Server Load Balancer

ユーザーガイド

ユーザーガイド

SLB インスタンス

パフォーマンス専有型インスタンス

Alibaba Cloudは、さまざまなリージョンでパフォーマンス占有型のインスタンスをリリースしました。 MaxConnection、CPS、QPSなどのパフォーマンスメトリックは、保証されたパフォーマンスのインスタンスSLAが含まれます。 対照的に、パフォーマンス共有型・インスタンスはパフォーマンス保証を提供しません。 Server Load Balancerリソースは、パフォーマンス共有型・インスタンス間で共有されます。

パフォーマンス専有型・インスタンスの3つの主要なメトリックは次のとおりです。

最大接続

SLBインスタンスへの最大接続数。 接続の最大数が仕様の限界に達すると、新しい接続が削除されます。

毎秒の接続（CPS）

1秒間に新しい接続が確立される速度。 CPSが仕様の限界に達すると、新しい接続が切断されます。

クエリ/秒（QPS）

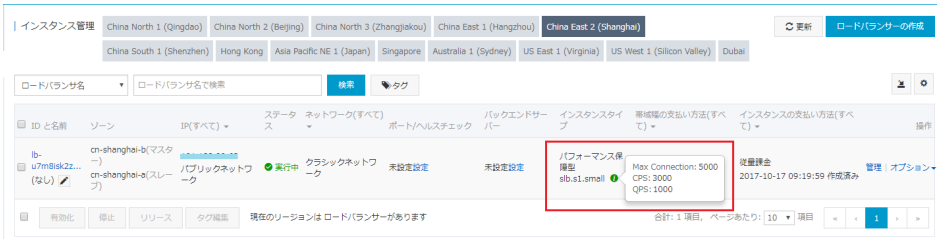
1秒あたりに処理できるHTTP / HTTPSクエリ/要求の数。これはレイヤー7リスナーに固有のものです。 QPSが仕様の限界に達すると、新しい接続が切断されます。

Alibaba Cloud Server Load Balancerには、パフォーマンス専有型・インスタンスの次の仕様が用意されています。

仕様		最大接続	CPS	QPS
仕様1	Small I	5000	3000	1000

	(slb.s1.small)			
仕様2	Standard I (slb.s2.small)	50000	5000	5000
仕様3	Standard II (slb.s2.medium)	100000	10000	10000
仕様4	Higher I (slb.s3.small)	200000	20000	20000
仕様5	Higher II (slb.s3.medium)	500000	50000	30000
仕様6	Super I (slb.s3.large)	1000000	100000	50000

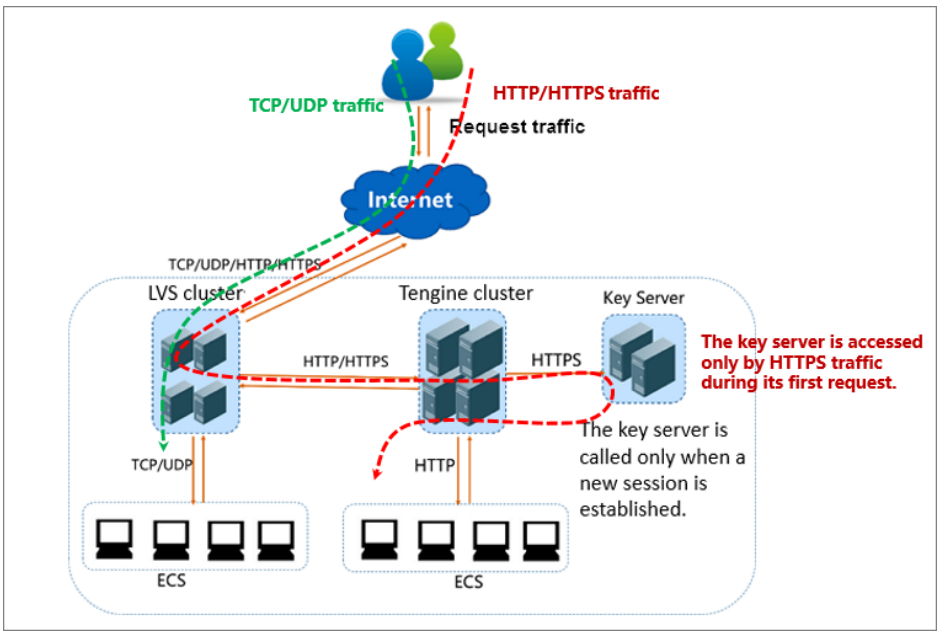
パフォーマンス専有型のインスタンスを起動する前に、以前に作成したすべてのインスタンスがパフォーマンスの共有インスタンスになります。 パフォーマンス専有型のインスタンスの場合、次の図に示すように、コンソールで仕様を表示できます。



ネットワークトラフィック

インバウンドネットワークトラフィック

Server Load Balancer は、コンソールまたは API で構成された転送ルールに従って着信トラフィックを配信します。次の図は、ネットワークのトラフィックフローを示しています。



TCP/UDP プロトコルまたは HTTP/HTTPS プロトコルにかかわらず、着信トラフィックは最初に LVS クラスタを介して転送する必要があります。

多数の着信トラフィックは、 LVS クラスタ内のすべてのノードサーバに均等に分散され、ノードサーバはセッションを同期させて高可用性を保証します。

レイヤ 4 Server Load Balancer （フロントエンドプロトコルは UDP または TCP ）の場合、LVS クラスタ内のノードサーバは、設定された転送ルールに従ってバックエンド ECS インスタンスにリクエストを直接配信します。

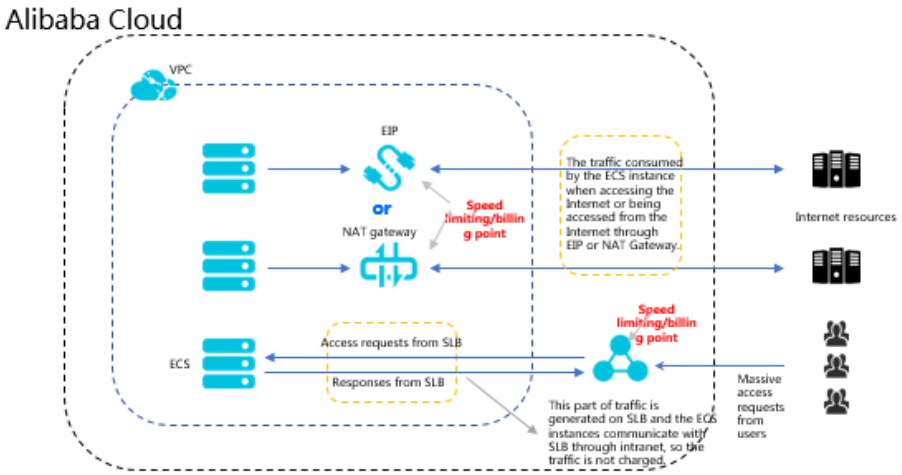
レイヤ 7 の Server Load Balancer （フロントエンドプロトコルは HTTP ）の場合、 LVS クラスタ内のノードサーバは、最初に Tengine クラスタにリクエストを配信します。その後、Tengine クラスタ内のノードサーバーは、構成された転送ルールに従ってバックエンド ECS インスタンスにリクエストを配信します。

レイヤ 7 Server Load Balancer （フロントエンドプロトコルは HTTPS ）の場合、リクエストの配布は HTTP プロトコルに似ています。しかし、バックエンド ECS インスタンスにリクエストを分配する前に、システムは鍵サーバを呼び出して証明書を検証し、データパケットを解読します。

アウトバウンドネットワークトラフィック

Server Load Balancer は、イントラネットを介してバックエンド ECS インスタンスと通信します。ただし、バックエンド ECS インスタンスから外部サービスを提供する場合や、バックエンド ECS インスタンスがインターネットにアクセスする必要がある場合は、 EIP または NAT Gateway 設定などのパブリック IP

を設定する必要があります。次の図は、アウトバウンドネットワークのトラフィックフローを示しています。



一般的な原則：トラフィックはどこから来るのですか：

Server Load Balancer からのトラフィックは、 Server Load Balancer で請求および速度制限が行われます。インバウンドトラフィックではなく、アウトバウンドトラフィックによって課金されます（ルールは将来変更される可能性があります）。 Server Load Balancer は、イントラネットを介してバックエンド ECS インスタンスと通信し、対応するトラフィックには課金されません。

EIP または NAT ゲートウェイからのトラフィックについては、課金および速度制限は、 EIP または NAT ゲートウェイ上で実行されます。 ECS インスタンスが作成されたときにパブリック IP が設定されている場合、請求および速度制限は ECS インスタンスで行われます。

Server Load Balancer は、インターネットからアクセスできるだけの容量を提供します。つまり、バックエンドの ECS インスタンスは、 Server Load Balancer によって転送されたリクエストに応答するときにのみインターネットにアクセスできます。バックエンドの ECS インスタンスからインターネットにアクティブにアクセスする場合は、 ECS インスタンスのパブリック IP （ EIP または NAT Gateway 設定）を設定する必要があります。

パブリック IP （ ECS インスタンスの作成時に設定される）、 EIP 、および NAT Gateway はすべて相互インターネットアクセス（アクセスする、またはアクセスされる）を達成できますが、トラフィックの転送やトラフィック負荷の分散はできません。

Server Load Balancer インスタンスの作成

前提条件

Server Load Balancerインスタンスを作成する前に、環境を正しく準備していることを確認してください。詳しくは、**データプランニング**を参照してください。

操作手順

Server Load Balancer 管理コンソールにログインします。

[**ロードバランサーの作成**] をクリックします。

Server Load Balancer インスタンスを設定します。

設定		説明
基本設定	リージョン	SLB の作成リージョンを選択します。 **注意**: SLB の作成リージョンはバックエンド ECS のと同一である必要があります。
	ゾーンタイプ	選択したリージョンのゾーンタイプを表示します。 クラウド製品のゾーンは、一連の独立したインフラストラクチャを指し、通常はインターネットデータセンター（IDC）によって表されます。異なるゾーンには独立したインフラストラクチャ（ネットワーク、電源、空調など）があります。したがって、あるゾーンのインフラストラクチャ障害は他のゾーンに影響しません。ゾーンは特定のリージョンに属し、単一のリージョンは 1 つ以上のゾーンを持つことができます。 - シングルゾーン：Server Load Balancer インスタンスは 1 つのゾーンにのみ展開されます。 - マルチゾーン：Server Load Balancer インスタンスは 2 つのゾーンに展開されます。デフォルトでは、プライマリゾーンのインス

		タンスはトラフィックの分散に使用されます。プライマリゾーンに障害が発生すると、バックアップゾーンのインスタンスが自動的にロードバランシングサービスを引き継ぎます。
	プライマリゾーン	Server Load Balancer インスタンスのプライマリゾーンを選択します。プライマリゾーンは通常の状態でトラフィックを伝送します。
	バックアップゾーン	Server Load Balancer インスタンスのバックアップゾーンを選択します。バックアップゾーンは、プライマリゾーンが使用できない場合にのみトラフィックを引き継ぎます。
ネットワークとインスタンスタイプ	インスタンスタイプ	ビジネスニーズに基づいてインターネットまたはイントラネットのインスタンスを作成することができます。システムはそれに応じてパブリック IP またはプライベート IP を割り当てます。詳細は **インスタンスとネットワークタイプ** を参照してください。 - インターネット: インターネットインスタンスは、リスナーの設定に従って、インターネットクライアントからのリクエストをバックエンド ECS インスタンスに転送します。 - イントラネット: イントラネットサーバーLoad Balancerインスタンスは、Server Load Balancerインスタンスのプライベートネットワークにアクセスできるクライアントからの要求のみを配布します。
	インスタンススペック	インスタンスのスペック仕様を選択します。 詳細は、 **インスタンスとネットワークの種類**を参照してください。
	ネットワークタイプ	インスタンスタイプは **イントラネット** の場合、ネットワークタイプの選

		択は必要です。 - クラッシクネットワーク: インスタンスの IP アドレスは Alibaba Cloud によって一元的に割り当てられ、管理されます。 - VPC: インスタンスの IP アドレスは、指定した VSwitch CIDR ブロックから割り当てられます。
	課金タイプ	課金方式を選択します。
購入プラン	数	購入台数を選択します。

[**今すぐ購入**] をクリックします。

Server Load Balancerインスタンスの管理

[Server Load Balancerコンソール]にログインし、**[インスタンス管理]** ページで、リージョン選択すると、選択したリージョンのすべてのServer Load Balancerインスタンスを表示できます。この他の操作も可能です：

Server Load Balancerインスタンス名を変更する

マウスカーソルをサーバーロードバランサIDに移動し、表示されている鉛筆アイコンをクリックして、インスタンス名を入力します。

Server Load Balancerインスタンスを停止する

実行中のServer Load Balancerインスタンスを選択し、ページの下部にある**[停止]**をクリックするか、**[オプション]** > **[停止]**をクリックします。

停止したServer Load Balancerインスタンスを開始する

停止したServer Load Balancerインスタンスを選択し、ページの下部にある **[有効化]** をクリックするか、または **[オプション]** > **[有効化]** をクリックします。

Server Load Balancerインスタンスをリリースする

Server Load Balancerインスタンスを選択し、ページの下部にある**[リリース]**をクリックするか、**[オプション]** > **[リリース]** をクリックします。**[リリース]**ダイアログで、インスタンスを即時にリリースするか、指定した時刻にインスタンスをリリースするかを選択します。

タグを設定する

タグを使用してインスタンスを統一された方法で分類および管理できます。 詳細はタグを使用してインスタンスを管理するを参照してください。

View Server Load Balancerのインスタンスの詳細

対象のServer Load BalancerインスタンスのIDをクリックするか、または **[管理]** をクリックして詳細を表示します。

詳細ページで、**[ご利用明細]**をクリックすると、Server Load Balancerサービスの詳細料金が表示されます。

インスタンスリスナー をクリックして、Server Load Balancerリスナーを表示および追加します。 詳しくは、リスナーの概要を参照してください。

バックエンドサーバーを追加および表示するには、[**サーバー**]をクリックします。 詳細は、バックエンドサーバーの概要を参照してください。

インスタンスのモニタリング をクリックすると、モニタ情報が表示され、アラームのメカニズムが設定されます。詳細はアラーム設定を参照してください。

タグを使用してインスタンスを管理する

タグの概要

Server Load Balancer はタグ機能があり、 Server Load Balancer インスタンスにタグを追加して分類するのに役立ちます。

各タグはキーと値で構成されています。タグを使用する場合は、次の制限に注意してください。

リージョン間でタグを使用することはできません。リージョン固有のリソースです。

タグを Server Load Balancer インスタンスに追加する必要があります。そうでない場合、タグは削除されます。

Server Load Balancer インスタンスには、最大 10 個のタグを追加できます。

インスタンスに追加される各タグのキーは一意でなければなりません。それ以外の場合は、同じキーのタグが上書きされます。

タグを追加

1. Server Load Balancer コンソールにログオンします。

   インスタンスページで、タグを追加するターゲットインスタンスを見つけて、**オプション> タグを編集**をクリックします。

   タグ編集ダイアログボックスで**作成**をクリックし、キーと値を入力します。**確認**をクリックします。

タグを使用してインスタンスの検索

インスタンスページで、**タグ**をクリックします。

検索条件として使用するキーと値を選択します。

指定されたタグを持つインスタンスが表示されます。タグの横にある削除アイコンをクリックすると、フィルタを消去できます。

タグの表示と検索

1. 左側のナビゲーションペインで、[**タグ**]をクリックします。作成されたすべてのタグがリストされます。
2. タグのキーを選択し、値を入力してタグを検索します。

タグの削除

関連するすべてのインスタンスからタグが削除されると、タグは削除されます。

インスタンスページで、タグを追加するターゲットインスタンスを見つけて、**その他> タグを編集**をクリックします。

タグの横にある削除アイコンをクリックし、**確認**をクリックします。

タグが複数のインスタンスに帰属する場合は、各インスタンスからタグを削除する必要があります。タグがどのインスタンスにも関連付けられていないと、システムから削除されます。

SLBインスタンススペックの変更手順

共有パフォーマンス・インスタンスを保証されたパフォーマンスのインスタンスに変更することも、保証されたパフォーマンスのインスタンスのキャパシティを変更することもできます。

構成を変更する前に、次の点に注意してください。

共有パフォーマンスのインスタンスを保証されたパフォーマンスのインスタンスに変更すると、サービスの短時間の切断が 10～30 秒間発生することがあります。

サービスがビジー状態でない場合は、この操作を実行することをお勧めします。

共有パフォーマンス・インスタンスを保証されたパフォーマンスのインスタンスに変更した後は、それを元に戻すことはできません。

保証されたパフォーマンスのインスタンスに変更した後、代わりに（slb.s1.small）容量を使用することができます。slb.s1.small の容量料金は回収されません。

手順

Server Load Balancer コンソールにログインする。

インスタンス管理をクリックし、対象インスタンスを確認し、 **オプション** > **変更設定** の順でクリックします。

設定のアップグレード ページで新しいスペックを選択し、 **有効化** をクリックし、インスタンスのスペックの変更を完了します。

変更は即時反映されます。

リスナー

リスナーの概要

リスナーの概要

Server Load Balancerインスタンスを作成したら、Server Load Balancerインスタンスのリスナーを構成する必要があります。リスナーは接続要求をチェックし、設定されたルールに従ってバックエンドサーバーに対し要求を配信します。

次の図に示すように、リスナーにはリスナー構成とヘルスチェック構成が含まれています。

リスナーの設定

Alibaba Cloudは、レイヤ4（TCPおよびUDPプロトコル）およびレイヤ7（HTTPおよびHTTPSプロトコル）のロードバランシングサービスを提供します。ビジネスニーズに応じてプロトコルを選択します。

プロトコル	説明	アプリケーションシナリオ
TCP	●接続指向のプロトコル。データを送受信できるようにするには、信頼できる接続をピアエンドと確立する必要があります。 ●送信元アドレスベースのセッション永続性。 ●ソースアドレスは、ネットワークレイヤーで使用できます。 ●高速なデータ送信	●ファイル伝送、電子メールの送受信、リモートログオンなど、低速の許容範囲を備えた信頼性とデータの正確性が高いシナリオに利用できます。 ●特別な要件がないWebアプリケーション。
UDP	●非接続指向のプロトコル。データを送信する前に、UDPは、相手と3回のハンドシェイクを行うのではなく、データパケットの送信を直接実行し、エラー回復とデータ再送信を行いません。 ●しかし、高速データ伝送では、信頼性は比較的低い	ビデオチャットやリアルタイムの財務諸表のプッシュなど、信頼性を超えるリアルタイムコンテンツを優先するシナリオに適用可能
HTTP	●主にデータをパッケージ化するために使用されるアプリケー	Webアプリケーションや小型モバイルゲームなどのデータコ

	ション層プロトコル。 ●Cookieベースのセッション永続性ソース ●IPアドレスを取得するには、X-Forward-Forを使用します。	ンテンツを認識する必要があるアプリケーションに適用できます。
HTTPS	●HTTPに似ていますが、不正なアクセスを防ぐ暗号化された接続があります。 ●統一された証明書管理サービス。証明書をServer>統一された証明書管理サービス。証明書をServer Load Balancerにアップロードすると、復号化操作はServer Load Balancerで直接完了します。	暗号化された通信が必要なアプリケーションなど。

ヘルスチェック設定

Server Load Balancerは、バックエンドサーバーのヘルスチェックを提供し、サービスの可用性を向上させます。

詳しくは、健康チェックの概要および健康チェックの設定を参照してください。

レイヤー4リスナー

レイヤー4リスナーを構成する

レイヤー 4 リスナーの概要

Alibaba Cloud は、レイヤ 4（TCP および UDP プロトコル）のロードバランシングサービスを提供します。レイヤー 4 リスナーは、HTTP ヘッダーを変更することなく、要求を直接バックエンド ECS インスタンスに転送します。

TCP プロトコル

接続指向のプロトコル。データを送受信できるようにするには、信頼できる接続をピア側と確立する必要があります。

ファイル転送、電子メールの送受信、リモートログオン、特別な要件なしの Web アプリケーションなど、信頼性とデータの正確性は高いが、低速に対する許容性が高いシナリオに適用できます。

UDP プロトコル 非接続指向のプロトコル。データを送信する前に、相手と 3 回のハンドシェイクを行うのではなく、データパケットの送信を直接実行し、エラー回復とデータ再送信を行いません。

ビデオチャットやリアルタイムの財務諸表のプッシュなど、信頼性を超えるリアルタイムコンテンツを優先するシナリオに適用できます。

UDP プロトコルリスナーを設定する場合は、次の制限に注意してください。

- リスナーごとの最大接続数：100,000。
- 現在、断片化されたパッケージはサポートされていません。
- 従来のネットワークで UDP プロトコルの実際のIPアドレスを取得することはサポートされていません。
- 次の 2 つのシナリオでは、UDP プロトコルリスナーの構成が有効になるまでに5分かかります。
    - バックエンドの ECS インスタンスを削除します。
    - バックエンド ECS インスタンスの重みを0に設定します。

レイヤー 4 リスナー構成

次の表に、TCP リスナーと UDP リスナーの構成を示します。

リスナー構成	説明
フロントエンドプロトコル[Port]	接続リクエストを受け取り、要求をバックエンドサーバーに転送するために使用されるフロントエンドプロトコルとポート。 レイヤー 4 リスナーを構成する場合は、TCP または UDP を選択します。 **注意**：Server Load Balancer インスタンス内のリスナーのフロントエンドポートは同じにすることはできません。
バックエンドプロトコル[ポート]	要求を受信するためにバックエンド ECS インスタンスでオープンされたポート。 バックエンドプロトコルは、フロントエンドプロトコルと同じです。 サーバー・グループではなくサーバー・プールにインスタンスを追加する場合、Server Load Balancer インスタンス内のリスナーのバックエンド・ポートは同じでなければなりません。

ピーク帯域幅	PayByBandwidth 請求方法のインスタンスでは、リスナーごとに異なる帯域幅のピークを設定して、リスナーによるトラフィックを制限できます。すべてのリスナーに設定された帯域幅の合計は、Server Load Balancer インスタンスに設定された帯域幅の合計量を超えることはできません。
スケジューリングアルゴリズム	サーバーロードバランサは、ラウンドロビン、重み付きラウンドロビン（WRR）、および重み付き最小接続（WLC）の 3 つ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。 ●ラウンドロビン：要求は、バックエンド ECS サーバー間で均等に分散されます。 ●加重ラウンドロビン（Weighted Round Robin：WRR）：ウェイトの高いバックエンドサーバーは、ウェイトの少ないサーバーよりも多くの要求を受け取ります。 ●加重最小接続（Weighted Least Connections：WLC）：加重値が高いサーバーは、一度にライブ接続の割合が高くなります。ウェイトが同じ場合、システムは、確立された接続数が最も少ないサーバーにネットワーク接続を送信します。
サーバー・グループを使用する	使用する場合は、リスナー・ディメンション内のバックエンド・サーバーを管理できます。 サーバーグループには、異なるポートを持つ複数のバックエンドサーバーが含まれています。サーバーグループを使用すると、異なるサーバーグループを持つ異なるリスナーを構成できます。したがって、リスナーは指定されたバックエンドサーバーに要求を転送できます。 **注意**：リスナーを使用してサーバーグループを構成すると、リスナーは選択したサーバーグループ内の ECS インスタンスに要求を転送します。Server Load Balancer インスタンスディメンションの追加された ECS インスタンスは、要求をもう受信しません。それ以外の場合、リスナーは、Server Load Balancer インスタンスディメンションに追加されたバックエンド ECS インスタンスに要求を転送します。詳細については、バックエンド ECS インスタンスの追加を参照してください。
サーバーグループタイプ	サーバーグループ機能を有効にしたら、使用するサーバーグループのタイプを選択します。 ●VServer Group：バックエンドサーバーとして追加された ECS インスタンスのグループ。VServer グループを使用すると、指定したバックエンドサーバーにトラフィックを配信し、要求ドメインと URL に基づいてトラフィックを分散するようにドメイン名/ URL 転送ルールを構成できます。詳細は、VServer グループの作成を

	参照してください。 ●マスター - スレーブサーバーグループ：2 つの ECS インスタンスのグループ。従来のプライマリバックアップ要求がある場合は、このタイプを選択できます。マスターサーバーが正常に動作すると、トラフィックはマスターサーバーに送られます。マスターサーバーが停止すると、トラフィックはスレーブサーバーに送られます。マスター/スレーブサーバーグループを使用すると、サービスの中断を避けることができます。詳細は、マスタ/スレーブサーバグループの作成を参照してください。
作成後にリスナーを自動的にアクティブ化	リスナーが作成されたらリスナーをアクティブにするかどうかを選択します。デフォルトの設定は、 **[有効]** です。

高度な設定

Real IPの取得	**レイヤ 4 リスナーの場合、クライアントの実際の IP を直接取得できます。注意：UDP プロトコルを使用する従来のネットワーク Server Load Balancer インスタンスでは、実 IP の取得機能はサポートされていません**
セッション持続性	セッション持続性を有効にするかどうかを選択します。有効化されている場合、クライアントからのすべての要求は、セッションの持続時間の間、同じバックエンドサーバーに送信されます。レイヤー 4 リスナーの場合、セッション持続性は IP に基づいています。同じ IP からの要求は、同じバックエンドサーバーに転送されます。 TCP リスナーの場合、セッションの永続性は IP アドレスに基づいています。同じ IP アドレスからのリクエストは、同じバックエンドサーバーに転送されます。注：セッションの永続性は、UDP リスナーではサポートされていません。
Connection Timeout	TCP接続のタイムアウト値を指定します。使用可能な値は10～900秒です。注：この設定は、TCPリスナーにのみ適用されます。
アクセス制御を有効にする	アクセス制御機能を有効にするかどうかを指定します。
アクセス制御方式	アクセス制御機能を有効にした後、アクセス制御方式を選択します。 **ホワイトリスト**:選択したアクセス制御リストのIPアドレスまたはCIDRブロックからの要求のみが転送されます。アプリケーションが特定のIPアドレスからのアクセスのみを許可するシナリオに適用されます。 ホワイトリストを有効にすると、ビジネス上のリスクが発生します。対応するアクセス制御リストにIPエントリを追加せずにホワイトリストを有効にすると、すべての要求が転送

	されます。 **ブラックリスト**:選択したアクセス制御リストのIPアドレスまたはCIDRブロックからの要求は転送されません。アプリケーションが特定のIPアドレスからのアクセスのみを拒否するシナリオに適用されます。対応するアクセス制御リストにIPエントリを追加せずにブラックリストを有効にすると、すべての要求が転送されます。
アクセス制御リストの選択	アクセスリストをホワイトリストまたはブラックリストとして選択します。詳細は**アクセス制御リストの設定**を参照してください。

レイヤー7リスナー

HTTPSリスナーの概要

データ転送のセキュリティ要件を満たすために、Server Load Balancer は HTTPS ロードバランシングをサポートしています。

サーバー負荷分散サーバーHTTPSリスナーは一方向認証と双方向認証の両方をサポートしています。サーバー証明書と CA 証明書を Server Load Balancer にアップロードするだけで、バックエンドサーバー上で構成を行う必要はありません。

HTTPS リスナーを設定するときは、次の点に注意してください。

HTTPS リスナーを設定する前に、次の表に示すように、必要な証明書を準備する必要があります。

証明書	説明	一方向認証に必要	双方向認証に必要
サーバー証明書	サーバーの識別に使用されます。 クライアントは、クライアントがそれを使用して、サーバーによって送信された証明書が認証局によって発行されているかどうかをチェックします。	はい。 Server Load Balancer にアップロードする必要があります。	はい。 Server Load Balancer にアップロードする必要があります。

クライアント証明書	お客様一の識別に使用されます。 サーバー側の通信におけるクライアントユーザーは、その真のアイデンティティを証明することができます。自己署名入りの CA 証明書でクライアント証明書に署名することができます。	いいえ。	はい。 クライアントにインストールする必要があります。
CA 証明書	サーバーは、セキュリティー保護された接続を開始する前に、CA 証明書を使用して、クライアント証明書の CA 署名を認証の一部として認証します。	いいえ。	はい。 サーバーロードバランサにアップロードする必要があります

Server Load Balancer に証明書をアップロードした後、Server Load Balancer コンソールで証明書の置換や削除などの証明書を管理できます。

証明書のアップロード、ロード、および検証には時間がかかるため、HTTPS リスナーをアクティブにするには通常 1〜3 分かかります。

HTTPS リスナーは、ECDHE メソッドを使用してキーを交換し、秘密鍵の転送をサポートしますが、BEGIN DH PARAMETERSなどのセキュリティ強化パラメータを含む PEM ファイルのアップロードはサポートしていません。

現在、Server Load Balancer HTTPS リスナーは SNI（Server Name Indication）をサポートしていません。

HTTPS リスナーのセッションチケットは 300 秒です。

実際のトラフィックは、プロトコルのハンドシェイクに使用されるトラフィックがあるため、課金トラフィックよりも大きくなります。

多数の新しい接続の場合、HTTPS リスナーは大量のトラフィックを使用します。

レイヤー7リスナーの構成

レイヤー7リスナーの概要

Alibaba Cloudは、レイヤー7（HTTPおよびHTTPSプロトコル）ロードバランシングサービスを提供します。原則として、レイヤー7リスナーはリバース・プロキシーの実装です。クライアントのHTTP要求がServer Load Balancerリスナーに到着すると、Server Load Balancerサーバーはバックエンドの ECSインスタンスとのTCP接続を確立します。つまり、バックエンドサーバーに直接パケットを転送するのではなく、新しいTCP接続を使用してHTTPプロトコルを接続してバックエンドにアクセスします。

HTTPプロトコル 主にデータをパッケージ化するために使用され、Webアプリケーションや小型モバイルゲームなどのデータコンテンツを認識する必要があるアプリケーションに適用可能なアプリケーション層プロトコルです。

HTTPSプロトコル 不正なアクセスを防止するための暗号化されたデータ送信で、暗号化された送信を必要とするアプリケーションに適用できます。

レイヤー7リスナー構成

リスナー構成	説明
Front-end Protocol [Port]	接続要求を受け取り、要求をバックエンドサーバーに転送するために使用されるフロントエンドプロトコルとポート。 レイヤー7リスナーを構成する場合は、HTTPまたはHTTPSを選択します。 **注**：Server Load Balancerインスタンス内のリスナーのフロントエンドポートは同じにすることはできません。
Backend Protocol [Port]	リクエストを受信するためにバックエンドECSインスタンスでオープンされたポート。 バックエンドプロトコルは、フロントエンドプロトコルと同じです。
Peak Bandwidth	PayByBandwidth請求方法のインスタンスでは、リスナーごとに異なる帯域幅のピークを設定して、リスナーによるトラフィックを制限できます。すべてのリスナーに設定された帯域幅の合計は、Server Load Balancerインスタンスに設定された帯域幅の合計量を超えることはできません。
Scheduling Algorithm	サーバーロードバランサは、ラウンドロビン、重

	み付きラウンドロビン（WRR）、および重み付き最小接続（WLC）の3つ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。 ●ラウンドロビン：要求は、バックエンドECSサーバー間で均等に分散されます。 ●加重ラウンドロビン（Weighted Round Robin：WRR）：ウェイトの高いバックエンドサーバーは、ウェイトの少ないサーバーよりも多くの要求を受け取ります。 ●加重最小接続（Weighted Least Connections：WLC）：加重値が高いサーバーは、一度にライブ接続の割合が高くなります。ウェイトが同じ場合、システムは、確立された接続数が最も少ないサーバーにネットワーク接続を送信します。
サーバー・グループを使用する	使用する場合は、リスナー・ディメンション内のバックエンド・サーバーを管理できます。 サーバーグループには、異なるポートを持つ複数のバックエンドサーバーが含まれています。サーバーグループを使用すると、異なるサーバーグループを持つ異なるリスナーを構成できます。したがって、リスナーは指定されたバックエンドサーバーに要求を転送できます。詳細は「VServerグループの作成」を参照してください。 **注意**：リスナーを使用してサーバーグループを構成すると、リスナーは選択したサーバーグループ内のECSインスタンスに要求を転送します。Server Load Balancerインスタンスディメンションの追加されたECSインスタンスは、要求をもう受信しません。それ以外の場合、リスナーは、Server Load Balancerインスタンスディメンションに追加されたバックエンドECSインスタンスに要求を転送します。詳細は「バックエンドECSインスタンスの追加」を参照ください。
Mutual Authentication	双方向HTTPS認証を有効にするかどうかを選択します。有効になっている場合は、サーバ証明書とCA証明書をサーバロードバランサにアップロードする必要があります。そうでない場合は、サーバ証明書だけが必要です。 **注意**：このオプションは、HTTPSリスナーでのみ使用できます。
サーバー証明書	サーバーが送信した証明書が信頼できるセンターによって署名されて発行されているかどうかをチェックするために、クライアント・ブラウザーが使用するサーバー証明書。 Alibaba Cloud Security証明書サービス、または他のサービスプロバイダからサーバ証明書を購入することができます。サーバー証明書は、サーバー負荷分散装置の証明書管理システムにアップロードする必要があります。 詳細に関しては 「サーバ証明書のア

	ップロード」を参照してください。 **注意**：このオプションは、HTTPSリスナーでのみ使用できます。
CA証明書	サーバーがクライアントのIDを確認するために使用する証明書。検証に失敗すると、接続は拒否されます。 CA証明書は、双方向認証が有効な場合にのみ必要です。自己署名CA証明書を使用して検証を行うことができます。詳細は、「**証明書の生成**」を参照してください。 **注意**：このオプションは、HTTPSプロトコルでのみ使用できます。
作成後にリスナーを自動的にアクティブ化	リスナーが構成された後にリスニングを使用可能にするかどうかを選択します。デフォルト設定は**有効化**です。

詳細設定

Real IPの取得	**レイヤ-7リスナーの場合、Server Load BalancerはHTTPヘッダーX-Forwarded-Forを使用してクライアントの実際のIPアドレスを取得します。**
セッション持続性	有効にすると、同じクライアントが同じバックエンドサーバーに送信されます。レイヤー7リスナーの場合、セッションの永続性はCookieに基づいています。次の2つのCookieメソッドがサポートされています。 Insert Cookie：Server Load Balancerはバックエンドサーバーからの最初の応答にセッションCookieを追加し、応答を送信したサーバーを識別します。次のリクエストにはクッキーの値が含まれ、リスナーはリクエストを同じバックエンドサーバーに配布します。このメソッドを使用すると、セッションのタイムアウトを指定するだけです。 Cookieを書き換えます：レスポンスにCookieの名前を設定できます。新しいCookieが設定されたことが検出されると、Server Load Balancerは元のCookieを上書きします。次回クライアントがServer Load Balancerにアクセスするための新しいCookieを保持すると、リスナーはその要求を以前に記録されたバックエンドサーバーに配信します。
アイドル接続タイムアウト	アイドル接続のタイムアウトを秒単位で指定します。有効な値：1-60。 現在、この機能はすべてのリージョンで利用可能です。
要求のタイムアウト	要求のタイムアウトを秒単位で指定します。有効な値：1-180。指定されたタイムアウト期間中に要求が受信されない場合、Server Load

	Balancerはその接続を閉じて、次の要求が来たときに接続を再開します。 現在、この機能はすべてのリージョンで利用可能です。
Gzip圧縮	特定の形式のファイルを圧縮するためにGzip圧縮を有効にするかどうかを選択します。現在、Gzipはext/xml、text/plain、text/css、application/javascript、application/x-javascriptapplication/rss+xml、application/atom+xml、application/xmlのファイルタイプをサポートしています。
追加するカスタムHTTPヘッダーを選択します。	X-Forwarded-For:このヘッダーを追加してクライアントIPを取得します。 X-Forwarded-Proto:このヘッダーを追加して、サーバーロードバランサへの接続に使用するプロトコルを取得します。 SLB-IP:このヘッダーを追加して、Server Load Balancerインスタンスのパブリックを取得します。 SLB-ID:このヘッダーを追加して、Server Load BalancerインスタンスのIDを取得します。
アクセス制御を有効にする	アクセス制御機能を有効にするかどうかを指定します。
アクセス制御方式	アクセス制御機能を有効にした後、アクセス制御方式を選択します。 **ホワイトリスト**:選択したアクセス制御リストのIPアドレスまたはCIDRブロックからの要求のみが転送されます。アプリケーションが特定のIPアドレスからのアクセスのみを許可するシナリオに適用されます。 ホワイトリストを有効にすると、ビジネス上のリスクが発生します。対応するアクセス制御リストにIPエントリを追加せずにホワイトリストを有効にすると、すべての要求が転送されます。 **ブラックリスト**:選択したアクセス制御リストのIPアドレスまたはCIDRブロックからの要求は転送されません。アプリケーションが特定のIPアドレスからのアクセスのみを拒否するシナリオに適用されます。対応するアクセス制御リストにIPエントリを追加せずにブラックリストを有効にすると、すべての要求が転送されます。
アクセス制御リストの選択	アクセスリストをホワイトリストまたはブラックリストとして選択します。詳細は**アクセス制御リストの設定**を参照してください。

HTTPSリスナーの構成（一方向の認証）

このチュートリアルでは、一方向認証でHTTPSリスナーを構成する手順を段階的に説明します。

HTTPSリスナー（一方向認証）を構成するには、次のタスクを実行します。

サーバー証明書をアップロードする

サーバー負荷分散装置を構成する

負荷分散サービスのテスト

Task 1: サーバー証明書をアップロードする

HTTPSリスナー（一方向認証）を構成する前に、Server Load Balancerの証明書管理システムにサーバー証明書をアップロードする必要があります。

[Server Load Balancerコンソール]にログインします。

左側のナビゲーションペインで、**「証明書」**をクリックし、**「証明書のアップロード」**をクリックします。

次のようにサーバー証明書を構成します。

認証書リージョン ： **China East 1（Hangzhou）**を選択します。

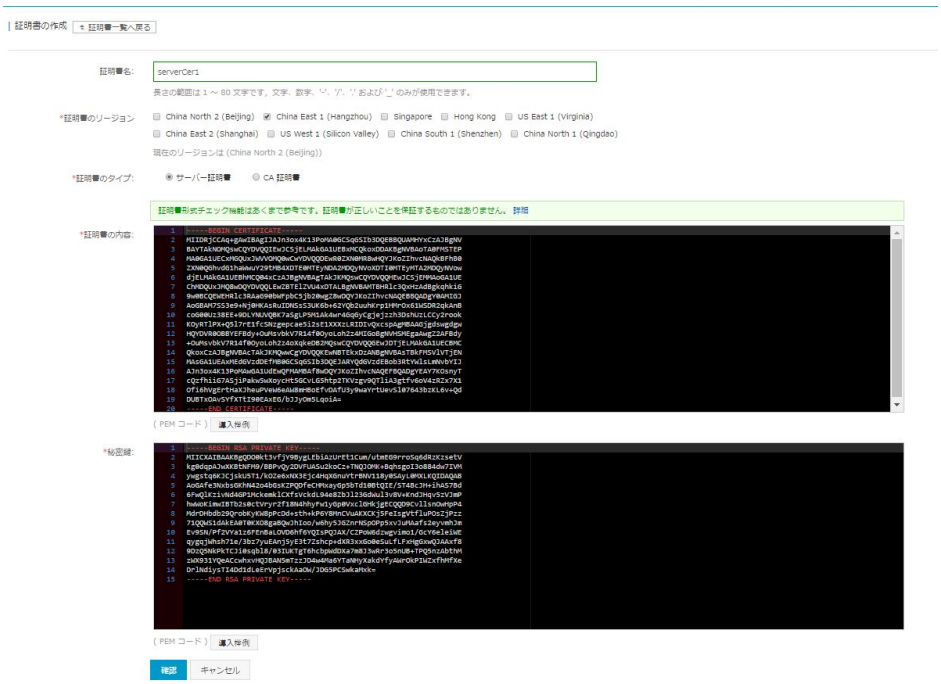
注意：証明書のリージョンは、Server Load Balancerインスタンスと同じである必要があります。

証明書のタイプ：**サーバー証明書**を選択します。

証明書の内容と秘密鍵：サーバー証明書の内容とサーバー証明書の秘密鍵をコピーします。**サンプルのインポート**をクリックして、証明書の有効な形式を表示できます。

証明書の形式については、証明書形式要件を参照してください。

確認をクリックします。



Task 2: サーバー負荷分散装置の構成

[Server Load Balancerコンソール]にログインします。

インスタンス管理ページで、 **ロードバランサーの作成**をクリックします。

インスタンスを構成し、**[今すぐ購入]**をクリックします。

> **注意**: インスタンスリージョンとして **中国東部 1** を選択し、ネットワークタイプとして**インターネット**を選択します。詳細は、Server Load Balancer インスタンスの作成を参照してください。

クリックし、ページ**インスタンス管理**に戻り、**中国東部1** リージョンを選択します。

Server Load BalancerインスタンスのIDをクリックします。

左側のナビゲーションペインで、 **インスタンスリスナー** をクリックし、 **リスナーの追加** をクリックします。

次のようにリスナーを構成します。

フロントエンドプロトコル [ポート]：HTTPS 443。

バックエンドプロトコル [ポート]：HTTP 80。

スケジューリングアルゴリズム：ラウンドロビン。

サーバー証明書：アップロードされたサーバー証明書を選択します。

[次のステップ]をクリックします。

デフォルトのヘルスチェック設定を使用し、**確認**をクリックして終了します。

左側のナビゲーションペインで、 **バックエンドサーバー** > **バックエンドサーバー** をクリックして、ECSインスタンスを追加します。

バックエンドサーバーの詳細については、バックエンドサーバーの概要を参照してください。

Task 3: 負荷分散サービスをテストする

WebブラウザにServer Load Balancerインスタンスのパブリックを入力し、リクエストがSSLプロトコ

ルで正しく処理されているかどうかを確認します。

HTTPSリスナーの構成（双方向認証）

このチュートリアルでは、双方向認証を使用してHTTPSリスナーをコンフィグレーションする手順を説明します。自己署名入りのCA証明書は、クライアント証明書の署名に使用されます。

HTTPSリスナー（双方向認証）を設定するには、次のタスクを実行します。

1. サーバー証明書の準備
2. Open SSLを使用してCA証明書を生成する
3. クライアント証明書を生成する
4. サーバー証明書とCA証明書をアップロードする
5. クライアント証明書をインストールする
6. Server Load Balancerインスタンスの構成
7. 負荷分散サービスのテスト

Step 1: サーバー証明書を準備する

HTTSリスナーを設定する前に、サーバー証明書を購入する必要があります。

Step 2: オープンSSLを使用してCA証明書を生成する

次のコマンドを実行して /rootディレクトリの下にcaフォルダを作成し、caフォルダの下に4つのサブフォルダを作成します。

```
 $ sudo mkdir ca
$ cd ca
$ sudo mkdir newcerts private conf server
```

- **newcerts**: CA証明書で署名された証明書を保存するために使用します。
- **private**: CA証明書の秘密鍵を保存するために使用されます。
- **conf**: 設定ファイルを保存するために使用します。
- **server**: サーバー証明書を保管するために使用されます。

confフォルダの下に以下の内容のopenssl.confファイルを作成します。

```
[ ca ]
```

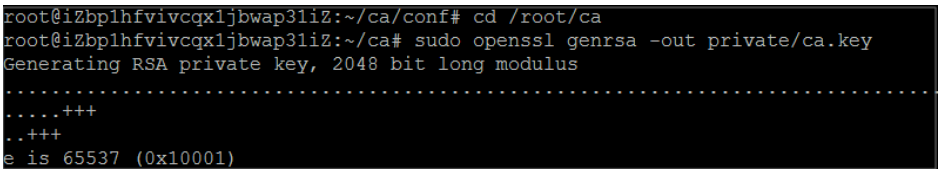
```
default_ca = foo
[ foo ]
dir = /root/ca
database = /root/ca/index.txt
new_certs_dir = /root/ca/newcerts
certificate = /root/ca/private/ca.crt
serial = /root/ca/serial
private_key = /root/ca/private/ca.key
RANDFILE = /root/ca/private/.rand
default_days = 365
default_crl_days= 30
default_md = md5
unique_subject = no
policy = policy_any

[ policy_any ]
countryName = match
stateOrProvinceName = match
organizationName = match
organizationalUnitName = match
localityName = optional
commonName = supplied
emailAddress = optional
```

次のコマンドを実行して、秘密鍵を生成します。

```
 $ cd /root/ca
$ sudo openssl genrsa -out private/ca.key
```

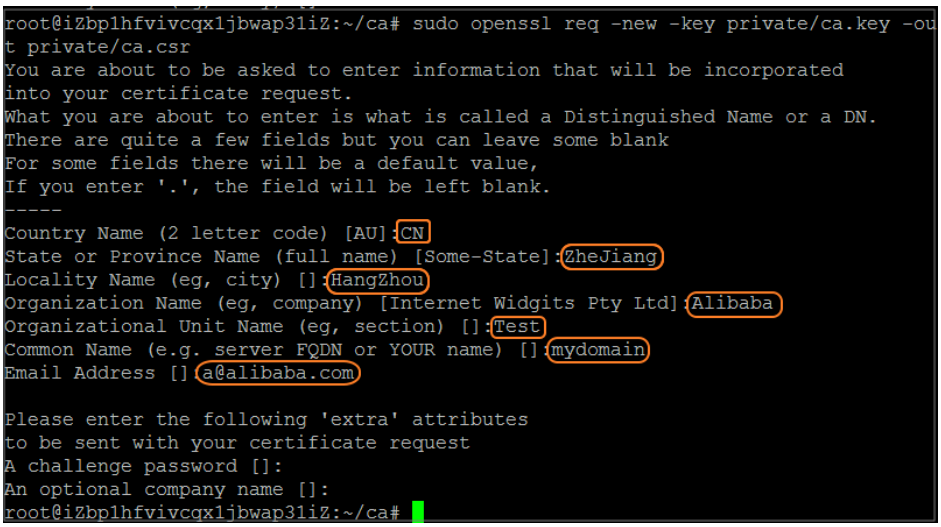
次の図は、鍵生成の例です。



次のコマンドを実行し、プロンプトに従って必要な情報を入力します。Enterを押して、証明書を生成するために使用されるcsrファイルを生成します。

$ sudo openssl req -new -key private/ca.key -out private/ca.csr

> **注意**: Common Nameの値のようなSLBインスタンスのドメイン名を入力します。



次のコマンドを実行して、crtファイルを生成します。

$ sudo openssl x509 -req -days 365 -in private/ca.csr -signkey private/ca.key -out private/ca.crt

次のコマンドを実行して、秘密鍵の開始シーケンス番号を4文字で設定します。

$ sudo echo FACE > serial

次のコマンドを実行して、CAキーライブラリを作成します。

$ sudo touch index.txt

次のコマンドを実行して、クライアント証明書を削除するための証明書失効リストを作成します。

$ sudo openssl ca -gencrl -out /root/ca/private/ca.crl -crldays 7 -config "/root/ca/conf/openssl.conf"

応答は次のとおりです：

Using configuration from /root/ca/conf/openssl.conf

Step 3: クライアント証明書を生成する

次のコマンドを実行して、クライアント証明書を格納する caフォルダの下にusersフォルダを生成します。

$ sudo mkdir users

次のコマンドを実行して、クライアント証明書のキーを作成します。

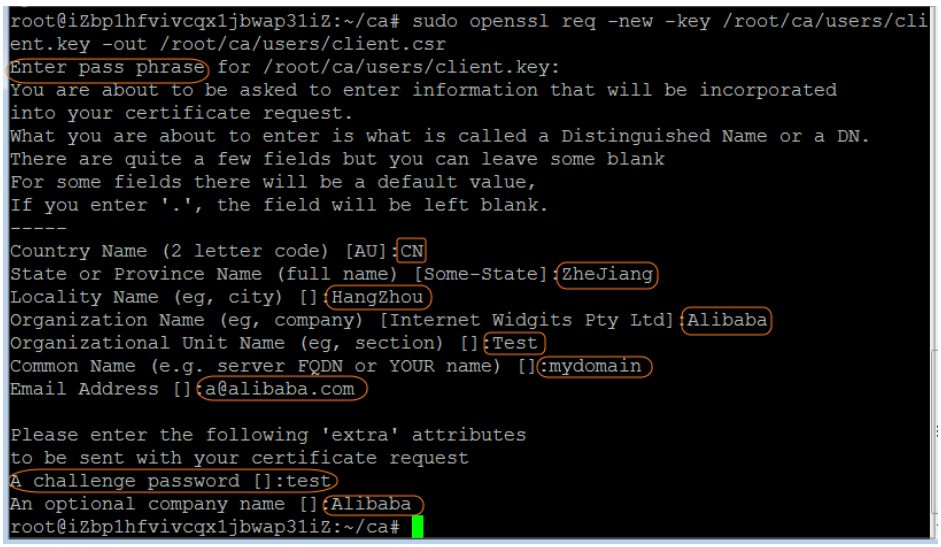
$ sudo openssl genrsa -des3 -out /root/ca/users/client.key 1024

注意: 入力されたパスフレーズは、このキーのフレーズです。

次のコマンドを実行して、証明書の署名を要求するための csrファイルを作成します。

$ sudo openssl req -new -key /root/ca/users/client.key -out /root/ca/users/client.csr

指示に従って、前の手順で設定したパスフレーズを入力します。



次のコマンドを実行して証明書に署名します。

$ sudo openssl ca -in /root/ca/users/client.csr -cert /root/ca/private/ca.crt -keyfile /root/ca/private/ca.key -out /root/ca/users/client.crt -config "/root/ca/conf/openssl.conf"

操作を確認するメッセージが表示されたら、 y を入力します。

```
root@iZbp1hfvivcqx1jbwap31iZ:~/ca# sudo openssl ca -in /root/ca/users/client.csr
 -cert /root/ca/private/ca.crt -keyfile /root/ca/private/ca.key -out /root/ca/us
ers/client.crt -config "/root/ca/conf/openssl.conf"
Using configuration from /root/ca/conf/openssl.conf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'CN'
stateOrProvinceName   :ASN.1 12:'ZheJiang'
localityName          :ASN.1 12:'HangZhou'
organizationName      :ASN.1 12:'Alibaba'
organizationalUnitName:ASN.1 12:'Test'
commonName            :ASN.1 12:'mydomain'
emailAddress          :IA5STRING:'a@alibaba.com'
Certificate is to be certified until Jun  4 15:28:55 2018 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
root@iZbp1hfvivcqx1jbwap31iZ:~/ca#
```

次のコマンドを実行して、証明書をほとんどのブラウザで認識可能な PKCS12ファイルに変換します。

$ sudo openssl pkcs12 -export -clcerts -in /root/ca/users/client.crt -inkey /root/ca/users/client.key -out /root/ca/users/client.p12

プロンプトが表示されたら、クライアントキーのパスワードを入力します。

クライアント証明書のエクスポートに使用するパスワードを入力します。

```
root@iZbp1hfvivcqx1jbwap31iZ:~/ca# sudo openssl pkcs12 -export -clcerts -in /roo
t/ca/users/client.crt -inkey /root/ca/users/client.key -out /root/ca/users/clien
t.p12
Enter pass phrase for /root/ca/users/client.key:
Enter Export Password:
Verifying - Enter Export Password:
root@iZbp1hfvivcqx1jbwap31iZ:~/ca#
```

作成した証明書を表示するには、次のコマンドを実行します。

```
cd users
ls
```

```
root@iZbp1hfvivcqx1jbwap31iZ:~/ca# cd users
root@iZbp1hfvivcqx1jbwap31iZ:~/ca/users# ls
client.crt  client.csr  client.key  client.p12
root@iZbp1hfvivcqx1jbwap31iZ:~/ca/users#
```

Step 4: サーバー証明書とCA証明書をアップロードする

[Server Load Balancerコンソール]にログインします。

左側のナビゲーションペインで、**[証明書]** をクリックし、 **[証明書の作成]**をクリックします。

次のようにサーバー証明書を構成します。

証明書リージョン ：**中国東部1（杭州）**を選択します。

注意: 証明書のリージョンは、Server Load Balancerインスタンスと同じである必要があります。

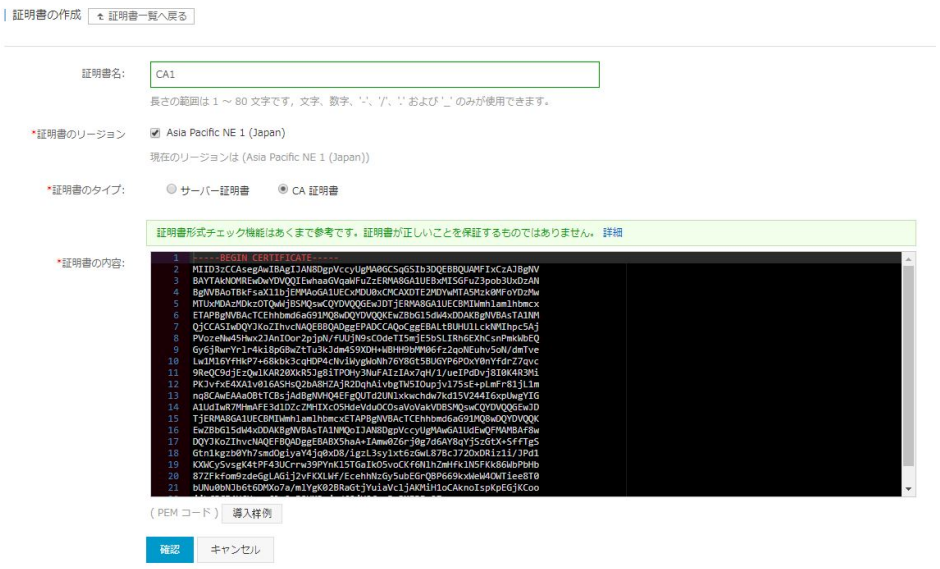
証明書のタイプ：**サーバー証明書**を選択します。

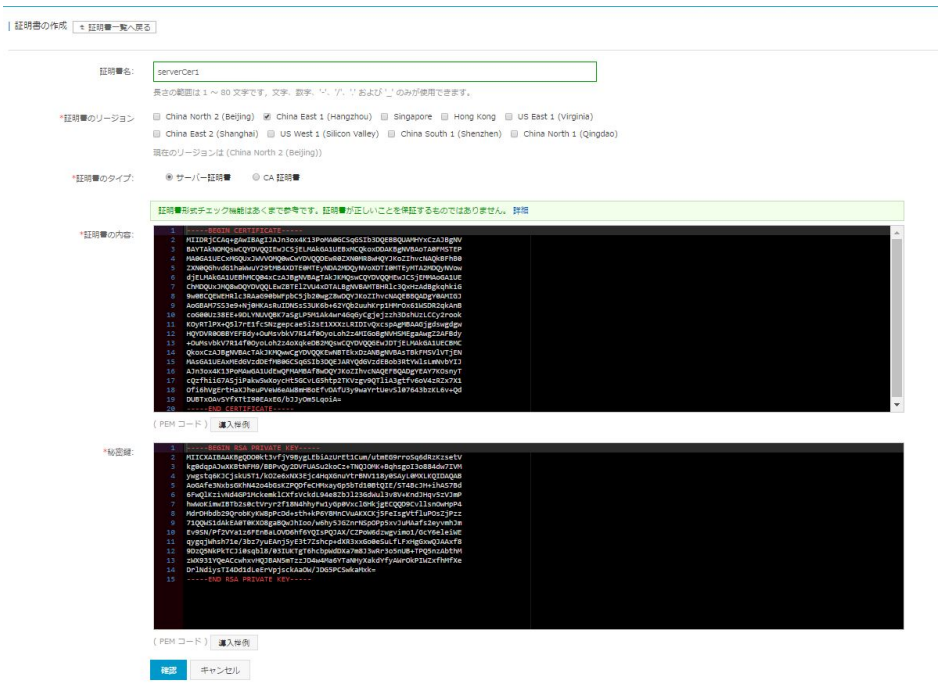
証明書の内容と秘密鍵：サーバー証明書の内容とサーバー証明書の秘密キーをコピーします。**サンプルのインポート**をクリックして、証明書の有効な形式を表示できます。

証明書の形式については、証明書形式要件を参照してください。

確認をクリックします。

前の手順を繰り返して、CA証明書をアップロードします。





Step 5: ライアント証明書をインストールする

Git Bashコマンドラインを開き、次のコマンドを実行してクライアント証明書をエクスポートします。

scp root@IPaddress:/root/ca/users/client.p12 ./

注意：IPaddressは、クライアント証明書が生成されるサーバのIPです。

クライアント証明書をインストールします。IEウェブブラウザを例とします：

IE Webブラウザを開き、**設定>インターネットオプション**を選択します。

コンテンツタブをクリックし、**証明書**をク リックします。

PKCS12ファイルをインポートします。

Step 6: サーバー・ロード・バランサのインスタンスを構成する

[Server Load Balancerコンソール]にログインします。

インスタンス管理ページで、 **Server Load Balancerの作成**をクリックします。

インスタンスを構成し、**[今すぐ購入]**をクリックします。

> **注意**: インスタンスリージョンとして China East 1 を選択し、ネットワークタイプとして**インターネット**を選択します。詳細は、Server Load Balancer インスタンスの作成を参照してください。

クリックし、ページ**インスタンス管理**に戻り、**中国東部1** リージョンを選択します。

Server Load BalancerインスタンスのIDをクリックします。

左側のナビゲーションペインで、 **インスタンスリスナー** をクリックし、 **リスナーの作成** をクリックします。

次のようにリスナーを構成します：

フロントエンドプロトコルとポート：HTTPS 443。

バックエンドプロトコルとポート：HTTP 80。

スケジューリングアルゴリズム：ラウンドロビン。

相互認証：有効。

サーバー証明書：アップロードされたサーバー証明書を選択します。

CA証明書：アップロードされたクライアント証明書を選択します。

[次のステップ]をクリックします。

デフォルトのヘルスチェック設定を使用し、**確認**をクリックして終了します。

左側のナビゲーションペインで、 **バックエンドサーバー** > **バックエンドサーバー** をクリックして、ECSインスタンスを追加します。

バックエンドサーバーの詳細については、バックエンドサーバーの概要を参照してください。

Step 7: 負荷分散サービスをテストする

WebブラウザにServer Load Balancerインスタンスのパブリックを入力し、要求がSSLプロトコルで正しく処理されているかどうかを確認します。

HTTP を HTTPS にリダイレクト

HTTPS は、HTTP のセキュア版です。HTTPS では、ブラウザとサーバーの間を流れるデータは暗号化されます。Server Load Balancer はサービスを妨げることなく、HTTP リクエストを HTTPS にリダイレクトします。

現在、HTTP のリダイレクト機能は、全てのリージョンでご利用いただけます。

前提条件

HTTPS リスナーを作成していること。詳細は、**HTTPS リスナーの設定** を参照のこと。

手順

Server Load Balancer コンソール にログインします。

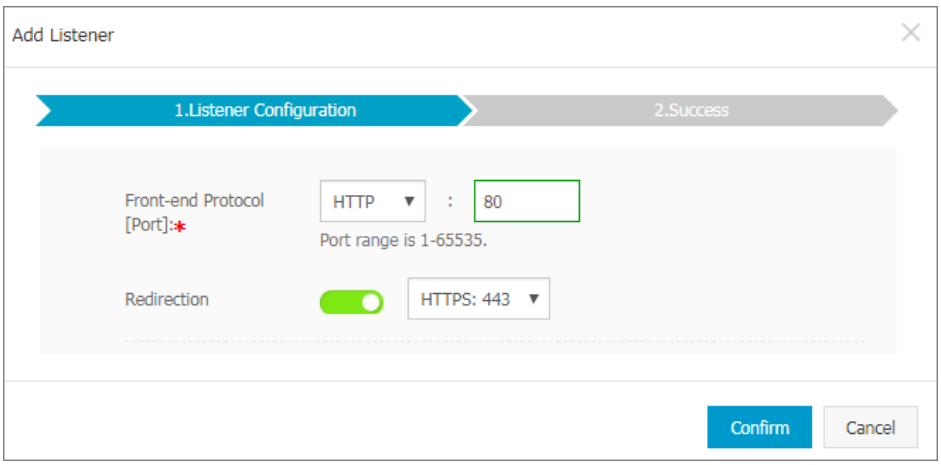
トップメニューより、SLB インスタンスの置かれているリージョンを選択します。

インスタンス 画面で、SLB インスタンスの ID をクリックします。

左ナビゲーション メニューにある、**リスナー**をクリックし、さらに**リスナー追加**をクリックします。

フロントエンド プロトコルとして **HTTP** を選択し、フロントエンド ポートには **80** を入力します。

リダイレクト機能を有効にし、**確認**をクリックします。



リダイレクト機能が有効になった後は、HTTP リクエストはすべて HTTPS リスナーにリダイレクトされ、HTTPS リスナーの設定に基いて割り振られます。

ドメイン名またはURL転送ルールの構成

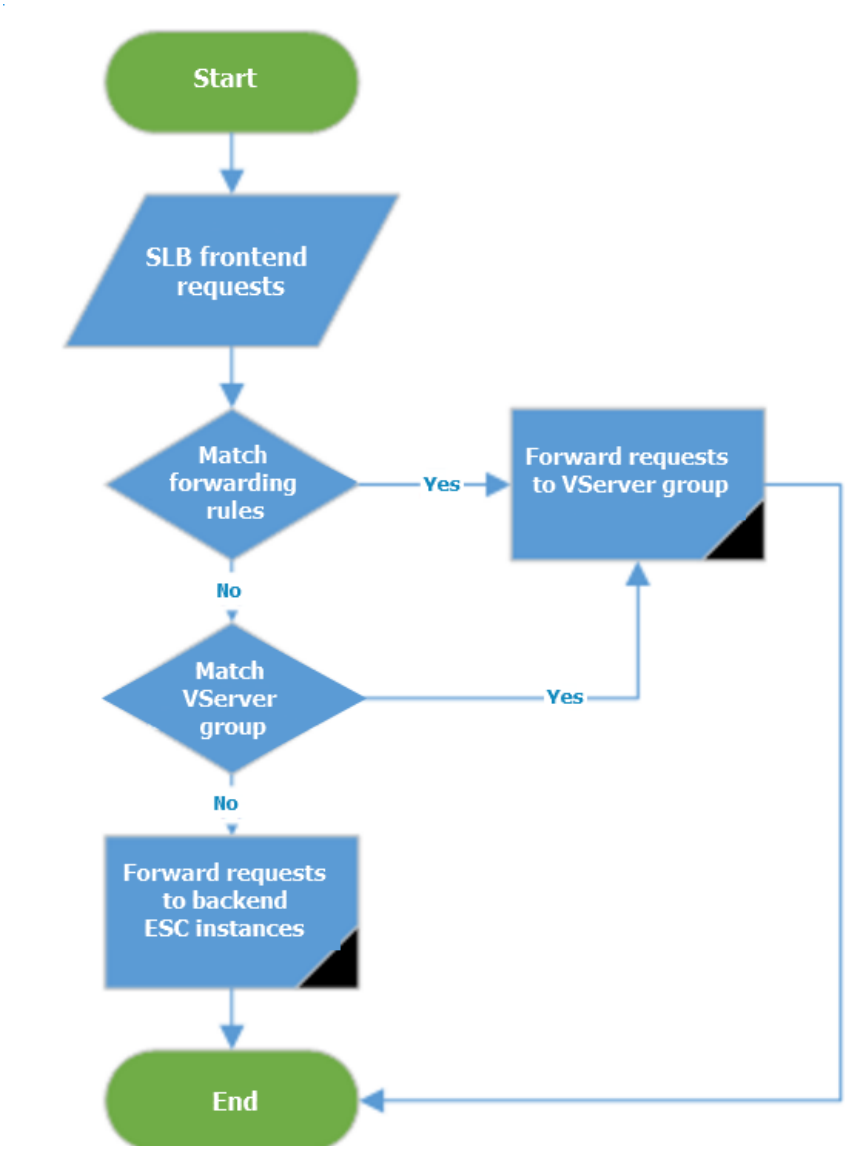
レイヤー7 Server Load Balancerは、異なる要求を異なるバックエンドサーバーに配布するために、ドメイン名または要求URLに基づく転送ルールの構成をサポートしています。レイヤ7リスナに対して、異なるVServerグループに関連付けられた異なる転送ルールを追加できます。

転送ルールを設定すると、システムはクライアントリクエストが設定された転送ルールと一致するかどうかをチェックします。

一致した場合、要求は転送ルールで指定されたVServerグループに転送されます。

そうでない場合、要求はリスナーで設定されたVServerグループに転送されます。

リスナーがVServerグループを使用していない場合、要求はサーバープールに追加されたバックエンドサーバーに転送されます。



ドメイン名とURL転送ルールの設定

ドメイン名、URL、またはその両方に基づいて転送ルールを構成できます。

ドメイン名のみ

ドメイン名に基づいて転送ルールを設定する場合は、URLフィールドを省略します。ドメイン名については、文字、数字、ダッシュ（ - ）、ピリオド（.）のみが許可されます。ドメイン名は、完全一致とワイルドカード一致の両方をサポートしています。例えば：

正確なドメイン名：www.aliyun.com

ワイルドカードドメイン名： *.aliyun.com*、 .market.aliyun.com

フロントエンド要求が複数のドメイン名ルールに同時に一致する場合、優先順位は次の表のようになります。

√は対応するルールが一致していることを表します。

モード	リクエストURL	ドメイン名		
		www.aliyun.com	*.aliyun.com	*.market.aliyun.com
完全一致	www.aliyun.com	√		
ワイルドカードマッチ	market.aliyun.com		√	
ワイルドカードマッチ	info.market.aliyun.com			√

URLのみ

要求パスに基づいて転送ルールを構成する場合は、ドメイン名フィールドを省略します。URL転送ルールを設定するときは、次のルールに注意してください。

- 文字、数字、ダッシュ（ - ）、スラッシュ（/）、ピリオド（%）、疑問符（?）、ハッシュキー（#）、アンパサンド（&）のみが許可されます。
- URLはスラッシュ（/）で始まる必要がありますが、スラッシュ（/）のみで構成することはできません。スラッシュ（/）のみを入力すると、URL転送ルールが正しく適用されます。

ドメイン名とURL

同じドメイン名ではなく異なるパスに基づいてトラフィックを転送する場合は、ドメイン名とURLを使用して転送ルールを構成できます。一致するURLがないためにアクセスエラーが発生しないように、ドメイン専用の転送ルールを設定することをおすすめします。

例えば、 *www.aaa.com* と *www.bbb.com* の2つのドメイン名があります。要件は、*www.aaa.com/index.html* からの要求をServerGroup1に転送し、 *xxx.html* からServerGroup2に他の要求を転送することです。次の転送ルールを設定する必要があります。*rule2* がなければ、www.aaa.comからのリクエストに404エラーが発生します。

ドメイン名とURL転送ルールを設定する

前提条件

レイヤー7（HTTP / HTTPS）リスナーを作成しました。そうでない場合は、**リスナーの概要**を参照してください。

VServerグループを作成しました。そうでない場合は、**VServerグループを作成する**を参照してください。

手順

Server Load Balancerコンソールにログオンします。

[**インスタンス管理**]をクリックし、対象のServer Load BalancerインスタンスのIDをクリックします。

左側のナビゲーションペインで、[**インスタンスリスナー**]をクリックします。

ターゲットのHTTP / HTTPSリスナーを見つけて、[**転送ルールの作成**]をクリックします。

[**ルール**]ダイアログで、[**転送ルールの作成**]をクリックします。

[**転送ルールの作成**]ダイアログで、ルールを設定して[**確認**]をクリックします。

[**転送ルールの作成**]をクリックして、別のルールを追加します。

ヘルスチェック

ヘルスチェック概要

Server Load Balancer は、ヘルスチェックを実行して、バックエンドサーバー（ECS インスタンス）のサービスの可用性をチェックします。

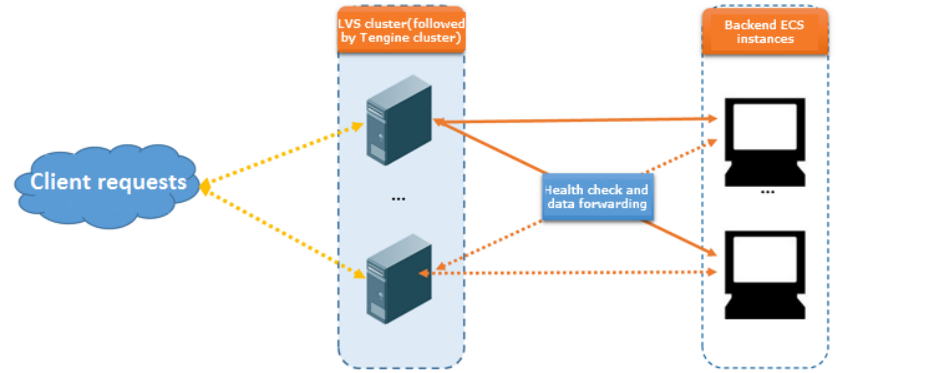
Server Load Balancer は正常なインスタンスにのみリクエストを配信します。Server Load Balancer は、インスタンスが正常でないことを検出すると、そのインスタンスへのリクエストの分散を停止し、正常な状態に復元されたときにインスタンスへのリクエストの分散を再開します。したがって、ヘルスチェックメカニズムにより、フロントエンドサービスの全体的な可用性が向上し、バックエンドサーバーが異常な場合のサービス可用性への影響が軽減されます。

ビジネスがトラフィック負荷に非常に敏感である場合、頻繁なヘルスチェックの検出が通常のサービスアクセスに影響する可能性があります。ヘルスチェックの頻度を減らしたり、ヘルスチェック間隔を長くしたり、HTTP ヘルスチェックをTCP ヘルスチェックに変更することで、影響を減らすことができます。サービスの可用性を確保するため、ヘルスチェックを停止することはお勧めしません。

ヘルスチェックプロセス

サーバーロードバランサはクラスタに展開されます。データ転送とヘルスチェックは、LVS クラスタと Tengine クラスタのノードサーバーによって同時に処理されます。

クラスタ内のノード・サーバーは、ヘルス・チェック構成に従って、独立してヘルス・チェックを実行します。ノードサーバーがヘルスチェック後に ECS インスタンスが正常でないことを検出すると、ノードサーバーは ECS インスタンスへのリクエストの配信を停止します。この操作は、すべてのノードサーバーで同期され、他のノードには影響しません。



ヘルスチェックを実行するために使用される IP アドレスの範囲は 100.64.0.0/10 です。バックエンド

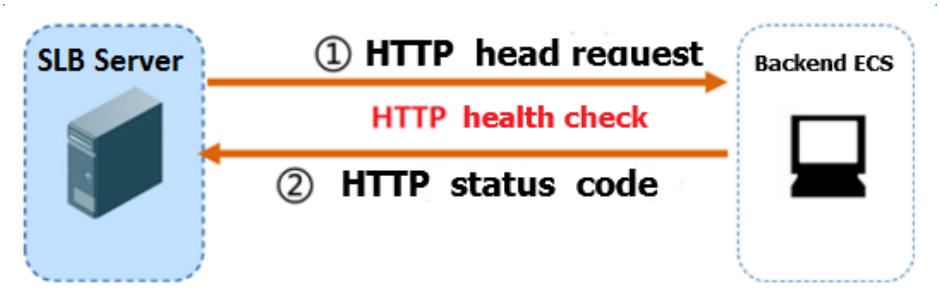
ECS インスタンスが IP テーブル転送またはその他のアクセス制御を有効にする場合は、この IP アドレス範囲へのアクセスを許可します。

ヘルスチェックの仕組み

HTTP/HTTPS リスナー

レイヤー7（HTTP または HTTPS）リスナーの場合、Server Load Balancer は、次の図に示すように、HTTP HEAD リクエストを送信してバックエンドサーバーの状態を検出します。

> HTTPS リスナーの場合、証明書は Server Load Balancer システムで管理されます。Server Load Balancer とバックエンド ECS インスタンス間のデータ転送（ヘルスチェックとデータ転送を含む）は、HTTPS を介して送信されなくなり、システムパフォーマンスが向上します。



ヘルスチェックのメカニズムは次のとおりです。

リスナーのヘルスチェック設定に従って、Tengine ノードサーバーは、HTTP HEAD リクエストを**ECS 内部 IP：ヘルスチェックポート**のバックエンド ECS インスタンスに送信します。

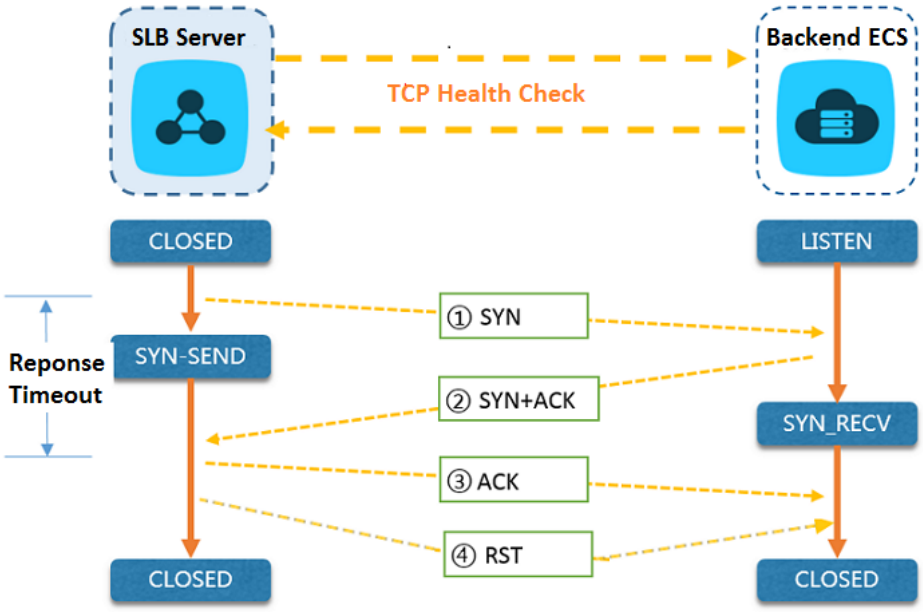
リクエストを受信した後、バックエンド ECS サーバーは、実行状態に基づいて HTTP ステータスコードを返します。

Tengine ノードサーバーが、構成された応答タイムアウト期間内にバックエンド ECS インスタンスからの応答を受信しない場合、ECS インスタンスは異常とみなされます。

Tengine ノードサーバーが、構成された応答タイムアウト期間内にバックエンド ECS インスタンスからの応答を受信し、返された HTTP ステータスコードが指定されたステータスコードである場合、ECS インスタンスは正常とみなされます。

TCPリスナー

TCP リスナーの場合、Server Load Balancer は、次の図に示すように、TCP 検出を送信してバックエンドサーバーの状態を検出します。



ヘルスチェックのメカニズムは次のとおりです。

リスナーのヘルスチェック設定に従って、LVS ノードサーバーは TCP SYN データパケットを**ECS 内部 IP：ヘルスチェックポート**のバックエンド ECS インスタンスに送信します。

リクエストを受信した後、バックエンド ECS インスタンスは、対応するポートが正常にリスニングしている場合は、TCP SYN および ACK パケットを返します。

LVS ノードサーバーが、構成された応答タイムアウト期間内にバックエンド ECS インスタンスから必要なデータパケットを受信しない場合、ECS インスタンスは異常とみなされ、LVS ノードサーバーは RST データパケットをバックエンド ECS サーバーに送信して TCP 接続します。

LVS ノードサーバーが、構成された応答タイムアウト期間内にバックエンド ECS インスタンスからデータパケットを受信すると、ECS インスタンスは正常と見なされ、LVS ノードサーバーは RST データパケットをバックエンド ECSサーバーに送信して TCP 接続を終了します。

注意：一般的に、TCP 接続を確立するために、TCP 3ウェイハンドシェイクが行われます。LVSノードサーバがバックエンド ECS インスタンスから SYN + ACK データパケットを受信すると、LVS ノードサーバは ACK データパケットを送信し、直ちに RST データパケットを送信して TCP 接続を終了します。

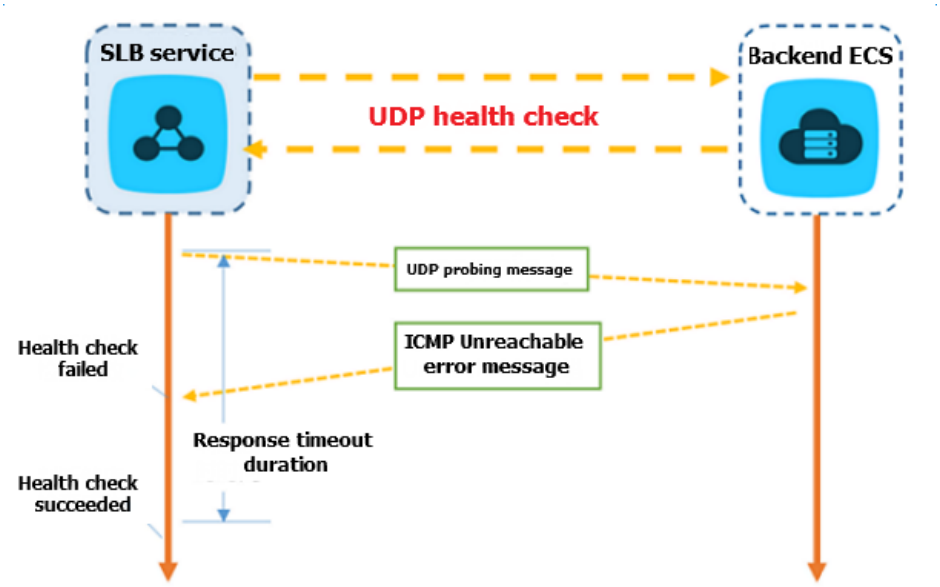
このメカニズムにより、バックエンド ECS インスタンスは、異常終了などの TCP 接続で発生したエラーを考慮し、**対応リセット**のような対応するエラーメッセージをスローする可能性があります。

ソリューション：

- HTTP ヘルスチェックを使用します。
- 実 IP 取得機能を有効にした後、ヘルスチェック IP に起因する接続エラーを無視できます。

UDP リスナー

UDP リスナーの場合、Server Load Balancer は、次の図に示すように、UDP パケット検出を介してバックエンドサーバーの状態を検出します。



ヘルスチェックのメカニズムは次のとおりです。

LVS ノード・サーバーは、ヘルス・チェック構成に従って**ECS 内部 IP：ヘルス・チェック・ポート**にある ECSインスタンスに UDP パケットを送信します。

ECS インスタンスの対応するポートが正常にリスニングしていない場合、システムは `port XX unreachable ‘のような ICMP エラーメッセージを返します。それ以外の場合は、メッセージは送信されません。

構成された応答タイムアウト期間内に LVS ノードサーバーが ICMP エラーメッセージを受信すると、ECS インスタンスは異常とみなされます。

LVS ノードサーバーが構成された応答タイムアウト時間内にメッセージを受信しない場合、ECS インスタンスは正常とみなされます。

注意：UDP ヘルスチェックの場合、以下の状況では、実際の ECS インスタンスのステータスとヘルスチェックの結果が異なる場合があります。

ECS インスタンスが Linux オペレーティングシステムを使用する場合、Linux の anti-ICMP 攻撃防御メカニズムは、大規模な同時シナリオで ICMP メッセージを送信する速度を制限します。ECS インスタンスが異常であっても、ICMPメッセージが受信されないため、Server Load Balancer は ECS インスタンスを正常に宣言することがあります。

ソリューション：コンソールで UDP ヘルスチェックのカスタムリクエストと応答の組を設定できます。次に、カスタムリクエストを使用してヘルスチェックを行うことができます。指定されたカスタム応答が返された場合、ECS インスタンスは正常とみなされます。それ以外の場合、ECS インスタンスは異常とみなされます。

ヘルスチェックステータスウィンドウ

ヘルスチェックの失敗に起因する頻繁なECSスイッチによるシステム可用性への影響を軽減するため、Server Load Balancer は、一定時間内に連続して成功または失敗した場合にのみ、ECS インスタンスを正常または異常に設定します。その時間枠はいくつかの要素に依存しています。

- 間隔（ヘルスチェックが実行される頻度）
- タイムアウト（応答を待つ時間）
- Threshold（成功または正常終了したヘルスチェックの連続回数）

正常性チェックウィンドウは次のように計算されます。

- 正常状態チェックウィンドウ=正常しきい値x正常チェック間隔
- 異常な状態チェックウィンドウ=（状態チェック間隔+応答タイムアウト）×異常しきい値

デフォルトでは、TCP/HTTP/HTTPS リスナーの正常状態チェックウィンドウは6秒で、異常状態チェックウィンドウは 21 秒です。

UDP リスナーの場合、正常状態チェックウィンドウは 15 秒で、異常状態チェックウィンドウは 45 秒です。

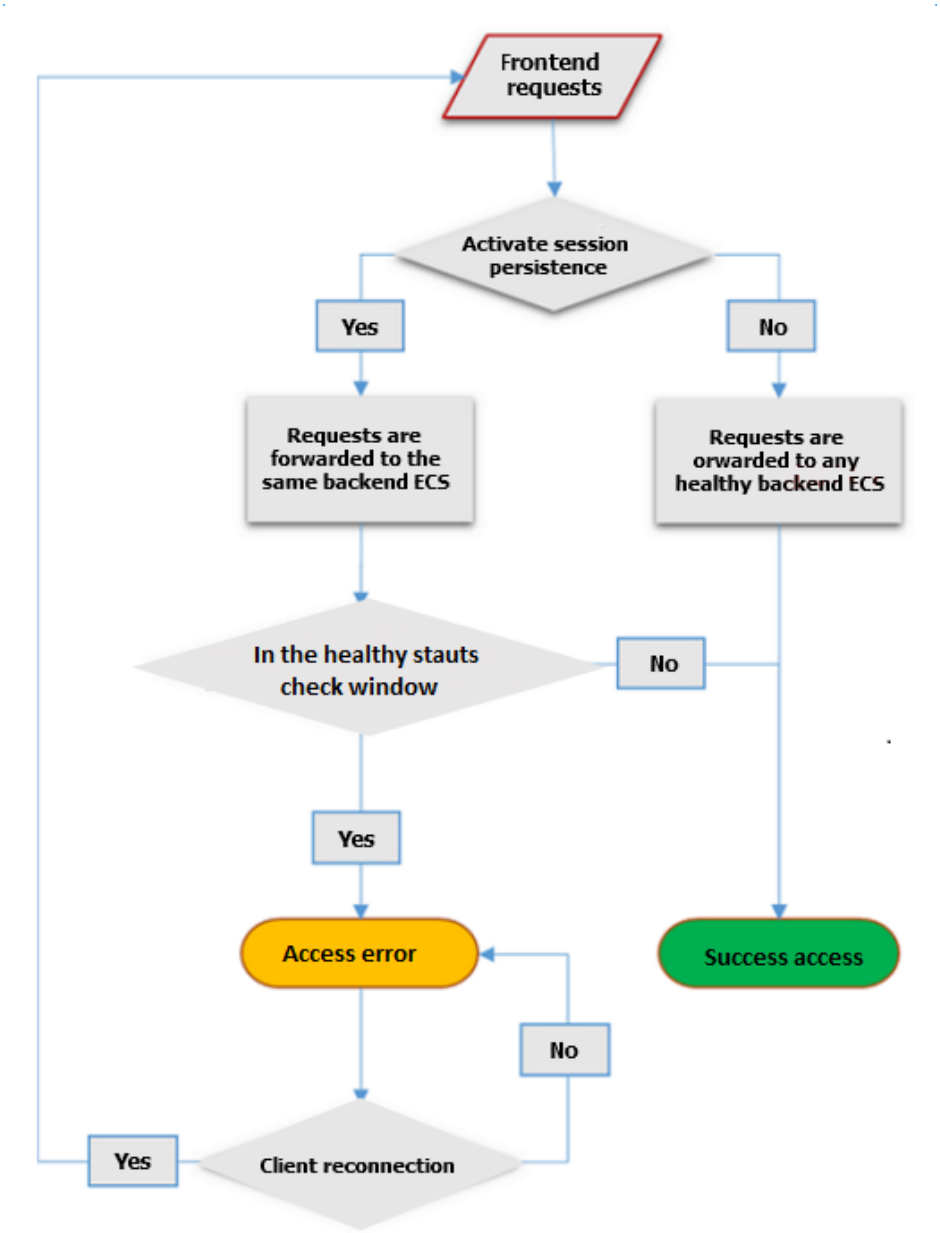
リクエストがサーバー負荷分散装置に到着した場合：

対象の ECS インスタンスのヘルスチェックが失敗した場合、そのサービスを復元するまで、リクエストはその ECS インスタンスに配信されません。

ターゲット ECS インスタンスのヘルスチェックが成功した場合、リクエストはそのヘルスチェックに配布されます。

異常な状態チェックウィンドウ中に要求が到着した場合、ECS インスタンスがチェックされており、異常であると宣言されていないため、リクエストは依然として ECS インスタンスに送信されます。正常状態チェックウィンドウ中にリクエストが到着した場合、ECS インスタンスが正常に宣言されると、サービスの可用性に影響はありません。デフォルトのヘルスチェック設定を使用するこ

とをお勧めします。



ヘルスチェックの設定

ヘルスチェックを設定する

リスナーのヘルスチェックは、コンソールまたは API を使用して構成できます。詳細については、ヘルスチ

ェック概要およびヘルスチェックのFAQを参照してください。

> **注**：TCP リスナーの場合は、TCP ヘルスチェックと HTTP ヘルスチェックの両方がサポートされています。

インスタンスの詳細ページで、**インスタンスリスナー** > **リスナーの作成**をクリックし、2 番目の手順でヘルスチェックを設定します。

ヘルスチェックのオプション

ヘルスチェックを設定するときは、デフォルト値を使用することをお勧めします。

オプション	説明
ドメイン名とヘルスチェック URL （HTTP ヘルスチェックのみ）	デフォルトで、 Server Load Balancer は、ヘルスチェックを行うために、バックエンド ECS インスタンスのイントラネット IP アドレスを使用して、アプリケーションサーバー上に構成されたデフォルトのホームページに HTTP ヘッドリクエストを送信します。 ヘルスチェックにアプリケーションサーバーのデフォルトホームページを使用しない場合は、ヘルスチェックの URL を指定する必要があります。

	一部のアプリケーションサーバーは要求のホストフィールドを確認するため、要求ヘッダーにホストフィールドが含まれている必要があります。ヘルスチェックでドメイン名が設定されている場合、 Server Load Balancer は、要求をバックエンドサーバーに転送するときに、ドメイン名をホストフィールドに追加します。ドメイン名が設定されていない場合、 Server Load Balancer によって転送された要求には、要求ヘッダーにホストフィールドが含まれません。その場合、ヘルスチェック要求はサーバーによって拒否され、ヘルスチェックは失敗する可能性があります。したがって、アプリケーションサーバーが要求のホストフィールドを検証する場合は、ヘルスチェックが機能するようにドメイン名を構成する必要があります。
通常ステータスコード （HTTP ヘルスチェックのみ）	ヘルスチェックが正常であることを示す HTTP ステータスコードを選択します。 デフォルト値：http_2xx および http_3xx。
ヘルスチェックポート	バックエンド ECS インスタンスにアクセスするためにヘルスチェックによって使用される検出ポート。 デフォルトでは、リスナーで設定されたバックエンドポートが使用されます。 **注：**リスナーが VServer グループまたはマスター / スレーブサーバーグループに関連付けられ、グループ内の ECS インスタンスが異なるポートを使用する場合は、このオプションを空のままにします。 Server Load Balancer は、各 ECS インスタンスのバックエンドポートを使用してヘルスチェックを行います。
応答タイムアウト	ヘルスチェックからの応答を待機する時間。ECS インスタンスが指定されたタイムアウト時間内に応答を送信しない場合、ヘルスチェックは失敗します。 有効な値：1 ～ 300 秒。デフォルト値は、UDP リスナーの場合は 10 秒、HTTP/HTTPS/TCP リスナーの場合は 5 秒です。
ヘルスチェックの間隔	連続した 2 回の健康診断の時間間隔。 LVS クラスタ内のすべてのノードサーバーは、間隔に応じて独立して同時に、バックエンド ECS インスタンスのヘルスチェックを実行します。各ノードサーバーのヘルスチェック時間が同期していないため、1 つの ECS インスタンスのヘルスチェック要求の統計にヘルスチェック間隔が反映されません。 有効な値は 1 ～ 50 秒です。デフォルト値は、UDP リスナーの場合は 5 秒、HTTP/HTTPS/TCP リスナーの場合は 2 秒です

	。
異常状態しきい値	同一の ECS インスタンス上で同じ LVS ノードサーバーによって実行された正常性チェックの連続失敗の回数（成功から失敗） 有効な値は 2 ～ 10 です。デフォルト値は 3 です。
正常状態しきい値	同じECSインスタンス上で同じLVSモード・サーバーによって実行されたヘルス・チェックの連続成功回数（失敗から成功まで）。 有効な値は 2 ～ 10 です。デフォルト値は 3 です。
ヘルスチェック要求と応答	UDP リスナーのヘルスチェックを設定するときは、 **ヘルスチェックレスポンス** にリクエストの内容（youraccountID など）を入力し、 **ヘルスチェックレスポンス** に期待されるレスポンス（slb123など）を入力します。対応するヘルスチェック応答ロジックをバックエンドサーバーのアプリケーションロジックに追加します。たとえば、 youraccountIDを受け取ったときにslb123を返します。 Server Load Balancerがバックエンドサーバーから期待される応答を受信すると、ヘルスチェックは成功します。それ以外の場合、ヘルスチェックは失敗します。この方法は、健康診断の信頼性を最大限に保証することができる。

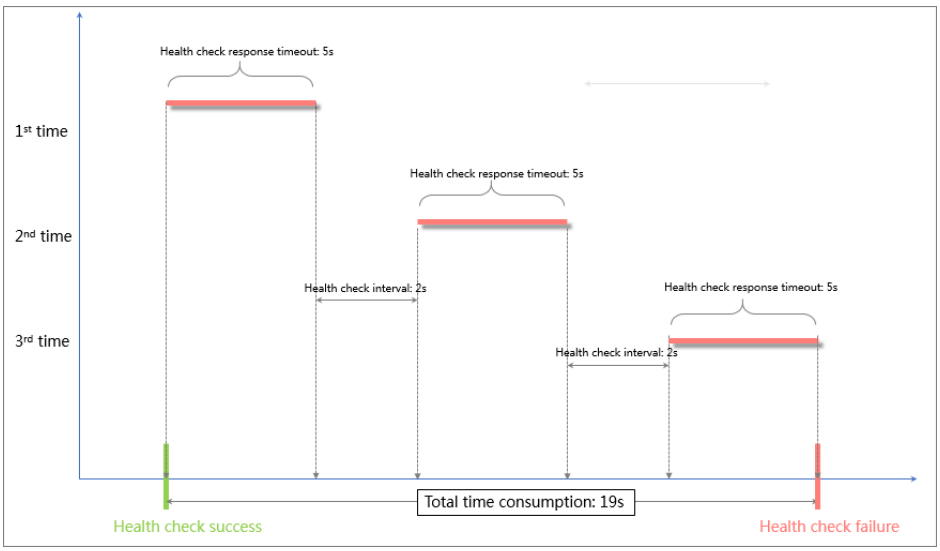
ヘルスチェック応答タイムアウトとヘルスチェック間隔の例

例として、次のヘルスチェック設定を行います。

- 応答タイムアウト：5 秒
- ヘルスチェック間隔：2 秒
- 正常な閾値：3 回
- 異常な閾値：3 回

ヘルスチェック失敗時間ウィンドウ=応答タイムアウト×異常状態しきい値+ヘルスチェック間隔×（異常状態しきい値 - 1）。すなわち、5x3 + 2x（3-1）= 19s。

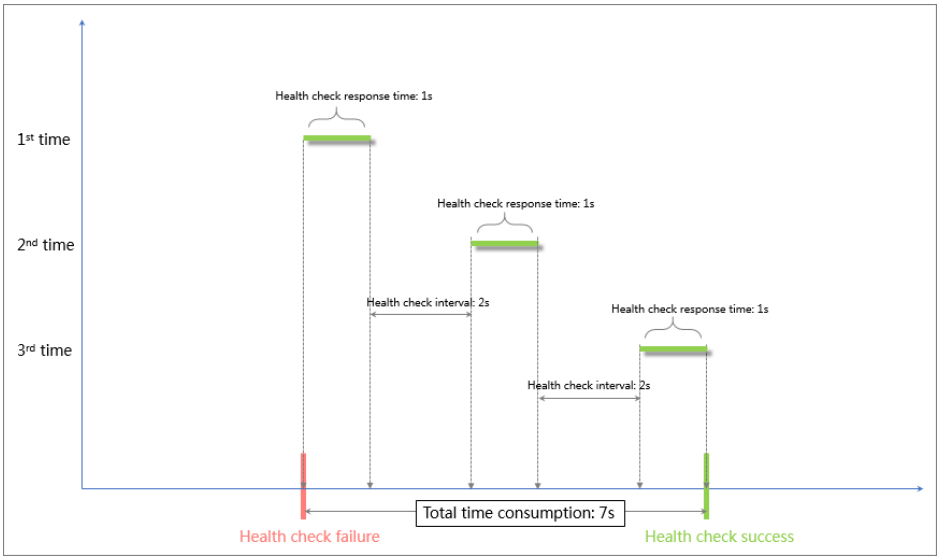
次の図は、正常でないバックエンドサーバーを宣言するプロセスを示しています。



ヘルスチェック成功時間ウィンドウ=ヘルスチェック応答時間×正常状態しきい値+ヘルスチェック間隔×（正常状態しきい値 -1）。すなわち、（1x3）+ 2x（3-1）= 7sである。

> **注**：**ヘルスチェック応答時間**は、正常な要求 - 応答メッセージが送受信される時間です。TCP ヘルスチェックが使用されている場合、Server Load Balancer はポートが開いているかどうかをチェックするだけなので、応答時間は非常に短く、ほとんど無視できます。HTTP ヘルスチェックを使用する場合、アプリケーションサーバーのパフォーマンスと負荷に応じて、通常は応答時間が秒単位で表示されます。

次の図は、健全なバックエンドサーバーを宣言するプロセスを示しています。



HTTP ヘルスチェックにおけるドメイン名の設定

HTTP ヘルスチェックを使用する場合は、ヘルスチェックのドメイン名を設定できますが、必須オプションではありません。一部のアプリケーションサーバーは要求のホストフィールドを確認するため、要求ヘッダ

ーにホストフィールドが含まれている必要があります。ヘルスチェックでドメイン名が設定されている場合、 Server Load Balancer は、要求をバックエンドサーバーに転送するときに、ドメイン名をホストフィールドに追加します。ドメイン名が設定されていない場合、 Server Load Balancer によって転送された要求には、要求ヘッダーにホストフィールドが含まれません。その場合、ヘルスチェック要求はサーバーによって拒否され、ヘルスチェックは失敗する可能性があります。したがって、アプリケーションサーバーが要求のホストフィールドを検証する場合は、ヘルスチェックが機能するようにドメイン名を構成する必要があります。

ヘルスチェックの無効化

ヘルスチェックが無効化された場合、リクエストが問題発生の ECS インスタンスに転送され、サービスの中断を招く可能性があります。一般的に、ヘルスチェックを無効化することはお勧めしません。

> **注意：** HTTP および HTTPS リスナーのヘルスチェックのみを無効化することができます。UDP リスナーと TCP リスナーのヘルスチェックは無効できません。

手順

Server Load Balancer コンソールにログインします。

「**インスタンス**」ページで、ターゲットインスタンスのIDをクリックします。

左側のナビゲーションペインで、**インスタンスリスナー**をクリックします。

ターゲットリスナーの横にある**設定**をクリックします。

リスナーの設定ダイアログボックスで、**次へ**をクリックします。

トグル**「ヘルスチェックを開く」**を**無効化**にします。

バックエンドサーバー

バックエンドサーバーの概要

バックエンドサーバとは何ですか？

バックエンドサーバーは、サーバー負荷分散装置インスタンスに追加するECSインスタンスで、分散要求を処理するために使用されます。

> **注意**：いつでもバックエンドECSインスタンスの数を増減できます。ただし、ヘルスチェック機能を有効にすることをお勧めします。サービスの安定性を維持するには、Sever Load Balancerインスタンスに少なくとも1つの通常のECSが存在する必要があります。

Server Load BalancerインスタンスにECSインスタンスを追加する場合は、次の点に注意してください。

Server Load Balancerは領域横断の展開をサポートしておらず、ECSインスタンスとServer Load Balancerインスタンスの領域が同じであることを確認します。

Server Load Balancerは、ECSインスタンスにデプロイされたアプリケーションが同じで、データが一貫している限り、ECSインスタンスで使用されるオペレーティングシステムを制限しません。メンテナンスのために、同じオペレーティングシステムを使用することをお勧めします。

最大50のリスナーをServer Load Balancerインスタンスに追加できます。各リスナーは、ECSにデプロイされたアプリケーションに対応しています。

追加されたECSインスタンスごとにウェイトを設定できます。高い重みのECSインスタンスは、より多くの接続要求を受信します。ECSインスタンスのサービス機能に基づいて重みを設定できます。

注意：セッション持続機能を有効にした場合、バックエンドECSへの分散要求は不均衡になる可能性があります。その場合は、セッション永続化機能をしばらく閉じて、問題がまだ存在するかどうかを確認することをお勧めします。

バックエンドサーバグループ

仮想IPアドレスを設定することにより、Server Load Balancerは、同じ地域にある追加されたECSインスタンスを高性能で高可用性のアプリケーションサービスプールに仮想化します。バックエンドサーバーはインスタンスレベルで管理されます。つまり、Server Load Balancerインスタンス内のリスナーは、リスナーに設定されているのと同じポート番号を持つ同じバックエンドECSインスタンスにのみ接続要求を配信できます。

また、ECSインスタンスをサーバーグループの方法で追加することもできます。リスナーごとに異なるサーバー・グループを使用できるため、Server Load Balancerインスタンス内の異なるリスナーは、サーバー・グループ内の異なるポート番号を持つ異なるECSインスタンスに接続要求を分散できます。

注意：リスナーを設定するときにサーバーグループを使用する場合、リスナーは関連するサーバーグループに要求を配信し、バックエンドサーバープール内のECSインスタンスに要求を配信しません。

次の2種類のサーバーグループがサポートされています。

- **マススレーブサーバーグループ**
  従来のマスターバックアップ要件がある場合、つまり、1つのバックエンドサーバーがマスターサーバーとして使用され、もう1つがスレーブサーバーとして使用される場合は、マスター/スレーブサーバーグループを作成できます。マスターサーバーが正常に動作すると、要求はそのマスターサーバーに配信されます。マスターサーバーが停止している場合、サービスの中断を避けるために、要求がスレーブサーバーに配信されます。マスター/スレーブサーバーグループは、レイヤー4リスナーに対してのみ使用できます。詳細は、「マスタスレーブサーバグループの作成」を参照してください。
- **VServerグループ**
  さまざまなバックエンドサーバーに異なる接続要求を配布する必要がある場合、またはドメイン名転送ルールを構成する場合は、VServerグループを使用できます。詳細については、「Create VServer groups」を参照してください。

バックエンドサーバーを追加する

前提条件

Server Load Balancerインスタンスの作成。

ECSインスタンスを作成し、ECSインスタンスにアプリケーションを配備して、分散された要求を処理します。以前にECSを使用していない場合は、ECSクイックスタートを参照してECSインスタンスを作成してください。

手順

[Server Load Balancer コンソール]にログインします。

[インスタンス管理]ページで、対象リージョンを選択します。

対象のServer Load BalancerインスタンスのIDをクリックします。

左側のナビゲーションペインで、 **[バックエンド サーバー]** をクリックします。

[ロードバランサーサーバーのプール] ページで、 **[追加されていないサーバー]** のタブをクリックします。

ターゲットECSインスタンスの横にある**追加**をクリックするか、ECSインスタンスを選択して、**バッチでの追加**をクリックします。

注：ECSインスタンスのネットワークタイプは、Server Load Balancerのインスタンスタイプに準拠している必要があります。詳細は、インスタンスとネットワークの種類を参照してください。

- Internet Server Load Balancerインスタンスの場合、従来のネットワーク内のECSインスタンスまたは同じVPC内のECSインスタンスを追加できます。
- VPCネットワーク内のイントラネットサーバーのLoad Balancerインスタンスの場合、Server Load Balancerインスタンスと同じVPC内にのみECSインスタンスを追加できます。
- クラシック・ネットワークのイントラネット・サーバー・ロード・バランサ・インスタンスの場合、クラシック・ネットワークにのみECSインスタンスを追加できます。

表示された**バックエンドサーバーの追加**ダイアログで、追加されたECSインスタンスの重みを指定してから、**確認**をクリックします。

より高い重みのECSインスタンスは、より多くの接続要求を受信します。ECSインスタンスのサービス機能に基づいて重みを設定できます。

> **注**：ウェイトが 0 に設定されている場合、ECSインスタンスにリクエストは送信されません。

追加されたECSインスタンスは、 **[追加されたサーバー]** のタブに表示されます。追加されたECSインスタンスの重量を削除または変更できます。

VServerグループを作成する

VServerグループを使用すると、リスナーディメンション内のバックエンドサーバを管理およびカスタマイズできます。つまり、Server Load Balancerインスタンスの下のリスナーは、異なるバックエンドサーバとドメイン名ベースの転送に異なる要求を分散する要件を満たす、異なるバックエンドサーバを使用できます。

リスナーを設定するときにVServerグループを使用する場合、リスナーは関連するVServerグループにリクエストを配信し、バックエンドサーバプール内のECSインスタンスにリクエストを配信しません。

Server Load Balancerインスタンスの場合、サーバプールにバックエンドサーバを追加し、VServerグループを構成し、同時にドメイン名転送ルールを構成すると、要求は次の順序で配布されます。

クライアント要求が設定済みドメイン名転送ルールと一致する場合、要求はそのルールに関連付けられたVServerグループに配信されます。

そうでない場合、要求はリスナーに関連付けられたVServerグループに配信されます。

リスナー用のVServerグループを設定していない場合、リクエストはバックエンドサーバープール内のECSインスタンスに配布されます。

VServerグループを使用する場合は、次の点に注意してください。

リスナーと同じリージョン内のバックエンドサーバーだけをVServerグループに追加できます。

1つのECSインスタンスを複数のVServerグループに追加できます。

1つのVServerグループを複数のリスナーに追加できます。

前提条件

Server Load Balancerインスタンスの作成。

ECSインスタンスを作成し、ECSインスタンスにアプリケーションを配備して、分散された要求を処理します。以前にECSを使用していない場合は、**ECSクイックスタート**を参照してECSインスタンスを作成してください。

手順

[Server Load Balancerコンソール] にログオンします。

[インスタンス管理]ページで、ターゲットサーバロードバランサのインスタンスリージョンを選択します。

対象のServer Load BalancerインスタンスのIDをクリックします。

左側のナビゲーションペインで、 **[サーバ]** > **[VServerグループ]** をクリックします。

[マスタースレイブサーバグループ] ページで、 **[VServerグループの作成]**をクリックします。

表示されたダイアログで、次の操作を完了します。

i. 「**グループ名**」フィールドにグループ名を入力します。

ECSインスタンスのネットワークタイプを選択します。

> **注**：ECSインスタンスのネットワークタイプは、Server Load Balancerのインスタンスタイプに準拠している必要があります。詳細は、**インスタンスとネットワークの種類**を参照してください。

i. インターネットServer Load Balancerインスタンスの場合、クラシックネットワーク内のECSインスタンスまたは同じVPC内のECSインスタンスを追加できます。
ii. VPCネットワーク内のイントラネットServer Load Balancerインスタンスの場合、Server Load Balancerインスタンスと同じVPC内にのみECSインスタンスを追加できます。
iii. クラシック・ネットワークのイントラネットServer Load Balancerインスタンスの場合、クラシック・ネットワークにのみECSインスタンスを追加できます。

[Available Server List] パネルからECSインスタンスを選択します。

各ECSインスタンスのポート番号と重みを設定し、**確認**をクリックします。

作成されたVServerグループが **VServerグループ** ページに表示されます。VServerグループのECSインスタンスを削除または追加することができます（**[編集]**をクリックします）。また、このVServerグループをインスタンスのリスナーまたは転送ルールに関連付けることもできます。

マスター/スレーブサーバーグループを作成する

前提条件

Server Load Balancerインスタンスの作成

ECSインスタンスを作成し、ECSインスタンスにアプリケーションを配備して、分散された要求を処理します。以前にECSを使用していない場合は、ECSクイックスタートを参照してECSインスタンスを作成してください。

手順

[Server Load Balancerコンソール] にログオンします。

[**インスタンス管理**]ページで、対象Server Load Balancerのインスタンスのリージョンを選択します。

対象のServer Load BalancerインスタンスのIDをクリックします。

左側のナビゲーションペインで、 **[バックエンドサーバー]** > **[マスタースレーブサーバグループ]** をクリックします。

マスタースレーブサーバグループ ページで、 **[マスタースレーブサーバグループを作成する]**をクリックします。

表示されたダイアログで、次の操作を完了します。

i. **[グループ名]**フィールドにグループ名を入力します。
ii. ECSインスタンスのネットワークタイプを選択します。

注意：ECSインスタンスのネットワークタイプは、Server Load Balancerのインスタンスタイプに準拠している必要があります。詳細は、インスタンスとネットワークの種類を参照してください。

i. Internet Server Load Balancerインスタンスの場合、従来のネットワーク内のECSインスタンスまたは同じVPC内のECSインスタンスを追加できます。
ii. VPCネットワーク内のIntranet Server Load Balancerインスタンスの場合、Server Load Balancerインスタンスと同じVPC内にのみECSインスタンスを追加できます。
iii. クラシック・ネットワークのIntranet Server Load Balancerインスタンスの場合、クラシック・ネットワークにのみECSインスタンスを追加できます。

iii. **[Available Server List]** パネルから2つのECSインスタンスを選択します。
iv. 各ECSインスタンスのポート番号を設定し、1つのECSインスタンスをスレーブサーバ

ーとして選択します。
v. **[確認]**をクリックします。

証明書管理

証明書形式要件

Server Load Balancer は PEM 以外の証明書形式をサポートしません。証明書が PEM 形式でない場合は、以下の「Server Load Balancer でサポートされる証明書形式と変換方法」のセクションを参照してください。

ルートCA発行の証明書

ルート CA から証明書を取得した場合、ブラウザーなどのアクセスデバイスで Web サイトが信用されるために必要なのはこの証明書だけです。

証明書発行時に関連の説明を提供します。ルールの説明に注意してください。

-----BEGIN CERTIFICATE-----, -----END CERTIFICATE----- 証明書の先頭と末尾にある内容。これらは一緒にアップロードする必要があります。

各行 64 文字で、最後の行は 64 文字を超えることはできません。最終行は64文字以下でも問題ありません。

空白文字を含むことはできません。

以下が、ルート CA が発行した証明書のサンプルです。

中間CA発行の証明書

中間CAの証明書を使用する場合、複数の中間証明書を入手することになります。

中間証明書の注意点を確認してください：

> 証明書の最初に 1 番目の証明を記載し、次にスペースを入れずに 2 番目の証明を記載します。
>
> 各行 64 文字で、最後の行は 64 文字を超えることはできません。最終行は64文字以下でも問題ありません。
>
> 空白文字を含むことはできません。
>
> 各証明書は証明書ルールに従う必要があります。

以下が、証明書チェーンのサンプルです。

```
- - - - - BEGIN CERTIFICATE - - - - -

- - - - - END CERTIFICATE - - - - -
- - - - - BEGIN CERTIFICATE - - - - -

- - - - - END CERTIFICATE - - - - -
- - - - - BEGIN CERTIFICATE - - - - -
```

```
- - - - - END CERTIFICATE - - - - -
```

RSA秘密鍵

証明書と一緒に秘密鍵をアップローとする場合は、以下のルールを確認する必要があります。

> ——-BEGIN RSA PRIVATE KEY——-, ——-END RSA PRIVATE KEY——-RSA 秘密鍵の先頭と末尾にある形式で、これらは一緒にアップロードする必要があります。
>
> 空白文字を含むことはできません。各行 64 文字で、最後の行は 64 文字を超えることはできません。最終行は64文字以下でも問題ありません。

注意: 秘密鍵が暗号化されている場合は、ヘッダーとフッターが- - - - - BEGIN PRIVATE KEY - - - - -,- - - - - END PRIVATE KEY - - - - - または - - - - - BEGIN ENCRYPTED PRIVATE KEY - - - - -, - - - - - END ENCRYPTED PRIVATE KEY - - - - -, もしくはProc-Type: 4で暗号化されています。暗号化する場合は、アップロード前に以下のコマンドを実行して暗号化します：

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

RSA 秘密鍵ファイルフォーマット

証明書の生成

概要

HTTPSリスナーを設定するとき、自己署名の CA 証明書を使用できます。この文書の手順に従って、CA 証明書を生成し、作成されたCA 証明書を使用してクライアント証明書を作成します。

OpenSSL を使用してCA 証明書を生成する

'/root' ディレクトリに ca フォルダーを作成して、ca フォルダーの下にサブフォルダーを 4 つ作成します。

```
 $ sudo mkdir ca
$ cd ca
$ sudo mkdir newcerts private conf server
```

- newcerts フォルダーは、CA によって署名されたデジタル証明書の保存に使用します (証明書のバックアップディレクトリ)。
- private フォルダーは、CA 秘密鍵の保存に使用します。
- conf フォルダーは、パラメーターを単純化するための設定ファイルの保存に使用します。
- server フォルダーは、サーバー証明書ファイルの保存に使用します。

conf ディレクトリの下に openssl.conf ファイルを作成し、その内容を以下のように編集します。

```
 [ ca ]
default_ca = foo
[ foo ]
dir = /root/ca
database = /root/ca/index.txt
new_certs_dir = /root/ca/newcerts
certificate = /root/ca/private/ca.crt
serial = /root/ca/serial
private_key = /root/ca/private/ca.key
RANDFILE = /root/ca/private/.rand
default_days = 365
default_crl_days= 30
default_md = md5
unique_subject = no
policy = policy_any
```

```
[ policy_any ]
countryName = match
stateOrProvinceName = match
organizationName = match
organizationalUnitName = match
localityName = optional
commonName = supplied
emailAddress = optional
```

次のコマンドを実行して、秘密鍵ファイルを生成します。

```
 $ cd /root/ca
$ sudo openssl genrsa -out private/ca.key
```

次の図は、秘密鍵生成の例です。

```
root@iZbp1hfvivcqx1jbwap31iZ:~/ca/conf# cd /root/ca
root@iZbp1hfvivcqx1jbwap31iZ:~/ca# sudo openssl genrsa -out private/ca.key
Generating RSA private key, 2048 bit long modulus
.....+++
..+++
e is 65537 (0x10001)
```

次のコマンドを実行し、コマンドの後に例に従って必要な情報を指定して、Enter キーを押して証明書を要求する csr ファイルを生成します。

```
$ sudo openssl req -new -key private/ca.key -out private/ca.csr
```

入力例:

```
root@iZbp1hfvivcqx1jbwap31iZ:~/ca# sudo openssl req -new -key private/ca.key -ou
t private/ca.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:ZheJiang
Locality Name (eg, city) []:HangZhou
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Alibaba
Organizational Unit Name (eg, section) []:Test
Common Name (e.g. server FQDN or YOUR name) []:mydomain
Email Address []:a@alibaba.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@iZbp1hfvivcqx1jbwap31iZ:~/ca#
```

注意: 「Common Name」にSLBインスタンスのドメイン名を入力します。

次のコマンドを実行して、crt ファイルを生成します。

```
$ sudo openssl x509 -req -days 365 -in private/ca.csr -signkey private/ca.key -out private/ca.crt
```

次のコマンドを実行して、キーの開始シリアル番号を設定します。シリアル番号には、任意の 4 文字を使用できます。

```
$ sudo echo FACE > serial
```

次のコマンドを実行して、CA キーライブラリを作成します。

```
$ sudo touch index.txt
```

次のコマンドを実行して、ユーザー証明書を削除するための証明書失効リストを作成します。

```
$ sudo openssl ca -gencrl -out /root/ca/private/ca.crl -crldays 7 -config
"/root/ca/conf/openssl.conf"
```

出力例:

```
Using configuration from /root/ca/conf/openssl.conf
```

クライアント証明書を生成します。

次のコマンドを実行して、キーを保存するユーザーディレクトリを作成します。

```
$ sudo mkdir users
```

次のコマンドを実行して、ユーザーのキーを作成します。

```
$ sudo openssl genrsa -des3 -out /root/ca/users/client.key 1024
```

注意：入力されたパスフレーズはキーのフレーズです。

次のコマンドを実行して、キーの証明書署名を要求する csr ファイルを作成します。

```
$ sudo openssl req -new -key /root/ca/users/client.key -out /root/ca/users/client.csr
```

確認のプロンプトが表示されたら、前の手順で設定したパスフレーズを入力します。users ディレクトリの下に client.csr ファイルが生成されます。

次のコマンドを実行し、CA 秘密鍵を使用してキーに署名します。

```
$ sudo openssl ca -in /root/ca/users/client.csr -cert /root/ca/private/ca.crt -keyfile
/root/ca/private/ca.key -out /root/ca/users/client.crt -config "/root/ca/conf/openssl.conf"
```

確認のプロンプトが表示されたら、両方とも「y」と入力します。users ディレクトリの下に client.crt ファイルが生成されます。

次のコマンドを実行して、証明書をほとんどのブラウザーで認識できる PKCS12 形式に変換します。

```
$ sudo openssl pkcs12 -export -clcerts -in /root/ca/users/client.crt -inkey /root/ca/users/client.key
-out /root/ca/users/client.p12`
```

- client.key のパスフレーズを入力します。
- 次のプロンプトが表示されたら、Export Password を入力します。このパスワードは、クライアントで証明書をインストールするときに必要です。
- users ディレクトリの下に client.p12 ファイルが生成されます。

以下のコマンドを実行して作成された証明書を表示します。

```
cd users
ls
```

証明書の変換

Server Load Balancer では PEM 証明書のみ利用できます。他の形式の証明書は PEM 形式に変換してから、Server Load Balancer にアップロードする必要があります。形式の変換には OpenSSL が推奨されます。広く使用されている証明書形式を PEM 形式に変換する方法を以下で説明します。

DER から PEM

DER 形式は通常 Java プラットフォームで使用されます。

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 から PEM

P7B 形式は通常 Windows Server と Tomcat で使用されます。

証明書の変換:

openssl pkcs7 -print_certs -in incertificate.p7b -out outcertificate.cer

PFX から PEM

PFX 形式は通常 Windows Server で使用されます。

秘密鍵の変換:

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

証明書の変換:

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

証明書のアップロード

Server Load Balancer には HTTPS 対応の証明書管理機能が備えられており、 証明書をバックエンドサーバーにデプロイせずに SLB 証明書管理システムに保存できます。Server Load Balancer の証明書は、リージョンごとに管理します。1 つの証明書を複数のリージョンで使用するには、証明書をアップロードするときに複数のリージョンを選択してください。リージョンごとに管理する仕様は、セキュリティとパフォーマンスを確保するために最優先されます。

以下の点に注意してください：

- 各ユーザーは、最大 100 個まで証明書を作成できます。
- 証明書は 1 つ以上のリスナーに適用できます。
- セキュリティとパフォーマンスを考慮して、複数のリージョンで証明書を使用する場合は、これらのすべてのリージョンでアップロードする必要があります。

操作手順

証明書の作成

Server Load Balancer コンソールにログインします。

左側のナビゲーションツリーから [証明書] を選択し、[証明書] ページを表示します。

右上隅にある [**証明書の作成**] をクリックします。

以下に従い、証明書の設定を行います。

Configuration	Description
証明書名	アップロードする証明書の名前を入力します。 名前は、文字、数字、ハイフン（ - ）、スラッシュ（/）、ピリオド（。）、およびアンダースコア（_）を含めて、1～80文字でなければなりません。
リージョン	証明書がアップロードされる地域を1つ以上選択します。 証明書とServer Load Balancerインスタンスでは、 リージョンは同じである必要があります。
証明書タイプ	証明書の種類を選択します。 - サーバー証明書：サーバー証明書をアップロードします。クライアントは、クライアントがそれを使用して、サーバーによって送信された証明書が認証局によって発行されているかどうかをチェックします。**注**：サーバ証明書の場合は、証明書の秘密鍵もアップロードする必要があります。 - CA証明書：CA証明書をアップロードします。サーバーは、セキュリティー保護された接続を開始する前に、CA証明書を使用して、クライアント証明書のCA署名を認証の一部として認証します。

証明書の内容	証明書の内容をエディタに貼り付けます。 PEM形式の証明書のみがサポートされています。有効な証明書形式を表示するには、[**サンプルをインポート**]をクリックします。詳細は、**証明書形式要件**を参照してください。
秘密鍵	エディタにサーバ証明書の秘密鍵を貼り付けます。 有効な書式を表示するには、[**サンプルをインポート**]をクリックします。キーが暗号化されている場合は、まずキーを変換する必要があります。 詳細は、**証明書形式要件**を参照してください。

証明書の置換え

適用シナリオ

次のシナリオでは証明書の置換えが必要です。

既存の証明書の有効期限が切れた場合は、新しい証明書を作成する必要があります。

Server Load Balancer の証明書を追加するときにエラーが発生する場合、秘密鍵が正しくない可能性があります。この場合、受け入れられた証明書に置き換える必要があります。

操作手順

証明書を作成してアップロードします。
詳細については、「**証明書の作成**」および「**証明書のアップロード**」を参照してください。

HTTPS リスナーを更新します。
詳細については、「**HTTP リスナーの設定**」を参照してください。

古い証明書を削除します。
[証明書]管理ページで、削除する証明書を見つけて **[削除]** をクリックし、証明書を削除します。

[確認]をクリックします。

アクセス制御リスト

アクセス制御リストの設定

Server Load Balancer には、アクセス制御機能があります。各リスナーにそれぞれのアクセス制御ルール（アクセス ホワイトリストあるいはブラックリスト）を設定することができます。 リスナーに対するアクセス制御を設定する前に、アクセス制御リストを作成しておかなければなりません。

現在、アクセス制御機能は次のリージョンでご利用いただけます。

- シンガポール
- オーストラリア（シドニー）
- マレーシア（クアラルンプール）
- 日本（東京）
- 米国（シリコンバレー）
- 米国（バージニア）
- ドイツ（フランクフルト）
- UAE（ドバイ）
- インド（ムンバイ）

制限

複数のアクセス制御リストを作成することができます。各リストには、複数の IP アドレス、あるいは、CIDR ブロックを含みます。アクセス制御リストの制限は次のとおりです。

リソース	制限
1 リージョンに対するアクセス制御リストの最大数	50
1 回に追加可能なIP アドレスの最大数	50
1 アクセス制御リストに対するエントリの最大数	300
1 リスナーに対するアクセス制御リストの最大数	50

アクセス制御リストを作成する

[Server Load Balancer コンソール]にログインします。

リージョンを選択します。

左ナビゲーション メニューにある、**アクセス制御**をクリックします。

アクセス制御リスト作成をクリックし、リスト名を入力し、**確認**をクリックします。

IP エントリーを追加する

[Server Load Balancer コンソール]にログインします。

リージョンを選択します。

左ナビゲーション メニューにある、**アクセス制御** をクリックします。

対象のアクセス制御リストを探し出し、**管理**をクリックします。

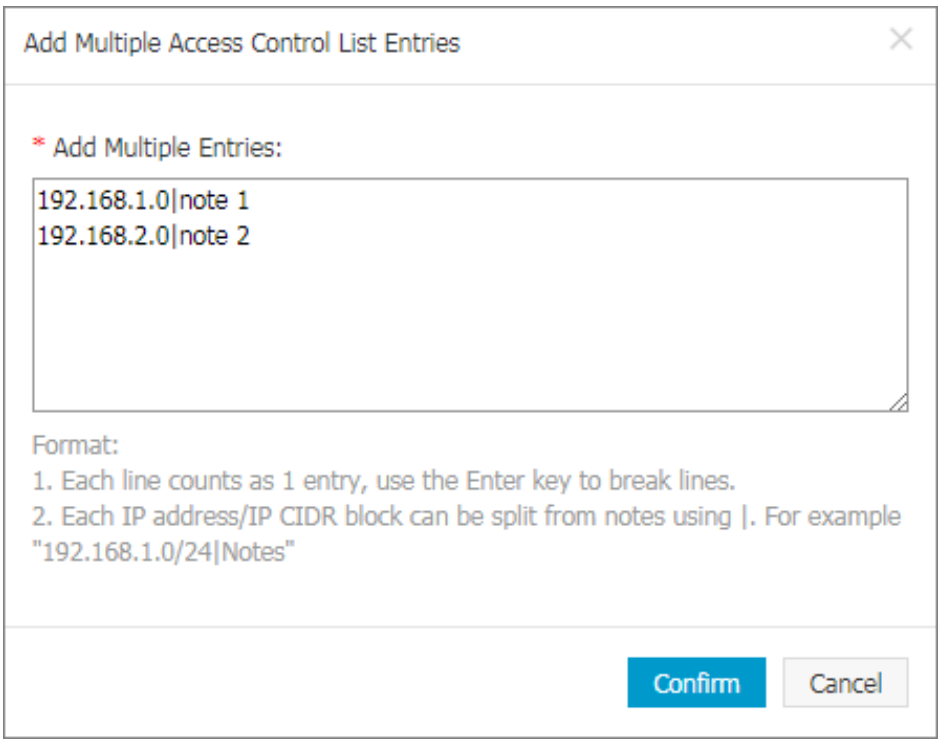
IP エントリを追加します。

エントリを複数追加をクリックし、表示されるダイアログボックスに IP アドレス、または、CIDR ブロックを 1 つ以上追加します。

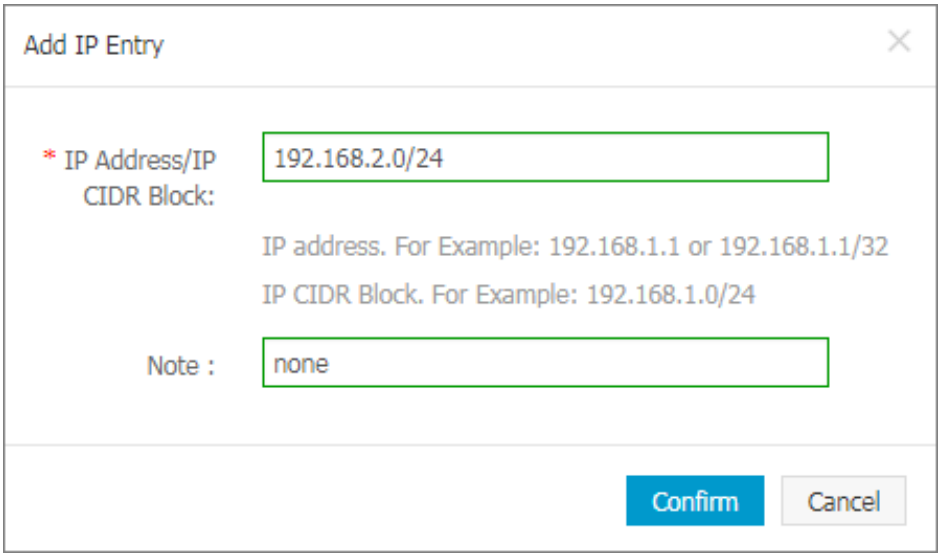
エントリーを追加する際には、次の点にご注意ください。

1 行 1 エントリとなります。Enter キーで改行します。

“|” を使用して IP アドレス や CIDR ブロックと注釈を分割します。たとえば、”192.168.1.0/24|注釈”。



エントリ追加をクリックし、IP アドレスや CIDR ブロックを追加します。 **確認**をクリックして、アクセス制御リストにエントリを追加します。



IP エントリを削除する

[Server Load Balancer コンソール]にログインします。

リージョンを選択します。

左ナビゲーション メニューにある**アクセス制御**をクリックします。

対象のアクセス制御リストを探し出し、**管理**をクリックします。

対象 IP エントリの**アクション**列にある**削除**をクリックします。あるいは、複数の IP エントリを選択してエントリ リストの下にある**削除**をクリックします。

表示されるダイアログボックスで、**確認**をクリックします。

アクセス制御を設定

Server Load Balancer （SLB）は、アクセス制御機能を提供します。リスナを作成するときに、異なるリスナに対して異なるホワイトリストまたはブラックリストを設定できます。詳細は、レイヤー 7 リスナーの構成とレイヤー 4 リスナーを構成するを参照してください。

また、リスナーの作成後にアクセス制御を変更または構成することもできます。このドキュメントでは、リスナーの作成後にアクセス制御を構成する方法について説明します。

アクセス制御を有効にする

前提条件

アクセス制御リストを作成しました。詳細は、アクセス制御リストの設定を参照してください。

リスナーを作成しました。

手順

[Server Load Balancer コンソール]にログオンします。

ターゲット SLB インスタンスのリージョンを選択します。

アクセス制御の設定が必要な SLB インスタンスの ID をクリックします。

左側のナビゲーションペインで、**リスナー**をクリックします。

インスタンスリスナーページで、ターゲットリスナーの**オプション**列で**アクセス制御の設定**をクリックします。

表示されたダイアログボックスで、アクセス制御を有効にし、アクセス制御方法とアクセス制御リストを選択します。 **確認**をクリックします。

アクセスリストは、ホワイトリストまたはブラックリストとして使用できます。

ホワイトリスト: 選択したアクセス制御リストの IP アドレスまたは CIDR ブロックからの要求のみが転送されます。アプリケーションが特定の IP アドレスからのアクセスのみを許可するシナリオに適用されます。

ホワイトリストを有効にすると、ビジネス上のリスクが発生します。対応するアクセス制御リストに IP エントリを追加せずにホワイトリストを有効にすると、リクエストは転送されません。

ブラックリスト: 選択したアクセス制御リストの IP アドレスまたは CIDR ブロックからのリクエストは転送されません。アプリケーションが特定の IP アドレスからのアクセスのみを拒否するシナリオに適用されます。

対応するアクセス制御リストに IP エントリを追加せずにブラックリストを有効にすると、すべてのリクエストが転送されます。

アクセス制御を無効にする

[Server Load Balancer コンソール]にログオンします。

ターゲット SLB インスタンスのリージョンを選択します。

アクセス制御の設定が必要な SLB インスタンスの ID をクリックします。

左側のナビゲーションペインで、**リスナー**をクリックします。

インスタンスリスナーページで、ターゲットリスナーの**オプション**列で**アクセス制御の設定**をクリックします。

表示されたダイアログボックスで、アクセス制御を無効にします。

ホワイトリストの設定

注意： Server Load Balancer （SLB）は、すべてのリージョンでブラックリスト機能をリリースしました。詳細は、アクセス制御リストの設定を参照してください。

SLB には、アクセス制御機能があります。各リスナーにそれぞれのホワイトリストまたはブラックリストを設定することができます。詳細は、レイヤー7リスナーの構成 および レイヤー 4 リスナーを構成するを参照のこと。

リスナーを作成後もアクセス制御の変更や設定ができます。本ドキュメントでは、リスナー作成後にアクセス制御を設定する方法をご紹介します。

アクセス制御の有効化

前提条件

アクセス制御リストを既に作成している。詳細は、アクセス制御リストの設定を参照参照してください。

リスナーを既に作成しています。

手順

[SLB コンソール]にログインします。

対象となる SLB インスタンスのリージョンを選択します。

アクセス制御を設定する必要のある SLB インスタンスの ID をクリックします。

左のナビゲーション メニューより、**リスナー**をクリックします。

リスナー画面で、対象リスナーの**アクション**列にある**詳細 > アクセス制御の設定**をクリックします。

表示されるダイアログボックスで、アクセス制御を有効にし、アクセス制御の方法およびアクセス制御リストを選択します。**確認**をクリックします。

アクセス リストは、ホワイトリストとして、あるいは、ブラックリストとして使用できます。

ホワイトリスト：選択したアクセス制御リスト内の IP アドレス、または、CIDR ブロックからのリクエストのみ転送されます。アプリケーションへのアクセスを特定の IP アドレスに限定する場合に適用します。

ホワイトリストを有効にすることにより、ビジネス リスクは高まります。該当するアクセス制御リストに IP アドレスを追加せずにホワイトリストを有効にした場合、すべてのリクエストが転送されません。

ブラックリスト：選択したアクセス制御リスト内の IP アドレス、または、CIDR ブロックからのリクエストは転送されません。特定の IP アドレスからのみアプリケーションへのアクセスを拒否する場合に適用します。

該当するアクセス制御リストに IP アドレスを全く追加せずにブラックリストを有効にした場合、すべてのリクエストが転送されます。

アクセス制御を無効にする

[SLB コンソール]にログインします。

対象となる SLB インスタントのリージョンを選択します。

アクセス制御の設定が必要な SLB インスタンスの ID をクリックします。

左のナビゲーション メニューより、**リスナー**をクリックします。

リスナー画面で、対象リスナーの**アクション**列にある**詳細 > アクセス制御の設定**をクリックします。

表示されるダイアログボックスで、アクセス制御を無効にします。

旧バージョンのアクセス制御リストを移行

リスナー用のホワイトリストをすでに設定している場合、システムはホワイトリストの IP アドレスまたは CIDR ブロックをアクセス制御リストに自動的に追加し、そのリストをリスナーに適用する機能を提供します。

古いバージョンのホワイトリストを新しいバージョンのアクセスコントロールに移行するには、次の手順を実行します。

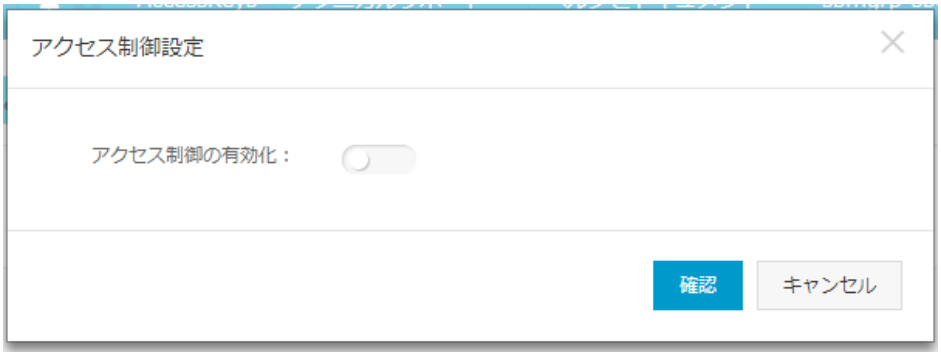
[Server Load Balancer コンソール]にログオンします。

ターゲット SLB インスタンスのリージョンを選択し、ターゲットインスタンスの ID をクリックします。

左側のナビゲーションペインで、**インスタンスリスナー**をクリックします。

ターゲットリスナーを見つけて、[**オプション]> [アクセス制御の設定]** をクリックします。

*アクセス制御の有効化をクリックします。



[**アクセス制御リストの作成]をクリックします。

適用をクリックして、リストをホワイトリストとしてリスナーに適用します。

注意: リストを適用しない場合、ホワイトリストは有効になりません。

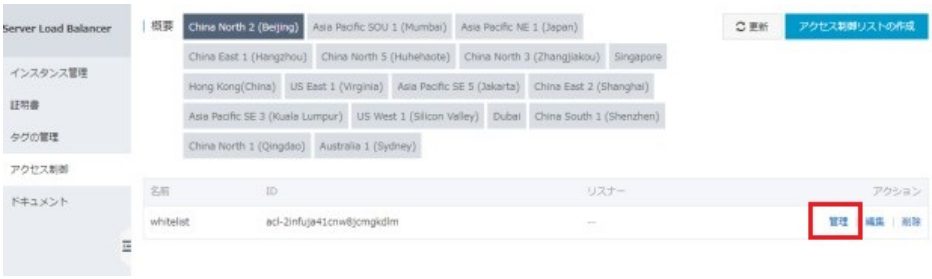
古いバージョンから移行されたアクセス制御リストを表示する

[Server Load Balancer コンソール]にログオンします。

ターゲットリージョンを選択します。

左側のナビゲーションペインで、[**アクセス制御**]をクリックします。

作成されたアクセス制御リストを検索し、バインドされたリスナーを表示します。また、**管理**をクリックして IP エントリを管理することもできます。



モニタリング

モニタリングデータの表示

Server Load Balancer では、さまざまな “Server Load Balancer ポート” のリアルタイムメトリックと履歴メトリックを表示することができます。

注意：バックエンド ECS がない、またはバックエンド ECS のヘルスチェックステータスが [異常] になっている Server Load Balancer インスタンスはメトリックデータを提供できません。したがって、Server Load Balancer インスタンスが正しく設定されていて、適切に動作していることを確認する必要があります。

手順

[Server Load Balancer コンソール]にログインします。

リージョンとインスタンス ID を選択します。

ページの一番上にあるリストから、表示する Server Load Balancer ポートのメトリックデータを選択します。

モニタリングチャートでメトリック項目をクリックします。

メトリクスをモニタする

モニタリング項目	説明
トラフィック	- インバウンドトラフィック：外部アクセスによって消費されたトラフィック。 - アウトバウンドトラフィック：サーバーロードバランサによって消費されるトラフィック。
パケット	- インバウンドパケット：1 秒間に受信されたリクエストパケット数。 - アウトバウンドパケット：1 秒間に送信された応答パケット数。
同時接続数	- アクティブ接続：確立された TCP 接続数。 - 非アクティブ接続：接続が確立されていない状態の TCP 接続数。 - 同時接続：TCP 接続の総数。
新規接続	統計期間にクライアントから Server Load Balancer に確立された新しい TCP 接続の平均

	数。
トラフィックの低下	- 破棄されたインバウンドトラフィック：1 秒あたりに破棄されたインバウンドトラフィックの量。 - 破棄されたアウトバウンドトラフィック：1 秒あたりに破棄されたアウトバウンドトラフィックの量。
破棄されたパケット	- 破棄されたインバウンドパケット：1 秒あたりに破棄されたインバウンドパケット数。 - 破棄されたアウトバウンドパケット：1 秒あたりに破棄されたアウトバウンドパケット数。
破棄された接続数	1 秒あたりに破棄された TCP 接続数。
レイヤー7 リスナーの QPS	1 秒あたりに処理できる HTTP/HTTPS リクエスト数。
レイヤー 7 リスナーの応答時間	Server Load Balancer の平均応答時間。
レイヤー7 リスナーのステータスコード（2XX）/（3xx）/（4xx）（5xx）（その他）	リスナーによって作成された HTTP 応答コードの平均数。
レイヤー7 リスナーの UpstreamCode 4XX/5XX	バックエンドサーバーによって作成された HTTP 応答コードの平均数。
レイヤー7 リスナーの UpstreamRT	バックエンドサーバーの平均応答時間。

アラーム設定

CloudMonitor サービスを有効にした後、[CloudMonitor コンソール]を使用してアラーム設定を構成できます。詳細については、Server Load Balancer の監視を参照してください。

注意：Server Load Balancer のリスナーまたはインスタンスが削除されると、アラーム設定もそれに対応して削除されます。

手順

[Server Load Balancer コンソール]にログオンする。

選択リージョンとターゲット・インスタンスの ID のリンクをクリックします。

インスタンスがリスナーとヘルスチェックで構成されていることを確認します。

左側のナビゲーションメニューで **[インスタンスのモニタリング]** をクリックします。

[しきい値警告設定]をクリックします。その後、CloudMonitor コンソールに移動します。

[アラームルールを作成]をクリックしてアラームを作成します。

Server Load Balancer マルチゾーン機能

Server Load Balancer のマルチゾーン展開とは何ですか？

クラウド製品ゾーンとは、一連の独立したインフラストラクチャを指します。異なるゾーンには独立したインフラストラクチャ（ネットワーク、電源、空調など）があり、あるゾーンのインフラストラクチャの障害は他のゾーンに影響しません。

より信頼性の高いサービスを提供するために、Alibaba Cloud Server Load Balancer （SLB）の複数のゾーンがすでに異なるリージョンに配置されており、データセンター間の災害復旧を実現しています。

- アクティブゾーンのデータセンターに障害があり、利用できない場合、 Server Load Balancerはスタンバイゾーンのデータセンターにすばやく切り替えて、30 秒以内にサービス機能を復元します。
- アクティブゾーンが復元されると、 Server Load Balancer 自動的にアクティブゾーンのデータセンターに戻り、サービスを提供します。

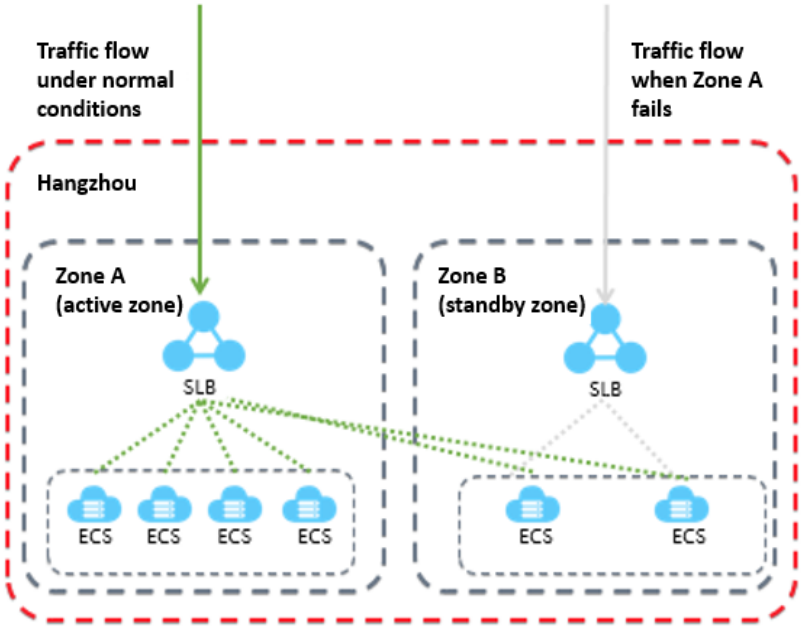
マルチゾーン展開をサポートするリージョンに Server Load Balancer インスタンスを作成して、サービスの可用性を向上させることができます。詳細については、Server Load Balancer 高可用性のベストプラクティスを参照してください。

注意：

Server Load Balancer は、ECS インスタンスと SLB インスタンスが同じリージョン内にある限り、異なるゾーンにある ECS インスタンスの接続をサポートします。SLB は、異なるゾーンの ECS インスタンスにトラフィックを分散できます。

通常の状態では、スタンバイゾーンの SLB インスタンスはスタンバイモードです。SLB インスタンスのモードを手動で変更することはできません。SLB インスタンスは、Alibaba Cloud がインスタンスの障害ではなく停電または光ケーブルの障害によって現在のゾーンが使用できないことが検出された場合にのみスタンバイゾーンに切り替わります。

SLB インスタンスと ECS インスタンスは異なるクラスタです。ゾーン A の SLB インスタンスが利用できない場合、ゾーン A の ECS インスタンスは必ずしも利用できないとは限りません。したがって、アクティブゾーンに障害が発生したために SLB インスタンスがスタンバイゾーンに切り替わると、SLB インスタンスはトラフィックをさまざまなゾーンの ECS インスタンスに分散できます。ただし、停電や光ケーブルの障害が 2 つのゾーンのすべてのクラスタに発生すると、関連するすべてのサービス（SLB と ECS を含むがこれに限定されない）は利用できません。



アクティブゾーンとスタンバイゾーンのリスト

次の表では、異なるリージョンのアクティブゾーンとスタンバイゾーンを表しています。DescribeZones API を呼び出して、使用可能なアクティブなゾーンとスタンバイゾーンを照会することもできます。

リージョン	ゾーンタイプ	ゾーン	
		プライマリゾーン	バックアップゾーン
中国東部 1 (杭州)	マルチゾーン	Zone B	Zone D
		Zone D	Zone E
		Zone E	Zone F
		Zone F	Zone E
中国東部 2 (上海)	マルチゾーン	プライマリゾーン	バックアップゾーン

		Zone A	Zone B
		Zone B	Zone A または Zone D
		Zone C	Zone B
		Zone D	Zone B
中国南部 1 (深圳)	マルチゾーン	**プライマリゾーン**	**バックアップゾーン**
		Zone A	Zone B
		Zone B	Zone A
		Zone C	Zone B
中国北部 1 (青島)	マルチゾーン	**プライマリゾーン**	**バックアップゾーン**
		Zone B	Zone C
		Zone C	Zone B
中国北部 2 (北京)	マルチゾーン	**プライマリゾーン**	**バックアップゾーン**
		Zone A	Zone B または Zone D
		Zone B	Zone A または Zone C
		Zone C	Zone B
		Zone D	Zone A
中国北部 3 (張家口)	シングルゾーン	Zone A	
中国北部 5 (フフホト)	シングルゾーン	Zone A	
UAE (ドバイ)	シングルゾーン	Zone A	
アジア太平洋 (シンガポール)	マルチゾーン	**プライマリゾーン**	**バックアップゾーン**
		Zone A	Zone B
		Zone B	Zone A
アジア太平洋 (シドニー)	シングルゾーン	Zone A	
アジア太平洋 (クアラルンプール)	シングルゾーン	Zone A	
アジア太平洋 (日本)	シングルゾーン	Zone A	
香港	マルチゾーン	**プライマリゾーン**	**バックアップゾーン**
		Zone B	Zone C
		Zone C	Zone B
米国東部 1 (バージニア)	シングルゾーン	Zone A	
米国西部 1 (シリコンバレー)	マルチゾーン	**プライマリゾーン**	**バックアップゾーン**
		Zone A	Zone B
		Zone B	Zone A

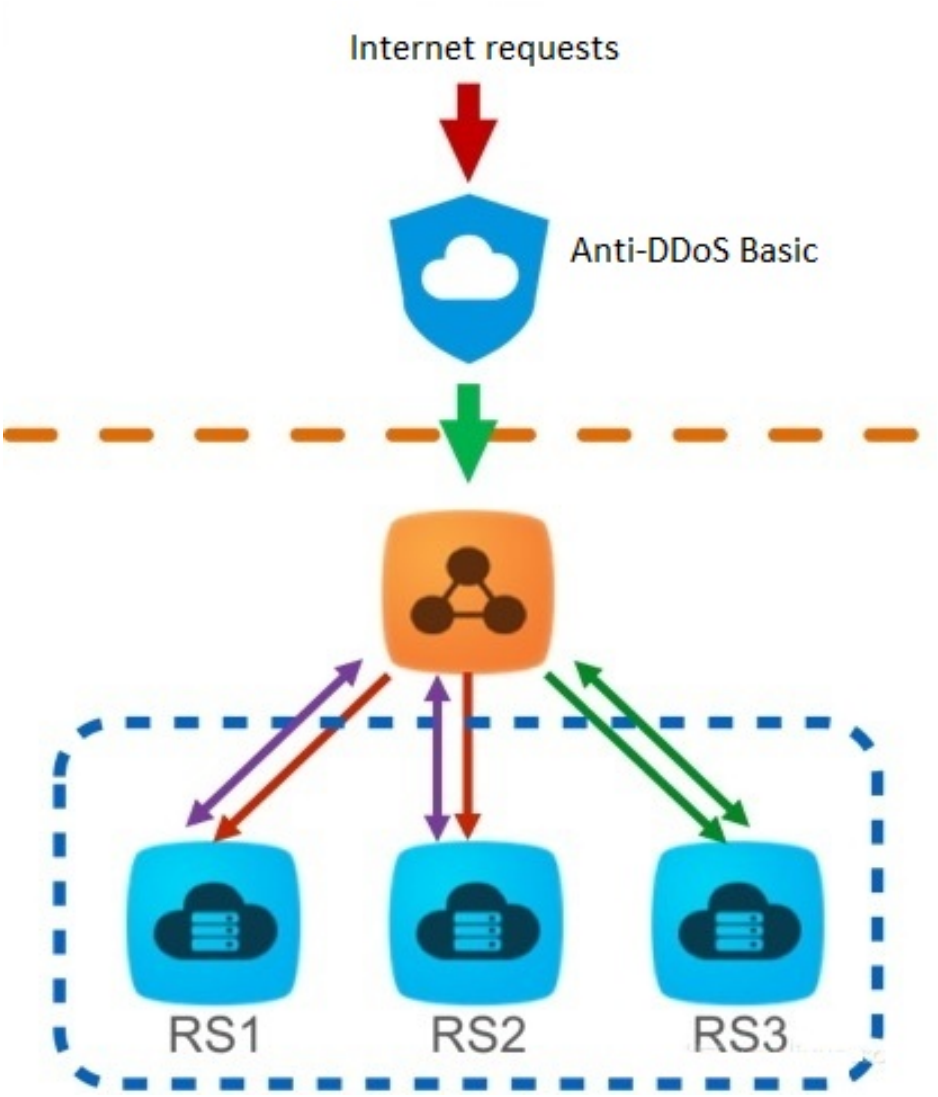
DDoS防御

SLB コンソールで、Internet Server Load Balancer インスタンスの Alibaba Cloud Security の閾値を表示できます。

> **注意:** この機能は、現在青島 (中国北部 1) 、 北京 (中国北部 2) 、 杭州 (中国東部 1) 、 上海 (中国東部 2) 、 深セン (中国南部 1) 、 香港 、シンガポール、 バージニア (米国東部 1)およびシリコンバレー (米国西部 1)のリージョンで利用可能です。

Anti-DDoS Basic

Alibaba Cloud は、Server Load Balancer に Anti-DDoS Basic（最大5 GB）を提供します。次の図に示すように、インターネットからのすべてのトラフィックは、まず Server Load Balancer に到着する前に Alibaba Cloud Security を通過する必要があります。Anti-DDoS Basic は一般的な DDoS 攻撃を駆除してフィルタリングし、SYN フラッド、UDP フラッド、ACK フラッド、ICMP フラッド、DNS クエリーフラッドなどの攻撃からサービスを保護します。



Anti-DDoS Basic は、Internet Server Load Balancer インスタンスの帯域幅に応じて、クリーニング閾値とブラックホール閾値を設定します。着信トラフィックが閾値に達すると、クリーニングまたはブラックホールがトリガーされます。

> クリーニング: インターネットからの攻撃トラフィックがクリーニング閾値を超えるか、特定の攻撃トラフィックモデルと一致すると、Alibaba Cloud Security は攻撃トラフィックのクリーンアップを開始します。クリーニング動作には、パケットフィルタリング、トラフィック速度制限、パケット速度制限などが含まれます。
>
> ブラックホール: インターネットからの攻撃トラフィック量がブラックホール閾値を超えると、ブラックホールが発生し、すべての受信トラフィックがドロップされます。

詳細については、Anti-DDoS Basic とはを参照してください。

閾値の参照

RAM アカウントを使用して SLB コンソールにログオンした後に閾値を表示できない場合は、RAM アカウントを最初に承認する必要があります。詳細については、Allow read-only access to Anti-DDoS Basic を参照してください。

閾値を表示するには、次の手順を実行します。

[Server Load Balancer コンソール]にログインします。

リージョンを選択して、そのリージョンの下のすべてのインスタンスを表示します。

ターゲットインスタンスの横にある DDoS アイコンにマウスポインタを合わせると、閾値が表示されます。詳細については、DDoS コンソールへのリンクをクリックしてください。

BPS クリーニング閾値: インバウンドトラフィックの量が BPS クリーニング閾値を超えると、クリーニングがトリガーされます。

PPS クリーニング閾値: 受信パケットの量が PPS クリーニング閾値を超えると、クリーニングがトリガーされます。

ブラックホール閾値: インバウンドトラフィックがブラックホール閾値を超えると、ブラックホールがトリガーされます。

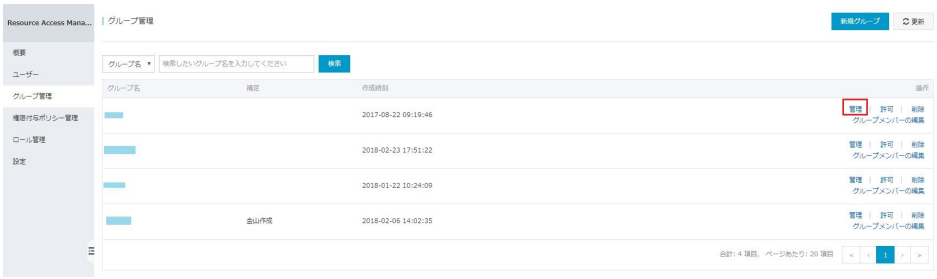
Anti-DDoS Basic への読み取り専用アクセスを許可する

Anti-DDoS Basic への RAM アカウントの読み取り専用アクセスを許可するには、次の手順に従います。

注意: プライマリアカウントを使用してアクセスを許可します。

プライマリアカウントを使用してアクセスを許可します。[RAM コンソール]にログオンします。

左側のナビゲーションペインで、**ユーザー**をクリックし、ターゲット RAM アカウントを見つけて**管理**をクリックします。



権限付与ポリシーをクリックし、**権限付与ポリシーの編集**をクリックします。

表示されたダイアログボックスで、使用可能な権限付与ポリシー名から AliyunYundunDDosReadOnlyAccessを検索し、選択した権限付与ポリシー名に追加して OK をクリックします。