップアップボックスでバインドしたいバックエンドサーバーを選択し、サービスポートと重みを設定することができます。バックエンドサービスをバインドする場合は、「CVM」または「ENI」を選択することができます。
CVM：CLBと共に、VPC下のすべてのCVMプライマリネットワークカードの主なプライベートIPをバインドすることができます。
ENI：CLBと共に、VPC下のCVMプライマリネットワークカードの主なプライベートIPを除く、プライマリネットワークカードのセカンダリプライベートIP、セカンダリネットワークカードのプライベートIPなどのすべてのENI IPをバインドすることができます。ENIのIPタイプの詳細については、ENI-関連コンセプトをご参照ください。



4. バインドが完了した設定についての詳細は、次のとおりです。



Serverless Cloud Function（SCF）のバインド

最終更新日：：2024-01-04 18:36:26

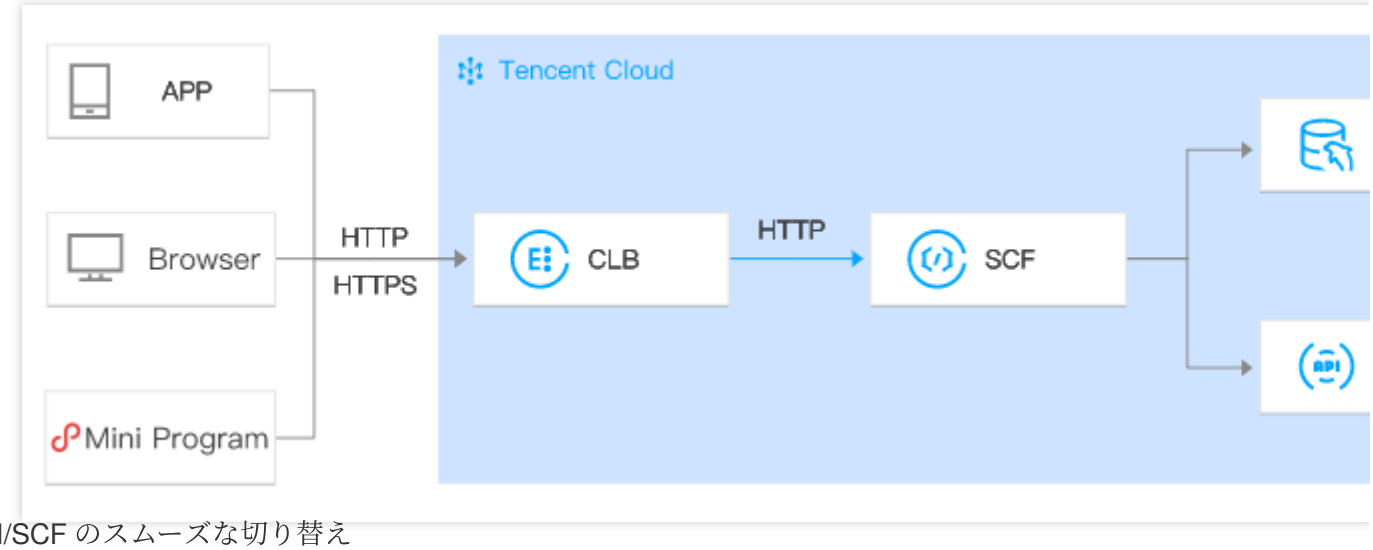
Serverless Cloud Function（SCF）の作成によってバックエンドWebサービスを実装後、CLBを使用してSCFをバインドし、外部にサービスを提供することができます。

背景情報

Serverless Cloud Function（SCF）はTencent Cloudが企業および開発者向けにご提供するサーバーレスな実行環境であり、これによってサーバーを購入および管理することなくコードを実行できます。SCFを作成すると、CLBトリガーを作成することでSCFとイベントを関連付けることができます。CLBトリガーはリクエスト内容をパラメータ形式でSCFに伝達し、SCFからの戻り値をレスポンスとしてリクエスト側に返します。

ユースケース

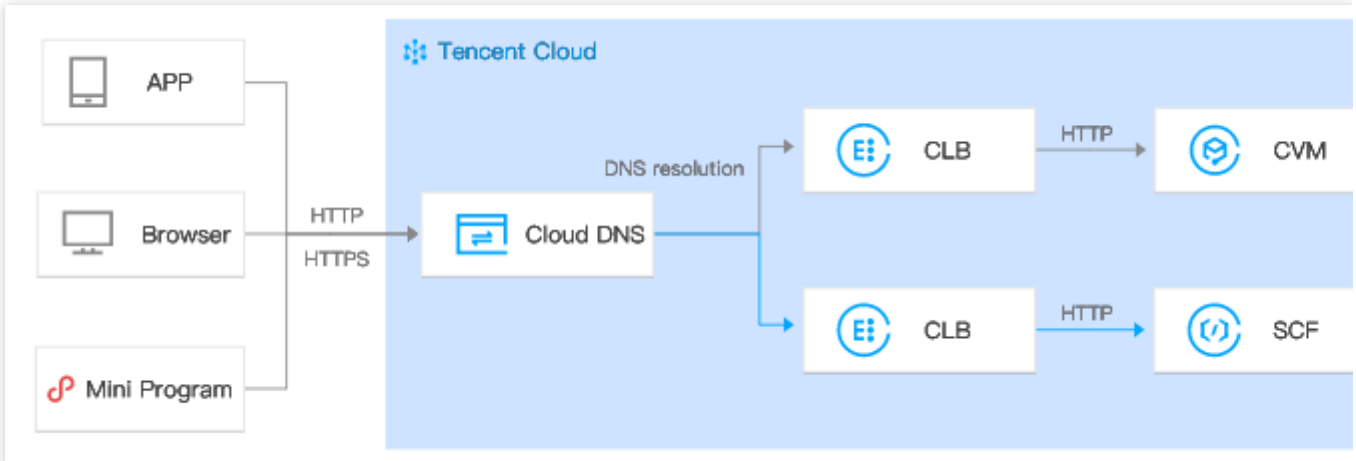
一般的な HTTP/HTTPS 接続
eコマース、ソーシャルネットワーキング、ツールなどのAppアプリケーション、個人ブログ、イベントページなどのWebアプリケーションのシーンなどに適しています。方法のフローは次のとおりです。
1. App、ブラウザ、H5、ミニプログラムなどからHTTP/HTTPSリクエストを送信し、CLBを介してSCFにアクセスします。
2. CLBによって証明書をアンインストールします。SCFはHTTPサービスの提供のみ必要です。
3. リクエストをSCFに転送し、続いてクラウドデータベースへの書き込みやその他のAPIの呼び出しなど、その後の処理を行います。



CVM/SCF のスムーズな切り替え

HTTP/HTTPSサービスをCVMからSCFに移行するシーン、CVM（SCF）サービスに問題が生じた場合にSCF（CVM）にスピーディーに移行するフェイルオーバーのシーンに適しています。方法のフローは次のとおりです。

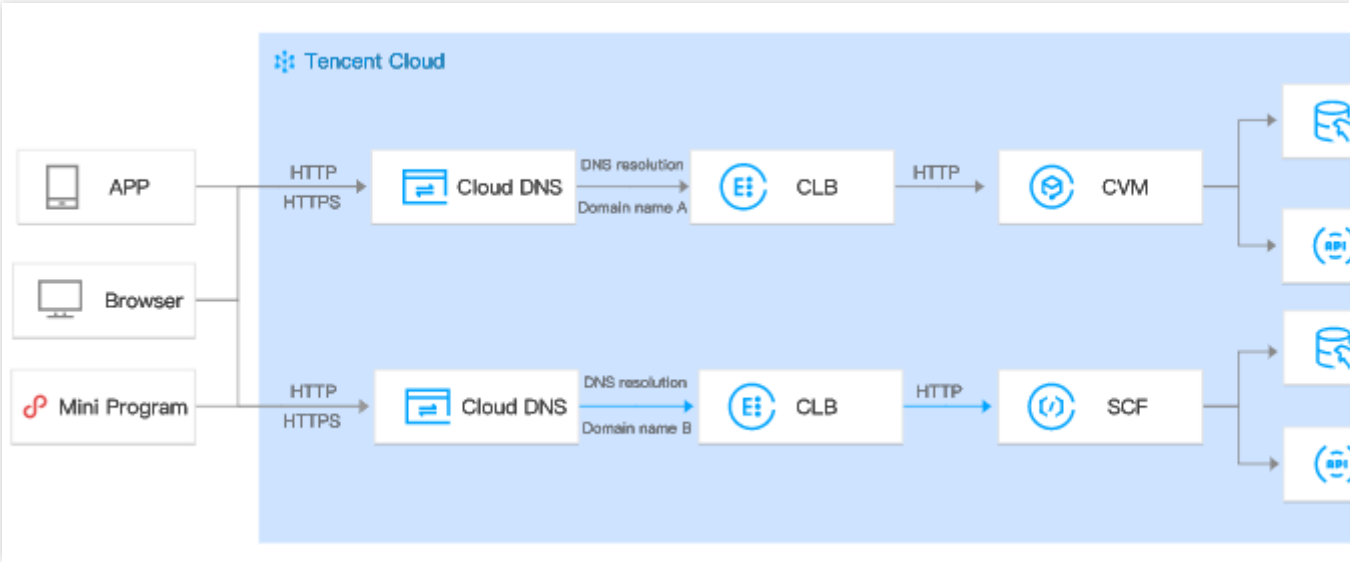
1. App、ブラウザ、H5、ミニプログラムなどからHTTP/HTTPSリクエストを送信します。
2. DNS解決によって、リクエストをCLBのVIPに解決します。
3. 1台のCLBからはリクエストをCVMに転送し、もう1台のCLBからはリクエストをSCFに転送します。
4. クライアントに影響することなく、CVMとSCFとの間でバックエンドサービスをスムーズに切り替えることができます。



CVM/SCF 業務分離

タイムセール、買い占めなどのシーンに適しています。高い弾力性が求められるサービスの処理にはSCFを、日常業務の処理にはCVMを使用します。

1. DNS解決によって、ドメイン名Aを1台のCLBのVIPに、ドメイン名Bをもう1台のCLBのVIPに、それぞれ解決します。
2. このうち1台のCLBからはリクエストをCVMに転送し、もう1台のCLBからはリクエストをSCFに転送します。



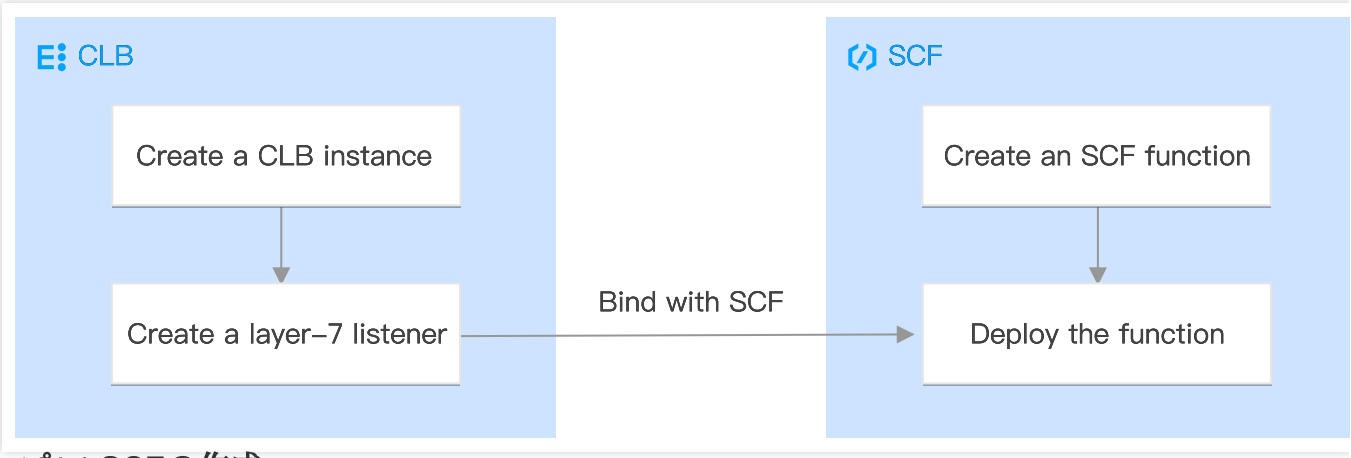
制限事項

SCFのバインドは広州、上海、北京、成都、中国香港、シンガポール、ムンバイ、東京、シリコンバレーリージョンでのみサポートしています。
SCFのバインドは標準アカウントタイプのみサポートしており、従来型アカウントタイプではサポートしていません。標準アカウントタイプへのアップグレードをお勧めします。詳細については、アカウントタイプアップグレードの説明をご参照ください。
SCFのバインドは基幹ネットワークタイプではサポートしていません。
CLBは同一リージョン下のすべてのSCFのバインドをデフォルトでサポートしています。異なるVPC間でのSCFバインドはサポート可能ですが、異なるリージョン間でのバインドはサポートしていません。
現在はIPv4、IPv6 NAT64バージョンのCLBのみSCFのバインドをサポートしています。IPv6バージョンは現時点ではサポートしていません。
SCFのバインドはレイヤー7（HTTP、HTTPS）リスナーのみサポートしており、レイヤー4（TCP、UDP、TCP SSL）リスナーおよびレイヤー7 QUICリスナーではサポートしていません。
CLBのSCFバインドは「Event関数」タイプのSCFのみサポートしています。

前提条件

1. CLBインスタンスの作成
2. HTTPリスナーの設定またはHTTPSリスナーの設定

操作手順

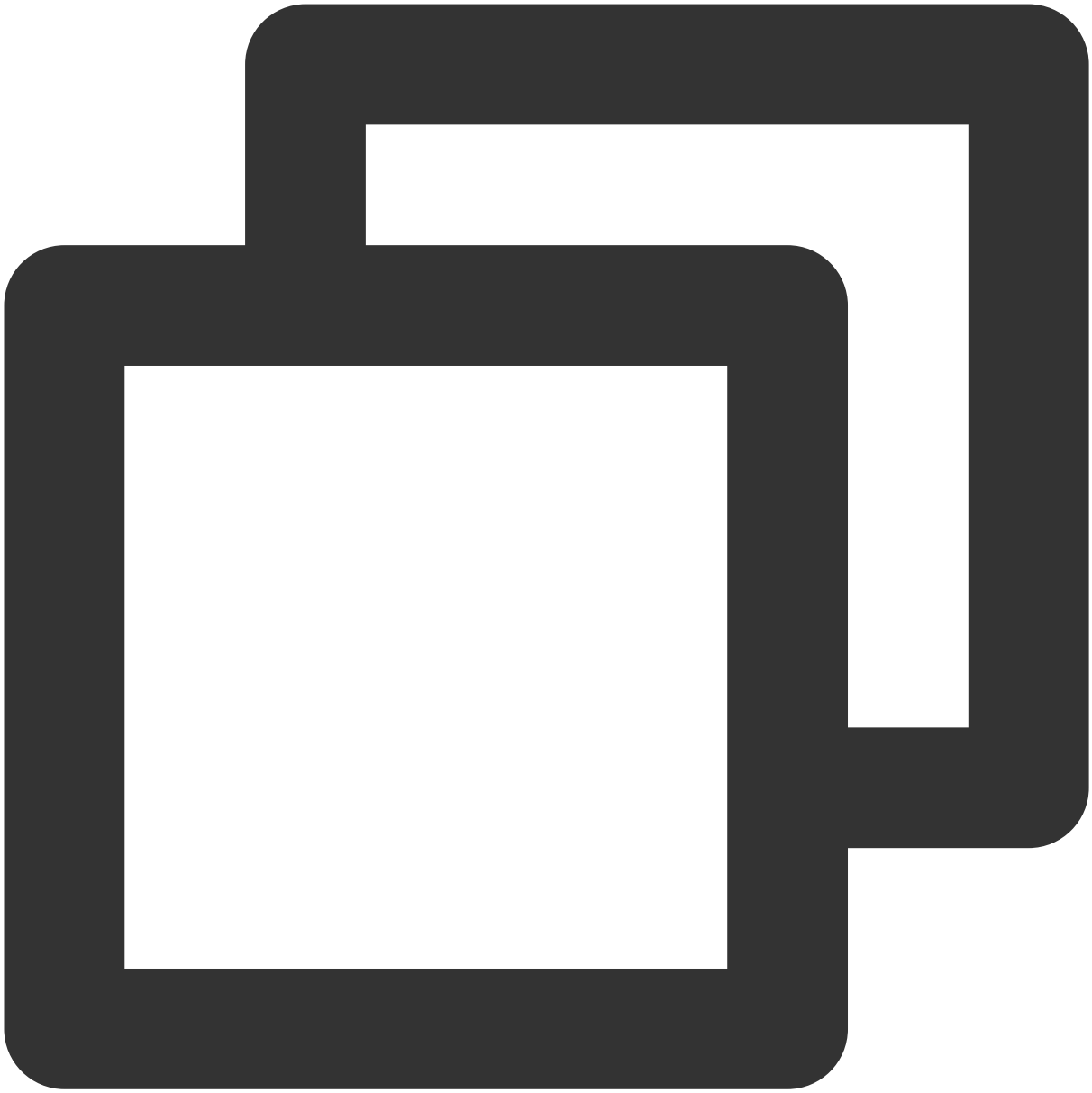


ステップ1：SCFの作成

1. SCFコンソールにログインし、左側のナビゲーションバーで【関数サービス】をクリックします。
2. 「関数サービス」ページで、【新規作成】をクリックします。
3. 関数サービスの「新規作成」ページで、作成方法は「カスタム作成」を選択し、関数名を入力し、リージョンはCLBインスタンスと同一のリージョンを、実行環境は「Python3.6」をそれぞれ選択します。関数コード入力ボックスに次のコードを入力し（ここではHello CLBを例とします）、【完了】をクリックします。

注意：

CLBにSCFをバインドする際は、特定のレスポンス統合形式によって返す必要があります。詳細については、統合レスポンスをご参照ください。



```
# -*- coding: utf8 -*-
import json
def main_handler(event, context):

 return {
     "isBase64Encoded": False,
     "statusCode": 200,
     "headers": {"Content-Type":"text/html"},
     "body": "<html><body><h1>Hello CLB</h1></body></html>"
}
```

ステップ2：SCFのデプロイ

1. 「関数サービス」ページのリストで、先ほど作成した関数名をクリックします。
2. 「関数管理」ページで、【関数コード】タブをクリックし、タブの下にある【デプロイ】をクリックします。

ステップ3：SCFのバインド

1. CLBコンソールにログインし、左側のナビゲーションバーで【インスタンス管理】をクリックします。
2. 「インスタンス管理」ページの「CLB」タブで、目的のインスタンスの右側にある「操作」列の【リスナーの設定】をクリックします。
3. HTTP/HTTPSリスナーリストで、SCFをバインドしたいリスナーを選択し、目的のリスナーの左側の【+】および表示されたドメイン名の左側の【+】をそれぞれクリックし、表示されたURLパスを選択して【バインド】をクリックします。
4. ポップアップした「バックエンドサービスのバインド」ダイアログボックスで、ターゲットタイプに「SCF」を選択し、ネームスペース、関数名およびバージョン/エイリアスを選択し、重みを設定して【確定】をクリックします。
5. 「リスナーの管理」タブに戻り、「転送ルールの詳細」エリアでCLBにバインド済みのSCF、すなわち作成済みのCLBトリガーを表示します。

説明：

SCFコンソールでCLBトリガーを作成し、CLBとSCFをバインドする方法も選択できます。詳細については、トリガーの作成をご参照ください。

結果の検証

1. SCFコンソールにログインし、左側のナビゲーションバーで【関数サービス】をクリックします。
2. 「関数サービス」ページのリストで、先ほど作成した関数名をクリックします。
3. 関数のページで、左側のリストの【トリガーの管理】をクリックします。
4. 「トリガーの管理」ページのトリガーで、アクセスパスをクリックします。
5. ブラウザでこのアクセスパスを開き、「Hello CLB」が表示された場合、関数のデプロイは成功です。

関連ドキュメント

SCF関数の作成

コンテナインスタンスのバインド

最終更新日：：2024-01-04 18:36:26

CLBのバックエンドサービスは、コンテナのインスタンスのバインドをサポートしています。

コンテナインスタンスの概要

EKS Container Instance（EKSCI）は、Elastic Kubernetes Service（EKS）が提供している、ユーザーがサーバーを購入することなく、また、K8Sクラスターをデプロイすることなく、すぐにコンテナのアプリケーションをデプロイすることができるサービス方式です。仮想マシンレベルのセキュリティ分離およびリソース隔離が提供され、すぐに使用できると同時に、仮想マシンよりも高速な起動速度およびリリース速度が提供されます。

Kubernetesクラスターと比較した場合、EKSCIはそのなかのPodに相当し、よりシンプルでより基本的なコンテナ化されたソリューションです。上位のワークロードのオーケストレーションやスケジューリングなどの管理機能を必要とせず、コンテナのリソースのスケジューリングと管理だけを必要としているのであれば、EKSCIを選択することは、より経済的で効率的です。EKSCIによって、下層のサーバー側の運用保守および管理業務を省くことができるため、アプリケーション層に集中することができるようになり、効率の向上とコストの節約ができます。
説明：
コンテナインスタンスは、現在ベータ版テスト中です。使用が必要な場合は、チケットを提出して、申請してください。

制限事項

CLBインスタンスタイプのみが、コンテナインスタンスのバインドをサポートしています。従来型CLBはサポートしていません。
VPCネットワークタイプのみが、コンテナインスタンスのバインドをサポートしています。基幹ネットワークはサポートしていません。
クロスリージョンバインディング2.0、ハイブリッドクラウドのデプロイは、すべてコンテナインスタンスのバインドをサポートしています。
レイヤー4、レイヤー7のリスナーは、すべてコンテナインスタンスのバインドをサポートしています。

前提条件

すでにチケットを提出して、コンテナインスタンスサービスのアクティブ化を申請していることが必要です。

CLBリスナーを作成済みである場合、TCPリスナーを例にした詳細について、TCPリスナーの設定をご参照ください。

操作手順

1. CLBコンソールにログインし、左側ナビゲーションバーの**インスタンス管理**をクリックします。
2. **インスタンス管理**ページの**CLB**タブで、目的のインスタンスの右側にある**操作**列の**リスナーの設定**をクリックします。
3. TCPリスナーリストから目的のリスナーを選択して、右側の**バインド**をクリックします。
4. ポップアップされた**バックエンドサービスのバインド**ダイアログボックスで、ターゲットタイプから**コンテナインスタンス**を選択し、バインド待ちのコンテナインスタンスにチェックを入れ、ポートと重みを設定後、**確定**をクリックします。

説明：
その他のVPCのコンテナインスタンスをバインドしたい場合は、その他のVPCを、このVPCと同じCCNインスタンスにバインドする必要があります。詳細については、ネットワークインスタンスのバインドをご参照ください。

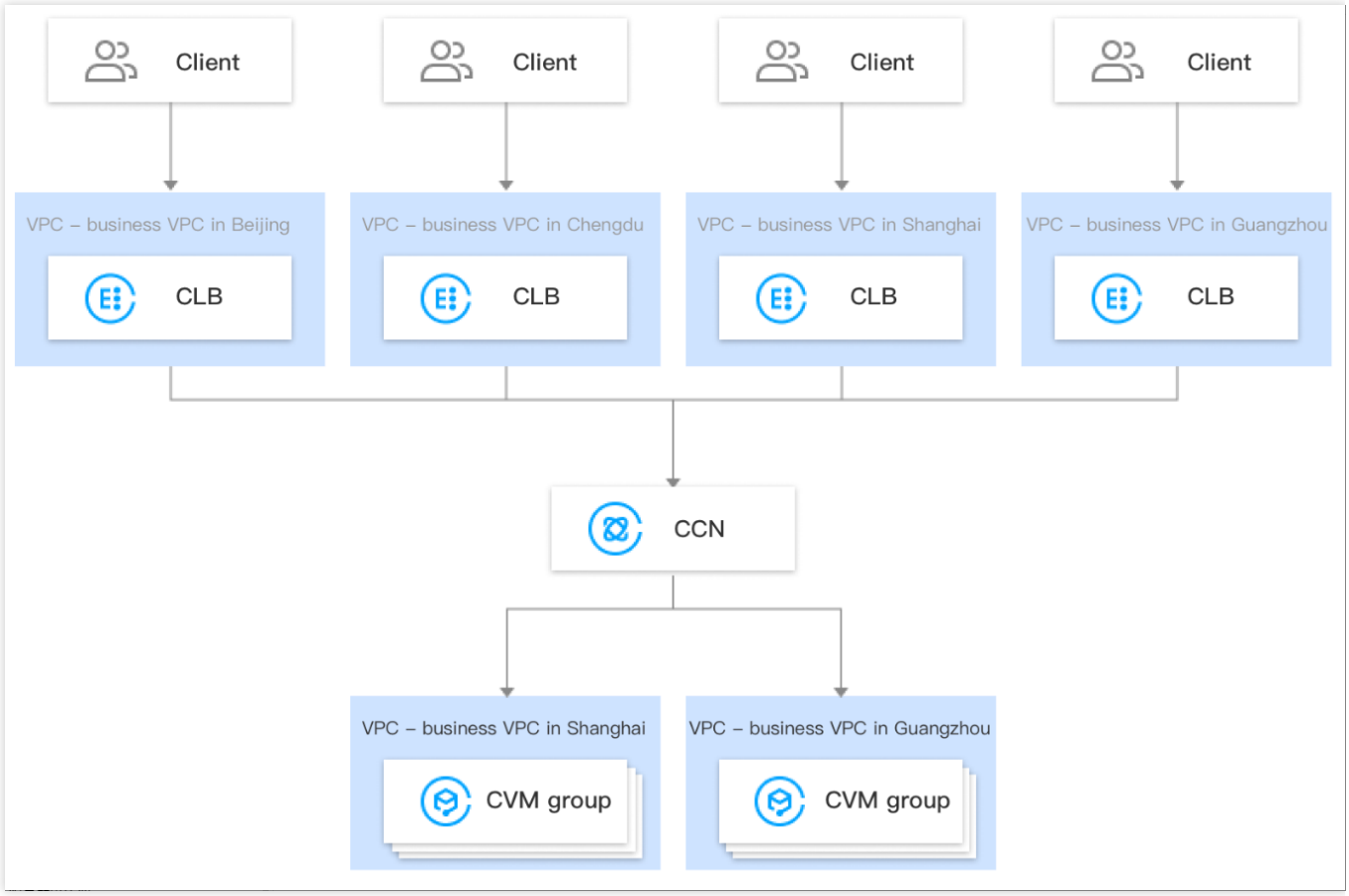
クロスリージョンバインディング2.0（新バージョン）

最終更新日：：2024-01-04 18:36:26

Cloud Load Balancer(CLB)は、CCN、クロスリージョンバインディングバックエンドサーバーをサポートしており、お客様が複数のバックエンドサーバーのリージョン、クロスVPC、クロスリージョンバインディングのバックエンドサーバーを選択できるようになっています。</td>
この機能は現在、ベータ版テスト段階です。この機能の体験を希望される場合は、ベータ版テスト申請を行ってください。
説明
クロスリージョンバインディング2.0は、現時点では従来型のCLBをサポートしていません。
この機能は、標準的なアカウントタイプでのみサポートされています。アカウントタイプが確定できない場合は、アカウントタイプの判断をご参照ください。
クロスリージョンバインディング2.0とハイブリッドクラウドデプロイは、セキュリティグループのデフォルト許可をサポートしていません。バックエンドサーバーでClient IPとサービスポートを許可してください。
クロスリージョンバインディング2.0とハイブリッドクラウドデプロイは、他のCLBインスタンスのバインドはサポートしていません（すなわち、CLBとCLBとのバインドはサポートしていません）。

ユースケース

1. P2Pなどのゲーム事業において、マルチサイト同一サーバーのシーンに対応します。お客様のバックエンドサービスクラスターが広州にあり、上海や北京など複数のリージョンでCLBを作成し、同じ広州のバックエンドサービスクラスターをバインドしたいと希望される場合、ゲームのアクセラレーションとトラフィック収束の役割を果たし、データ伝送品質を効果的に確保し、遅延を低減させます。
2. 金融ビジネスでの決済や注文・支払いといったシーンに対応し、主要活動でのデータ伝送品質とデータの整合性を効果的に保証します。



旧バージョンのクロスリージョンバインディングとの相違点

比較項目	クロスリージョンバインディング2.0（新バージョン）	クロスリージョンバインディング1.0（旧バージョン）
複数リージョンでの同時バインドサービスをサポートしていますか	サポートしています。 新バージョンのクロスリージョンバインディングCLBは、複数リージョンのCVMの同時バインドをサポートしています。 例えば、北京のCLBは、北京と上海のCVMを同時にバインドできます。	サポートしていません。 旧バージョンのクロスリージョンバインディングCLBは、1つのリージョンのCVMのバインドのみをサポートしています。 例えば、北京のCLBは上海のCVMをバインドできますが、北京と上海のCVMの同時バインドはできません。
クロスドメインから非クロスドメインへの変更をサポートしていますか	サポートしています。新バージョンのクロスリージョンバインディングは、もとの同一リージョンバインディングへの変更をサポートしています。	サポートしていません。旧バージョンのクロスリージョンバインディングでバックエンドインスタンスのリージョン属性を変更した後、このリージョンがCLBリージョンと異なる場合、

		もとの同一リージョンバインディングに変更できません。
CLBタイプをサポートしています	パブリックネットワークCLBとプライベートネットワークCLBをサポートしています。	パブリックネットワークCLBをサポートしています。
CVMリリース時にCLBを自動的にバインド解除しますか	同一リージョンでバインドする際の自動バインド解除： CLBが同一リージョンのCVMにバインドされている場合、このCVMがリリースされると、CLBは自動的にこのCVMのバインドを解除します。 クロスリージョンバインディングの際の自動バインド解除： CLBがクロスリージョンバインディングCVMの場合、このCVMがリリースされても、CLBは自動的にこのCVMとのバインドを解除することはなく、手動でバインドを解除する必要があります	同一リージョンでバインドする際の自動バインド解除： CLBが同一リージョンのCVMにバインドされている場合、このCVMがリリースされると、CLBは自動的にこのCVMのバインドを解除します。 クロスリージョンバインディングの際の自動バインド解除： CLBがクロスリージョンバインディングCVMの場合、このCVMがリリースされると、CLBは自動的にこのCVMとのバインドを解除します。
お得な価格ですか	クラウドネットワーク課金によって、きめ細かなコスト計算が可能になり、価格も下がります。	日95課金です。

制限条件

クロスネットワークバインディングのバックエンドサーバーは、現時点では従来型のCLBをサポートしていません。
この機能は、標準的なアカウントタイプでのみサポートされています。アカウントタイプが確定できない場合は、アカウントタイプの判断をご参照ください。
VPCのみサポートし、基幹ネットワークではサポートしていません。
この機能は、IPv4とIPv6のNAT64バージョンのCLBインスタンスの両方でサポートされています。IPv6バージョンのインスタンスは、デュアルスタックミックスバインド機能を有効にする必要があります。有効にすると、レイヤー7リスナーは、IPv4とIPv6のバックエンドサーバーを同時にバインドできるようになります。レイヤー7リスナーがIPv4 IPとミックスバインドする場合、クロスリージョンバインディング2.0とハイブリッドクラウドデプロイがサポートされます。IPv6バージョンのインスタンスがIPv6のバックエンドサーバーにバインドされる場合、クロスリージョンバインディング2.0とハイブリッドクラウドデプロイはサポートされません。
クロスリージョンバインディング2.0とハイブリッドクラウドデプロイは、セキュリティグループのデフォルト許可をサポートしていません。バックエンドサーバーでClient IPとサービスポートを許可してください。

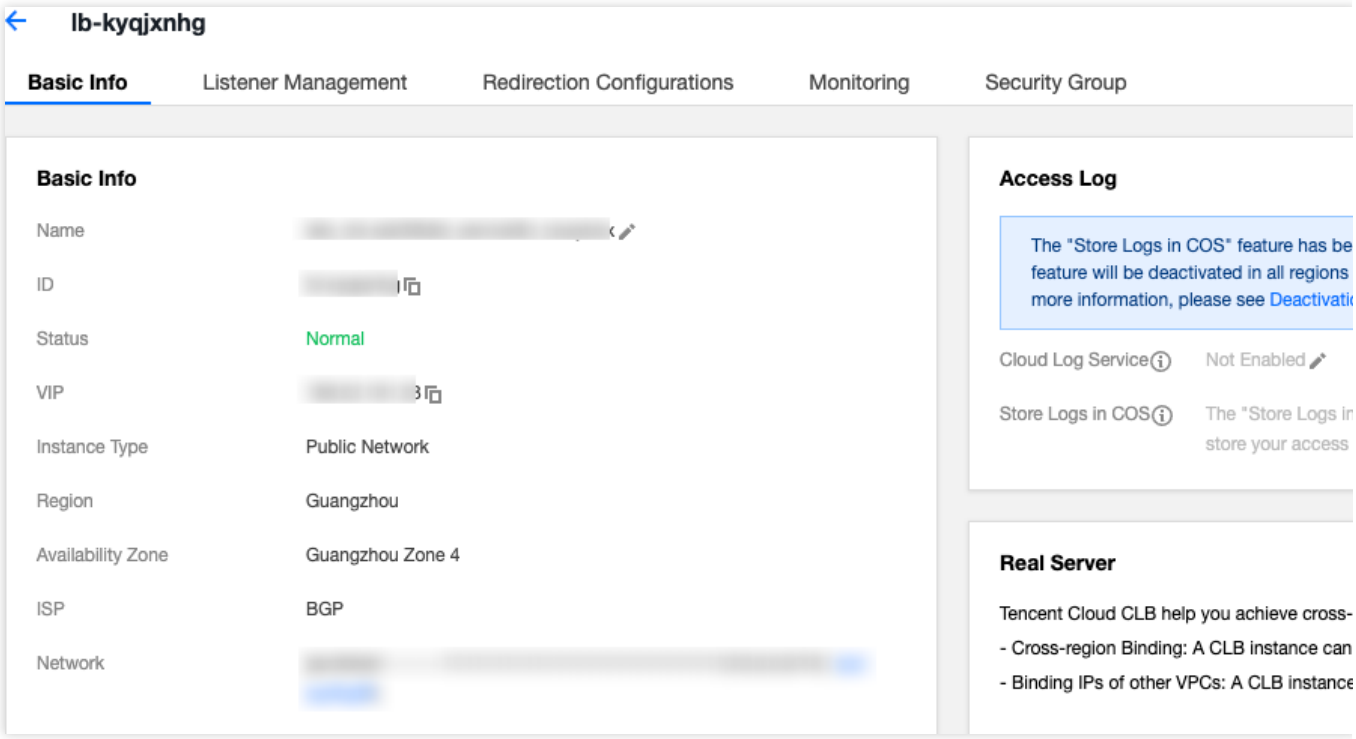
クロスリージョンバインディング2.0とハイブリッドクラウドデプロイは、他のCLBインスタンスのバインドはサポートしていません（すなわち、CLBとCLBとのバインドはサポートしていません）。
レイヤー4/7リスナーは、どちらもクライアントIPの取得をサポートしています。レイヤー4CLBがバックエンドCVMで取得したソースIPは、クライアントIPとなります。レイヤー7CLBは、X-Forwarded-Forまたはremote_addrフィールドからクライアントIPを取得する必要があります。詳細については、クラウド上のIPバインドシーンでのクライアントリアルIPの取得をご参照ください。

前提条件

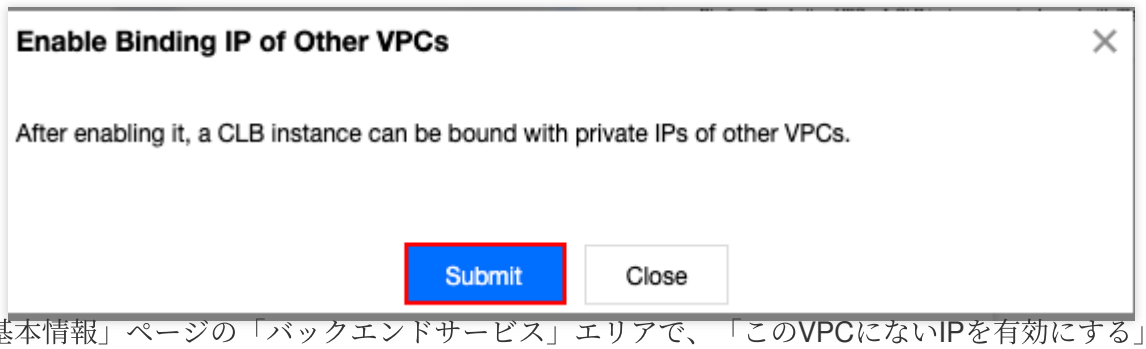
1. ベータ版テストを申請済みであること。中国本土のクロスリージョンバインディングについては、ベータ版テスト申請、中国本土以外のクロスリージョンバインディングについては、ビジネス申請から申請してください。
2. CLBインスタンスを作成済みであること。詳細については、CLBインスタンスの作成をご参照ください。
3. CCNインスタンスを作成済みであること。詳細については、CCNインスタンスの新規作成をご参照ください。
4. バインドしたいターゲットVPCを作成済みのCCNインスタンスにバインドします。詳細については、ネットワークインスタンスのバインドをご参照ください。

操作手順

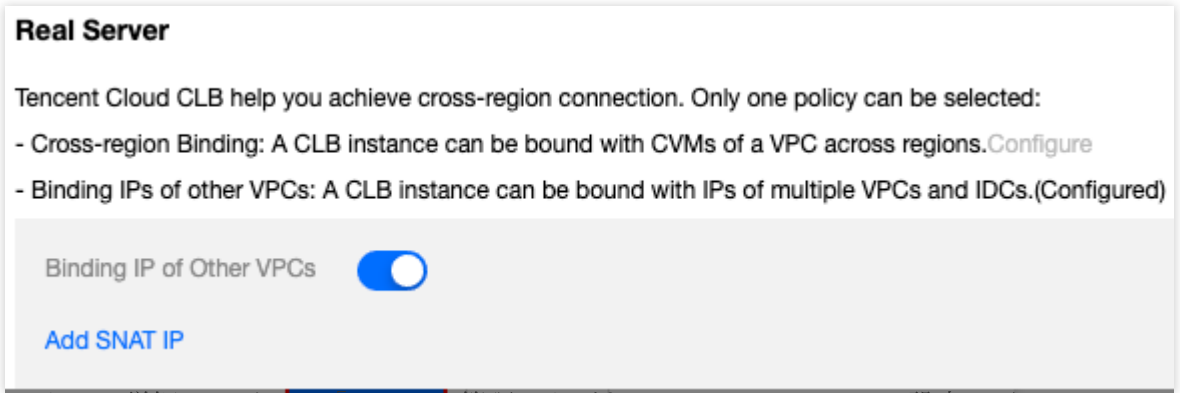
1. CLBコンソールにログインします。
2. インスタンス詳細ページでターゲットCLBインスタンスを見つけ、インスタンスIDをクリックします。
3. 「基本情報」ページの「バックエンドサービス」エリアで、**設定をクリック**をクリックして、このVPCにないプライベートIPをバインドします。



4. ポップアップした「このVPCにないIPを有効にする」ダイアログボックスで、**送信**をクリックします。

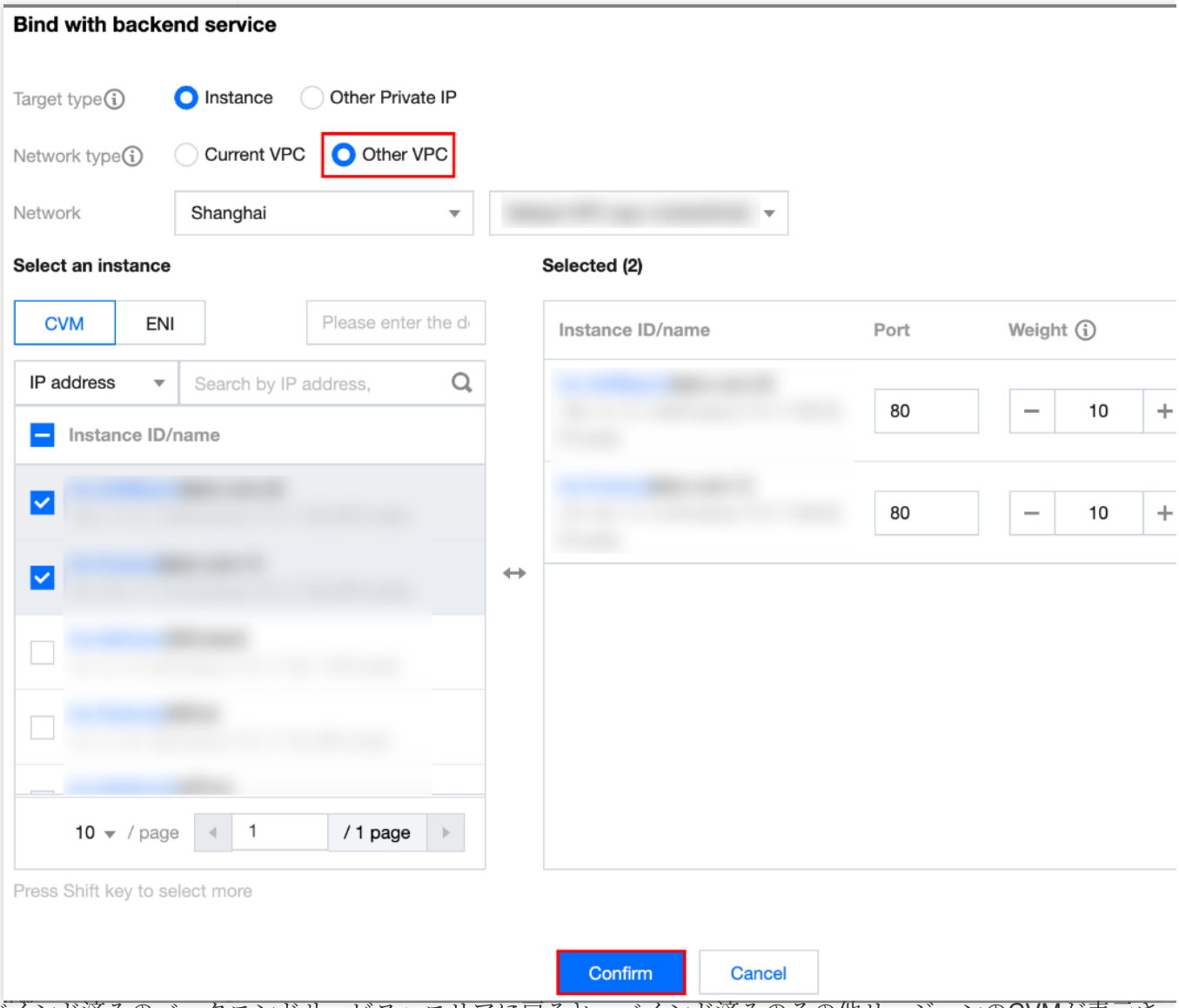


5. 「基本情報」ページの「バックエンドサービス」エリアで、「このVPCにないIPを有効にする」スイッチがオンになっていることを確認します。オンになっていれば、クラウド上のIPをバインドできます。



6. インスタンス詳細ページで、「リスナー管理」タブをクリックし、リスナー設定モジュールで、CLBインスタンスにバックエンドサービスをバインドします。詳細については、CLBバックエンドCVMの追加をご参照ください。

7. ポップアップした「バックエンドサービスのバインド」ダイアログボックスで「その他のVPC」を選択し、**CVM**をクリックします。関連付けたいCVM（複数選択可）を選択し、関連のCVMの転送を希望するポートと重みを入力します。詳細については、サーバーの一般的なポートをご参照ください。その後**OK**をクリックします。



8. 「バインド済みのバックエンドサービス」エリアに戻ると、バインド済みのその他リージョンのCVMが表示されます。

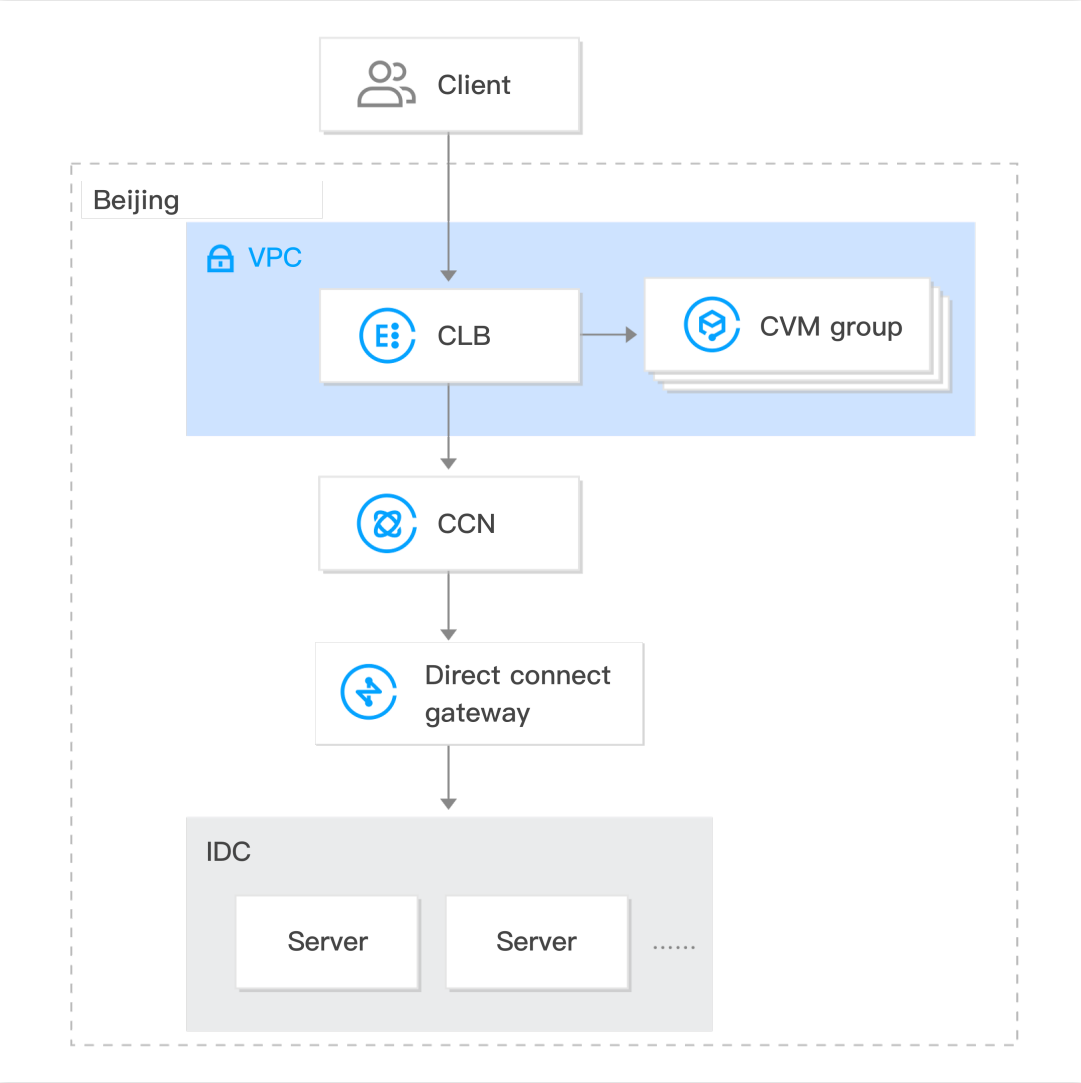
ハイブリッドクラウドのデプロイ

最終更新日：：2024-01-04 18:36:26

ハイブリッドクラウドのデプロイシーンでは、CLBを使用して、クラウドのローカルデータセンター(IDC)内のIPを直接バインドすると、VPCとIDCにまたがるバックエンドサーバーをバインドできます。
この機能は現在、ベータ版テスト段階です。この機能の体験を希望される場合、中国本土のクロスリージョンバインディングについては、ベータ版テスト申請、中国本土以外のクロスリージョンバインディングについては、ビジネス申請から申請してください。

ソリューションの優位性

ハイブリッドクラウドをすばやく構築し、クラウド内外をシームレスに接続します。CLBは、クラウド上のVPC内のサーバーとクラウド外のIDCルーム内のサーバーの両方に同時にリクエストを転送できます。
Tencent Cloudの高品質なパブリックネットワークアクセス機能を多重化します。
レイヤー4/7アクセス、ヘルスチェック、セッション維持など、Tencent Cloud CLBの豊富な機能特性を多重化します。
プライベートネットワークはCCNで相互接続されており、品質確保のためのきめ細かいルート選択や、コスト削減のための多様な階層型課金をサポートしています。



制限条件

クロスリージョンバインディング2.0は、現時点では従来型のCLBをサポートしていません。
この機能は、標準的なアカウントタイプでのみサポートされています。アカウントタイプが確定できない場合は、アカウントタイプの判断をご参照ください。
VPCのみサポートし、基幹ネットワークではサポートしていません。
この機能は、IPv4とIPv6のNAT64バージョンのCLBインスタンスの両方でサポートされています。IPv6バージョンのインスタンスは、デュアルスタックミックスバインド機能を有効にする必要があります。有効にすると、レイヤー7リスナーは、IPv4とIPv6のバックエンドサーバーを同時にバインドできるようになります。レイヤー7リスナーがIPv4 IPとミックスバインドする場合、クロスリージョンバインディング2.0とハイブリッドクラウドデプロイがサポートされます。IPv6バージョンのインスタンスがIPv6のバックエンドサーバーにバインドされる場合、クロスリージョンバインディング2.0とハイブリッドクラウドデプロイはサポートされません。

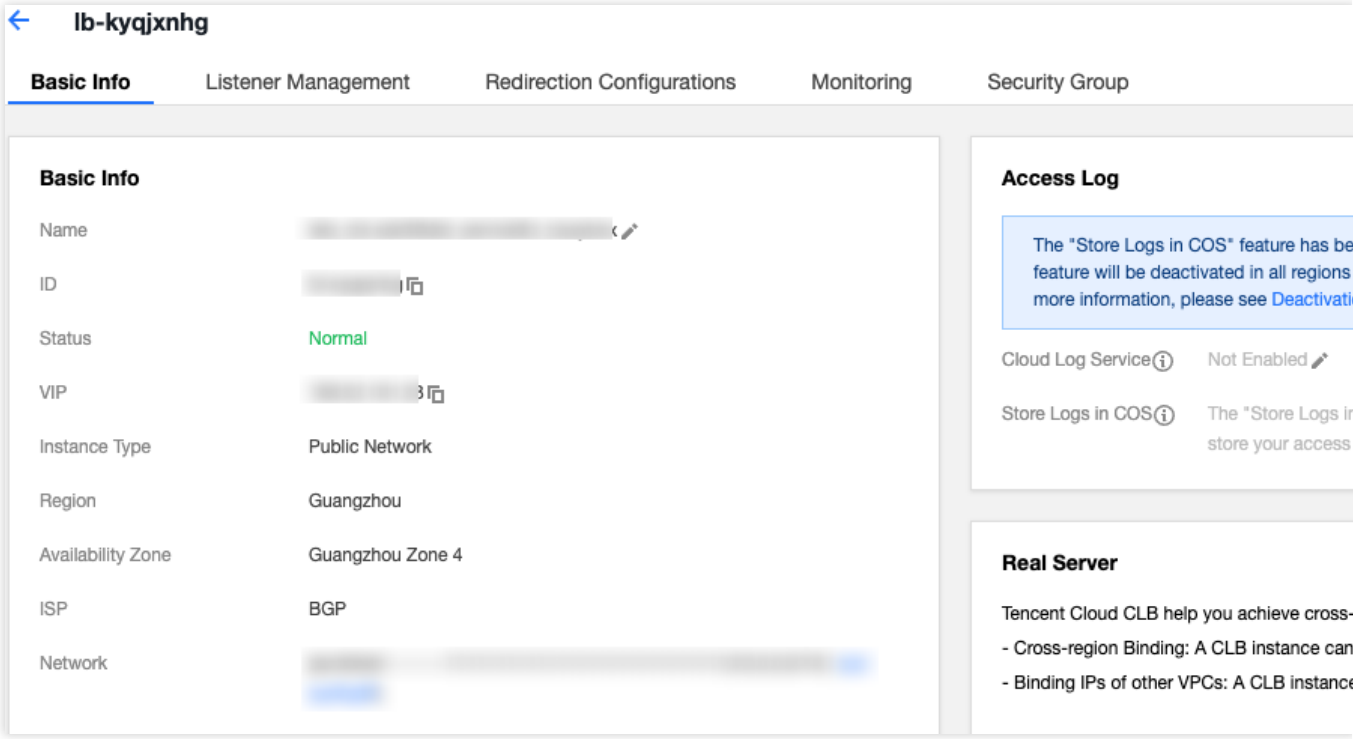
クロスリージョンバインディング2.0とハイブリッドクラウドデプロイは、セキュリティグループのデフォルト許可をサポートしていません。バックエンドサーバーでClient IPとサービスポートを許可してください。
クロスリージョンバインディング2.0とハイブリッドクラウドデプロイは、他のCLBインスタンスのバインドはサポートしていません（すなわち、CLBとCLBとのバインドはサポートしていません）。
この機能は現在、広州、上海、済南、杭州、合肥、北京、天津、成都、重慶、中国香港、シンガポール、シリコンバレーのみでサポートされています。
TCPとTCP SSLリスナーは、RSで汎用TOA 経由でソースIPを取得する必要があります。詳細については、ハイブリッドクラウドのデプロイシーンでのTOAによるクライアントリアルIPの取得をご参照ください。
HTTとHTTPSのリスナーは、X-Forwarded-For(XFF)経由でソースIPを取得する必要があります。
UDPリスナーでは、ソースIPの取得はサポートされていません。

前提条件

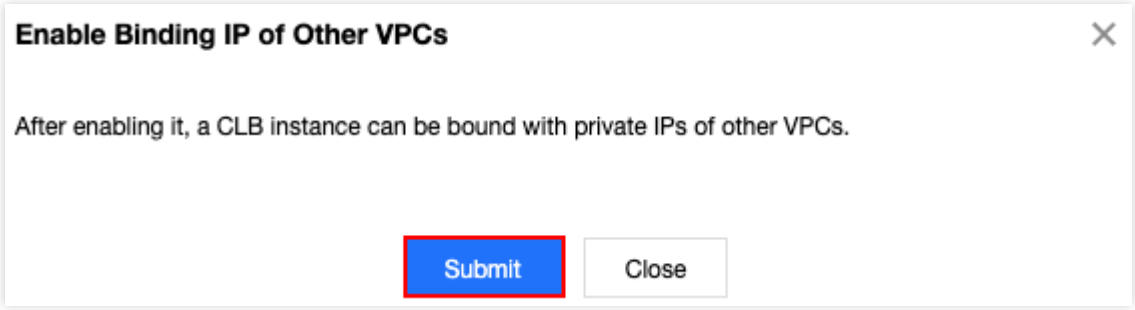
1. ベータ版テストを申請済みであること。中国本土のクロスリージョンバインディングについては、ベータ版テスト申請、中国本土以外のクロスリージョンバインディングについては、ビジネス申請から申請してください。
2. CLBインスタンスを作成済みであること。詳細については、CLBインスタンスの作成をご参照ください。
3. Cloud Connect Network(CCN)インスタンスを作成済みであること。詳細については、CCNインスタンスの新規作成をご参照ください。
4. IDCにバインドされた専用ゲートウェイと、バインドしたいターゲットVPCを作成済みのCCNインスタンスにバインドします。詳細については、ネットワークインスタンスのバインドをご参照ください。

操作手順

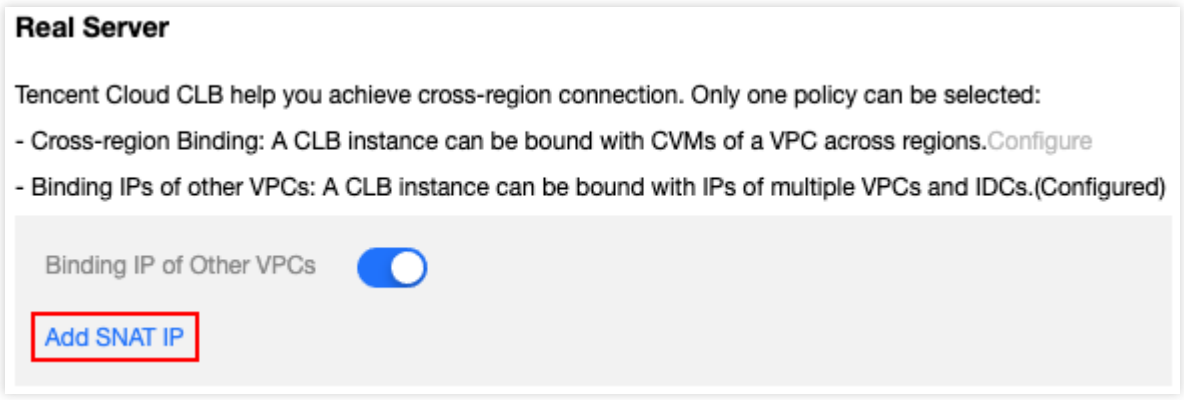
1. CLBコンソールにログインします。
2. CLB「インスタンス管理」ページでターゲットCLBインスタンスを見つけ、インスタンスIDをクリックします。
3. 「基本情報」ページの「バックエンドサービス」エリアで、設定をクリックをクリックして、このVPCにないプライベートIPをバインドします。



4. ポップアップした「このVPCにないIPを有効にする」ダイアログボックスで、送信をクリックします。

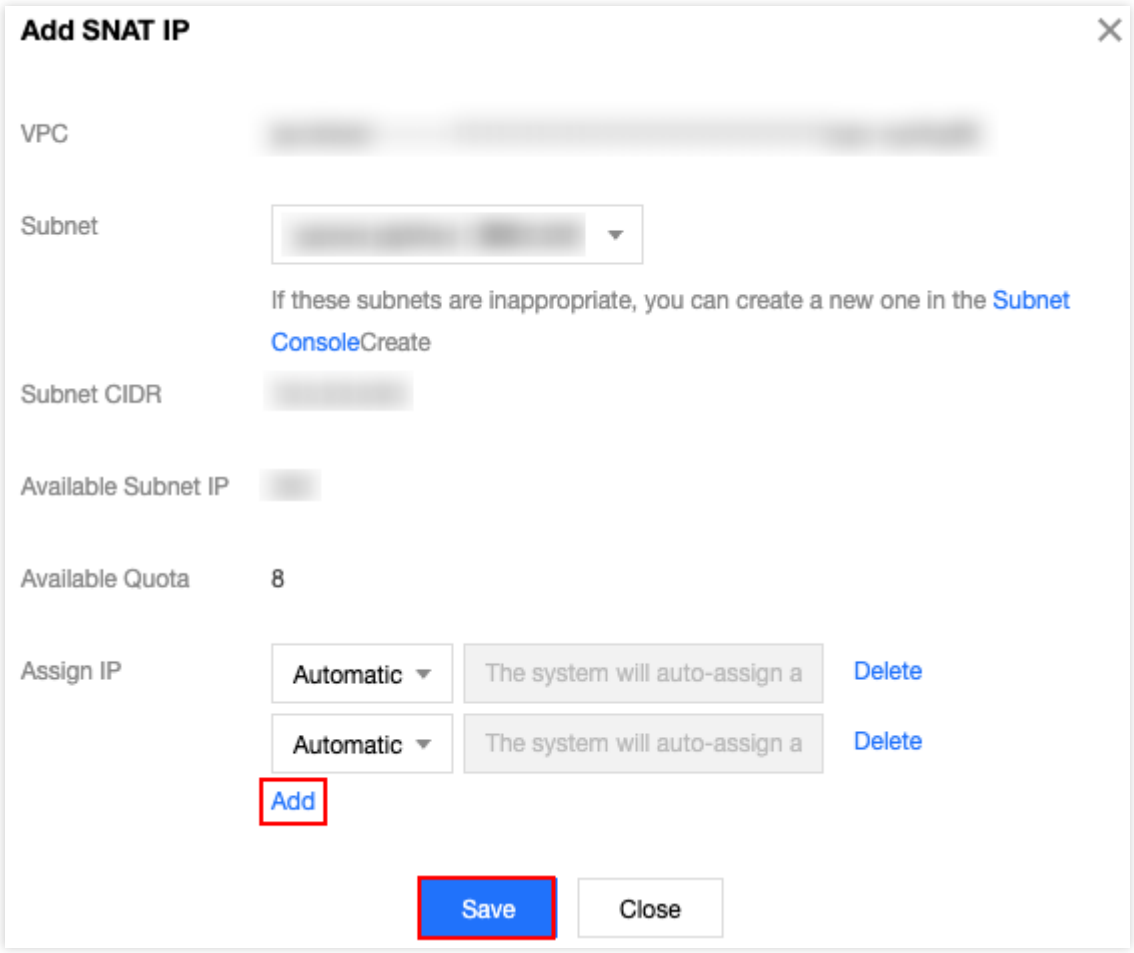


5. 「基本情報」ページの「バックエンドサービス」エリアで、SNAT IPの追加をクリックします。



6. ポップアップした「SNAT IPの追加」ダイアログボックスで、「サブネット」を選択し、追加をクリックしてIPを割り当て、最後に保存をクリックします。

説明：
SNAT IPは、主にハイブリッドクラウドデプロイにおいてIDC内のサーバーにリクエストを転送するシーンで使用されます。CLBを使用してCCNに接続されたIDC内のIPをバインドする場合、SNAT IPを割り当てる必要があります。SNAT IPは、お客様のVPCのプライベートIPです。
1つのCLBインスタンスに対して、最大10個のSNAT IPを設定できます。
1つのCLBインスタンスの1つのルールが1つのSNAT IPを設定し、1つのバックエンドサービスをバインドした場合の最大接続数は55,000になります。SNAT IPやバックエンドサービスを追加すると、接続数はそれに比例して増加します。例えば、1つのCLBインスタンスが2つのSNAT IPを設定すると、バックエンドは10ポートをバインドします。この場合、このCLBインスタンス数は2x10x5.5万=110万個となります。接続数に応じて、SNAT IPの割り当て数を評価することができます。
SNAT IPを削除すると、そのSNAT IPの接続がすべて切断されますので、慎重に操作してください。



7. インスタンス詳細ページで、「リスナー管理」タブをクリックし、リスナー設定モジュールで、CLBインスタンスにバックエンドサービスをバインドします。詳細については、CLBバックエンドCVMの追加をご参照ください。
8. ポップアップした「バックエンドサービスのバインド」ダイアログボックスで、「その他のプライベートIPの追加」を選択し、プライベートIPの追加をクリックしてバインドしたIDCプライベートIPアドレスを入力し、ポー

トと重みを入力します。詳細については、サーバーの一般的なポートをクリックし、最後に確認をクリックします。
9. 「バインド済みのバックエンドサービス」エリアに戻ると、バインド済みのIDCのプライベートIPが表示されます。

関連ドキュメント

クロスリージョンバインディング2.0（新バージョン）

バックエンドCVMのセキュリティグループ設定

最終更新日：：2024-01-04 18:36:26

CVMセキュリティグループの概要

CLBのバックエンドCVMインスタンスはセキュリティグループによってアクセスを制御し、ファイアウォールの役割をさせることができます。
1つまたは複数のセキュリティグループをバックエンドCVMと関連付け、さらに各セキュリティグループに1つまたは複数のルールを追加することで、さまざまなサーバーのトラフィックアクセス権限を制御することができます。セキュリティグループのルールはいつでも変更でき、新ルールはそのセキュリティグループに関連付けられたすべてのインスタンスに自動的に適用されます。その他の情報に関しては、セキュリティグループ操作ガイドをご参照ください。VPC 環境では、ネットワークACLを使用してアクセス制御を行うこともできます。

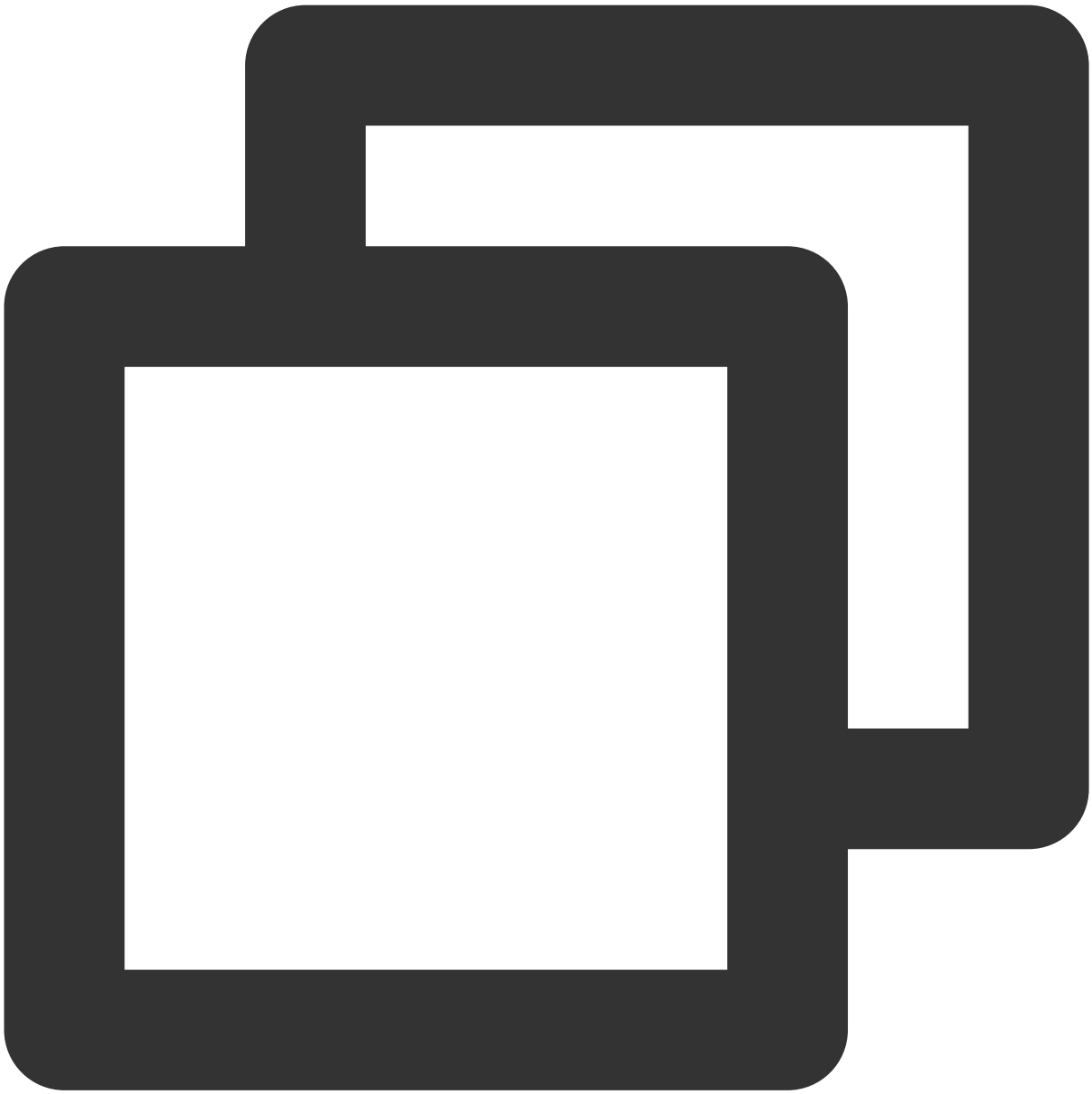
CVMセキュリティグループ設定の説明

CVMセキュリティグループでは、Client IPおよびサービスポートを開放する必要があります。
CLBを使用して業務トラフィックをCVMに転送する場合、ヘルスチェック機能を保障するため、CVMセキュリティグループに次の設定を行う必要があります。
1. パブリックネットワークCLB：バックエンドCVMのセキュリティグループでCLBのVIPを開放する必要がある場合、CLBはVIPを使用してバックエンドCVMのヘルスステータスをチェックします。
2. プライベートネットワークCLB：
プライベートネットワークCLB（旧「アプリケーション型プライベートネットワークCLB」）については、CLBがVPCネットワークにある場合、 バックエンドCVMのセキュリティグループ上でCLBのVIP（ヘルスチェック用）を開放する必要があります。CLBが基幹ネットワークにある場合は、バックエンドCVMのセキュリティグループ上で設定を行う必要はなく、ヘルスチェックIPがデフォルトで開放されています。

CVMセキュリティグループ設定の例

次の例は、CLBからCVMにアクセスする場合の、CVMセキュリティグループの設定例です。CLB上でもセキュリティグループを設定する場合は、CLBセキュリティグループの設定
をご参照の上、CLB上のセキュリティグループルールを設定してください。
ユースケース1：
パブリックネットワークCLBで、リスナーをTCP:80リスナーに、バックエンドサービスポートを8080にそれぞれ

設定し、Client IP（ClientA IPおよびClientB IP）にのみCLBへのアクセスを許可したい場合、バックエンドサーバーセキュリティグループのインバウンドルールの設定は次のようになります。

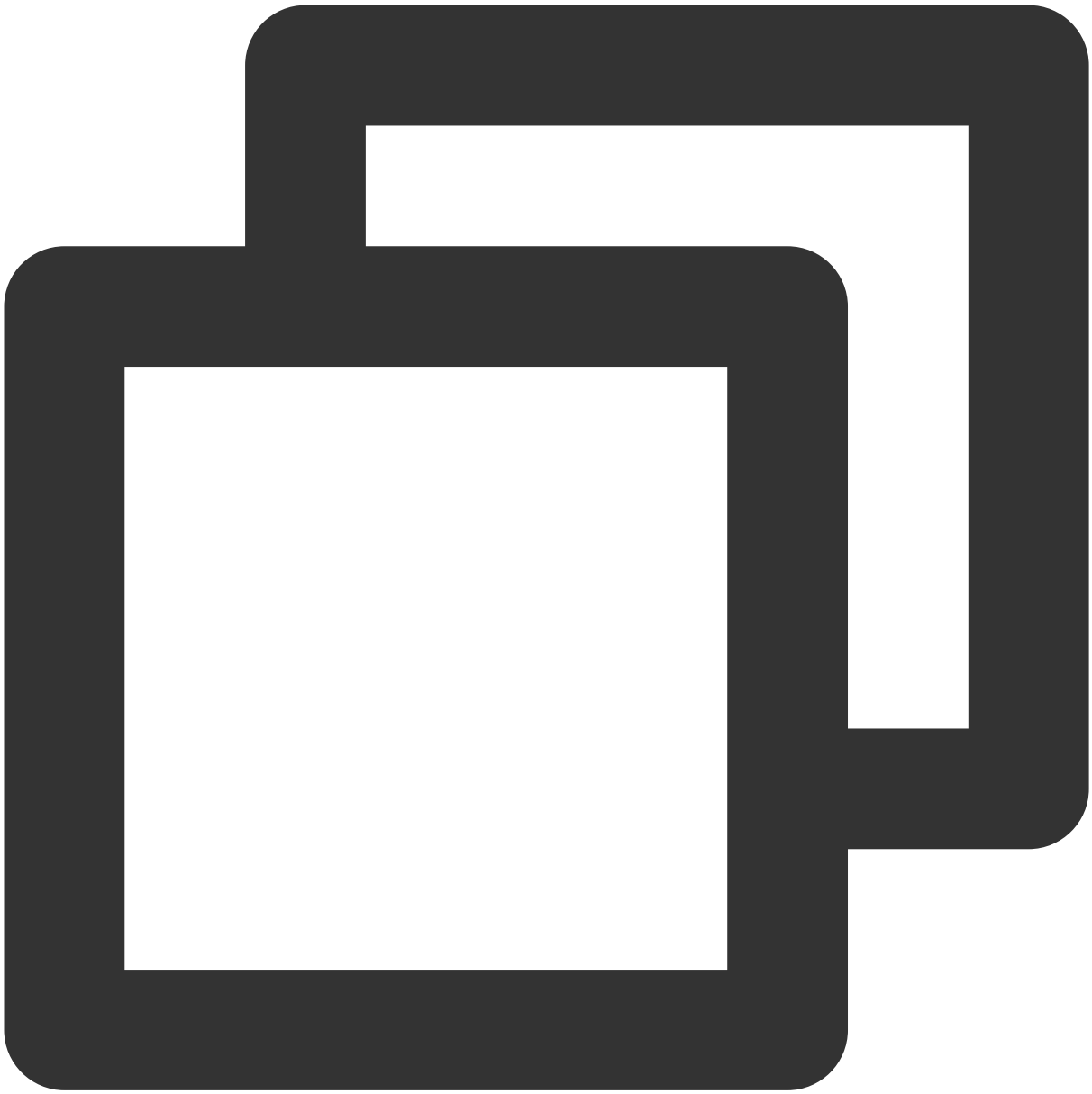


```
ClientA IP + 8080 allow
ClientB IP + 8080 allow
CLB VIP    + 8080 allow
0.0.0.0/0  + 8080 drop
```

ユースケース2：

パブリックネットワークCLBで、リスナーをHTTP:80リスナーに、バックエンドサービスポートを8080にそれぞ

れ設定し、すべてのClient IPに正常なアクセスを開放したい場合、バックエンドサーバーセキュリティグループのインバウンドルールの設定は次のようになります。



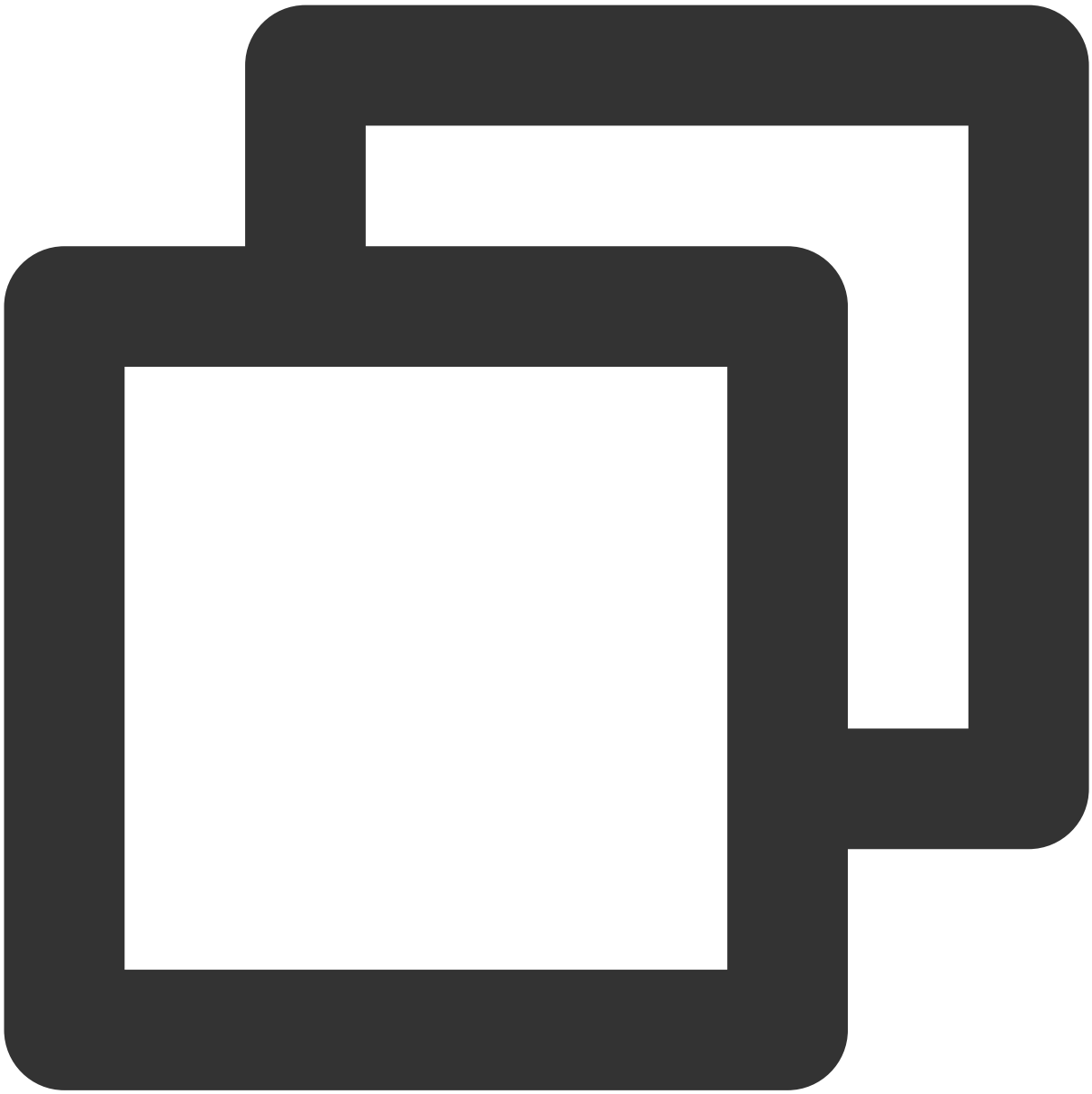
```
0.0.0.0/0 + 8080 allow
```

ユースケース3：

プライベートネットワークCLB（旧「アプリケーション型プライベートネットワークCLB」）で、ネットワークタイプがVPCネットワークの場合に、CVMのセキュリティグループ上でCLBのVIPを開放してヘルスチェックを行う必要があるとします。このCLBにTCP:80リスナーを設定し、バックエンドサービスポートを8080とし、Client IP（ClientA IPおよびClientB IP）にのみCLBのVIPへのアクセスを許可し、なおかつClient IPがそのCLBにバインドさ

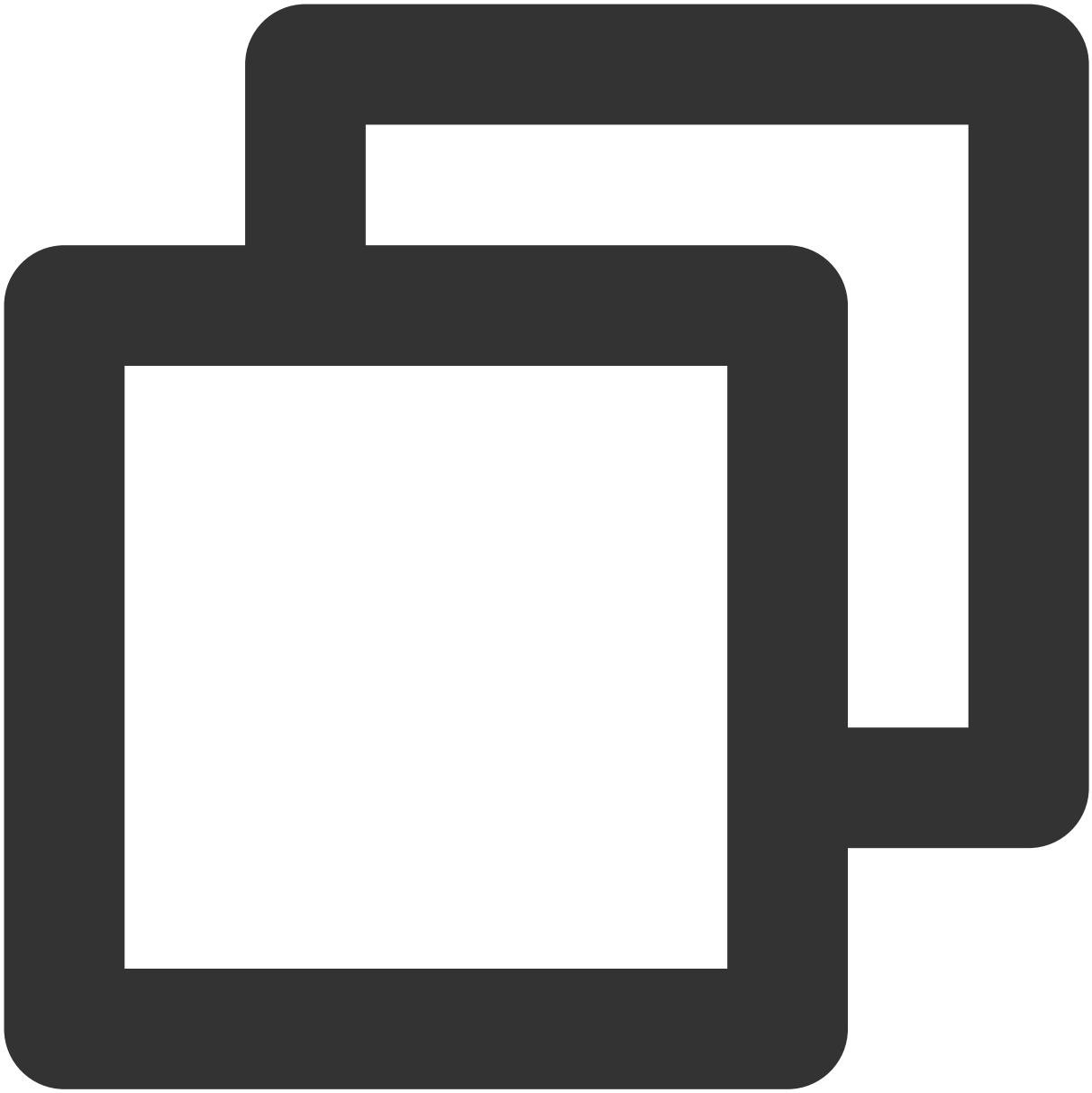
れたバックエンドホストにのみアクセスできるよう制限したい場合です。

a. バックエンドサーバーセキュリティグループのインバウンドルールの設定は次のようになります。



```
ClientA IP + 8080 allow
ClientB IP + 8080 allow
CLB VIP    + 8080 allow
0.0.0.0/0  + 8080 drop
```

b. Clientとして用いるサーバーのセキュリティグループのアウトバウンドルールの設定は次のようになります。



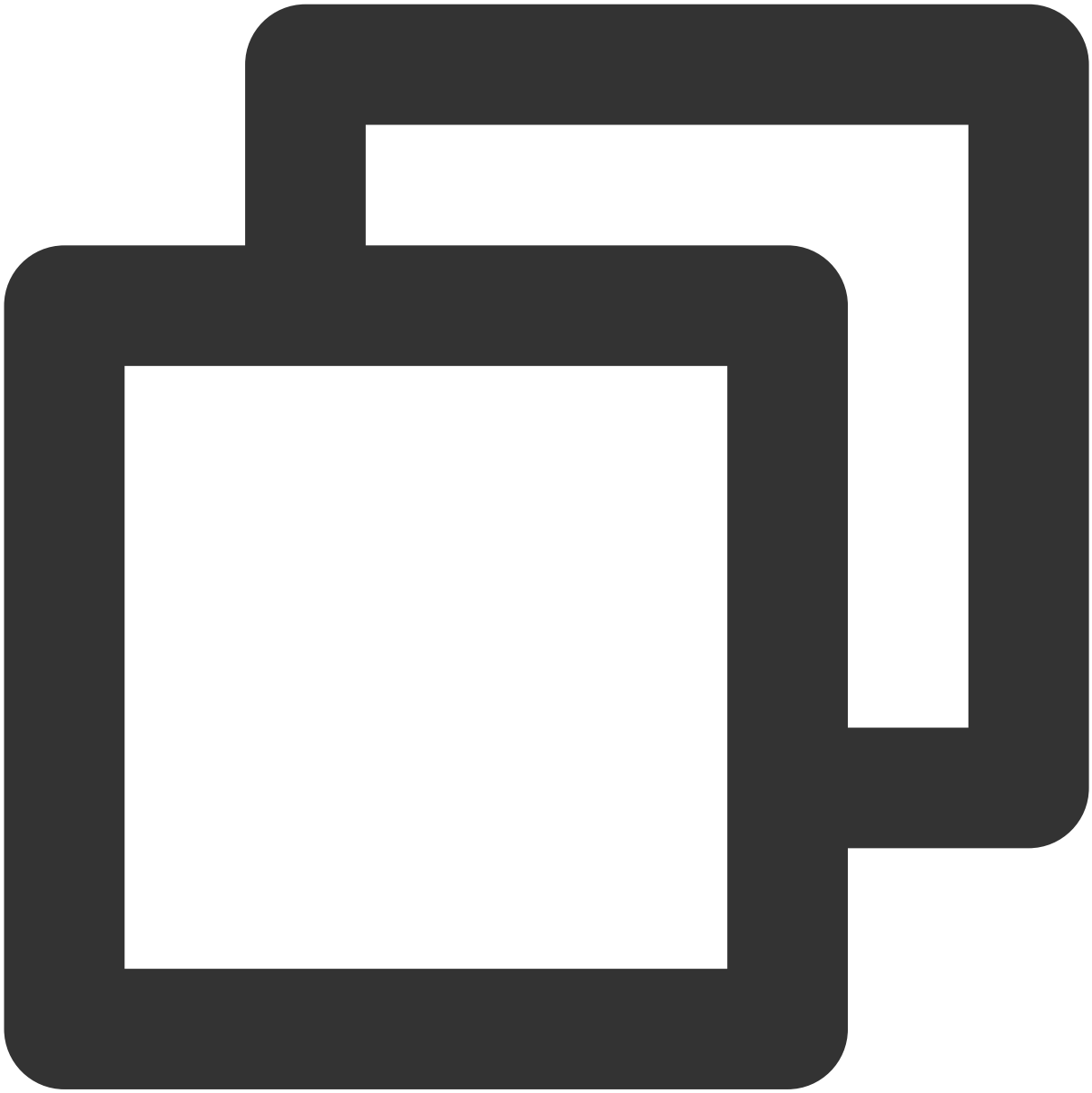
```
CLB VIP     + 8080 allow
0.0.0.0/0   + 8080 drop
```

ユースケース4：ブラックリスト

ユーザーがいくつかのClient IPをブラックリストに設定し、そのアクセスを拒否したい場合は、クラウドサービスに関連付けたセキュリティグループによって実現することができます。セキュリティグループのルールは次の手順に従って設定する必要があります。

アクセスを拒否したいClient IP + ポートをセキュリティグループに追加し、ポリシー欄でこのIPからのアクセス拒否を選択します。

設定完了後、セキュリティグループルールを1つ追加し、このポートを全IPからのアクセスにデフォルトで開放します。
設定完了後、セキュリティグループルールは次のようになります。



```
clientA IP + port drop
clientB IP + port drop
0.0.0.0/0  + port accept
```

注意：
上記の設定手順には**順序要件**があります。順序を逆にすると、ブラックリストの設定が無効になります。

セキュリティグループはステートフルであるため、上記の設定はいずれも**インバウンドルール**の設定となり、アウトバウンドルールは特に設定する必要はありません。

CVMセキュリティグループの操作ガイド

コンソールを使用してバックエンドサーバーセキュリティグループを管理する

1. CLBコンソールにログインし、該当するCLBインスタンスIDをクリックしてCLB詳細ページに進みます。
2. CLBにバインドしたCVMのページで、該当するバックエンドサーバーIDをクリックしてCVM詳細ページに進みます。
3. **セキュリティグループ**のオプションタブをクリックすると、セキュリティグループのバインド/バインド解除を行うことができます。

Tencent Cloud APIを使用してバックエンドサーバーセキュリティグループを管理する

セキュリティグループバインドインターフェースおよびセキュリティグループバインド解除インターフェースをご参照ください。

ヘルスチェック
ヘルスチェックの概要

最終更新日：：2024-01-04 18:36:26

CLBはヘルスチェックによってバックエンドサービスの可用性を判断し、バックエンドサービスの異常によるフロントエンドへの影響を回避することで、業務全体の可用性を向上させます。
ヘルスチェックを有効化すると、バックエンドサーバーの重み付けの大小にかかわらず（重みが0の場合も含めて）、CLBインスタンスは常にヘルスチェックを実行します。ヘルスチェックのステータスはインスタンスリストページの「ヘルスチェック」列か、リスナーにバインドしたバックエンドサービスの詳細ページで確認することができます。
バックエンドサーバーインスタンスが異常と判定された場合、CLBインスタンスは新しいリクエストを異常なバックエンドサーバーには転送せず、他の正常なバックエンドサーバーに自動的に転送します。
異常なインスタンスが正常に復旧すると、CLBはそのインスタンスをCLBサービスに復帰させ、リクエストの転送を再開します。
ヘルスチェックの結果、バックエンドサービスのすべてに異常が検出された場合、リクエストはすべてのバックエンドサーバーに転送されます。
ヘルスチェックを無効化すると、CLBはすべてのバックエンドサーバー（異常なバックエンドサーバーも含めて）にトラフィックを転送します。このため、ヘルスチェックを有効化し、CLBが異常なバックエンドサーバーを自動的にチェックして削除できるようにしておくことを強く推奨します。
デフォルトのパッシブヘルスチェックは、レイヤー4のTCP SSLリスナーやレイヤー7のHTTP/HTTPSリスナーには、パッシブヘルスチェック機能がデフォルトで設定されています（デフォルトで有効であり、無効化はサポートされません）。CLBは、トラフィックをバックエンドサービスに転送すると同時にバックエンドサービスのヘルスステータスを記録します。転送に失敗した場合、他のバックエンドサービスへの転送を再試行すると同時に、このバックエンドサービスの失敗回数を累計1回とします。失敗回数の累計が3に達した場合、バックエンドサービスは10秒間ブロックされます。ブロック時間が終了すると、トラフィック転送は再開され、バックエンドサービスのヘルスステータスが引き続き記録されます。

ヘルスチェックステータス

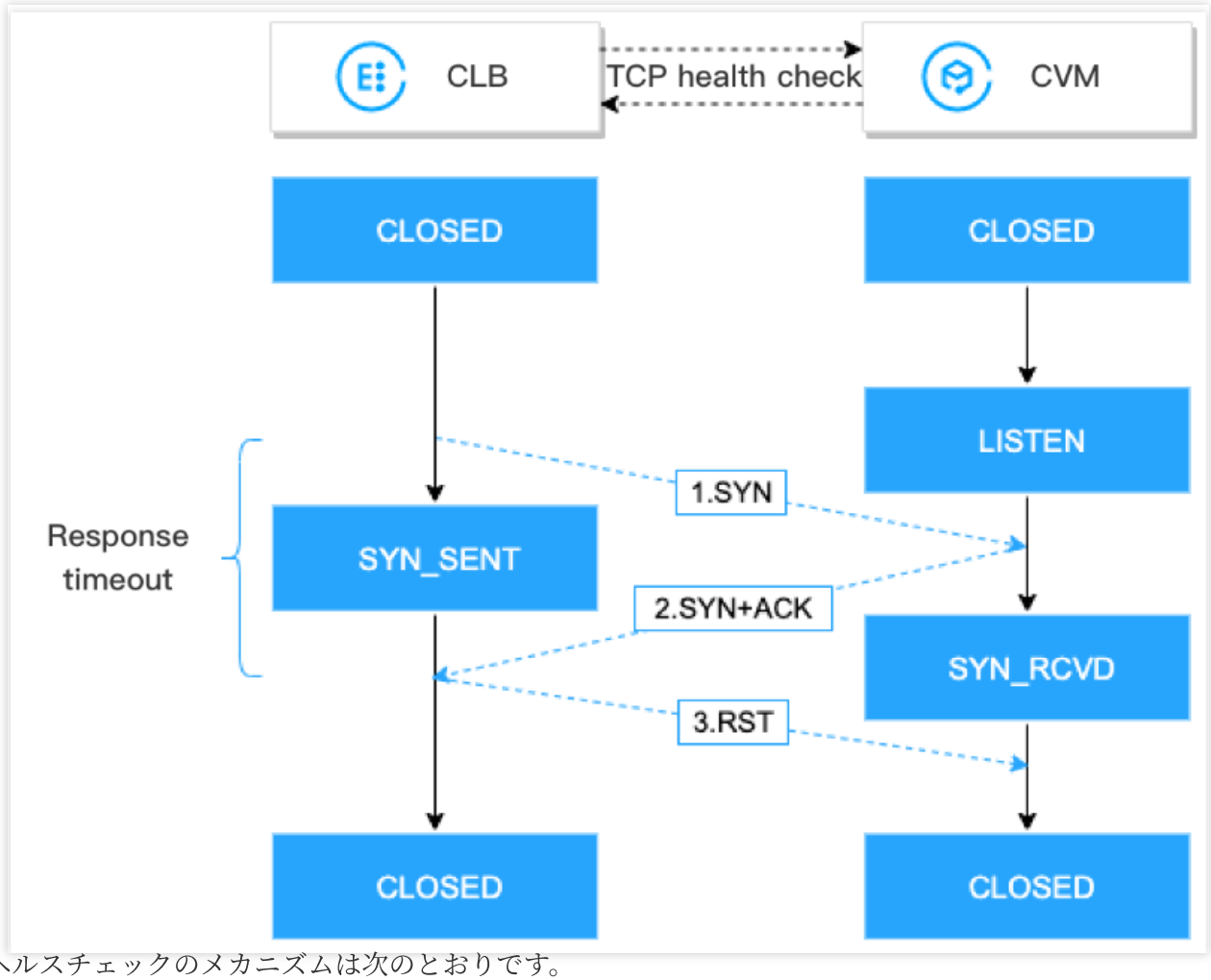
ヘルスチェックでの検出状況に基づく、バックエンドサーバーのヘルスチェックステータスは次のとおりです。

ステータス	説明	トラフィックを転送するかどうか
検出中	新たにバインドしたバックエンドサーバーの、チェック間隔×正常閾値の時間内におけるステータスです。	CLBは、「検出中」のバックエンドサービスにトラフィックを転送しません。

	例えば、チェック間隔が2秒、正常閾値が3回の場合、6秒間のステータスを表します。	
ヘルス	バックエンドサービスが正常な場合	CLBは、「正常」なバックエンドサービスにトラフィックを転送します。
異常	バックエンドサービスが異常な場合	CLBは、「異常」なバックエンドサービスにトラフィックを転送しません。 レイヤー4リスナーまたはレイヤー7URLルールでは、CLBがすべてのバックエンドサービスが異常であることを検出した場合、all-dead-all-aliveロジックがアクティブ化され、リクエストがすべてのバックエンドサービスに転送されます。
すでにオフです	ヘルスチェックをオフにする	CLBはバックエンドサービスにトラフィックを転送します。

TCPヘルスチェック

レイヤー4TCPリスナーについては、TCPヘルスチェックを設定できます。TCPヘルスチェックではSYNパケット、すなわちTCPの3ウェイハンドシェイクの開始によって、バックエンドサーバーのステータス情報を取得します。もしくは、カスタムプロトコルのリクエスト内容および返される結果によって、バックエンドサーバーのステータス情報を取得することもできます。



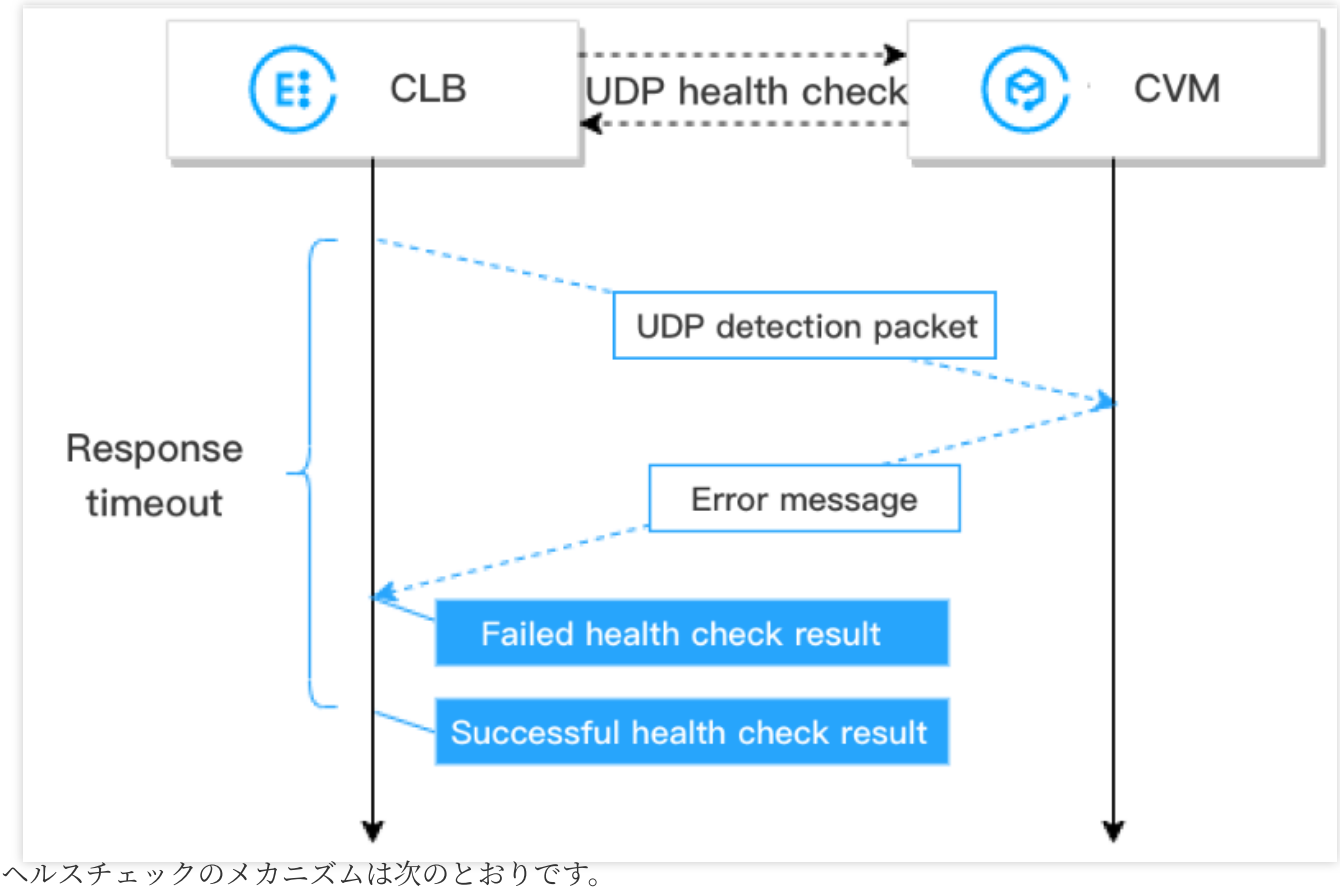
TCPヘルスチェックのメカニズムは次のとおりです。

1. CLBがバックエンドサーバー（プライベートIPアドレス+ヘルスチェックポート）にSYN接続リクエストメッセージを送信します。
2. バックエンドサーバーはSYNリクエストメッセージを受信後、対応するポートが正常なリスニング状態にある場合は、SYN+ACKレスポンスメッセージを返します。
3. レスポンスタイムアウト時間内にCLBがバックエンドサーバーから返されたSYN+ACKレスポンスメッセージを受信した場合、サービスは正常に実行されていることを表し、ヘルスチェックは成功と判定されます。CLBはバックエンドサーバーにRSTリセットメッセージを送信し、TCP接続を中断します。
4. レスポンスタイムアウト時間内にCLBがバックエンドサーバーから返されたSYN+ACKレスポンスメッセージを受信しなかった場合、サービスの実行が異常であることを表し、ヘルスチェックは失敗したと判定されます。CLBはバックエンドサーバーにRSTリセットメッセージを送信し、TCP接続を中断します。

UDPヘルスチェック

レイヤー4 UDPリスナーについては、UDPヘルスチェックを設定できます。UDPヘルスチェックでは、`Ping`コマンドおよびヘルスチェックポートへのUDPチェックメッセージの送信によってヘルスステータスを取得しま

す。もしくは、カスタムプロトコルのリクエスト内容および返される結果によってバックエンドサーバーのステータス情報を取得することもできます。



UDPヘルスチェックのメカニズムは次のとおりです。

1. CLBがバックエンドサーバーのプライベートIPアドレスに対し、`Ping` コマンドを送信します。
2. CLBがバックエンドサーバー（プライベートIPアドレス+ヘルスチェックポート）にUDPチェックメッセージを送信します。
3. `Ping` に成功し、なおかつレスポンスタイムアウト時間内に、バックエンドサーバーからエラーメッセージ `port XX unreachable` が返されなかった場合、サービスは正常であることを表し、ヘルスチェックは成功と判定されます。
4. `Ping` に失敗するか、またはレスポンスタイムアウト時間内に、システムがバックエンドサーバーから返されたエラーメッセージ `port XX unreachable` を受信した場合、サービスに異常があることを表し、ヘルスチェックは失敗と判定されます。

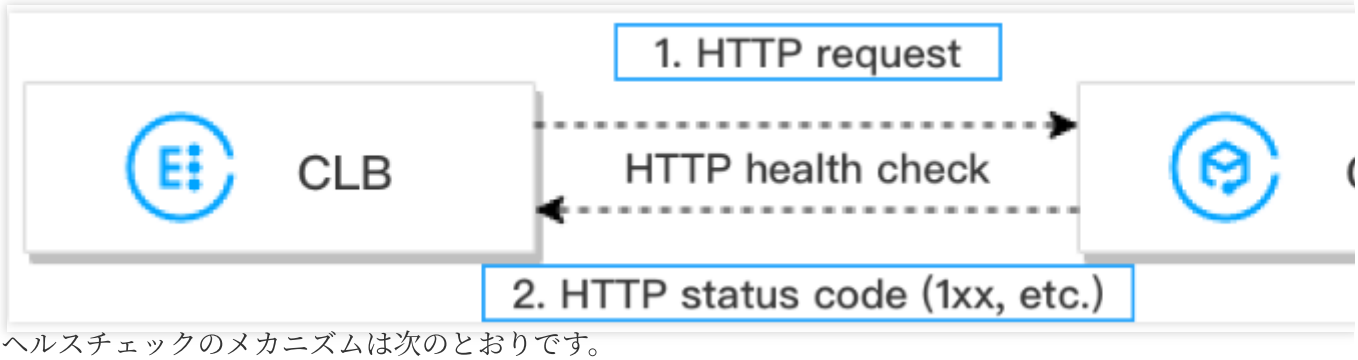
ご注意：

1. UDPヘルスチェックはICMPプロトコルに依存しているため、バックエンドサーバーはICMPパケット（`Ping` をサポート）、ICMPポート到達不能パケット（ポートチェックをサポート）を返せるよう許可する必要があります。
2. バックエンドサーバーがLinuxサーバーの場合、多数同時実行のシーンにおいて、LinuxはICMP攻撃からの保護メカニズムを備えるため、サーバーからのICMPパケット送信の速度を制限します。この場合、バックエンドサービスに異常が生じていても、CLBに `port XX unreachable` を返すことができないため、CLBはICMP応答を受

信していないことからヘルスチェックを成功と判定し、最終的にバックエンドサービスの真のステータスがヘルスチェックと一致しなくなります。
対処方法：UDPヘルスチェックの設定の際に、バックエンドサーバーに指定の文字列を送信するよう入力および出力をカスタム設定し、CLBが指定の応答を受信した場合のみヘルスチェック成功と判断するようにします。この方法はバックエンドサーバーに依存しますので、バックエンドサーバーはヘルスチェックの入力を処理し、指定の出力を返す必要があります。

HTTPヘルスチェック

レイヤー4 TCPリスナーおよびレイヤー7 HTTP/HTTPSリスナーについては、HTTPヘルスチェックを設定でき、HTTPリクエストを送信することでバックエンドサーバーのステータス情報を取得できます。



HTTPヘルスチェックのメカニズムは次のとおりです。
1. CLBがヘルスチェック設定に基づいて、バックエンドサーバー（プライベートIPアドレス+ヘルスチェックポート+チェックパス）にHTTPリクエストを送信します（チェックドメイン名を選択して設定できます）。
2. バックエンドサーバーはリクエストを受信後、該当するHTTPステータスコードを返します。
3. レスポンスタイムアウト時間内にCLBがバックエンドサーバーから返されたHTTPステータスコードを受信し、それが設定したHTTPステータスコードと一致した場合、ヘルスチェックは成功と判定され、そうでない場合は失敗と判定されます。
4. レスポンスタイムアウト時間内にCLBがバックエンドサーバーから返されたHTTPステータスコードを受信しなかった場合、ヘルスチェックは失敗と判定されます。
説明：
レイヤー7 HTTPSリスナーについては、HTTPSリスナーの転送ルールのバックエンドプロトコルにHTTPを選択した場合、ヘルスチェックにはHTTPヘルスチェックを使用します。HTTPSを選択した場合、ヘルスチェックにはHTTPSヘルスチェックを使用します。
HTTPSヘルスチェックは、基本的にHTTPヘルスチェックと類似しています。相違点は、HTTPSヘルスチェックはHTTPSリクエストを送信することによって、返されたHTTPSステータスコードに基づいてバックエンドサーバーのステータス情報を判断することです。

ヘルスチェックの時間範囲

CLBのヘルスチェックメカニズムは業務の可用性を効果的に向上させます。ヘルスチェックの頻繁な失敗に伴う切り替えがシステムの可用性に影響を与えることを防ぐため、ヘルスチェックはヘルスチェックの時間範囲内で複数回連続して成功または失敗した場合のみ、正常と異常のステータスを切り替えます。ヘルスチェックの時間範囲は次の要因によって決定されます。

ヘルスチェックの設定	説明	デフォルト値
レスポンスタイムアウト	ヘルスチェックのレスポンスの最大タイムアウト時間です。 バックエンドサーバーがタイムアウト時間内に正しくレスポンスしない場合は、ヘルスチェックに異常があると判断されます。 設定可能範囲は2～60秒です。	2秒
チェック間隔	CLBがヘルスチェックを行う時間の間隔です。 設定可能範囲は5～300秒です。	5秒
不健全なしきい値	n回（nには数値を入力）連続してヘルスチェック失敗の結果を受信した場合に、異常であると認識し、コンソールで失敗と表示します。 設定可能範囲は2～10回です。	3回
健全なしきい値	n回（nには数値を入力）連続してヘルスチェック成功の結果を受信した場合に、正常であると認識し、コンソールで成功と表示します。 設定可能範囲は2～10回です。	3回

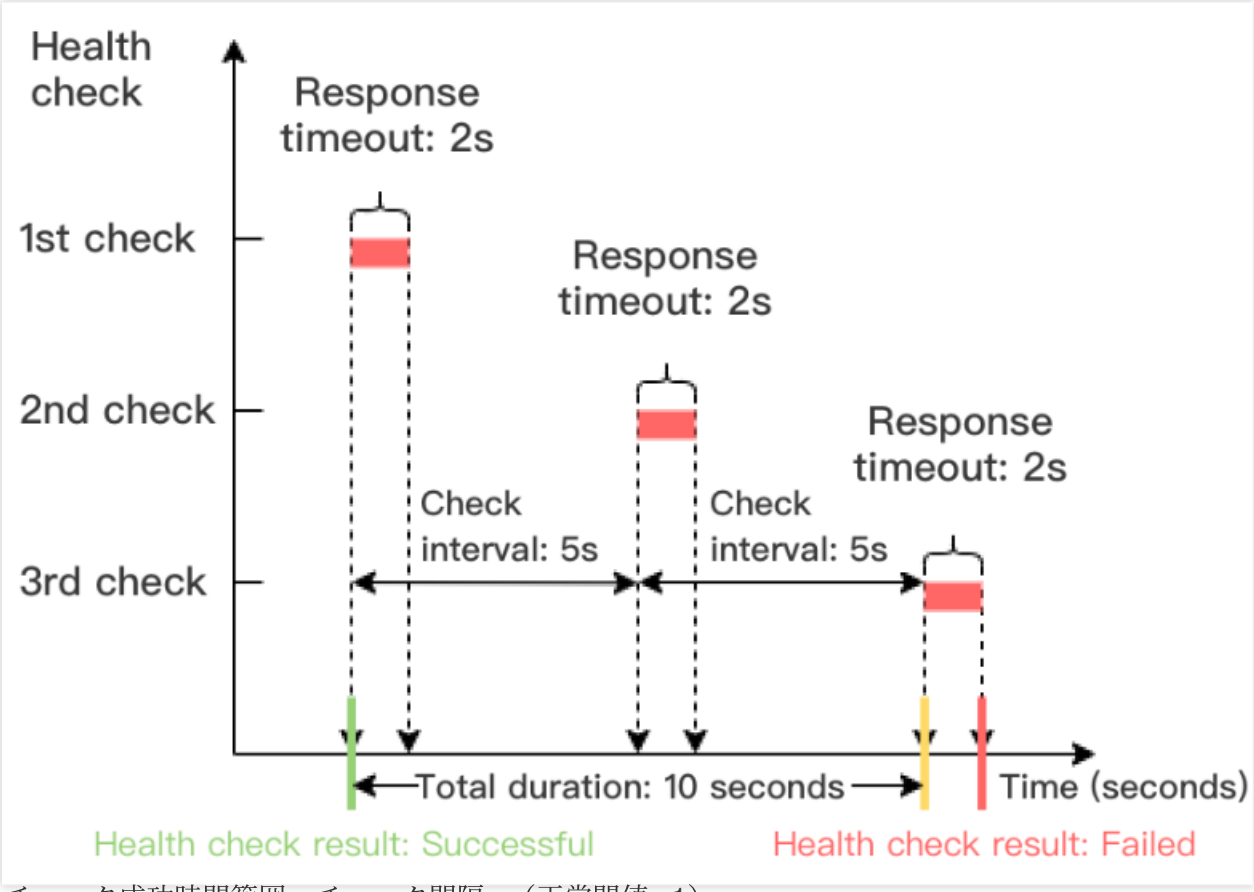
レイヤー4ヘルスチェック時間範囲の計算方法は次のとおりです。

説明：

レイヤー4ヘルスチェック、すなわちTCPヘルスチェックまたはUDPヘルスチェックでは、チェックに成功したか、またはレスポンスタイムアウトかにかかわらず、前後2回の間の送信パケットのチェック間隔はすべて設定済みのチェック間隔となります。

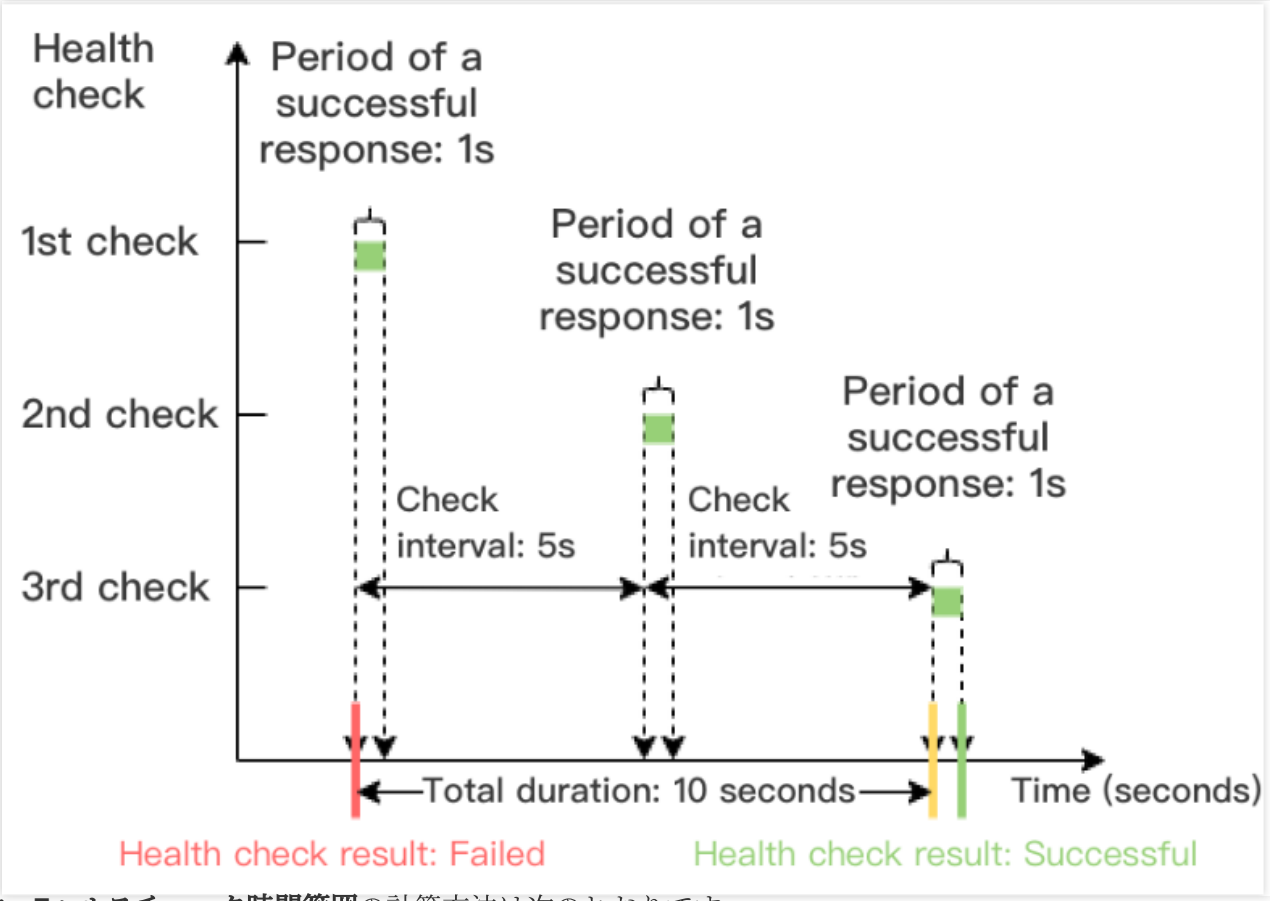
ヘルスチェック失敗時間範囲 = チェック間隔 ×（異常閾値 - 1）

下図はヘルスチェックのレスポンスタイムアウト時間が2秒、チェック間隔が5秒、異常閾値が3回の場合の例です。ヘルスチェック失敗時間範囲は、5×（3-1）= 10sとなります。



ヘルスチェック成功時間範囲 = チェック間隔 ×（正常閾値 - 1）

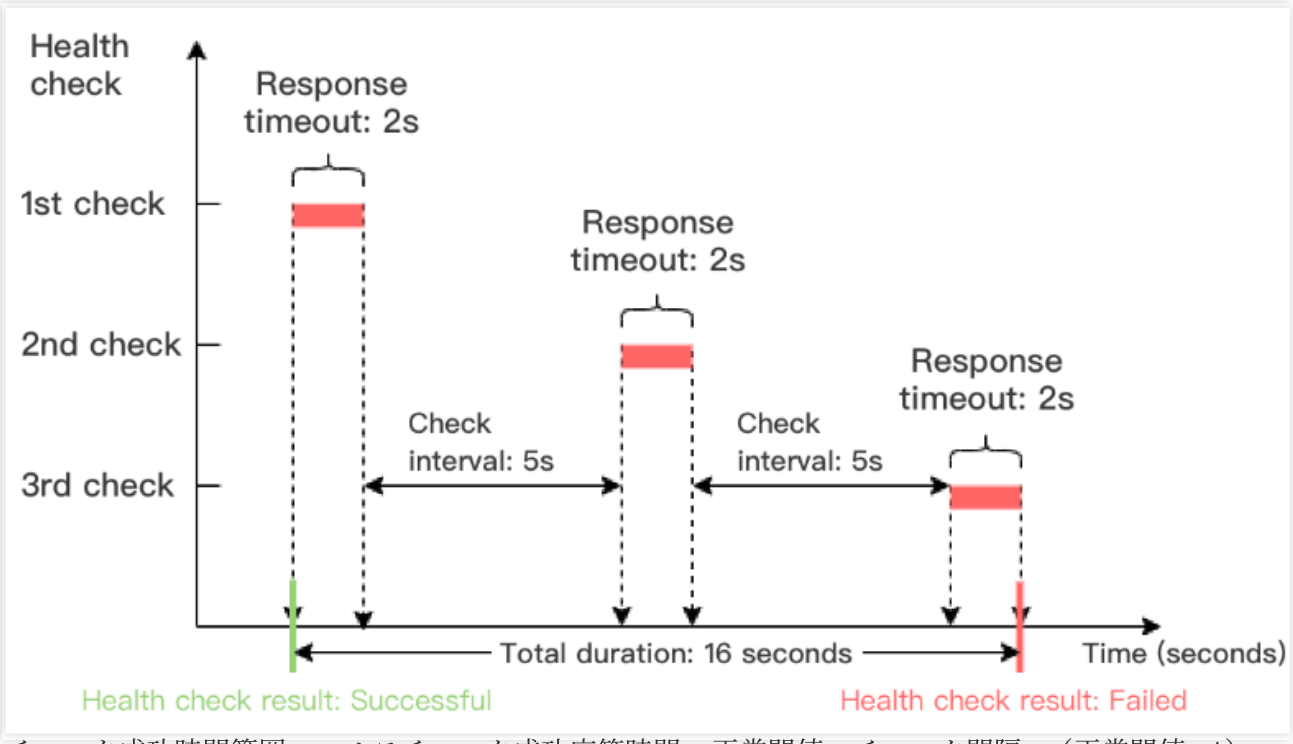
下の図はヘルスチェック成功応答時間が1秒、チェック間隔が5秒、正常閾値が3回の場合の例です。ヘルスチェック成功時間範囲 = 5 x（3-1）= 10秒となります。



レイヤー7ヘルスチェック時間範囲の計算方法は次のとおりです。

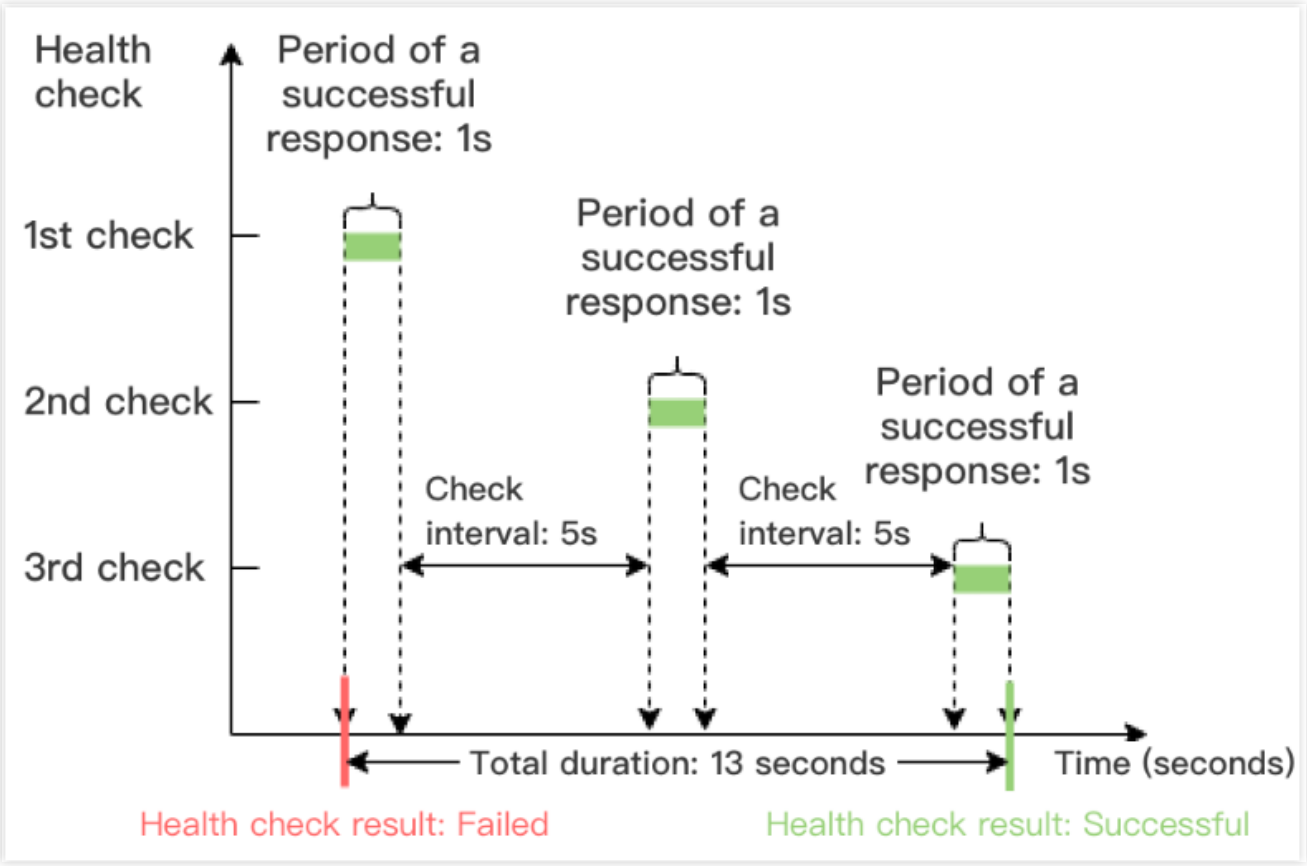
ヘルスチェック失敗時間範囲 = 応答タイムアウト時間 × 異常閾値 + チェック間隔 ×（異常閾値 - 1）

下の図はヘルスチェックの応答タイムアウト時間が2秒、チェック間隔が5秒、異常閾値が3回の場合の例です。ヘルスチェック失敗時間範囲 = 2 x 3 + 5 x（3-1）= 16秒となります。



ヘルスチェック成功時間範囲 = ヘルスチェック成功応答時間 × 正常閾値 + チェック間隔 ×（正常閾値 - 1）

下図はヘルスチェック成功レスポンス時間が1秒、チェック間隔が5秒、正常閾値が3回の場合の例です。ヘルスチェック成功時間範囲 = 1×3+ 5×（3-1）=13sとなります。



ヘルスチェック検出識別子

CLBでヘルスチェックを有効化すると、バックエンドサーバーは正常な業務リクエストに加えて、ヘルスチェックリクエストも受信することになります。ヘルスチェックリクエストは次の識別子を有します。
ヘルスチェックプローブリクエストのソースIPは、CLBのVIPまたは100.64.0.0/10ネットワークセグメントです。
レイヤー4(TCP、UDP、TCP SSL)リスナーのヘルスチェックリクエストには、「HEALTH CHECK」の識別子が含まれます。
レイヤー7(HTTP、HTTPS)リスナーのヘルスチェックリクエストHeaderのuser-agentは、「clb-healthcheck」です。
説明：
従来型プライベートネットワークCLBでは、ヘルスチェックのソースIPは `169.254.128.0/17` ネットワークセグメントです。
基幹ネットワークのプライベートネットワークCLBでは、ヘルスチェックのソースIPはサーバーの物理IPとなります。

関連ドキュメント

ヘルスチェックの設定
ヘルスチェックログの設定
アラートポリシーの設定

ヘルスチェックの設定

最終更新日：：2023-05-09 15:10:44

リスナーの設定の際にヘルスチェック機能を有効化し、バックエンドサービスの可用性を判断することができます。ヘルスチェックの詳細については、ヘルスチェックの概要をご参照ください。

制限事項

IPv6バージョンのCLBのTCPリスナーでは、HTTPヘルスチェックおよびカスタムプロトコルヘルスチェックはサポートしていません。
IPv6バージョンのCLBのUDPリスナーでは、ポートチェック方式のヘルスチェックはサポートしていません。

前提条件

1. CLBインスタンスを作成済みであること。詳細については、CLBインスタンスの作成をご参照ください。
2. CLBリスナーを作成済みであること。
TCPリスナーの作成については、TCPリスナーの設定をご参照ください。
UDPリスナーの作成については、UDPリスナーの設定をご参照ください。
TCP SSLリスナーの作成については、TCP SSLリスナーの設定をご参照ください。
HTTPリスナーの作成については、HTTPリスナーの設定をご参照ください。
HTTPSリスナーの作成については、HTTPSリスナーの設定をご参照ください。

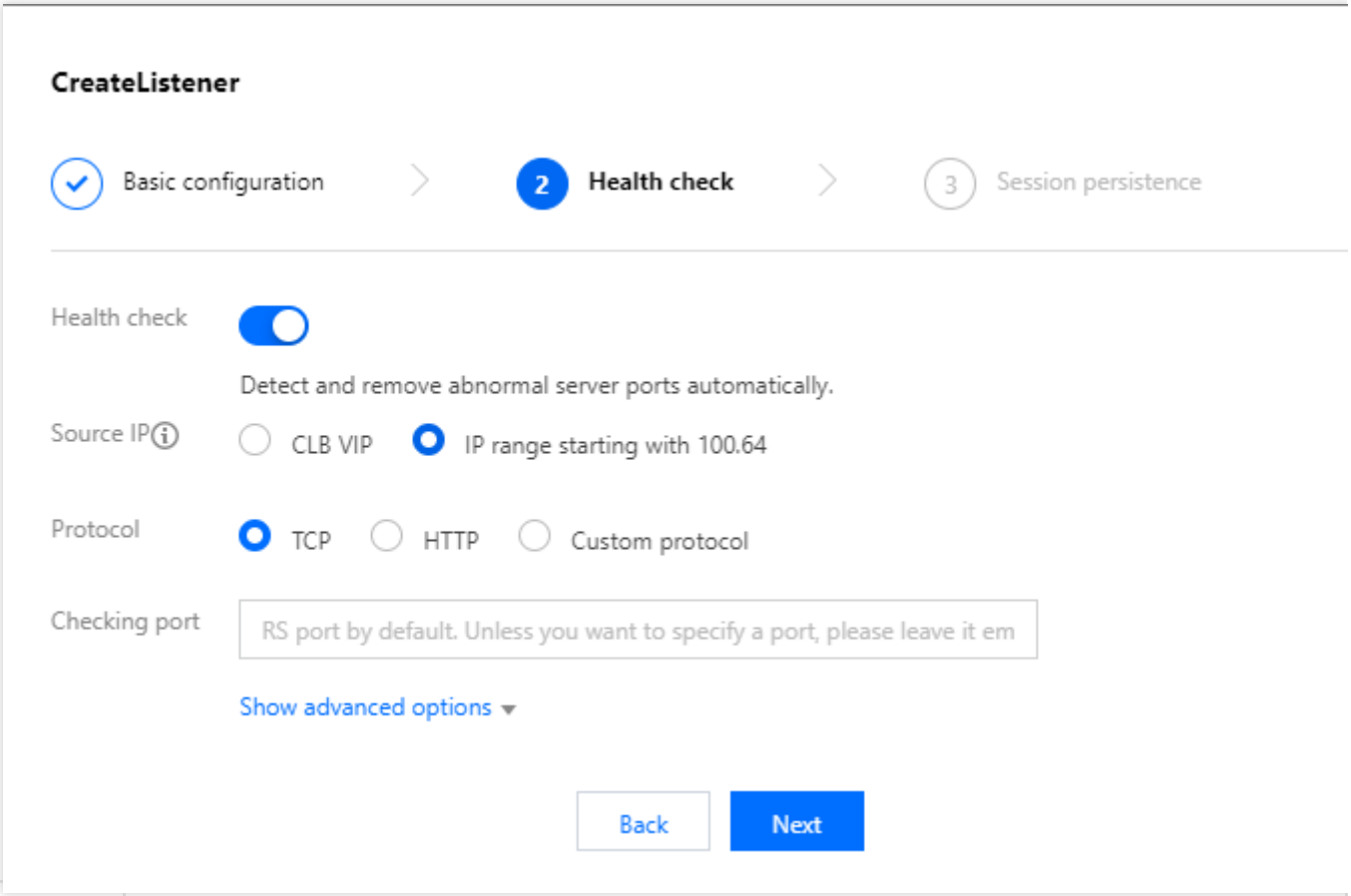
TCPリスナー

レイヤー4 TCPリスナーは、レイヤー4 TCP、レイヤー7 HTTPおよびカスタムプロトコルの3タイプのヘルスチェックをサポートしています。
TCPヘルスチェックでは、SYNパケット、すなわちTCPの3ウェイハンドシェイクの開始によって、バックエンドサーバーのステータス情報を取得します。
HTTPヘルスチェックでは、HTTPリクエストの送信によって、バックエンドサーバーのステータス情報を取得します。
カスタムプロトコルはアプリケーション層プロトコルの入力および出力内容をカスタマイズすることによって、バックエンドサーバーのステータス情報を取得します。

TCPヘルスチェックの設定

1. 前提条件を参照し、「ヘルスチェック」タブで操作を行います。

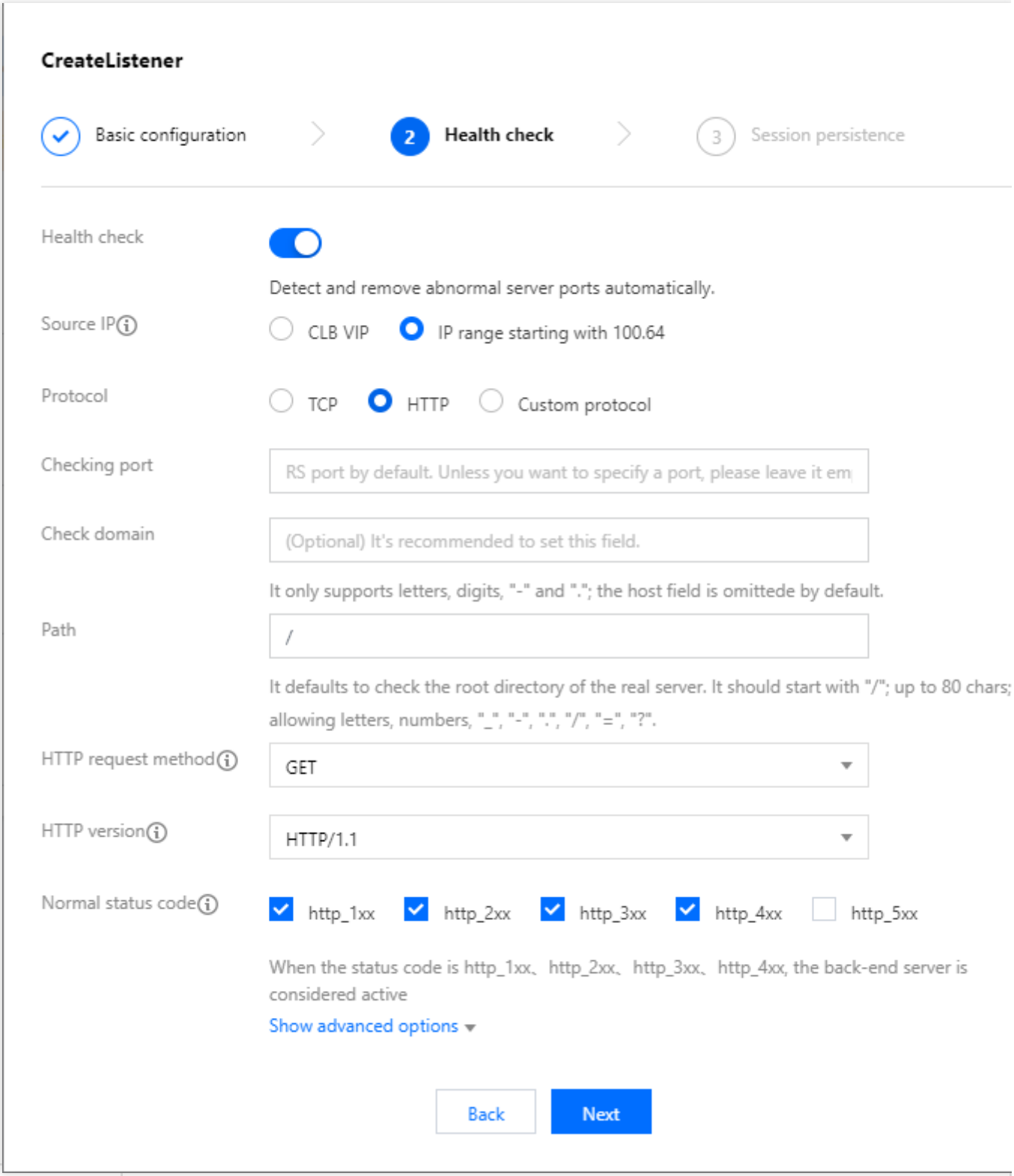
2. 「ヘルスチェックタブで、「TCP」チェック方法を選択します。



パラメータ	説明
ヘルスチェック	ヘルスチェック機能を有効化または無効化できます。ヘルスチェックを有効化すると、異常なバックエンドサーバーポートを自動的にチェックして削除する際に役立ちますので、ヘルスチェックを有効化しておくことをお勧めします。
ヘルスチェックソースIP	ヘルスチェックプローブパケットのソースIPで、デフォルトでは100.64.0.0/10ネットワークセグメントです。このネットワークセグメントを使用して、Tencent Kubernetes Engine(TKE)で発生するコンテナのループバックの問題を解決することができます。既存ユーザーは、ヘルスプローブのソースIPとして、CLBのVIPを選択できます。
チェック方法	TCPヘルスチェックを設定する場合は、「TCP」を選択します。
チェックポート	入力必須ではありません。ポートを入力しない場合は、デフォルトでバックエンドサーバーポートとなります。特定のポートを指定したい場合を除き、入力しないことをお勧めします。
高度なオプションの表示	詳細については、高度なオプションをご参照ください。

HTTPヘルスチェックの設定

1. 前提条件を参照し、「ヘルスチェック」タブで操作を行います。
2. 「ヘルスチェック」タブで、「HTTP」チェック方法を選択します。



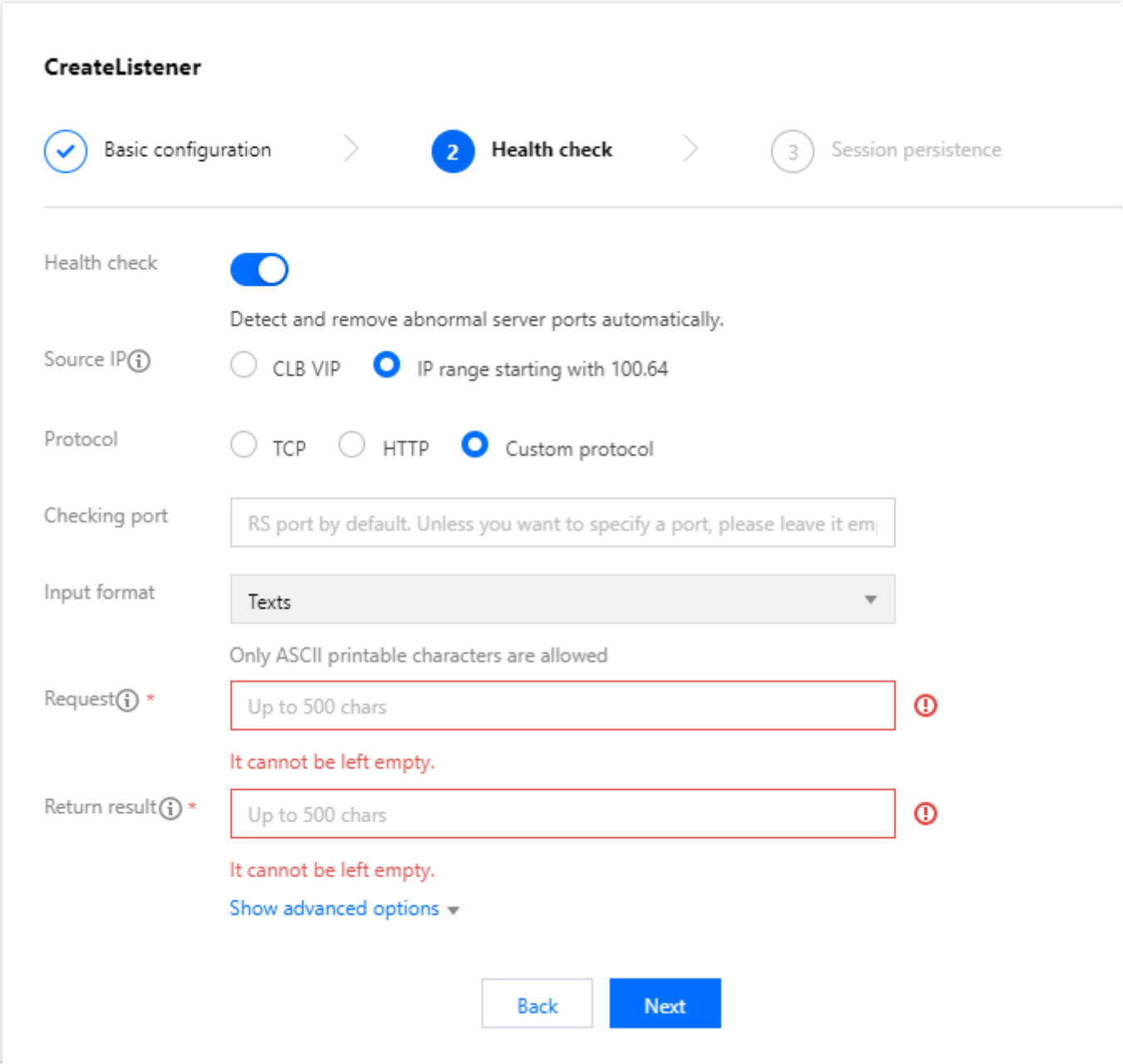
パラメータ	説明
ヘルスチェッ	ヘルスチェック機能を有効化または無効化できます。ヘルスチェックを有効化すると、異

ク	常なバックエンドサーバーポートを自動的にチェックして削除する際に役立ちますので、ヘルスチェックを有効化しておくことをお勧めします。
ヘルスチェックソースIP	ヘルスチェックプローブパケットのソースIPで、デフォルトでは100.64.0.0/10ネットワークセグメントです。このネットワークセグメントを使用して、Tencent Kubernetes Engine(TKE)で発生するコンテナのループバックの問題を解決することができます。既存ユーザーは、ヘルスプローブのソースIPとして、CLBのVIPを選択できます。
チェック方法	HTTPヘルスチェックを設定する場合は、「HTTP」を選択します。
チェックポート	入力必須ではありません。ポートを入力しない場合は、デフォルトでバックエンドサーバーポートとなります。特定のポートを指定したい場合を除き、入力しないことをお勧めします。
チェックドメイン名	ヘルスチェックドメイン名 長さ制限：1～80文字です。 デフォルトではドメイン名を転送します。 正規表現をサポートしていません。転送ドメイン名がワイルドカードドメイン名の場合は、ある特定のドメイン名（非正規表現）をヘルスチェックドメイン名として指定する必要があります。 サポートされている文字セットは、a-z 0-9 . -です。
チェックパス	ヘルスチェックパス ： 長さ制限：1～200文字です。 デフォルトでは/であり、必ず/で始めなければなりません。 正規表現をサポートしていません。ある特定のURLパス（静的ページ）を指定してヘルスチェックを行うことをお勧めします。 サポートされている文字セットは、a-z A-Z 0-9 . - _ / = ?です。
HTTPリクエスト方法	ヘルスチェックのHTTPリクエストメソッドです。GETまたはHEADを選択でき、デフォルトではGETです。 HEADメソッドを使用する場合、サーバーはHTTPヘッダー情報のみを返すため、バックエンドのオーバーヘッドを低減し、リクエスト効率を向上させることができます。対応するバックエンドサービスがHEADをサポートしている必要があります。 GETメソッドを使用する場合は、バックエンドサービスがGETをサポートしていれば使用可能です。
HTTPバージョン	バックエンドサービスのHTTPバージョンです。 バックエンドサーバーがサポートするHTTPバージョンが1.0の場合は、リクエストのHostフィールドの検証は不要、つまりチェックドメイン名を設定する必要はありません。 バックエンドサーバーがサポートするHTTPバージョンが1.1の場合は、リクエストのHostフィールドの検証が必要、つまりチェックドメイン名を設定する必要があります。 **説明**：HTTP /1.1バージョンを選択した時に、チェックするドメイン名をまだ設定していない場合、HTTP標準プロトコルに基づき、バックエンドサーバーは400エラーコードを返し、ヘルスチェックが異常であることを示します。この場合、正常なステータスコードhttp_4xxをチェックすることをお勧めします。

正常なステータスコード	ステータスコードが選択したステータスコードの場合、バックエンドサーバーは稼働している、つまりヘルスチェックは正常であるとみなされます。http_1xx、http_2xx、http_3xx、http_4xx、http_5xxを選択できます。
高度なオプションの表示	詳細については、高度なオプションをご参照ください。

カスタムプロトコルヘルスチェックの設定

1. 前提条件を参照し、「ヘルスチェック」タブで操作を行います。
2. 「ヘルスチェック」タブで、「カスタムプロトコル」チェック方法を選択します。



パラメータ	説明

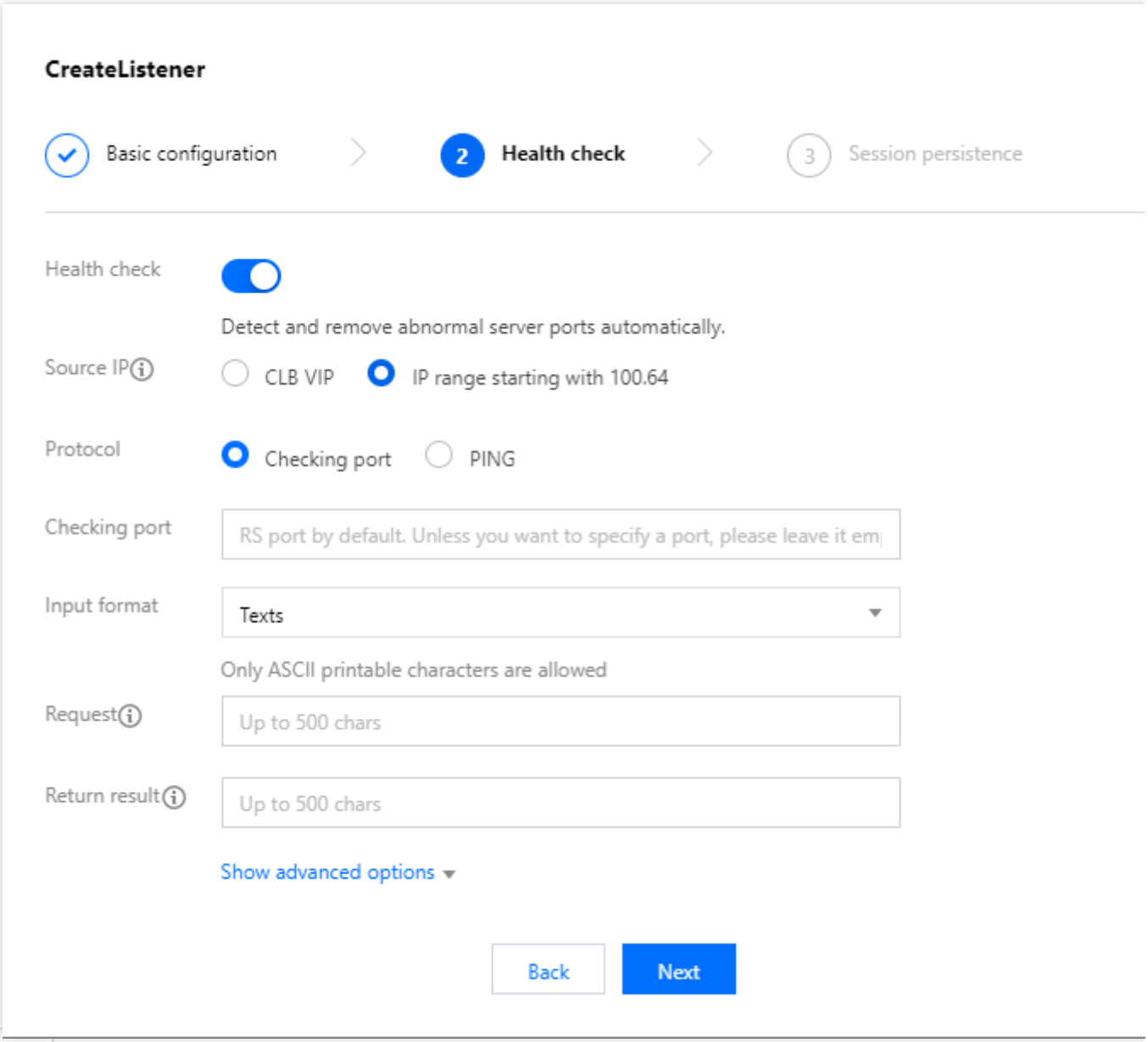
ヘルスチェック	ヘルスチェック機能を有効化または無効化できます。ヘルスチェックを有効化すると、異常なバックドサーバーポートを自動的にチェックして削除する際に役立ちますので、ヘルスチェックを有効化しくことをお勧めします。
ヘルスチェックソースIP	ヘルスチェックプローブパケットのソースIPで、デフォルトでは100.64.0.0/10ネットワークセグメンす。このネットワークセグメントを使用して、Tencent Kubernetes Engine(TKE)で発生するコンテナのループバックの問題を解決することができます。既存ユーザーは、ヘルスプローブのソースIPとして、CLBのVIPを選択できます。
チェック方法	「カスタムプロトコル」を選択する場合、カスタムプロトコルヘルスチェックを設定することになりす。TCPの、HTTP以外のプロトコルに適用可能です。
チェックポート	入力必須ではありません。ポートを入力しない場合は、デフォルトでバックエンドサーバーポートとます。特定のポートを指定したい場合を除き、入力しないことをお勧めします。
入力形式	テキスト入力と16進数入力をサポートしています。 入力形式をテキストにするとは、テキストをバイナリーに変換してリクエストを送信し、返された結の比較を行うことです。 入力形式を16進数にするとは、16進数をバイナリーに変換してリクエストを送信し、返された結果と較を行うことです。
チェックリクエスト	カスタムヘルスチェックリクエストの内容であり、入力必須です。例えば、DNSサービスを検出するのチェックリクエストの例は、F13E010000010000000000000377777704746573740363 6F6D0774656E63656E7403636F6D000001のようになります。
返されたチェック結果	ヘルスチェックリクエストをカスタマイズする場合は、返されるヘルスチェック結果を入力する必要ります。例えば、DNSサービスを検出する場合に返されるチェック結果の例は、F13Eのようになりま
高度なオプションの表示	詳細については、高度なオプションをご参照ください。

UDPリスナー

UDPリスナーはUDPヘルスチェックをサポートしています。これにはポートチェックとPINGの2つのチェックのタイプが含まれます。

UDPヘルスチェックの設定-ポートチェック

1. 前提条件を参照し、「ヘルスチェック」タブで操作を行います。
2. 「ヘルスチェック」タブで、「カスタム」チェック方法を選択します。

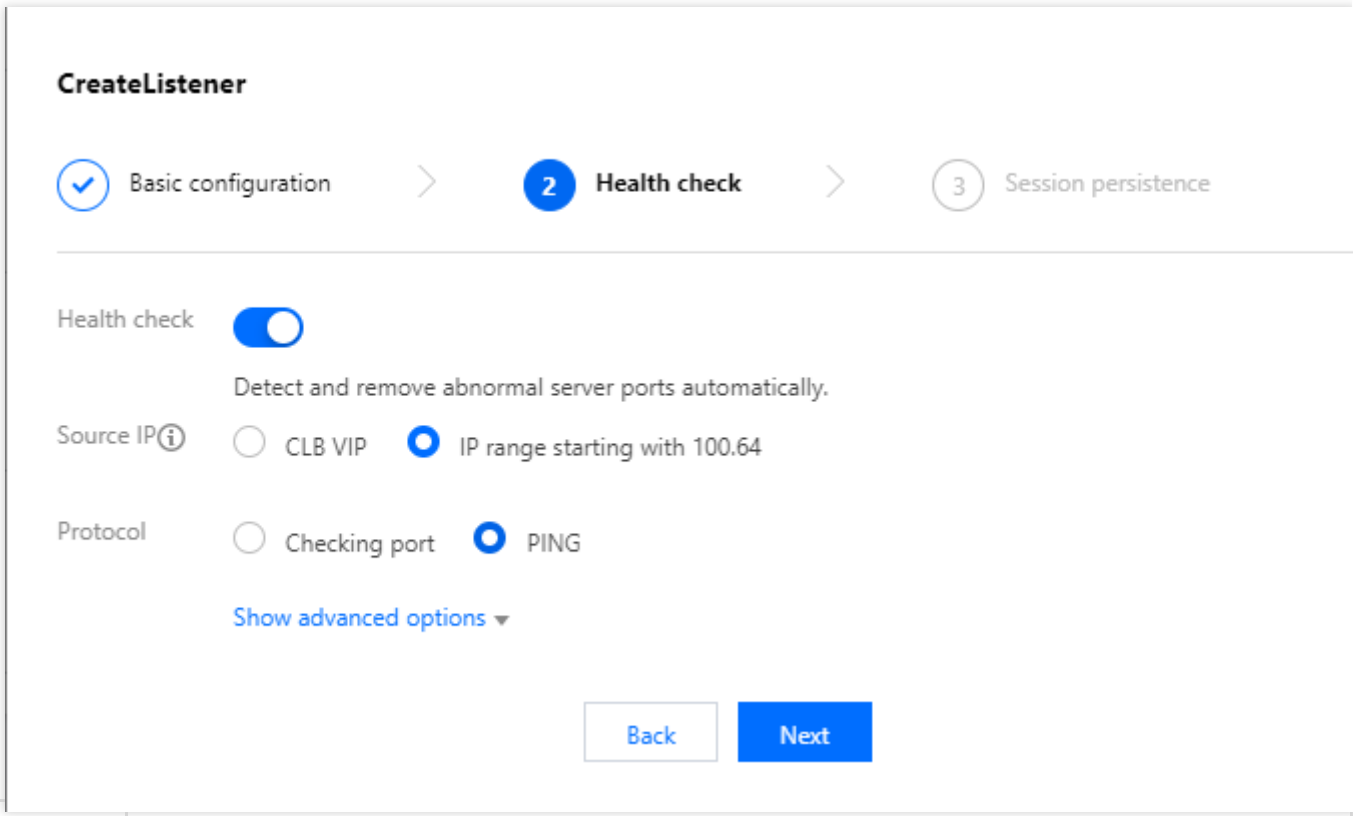


パラメータ	説明
ヘルスチェック	ヘルスチェック機能を有効化または無効化できます。ヘルスチェックを有効化すると、異常なバックドサーバーポートを自動的にチェックして削除する際に役立ちますので、ヘルスチェックを有効化しくことをお勧めします。
ヘルスチェックソースIP	ヘルスチェックプローブパケットのソースIPで、デフォルトでは100.64.0.0/10ネットワークセグメンす。このネットワークセグメントを使用して、Tencent Kubernetes Engine(TKE)で発生するコンテナのループバックの問題を解決することができます。既存ユーザーは、ヘルスプローブのソースIPとして、CLBのVIPを選択できます。
チェッ	「カスタム」を選択すると、ヘルスプローブのソースIPがバックエンドサーバーにUDPプローブメッ

ク方法	ジを送信することにより、バックエンドサーバーのステータス情報を取得することになります。
チェックポート	入力必須ではありません。ポートを入力しない場合は、デフォルトでバックエンドサーバーポートとます。特定のポートを指定したい場合を除き、入力しないことをお勧めします。
入力形式	テキスト入力と16進数入力をサポートしています。 入力形式をテキストにするとは、テキストをバイナリーに変換してリクエストを送信し、返された結の比較を行うことです。 入力形式を16進数にするとは、16進数をバイナリーに変換してリクエストを送信し、返された結果と較を行うことです。
チェックリクエスト	カスタムヘルスチェックリクエストの内容です。例えば、DNSサービスを検出する場合のチェックリストの例は、 F13E010000010000000000003777777047465737403636F6D0774656E63656E7403636F6D000001のようになります。
返されたチェック結果	ヘルスチェックリクエストをカスタマイズする場合は、返されるヘルスチェック結果を設定する必要ります。例えば、DNSサービスを検出する場合に返されるチェック結果の例は、F13Eのようになりま
高度なオプションの表示	詳細については、高度なオプションをご参照ください。

UDPヘルスチェックの設定-PING

1. 前提条件を参照し、「ヘルスチェック」タブで操作を行います。
2. 「ヘルスチェック」タブで、「PING」チェックプロトコルを選択します。

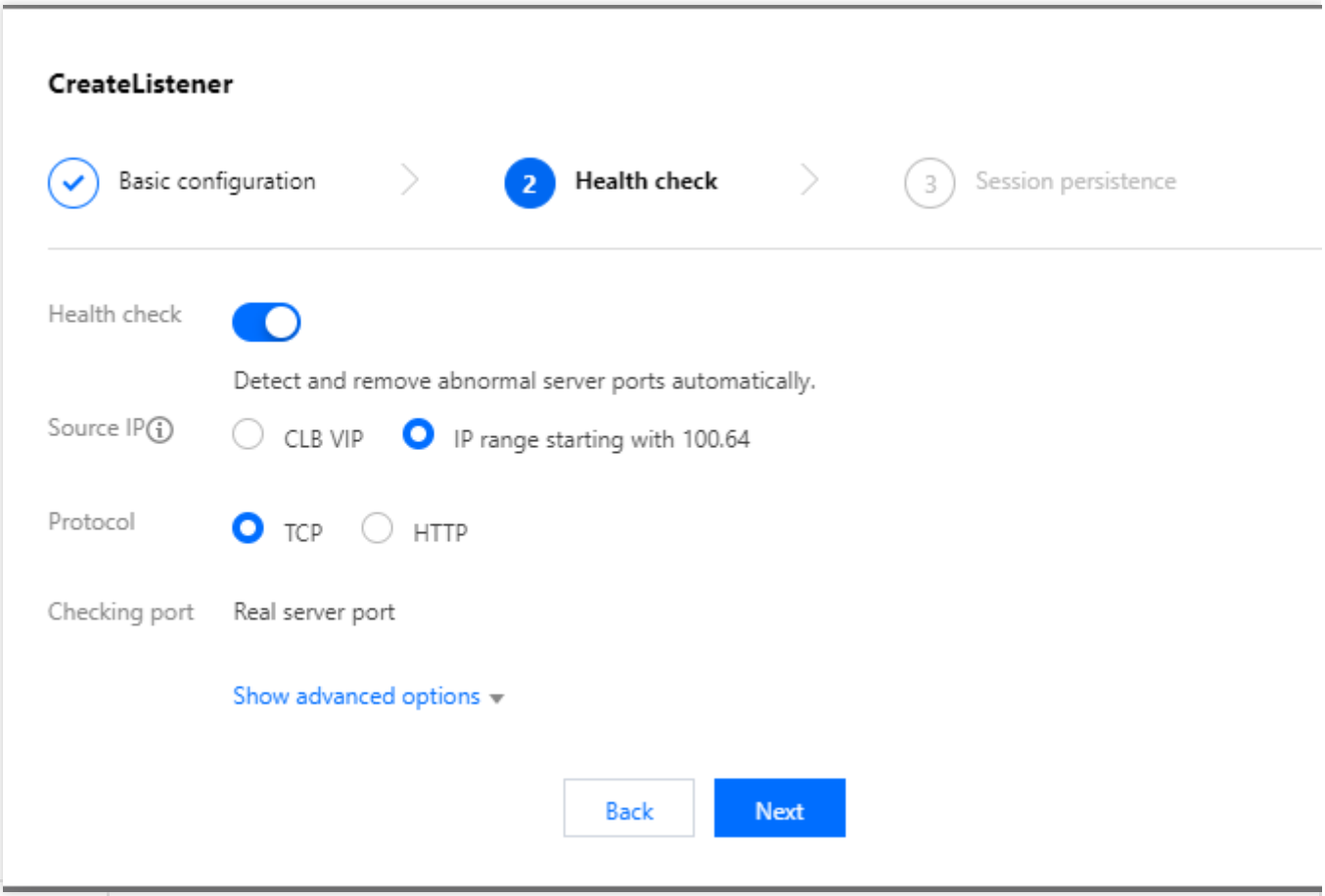


パラメータ	説明
ヘルスチェック	ヘルスチェック機能を有効化または無効化できます。ヘルスチェックを有効化すると、異常なバックエンドサーバーポートを自動的にチェックして削除する際に役立ちますので、ヘルスチェックを有効化しておくことをお勧めします。
ヘルスチェックソースIP	ヘルスチェックプローブパケットのソースIPで、デフォルトでは100.64.0.0/10ネットワークセグメントです。このネットワークセグメントを使用して、Tencent Kubernetes Engine(TKE)で発生するコンテナのループバックの問題を解決することができます。既存ユーザーは、ヘルスプローブのソースIPとして、CLBのVIPを選択できます。
チェックプロトコル	「PING」を選択することは、バックエンドサーバーのIPアドレスをPingすることでバックエンドサーバーのステータス情報を取得することを意味します。
高度なオプションの表示	詳細については、高度なオプションをご参照ください。

TCP SSLリスナー

TCPヘルスチェックの設定

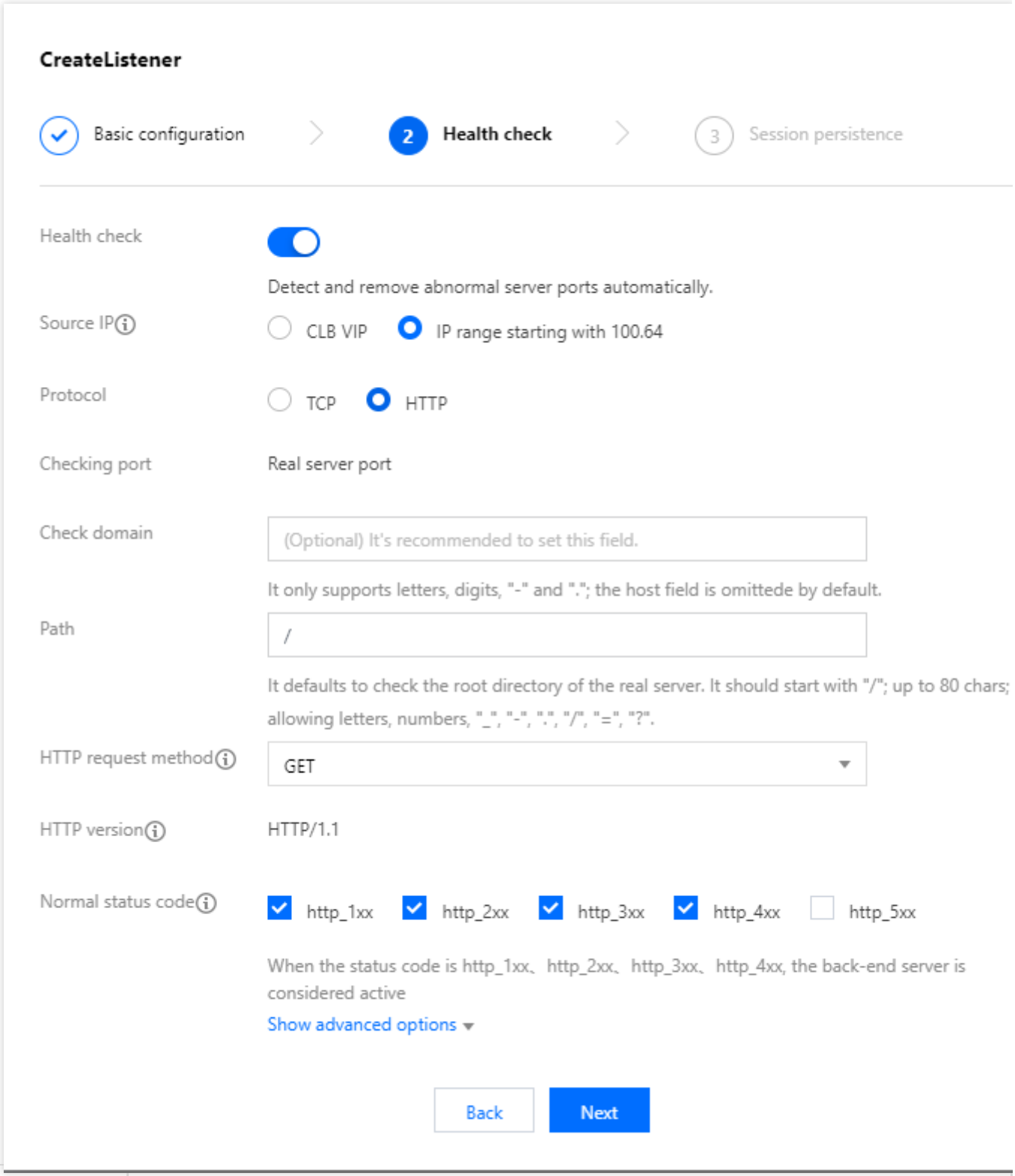
1. 前提条件を参照し、「ヘルスチェック」タブで操作を行います。
2. 「ヘルスチェック」タブで、「TCP」チェック方法を選択します。



パラメータ	説明
ヘルスチェック	ヘルスチェック機能を有効化または無効化できます。ヘルスチェックを有効化すると、異常なバックエンドサーバーポートを自動的にチェックして削除する際に役立ちますので、ヘルスチェックを有効化しておくことをお勧めします。
ヘルスチェックソースIP	ヘルスチェックプローブパケットのソースIPで、デフォルトでは100.64.0.0/10ネットワークセグメントです。このネットワークセグメントを使用して、Tencent Kubernetes Engine(TKE)で発生するコンテナのループバックの問題を解決することができます。既存ユーザーは、ヘルスプローブのソースIPとして、CLBのVIPを選択できます。
チェック方法	TCPヘルスチェックを設定する場合は、「TCP」を選択します。
チェックポート	TCP SSLリスナーのヘルスチェックポートは、リスニングポートと同じです。
高度なオプションの表示	詳細については、高度なオプションをご参照ください。

HTTPヘルスチェックの設定

1. 前提条件を参照し、「ヘルスチェック」タブで操作を行います。
2. 「ヘルスチェック」タブで、「HTTP」チェック方法を選択します。



パラメータ	説明
ヘルスチェック	ヘルスチェック機能を有効化または無効化できます。ヘルスチェックを有効化すると、異常なバックエンドサーバーポートを自動的にチェックして削除する際に役立ちますので、

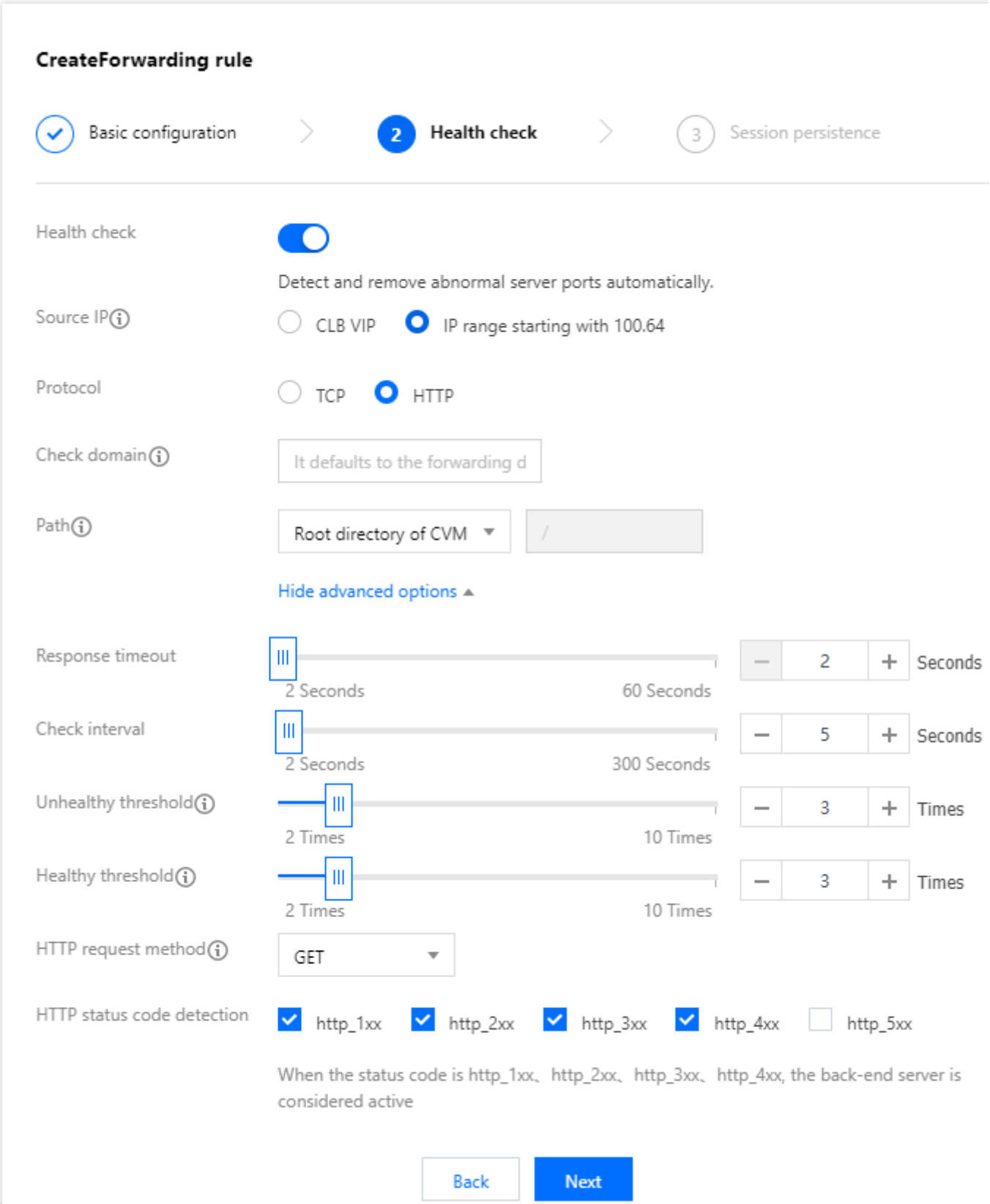
	ヘルスチェックを有効化しておくことをお勧めします。
ヘルスチェックソースIP	ヘルスチェックプローブパケットのソースIPで、デフォルトでは100.64.0.0/10ネットワークセグメントです。このネットワークセグメントを使用して、Tencent Kubernetes Engine(TKE)で発生するコンテナのループバックの問題を解決することができます。既存ユーザーは、ヘルスプローブのソースIPとして、CLBのVIPを選択できます。
チェック方法	HTTPヘルスチェックを設定する場合は、「HTTP」を選択します。
チェックポート	TCP SSLリスナーのヘルスチェックポートは、リスニングポートと同じです。
チェックドメイン名	ヘルスチェックドメイン名 長さ制限：1～80文字です。 デフォルトではドメイン名を転送します。 正規表現をサポートしていません。転送ドメイン名がワイルドカードドメイン名の場合は、ある特定のドメイン名（非正規表現）をヘルスチェックドメイン名として指定する必要があります。 サポートされている文字セットは、a-z 0-9 . -です。
チェックパス	ヘルスチェックパス ： 長さ制限：1～200文字です。 デフォルトでは/であり、必ず/で始めなければなりません。 正規表現をサポートしていません。ある特定のURLパス（静的ページ）を指定してヘルスチェックを行うことをお勧めします。 サポートされている文字セットは、a-z A-Z 0-9 . - _ / = ?です。
HTTPリクエスト方法	ヘルスチェックのHTTPリクエストメソッドです。GETまたはHEADを選択でき、デフォルトではGETです。 HEADメソッドを使用する場合、サーバーはHTTPヘッダー情報のみを返すため、バックエンドのオーバーヘッドを低減し、リクエスト効率を向上させることができます。対応するバックエンドサービスがHEADをサポートしている必要があります。 GETメソッドを使用する場合は、バックエンドサービスがGETをサポートしていれば使用可能です。
HTTPバージョン	バックエンドサービスのHTTPバージョンで、HTTP1.1バージョンのみをサポートしています。バックエンドサービスは、リクエストのHostフィールドの検証すなわちドメイン名のチェックを設定する必要があります。 **説明**：チェックするドメイン名をまだ設定していない場合、HTTP標準プロトコルに基づき、バックエンドサーバーは400エラーコードを返し、ヘルスチェックが異常であることを示します。正常なステータスコードhttp_4xxをチェックすることをお勧めします。
正常なステータスコード	ステータスコードが選択したステータスコードの場合、バックエンドサーバーは稼働している、つまりヘルスチェックは正常であるとみなされます。http_1xx、http_2xx、http_3xx、http_4xx、http_5xxを選択できます。
高度なオプ	詳細については、高度なオプションをご参照ください。

ションの表示	

HTTPリスナー

HTTPヘルスチェックの設定

1. 前提条件を参照し、「ヘルスチェック」タブで操作を行います。



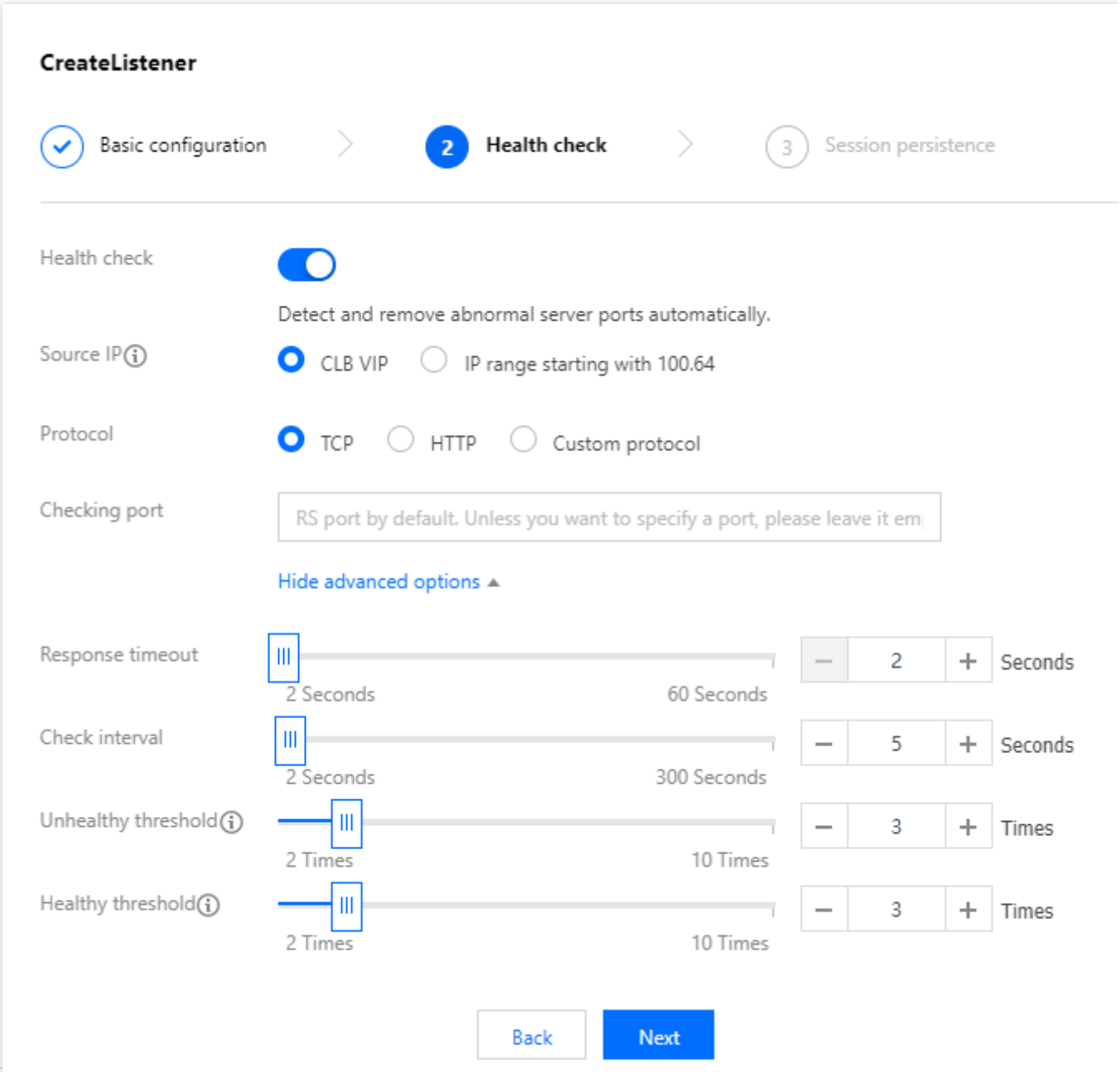
パラメータ	説明
ヘルスチェック	ヘルスチェック機能を有効化または無効化できます。ヘルスチェックを有効化すると、異常なバックエンドサーバーポートを自動的にチェックして削除する際に役立ちますので、ヘルス

	チェックを有効化しておくことをお勧めします。
ヘルスチェックソースIP	ヘルスチェックプローブパケットのソースIPで、デフォルトでは100.64.0.0/10ネットワークセグメントです。このネットワークセグメントを使用して、Tencent Kubernetes Engine(TKE)で発生するコンテナのループバックの問題を解決することができます。既存ユーザーは、ヘルスプローブのソースIPとして、CLBのVIPを選択できます。
チェックドメイン名	ヘルスチェックドメイン名 長さ制限：1～80文字です。 デフォルトではドメイン名を転送します。 正規表現をサポートしていません。転送ドメイン名がワイルドカードドメイン名の場合は、ある特定のドメイン名（非正規表現）をヘルスチェックドメイン名として指定する必要があります。 サポートされている文字セットは、a-z 0-9 . -です。
チェックパス	ヘルスチェックパスは、バックエンドサーバーのルートディレクトリまたは指定のURLに設定できます。 長さ制限：1～200文字です。 デフォルトでは/であり、必ず/で始めなければなりません。 正規表現をサポートしていません。ある特定のURLパス（静的ページ）を指定してヘルスチェックを行うことをお勧めします。 サポートされている文字セットは、a-z A-Z 0-9 . - _ / = ?です。
レスポンスタイムアウト	ヘルスチェックのレスポンスの最大タイムアウト時間です。 バックエンドCVMからタイムアウト時間内に正確なレスポンスがない場合は、ヘルスチェックに異常があると判断されます。 設定可能範囲は2～60秒です。
チェック間隔	CLBがヘルスチェックを行う時間の間隔です。 設定可能範囲は2～300秒です。
不健全なしきい値	n回（nには数値を入力）連続してヘルスチェック失敗の結果を受信した場合に、異常であると認識し、コンソールで異常と表示します。 設定可能範囲は2～10回です。
健全なしきい値	n回（nには数値を入力）連続してヘルスチェック成功の結果を受信した場合に、正常であると認識し、コンソールで正常と表示します。 設定可能範囲は2～10回です。
HTTPリクエスト方法	ヘルスチェックのHTTPリクエストメソッドです。GETまたはHEADを選択でき、デフォルトではGETです。 HEADメソッドを使用する場合、サーバーはHTTPヘッダー情報のみを返すため、バックエンドのオーバーヘッドを低減し、リクエスト効率を向上させることができます。対応するバックエンドサービスがHEADをサポートしている必要があります。 GETメソッドを使用する場合は、バックエンドサービスがGETをサポートしていれば使用可能です。

正常なステータスコード	ステータスコードが選択したステータスコードの場合、バックエンドサーバーは稼働している、つまりヘルスチェックは正常であるとみなされます。http_1xx、http_2xx、http_3xx、http_4xx、http_5xxを選択できます。

TCPヘルスチェックの設定

1. 前提条件を参照し、「ヘルスチェック」タブで操作を行います。
2. 「ヘルスチェック」タブで、「TCP」チェックプロトコルを選択します。



パラメータ	説明
ヘルスチェック	ヘルスチェック機能を有効化または無効化できます。ヘルスチェックを有効化すると、異常なバックエンドサーバーポートを自動的にチェックして削除する際に

	役立ちますので、ヘルスチェックを有効化しておくことをお勧めします。
ヘルスソースプローブIP	ヘルスチェックプローブパケットのソースIPで、デフォルトでは100.64.0.0/10ネットワークセグメントです。このネットワークセグメントを使用して、Tencent Kubernetes Engine(TKE)で発生するコンテナのループバックの問題を解決することができます。既存ユーザーは、ヘルスプローブのソースIPとして、CLBのVIPを選択できます。
チェックプロトコル	TCPヘルスチェックを設定する場合は、「TCP」を選択します。
高度なオプションの表示	詳細については、高度なオプションをご参照ください。

HTTPSリスナー

説明：
HTTPSリスナーの転送ルールでバックエンドプロトコルにHTTPプロトコルを選択している場合、ヘルスチェックにはHTTPヘルスチェックを使用します。HTTPSプロトコルを選択している場合、ヘルスチェックにはHTTPSヘルスチェックを使用します。
HTTPSリスナーのヘルスチェックの設定については、上記のHTTPSリスナーのヘルスチェックをご参照ください。

高度なオプション

ヘルスチェックの設定	説明	デフォルト値
レスポンスタイムアウト	ヘルスチェックのレスポンスの最大タイムアウト時間です。 バックエンドCVMからタイムアウト時間内に正確なレスポンスがない場合は、ヘルスチェックに異常があると判断されます。 設定可能範囲は2～60秒です。	2秒
チェック間隔	CLBがヘルスチェックを行う時間の間隔です。 設定可能範囲は2～300秒です。	5秒
不健全なしきい値	n回（nには数値を入力）連続してヘルスチェック失敗の結果を受信した場合に、異常であると認識し、コンソールで**異常**と表示します。 設定可能範囲は2～10回です。	3回
健全なしきい値	n回（nには数値を入力）連続してヘルスチェック成功の結果を受信した場合に、正常であると認識し、コンソールで**正常**と表示します。	3回

	設定可能範囲は2～10回です。	

関連ドキュメント

ヘルスチェックの概要
アラートポリシーの設定

ヘルスチェックのソースIP　非VIPをサポート

最終更新日：：2024-01-04 18:36:26

ここでは、CLBヘルスチェックのソースIPを、CLBの仮想サービスアドレス（VIP）から `100.64.0.0/10` ネットワークセグメントに設定する方法について、TCPリスナーを例としてご説明します。

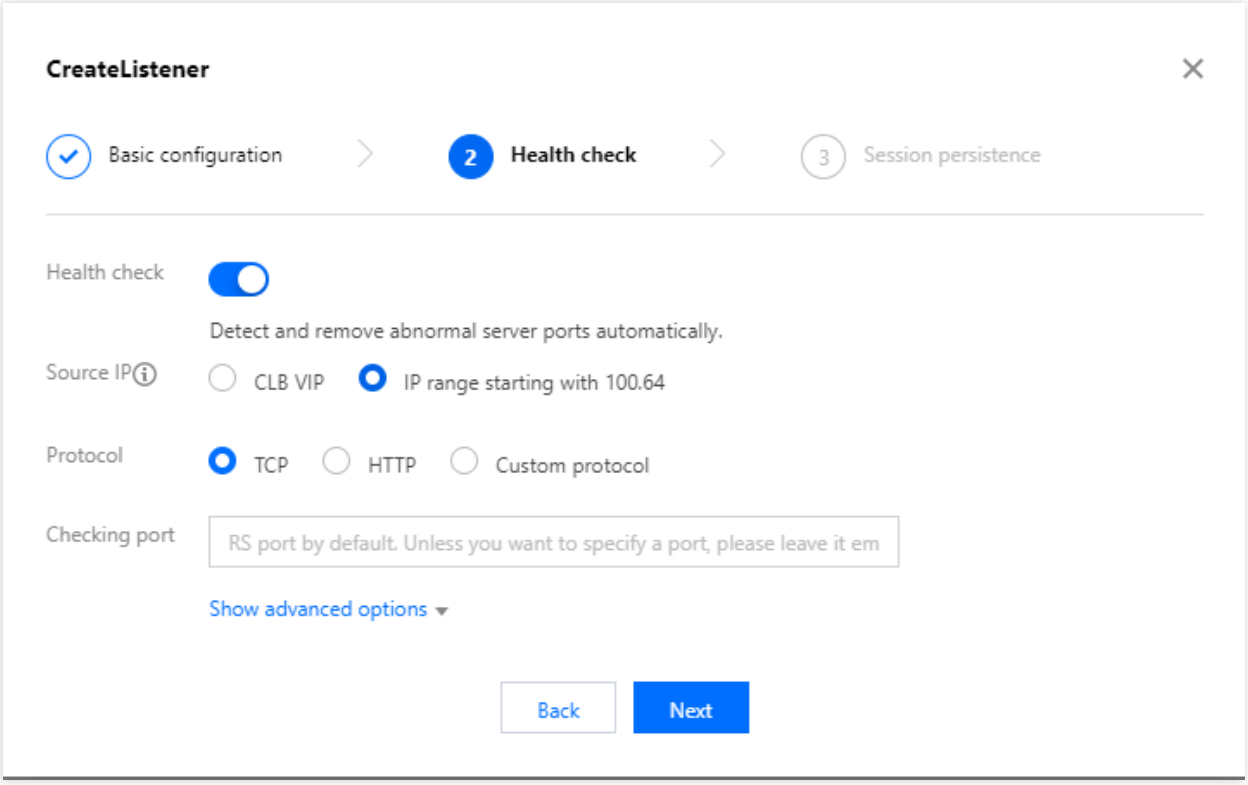
シナリオ

1. **バックエンドサーバーのセキュリティグループの集約**
ヘルスチェックのソースIPを100.64.0.0/10のネットワークセグメントに集約します。
2. **自作Kubernetesクラスタープライベートネットワークのループバック問題の解決**
K8sサービスは、クラスター内とクラスター外の両方で公開する必要があります。クラスター内はクラスター内CLB(IPVS)によって、クラスター外はプライベートネットワークCLBによって実装されます。IPVSは、プライベートネットワークCLBのIPアドレスをローカルのインターフェースにバインドします。これにより、クラスター内のプライベートネットワークCLBへアクセスするためのアドレスには、実際はクラスター内のIPVS CLBが使用されます。
TKEでは、プライベートネットワークCLBがCLBのVIPアドレスをヘルスチェックのソースIPとして使用します。これは、ネイティブのK8s実装のIPVSにバインドされたアドレスと競合するため、プライベートネットワークCLBのヘルスチェックは失敗します。
ヘルスチェックのソースIPを `100.64.0.0/10` ネットワークセグメントに設定することにより、アドレスの競合を回避し、ヘルスチェックの失敗問題を解決できます。

処理手順

1. CLBコンソールにログインします。
2. **インスタンス管理**ページの左上隅でリージョンを選択し、インスタンスリストから対象のインスタンスを見つけ、**操作**列の**リスナーの設定**をクリックします。
3. **リスナー管理**タブで、対象のリスナーを見つけ、リスナー右側の

アイコンをクリックして、リスナーを編集します。
4. ポップアップした**リスナーの編集**ダイアログボックスで、**次へ**から**ヘルスチェック**タブをクリックします。
5. **ヘルスチェック**タブで、ヘルスチェックのソースIPに**100.64.0.0/10ネットワークセグメント**を選択し、**次へ**をクリックしてから**送信**をクリックします。



よくあるご質問

ヘルスプローブのソースIPを100.64.0.0/10ネットワークセグメントに切り替えると、どんなメリットがありますか。

ヘルスプローブのソースIPにネットワークセグメント100.64.0.0/10を使用する場合、バックエンドサーバーのセキュリティグループ内に、このネットワークセグメントの許可ポリシーを追加設定する必要はありません。バックエンドサーバー内にiptablesなどの他のセキュリティポリシーを設定している場合は、このネットワークセグメントを必ず許可しなければヘルスチェックに失敗します。
バックエンドサーバー集約のセキュリティポリシーは、100.64.0.0/10のネットワークセグメントに統一されています。
100.64.0.0/10ネットワークセグメントはTencent Cloudの内部アドレスであり、このセグメントにユーザーを割り当てることはできないため、アドレス競合の問題は発生しません。

100.64.0.0/10ネットワークセグメントをヘルスプローブのソースIPとして使用する場合、固定IPとなりますか。

プローブIPとして使用されるのは固定IPではなく、100.64.0.0/10ネットワークセグメント内のいずれかのIPです。

関連ドキュメント

ヘルスチェックの設定

ヘルスチェックプローブ識別子

証明書管理
証明書の管理

最終更新日：：2024-01-04 18:36:26

CLBのHTTPSリスナーを設定する際に、 SSL証明書サービスの証明書を直接使用するか、もしくはサードパーティが発行した必要なサーバー証明書とSSL証明書をCLBにアップロードすることができます。

証明書の要件

CLBはPEM形式の証明書のみサポートしています。証明書をアップロードする前に、証明書、証明書チェーン、秘密鍵が形式の要件に合っていることを確認してください。証明書の要件については、証明書の要件および証明書形式の変換をご参照ください。

証明書の暗号化アルゴリズム

CLBがサポートする証明書暗号化アルゴリズムはECC暗号化アルゴリズムおよびRSA暗号化アルゴリズムを含みます。暗号化アルゴリズムの具体的な内容についてはRSA暗号化アルゴリズムとECC暗号化アルゴリズムの違いで確認できます。

説明：

HTTPSリスナーのSSL解析のサーバー証明書は2つの証明書の設定をサポートしています。すなわち2種類の異なる暗号化アルゴリズムタイプの証明書です。詳細については、HTTPSリスナーの設定をご参照ください。

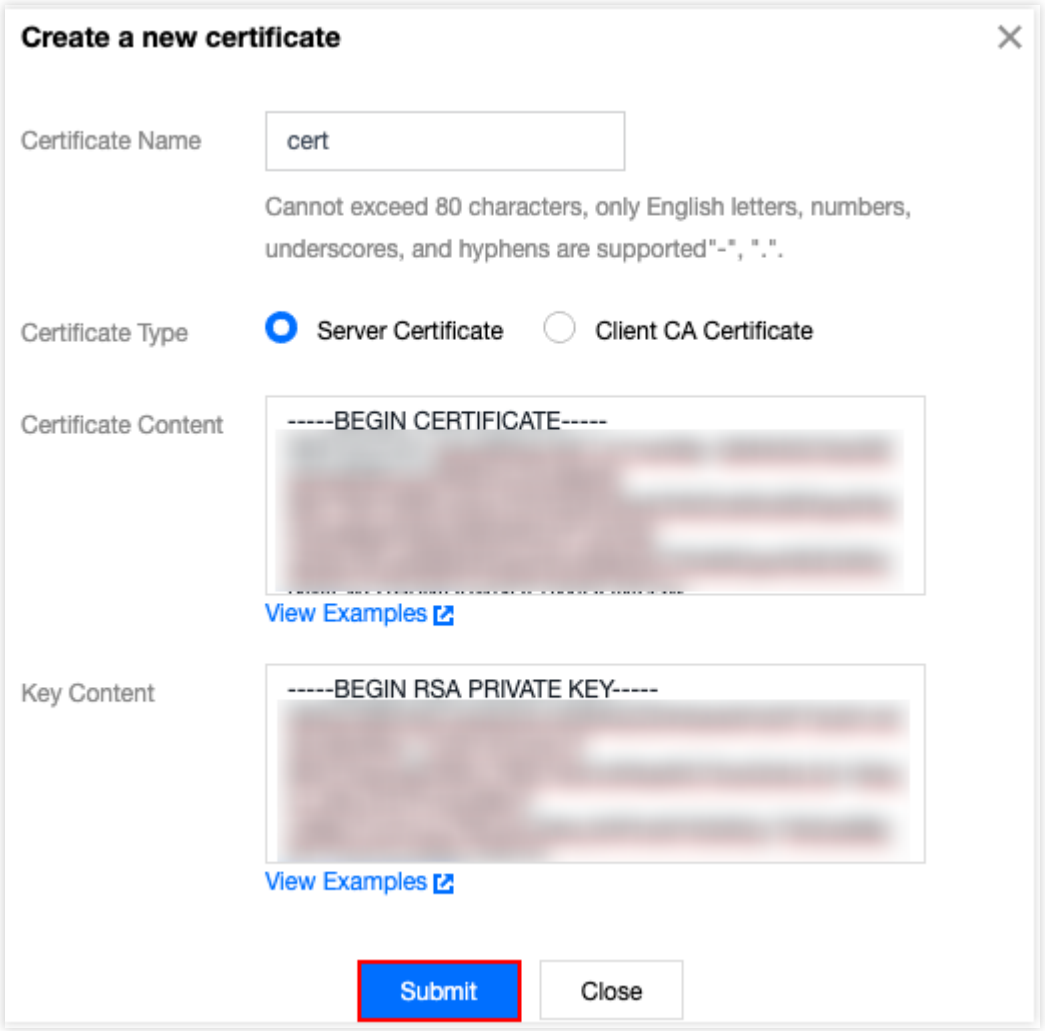
リスナータイプ	単一の証明書の設定がサポートする暗号化アルゴリズム	2つの証明書の設定がサポートする暗号化アルゴリズム
HTTPS	RSAまたはECC	RSAおよびECC
TCP_SSL、QUIC	RSAまたはECC	2種類の異なるタイプの暗号化アルゴリズムの証明書の設定はサポートしていません
TCP、UDP、HTTP	証明書の設定はサポートしていません	証明書の設定はサポートしていません

証明書構成

HTTPSリスナーに証明書を設定する方法には次の2種類があります。
SNIを有効化せず、リスナーのディメンションで証明書を設定します。このリスナー下のすべてのドメイン名が同一の証明書を使用することになります。詳細については、リスナーディメンションでの証明書設定をご参照ください。
SNIを有効化し、ドメイン名のディメンションで証明書を設定します。このリスナー下でドメイン名ごとに異なる証明書を設定できます。詳細については、ドメイン名ディメンションでの証明書設定をご参照ください。

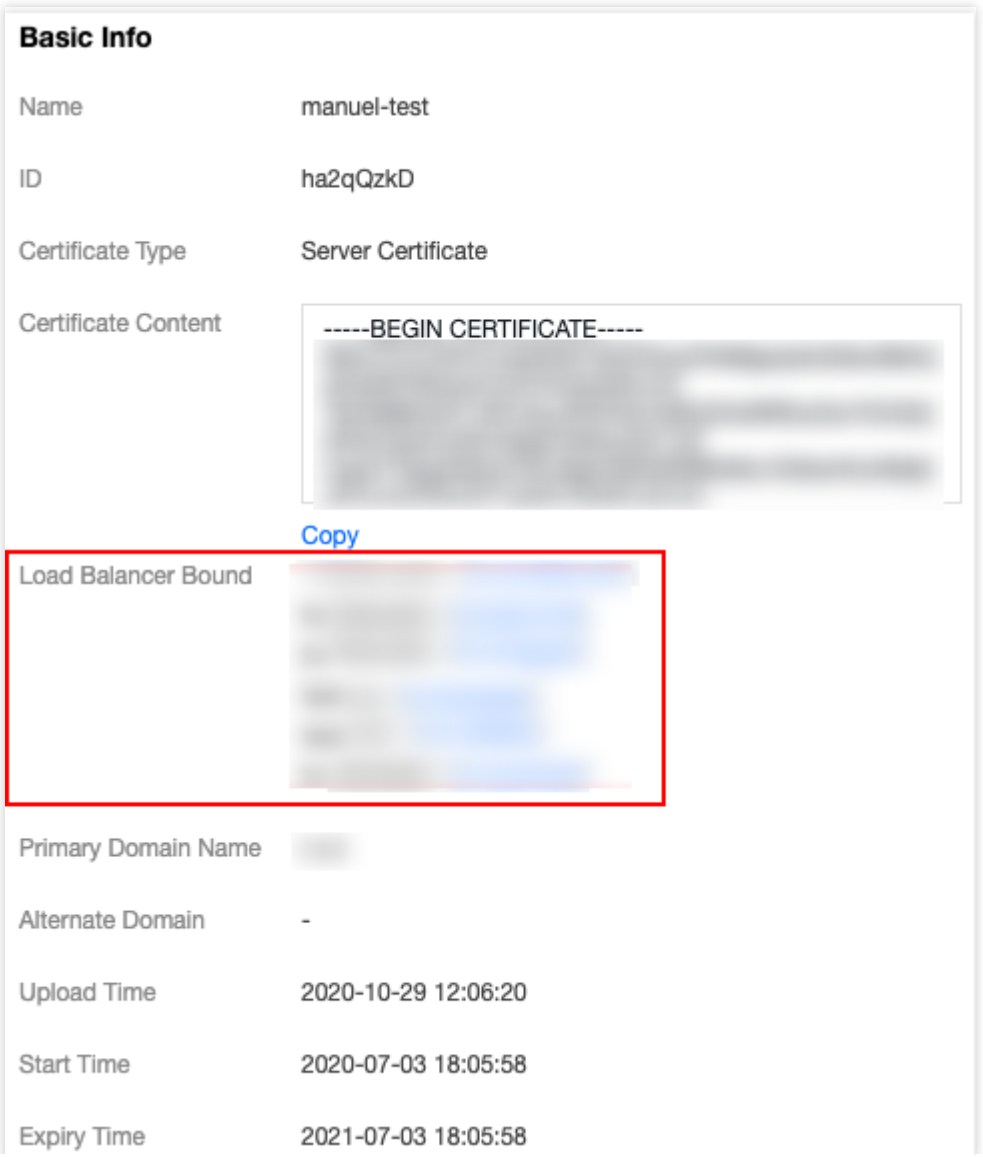
証明書の更新

証明書が期限切れとなることでサービスに影響することがないよう、証明書は有効期限までに更新してください。
説明：
証明書を更新すると、すぐに有効化され、システムは古い証明書を削除せず、新しい証明書を生成します。この証明書を使用するすべてのCLBインスタンスで、証明書が自動的に更新されます。
1. CLBコンソールにログインします。
2. 左側ナビゲーションバーで**証明書管理**をクリックします。
3. **証明書管理**ページの証明書リストで、目的の証明書の右側にある**操作**列の**更新**をクリックします。
4. ポップアップした「証明書の新規作成」ダイアログボックスで、新しい証明書の証明書内容およびキー内容を入力し、**送信**をクリックします。



証明書に関連付けられたCLBの確認

1. CLBコンソールにログインします。
2. 左側ナビゲーションバーで**証明書管理**をクリックします。
3. **証明書管理**ページの証明書リストで、目的の証明書のIDをクリックします。
4. **基本情報**ページで、証明書に関連付けられているCLBインスタンスを確認します。



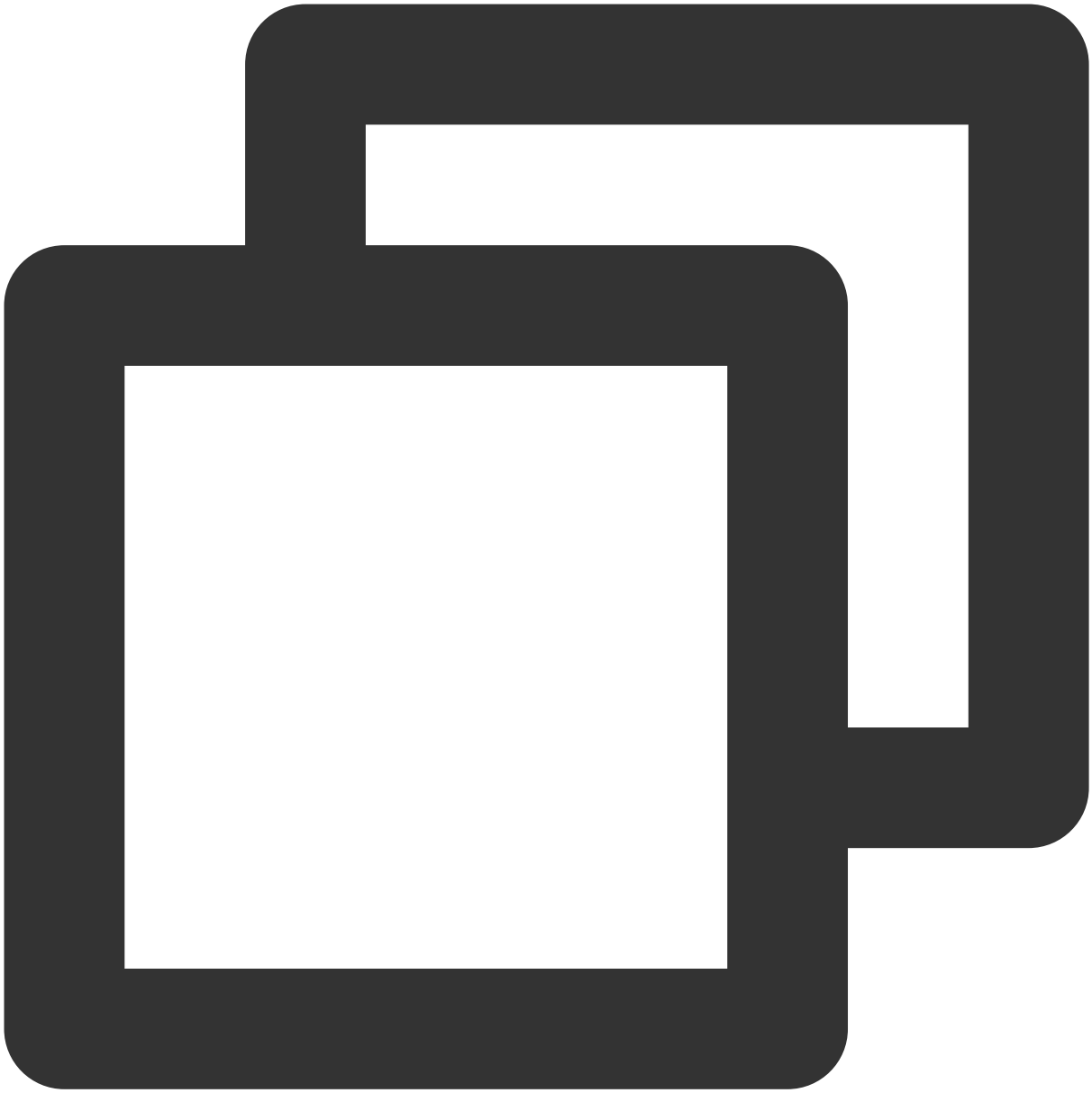
証明書の要件および証明書形式の変換

最終更新日：：2024-01-04 18:36:26

ここではSSL証明書の要件および証明書形式の変換についてご説明します。

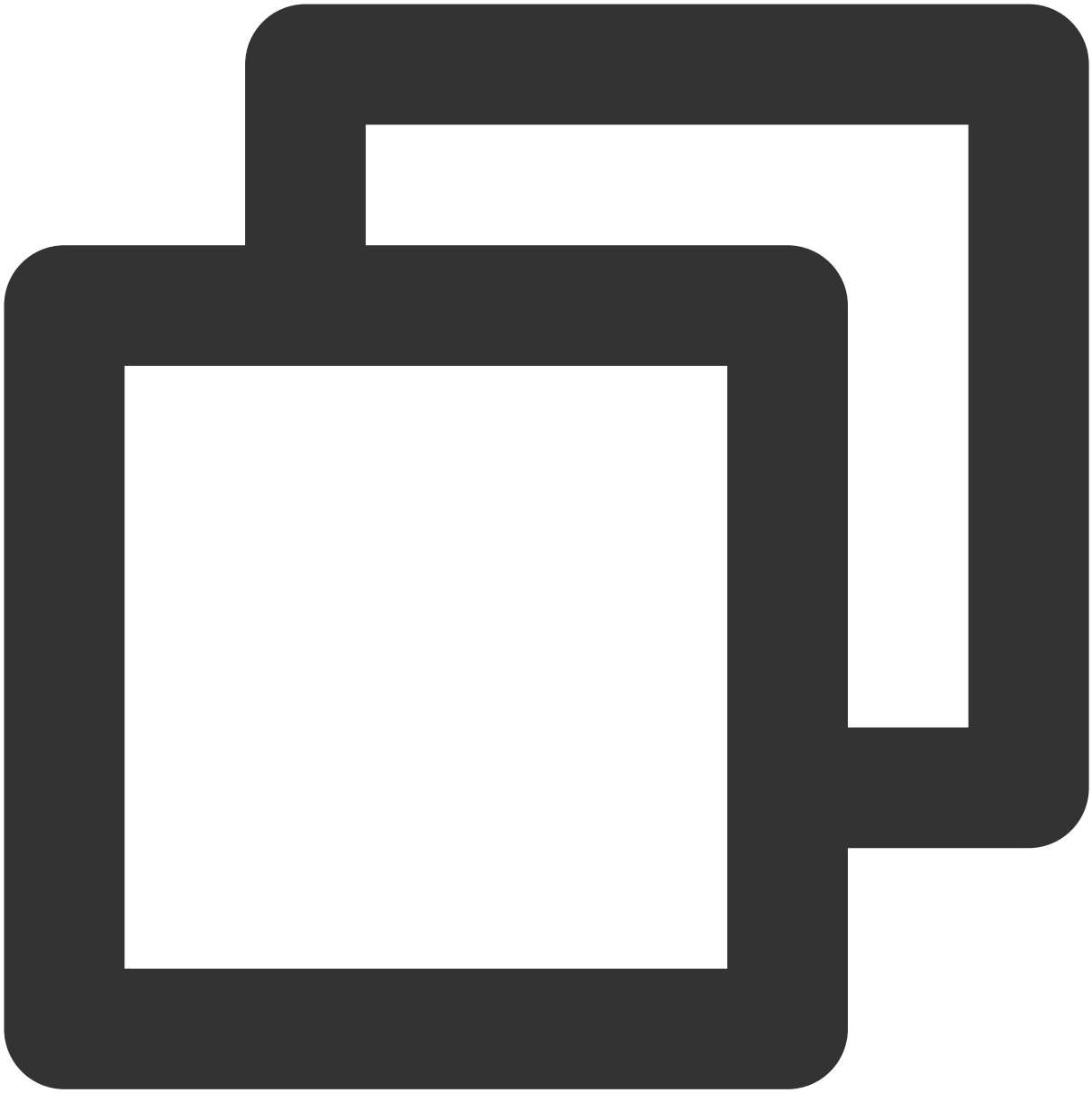
一般的な証明書申請のフロー

1. OpenSSLツールを使用して、ローカルで秘密鍵ファイルを生成します。その中の `privateKey.pem` が秘密鍵ファイルですので、適切に保管してください。



```
openssl genrsa -out privateKey.pem 2048
```

2. OpenSSLツールを使用して、証明書リクエストファイルを生成します。その中の `server.csr` が証明書リクエストファイルです。証明書の申請に使用できます。



```
openssl req -new -key privateKey.pem -out server.csr
```

3. 証明書リクエストファイルの内容を取得し、CAなどの機関のサイトに送信して証明書の申請を行います。

証明書形式の要件

ユーザーが申請する必要がある証明書は、Linux環境のPEM形式の証明書です。CLBは他の形式の証明書をサポートしていません。その他の形式の証明書の場合は、下記の証明書のPEM形式への変換の説明の内容をご参照くだ

さい。
root CA機関によって発行された証明書の場合は、取得した証明書が唯一のものであり、追加の証明書は必要ありません。これを設定したサイトは、ブラウザなどのアクセスデバイスによって信頼できるとみなされます。
中間CA機関によって発行された証明書の場合は、取得した証明書ファイルに複数の証明書が含まれるため、サーバー証明書と中間証明書を手動で合わせてアップロードする必要があります。
証明書に証明書チェーンがある場合は、証明書チェーンの内容をPEM形式の内容に変換し、証明書の内容と合わせてアップロードしてください。
結合ルール：サーバー証明書を最初に、中間証明書を2番目に配置し、間に空白行を空けずに結合します。
説明：
通常、機関が証明書を発行する際には、それに対応する説明が添付されていますので、よくお読みください。
証明書形式および証明書チェーン形式の例
証明書形式および証明書チェーン形式の例は次のとおりです。形式が正しいことを確認してからアップロードしてください。
1. root CA機関が発行した証明書：証明書形式はLinux環境のPEM形式です。サンプルは次のようになります。

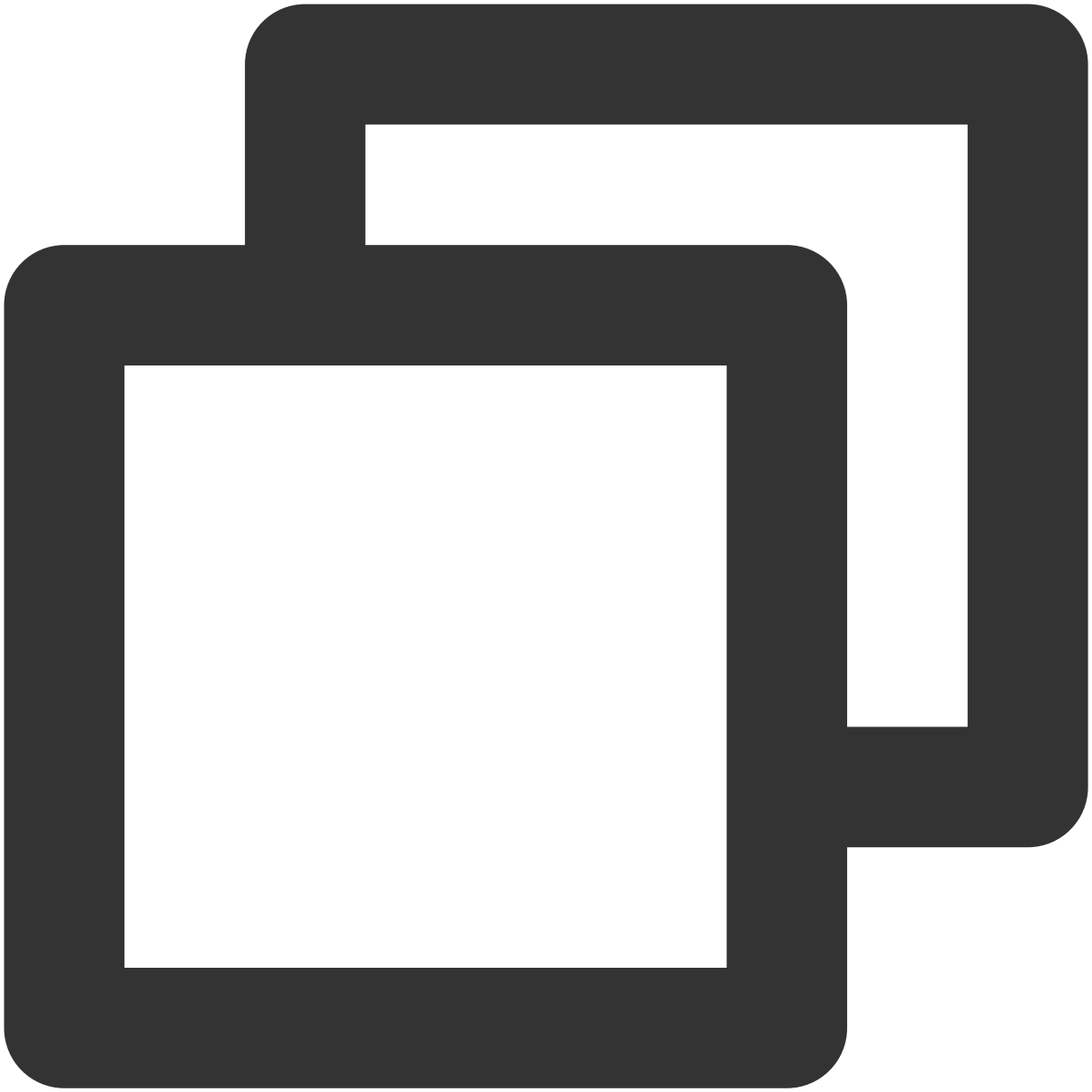
証明書ルールは次のとおりです。
[——-BEGIN CERTIFICATE——-、——-END CERTIFICATE——-] を先頭と末尾にします。これらの内容を合わせてアップロードしてください。
1行の文字数は64文字とし、最後の1行は64文字以内とします。
2. 中間機関が発行した証明書チェーン：
——-BEGIN CERTIFICATE——-

——-END CERTIFICATE——-
——-BEGIN CERTIFICATE——-
——-END CERTIFICATE——-
——-BEGIN CERTIFICATE——-
——-END CERTIFICATE——-
証明書チェーンルールは次のとおりです。
証明書の間には空白行があってはなりません。
各証明書が上記の証明書形式の要件を遵守していることとします。

RSA秘密鍵形式の要件

サンプルは次のようになります。

RSA秘密鍵にはすべての秘密鍵（RSAおよびDSA）、公開鍵（RSAおよびDSA）および(x509)証明書を含めることができます。これはBase64でエンコードされたDER形式のデータを使用して保存し、ASCIIヘッダーで囲むため、システム間のテキスト形式での伝送に適しています。
RSA秘密鍵のルール：
[——-BEGIN RSA PRIVATE KEY——-、——-END RSA PRIVATE KEY——-]を先頭と末尾にします。これらの内容を合わせてアップロードしてください。
1行の文字数は64文字とし、最後の1行は64文字に満たなくても構いません。
上記の方法で[——-BEGIN PRIVATE KEY——-、——-END PRIVATE KEY——-]形式の使用可能な秘密鍵を生成していない場合は、次の方法で使用可能な秘密鍵に変換することができます。



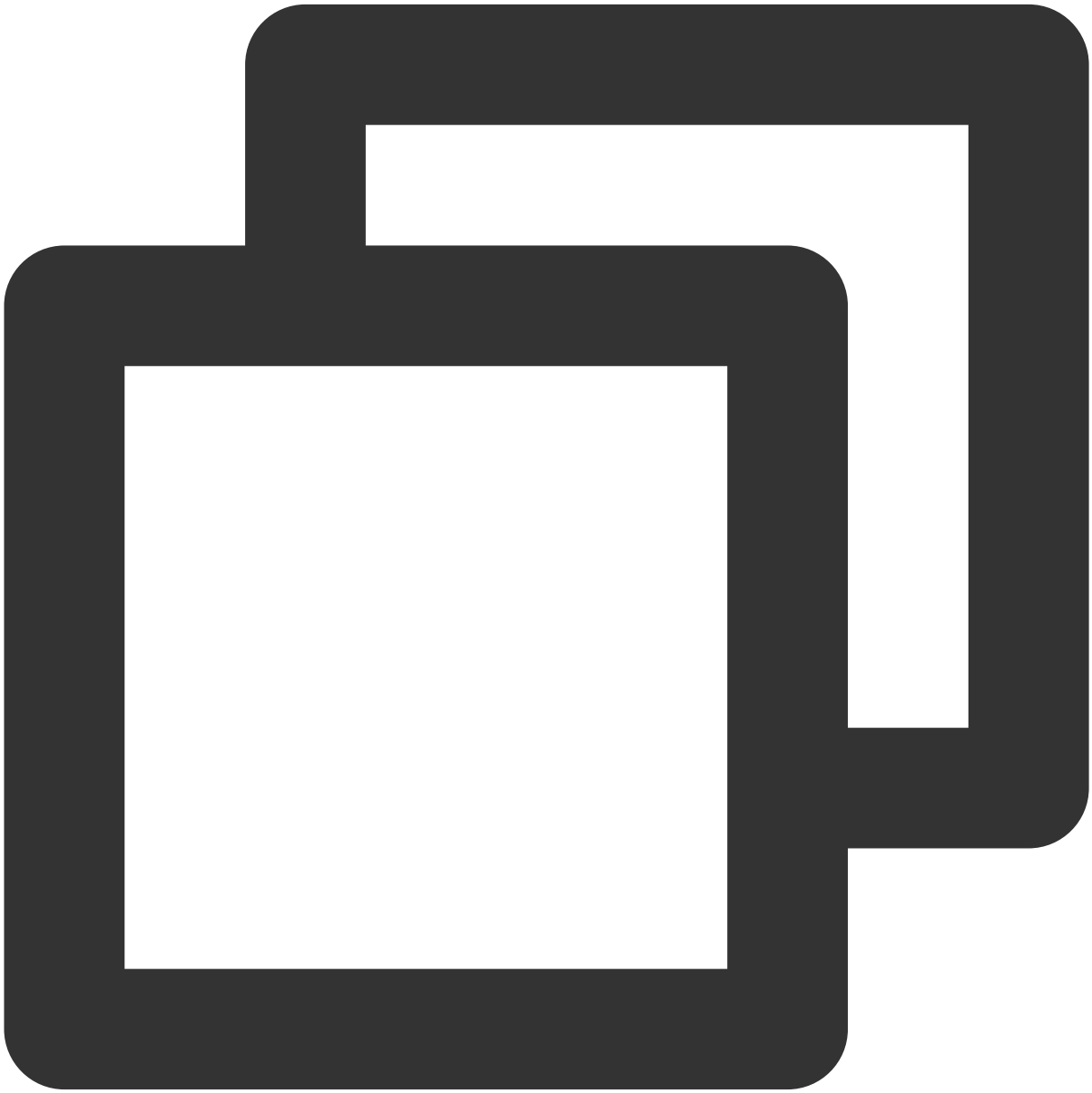
```
openssl rsa -in old_server_key.pem -out new_server_key.pem
```

その後、new_server_key.pemの内容を証明書と共にアップロードします。

証明書のPEM形式への変換の説明

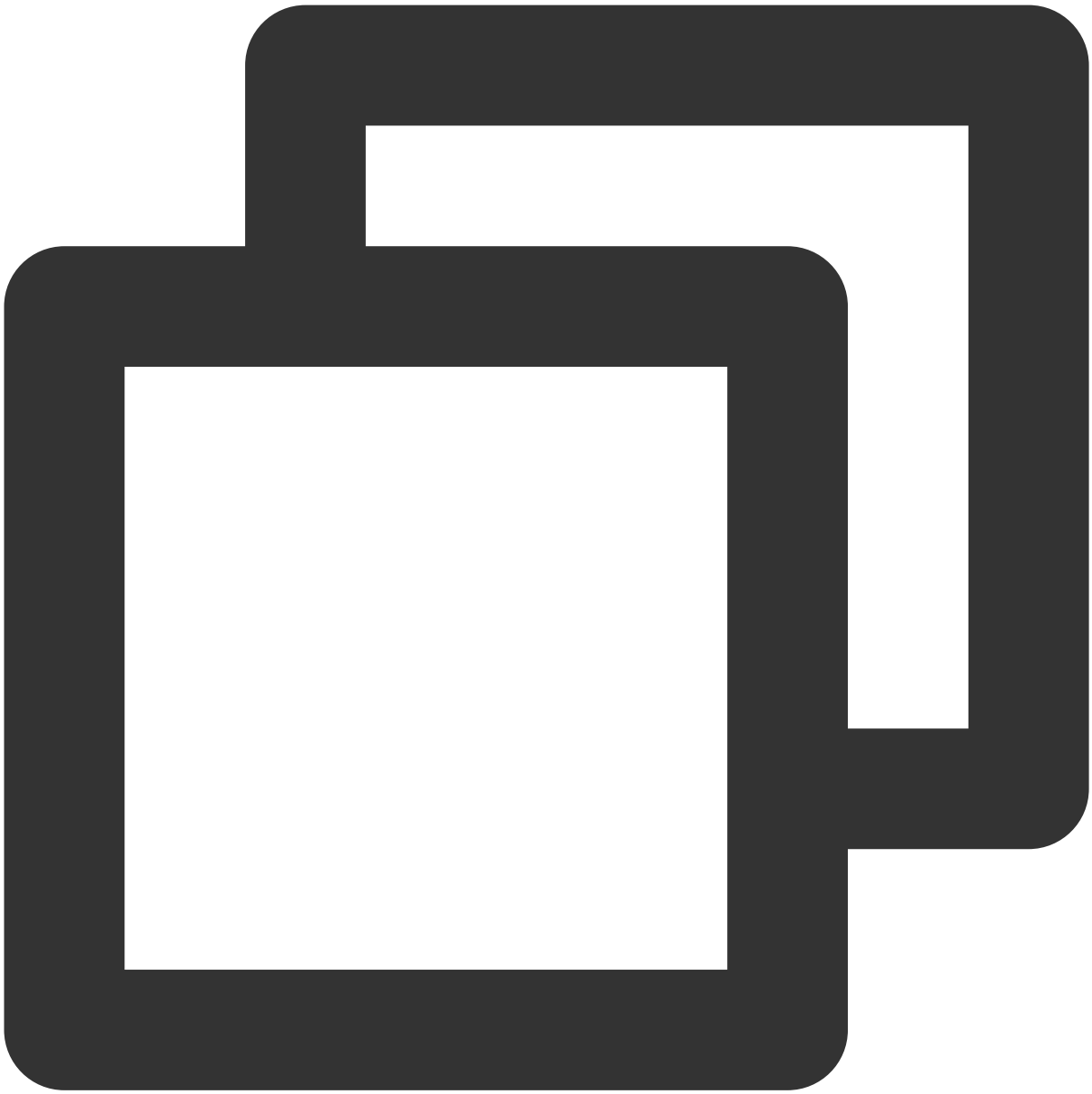
現在CLBはPEM形式の証明書のみサポートしており、他の形式の証明書はPEM形式に変換してからでなければCLBにアップロードできません。変換はopensslツールによって行うことをお勧めします。証明書の形式をPEM形

式に変換する、一般的ないくつかの方法を次に挙げます。
DER を PEM に変換
P7B を PEM に変換
PFX を PEM に変換
CER/CRT を PEM に変換
DER形式は一般的にJavaプラットフォームで用いられます。
証明書の変換：



```
openssl x509 -inform der -in certificate.cer -out certificate.pem
```

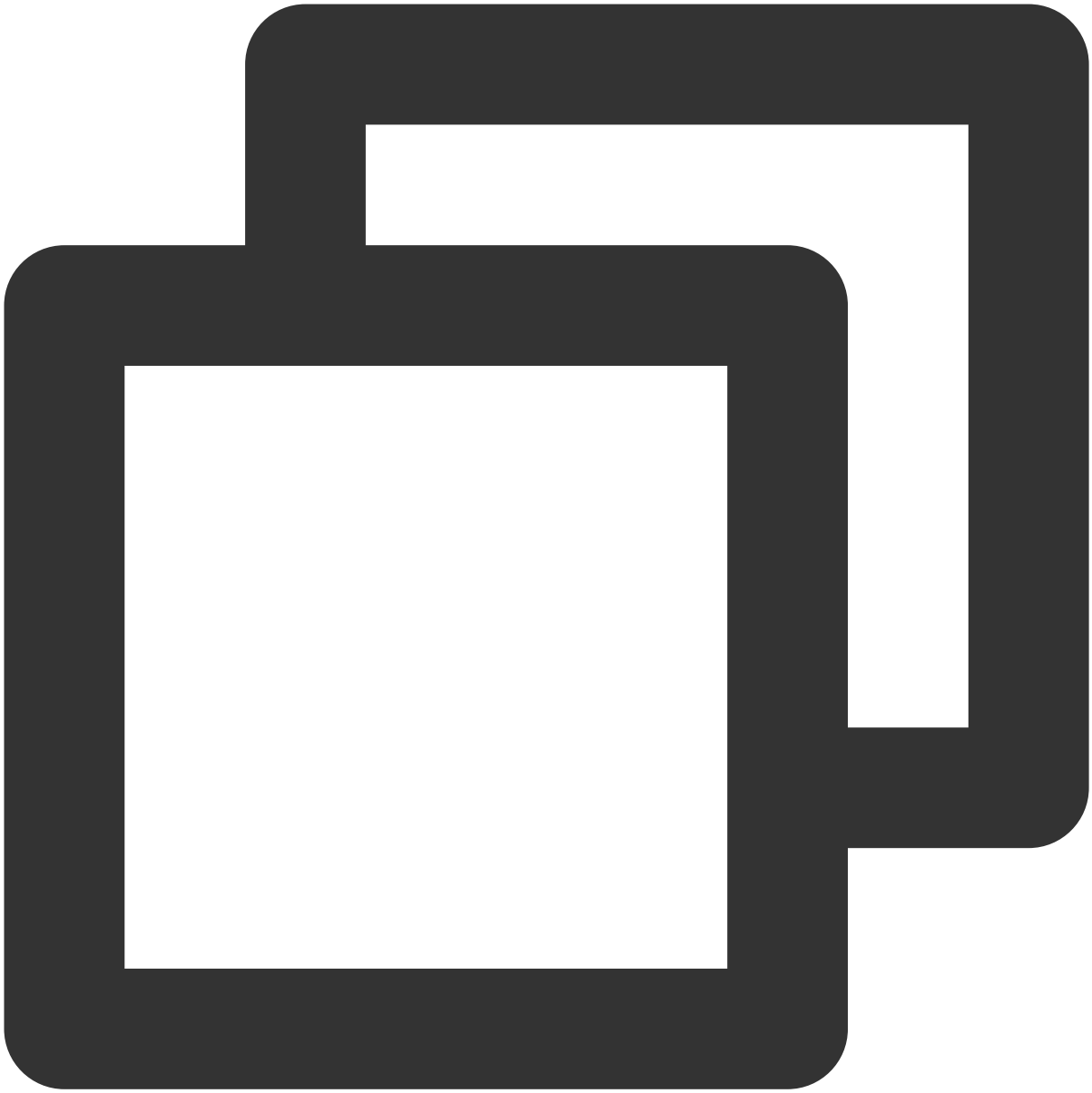
秘密鍵の変換：



```
openssl rsa -inform DER -outform PEM -in privatekey.der -out privatekey.pem
```

P7B形式は一般的にWindows ServerおよびTomcatで用いられます。

証明書の変換：



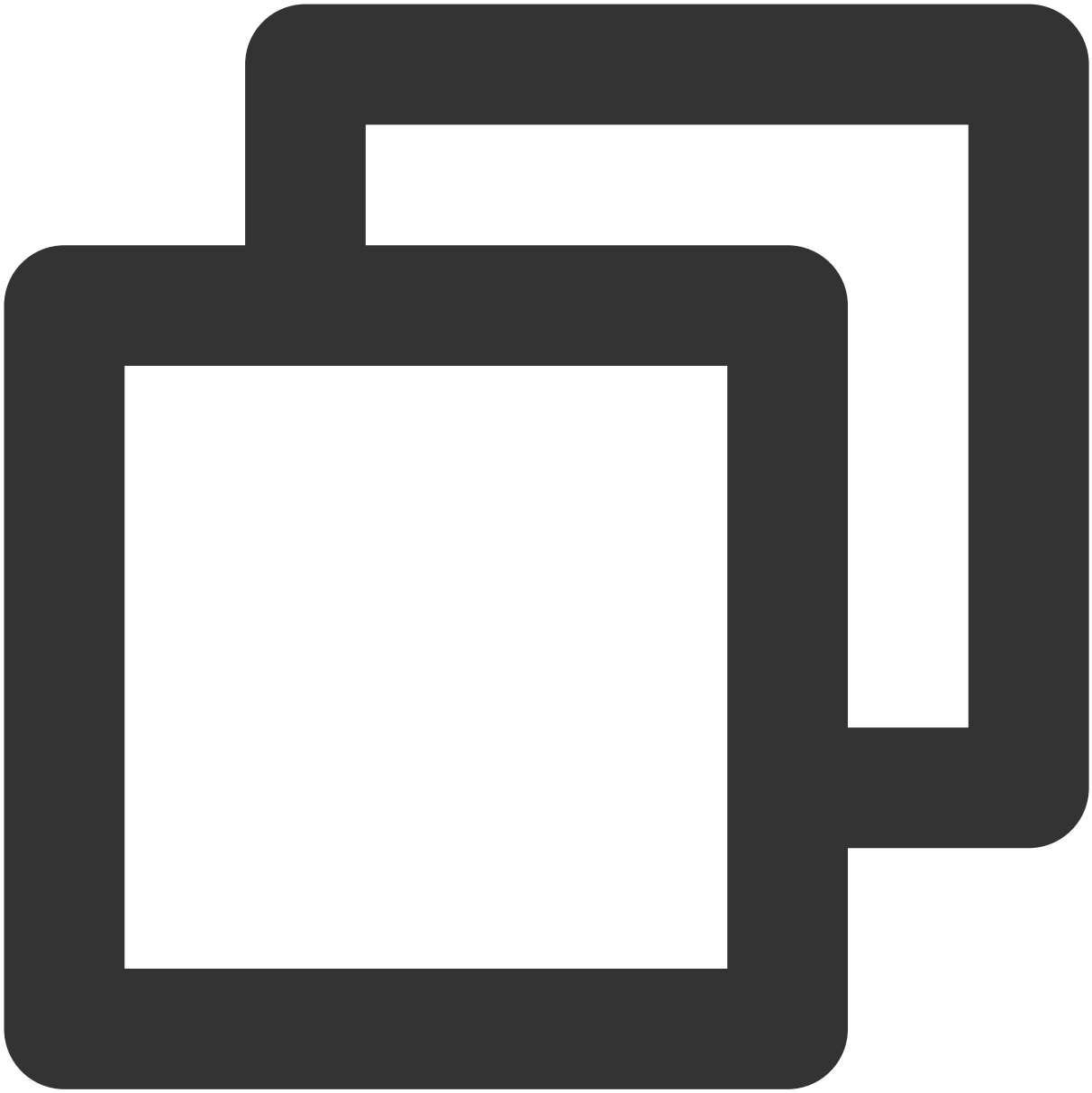
```
openssl pkcs7 -print_certs -in incertificat.p7b -out outcertificate.cer
```

outcertificat.cerの中の[——-BEGIN CERTIFICATE——-、——-END CERTIFICATE——-]の内容を取得し、証明書としてアップロードします。

秘密鍵の変換：秘密鍵は通常、IISサーバーからエクスポートできます。

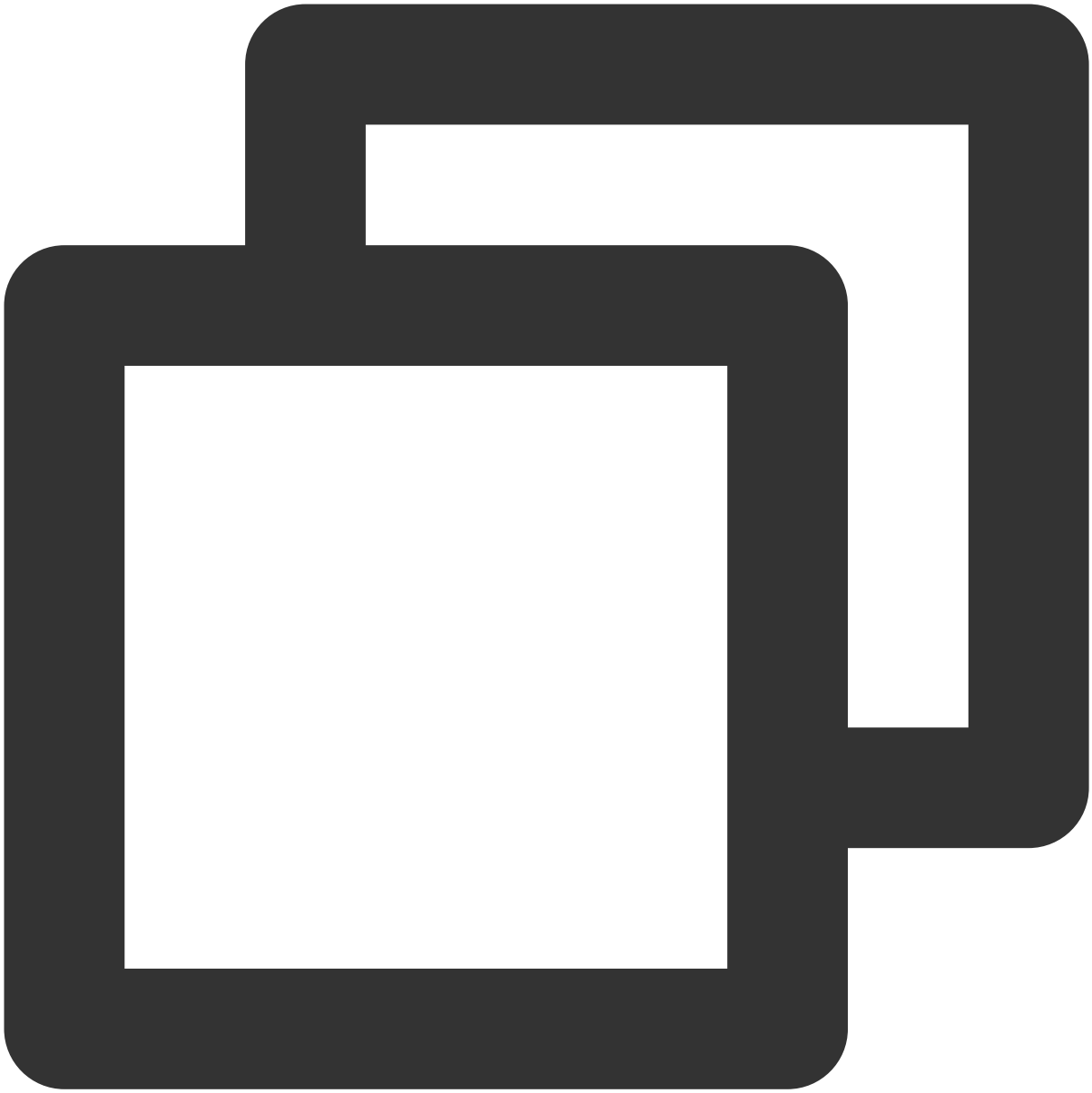
PFX形式は一般的にWindows Serverで用いられます。

証明書の変換：



```
openssl pkcs12 -in certname.pfx -nokeys -out cert.pem
```

秘密鍵の変換：



```
openssl pkcs12 -in certname.pfx -nocerts -out key.pem -nodes
```

CER/CRT形式の証明書については、証明書ファイルの拡張子を直接変更する方法で変換することができます。例えば、証明書ファイル「servertest.crt」は「servertest.pem」に直接リネームできます。

SSL単方向認証および双方向認証の説明

最終更新日：：2024-01-04 18:36:26

Secure Sockets Layer（SSL）とは、ネットワーク通信に安全性およびデータ完全性を提供するためのセキュリティプロトコルの一種です。ここでは主にSSLの単方向認証および双方向認証についてご説明します。

説明：

CLBはTCP SSLリスナーまたはHTTPSリスナーを作成する際、SSL の解析メソッドとして単方向認証か双方向認証かを選択することができます。詳細については、TCP SSLリスナーの設定、HTTPSリスナーの設定をご参照ください。

SSL単方向認証と双方向認証の違い

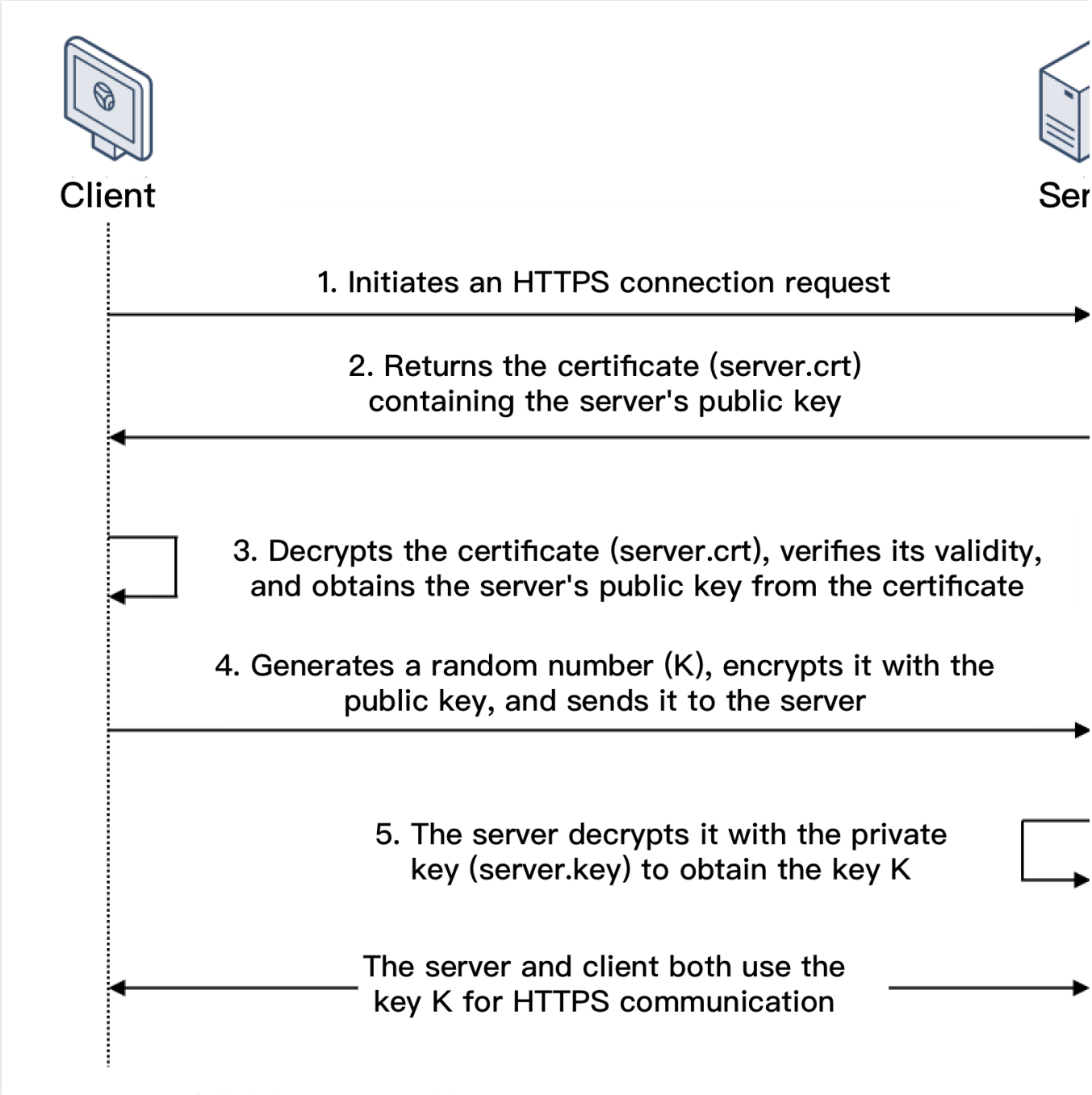
SSL単方向認証はクライアントが証明書を所有する必要がなく、サーバーのみ証明書が必要です。SSL双方向認証ではクライアントとサーバーの両方が証明書を所有する必要があります。

SSL単方向認証はSSL双方向認証の認証プロセスと異なり、サーバーでクライアント証明書の検証と暗号化方式のネゴシエーションを行う必要がなく、サーバーからクライアントへも暗号化されていない暗号化方式が送信されます（SSL認証プロセスの安全性に影響はありません）。

一般的に、Webアプリケーションはユーザー数が非常に多く、通信層でユーザーのID認証を行う必要がないため、SSL単方向認証の設定で十分です。ただし、一部の金融業界ユーザーのアプリケーションアクセスでは、クライアントのID認証が要求される可能性があり、この場合はSSL双方向認証が必要です。

SSL単方向認証

SSL単方向認証ではサーバーのIDのみ検証する必要があり、クライアントのIDを検証する必要はありません。SSL単方向認証のフローは次の図のとおりです。

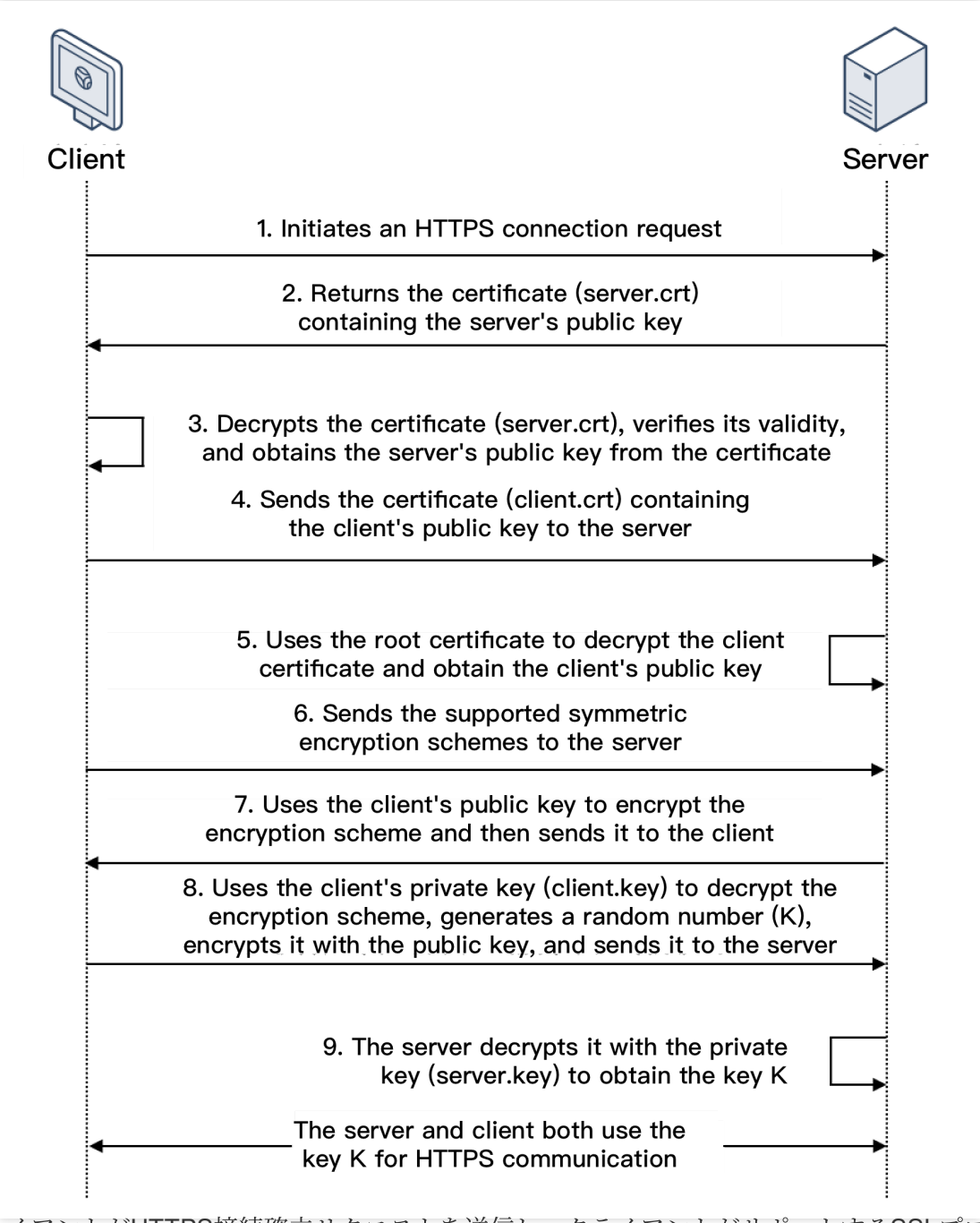


1. クライアントがHTTPS接続確立リクエストを送信し、クライアントがサポートするSSLプロトコルバージョン番号、暗号化アルゴリズムの種類、生成する乱数などの情報をサーバーに送信します。
2. サーバーはSSLプロトコルバージョン番号、暗号化アルゴリズムの種類、生成する乱数などの情報、サーバーの証明書（server.crt）をクライアントに返します。
3. クライアントは証明書（server.crt）の有効性を検証し、この証明書からサーバーの公開鍵を取得します。
証明書が期限切れになっていないかを確認します。
証明書が取り消されていないかを確認します。
証明書が信頼できるかどうかを確認します。
受信した証明書内のドメイン名とリクエストのドメイン名が一致しているかどうかを確認します。

4. 証明書が検証に合格すると、クライアントは乱数（キーK）を生成し、通信のプロセスで共通鍵暗号化のキーとして用います。さらにサーバー証明書の公開鍵を使用して暗号化した後、サーバーに送信します。
5. サーバーはクライアントから送信された暗号化情報を受信した後、秘密鍵（server.key）を使用して復号し、共通暗号化鍵（キーK）を取得します。
それ以降のセッションでは、クライアントとサーバーはその共通暗号化鍵（キーK）を使用して通信を行うことで、通信プロセスにおける情報のセキュリティを保証します。

SSL双方向認証

SSL双方向認証ではクライアントとサーバーのIDを検証する必要があります。SSL双方向認証のフローは次の図のとおりです。



1. クライアントがHTTPS接続確立リクエストを送信し、クライアントがサポートするSSLプロトコルバージョン番号、暗号化アルゴリズムの種類、生成する乱数などの情報をサーバーに送信します。
2. サーバーはSSLプロトコルバージョン番号、暗号化アルゴリズムの種類、生成する乱数などの情報、サーバーの証明書（server.crt）をクライアントに返します。
3. クライアントは証明書（server.crt）の有効性を検証し、この証明書からサーバーの公開鍵を取得します。
証明書が期限切れになっていないかを確認します。
証明書が取り消されていないかを確認します。
証明書が信頼できるかどうかを確認します。
受信した証明書内のドメイン名とリクエストのドメイン名が一致しているかどうかを確認します。

4. サーバーがクライアントにクライアントの証明書（client.crt）を送信するよう要求し、クライアントは自身の証明書をサーバーに送信します。
5. サーバーはクライアントの証明書（client.crt）を検証し、検証に合格すると、サーバーはルート証明書（root.crt）を使用してクライアント証明書を復号し、クライアントの公開鍵を取得します。
6. クライアントはサーバーに、自身のサポートする共通鍵暗号化方式を送信します。
7. サーバーはクライアントから送信された共通鍵暗号化方式の中から、暗号化の程度が最も高い暗号化方式を選択し、クライアントの公開鍵を使用して暗号化した後、クライアントに返します。
8. クライアントはクライアントの秘密鍵（client.key）を使用して暗号化方式を復号し、乱数（キーK）を生成し、通信のプロセスで共通鍵暗号化のキーとして用います。その後、サーバー証明書の公開鍵を使用して暗号化した後、再びサーバーに送信します。
9. サーバーはクライアントから送信された暗号化情報を受信した後、サーバーの秘密鍵（server.key）を使用して復号し、共通暗号化鍵（キーK）を取得します。
それ以降のセッションでは、クライアントとサーバーはその共通暗号化鍵（キーK）を使用して通信を行うことで、通信プロセスにおける情報のセキュリティを保証します。

関連ドキュメント

証明書の要件および証明書形式の変換

ログ管理
アクセスログの概要

最終更新日：：2024-01-04 18:36:26

CLBのアクセスログは各クライアントリクエストの詳細情報を収集し、リクエスト時間、リクエストパス、クライアントIPおよびポート、戻りコード、応答時間などの情報をログに記録します。アクセスログは、クライアントリクエストの把握、トラブルシューティングの補助、ユーザー行動の分析と整理などに役立ちます。

説明：

アクセスログの設定をサポートしているのはレイヤー7 CLBのみであり、レイヤー4CLBではサポートしていません。

現在、アクセスログの設定は一部のリージョンでのみサポートしています。詳細については、CLSの アベイラビリティリージョン をご参照ください。

ストレージ方式

CLBのアクセスログはCloud Log Service（CLS）をサポートしています。CLSはワンストップ式のログサービスプラットフォームであり、ログの収集、ログの保存や、ログの検索分析、リアルタイム消費、ログ配信などのさまざまなサービスを提供し、ログによるユーザーの業務運営、セキュリティモニタリング、ログ審査、ログ分析などの問題解決を支援します。

機能の特徴	アクセスログのCLSへの保存設定
ログ取得の時間粒度	分レベル
オンライン検索	サポートあり
検索構文	全文検索、キー値検索、あいまいキーワード検索などがあります。詳細については、検索ルールをご参照ください。
サポートリージョン	リージョンサポートの詳細については、CLSのアベイラビリティリージョンをご参照ください。
サポートタイプ	パブリックネットワーク/プライベートネットワークCLBをサポート
アップストリームリンクおよびダウンストリームリンク	CLSはログのCOSへの配信をサポートしており、CKafkaを使用してログを消費できます。
ログの保存	Tencent Cloudはデフォルトではアクセスログの保存をコミットしていません。業務上必要な場合は、アクセスログのCLSへの保存をご自身で設定してください。

関連操作

アクセスログのCLSへの保存設定

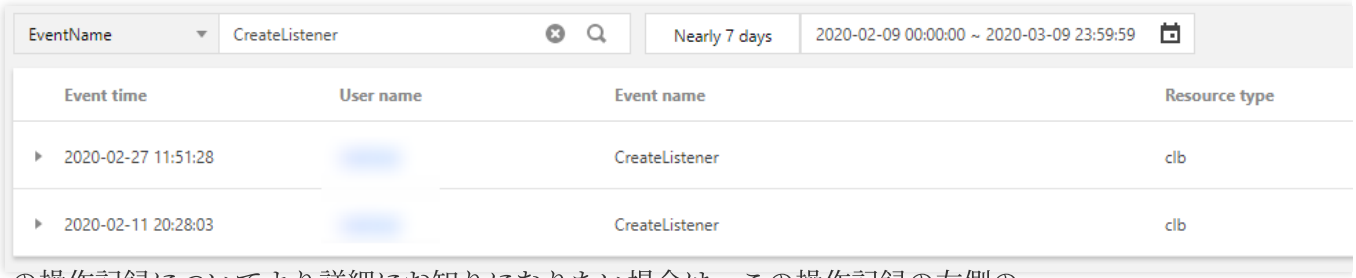
操作ログの確認

最終更新日：：2024-01-04 18:36:26

CLBの操作記録は、CloudAuditコンソールで照会、ダウンロードすることができます。
CloudAuditはTencent Cloudアカウントに対する規制、コンプライアンスチェック、操作の審査およびリスク審査のサポートを行うサービスです。CloudAuditはTencent Cloudアカウントの活動に伴うイベント履歴を提供します。これらの活動には、Tencent Cloudの管理コンソール、APIサービス、コマンドラインツール、その他のTencent Cloudサービスによって実行される操作が含まれます。このイベント履歴によって、安全性分析、リソース変更の追跡およびトラブルシューティングの作業を簡略化することができます。

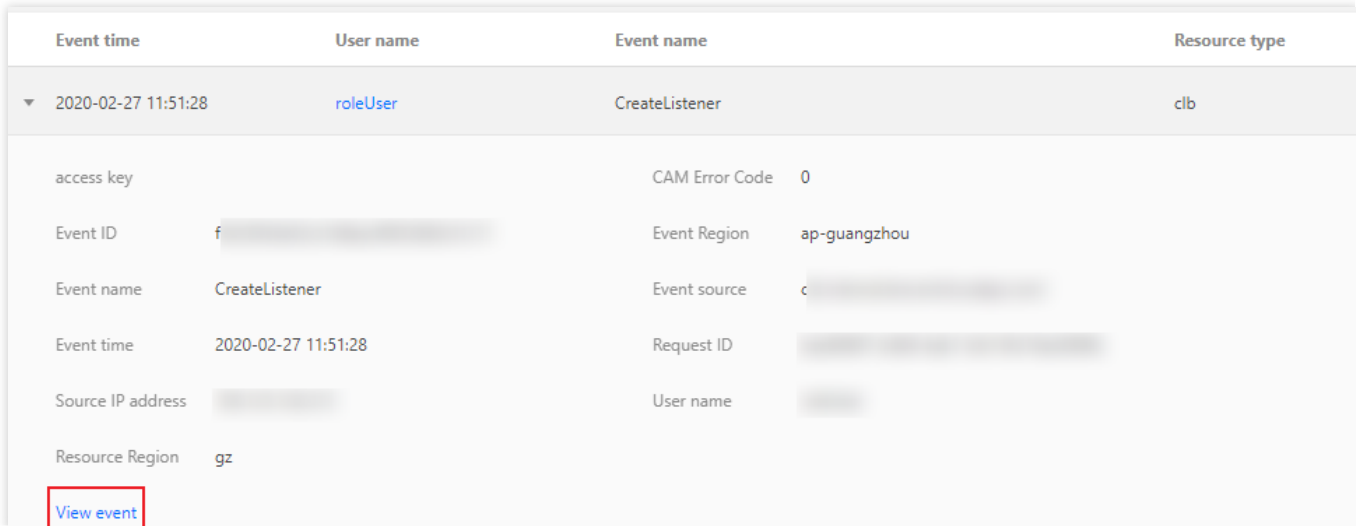
操作手順

1. CloudAuditコンソールにログインします。
2. 左側ナビゲーションで**操作の記録**をクリックし、「操作の記録」ページに進みます。またはCLBコンソールにログインし、ページ右上隅の**CloudAudit**を選択すると、すぐに操作記録ページに進むことができます。
3. 操作記録ページで、ユーザー名、リソースタイプ、リソース名、イベントソース、イベントIDなどに基づいて操作の記録を照会できます。デフォルトの状態では一部のデータだけが表示されており、ページ下部で**クリックしてさらにロード**をクリックすると、その他の記録を取得することができます。



4. 単一の操作記録についてより詳細にお知りになりたい場合は、この操作記録の左側の

をクリックすると、アクセスキー、エラーコード、イベントIDなどの操作記録の詳細を確認できます。また、**イベントの表示**をクリックすると、イベントの関連情報を知ることができます。



アクセスログの設定

最終更新日：：2024-01-04 18:36:26

CLBはレイヤー7（HTTP/HTTPS）アクセスログ（Access Log）の設定をサポートしています。アクセスログは、クライアントリクエストの把握、トラブルシューティングの補助、ユーザー行動の分析と整理などに役立ちます。現在アクセスログはCLSへの保存をサポートしており、分単位でのログレポート、オンラインマルチルール検索をサポートしています。

CLBのアクセスログは主にトラブルシューティングに用いられ、業務上の問題を迅速に特定する上で役立ちます。アクセスログの機能には、ログレポート、ログのストレージと照会があります。

ログレポートはベストエフォートサービス（Best-Effort Service）です。業務の転送を優先的に保障した後にログレポートを保障します。

ログのストレージと照会では、現在使用中のストレージサービスに基づいてサービス品質保証（SLA）を提供します。

説明：

現在CLBはレイヤー7プロトコル（HTTP/HTTPS）のみ、アクセスログをCLSに保存する設定をサポートしています。レイヤー4プロトコル（TCP/UDP/TCP SSL）ではアクセスログをCLSに保存する設定をサポートしていません。

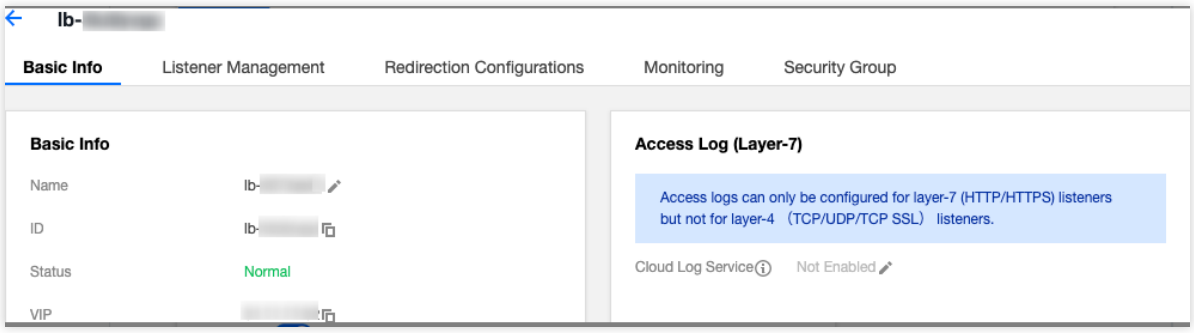
CLBによるアクセスログのCLSへの保存設定機能は無料です。ユーザーにはCLSの料金のみがかかります。

この機能は現在一部のリージョンでのみサポートされています。実際には、コンソールのサポートリージョンに準じます。

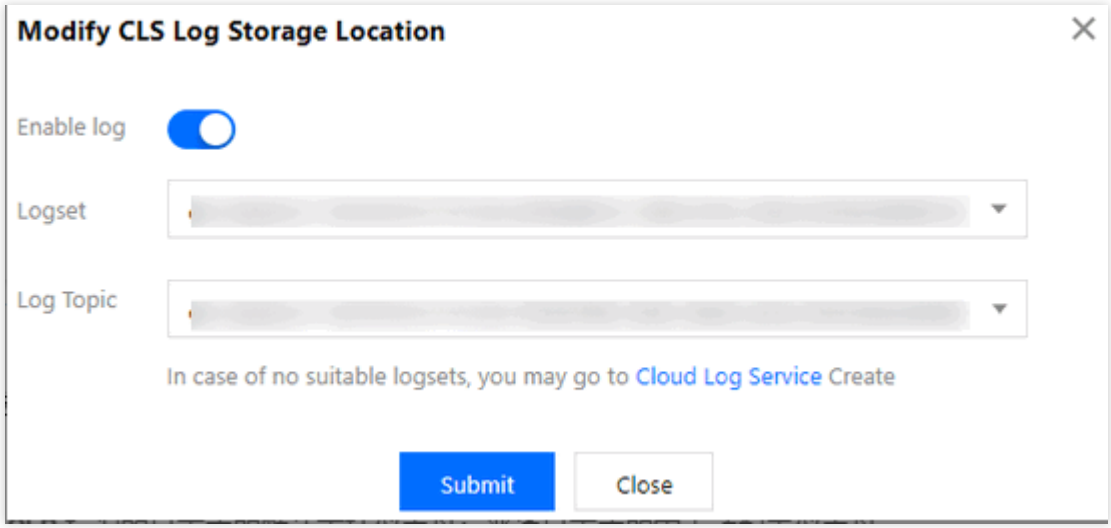
方法1：単一のインスタンスにアクセスログを設定する

ステップ1：アクセスログのCLSへの保存の有効化

1. CLBコンソールにログインし、左側ナビゲーションバーの**インスタンス管理**をクリックします。
2. **インスタンス管理**ページで、目的のCLB IDをクリックします。
3. **基本情報**ページの「アクセスログ（レイヤー7）」モジュールで、鉛筆のアイコンをクリックします。



4. ポップアップした**CLSログ保存場所の変更**ダイアログボックスで**ログの有効化**を開き、アクセスログを保存するログセットおよびログトピックを選択し、**送信**をクリックします。ログセットまたはログトピックを作成していない場合は、関連リソースの新規作成をクリックしてから、具体的な保存場所を選択してください。



説明：

clb_logsetログセット下の、CLBの表示があるログトピックを選択することをお勧めします。CLBの表示があるログトピックと一般のログトピックとの違いは次の点にあります。

CLBの表示があるログトピックでは、インデックスはデフォルトで自動作成されます。一般のログトピックでは手動でインデックスを作成する必要があり、作成しなければ検索がサポートされません。

CLBの表示があるログトピックはデフォルトでダッシュボードをサポートします。一般のログトピックでは手動でダッシュボードを設定する必要があります。

5. 設定完了後にログセットまたはログトピックをクリックすると、CLSコンソールの検索分析ページにリダイレクトされます。

6. （オプション）アクセスログを無効化したい場合は、再度鉛筆のアイコンをクリックし、ポップアップした**CLSログ保存場所の変更**ダイアログボックスで無効化を行い、送信するだけで可能です。

ステップ2：ログトピックのインデックスの設定

説明：

単一のインスタンスに設定するアクセスログのログトピックには必ずインデックスを設定しなければならず、そうしなければログが検索できなくなります。

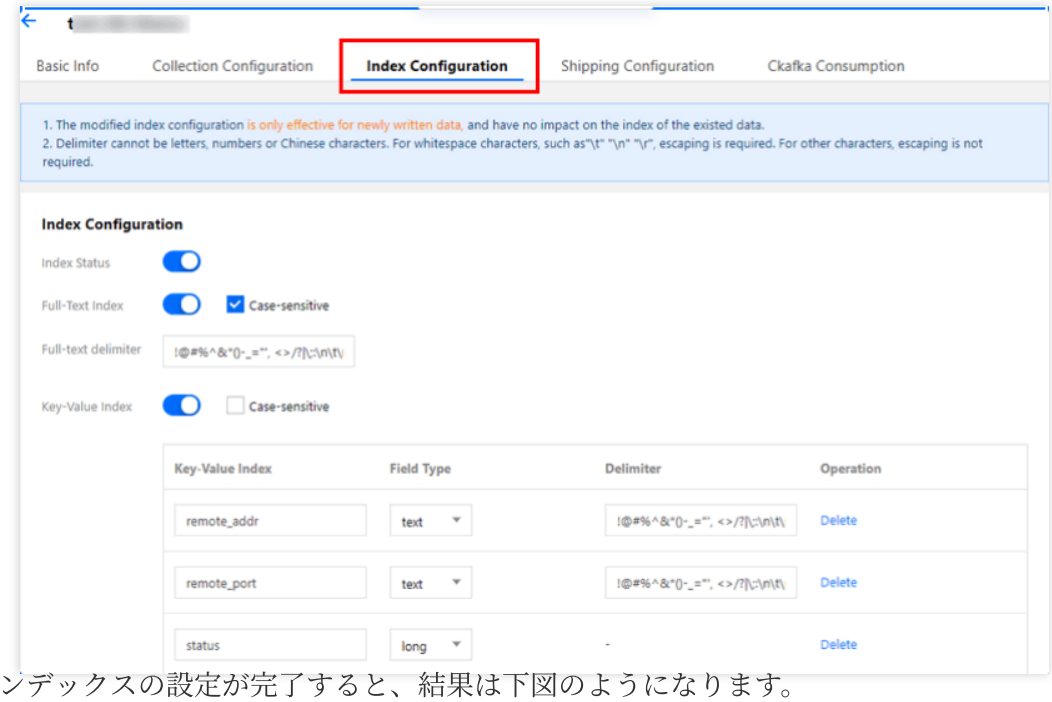
設定を推奨するインデックスは次のとおりです。

キー値インデックス	フィールドタイプ	区切り文字
server_addr	text	区切り文字は設定不要
server_name	text	区切り文字は設定不要
http_host	text	区切り文字は設定不要

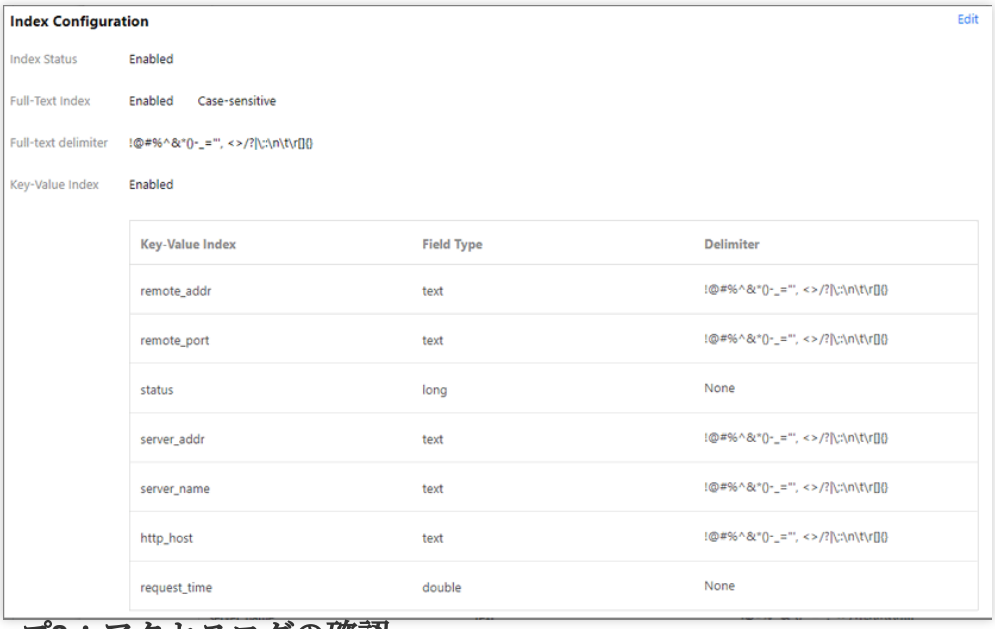
status	long	-
vip_vpcid	long	-

具体的な操作は次のとおりです。

1. CLSコンソールにログインし、左側のナビゲーションバーで**ログトピック**をクリックします。
2. **ログトピック**ページで、目的のログトピックIDをクリックします。
3. ログトピック詳細ページで**インデックスの設定**タブをクリックし、右上隅の**編集**をクリックするとインデックスを追加できます。インデックスフィールドの設定説明については、インデックスの有効化をご参照ください。

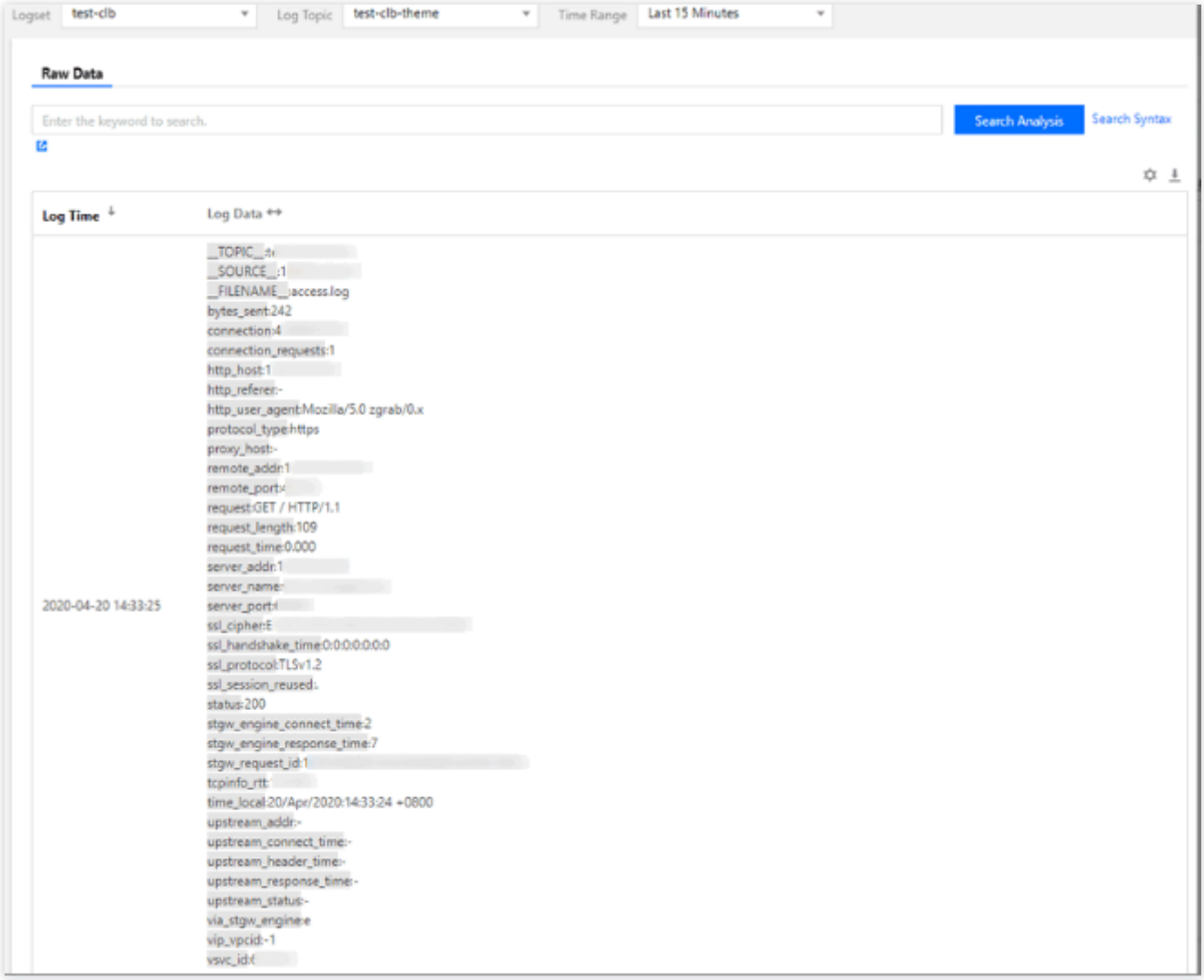


4. インデックスの設定が完了すると、結果は下図のようになります。



ステップ3：アクセスログの確認

1. CLSコンソールにログインし、左側のナビゲーションバーの**検索分析**をクリックします。
2. **検索分析**ページで、ログセット、ログトピックおよび時間範囲を選択し、**検索分析**をクリックすると、CLBがCLSに送信したアクセスログを検索できます。検索構文の詳細については、構文とルールをご参照ください。



方法2：アクセスログの一括設定

ステップ1：ログセットとログトピックの作成

アクセスログをCLSに保存するよう設定したい場合は、先にログセットとログトピックを作成する必要があります。

ログセットとログトピックを作成済みの場合は、スキップしてステップ2から操作を開始することができます。

1. CLBコンソールにログインし、左側ナビゲーションバーの**アクセスログ**をクリックします。
2. **アクセスログ**ページの左上隅で所属リージョンを選択し、**ログセット情報**のエリアで**ログセットの作成**をクリックします。
3. ポップアップした**ログセットの作成**ダイアログボックスで保存期間を設定し、**保存**をクリックします。

説明：

各リージョンにつき、作成できるログセットは1つのみです。ログセット名は「clb_logset」となります。

4. **アクセスログ**ページの**ログトピック**のエリアで**ログトピックの新規作成**をクリックします。

5. ポップアップした**ログトピックの追加**ダイアログボックスで、ストレージタイプとログの保存期間を選択した後、左側のCLBインスタンスを選択して右側のリストに追加し、**保存**をクリックします。

説明：

ストレージタイプには標準ストレージと低頻度ストレージがあります。詳細については、ストレージタイプの概要をご参照ください。

ログの保存は永久保存および固定期間での保存をサポートしています。

ログトピックを新規作成する際は、CLBインスタンスを追加するかどうかを選択できます。ログトピックリストの右側の**操作**列で**管理**をクリックすると、CLBインスタンスを再度追加できます。各CLBインスタンスは1つのログトピックにのみ追加できます。

1つのログセットに複数のログトピック（Topic）を作成することができます。さまざまなCLBログをさまざまなログトピックに保存することが可能であり、これらのログトピックにはデフォルトで**CLB**の表示が付帯します。

6. （オプション）アクセスログを無効化したい場合は、ログトピックリストの右側の**操作**列で**停止**をクリックし、ログの配信を停止します。

ステップ2：アクセスログの確認

CLBはアクセスログの変数をキー値とするインデックスを自動的に設定しているため、手動でインデックスを設定する必要はありません。検索分析によってそのままアクセスログの照会を行うことができます。

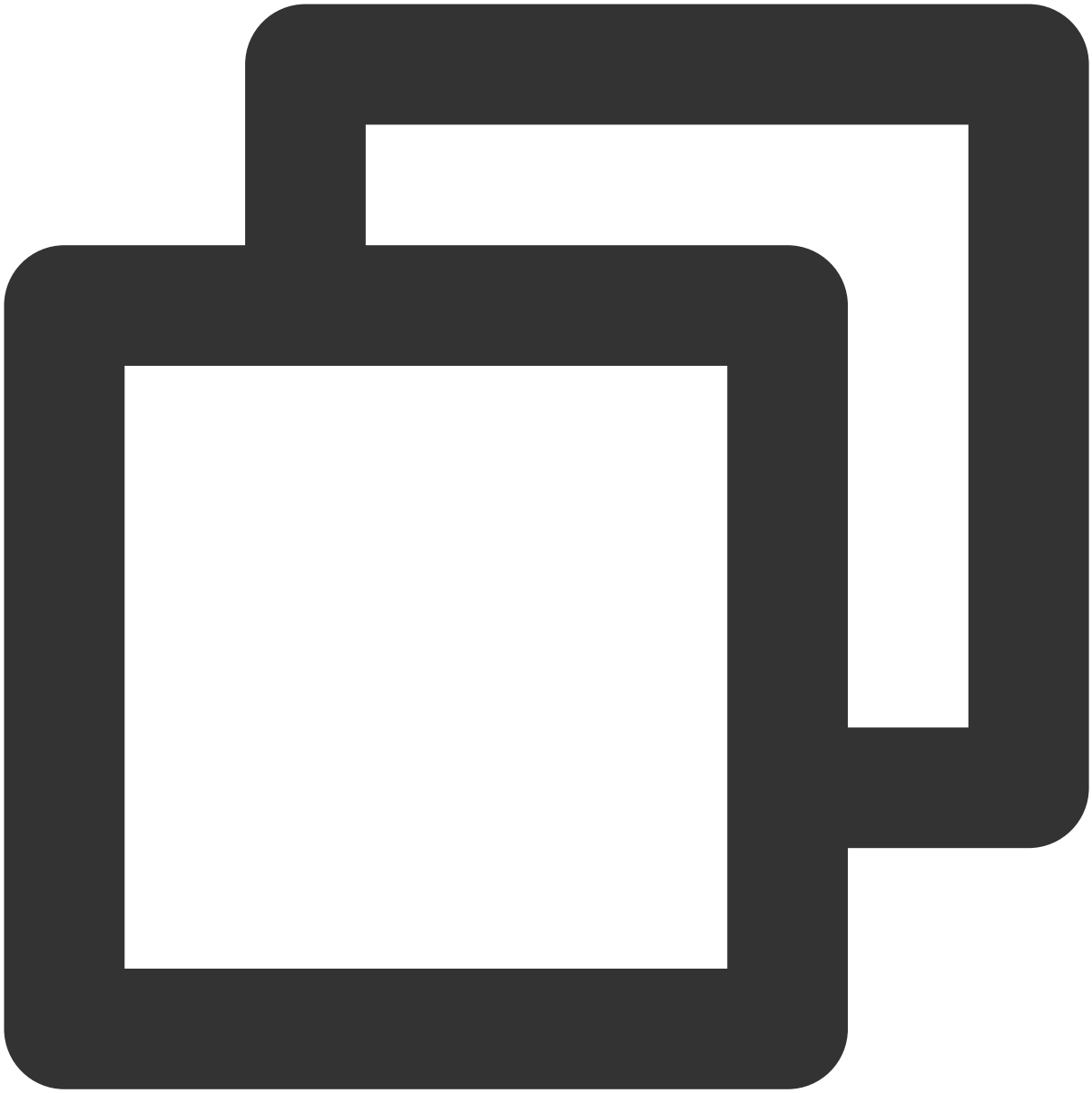
1. CLBコンソールにログインし、左側ナビゲーションバーの**アクセスログ**をクリックします。
2. 目的のログトピック右側の**操作**列の**検索**をクリックし、CLSコンソールの「検索分析」ページにリダイレクトします。
3. **検索分析**ページの入力ボックスに検索分析語を入力し、時間範囲を選択して**検索分析**をクリックすると、CLBがCLSに送信したアクセスログを検索できます。

説明：

検索構文の詳細については、構文とルールをご参照ください。

ログ形式および変数の説明

ログ形式



```
[$stgw_request_id] [$time_local] [$protocol_type] [$server_addr:$server_port]  [$se
```

フィールドタイプ

CLSは現在、次の3種類のフィールドタイプをサポートしています。

名前	タイプ説明
text	テキストタイプ
long	整数値タイプ（Int 64）

double	浮動小数点数値タイプ（64 bit）

ログ変数の説明

変数名	説明	フィールドタイプ
stgw_request_id	リクエストID	text
time_local	アクセスの時刻とタイムゾーンです。例えば「01/Jul/2019:11:11:00 +0800」の場合、最後の「+0800」は属するタイムゾーンがUTCの8時間後、すなわち北京時間であることを表します。	text
protocol_type	プロトコルタイプ（HTTP/HTTPS/SPDY/HTTP2/WS/WSS）。	text
server_addr	CLBのVIP。	text
server_port	CLBのVPort、すなわちリスニングポートです。	long
server_name	ルール化されたserver_name。CLBのリスナー内に設定されたドメイン名です。	text
remote_addr	クライアントIP	text
remote_port	クライアントのポートです。	long
status	CLBがクライアントに返すステータスコードです。	long
upstream_addr	RSアドレスです。	text
upstream_status	RSがCLBに返すステータスコードです。	text
proxy_host	stream IDです。	text
request	リクエスト行です。	text
request_length	クライアントから受信したリクエストバイト数です。	long
bytes_sent	クライアントに送信したバイト数です。	long
http_host	リクエストドメイン名、すなわちHTTP ヘッダー内のHostです。	text
http_user_agent	HTTPプロトコルヘッダーのuser_agent フィールドです。	text
http_referer	HTTPリクエストのソースです。	text
http_x_forward_for	HTTPリクエスト内のx-forward-for headerの内容です。	text

request_time	リクエストの処理時間です。クライアントからの最初のバイトの受信時から、クライアントに最後のバイトを送信するまでの時間です。クライアントリクエストがCLBに到着し、CLBがリクエストをRSに転送し、RSが応答データをCLBに送信し、CLBがデータをクライアントに転送するまでのすべての時間が含まれます。単位: 秒。	double
upstream_response_time	バックエンドリクエスト全体が消費する時間：CONNECT RSを開始してからRSから応答を受信するまでの時間です。単位: 秒。	double
upstream_connect_time	およびRSがTCP接続を確立する所要時間：CONNECT RSを開始してからHTTPリクエストの送信を開始するまでの時間です。	double
upstream_header_time	RSからHTTPヘッダーの受信を完了する所要時間：CONNECT RSを開始してからRSからHTTPレスポンスヘッダーの受信を完了するまでの時間です。	double
tcpinfo_rtt	TCP接続のRTTです。	long
connection	接続IDです。	long
connection_requests	接続のリクエスト個数です。	long
ssl_handshake_time	SSLハンドシェイクの各段階の消費時間を記録します。形式はx:x:x:x:x:x:xです。このうち、コロンで区切られた文字列は、単位がmsで、各段階の消費時間が1ms未満の場合は0と表示されます。 最初のフィールドはSSLセッションを再利用したかどうかを表します。 2番目のフィールドはハンドシェイク全体の時間を表します。 3~7はSSLの各段階の消費時間を表します。 3番目のフィールドはCLBのclient hello受信からserver hell done送信までの時間を表します。 4番目のフィールドはCLBのserver証明書送信開始からserver証明書送信完了までの時間を表します。 5番目のフィールドはCLBの署名計算からserver key exchange送信完了までの時間を表します。 6番目のフィールドはCLBのclient key exchange受信開始からclient key exchange受信完了までの時間を表します。 7番目のフィールドはCLBのclient key exchange受信からserver finished送信までの時間を表します。	text
ssl_cipher	SSL暗号スイートです。	text
ssl_protocol	SSLプロトコルバージョンです。	text
vip_vpcid	CLBインスタンスが所属するプライベートネットワーク ID。パブ	long

	リックネットワークCLBの値は-1です。	
request_method	リクエスト方式は、POSTおよびGETリクエストをサポートしています。	text
uri	リソース識別子です。	text
server_protocol	CLBのプロトコルです。	text

デフォルトで検索をサポートするログ変数

「CLB」の表示があるログセットの、デフォルトで検索をサポートするフィールドは次のとおりです。

インデックスフィールド	説明	フィールドタイプ
time_local	アクセスの時刻とタイムゾーンです。例えば「01/Jul/2019:11:11:00 +0800」の場合、最後の「+0800」は属するタイムゾーンがUTCの8時間後、すなわち北京時間であることを表します。	text
protocol_type	プロトコルタイプ（HTTP/HTTPS/SPDY/HTTP2/WS/WSS）。	text
server_addr	CLBのVIP。	text
server_name	ルール化されたserver_name。CLBのリスナー内に設定されたドメイン名です。	text
remote_addr	クライアントIP	text
status	CLBがクライアントに返すステータスコードです。	long
upstream_addr	RSアドレスです。	text
upstream_status	RSがCLBに返すステータスコードです。	text
request_length	クライアントから受信したリクエストバイト数です。	long
bytes_sent	クライアントに送信したバイト数です。	long
http_host	リクエストドメイン名、すなわちHTTP ヘッダー内のHostです。	text
request_time	リクエストの処理時間です。クライアントからの最初のバイトの受信時から、クライアントに最後のバイトを送信するまでの時間です。クライアントリクエストがCLBに到着し、CLBがリクエストをRSに転送し、RSが応答データをCLBに送信し、CLBがデータをクライアントに転送するまでのすべての時間が含まれます。単位:秒。	double

upstream_response_time	バックエンドリクエスト全体が消費する時間：CONNECT RSを開始してからRSから応答を受信するまでの時間です。単位: 秒。	double

ログサンプリング

最終更新日：：2024-01-04 18:36:26

レイヤー7アクセスログまたはヘルスチェックログを有効化すると、ログの量が大きいシナリオに対して、全量ログレポートはログコストが高くなる可能性があります。CLBは一部のログのサンプリングをサポートし、データの報告量を減少させることによってログコストを削減します。

説明：

CLBはアクセスログの設定およびヘルスチェックログをログサービスCLSに記録することをサポートし，ログデータの検索分析、可視化およびアラートなどのサービスを実現します。Tencent Cloud Log Service(CLS)は独立した課金製品です。課金基準についてはCLS課金の詳細をご参照ください。

前提条件

アクセスログのログセットおよびログトピックを作成済みであること。詳細については、アクセスログの設定をご参照ください。

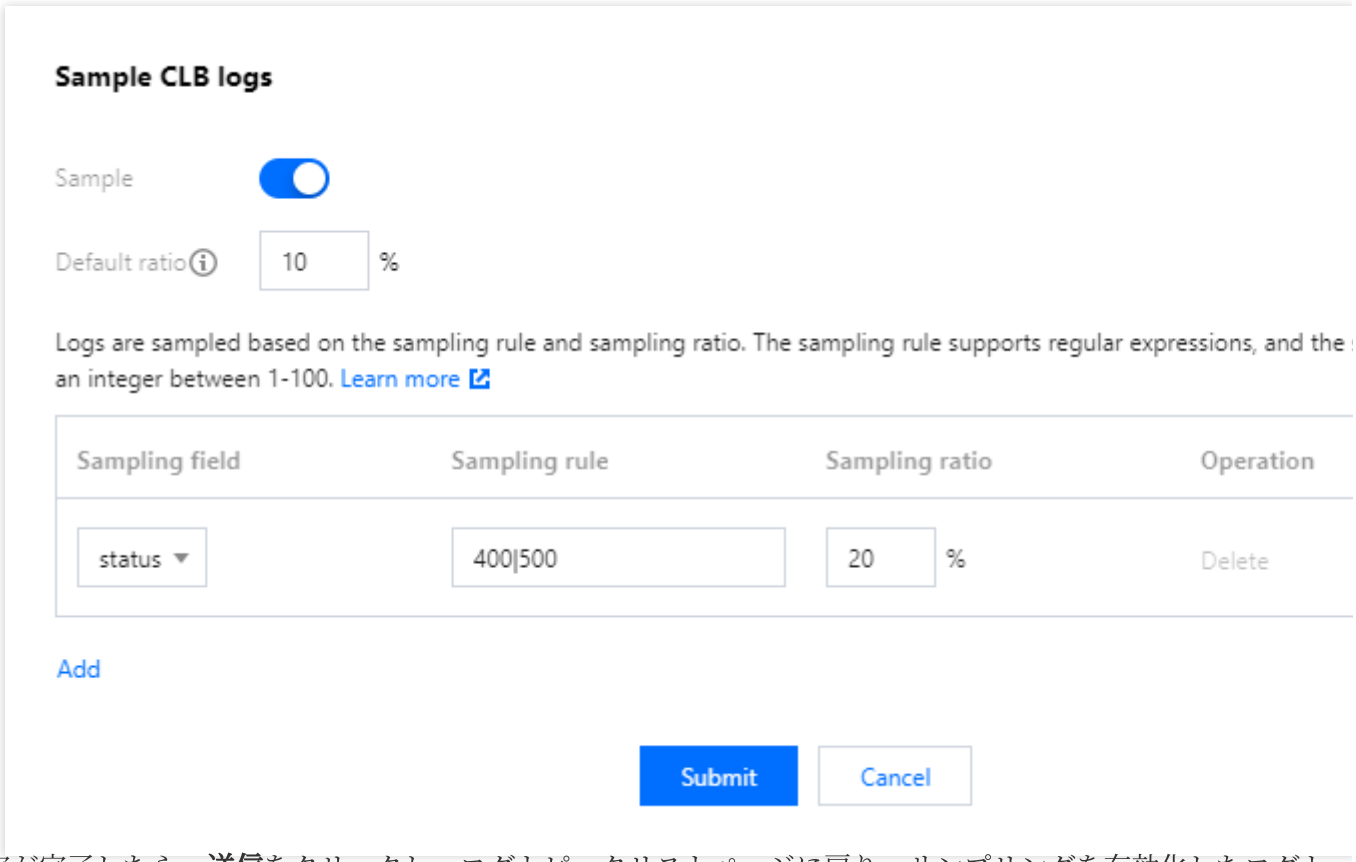
ヘルスチェックログのログセットおよびログトピックを作成済みであること。詳細については、ヘルスチェックログの設定をご参照ください。

レイヤー7アクセスログのサンプリング

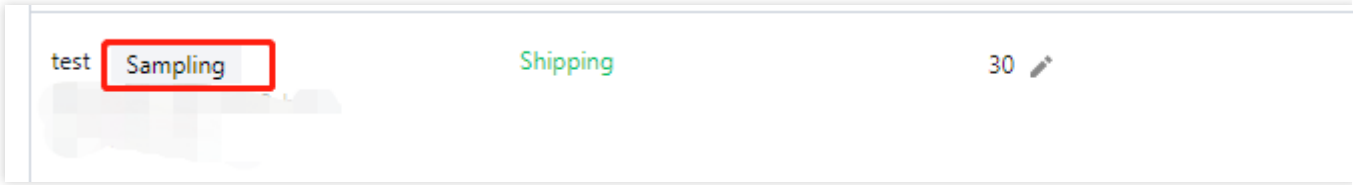
1. CLBコンソールにログインし、左側ナビゲーションバーの**アクセスログ > ログリスト**を選択します。
2. **アクセスログ**詳細ページ左上隅で所在リージョンを選択し、ログトピックリストで目標のログトピックを見つけ、**操作**列の**その他 > サンプリング**を選択します。
3. ポップアップした**CLBログサンプリング管理**ダイアログボックスで、サンプリングを有効化し、必要に応じてパラメータを設定します。

パラメータ	説明
サンプリングの有効化/無効化	有効化すると、ログのサンプリングをサポートします。 無効化すると、ログを全量収集し、サンプリングを行いません。
デフォルトのサンプリング比率	ログサンプリングのサンプリングルールを設定すると、このサンプリングルールに一致しないログはデフォルトサンプリング比率に従ってログ収集を行います。1～100の整数の入力をサポートしています。
サンプリングフィールド	現在サンプリングをサポートしているログフィールドはstatusコードです。
サンプリングルール	サンプリングルールは正規表現をサポートしています。例えばstatusコードが400

	または500のログをサンプリングしたい場合、サンプリングルールを400|500に設定することができます。
サンプリング比率	サンプリングを定義するために使用される比率です。1～100の整数の入力をサポートしています。
操作	サンプリングルールの削除を選択することができます。
追加	現在のサンプリングルールがニーズを満たしていない場合、サンプリングルールの追加を継続することを選択できます。各ログトピックは最大5つのサンプリングルールの設定をサポートしています。



4. 設定が完了したら、**送信**をクリックし、ログトピックリストページに戻り、サンプリングを有効化したログトピックは**サンプリング**タグが追加されます。



ヘルスチェックログのサンプリング

1. CLBコンソールにログインし、左側ナビゲーションバーの**ヘルスチェックログ**を選択します。

2. 残りのステップは上記のサンプリングレイヤー7アクセスログをご参照ください。

関連ドキュメント

アクセスログの設定
ヘルスチェックログの設定

ヘルスチェックログの設定

最終更新日：：2024-01-04 18:36:26

ヘルスチェックログを確認したい場合は、まずログをCloud Log Service（CLS）に保存し、CLSで確認する必要があります。CLBはヘルスチェックログのCLSへの保存をサポートしており、分単位でのログレポートおよびオンラインマルチルール検索が可能です。ヘルスチェックでの異常の原因をトラブルシューティングし、問題を迅速に特定する上で役立ちます。

説明：

ヘルスチェックログ機能は現在ベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。

ヘルスチェックログ機能にはログレポート、ログのストレージと照会があります。

ログレポート：業務の転送を優先的に保障した後にログレポートを保障します。

ログのストレージと照会：現在使用中のストレージサービスに基づいてサービス品質保証（SLA）を提供します。

制限事項

CLBのレイヤー4、レイヤー7プロトコルはどちらもヘルスチェックログのCLSへの設定をサポートしています。

CLBによるヘルスチェックログのCLSへの保存設定機能は無料です。ユーザーにはCLSの料金のみがかかります。

この機能をサポートしているのはCLB（旧「アプリケーション型CLB」）インスタンスタイプのみです。従来型CLBインスタンスタイプはサポートしていません。

この機能をサポートしているのはIPバージョンがIPv4およびIPv6 NAT64のインスタンスのみです。IPv6バージョンのインスタンスは現時点ではサポートしていません。

この機能は現在一部のリージョンでのみサポートされています。実際には、コンソールのサポートリージョンに準じます。

ステップ1：ロール権限の追加

CLSをアクティブ化していない場合は、先にCLSのアクティブ化を行ってからロール権限を追加してください。

1. CLBコンソールにログインし、左側ナビゲーションバーの**ヘルスチェックログ**をクリックします。
2. 「ヘルスチェックログ」ページで**今すぐアクティブにする**をクリックし、ポップアップしたダイアログボックスで**権限を承認してアクティブにする**をクリックします。
3. CAMコンソールにリダイレクトし、「ロール管理」ページで**権限承認に同意**をクリックします。

ステップ2：ログセットとログトピックの作成

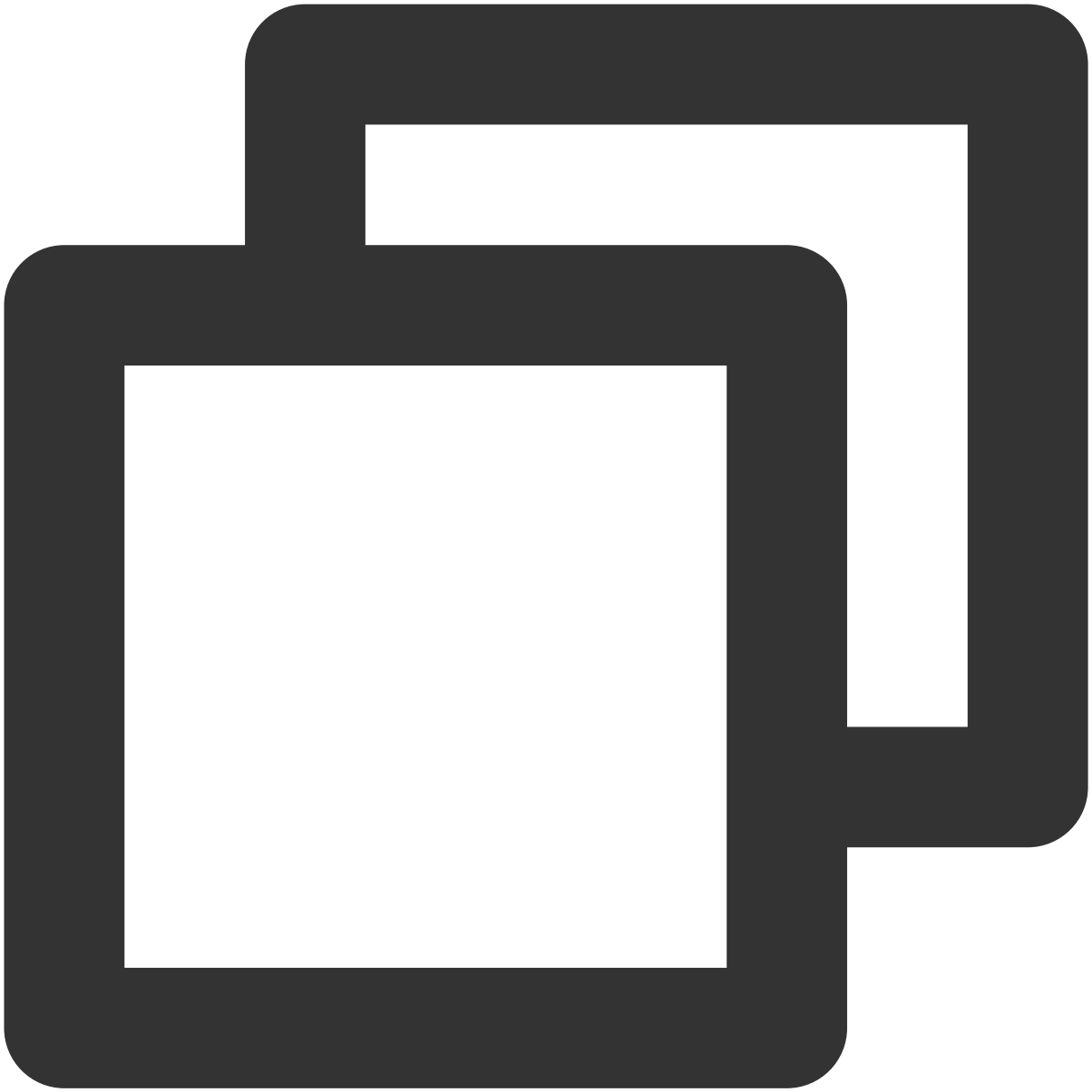
ヘルスチェックログをCLSに保存するよう設定したい場合は、先にログセットとログトピックを作成する必要があります。
ログセットとログトピックを作成済みの場合は、スキップしてステップ3 から操作を開始することができます。
1. CLBコンソールにログインし、左側ナビゲーションバーの**ヘルスチェックログ**をクリックします。
2. **ヘルスチェックログ**ページの左上隅で所属リージョンを選択し、**ログセット情報**のエリアで**ログセットの作成**をクリックします。
3. ポップアップした**ログセットの作成**ダイアログボックスで保存期間を設定し、**保存**をクリックします。
4. **ヘルスチェックログ**ページの**ログトピック**のエリアで**ログトピックの新規作成**をクリックします。
5. ポップアップした**ログトピックの追加**ダイアログボックスで、ストレージタイプとログの保存期間を選択した後、左側のCLBインスタンスを選択して右側のリストに追加し、**保存**をクリックします。
説明：
ストレージタイプには標準ストレージと低頻度ストレージがあります。詳細については、ストレージタイプの概要をご参照ください。
ログの保存は永久保存および固定期間での保存をサポートしています。
ログトピックを新規作成する際は、CLBインスタンスを追加するかどうかを選択できます。ログトピックリストの右側の**操作**列で**管理**をクリックすると、CLBインスタンスを再度追加できます。各CLBインスタンスは1つのログトピックにのみ追加できます。
1つのログセットに複数のログトピック（Topic）を作成することができます。さまざまなCLBログをさまざまなログトピックに保存することが可能であり、これらのログトピックにはデフォルトで「CLB」の表示が付帯します。
6. （オプション）ヘルスチェックログを無効化したい場合は、ログトピックリストの右側の**操作**列で**停止**をクリックし、ログの配信を停止します。

ステップ3：ヘルスチェックログの確認

CLBはヘルスチェックログの変数をキー値とするインデックスを自動的に設定しているため、手動でインデックスを設定する必要はありません。検索分析によってそのままヘルスチェックログの照会を行うことができます。
1. CLBコンソールにログインし、左側ナビゲーションバーの**ヘルスチェックログ**をクリックします。
2. 「ヘルスチェックログ」ページの左上隅で所属リージョンを選択し、「ログトピック」エリアで右側の「操作」列の**検索**をクリックし、CLSコンソールにリダイレクトします。
3. CLSコンソールで、左側ナビゲーションバーの**検索分析**をクリックします。
4. **検索分析**ページの入力ボックスに検索分析語を入力し、時間範囲を選択して**検索分析**をクリックすると、CLBがCLSに送信したヘルスチェックログを検索できます。
説明：
検索構文の詳細については、構文とルールをご参照ください。

ヘルスチェックログの形式と説明

ログ形式



```
[$protocol][$rsport][$rs_vpcid][$vport][$vpcid][$time][$vip][$rsip][$status][$domai
```

ログ変数の説明

変数名	説明	フィールドタイプ

protocol	プロトコルタイプ（HTTP/HTTPS/SPDY/HTTP2/WS/WSS）。	text
rsport	バックエンドRSポートです。	long
rs_vpcid	バックエンドRSの所属プライベートネットワークID。パブリックネットワークCLBのvip_vpcidは-1です。	long
vport	CLBのVPort、すなわちリスニングポートです。	long
vpcid	CLB VIPの所属プライベートネットワークID。パブリックネットワークCLBのvip_vpcidは-1です。	long
time	アクセスの時刻とタイムゾーンです。例えば「01/Jul/2019:11:11:00 +0800」の場合、最後の「+0800」は属するタイムゾーンがUTCの8時間後、すなわち北京時間であることを表します。	text
vip	CLBのVIP。	text
rsip	バックエンドRSのIPアドレスです。	text
status	現在のヘルスチェックステータス： true：健康であることを示します false：異常であることを示します	text
domain	ヘルスチェックドメイン名。リスナーがレイヤー4リスナーの場合、ヘルスチェックドメイン名はなく、このパラメータは空白です。	text
url	ヘルスチェック URL。リスナーがレイヤー4リスナーの場合、ヘルスチェックURLはなく、このパラメータは空白です。	text

関連ドキュメント

CLSクイックスタート

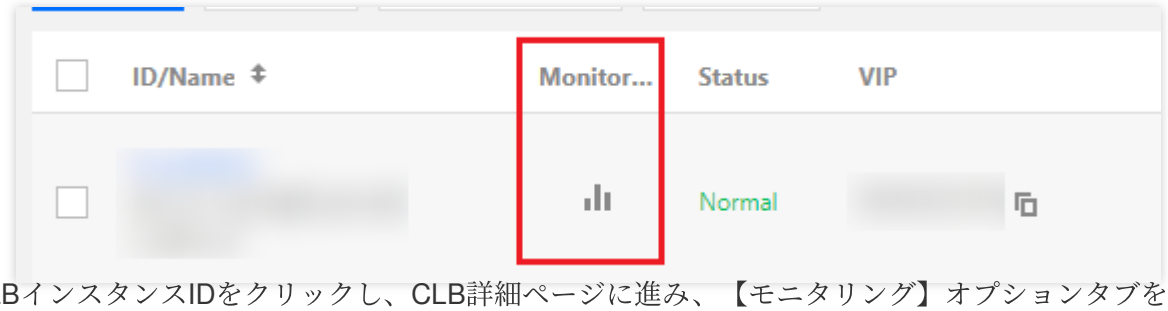
監視アラート
監視データの取得

最終更新日：：2024-01-04 18:36:26

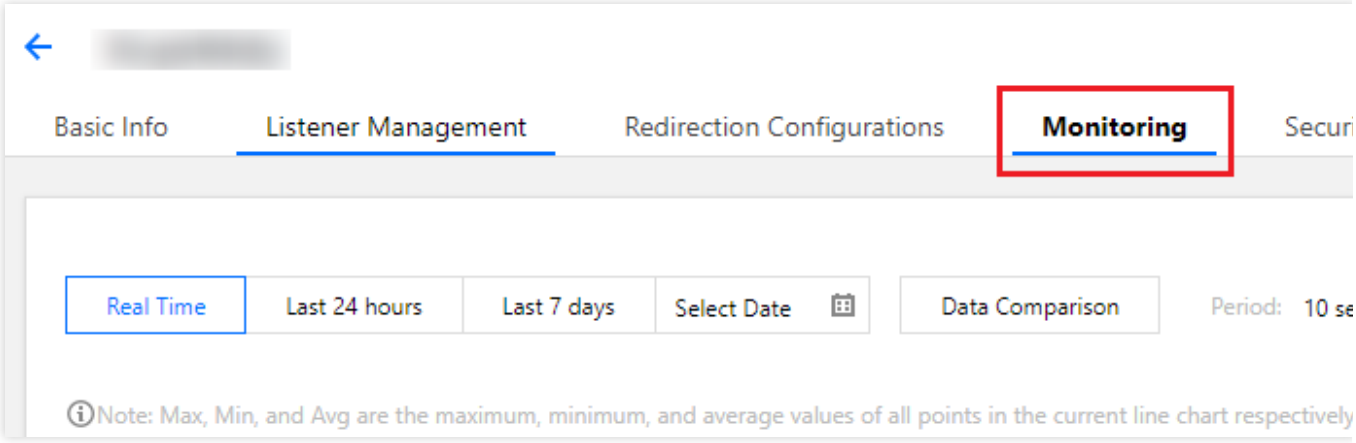
Tencent CloudのCloud Monitor（CM）はCLBおよびバックエンドインスタンスにデータ収集およびデータ表示機能を提供します。Tencent Cloud CMを使用すると、CLBの統計データを確認し、システムが正常に動作しているかを検証できるほか、それに応じたアラートを作成することもできます。CMに関するその他の情報については、CM製品ドキュメントをご参照ください。
Tencent Cloudはデフォルトですべてのユーザーに CM機能をご提供しています。手動でのアクティブ化は必要なく、CLBを使用するだけで、CMが関連の監視データの収集をサポートします。CLBの監視データは次のいくつかの方法で確認できます。

CLBコンソール

1. CLBコンソールにログインし、CLBインスタンスIDの横の監視アイコンをクリックすると、監視フローティングウィンドウから各インスタンスのパフォーマンスデータをすぐに閲覧することができます。



2. CLBインスタンスIDをクリックし、CLB詳細ページに進み、【モニタリング】オプションタブをクリックすると、現在のCLBインスタンスの監視データを確認できます。



CMコンソール

CMコンソールにログインし、左側ナビゲーションバーの「クラウド製品監視」モジュールの【Cloud Load Balancer-CLB】をクリックし、CLBインスタンスIDをクリックして監視詳細ページに進むと、そのCLBインスタンスの監視データを確認できます。インスタンスを表示すると、リスナー、バックエンドサーバーなどの監視情報を確認できます。

API方式

GetMonitorDataインターフェースを使用して全製品の監視情報を取得することができます。具体的な内容については指標のモニタリングデータのプルをご参照ください。CLBのネームスペースについては、パブリックネットワークCLBの監視指標、プライベートネットワークCLBレイヤー4プロトコルの監視指標をご参照ください。

監視指標の説明

最終更新日：：2024-01-04 18:36:26

Tencent Cloud Observability Platform (TCOP) は実行中のCLBインスタンスからオリジナルデータを収集し、わかりやすいチャート形式でデータを表示します。統計データはデフォルトで1か月間保存されます。インスタンスの1か月間の実行状況を観察することで、アプリケーションサービスの実行状況をより適切に把握することができます。

CLBの監視データはTCOPコンソールで確認することをお勧めします。**クラウド製品監視** > **Cloud Load Balancer-CLB**を選択し、CLBインスタンスIDをクリックして監視詳細ページに進み、そのCLBインスタンスの監視データを確認します。インスタンスを表示すると、リスナー、バックエンドサーバーなどの監視情報を確認できます。

説明：

ここに記載する指標はすべて基本指標です。より幅広い監視機能が必要な場合は、高度な指標を有料でアクティブ化できます。

CLBの高度な指標には、インスタンスディメンションの最大接続数使用率（ConcurConnVipRatio）および新規接続数使用率（NewConnVipRatio）の指標が含まれます。

現在はLCUタイプのCLBインスタンスに限り、最大接続数使用率、新規接続数使用率の指標をアクティブ化するとデータがレポートされます。共有タイプのCLBインスタンスでは、現時点ではデータがレポートされません。

CLBインスタンスディメンション

指標の英語名	指標の日本語名	指標の説明	単位	統計周期（秒）
ClientConnum	クライアントからLBへのアクティブな接続数	統計周期内のある時点における、クライアントからCLBまたはリスナーへのアクティブな接続数です。	個	10、60、300
ClientInactiveConn	クライアントからLBへの非アクティブな接続数	統計周期内のある時点における、クライアントからCLBまたはリスナーへの非アクティブな接続数です。	個	10、60、300
ClientConcurConn	クライアントからLBへの同時接続数	統計周期内のある時点における、クライアントからCLBまたはリスナーへの同時接続数です。	個	10、60、300
ClientNewConn	クライアントからLBへの新規接続	統計周期内におけるクライアントからCLBまたはリスナーへの新規	個/秒	10、60、300

	数	接続数です。		
ClientInpkg	クライアントからLBへのインバウンドパケット	統計周期内におけるクライアントがCLBへ1秒あたりに送信するデータパケット数です。	個/秒	10、60、300
ClientOutpkg	クライアントからLBへのアウトバウンドパケット	統計周期内でCLBがクライアントへ1秒あたりに送信するデータパケット数です。	個/秒	10、60、300
ClientAccIntraffic	クライアントからLBへのインバウンドトラフィック	統計周期内におけるクライアントからCLBに流入するトラフィックです。	MB	10、60、300
ClientAccOuttraffic	クライアントからLBへのアウトバウンドトラフィック	統計周期内におけるCLBからクライアントに流出するトラフィックです。	MB	10、60、300
ClientOuttraffic	クライアントからLBへのアウトバウンド帯域幅	統計周期内におけるCLBからクライアントへの流出に使用する帯域幅です。	Mbps	10、60、300
ClientIntraffic	クライアントからLBへのインバウンド帯域幅	統計周期内におけるクライアントからCLBへの流入に使用する帯域幅です。	Mbps	10、60、300
OutTraffic	LBからバックエンドへのアウトバウンド帯域幅	統計周期内におけるバックエンドサーバーからCLBへの流出に使用する帯域幅です。	Mbps	60、300
InTraffic	LBからバックエンドへのインバウンド帯域幅	統計周期内におけるCLBからバックエンドサーバーへの流入に使用する帯域幅です。	Mbps	60、300
AccOuttraffic	LBからバックエンドへのアウトバウンドトラフィック	統計周期内におけるバックエンドサーバーからCLBへ流出するトラフィックです。 この指標は、パブリックネットワークのCLBインスタンスでのみサポートされ、プライベートネットワークCLBではサポートされません。	MB	10、60、300、3600
DropTotalConns	破棄接続数	統計周期内でCLBまたはリスナーで破棄される接続数です。	個	10、60、300

		この指標は標準的なアカウントタイプでのみサポートされており、従来型のアカウントタイプではサポートされていません。アカウントタイプが確定できない場合は、アカウントタイプの判断をご参照ください。		
InDropBits	破棄インバウンド帯域幅	統計周期内でクライアントがパブリックネットワークを介してCLBにアクセスする際に破棄される帯域幅です。 この指標は標準的なアカウントタイプでのみサポートされており、従来型のアカウントタイプではサポートされていません。アカウントタイプが確定できない場合は、アカウントタイプの判断をご参照ください。	バイト	10、60、300
OutDropBits	破棄アウトバウンド帯域幅	統計周期内でCLBがパブリックネットワークにアクセスする際に破棄される帯域幅です。 この指標は標準的なアカウントタイプでのみサポートされており、従来型のアカウントタイプではサポートされていません。アカウントタイプが確定できない場合は、アカウントタイプの判断をご参照ください。	バイト	10、60、300
InDropPkts	破棄流入データパケット	統計周期内でクライアントがパブリックネットワークを介してCLBにアクセスする際に破棄されるデータパケットです。 この指標は標準的なアカウントタイプでのみサポートされており、従来型のアカウントタイプではサポートされていません。アカウントタイプが確定できない場合は、アカウントタイプの判断をご参照ください。	個/秒	10、60、300
OutDropPkts	破棄流出データパケット	統計周期内でCLBがパブリックネットワークにアクセスする際に破棄されるデータパケットです。	個/秒	10、60、300

		この指標は標準的なアカウントタイプでのみサポートされており、従来型のアカウントタイプではサポートされていません。アカウントタイプが確定できない場合は、アカウントタイプの判断をご参照ください。		
DropQps	破棄QPS	統計周期内でCLBまたはリスナーで破棄されるリクエスト数です。この指標はレイヤー7リスナーに固有のもので、標準的なアカウントタイプでのみサポートされており、従来型のアカウントタイプではサポートされていません。アカウントタイプが確定できない場合は、アカウントタイプの判断をご参照ください。	個	60、300
IntrafficVipRatio	インバウンド帯域幅使用率	統計周期内でクライアントがパブリックネットワークを介してCLBにアクセスする際に使用する帯域幅の使用率です。 この指標は標準アカウントタイプのみサポートしており、従来型アカウントタイプではサポートしていません。アカウントタイプの判断方法については、アカウントタイプの判断をご参照ください。この指標はベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。	%	10、60、300
OuttrafficVipRatio	アウトバウンド帯域幅使用率	統計周期内でCLBがパブリックネットワークにアクセスする際に使用する帯域幅の使用率です。 この指標は標準アカウントタイプのみサポートしており、従来型アカウントタイプではサポートしていません。アカウントタイプの判断方法については、アカウントタイプの判断をご参照ください。この指標はベータ版テスト段階です。ご利用を希望される場合は、	%	10、60、300

		チケット申請を提出してください。		
ReqAvg	平均リクエスト時間	統計周期内におけるCLBの平均リクエスト時間です。 この指標はレイヤー7リスナーに固有の指標です。	ミリ秒	60、300
ReqMax	最大リクエスト時間	統計周期内におけるCLBの最大リクエスト時間です。 この指標はレイヤー7リスナーに固有の指標です。	ミリ秒	60、300
RspAvg	平均応答時間	統計周期内におけるCLBの平均レスポンス時間です。 この指標はレイヤー7リスナーに固有の指標です。	ミリ秒	60、300
RspMax	最大レスポンス時間	統計周期内におけるCLBの最大レスポンス時間です。 この指標はレイヤー7リスナーに固有の指標です。	ミリ秒	60、300
RspTimeout	レスポンスタイムアウト個数	統計周期内におけるCLBのレスポンスタイムアウトの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
SuccReq	1分あたりのリクエスト成功数	統計周期内におけるCLBの1分あたりのリクエスト成功数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
TotalReq	1秒あたりのリクエスト数	統計周期内におけるCLBの1秒あたりのリクエスト数です。 この指標はレイヤー7リスナーに固有の指標です。	個	60、300
ClbHttp3xx	CLBが返した3xxステータスコード	統計周期内でCLBが返した3xxステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
ClbHttp4xx	CLBが返した4xxステータスコード	統計周期内でCLBが返した4xxステータスコードの数（CLBとバッ	個/分	60、300

		クエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。		
ClbHttp5xx	CLBが返した5xxステータスコード	統計周期内でCLBが返した5xxステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
ClbHttp404	CLBが返した404ステータスコード	統計周期内でCLBが返した404ステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
ClbHttp499	CLBが返した499ステータスコード	統計周期内でCLBが返した499ステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
ClbHttp502	CLBが返した502ステータスコード	統計周期内でCLBが返した502ステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
ClbHttp503	CLBが返した503ステータスコード	統計周期内でCLBが返した503ステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
ClbHttp504	CLBが返した504ステータスコード	統計周期内でCLBが返した504ステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300

Http2xx	2xxステータスコード	統計周期内におけるバックエンドサーバーが返した2xxステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http3xx	3xxステータスコード	統計周期内におけるバックエンドサーバーが返した3xxステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http4xx	4xxステータスコード	統計周期内におけるバックエンドサーバーが返した4xxステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http5xx	5xxステータスコード	統計周期内におけるバックエンドサーバーが返した5xxステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http404	404ステータスコード	統計周期内におけるバックエンドサーバーが返した404ステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http499	499ステータスコード	統計周期内におけるバックエンドサーバーが返した499ステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http502	502ステータスコード	統計周期内におけるバックエンドサーバーが返した502ステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http503	503ステータスコード	統計周期内におけるバックエンドサーバーが返した503ステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300

Http504	504ステータスコード	統計周期内におけるバックエンドサーバーが返した504ステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
OverloadCurConn	SNAT同時接続数	統計周期内におけるCLBのSNAT IPの1分あたりの同時接続数です。 この指標はベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。	個/分	60
ConnRatio	SNATポート使用率	統計周期内におけるCLBのSNAT IPのポート使用率です。 ポート使用率 = SNAT同時接続数 / (SNAT IP数×55000×サーバー数）。 この指標はベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。	%	60
SnatFail	SNAT失敗数	統計周期内におけるCLBのSNAT IPとバックエンドサーバー間で接続に失敗した1分あたりの回数です。 この指標はベータ版テスト段階です。ご利用を希望される場合は、チケット申請を提出してください。	個/分	60
UnhealthRsCount	ヘルスチェック異常数	統計周期内におけるCLBのヘルスチェックの異常数です。	個	60、300

レイヤー4リスナー（TCP/UDP）ディメンション

レイヤー4リスナーは以下の3つのディメンションで、下表の各監視指標の確認をサポートします。
リスナーディメンションです。
バックエンドサーバーディメンションです。
バックエンドサービスのポートディメンションです。

指標の英語名	指標の日本語名	指標の説明	単位	統計周期

				（秒）
ClientConnum	クライアントからLBへのアクティブな接続数	統計周期内のある時点における、クライアントからCLBまたはリスナーへのアクティブな接続数です。	個	10、60、300
ClientNewConn	クライアントからLBへの新規接続数	統計周期内におけるクライアントからCLBまたはリスナーへの新規接続数です。	個/秒	10、60、300
ClientInpkg	クライアントからLBへのインバウンドパケット	統計周期内におけるクライアントがCLBへ1秒あたりに送信するデータパケット数です。	個/秒	10、60、300
ClientOutpkg	クライアントからLBへのアウトバウンドパケット	統計周期内でCLBがクライアントへ1秒あたりに送信するデータパケット数です。	個/秒	10、60、300
ClientAccIntraffic	クライアントからLBへのインバウンドトラフィック	統計周期内におけるクライアントからCLBに流入するトラフィックです。	MB	10、60、300
ClientAccOuttraffic	クライアントからLBへのアウトバウンドトラフィック	統計周期内におけるCLBからクライアントに流出するトラフィックです。	MB	10、60、300
ClientOuttraffic	クライアントからLBへのアウトバウンド帯域幅	統計周期内におけるCLBからクライアントへの流出に使用する帯域幅です。	Mbps	10、60、300
ClientIntraffic	クライアントからLBへのインバウンド帯域幅	統計周期内におけるクライアントからCLBに流入するトラフィックです。	Mbps	10、60、300
OutTraffic	LBからバックエンドへのアウトバウンド帯域幅	統計周期内におけるバックエンドサーバーからCLBへの流出に使用する帯域幅です。	Mbps	60、300
InTraffic	LBからバックエンドへのインバウンド帯域幅	統計周期内におけるCLBからバックエンドサーバーへの流入に使用する帯域幅です。	Mbps	60、300
OutPkg	LBからバックエンドへのアウトバウンドパケット	統計周期内におけるバックエンドサーバーがCLBへ1秒あたりに送信するデータパケット数です。	個/秒	60、300
InPkg	LBからバックエン	統計周期内でCLBがバックエンド	個/秒	60、300

	ドへのインバウンドパケット	サーバーへ1秒あたりに送信するデータパケット数です。		
AccOuttraffic	LBからバックエンドへのアウトバウンドトラフィック	統計周期内におけるバックエンドサーバーからCLBへ流出するトラフィックです。 この指標は、パブリックネットワークのCLBインスタンスでのみサポートされ、プライベートネットワークCLBではサポートされません。	MB	10、60、300、3600
ConNum	LBからバックエンドへの接続数	統計周期内におけるCLBからバックエンドサーバーへの接続数です。	個	60、300
NewConn	LBからバックエンドへの新規接続数	統計周期内におけるCLBからバックエンドサーバーへの新規接続数です。	個/分	60、300
UnhealthRsCount	ヘルスチェック異常数	統計周期内におけるCLBのヘルスチェックの異常数です。	個	60、300

レイヤー7リスナー（HTTP/HTTPS）ディメンション

レイヤー7リスナーは以下の3つのディメンションで、下表の各監視指標の確認をサポートします。
リスナーディメンションです。
バックエンドサーバーディメンションです。
バックエンドサービスのポートディメンションです。

指標の英語名	指標の日本語名	指標の説明	単位	統計周期（秒）
ClientConnum	クライアントからLBへのアクティブな接続数	統計周期内のある時点における、クライアントからCLBまたはリスナーへのアクティブな接続数です。	個	10、60、300
ClientNewConn	クライアントからLBへの新規接続数	統計周期内におけるクライアントからCLBまたはリスナーへの新規接続数です。	個/秒	10、60、300
ClientInpkg	クライアントからLBへのインバウンドパケット	統計周期内におけるクライアントがCLBへ1秒あたりに送信するデータパケット数です。	個/秒	10、60、300

ClientOutpkg	クライアントからLBへのアウトバウンドパケット	統計周期内でCLBがクライアントへ1秒あたりに送信するデータパケット数です。	個/秒	10、60、300
ClientAccIntraffic	クライアントからLBへのインバウンドトラフィック	統計周期内におけるクライアントからCLBに流入するトラフィックです。	MB	10、60、300
ClientAccOuttraffic	クライアントからLBへのアウトバウンドトラフィック	統計周期内におけるCLBからクライアントに流出するトラフィックです。	MB	10、60、300
ClientOuttraffic	クライアントからLBへのアウトバウンド帯域幅	統計周期内におけるCLBからクライアントへの流出に使用する帯域幅です。	Mbps	10、60、300
ClientIntraffic	クライアントからLBへのインバウンド帯域幅	統計周期内におけるクライアントからCLBに流入するトラフィックです。	Mbps	10、60、300
OutTraffic	LBからバックエンドへのアウトバウンド帯域幅	統計周期内におけるバックエンドサーバーからCLBへの流出に使用する帯域幅です。	Mbps	60、300
InTraffic	LBからバックエンドへのインバウンド帯域幅	統計周期内におけるCLBからバックエンドサーバーへの流入に使用する帯域幅です。	Mbps	60、300
OutPkg	LBからバックエンドへのアウトバウンドパケット	統計周期内におけるバックエンドサーバーがCLBへ1秒あたりに送信するデータパケット数です。	個/秒	60、300
InPkg	LBからバックエンドへのインバウンドパケット	統計周期内でCLBがバックエンドサーバーへ1秒あたりに送信するデータパケット数です。	個/秒	60、300
AccOuttraffic	LBからバックエンドへのアウトバウンドトラフィック	統計周期内におけるバックエンドサーバーからCLBへ流出するトラフィックです。 この指標は、パブリックネットワークのCLBインスタンスでのみサポートされ、プライベートネッ	MB	10、60、300、3600

		トワークCLBではサポートされません。		
ConNum	LBからバックエンドへの接続数	統計周期内におけるCLBからバックエンドサーバーへの接続数です。	個	60、300
NewConn	LBからバックエンドへの新規接続数	統計周期内におけるCLBからバックエンドサーバーへの新規接続数です。	個/分	60、300
ReqAvg	平均リクエスト時間	統計周期内におけるCLBの平均リクエスト時間です。 この指標はレイヤー7リスナーに固有の指標です。	ミリ秒	60、300
ReqMax	最大リクエスト時間	統計周期内におけるCLBの最大リクエスト時間です。 この指標はレイヤー7リスナーに固有の指標です。	ミリ秒	60、300
RspAvg	平均応答時間	統計周期内におけるCLBの平均レスポンス時間です。 この指標はレイヤー7リスナーに固有の指標です。	ミリ秒	60、300
RspMax	最大レスポンス時間	統計周期内におけるCLBの最大レスポンス時間です。 この指標はレイヤー7リスナーに固有の指標です。	ミリ秒	60、300
RspTimeout	レスポンスタイムアウト個数	統計周期内におけるCLBのレスポンスタイムアウトの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
SuccReq	1分あたりのリクエスト成功数	統計周期内におけるCLBの1分あたりのリクエスト成功数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
TotalReq	1秒あたりのリクエスト数	統計周期内におけるCLBの1秒あたりのリクエスト数です。 この指標はレイヤー7リスナーに固有の指標です。	個	60、300
ClbHttp3xx	CLBが返した3xxステータスコー	統計周期内でCLBが返した3xxステータスコードの数（CLBとバッ	個/分	60、300

	ド	クエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。		
ClbHttp4xx	CLBが返した4xxステータスコード	統計周期内でCLBが返した4xxステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
ClbHttp5xx	CLBが返した5xxステータスコード	統計周期内でCLBが返した5xxステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
ClbHttp404	CLBが返した404ステータスコード	統計周期内でCLBが返した404ステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
ClbHttp499	CLBが返した499ステータスコード	統計周期内でCLBが返した499ステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
ClbHttp502	CLBが返した502ステータスコード	統計周期内でCLBが返した502ステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
ClbHttp503	CLBが返した503ステータスコード	統計周期内でCLBが返した503ステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300

ClbHttp504	CLBが返した504ステータスコード	統計周期内でCLBが返した504ステータスコードの数（CLBとバックエンドサーバーが返したコードの合計）です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http2xx	2xxステータスコード	統計周期内におけるバックエンドサーバーが返した2xxステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http3xx	3xxステータスコード	統計周期内におけるバックエンドサーバーが返した3xxステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http4xx	4xxステータスコード	統計周期内におけるバックエンドサーバーが返した4xxステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http5xx	5xxステータスコード	統計周期内におけるバックエンドサーバーが返した5xxステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http404	404ステータスコード	統計周期内におけるバックエンドサーバーが返した404ステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http499	499ステータスコード	統計周期内におけるバックエンドサーバーが返した499ステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http502	502ステータスコード	統計周期内におけるバックエンドサーバーが返した502ステータスコードの数です。	個/分	60、300

		この指標はレイヤー7リスナーに固有の指標です。		
Http503	503ステータスコード	統計周期内におけるバックエンドサーバーが返した503ステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
Http504	504ステータスコード	統計周期内におけるバックエンドサーバーが返した504ステータスコードの数です。 この指標はレイヤー7リスナーに固有の指標です。	個/分	60、300
UnhealthRsCount	ヘルスチェック異常数	統計周期内におけるCLBのヘルスチェックの異常数です。	個	60、300

説明：

特定のリスナー配下のバックエンドサーバーの監視データを確認したい場合は、CLBコンソールにログインし、CLBインスタンスIDの横の監視アイコンをクリックすると、監視フローティングウィンドウから各インスタンスのパフォーマンスデータをすぐに閲覧することができます。

関連ドキュメント

パブリックネットワークCLBの監視指標

アラートポリシーの設定

最終更新日：：2024-01-04 18:36:26

ここではアラートポリシーの作成方法についてご説明します。

ユースケース

Tencent Cloud Observability Platformがサポートする監視のタイプについて、パフォーマンス消費クラス指標の閾値アラートを設定できます。また、クラウド製品インスタンスまたはプラットフォームの基盤インフラストラクチャのサービスステータスについてもイベントアラートを設定でき、異常発生時に速やかに通知して措置をとれるようにします。アラートポリシーは、名称、ポリシータイプ、アラートトリガー条件、アラートオブジェクト、アラート通知テンプレートという5つの必要な部分によって構成されています。アラートポリシーの作成は次のガイドに基づいて行うことができます。

基本概念

用語	定義
アラートポリシー	アラート名、アラートポリシータイプ、アラートトリガー条件、アラートオブジェクト、アラート通知テンプレートで構成されます
アラートポリシータイプ	アラートポリシータイプはポリシーのカテゴリーを表すために用いられ、タイプはクラウド製品に対応しています。例えば、CVMポリシーを選択した場合、CPU使用率、ディスク使用率などの指標のアラートをカスタマイズできます
アラートトリガー条件	指標、比較関係、閾値、統計粒度、継続的なN個のデータ監視ポイントで構成されるセマンティック条件です
監視タイプ	クラウド製品監視、アプリケーションパフォーマンス管理、Real User Monitoring（RUM）、Cloud Automated Testing（CAT）が含まれます
通知テンプレート	複数のポリシー用に、ワンクリックで何度も使用できるテンプレートです。さまざまなシーンでのアラート通知受信に適用できます。詳細については、アラート通知テンプレートの新規作成をご参照ください

操作手順

1. Tencent Cloud Observability Platformにログインします。

2. **アラート設定** > **アラートポリシー**をクリックし、アラートポリシー設定ページに進みます。
3. **追加**をクリックし、アラートポリシーを設定します。設定に関する説明は次のとおりです。

設定タイプ	設定項目	説明
基本情報	ポリシー名	カスタムポリシー名
	備考	カスタムポリシーの備考
	監視タイプ	クラウド製品監視、アプリケーションパフォーマンス管理、Real User Monitoring（RUM）、Cloud Automated Testing（CAT）をサポートしています
	ポリシータイプ	監視したいクラウド製品のポリシータイプを選択します
	所属プロジェクト	所属プロジェクトには、次の2つの機能があります。 アラートポリシーを管理します。所属するプロジェクトを設定すると、アラートポリシーリストのプロジェクトにあるアラートポリシーをすぐにフィルタリングできます。 インスタンスを管理します。必要に応じてプロジェクトを選択すると、アラートオブジェクトのプロジェクトにあるインスタンスをすぐに選択することができます。ビジネスタイプに基づいて、クラウド製品をそれぞれのプロジェクトに割り当てることができます。プロジェクトを作成するには、プロジェクト管理をご参照ください。プロジェクトを作成すると、各クラウド製品のコンソールで、各クラウド製品のリソースにプロジェクトを割り当てることができます。一部のクラウド製品はプロジェクトの割り当てをサポートしていません（例えば、TencentDB for MySQLについてはインスタンスのプロジェクトの指定ガイドをご参照の上、インスタンスを対応するプロジェクトに割り当てることができます）。プロジェクトの権限がない場合は、Cloud Access Management をご参照の上、権限を承認してください。
アラートルールの設定	アラートオブジェクト	インスタンスIDを選択すると、このアラートポリシーがユーザーの選択したインスタンスにバインドされます。 インスタンスグループを選択すると、このアラートポリシーがユーザーの選択したインスタンスグループにバインドされます。 すべてのオブジェクトを選択すると、このアラートポリシーは、現在のアカウントに権限があるすべてのインスタンスにバインドされます。
	手動設定（インジケータアラート）	アラートトリガー条件：指標、比較関係、閾値、統計粒度、継続的なN個のデータ監視ポイントで構成されるセマンティック条件です。チャート上の指標のトレンドに応じ

		て、アラートの閾値を設定することができます。例えば、指標をCPU使用率、比較関係を>、閾値を80%、統計粒度を5分、継続的データ監視ポイントを2データポイントとします。これは、5分に1回CPU使用率を収集し、あるCVMのCPU使用率が2回連続して80%を超えると、アラートがトリガーされることを意味します。 アラート頻度：アラートルールごとに、繰り返し通知ポリシーを設定できます。すなわち、アラートが発生した時に、そのアラートが特定の頻度で繰り返し通知されるように定義できます。 繰り返しなし、5分、10分、周期的指数関数的な増加...などの繰り返し頻度から選択することができます。 周期的指数関数的な増加とは、そのアラートが、1回、2回、4回、8回...という2のN乗回でトリガーされた場合に、アラート情報をユーザーに向けて送信することを意味しています。つまり、アラート情報の送信間隔を長くしていくほど、アラートの繰り返しによる煩わしさをある程度回避できます。 繰り返しアラートのデフォルトロジック：アラートが発生してから24時間以内に、繰り返し通知用に設定した頻度に基づき、繰り返しアラート通知が送信されます。アラート発生からまるまる24時間が経過すると、1日1回アラート通知が送信されるようになります。
	手動設定（イベントアラート）	クラウド製品リソースまたは基盤となるインフラストラクチャサービスに異常が発生した場合、イベントアラートを作成して、講じる対策について速やかにお知らせすることができます。
	テンプレートの選択	テンプレートボタンを選択し、ドロップダウンリストから設定済みのテンプレートを選択します。具体的な設定については、トリガー条件テンプレートの設定をご参照ください。新規作成したテンプレートが表示されない場合は、右側の**更新**をクリックすると、トリガーするアラートテンプレートの選択リストが更新されます。
アラート通知の設定	アラート通知	システムプリセット通知テンプレートとユーザーカスタム通知テンプレートの選択がサポートされます。各アラートポリシーは、最大で3つの通知テンプレートにのみバインドできます。詳細については、通知テンプレートをご参照ください
高度な設定	自動スケーリング	有効にして正常に設定されると、アラート条件に達した場合、自動スケーリングポリシーがトリガーされ、容量が縮小または拡張されます

4. 上記の情報を設定し、**保存**をクリックすれば、アラートポリシーの作成は完了です。

説明：

CVMアラートで正常にアラートを送信するには、CVMインスタンスの監視エージェントのインストールを行って監視指標データを報告する必要があります。クラウド製品監視ページで、監視agentをインストールしていないCVMを確認し、IPリストをダウンロードすることができます。

アラート指標の説明

最終更新日：：2024-01-04 18:36:26

アラートの説明

注目するインスタンス指標についてアラートを作成することで、CLBインスタンスが実行状態においてある条件に達した際、関心のあるユーザーグループに対し速やかにアラート情報を送信することができます。これにより、異常な状態を速やかに発見してそれに応じた措置を確実にとることができ、システムの安定性と信頼性を維持することができます。CLBのアラートポリシーには次のタイプがあります。
パブリックネットワークリスナー
プライベートネットワークリスナー
サーバーポート（その他）
リスナーディメンションです。
サーバーポートディメンション
サーバーポート（従来型プライベートネットワーク）
レイヤー7プロトコル監視

パブリックネットワークリスナー/プライベートネットワークリスナー

現在、パブリックネットワークCLBとプライベートネットワークCLBはいずれもリスナーディメンションでのアラートをサポートしています。具体的な指標は次のとおりです。

指標	単位	説明
インバウンド帯域幅	Mbps	統計周期内で、クライアントがパブリックネットワーク経由でCLBにアクセスする際に使用した帯域幅です。
アウトバウンド帯域幅	Mbps	統計周期内で、CLBがパブリックネットワークにアクセスする際に使用した帯域幅です。
インバウンドパケット数	個/s	統計周期内で、CLBが1秒間に受信したリクエストデータパケット数です。
アウトバウンドパケット数	個/s	統計周期内で、CLBが1秒間に送信したデータパケット数です。

サーバーポート（その他）

従来型のプライベートネットワークCLBを除くすべてのCLBは、次の2つのディメンションのアラートをサポートしています。

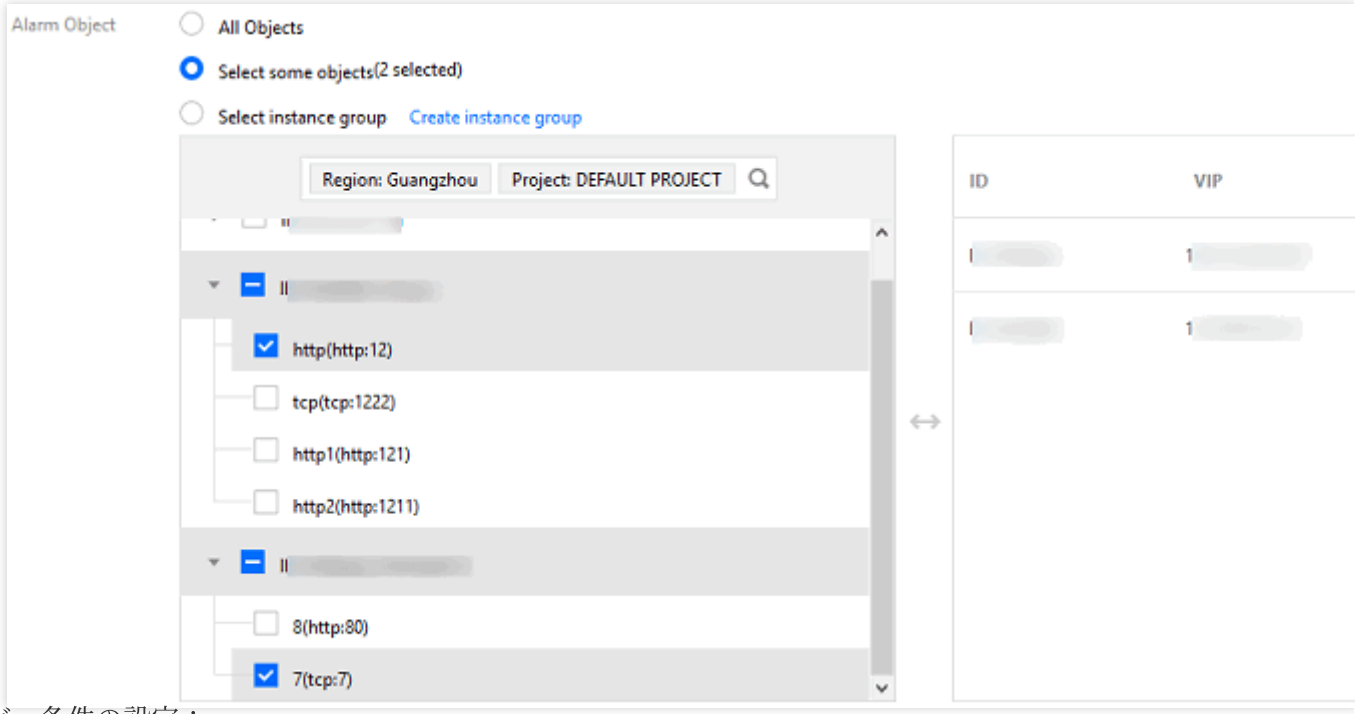
1. リスナーディメンション

あるリスナーのバックエンドサーバーの異常ポート数を設定し、そのリスナーにバインドしたすべてのサーバーポートの異常を統計することで、設定した閾値に基づいてアラートを発出できます。下の図の設定は、選択したリスナー下のすべてのバックエンドサーバーの異常ポート数を1分に1回収集し、異常ポート数が2回連続して10個/秒を超えるとアラートをトリガーし、かつ1日1回警告することを表します。

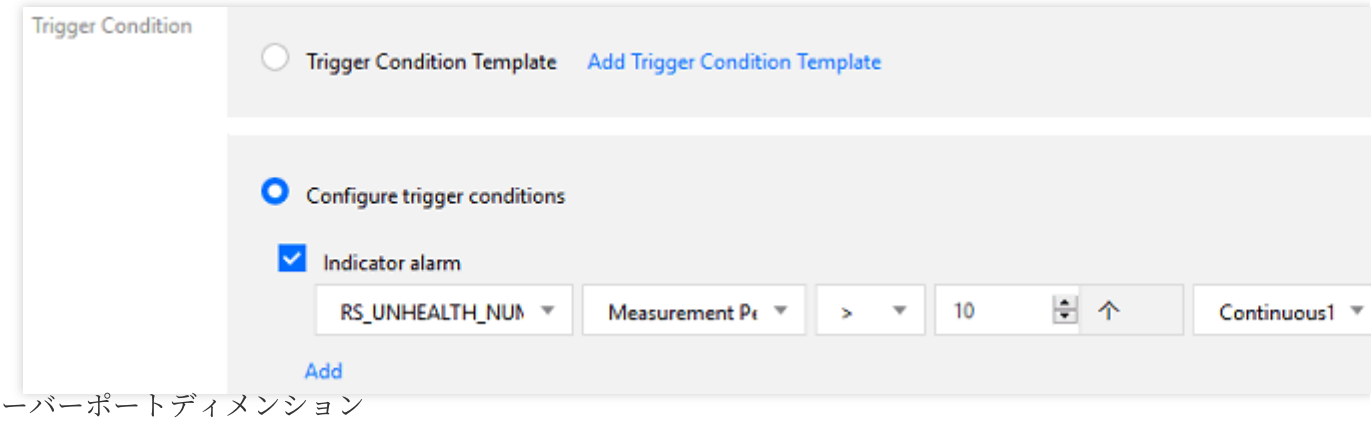
説明：

リスナーディメンションのアラートをご希望の場合は、チケット申請を提出してください。

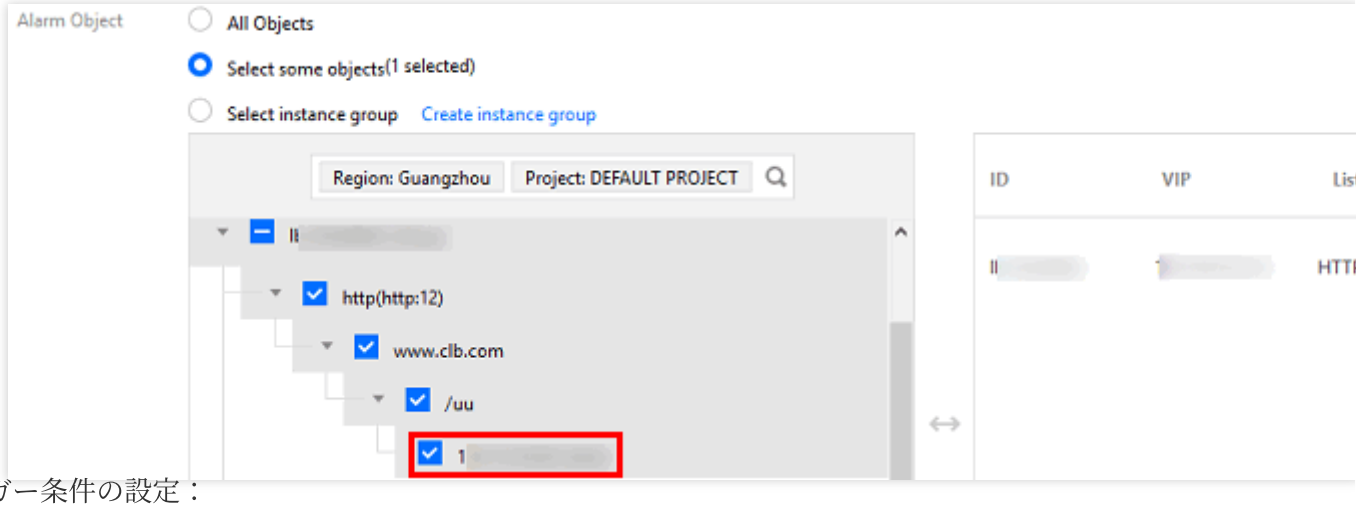
アラートオブジェクトの設定：



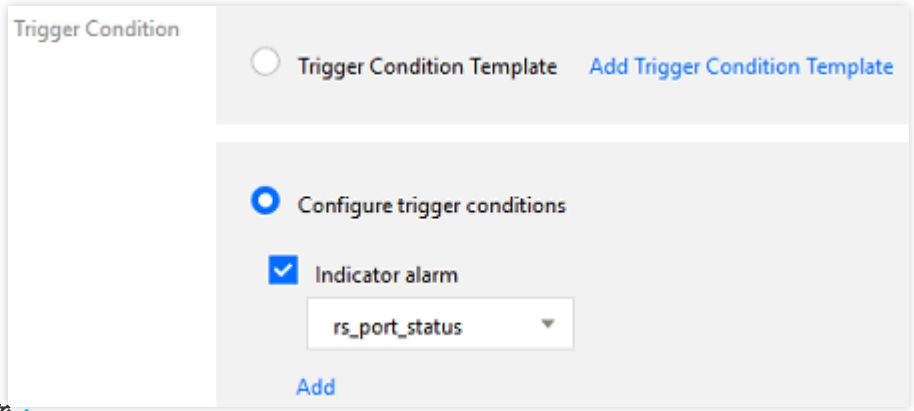
トリガー条件の設定：



2. サーバーポートディメンション
あるリスナーにバインドされたあるバックエンドサーバーのあるポートの異常アラートを設定し、そのポートに異常があればアラートを送信することができます。
アラートオブジェクトの設定：



トリガー条件の設定：



ご注意：
バックエンドサーバーポートの異常とは、バックエンドサーバーのそのポートが使用できなくなったことをCLBが検知したことを意味します。ポート異常はわずかなネットワークジッターによってもトリガーされる場合があります。

リスナーディメンションの統計には、そのリスナー下のすべてのバックエンドサービスポートのステータスが含まれ、単一のアラートを閾値アラートに集約します。ネットワークジッターの影響を低減するため、リスナーディメンションのアラートを使用することをお勧めします。

サーバーポート（従来型プライベートネットワーク）

従来型プライベートネットワークCLBではサーバーポート異常アラートを設定することができます。具体的な設定は「サーバーポート（その他）-サーバーポートディメンション」の設定と同様です。
あるリスナーにバインドされたあるバックエンドサーバーのあるポートの異常アラートを設定し、そのポートに異常があればアラートを送信することができます。

レイヤー7プロトコル監視

すべてのレイヤー7リスナー（HTTP/HTTPS）について、レイヤー7独自の監視指標を含むアラートポリシーを設定できます。具体的な指標は次のとおりです。

指標	単位	説明
インバウンド帯域幅	Mbps	統計周期内で、クライアントがパブリックネットワーク経由でCLBにアクセスする際に使用した帯域幅です。
アウトバウンド帯域幅	Mbps	統計周期内で、CLBがパブリックネットワークにアクセスする際に使用した帯域幅です。
インバウンドパケット数	個/s	統計周期内で、CLBが1秒間に受信したリクエストデータパケット数です。
アウトバウンドパケット数	個/s	統計周期内で、CLBが1秒間に送信したデータパケット数です。
新規接続数	個	統計周期内における1分間の新規接続の個数です。
アクティブ接続数	個	統計周期内における1分間のアクティブ接続の個数です。
平均応答時間	ms	統計周期内におけるCLBの平均応答時間です。
最大応答時間	ms	統計周期内におけるCLBの最大応答時間です。
2xxステータスコード	個	統計周期内で、バックエンドサーバーが返した2xxステータスコードの数です。
3xxステータスコード	個	統計周期内で、バックエンドサーバーが返した3xxステータスコードの数です。

4xxステータスコード	個	統計周期内で、バックエンドサーバーが返した4xxステータスコードの数です。
5xxステータスコード	個	統計周期内で、バックエンドサーバーが返した5xxステータスコードの数です。
404ステータスコード	個	統計周期内で、バックエンドサーバーが返した404ステータスコードの数です。
502ステータスコード	個	統計周期内で、バックエンドサーバーが返した502ステータスコードの数です。
CLBが返した3xxステータスコード	個	統計周期内で、CLBが返した3xxステータスコードの数です。
CLBが返した4xxステータスコード	個	統計周期内で、CLBが返した4xxステータスコードの数です。
CLBが返した5xxステータスコード	個	統計周期内で、CLBが返した5xxステータスコードの数です。
CLBが返した404ステータスコード	個	統計周期内で、CLBが返した404ステータスコードの数です。
CLBが返した502ステータスコード	個	統計周期内で、CLBが返した502ステータスコードの数です。

Cloud Access Management
概要

最終更新日：：2024-01-04 18:36:26

Cloud Load Balancer（CLB）、CVM、TencentDBなどのサービスを使用する場合、これらのサービスは管理者がそれぞれ異なりますが、いずれの管理者もクラウドアカウントキーを共有するため、次の問題が存在します。
キーが複数の人に共有されるため、機密漏洩リスクが高くなります。
他の人のアクセス権限を制限することはできませんので、誤操作によりセキュリティリスクが発生する可能性があります。
Cloud Access Management（CAM）は、Tencent Cloudアカウント下のリソースへのアクセス権限の管理に用いられます。CAMを使用することで、ID管理とポリシー管理によって、どのサブアカウントにどのリソースの操作権限を与えるかを制御することができます。
例えば、アカウント下に複数のCLBインスタンスがあり、異なるプロジェクトにデプロイされている場合、権限制御を強化し、リソースの権限承認を行うため、プロジェクトAの管理者に権限承認ポリシーをバインドすることができます。このポリシーでは、この管理者だけがプロジェクトA下のCLBリソースを操作できるように規定します。
サブアカウントのCLB関連リソースへのアクセス管理を行う必要がない場合は、このセクションをスキップできます。この部分をスキップしても、ドキュメントのそれ以外の内容の理解と利用には影響しません。

CAMの基本概念

ルートアカウントはサブアカウントにポリシーをバインドすることで権限承認を行います。ポリシーの設定は、**[API、リソース、ユーザー/ユーザーグループ、許可/拒否、条件]**の次元まで精密に行うことができます。
1. アカウント
ルートアカウント
Tencent Cloudのリソースの帰属先であり、リソース使用量の計算と課金における基本主体です。Tencent Cloudサービスにログインできます。
サブアカウント
ルートアカウントが作成するアカウントであり、確実なIDおよびIDクレデンシャルを有し、なおかつTencent Cloudコンソールにログインできます。ルートアカウントは複数のサブアカウント(ユーザー)を作成することができます。**サブアカウントはデフォルトではリソースを所有せず、所属するルートアカウントによる権限承認を受ける必要があります。**
ID証明書
ログイン証明書とアクセス証明書の2種類があります。**ログイン証明書**はユーザーのログイン名とパスワードを指し、**アクセス証明書**はTencent Cloud APIのキー（SecretIdおよびSecretKey）を指します。

2. リソースと権限

リソース

リソースとは、クラウドサービスにおいて操作の対象となるものであり、例えばCVMインスタンス、VPCインスタンスなどがあります。

権限

権限とは、ある何人かのユーザーに対し、あるいくつかの操作の実行を許可または拒否することを指します。デフォルトでは、**ルートアカウントはその名前の下にあるすべてのリソースへのアクセス権限を有する**一方、**サブアカウントにはルートアカウント下の何らかのリソースへのアクセス権限がありません**。

ポリシー

ポリシーは、1つまたは複数の権限を定義および説明する構文仕様です。**ルートアカウント**はユーザー/ユーザーグループに**ポリシーをバインドする**ことによって権限承認を行います。

その他の関連情報については、CAMの概要をご参照ください。

関連ドキュメント

ターゲット	リンク
ポリシーとユーザー間の関係を理解する	ポリシー管理
ポリシーの基本構造を理解する	ポリシー構文
CAMをサポートしている他の製品について理解する	CAMをサポートしているクラウドサービスのリスト

権限承認の定義

最終更新日：：2024-01-04 18:36:26

CAMで権限承認が可能なCLBのリソースタイプ

リソースタイプ	承認ポリシーにおけるリソースの記述メソッド
CLBインスタンス	`qcs::clb:$region::clb/$loadbalancerid`
CLBバックエンドサーバー	`qcs::cvm:$region:$account:instance/$cvminstanceid`

そのうち：
- `$region` はすべて、あるregionのIDとしなければなりません。空にすることができます。
`$account` はすべて、リソース所有者のAccountIdとするか、または「*」としなければなりません。
`$loadbalancerid` はすべて、あるloadbalancerのIDとするか、または「*」としなければなりません。
以下同様とします。

CAMでCLBの権限承認を行うことができるインターフェース

CAMではCLBリソースに対し、次のActionの権限承認を行うことができます。

インスタンス関連

APIの操作	リソース説明	インターフェースの説明
DescribeLoadBalancers	CLBインスタンスリストの照会	`*` インターフェースにのみ認証を行います
CreateLoadBalancer	CLBの購入	`qcs:$projectid:clb:$region:$account:clb/*`

DeleteLoadBalancers	CLBの削除	`qcs::clb:$region:$account:clb/$loadbalancerid`
ModifyLoadBalancerAttributes	CLB属性情報の変更	`qcs::clb:$region:$account:clb/$loadbalancerid`
ModifyForwardLBName	CLBの名前の変更	`qcs::clb:$region:$account:clb/$loadbalancerid`
SetLoadBalancerSecurityGroups	CLBインスタンスのセキュリティグループの設定	`qcs::clb:$region:$account:clb/$loadbalancerid`

リスナー関連

APIの操作	リソース説明	インターフェースの説明
DeleteLoadBalancerListeners	CLBリスナーの削除	`qcs::clb:$region:$account:clb/$loadbalance`
DescribeLoadBalancerListeners	CLBリスナーリス	`qcs::clb:$region:$account:clb/$loadbalance`

	トの取得	
ModifyLoadBalancerListener	CLBリスナー属性の変更	`qcs::clb:$region:$account:clb/$loadbalance`
CreateLoadBalancerListeners	CLBリスナーの作成	`qcs::clb:$region:$account:clb/$loadbalance`
DeleteForwardLBListener	CLBリスナーの削除（レイヤー4およびレイヤー7）	`qcs::clb:$region:$account:clb/$loadbalance`
ModifyForwardLBSeventhListener	CLBレイヤー7リスナーの属性の変更	`qcs::clb:$region:$account:clb/$loadbalance`
ModifyForwardLBFourthListener	CLBレイヤー4リスナー	`qcs::clb:$region:$account:clb/$loadbalance`

	属性の変更	
DescribeForwardLBListeners	CLBリスナーリストの照会	`qcs::clb:$region:$account:clb/$loadbalance`
CreateForwardLBSeventhLayerListeners	レイヤー7CLBリスナーの作成	`qcs::clb:$region:$account:clb/$loadbalance`
CreateForwardLBFourthLayerListeners	レイヤー4CLBリスナーの作成	`qcs::clb:$region:$account:clb/$loadbalance`

CLBドメイン名 + URL関連

APIの操作	リソース説明	インターフェースの説明
ModifyForwardLBRulesDomain	CLBリスナー転送ルールのドメイン名の変更	`qcs::clb:$region:$account:clb/$loadbalancerid`
CreateForwardLBListenerRules	CLBリスナー転送ルールの作成	`qcs::clb:$region:$account:clb/$loadbalancerid`

DeleteForwardLBListenerRules	レイヤー7CLBリスナールールの削除	`qcs::clb:$region:$account:clb/$loadbalancerid`
DeleteRewrite	CLB転送ルール間のリダイレクト関係の削除	`qcs::clb:$region:$account:clb/$loadbalancerid`
ManualRewrite	CLB転送ルールのリダイレクト関係の手動追加	`qcs::clb:$region:$account:clb/$loadbalancerid`
AutoRewrite	CLB転送ルールのリダイレクト関係の自動生成	`qcs::clb:$region:$account:clb/$loadbalancerid`

バックエンドサーバー関連

APIの操作	リソース説明	インターフェースの説明
ModifyLoadBalancerBackends	CLBバックエンドサーバーの重みの変更	`qcs::clb:$region:$account:clb/$loa`
DescribeLoadBalancerBackends	CLBが	`qcs::clb:$region:$account:clb/$loa`

	バインドするバックエンドサーバーリストの取得	
DeregisterInstancesFromLoadBalancer	バックエンドサーバーのバインド解除	`qcs::clb:$region:$account:clb/$loa`
RegisterInstancesWithLoadBalancer	バックエンドサーバーのCLBへのバインド	`qcs::clb:$region:$account:clb/$loa`
DescribeLBHealthStatus	CLBヘルスステータスの照会	`qcs::clb:$region:$account:clb/$loa`
ModifyForwardFourthBackendsPort	レイヤー4リスナー転送ルールのCVMのポートの変更	`qcs::clb:$region:$account:clb/$loa`
ModifyForwardFourthBackendsWeight	レイヤー4リスナー転送ルー	`qcs::clb:$region:$account:clb/$loa`

	ルのCVMの重みの変更	
RegisterInstancesWithForwardLBSeventhListener	CVMをCLBレイヤー7リスナーの転送ルールにバインドする	`qcs::clb:$region:$account:clb/$loa`
RegisterInstancesWithForwardLBFourthListener	CVMをCLBレイヤー4リスナーの転送ルールにバインドする	`qcs::clb:$region:$account:clb/$loa`
DeregisterInstancesFromForwardLBFourthListener	CLBレイヤー4リスナー転送ルールのCVMをバインド解除する	`qcs::clb:$region:$account:clb/$loa`
DeregisterInstancesFromForwardLB	CLBレイヤー7リスナー転送ルールの	`qcs::clb:$region:$account:clb/$loa`

	CVMをバインド解除する	
ModifyForwardSeventhBackends	レイヤー7リスナー転送ルールのCVMの重みを変更する	`qcs::clb:$region:$account:clb/$loa`
ModifyForwardSeventhBackendsPort	レイヤー7リスナー転送ルールのCVMのポートを変更する	`qcs::clb:$region:$account:clb/$loa`
DescribeForwardLBBackends	CLBのCVMリストの照会	`qcs::clb:$region:$account:clb/$loa`
DescribeForwardLBHealthStatus	CLBヘルスチェックステータスの照会	`qcs::clb:$region:$account:clb/*`
ModifyLoadBalancerRulesProbe	CLBリスナー転送ルールのヘルス	`qcs::clb:$region:$account:clb/$loa`

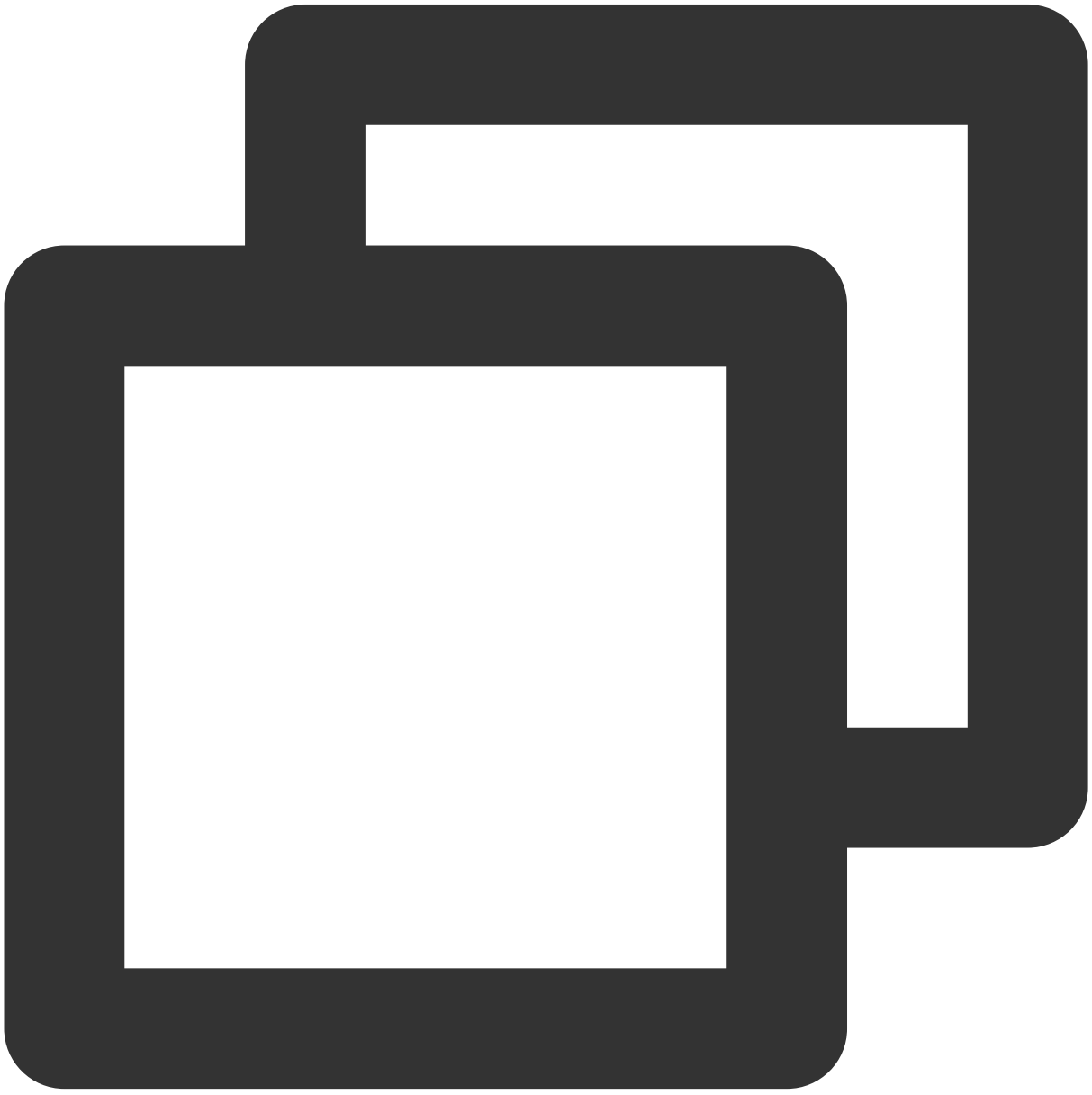
	チェックおよび転送パスの変更	

ポリシーの例

最終更新日：：2024-01-04 18:36:26

すべてのCLBの全読み取り書き込みポリシー

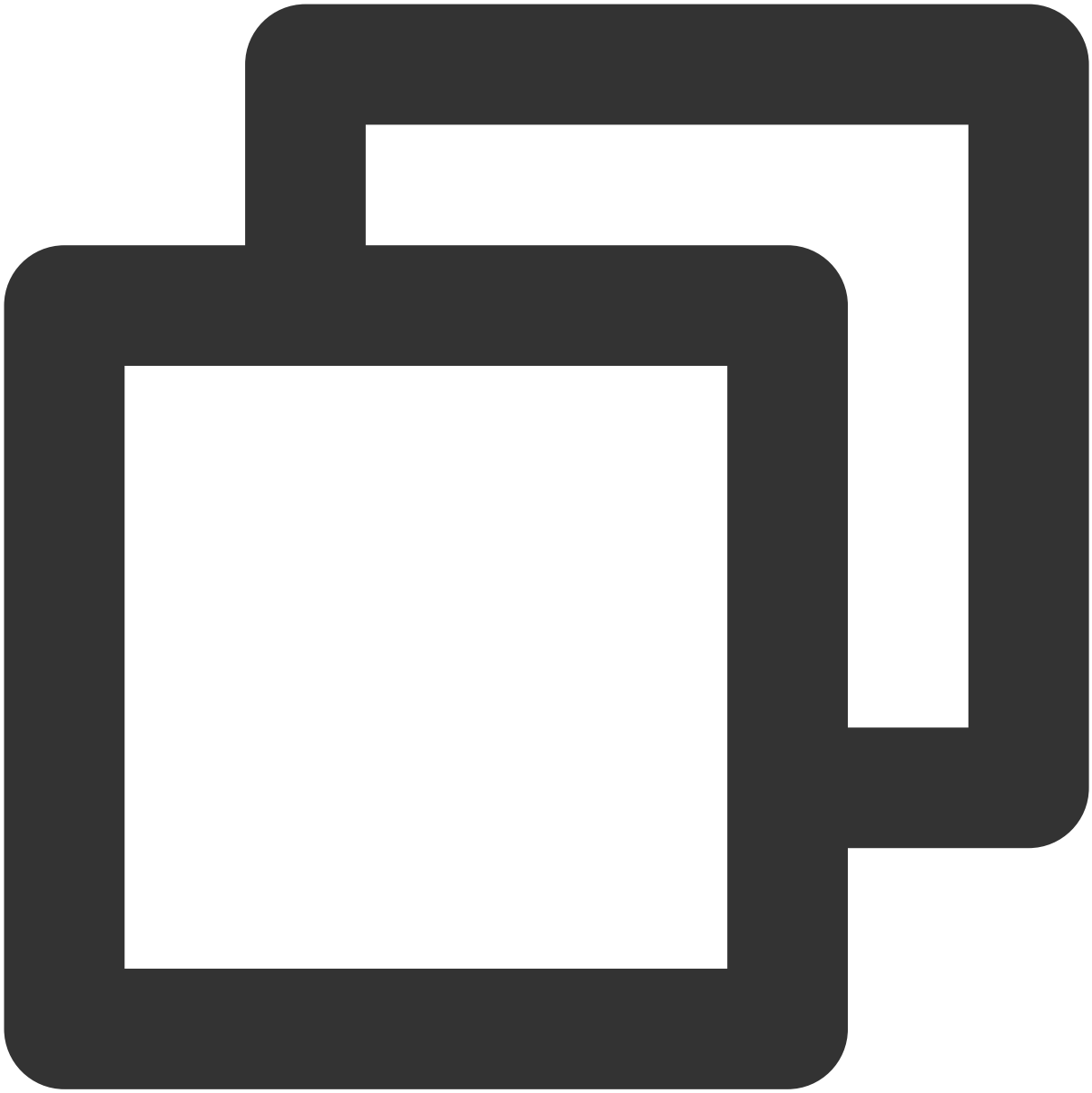
サブアカウントにCLBサービスの完全な管理権限（作成、管理などの全操作）を承認します。
ポリシー名：CLBResourceFullAccess



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

すべてのCLBの読み取り専用ポリシー

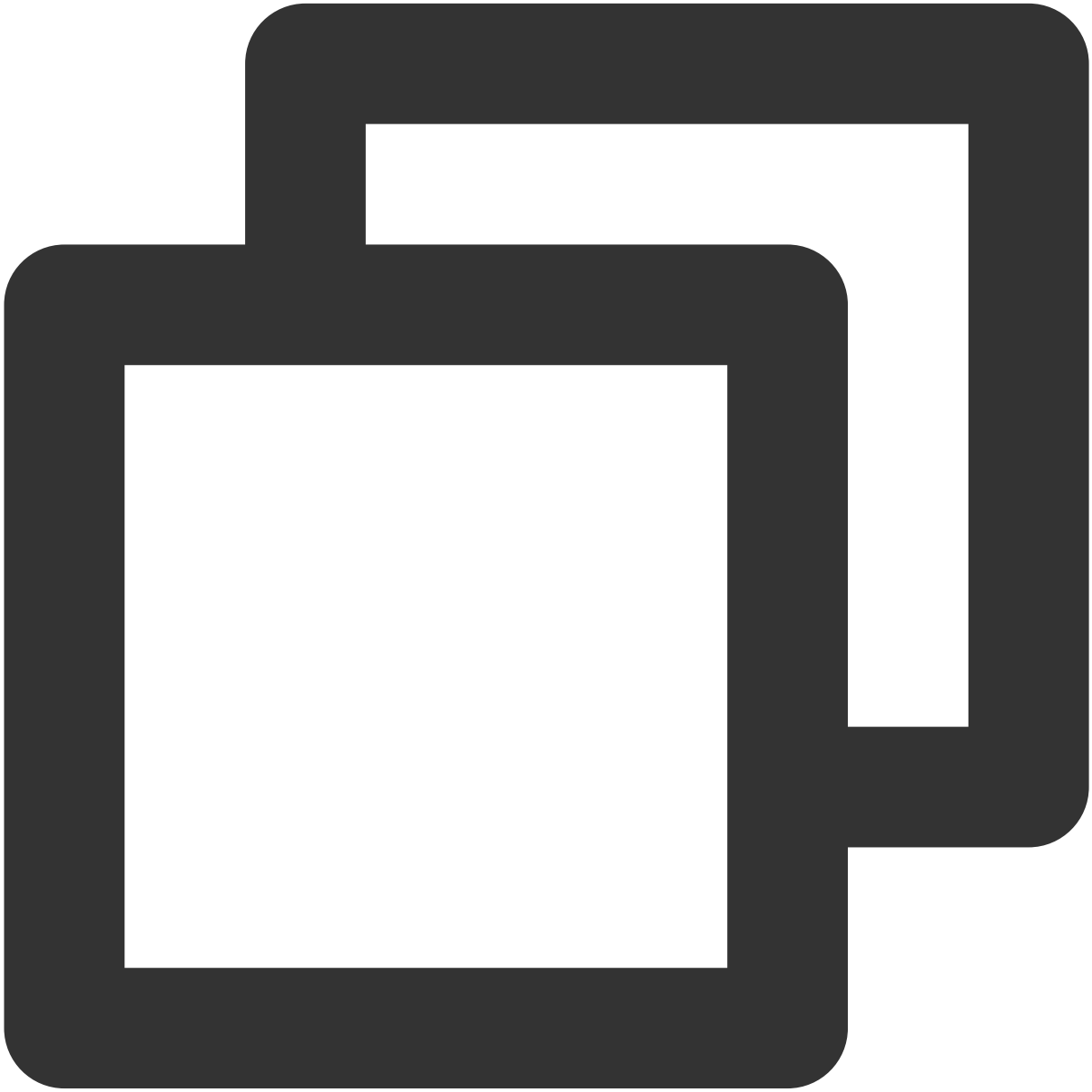
サブアカウントに、CLBに読み取り専用でアクセスできる権限（すなわち、すべてのCLB下のすべてのリソースを見ることができる権限）を承認します。ただしサブアカウントはそれらの作成、更新または削除を行うことはできません。コンソールでのリソース操作の前提は、そのリソースを見ることができることであるため、サブアカウントのCLB全読み取り権限をアクティブ化することをお勧めします。
ポリシー名： CLBResourceReadOnlyAccess



```
{
  "version": "2.0",
  "statement": [{
      "action": [
          "name/clb:Describe*"
      ],
      "resource": "*",
      "effect": "allow"
  }]
}
```

あるタグ下のCLBの全読み取り書き込みポリシー

サブアカウントに、あるタグ（タグキーはtagkey、タグ値はtagvalue）下のCLBの完全な管理権限（インスタンス管理、リスナー管理などの全操作）を承認します。
CLBインスタンスはタグ設定およびタグ使用認証をサポートしています。



```
{
    "version":"2.0",
    "statement":[
        {
```

```
            "effect":"allow",
            "action":"*",
            "resource":"*",
            "condition":{
                "for_any_value:string_equal":{
                    "qcs:tag":[
                        "tagkey&tagvalue"
                    ]
                }
            }
        }
    ]
}
```

従来型CLB
従来型CLBの概要

最終更新日：：2024-01-04 18:36:26

概要

従来型CLBの設定はシンプルで、簡単なCLBシーンにおいてサポートをしています。
従来型**パブリックネットワーク**CLB：TCP/UDP/HTTP/HTTPSプロトコルをサポートしています。
従来型**プライベートネットワーク**CLB：TCP/UDPプロトコルをサポートしています。
CLBの2種類のインスタンスタイプ：CLB（これまで「アプリケーション型CLB」とも呼ばれていたもの）と従来型CLBがあります。
CLBは、従来型CLBのすべての機能をカバーしています。製品の機能、製品の性能などのあらゆる面から考えても、使用するインスタンスタイプにCLBをお勧めします。両者の詳細な比較については、インスタンスタイプをご参照ください。
ご注意：
現在、Tencent Cloudアカウントには標準アカウントタイプと従来型アカウントタイプがあり、2020年6月17日0時以降に登録したアカウントはすべて標準アカウントタイプとなります。この時点より前に登録したアカウントについては、コンソールでアカウントタイプを確認してください。具体的な操作については、アカウントタイプの判断をご参照ください。標準アカウントタイプによる従来型CLBのサポートはなくなるため、購入するインスタンスはすべてCLBとなります。
ここでは、従来型CLBインスタンスを紹介します。インスタンスを作成後、インスタンスにリスナーを設定する必要があります。リスナーはCLBインスタンス上のリクエストを監視し、バランシングポリシーに基づいてトラフィックをバックエンドサーバーに振り分ける役割を担います。

リスナーの設定の説明

CLBリスナーには次の設定が必要です。
1. リスニングプロトコルおよびリスニングポートについて、CLBのリスニングポートはフロントエンドポートとも呼ばれ、リクエストを受信してバックエンドサーバーにリクエストを転送するためのポートとして用いられます。
2. バックエンドポートは、CVMがサービスを提供するためのポートで、CLBからのトラフィックを受信して処理します。
3. リスナーポリシーには、バランシングポリシー、セッション維持などがあります。
4. ヘルスチェックポリシーです。

5. バックエンドサービスをバインドし、バックエンドサーバーのIPを選択します。

説明：

従来型CLBにおいて、複数のリスナーを設定した場合、複数のバックエンドCVMがバインドされるため、各リスナーは、その設定に従ってすべてのバックエンドサーバーに転送を行います。

サポートするプロトコルタイプ

CLBリスナーは、CLBインスタンス上のレイヤー4およびレイヤー7リクエストを監視し、これらのリクエストをバックエンドサーバーに振り分けることができ、その後バックエンドサーバーがリクエストを処理します。レイヤー4およびレイヤー7CLBの主な違いは、ユーザーのリクエストに対しロードバランシングを行う際に、トラフィックの転送をレイヤー4プロトコルとレイヤー7プロトコルのどちらに基づいて行うかという点にあります。

レイヤー4プロトコル：トランスポート層プロトコルで、TCPおよびUDPが含まれます。

レイヤー7プロトコル：アプリケーション層プロトコルで、HTTPおよびHTTPSが含まれます。

説明：

1. 従来型CLBは、主にVIP + Portによってリクエストを受信し、トラフィックをバックエンドサーバーに分配します。レイヤー7プロトコルは、ドメイン名およびURLパスベースの転送をサポートしていません。
2. 従来型プライベートネットワークCLBは、レイヤー4プロトコルのみをサポートしており、レイヤー7プロトコルはサポートしていません。
3. 上記の高度な機能のサポートが必要な場合は、CLBを直接使用してください。従来型CLBではない場合の詳細については、インスタンスタイプをご参照ください。

ポートの設定

リスニングポート（フロントエンドポート）	サービスポート（バックエンドポート）	説明
CLBがサービスを提供する際に、リクエストを受信してバックエンドサーバーにリクエストを転送するポートです。 ユーザーは、1～65535番ポートに、21（FTP）、25（SMTP）、80（HTTP）、443（HTTPS）などのCLBを設定できます。	CLBがサービスを提供する際に、リクエストを受信してバックエンドサーバーにリクエストを転送するポートです。 ユーザーは、1～65535番ポートに、21（FTP）、25（SMTP）、80（HTTP）、443（HTTPS）などのCLBを設定できます。	同一のCLBインスタンス内ではリスニングポートの重複はできません。 例えば、リスナーのTCP:80とリスナーのHTTP:80を同時に作成することはできません。 TCPとUDPプロトコルのポートのみが重複できます。 例えば、リスナーのTCP:80とリスナーのUDP:80であれば、同時に作成することができます。 サービスポートは、同じCLBインスタンス内で重複することができます。 例えば、リスナーの HTTP:80とリスナーのHTTPS:443は、同じ

		CVMの同じポートを同時にバインドすることができます。

従来型CLBの設定

最終更新日：：2024-01-04 18:36:26

従来型CLBインスタンスを作成後、インスタンスにリスナーを設定する必要があります。リスナーはCLBインスタンス上のリクエストを監視し、バランシングポリシーに基づいてトラフィックをバックエンドサーバーに振り分ける役割を担います。

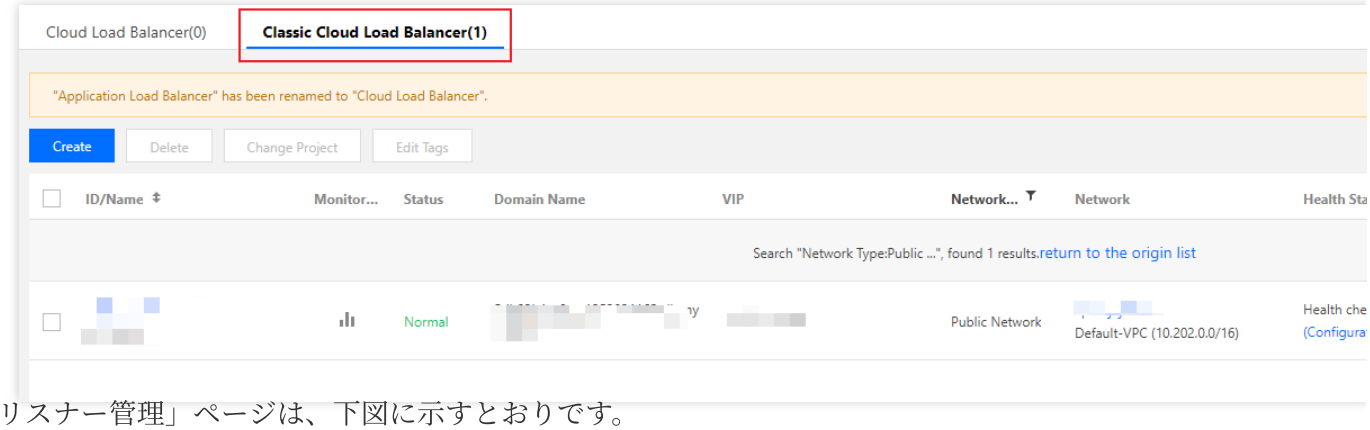
前提条件

CLBインスタンスの作成が必要です。そのうち、インスタンスタイプでは「従来型CLB」を選択します。
ご注意：
現在、Tencent Cloudアカウントは、標準アカウントタイプと従来型アカウントタイプがあり、2020年6月17日0時以降に登録したアカウントは、すべて標準アカウントタイプとなります。この時点より前に登録したアカウントは、コンソールでアカウントタイプを確認してください。具体的な操作については、アカウントタイプの判断をご参照ください。標準アカウントタイプによる従来型CLBのサポートはなくなるため、購入するインスタンスはすべてCLBとなります。

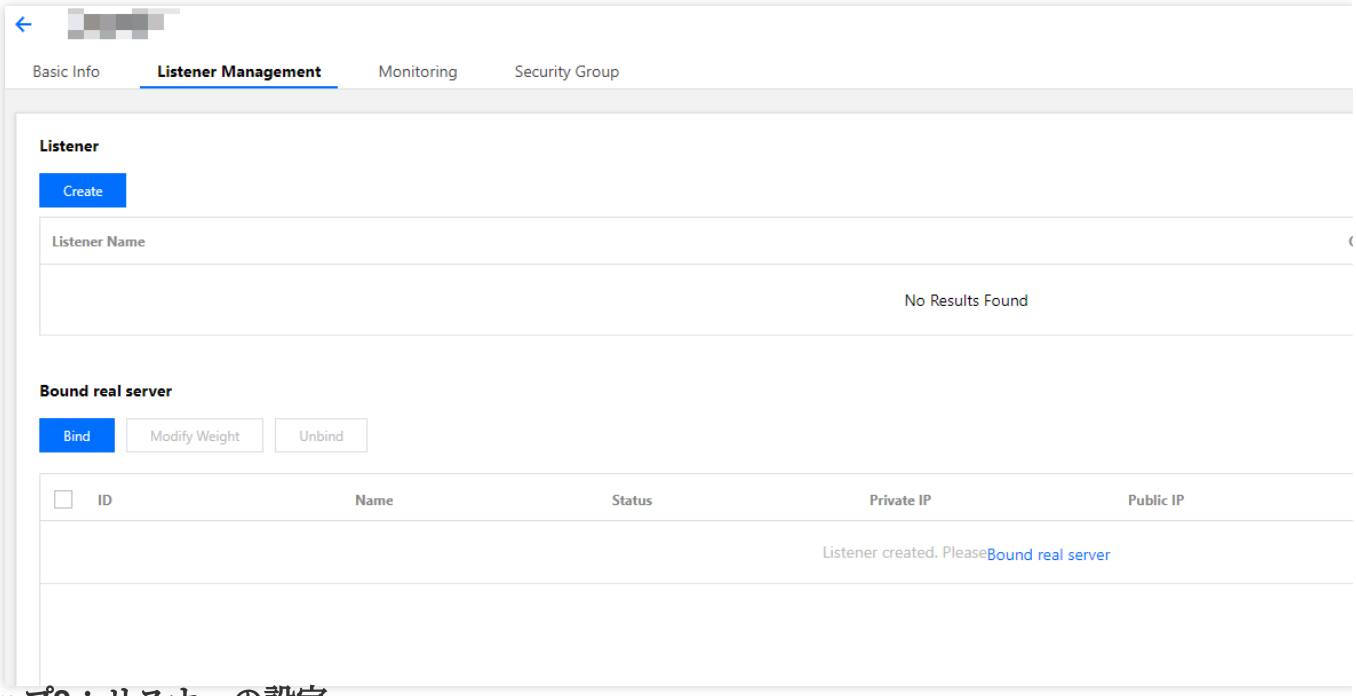
リスナーの設定

ステップ1：リスナー管理ページを開きます

1. CLBコンソールにログインします。
2. 左側のナビゲーションバーで、**インスタンス管理**を選択します。
3. インスタンスリストページで設定が必要なインスタンスIDをクリックし、インスタンス詳細ページに進みます。
4. **リスナー管理**タブをクリックします。また、リストページの操作バーで**リスナーの設定**をクリックすることもできます。



5. 「リスナー管理」ページは、下図に示すとおりです。



ステップ2：リスナーの設定

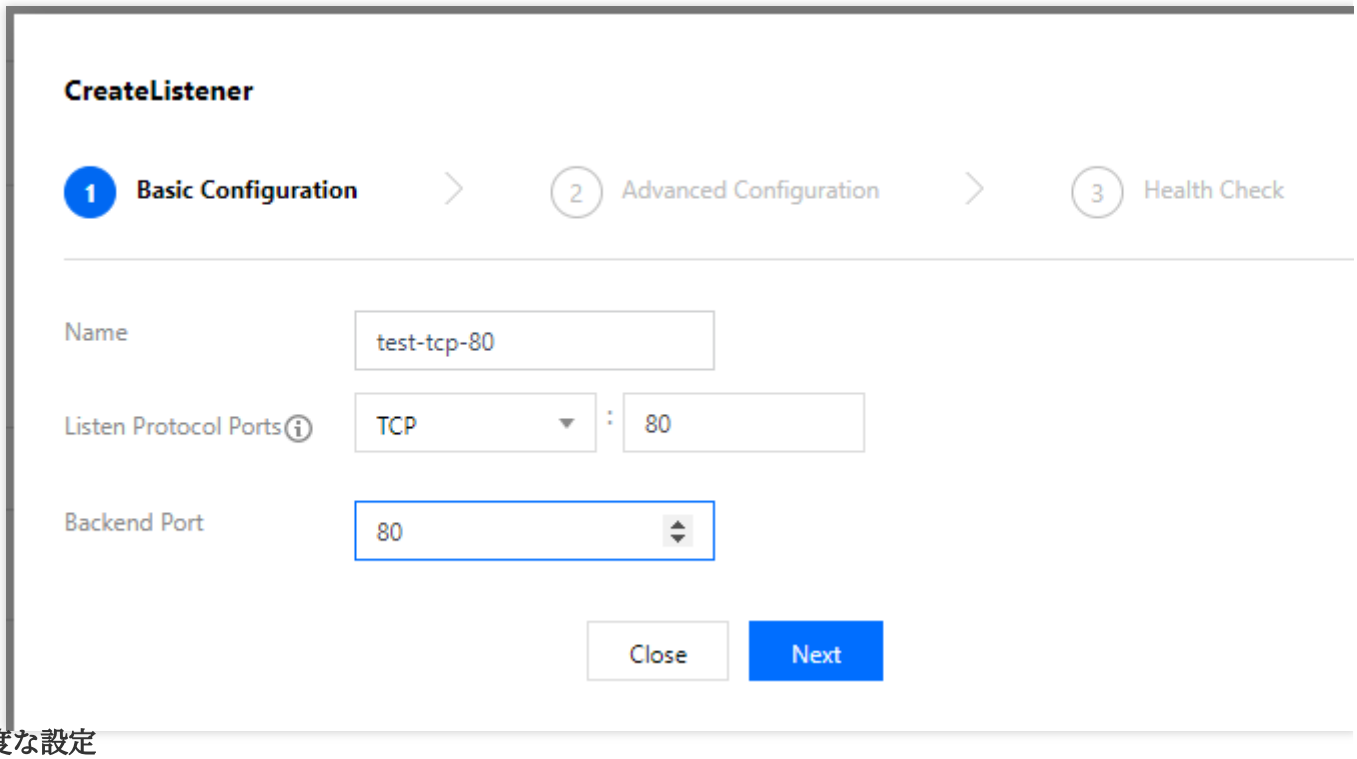
「リスナー」モジュールで、**新規作成**をクリックし、ポップアップボックスでTCPリスナーを設定します。

1. **基本設定**

リスナーの基本設定	説明	事例
名前	リスナーの名前	test-tcp-80
リスニングプロトコルポート	リスナーのプロトコルおよびリスニングポート リスニングプロトコル：CLBがサポートしているプロトコルには、TCP、UDP、HTTP、HTTPSが含まれており、この例では、TCPを選択しています。 リスニングポート：リクエストを受信してバックエンドサーバーにリクエストを転送するために使用するポートで、ポート範囲は1～65535です。 同一CLBインスタンス内で、リスニングポートは重複できません。	TCP:80

バックエンドポート	CVMが提供するサービスのポートは、CLBからのトラフィックを受信して処理します。	80

TCPリスナーの作成における具体的な基本設定は、下図に示すとおりです。

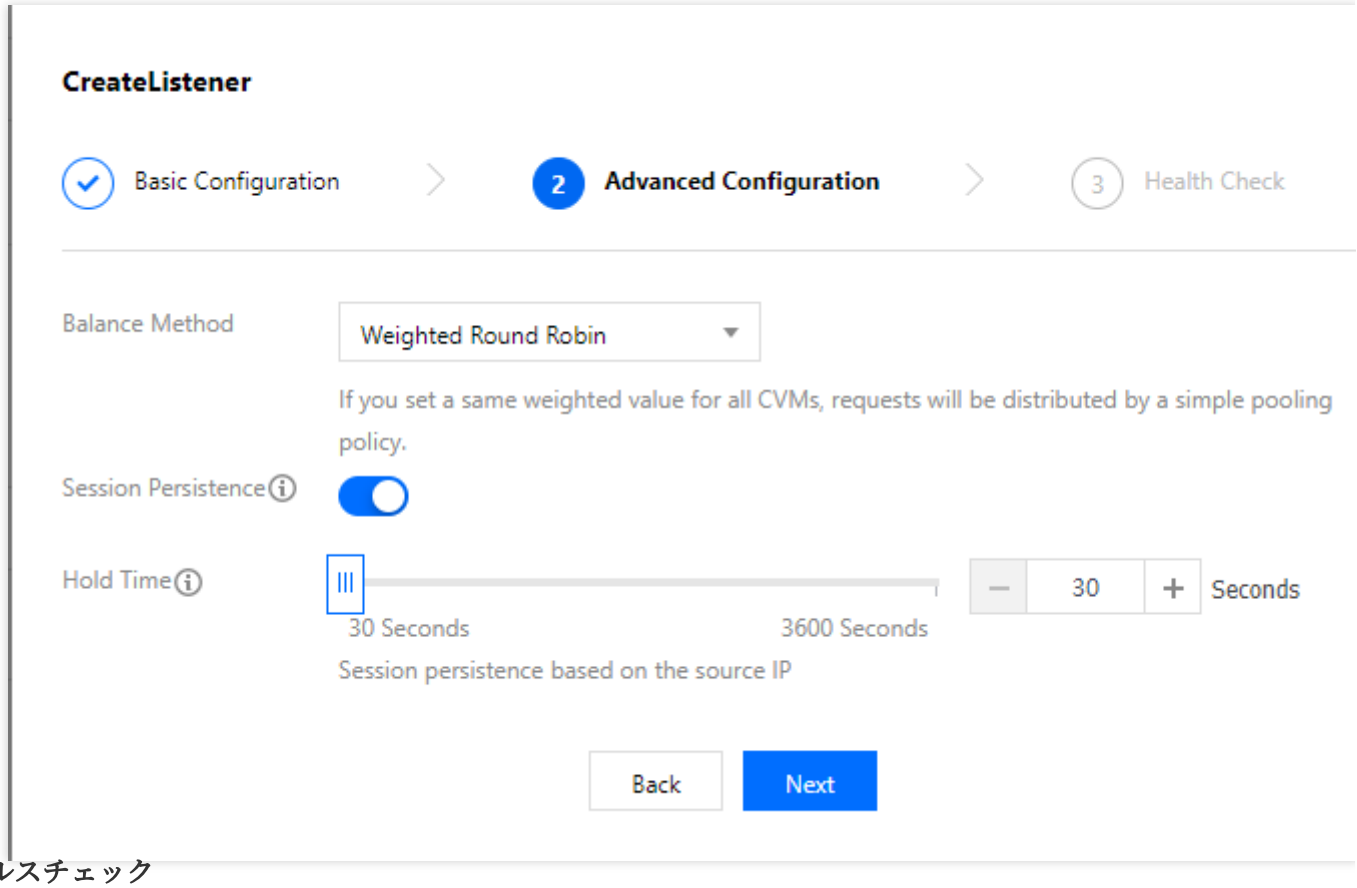


2. **高度な設定**

高度な設定	説明	事例
バランシング方式	TCPリスナーでは、CLBは重み付けラウンドロビン（WRR）および重み付け最小接続（WLC）の2種類のスケジューリングアルゴリズムをサポートしています 重み付けラウンドロビンアルゴリズム：バックエンドサーバーの重みに基づき、順番にリクエストを異なるサーバーに配信します。重み付けラウンドロビンアルゴリズムは**新規接続数**に基づいてスケジューリングし、重みの高いサーバーがラウンドロビンされる回数（確率）が高くなるほど、同じ重みのサーバーは同じ数の接続数を処理します。 重み付け最小接続：サーバーの現在アクティブな接続数に基づいてサーバーの負荷状況を推定します。重み付け最小接続はサーバー負荷および重みに基づいて総合的にスケジューリングし、重み値が同じ場合、現在の接続数が少ないバックエンドサーバーほどラウンドロビンされる回数（確率）も高くなります。	重み付けラウンドロビン
セッション維持のステータス	セッションの維持をオンまたはオフにします セッションの維持を有効化すると、CLBリスナーは同一クライアントからのアクセスリクエストを同一のバックエンドサーバーに配信します。 TCPプロトコルはクライアントIPアドレスのセッションの維持に基づき、同一IPアドレスからのアクセスリクエストを同一のバックエンドサーバーに転送します。	オン

	重み付けラウンドロビンスケジューリングはセッションの維持をサポートします。重み付け最小接続スケジューリングはセッション維持機能の有効化をサポートしていません。	
セッションの維持時間	セッションの維持時間 維持時間を超え、接続中に新たなリクエストがない場合は、自動的にセッションの維持が切断されます。 設定可能範囲は30～3600秒です。	30s

具体的な設定については、下図に示すとおりです。

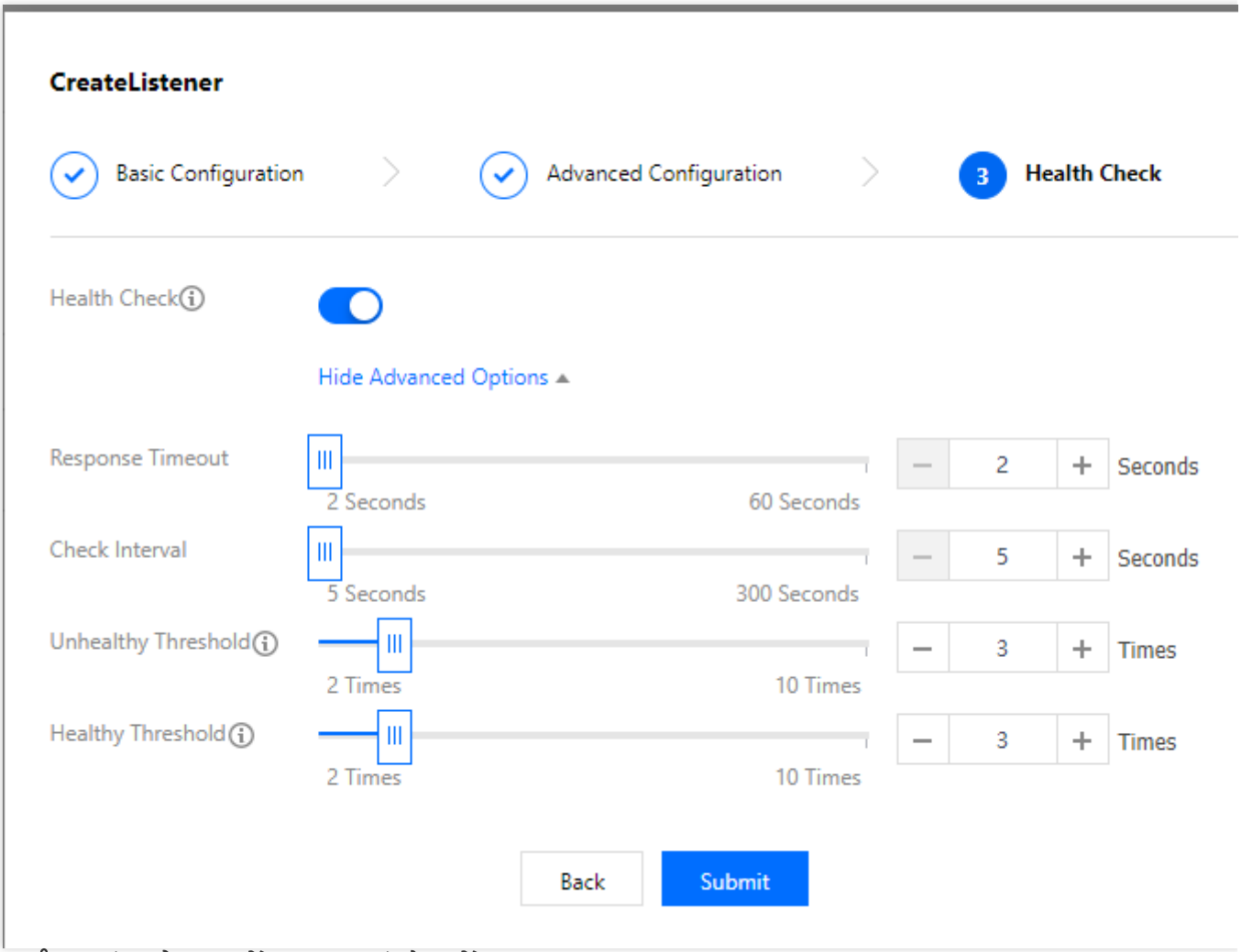


3. **ヘルスチェック**

ヘルスチェックの設定	説明	事例
ヘルスチェックステータス	ヘルスチェックをオンまたはオフにします。TCPリスナーでは、CLBインスタンスが指定のサーバーポートにSYNパッケージを送信し、ヘルスチェックを行います。	オン
チェックプロトコル	補足待機中	補足待機中
チェックポート	補足待機中	補足待機中

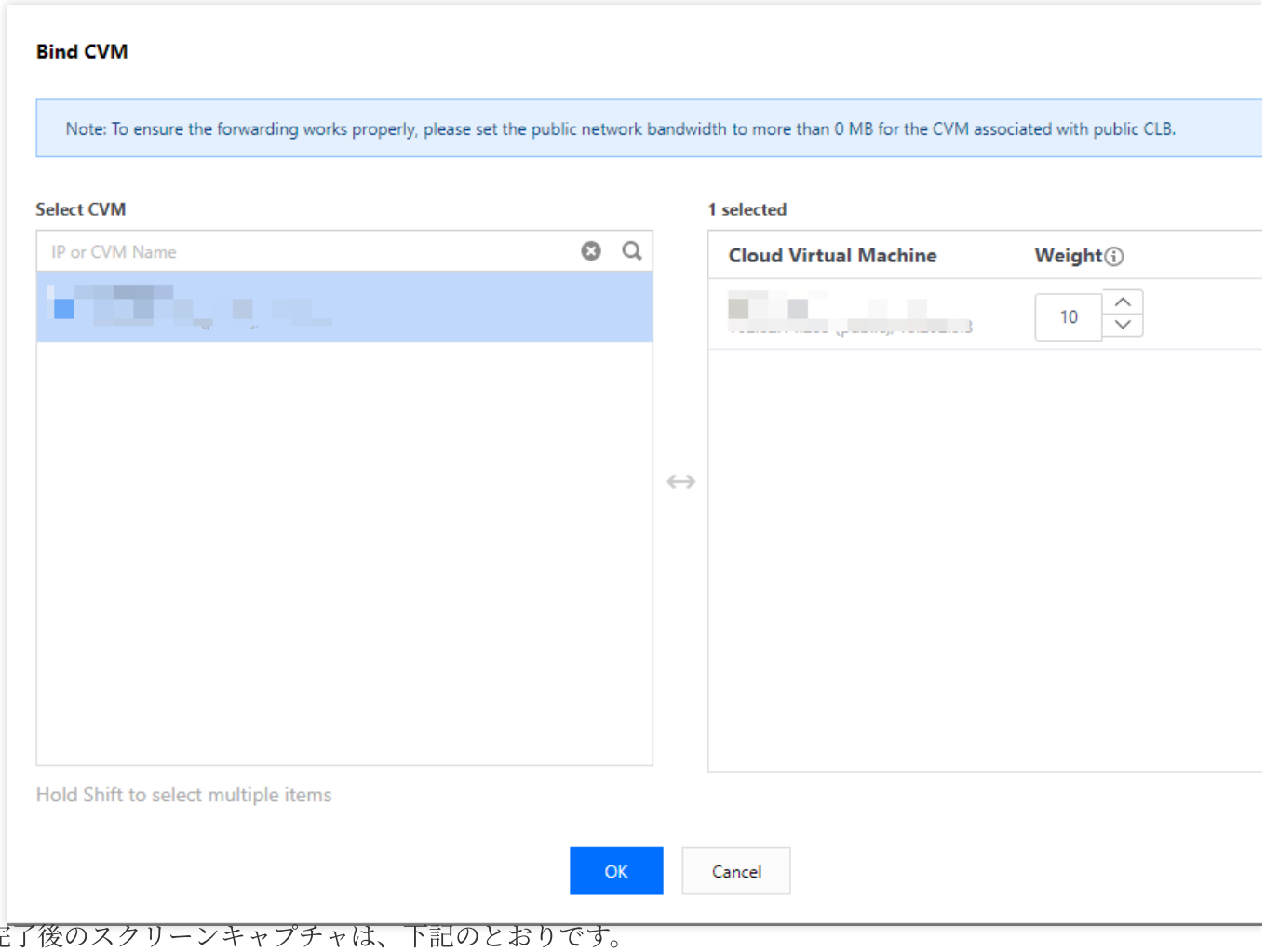
レスポンスタイムアウト	ヘルスチェックのレスポンスの最大タイムアウト時間です。 バックエンドCVMからタイムアウト時間内に正確なレスポンスがない場合は、ヘルスチェックに異常があると判断されます。 設定可能範囲：2～60秒で、デフォルト値は2秒となっています。	2s
チェック間隔	CLBがヘルスチェックを行う時間の間隔です。 設定可能範囲：5～300秒で、デフォルト値は5秒となっています。	5s
不健全なしきい値	n回（nには数値を入力）連続してヘルスチェック失敗の結果を受信した場合に、異常であると認識し、コンソールで**異常**と表示します。 設定可能範囲：2～10回で、デフォルト値は3回となっています。	3回
健全なしきい値	n回（nには数値を入力）連続してヘルスチェック成功の結果を受信した場合に、正常であると認識し、コンソールで**正常**と表示します。 設定可能範囲：2～10回で、デフォルト値は3回となっています。	3回

ヘルスチェックの具体的な設定については、下図に示すとおりです。

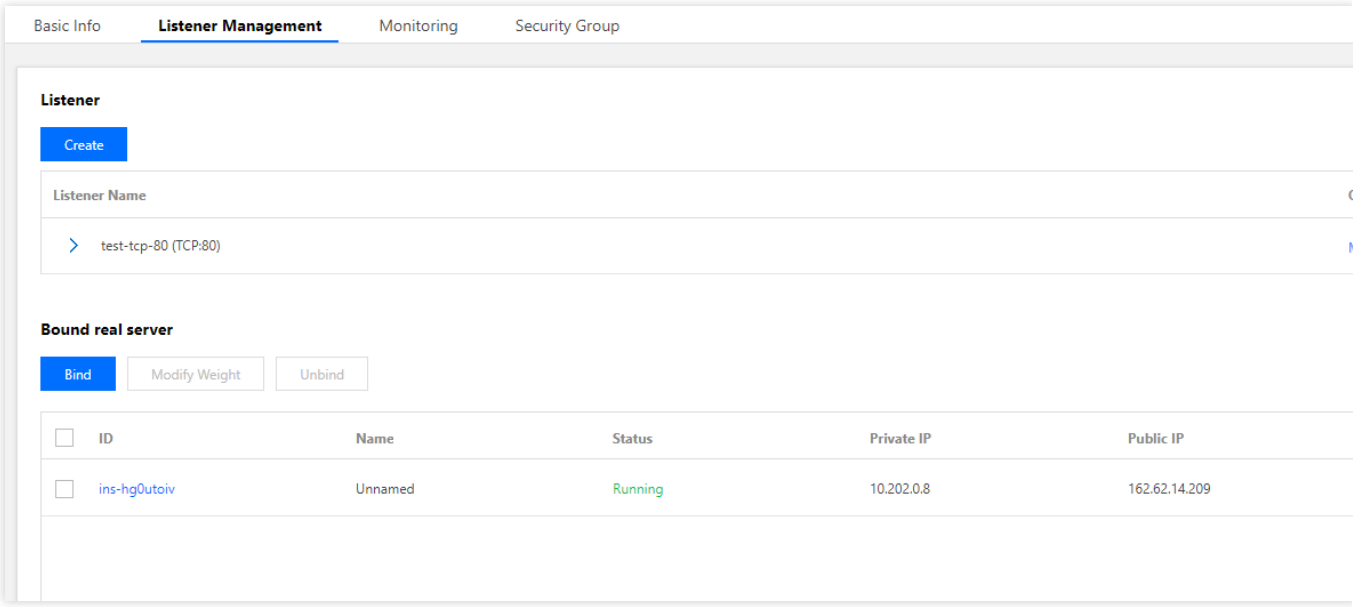


ステップ3：バックエンドCVMのバインド

「リスナー管理」ページで、**バインド**ボタンをクリックし、ポップアップボックスからバインドしたいバックエンドCVMを選択します。バインドの詳細については、下記のとおりです。



設定完了後のスクリーンキャプチャは、下記のとおりです。



説明：

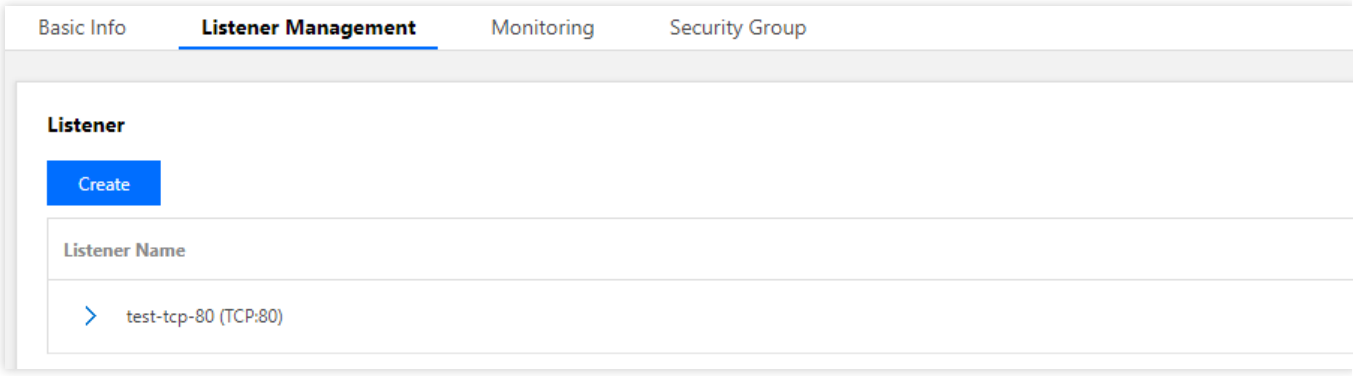
従来型CLBにおいて、複数のリスナーを設定した場合、複数のバックエンドCVMがバインドされるため、各リスナーは、その設定に従ってすべてのバックエンドサーバーに転送を行います。

ステップ4：セキュリティグループ（オプション）

CLBのセキュリティグループを設定してパブリックネットワークトラフィックの分離を行うことができます。詳細については、CLBセキュリティグループの設定をご参照ください。

ステップ5：リスナーの変更/削除（オプション）

作成済みのリスナーを変更または削除したい場合は、「リスナー管理」ページで、作成済みのリスナーを選択し、**変更**または**削除**を選択して操作を完了させてください。



従来型CLBの管理バックエンドCVM

最終更新日：：2024-01-04 18:36:26

従来型CLBは、正常に動作しているバックエンドCVMインスタンスにリクエストをルーティングします。従来型CLBを初めて使用する際または業務上のニーズに応じてバックエンドサーバーの数を追加または削除したい場合は、本テキストのガイドに従って操作することができます。

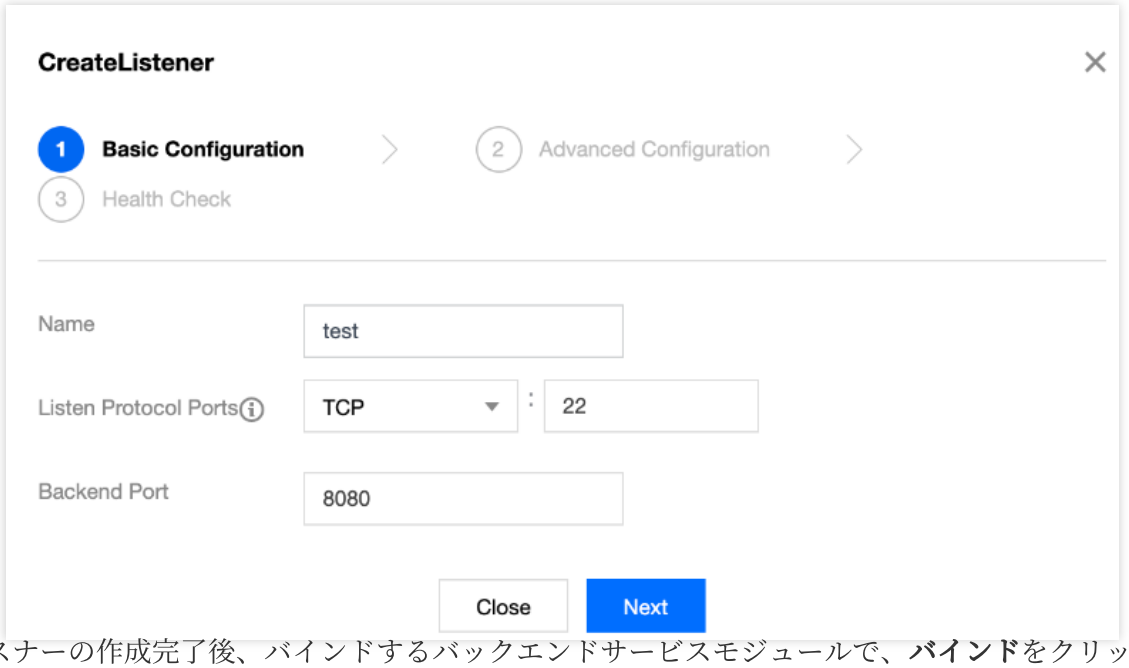
前提条件

従来型CLBインスタンスを作成済みで、リスナーの設定をしていることが必要です。詳細については、従来型CLBのクイックスタートをご参照ください。

操作手順

従来型CLBのバックエンドサーバーの追加

説明：
従来型CLBインスタンスがある自動スケーリンググループに関連付けられている場合、このグループのCVMが、従来型CLBのバックエンドCVMに自動的に追加されます。自動スケーリンググループから削除されたCVMインスタンスは、従来型CLBのバックエンドCVMからも自動的に削除されます。
APIを使用してバックエンドサーバーを追加したい場合は、バックエンドサービスの従来型CLBへのバインドインターフェースの説明をご参照ください。
1. CLBコンソールにログインします。
2. 「インスタンス管理」ページで、**従来型CLB**をクリックします。
3. 目標とする従来型CLBインスタンスの右側の操作リストで、**リスナーの設定**をクリックします。
4. リスナーモジュールの設定で、**作成**をクリックします。
5. 「リスナーの作成」のポップアップウィンドウで、「バックエンドポート」（ポートの選択については、サーバーの一般的なポートをご参照ください）およびその他の関連するフィールドを入力し、**次のステップ**をクリックして、引き続き設定を完了させます。詳細については、従来型CLBの設定をご参照ください。
説明：
従来型CLBは、**リスナーの作成フェーズ**で、バックエンドサーバーのポートを指定する必要があります。



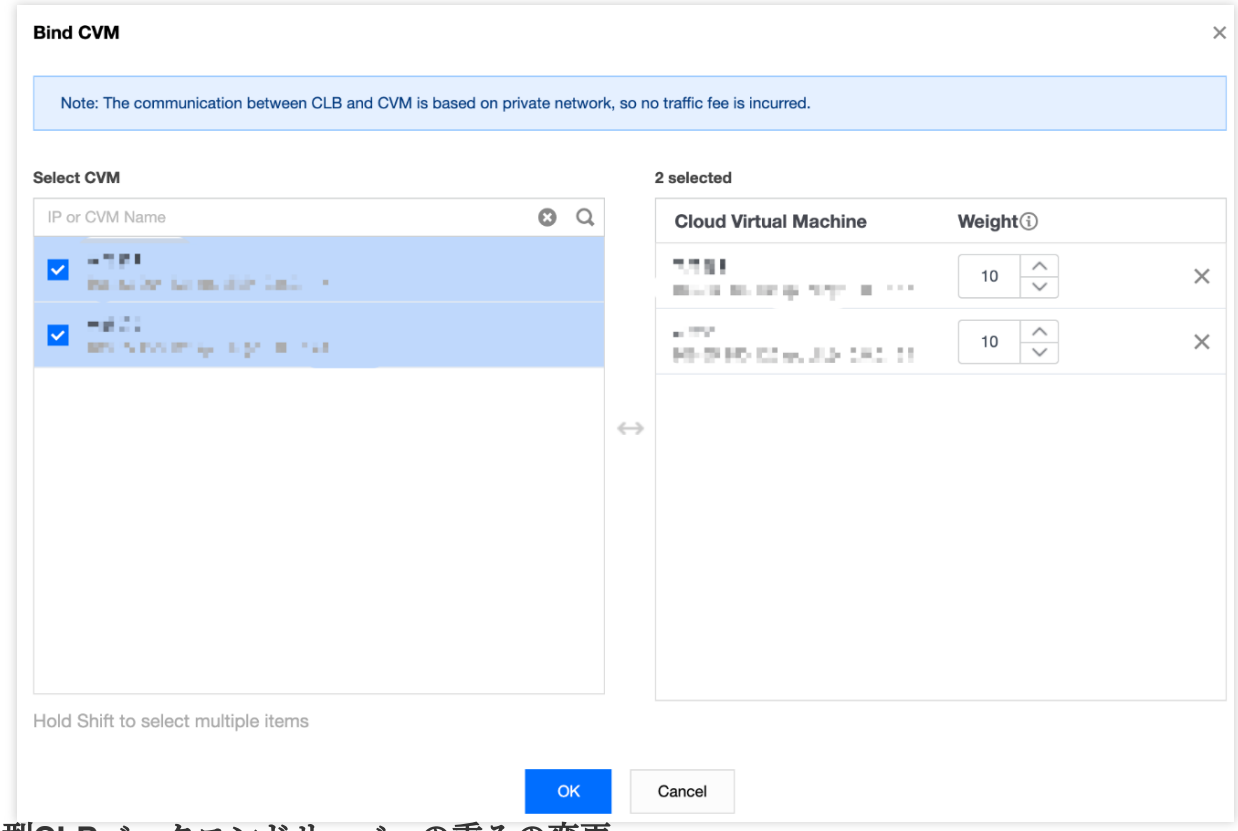
6. リスナーの作成完了後、バインドするバックエンドサービスモジュールで、**バインド**をクリックします。
7. 「CVMのバインド」のポップアップウィンドウで、バインドしたいCVMにチェックを入れ、「重み」の箇所で重みの情報を入力し、**確定**をクリックします。

説明：

ポップアップボックスには、同一のリージョン、同一のネットワーク環境において、隔離されていない、期限切れではない、帯域幅（ピーク値）が0ではない、選択可能なCVMのみが表示されます。

複数のバックエンドサーバーをバインドする場合、CLBはHashアルゴリズムに基づきトラフィックを転送することで、負荷分散の役割を果たします。

重みが高いほど、転送されるリクエストの数は多くなります。デフォルトでは10、設定可能範囲は0～100です。重みを0に設定すると、そのサーバーは新しいリクエストを受信しなくなります。セッション維持を有効にしていると、バックエンドサーバーのリクエストが不均一になる可能性があります。詳細については、バランシングアルゴリズムの選択と重みの設定の例をご参照ください。



従来型CLBバックエンドサーバーの重みの変更

説明：

従来型CLBは、APIを使用したバックエンドサーバーの重みの変更を、現時点ではサポートしていません。

1. CLBコンソールにログインします。
2. 「インスタンス管理」ページで、**従来型CLB**をクリックします。
3. 目標とする従来型CLBインスタンスの右側の操作リストで、**リスナーの設定**をクリックします。
4. バックエンドサービスのモジュールをバインドして、関連するサーバーの重みを変更します。

説明：

重みが高いほど、転送されるリクエストの数は多くなります。デフォルトでは10、設定可能範囲は0～100です。重みを0に設定すると、そのサーバーは新しいリクエストを受信しなくなります。セッション維持を有効にしていると、バックエンドサーバーのリクエストが不均一になる可能性があります。詳細については、バランシングアルゴリズムの選択と重みの設定の例をご参照ください。

方法1：あるサーバーの重みを単独で変更します。

4.1.1 重みを変更したいサーバーを見つけ、カーソルを対応する重みの上方に移動させて、

編集ボタンをクリックします。

4.1.2 「重みの変更」ポップアップウィンドウに、変更後の重み値を入力し、**送信**をクリックします。

方法2：いくつかのサーバーの重みを一括変更します。

説明：

一括変更した後のサーバーの重みはすべて同じになります。

4.1.1 サーバーの前にあるチェックボックスをクリックして複数のサーバーを選択し、リストの上方で**重みの変更**をクリックします。

4.1.2 「重みの変更」ポップアップウィンドウに、変更後の重み値を入力し、**送信**をクリックします。

従来型CLBのバックエンドサーバーのバインド解除

説明：

バックエンドサーバーのバインドを解除すると、従来型CLBインスタンスとCVMインスタンスの関連付けが解除され、従来型CLBからのリクエスト転送はその時点で停止します。

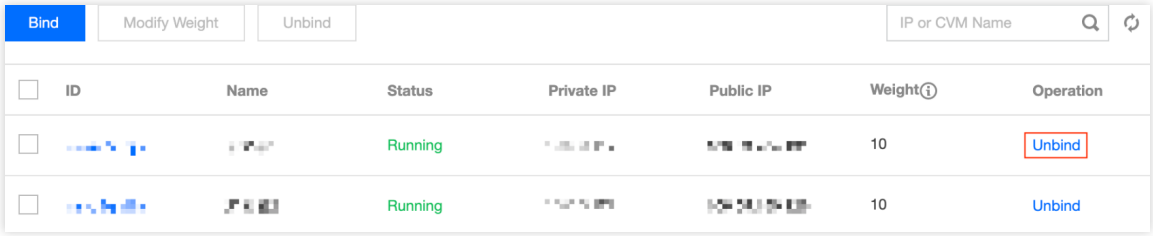
バックエンドサーバーのバインドを解除しても、CVMのライフサイクルには影響はありません。再度バックエンドサーバークラスターに追加することもできます

APIを使用してバックエンドサーバーのバインドを解除したい場合は、従来型CLBのバックエンドサーバーのバインドの解除インターフェースの説明をご参照ください。

1. CLBコンソールにログインします。
2. 「インスタンス管理」ページで、**従来型CLB**をクリックします。
3. 目標とする従来型CLBインスタンスの右側の操作リストで、**リスナーの設定**をクリックします。
4. バックエンドサービスのモジュールをバインドして、バインド済みのサーバーのバインドを解除します。

方法1：あるサーバーのバインドを単独で解除します。

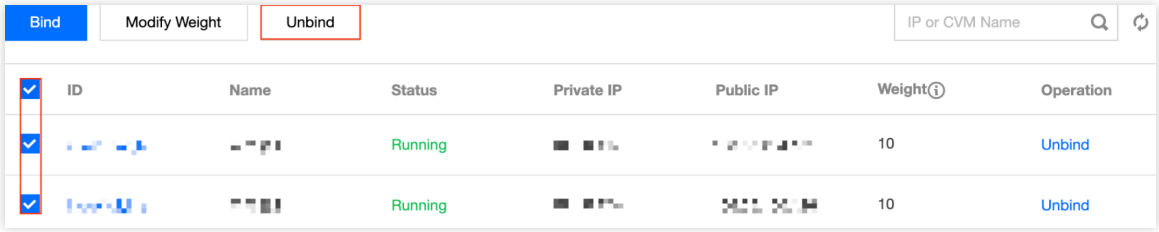
4.1.1 バインドを解除したいサーバーを見つけ、右側の操作バーで**バインド解除**をクリックします。



4.1.2 「バックエンドサービスのバインド解除」ポップアップウィンドウで、バインドを解除するサービスを確認し、**送信**をクリックします。

方法2：いくつかのサーバーのバインドを一括解除します。

4.1.1 サーバーの前にあるチェックボックスをクリックして複数のサーバーを選択し、リストの上方で**バインド解除**をクリックします。



4.1.2 「バックエンドサービスのバインド解除」ポップアップウィンドウで、バインドを解除するサービスを確認し、**送信**をクリックします。