# Server Load Balancer

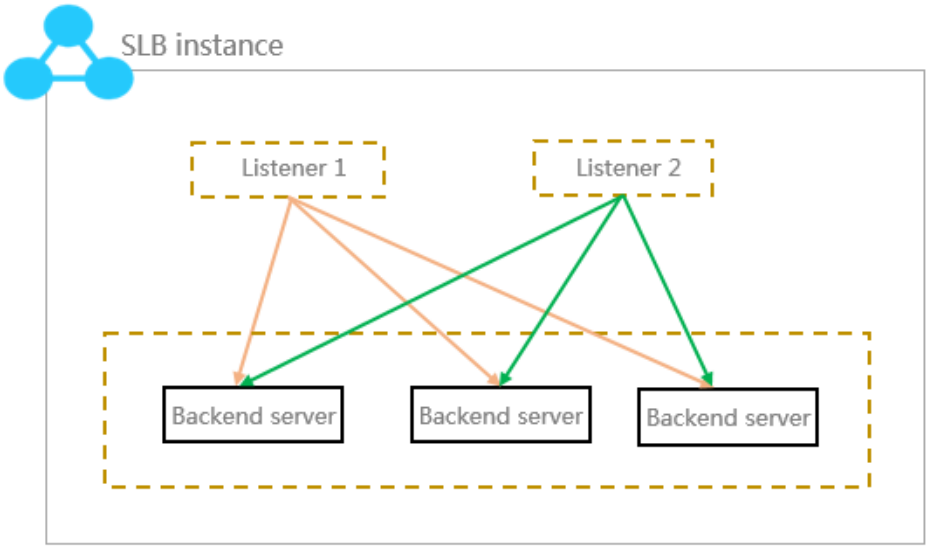
## ユーザーガイド

MORE THAN JUST CLOUD | Alibaba Cloud

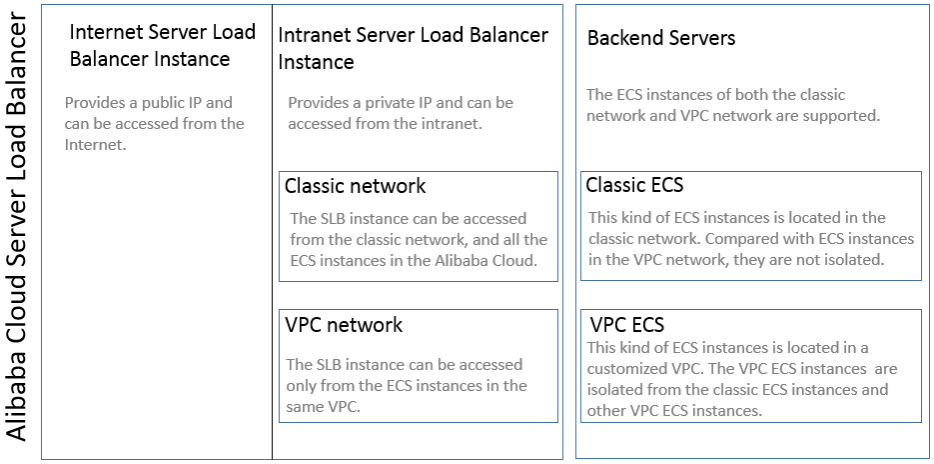
# ユーザーガイド

# SLB インスタンス

Server Load Balancerインスタンスは、Server Load Balancerサービスの実行中のエンティティです。




Alibaba Cloudは、インターネットSLBとイントラネットSKBの2種類の負荷分散サービスを提供します Server Load Balancerインスタンスを作成するとき、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Server Load Balancerインスタンスに割り当てます。



# インターネットサーバーロードバランサ

インターネットServer Load Balancerインスタンスは、構成された転送ルールに従ってバックエンドのECSサーバーにインターネットを介してクライアント要求を配布します。

インターネット Server Load Balancerインスタンスを作成すると、システムはインスタンスにパブリックIPを割り当てます。 ドメイン名をパブリックIPに解決して、パブリック・サービスを提供することができます。

# イントラネットサーバーロードバランサ

イントラネットServer Load Balancerインスタンスは、Alibaba Cloudイントラネット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、Alibaba Cloudでのみ使用できます。

イントラネットServer Load Balancerインスタンスを作成すると、システムはインスタンスにプライベートIPを割り当てます。

イントラネットServer Load Balancerインスタンスの場合、ネットワークタイプに応じて、次のようにさらに分類できます。

クラシックネットワーク

イントラネットServer Load Balancerインスタンス用のクラシックネットワークを選択すると、Server Load BalancerインスタンスのIPがAlibaba Cloudによって割り当てられ、維持されます。

VPCネットワーク

イントラネットServer Load Balancerインスタンス用のVPCネットワークを選択すると、Server Load BalancerインスタンスのIPが、そのインスタンスが属するVSwitchのCIDRから割り当てられます。VPC Sever Load Balancerインスタンスは、同じVPC内のECSインスタンスによってのみアクセスできます。

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 (slb.s1.small) | 5000 | 3000 | 1000 |
| 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。

# 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。<br>**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。<br>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。<br>- 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。<br>- 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。<br>- 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。<br>- 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。<br>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。<br>- 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。<br>- 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コンソールページで、[ リージョン選択すると、選択したリージョンのサーバーのすべての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。

# タグの概要

Server Load Balancerにはタグ機能があり、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、キーと値のペア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。

タグを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。

インスタンスに最大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。

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

# タグの追加

Server Load Balancer コンソールへログインします。

**インスタンス管理**ページで、タグを追加するターゲットインスタンスを選択します。

**タグの編集** をクリックするか **オプション** > **タグの編集**をクリックします。

**タグの編集** ダイアログで、**作成** をクリックし、次にキーと値をクリックします。 **確認** をクリックしてタグを追加します。

# タグを使用するインスタンス検索

**インスタンス管理**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、フィルタをクリアできます。

# タグの表示と検索

Server Load Balancerの左側のナビゲーションペインで、**タグの管理**をクリックします。

作成されたすべてのタグが一覧表示されます。

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

タグは、関連付けられたインスタンスから削除されると削除されます。

**インスタンス管理** ページで、ターゲットインスタンスを選択します。

**タグの編集**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、次に**確認**をクリックして、インスタンスからタグを削除します。

タグは複数のインスタンスに付けられた場合、他のインスタンスからタグを削除する必要があります。タグがインスタンスに関連付けられていない場合は、システムから削除されます。

# バックエンドサーバー

## 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。

- **マスター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。

# 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。

# 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。

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。

# リスナー

## 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。<br>●送信元アドレスベースのセッション永続性。<br>●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。<br>●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。<br>●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。 | 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 |

| | | |
|---|---|---|
| | ●しかし、高速データ伝送では、信頼性は比較的低い | |
| 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ション層プロトコル。<br>●Cookieベースのセッション永続性ソース<br>●IPアドレスを取得するには、X-Forward-Forを使用します。 | 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。<br>●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。

HTTP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。<br><br>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。<br><br>**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。<br><br>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。<br><br>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。<br><br>●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。<br><br>●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。<br><br>●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。<br><br>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。<br><br>**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。<br><br>●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。<br><br>●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。<br><br>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。 |

# レイヤー7リスナー

## 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。<br><br>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。<br><br>**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。<br><br>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。<br><br>●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。<br><br>●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。<br><br>●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。<br><br>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。<br><br>**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。<br><br>**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。<br><br>**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。<br><br>**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。<br><br>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。<br><br>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。 |
| 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。<br><br>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。<br><br>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IPを取得します。<br>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。 |

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。<br>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。<br>Server Load Balancerにアップロードする必要があります。 | はい。<br>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 | お客様ーの識別に使用されます。<br>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。<br>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。<br>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。

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。

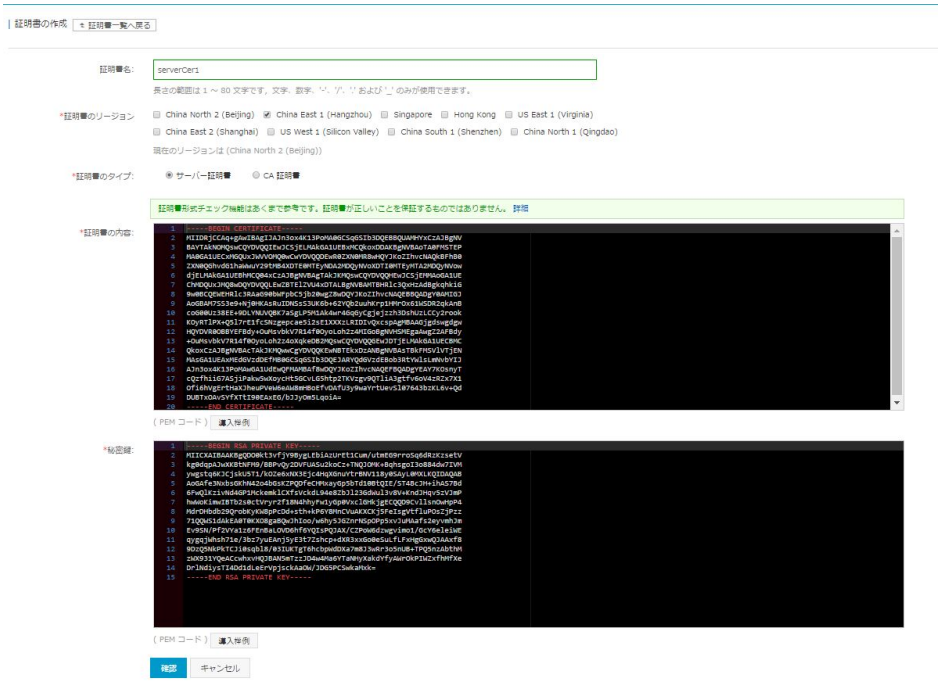
**注意**：証明書のリージョンは、Server Load Balancerインスタンスと同じである必要があります。

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 IP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。

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

# Task 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。

# Task 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。

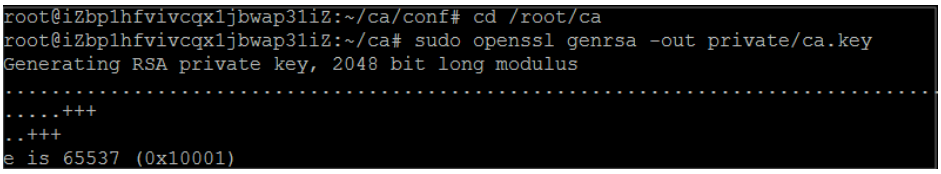
```
 [ ca ]
default_ca = foo
[ foo ]
dir = /root/ca
database = /root/ca/index.txt
new_certs_dir = /root/ca/newcerts
certificate = /root/ca/private/ca.crt
serial = /root/ca/serial
private_key = /root/ca/private/ca.key
RANDFILE = /root/ca/private/.rand
default_days = 365
default_crl_days= 30
default_md = 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
```

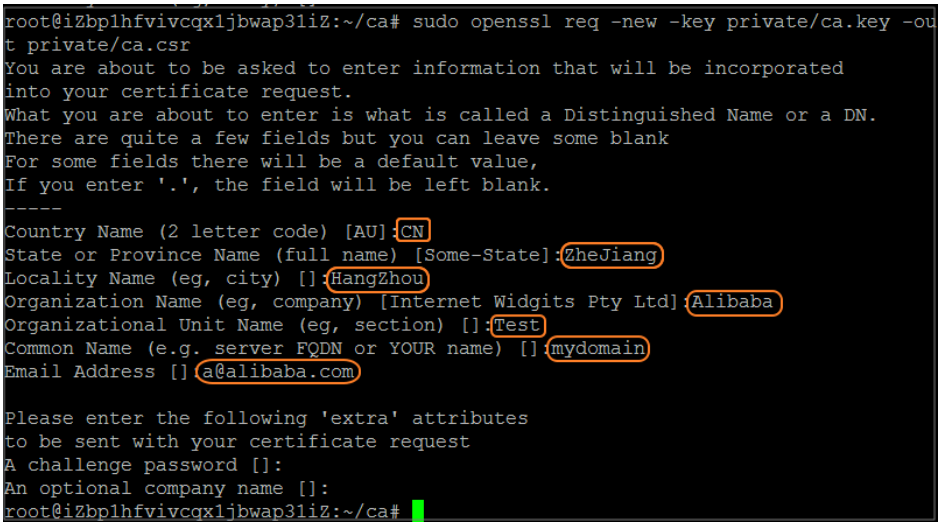
次の図は、鍵生成の例です。




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

## Task 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。

```
root@iZbp1hfvivcqx1jbwap31iZ:~/ca# sudo openssl req -new -key /root/ca/users/cli
ent.key -out /root/ca/users/client.csr
Enter pass phrase for /root/ca/users/client.key:
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 []:test
An optional company name []:Alibaba
root@iZbp1hfvivcqx1jbwap31iZ:~/ca#
```

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。

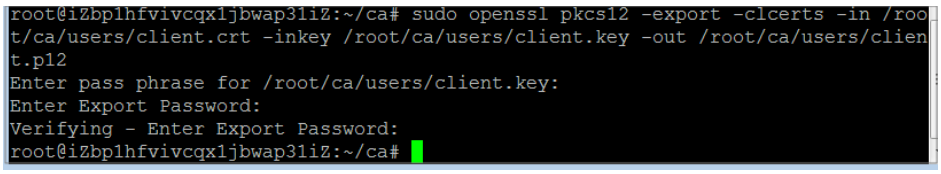
```
root@iZbp1hfvivcqx1jbwap31iZ:~/ca# sudo openssl ca -in /root/ca/users/client.csr
 -cert /root/ca/private/ca.crt -keyfile /root/ca/private/ca.key -out /root/ca/us
ers/client.crt -config "/root/ca/conf/openssl.conf"
Using configuration from /root/ca/conf/openssl.conf
Check that the request matches 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

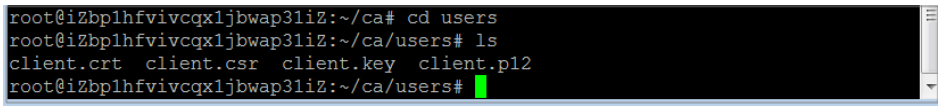
プロンプトが表示されたら、クライアントキーのパスワードを入力します。

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。




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
```




# Task 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。

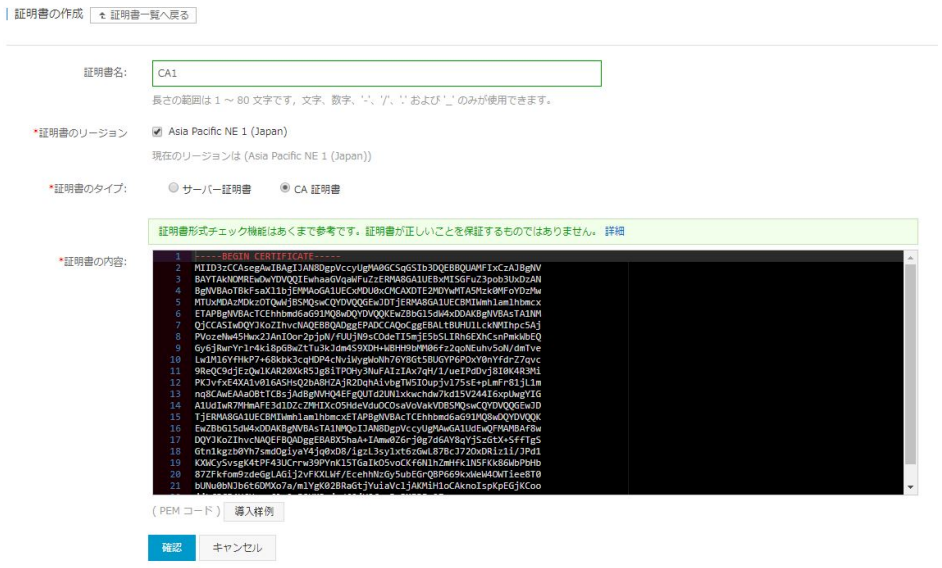
証明書のタイプ：**サーバー証明書**を選択します。

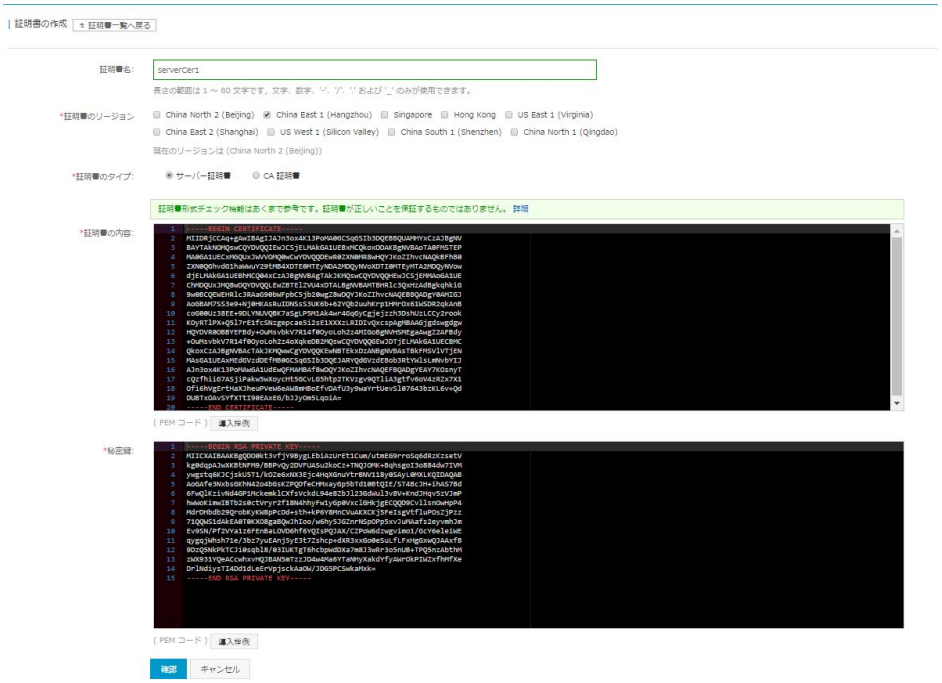
証明書の内容と秘密鍵：サーバー証明書の内容とサーバー証明書の秘密キーをコピーします。**サンプルのインポート**をクリックして、証明書の有効な形式を表示できます。

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。




# Task 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。

# Task 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。

# Task 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 IP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。

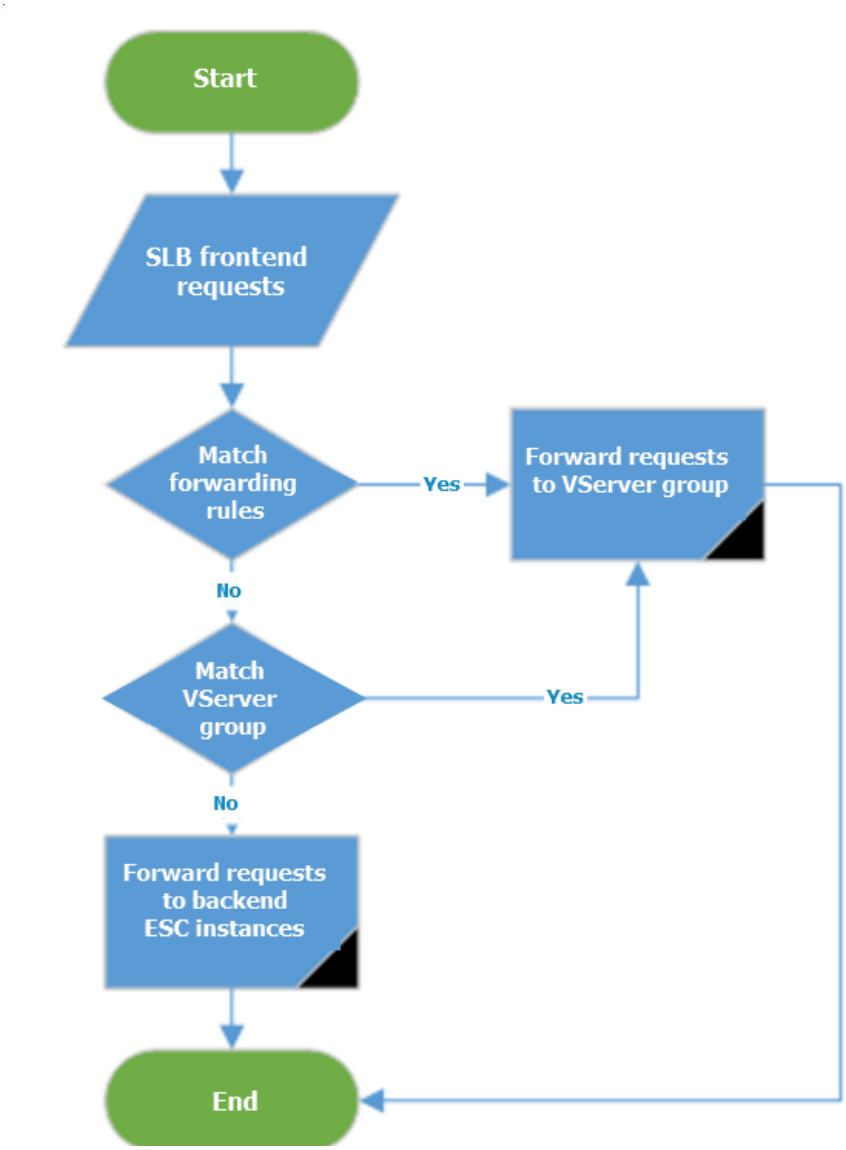
レイヤー7 Server Load Balancerは、異なる要求を異なるバックエンドサーバーに配布するために、ドメイン名または要求URLに基づく転送ルールの構成をサポートしています。レイヤ7リスナに対して、異なるVServerグループに関連付けられた異なる転送ルールを追加できます。

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。

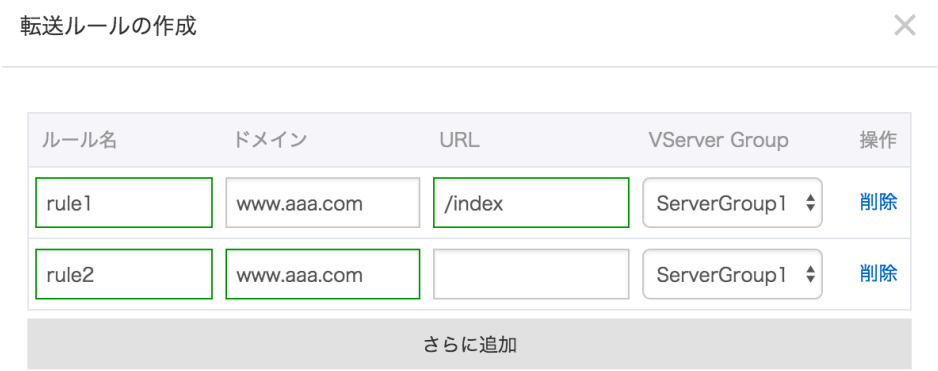
- 文字、数字、ダッシュ（ - ）、スラッシュ（ / ）、ピリオド（ % ）、疑問符（ ？ ）、ハッシュキー（ # ）、アンパサンド（ & ）のみが許可されます。
- URLはスラッシュ（ / ）で始まる必要がありますが、スラッシュ（ / ）のみで構成することはできません。スラッシュ（ / ）のみを入力すると、URL転送ルールが正しく適用されます。

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

## ヘルスチェックの設定

**[インスタンスの詳細]** ページで、**[リスナー]** > **[リスナーの追加]**をクリックします。

ヘルスチェックを設定します。
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。

**注意**：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。

# ヘルスチェックのオプション

次の表に、ヘルスチェックのオプションを示します。

| 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 ) | デフォルトでは、サーバーロードバランサは、*ECS プライベート IP : ヘルスチェックポート* を介してアプリケーションサーバー上に構成された既定のページにヘルスチェック HTTP ヘッダーリクエストを送信し、ヘルスチェックを行います。<br><br>デフォルトのページを使用しない場合は、チェックするドメイン名とページの URL を設定する必要があります。<br>- ドメイン名 : 確認するページのドメイン名と特定の URL を入力します。リクエストヘッダーに host パラメータを設定した場合は、ドメイン名を設定する必要はなく、URL だけが必要です。<br>- ヘルスチェック URL : チェックに使用されるファイルページの URL。静的ページを使用することをお勧めします。 |

| | |
|---|---|
| 通常のステータスコード ( HTTP ヘルスチェックのみ ) | リターンステータスコード。 ECS インスタンスは、指定されたステータスコードがレスポンスタイムアウト時間内に返された場合にのみ正常とみなされます。<br><br>デフォルト値 : http_2xx および http_3xx。 |
| ヘルスチェックポート | ECSインスタンスとの接続に使用するポート。<br><br>デフォルトでは、リスナーの設定時に指定されたバックエンドポートが使用されます。<br><br>**注意**：リスナーがサーバーグループを構成し、サーバーグループ内の ECS インスタンスが異なるポート番号を使用する場合は、このオプションを空のままにします。 |
| レスポンスタイムアウト | ヘルスチェックからのレスポンスを待機する時間。指定されたタイムアウト時間内にレスポンスが送信されない場合、ECS インスタンスは健全でないとみなされます。<br><br>有効値 : 1-300 秒。 UDP リスナーのデフォルト値は 10 秒で、HTTP/HTTPS/TCP リスナーのデフォルト値は 5 秒です。 |
| ヘルスチェック間隔 | ヘルスチェックが実行される頻度。<br><br>LVS クラスタ内のすべてのノードサーバーは、指定された間隔に従って独立してヘルスチェックを並列に実行します。各ノードサーバーのヘルスチェック時間は同期していないため、1 つの ECS でのヘルスチェック要求の統計情報には、ヘルスチェック間隔が反映されません。<br><br>有効値 : 1-50 秒。 UDP リスナーのデフォルト値は 5 秒で、HTTP/HTTPS/TCP リスナーの場合は 2 秒です。 |
| 異常なしきい値 | ECS サーバー上の連続したヘルスチェック失敗の数 ( 成功から失敗まで ) 。<br><br>有効値 : 2-10、デフォルト値は 3 です。 |
| 正常なしきい値 | ECS サーバー上での連続した正常性チェックの成功回数 ( 失敗から成功まで ) 。<br><br>有効値 : 2-10、デフォルト値は 3 です。 |
| ヘルスチェックリクエストとレスポンス | UDP リスナーのヘルスチェックを構成するときに、ヘルスチェック要求とレスポンスをカスタマイズできます。<br><br>**ヘルスチェックリクエスト**フィールド ( *youraccountID* など ) にリクエストを入力し、**ヘルスチェックレスポンス**フィールド ( *slb123* など ) に期待レスポンスを入力します。<br><br>さらに、アプリケーションサーバーにヘルスレスポンスロジックを追加する必要があります。たと |

| | |
|---|---|
| | えば、 *youraccountID* 、 *slb123* のリクエストを受け取った場合、リターンされます。<br><br>期待されるレスポンスが返された場合、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。 |

ホワイトリストはアクセス制御モードで、Server Load Balancerリスナーへのアクセスに許可されているIPアドレスを構成できます。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。

**注意：**

- ホワイトリストが設定されると、ホワイトリストのIPアドレスのみがServer Load Balancerリスナーにアクセスできるため、意図しないビジネスリスクが発生する可能性があります。
- ホワイトリストを設定せずにアクセス制御を有効にすると、Server Load Balancerリスナーへのアクセスは許可されません。
- ホワイトリストを設定すると、Server Load Balancerインスタンスへのアクセスが中断されることがあります。

## 操作手順

[Server Load Balancerコンソール]にログオンします。

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。

左のナビゲーションバーで**インスタンスリスナー**をクリックします。

目的のリスナーの操作列で、**オプション > アクセス制御の設定**.をクリックします。

この機能を有効にするには、Off状態の[**アクセス制御を有効化**]スイッチをクリックします。

このホワイトリストへのアクセスを許可するIPアドレスを**ホワイトリスト設定**ボックスに入力します。

単一のIPアドレスまたはIPネットワークセグメントを入力できます。複数のIPアドレスはコンマで区切り、重複したIPアドレスは許可されません。最大300のIPアドレスを入力できます。

**OK** をクリックします。

# インスタンスの帯域幅の共有

## ピーク帯域幅を設定する

PayByBandwidth請求メソッドのインスタンスの場合、インスタンスに追加されたリスナーはインスタンス帯域幅を共有します。

リスナーを追加するときに、リスナーにピーク帯域幅を設定するかどうかを選択できます。

構成：リスナーのピーク帯域幅を設定して、トラフィックを制限することができます。ただし、すべてのリスナーのピーク帯域幅の合計がインスタンスの帯域幅を超えることはできません。

無制限：リスナーにピーク帯域幅を設定しないと、他のリスナーとインスタンス帯域幅を共有します。

## どのように帯域幅を共有するのですか？

10 MBの帯域幅を持つServer Load Balancerインスタンスを作成したとします。次に、2つのリスナー（リス

ナーAとリスナーB）をインスタンスに追加し、リスナーAのピーク帯域幅を10 MBに設定し、リスナーBのピーク帯域幅を設定なしにします。

リスナーAとリスナーBの間の帯域幅使用率は次のとおりです。

トラフィックがリスナーAおよびCに到着しない場合、リスナーBは残りの帯域幅（10 MB∽4 MB = 6 MB）までしか使用できません。

リスナーAを通過するトラフィックが残りの帯域幅6 MBを超えている時にリスナーCに到着するトラフィックがない場合、リスナーBに到着する新しい接続はドロップされ、リスナーAの受信トラフィックはピーク帯域幅4 MBを超えません。

リスナーAがフルスピード（ピーク帯域幅4 MB）で実行され、リスナーBとCに到着するトラフィックが多い場合、リスナーBとCは残りの帯域幅（6 MB）を共有します。この場合、リスナーAを通過するトラフィックは、リスナーBおよびCの影響を受けず、予約ピーク帯域幅（4 MB）が保証されます。リスナーBとCのトラフィックが同じ場合、2つのリスナーが占める帯域幅は等しくなります。

したがって、リスナーに設定されたピーク帯域幅はリソース予約であり、コアビジネスには常に十分な帯域幅があることが保証されます。

# 証明書管理

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。

```
-----BEGIN CERTIFICATE-----
MIIE+TCCA+GgAwIBAgIQU306HIX4KsioTW1s2A2krTANBgkqhkiG9w0BAQUFADCB
tTELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMR8wHQYDVQQL
ExZWZXJpU2lnbiBUcnVzdCBOZXR3b3JrMTswOQYDVQQLEzJUZXJtcyBvZiB1c2Ug
YXQgaHR0cHM6Ly93d3cudmVyaXNpZ24uY29tL3JwYSoAYykwOTEvMC0GA1UEAxMm
VmVyaVNpZ24gQ2xhc3MgMyBTZWN1cmUgU2VydmVyIENBIC0gRzIwHhcNMTAxMDA4
MDAwMDAwWhcNMTMxMDA3MjM1OTU5WjBqMQswCQYDVQQGEwJVUzETMBEGA1UECBMK
V2FzaGluZ3RvbjEQMA4GA1UEBxQHU2VhdHRsZTEYMBYGA1UEChQPQW1hem9uLmNv
bSBJbmMuMRowGAYDVQQDFBFpYW0uYW1hem9uYXdzLmNvbTCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEA3Xb0EGea2dB8QGEUwLcEpwvGawEkUdLZmGL1rQJZdeeN
3vaF+ZTm8Qw5Adk2Gr/RwYXtpx04xvQXmNm+9YmksHmCZdruCrW1eN/P9wBfqMMZ
X964CjVov3NrF5AuxU8jgtw0yu//C3hWnOuIVGdg76626ggOoJSaj48R2n0MnVcC
AwEAAaOCAdEwggHNMAkGA1UdEwQCMAAwCwYDVR0PBAQDAgWgMEUGA1UdHwQ+MDww
OqA4oDaGNGh0dHA6Ly9TVlJTZWN1cmUtRzItY3JsLnZlcmlzaWduLmNvbS9TVlJT
ZWN1cmVHMi5jcmwwRAYDVR0gBD0wOzA5BgtghkgBhvhFAQcXAzAqMCgGCCsGAQUF
BwIBFhxodHRwczovL3d3dy52ZXJpc2lnbi5jb20vcnBhMB0GA1UdJQQWMBQGCCsG
AQUFBwMBBggrBgEFBQcDAjAfBgNVHSMEGDAWgBSl7wsRzsBBA6NKZZBIshzgVy19
RzB2BggrBgEFBQcBAQRqMGgwJAYIKwYBBQUHMAGGGGh0dHA6Ly9vY3NwLnZlcmlz
aWduLmNvbTBABggrBgEFBQcwAoY0aHR0cDovL1NWUlNlY3VyZS1HMi1haWEudmVy
aXNpZ24uY29tL1NWUlNlY3VyZUcyLmNlcjBuBggrBgEFBQcBDARiMGChXqBcMFow
WDBWFglpbWFnZS9naWYwITAfMAcGBSsOAwIaBBRLa7kolgYMu9BSOJsprEsHiyEF
GDAmFiRodHRwOi8vbG9nby52ZXJpc2lnbi5jb20vdnNsb2dvMS5naWYwDQYJKoZI
hvcNAQEFBQADggEBALpFBXeG782QsTtGwEE9zBcVCuKjrsl3dWK1dFiq3OP4y/Bi
ZBYEywBt8zNuYFUE25Ub/zmvmpe7p0G76tmQ8bRp/4qkJoiSesHJvFgJ1mksr3IQ
3gaE1aN2BSUIHxGLn9N4F09hYwwbeEZaCxfgBiLdEIodNwzcvGJ+2LlDWGJOGrNI
NM856xjqhJCPxYzk9buuCl1B4Kzu0CTbexz/iEgYV+DiuTxcfA4uhwMDSe0nynbn
1qiwRk450mCOnqH4ly4P4lXo02t4A/DI1I8ZNct/Qfl69a2Lf6vc9rF7BELT0e5Y
R7CKx7fc5xRaeQdyGj/dJevm9BF/mSdnclS5vas=
-----END CERTIFICATE-----
```

# 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

# 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
```

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
```

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。<br>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。<br>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。<br>- 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。<br>- 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。<br>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。<br>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。 |

# 概要

Server Load Balancer では、ユーザーへのデータ転送を安全に行うため、サーバー側およびクライアント側での HTTPS 相互認証をサポートしています。CA 証明書とサーバー証明書の両方を HTTPS リスナーにバインドし、HTTPS 相互認証を有効にしてください。

# 操作手順

1. 証明書 (サーバー証明書、CA 証明書、クライアント証明書など) を生成します。詳細については、**証明書の生成**を参照してください。

   クライアント証明書をインポートします。

   i. クライアント証明書をダウンロードします。
   ii. インポートプログラムをダブルクリックして実行します。
   iii. 証明書のパスワード入力します。
   iv. インポートした証明書を保存する場所を選択して、インポートを完了します。

HTTPS 相互認証を設定します。

i. 生成したサーバー証明書と CA 証明書をアップロードします。詳細については、**証明書のアップロード**を参照してください。

Server Load Balancer コンソールで新しい HTTPS リスナーを作成するか、既存の HTTPS リスナーを設定するときは (リスナーを作成および設定する方法の詳細については、**リスナーの設定**を参照してください)、[**相互認証**] をクリックして機能を有効にし、アップロードされたサーバー証明書および CA 証明書を選択します。

リスナーを設定すると、HTTPS 相互認証が正常に設定されます。

# 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。

# モニタリング

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。<br>- 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。<br>- 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。<br>- 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。<br>- 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。<br>- 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。<br>- 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。 |

# 前提条件

CloudMonitor サービスが有効化されていること。

## 手順

[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、対象インスタンスを確認します。

インスタンスを選択して、対応する [操作] 列の [**管理**] をクリックし、[詳細] ページを表示します。

左側のナビゲーションツリーから [インスタンスのモニタリング] を選択し、[インスタンスのモニタリング] ページを表示します。

右上の [**しきい値警告設定**] をクリックし、[CloudMonitor] > [Server Load Balancer] > [アラームルール] ページにアクセスします。

[**新しいアラームルール**] をクリックして、アラームを作成します。

アラームルール設定の詳細はServer Load Balancer モニタリングを参照してください。

# インスタンス管理

リスナー構成は、主にレイヤー4(TCP/UDP)サーバー・ロード・バランサーおよびレイヤー7(HTTP/HTTPS)サーバー・ロード・バランサーに適用されます。最大50のリスナーでインスタンスを構成できます。

## TCPリスナーの設定

### アプリケーションのシナリオ

TCPは、データを送受信できるようになる前に、他のピアエンドとの信頼性の高い接続を必要とする接続指向のプロトコルです。ファイル転送、電子メール、リモートログオンなどのシナリオのような、スピードの要件は低くなりますが、高い信頼性とデータの正確性が要求されるシナリオに適用できます。

### 推奨構成

ユーザサービスとアプリケーションステータスの迅速なコンバージェンスを促進するため、推奨されるパラ

メータ設定は次のとおりです。

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

ヘルスチェック応答のタイムアウト値を下げてレスポンスを速めますが、サービスがその値より短い時間内に処理されることを確認してください。推奨されるECSヘルスチェック失敗および成功応答時間は、次のとおりです。

- ネットワーク例外の場合のECSヘルスチェック失敗応答時間：(2 + 5)× 3 = 21秒
- ECSヘルスチェック成功応答時間：2 × 3 = 6秒

# UDPリスナーの設定

## アプリケーションのシナリオ

UDPは非接続指向のプロトコルです。データを送信する前に、相手とのハンドシェイクの代わりに直接パケット送信を行います。ビデオチャット、リアルタイムの財務相場の推移、DNSおよびIoTなど、信頼性に関するリアルタイム情報に優先するシナリオに主に適用されます。

## 推奨構成

ユーザサービスとアプリケーションステータスの迅速なコンバージェンスを促進するため、推奨されるパラメータ設定は次のとおりです。

- 応答タイムアウト時間：10秒
- ヘルスチェック間隔：5秒
- 不健全な閾値：3
- 正常な閾値：3

レスポンスのタイムアウト値を小さくすると応答速度が速くなりますが、サービスがその値よりも短い時間内に処理されるようにしてください。推奨されるECSヘルスチェック失敗および成功応答時間は、次のとおりです。

- ネットワーク例外の場合のECSヘルスチェック失敗応答時間：(10 + 5)× 3 = 45秒
- ECSヘルスチェック成功応答時間：5 × 3 = 15秒

# HTTPリスナーの設定

## アプリケーションのシナリオ

HTTPは、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ション層プロトコルであり、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識別する必要のあるアプリケーションに適用されます。

## 推奨構成

ユーザサービスとアプリケーションステータスの迅速なコンバージェンスを促進するため、推奨されるパラメータ設定は次のとおりです。

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

レスポンスのタイムアウト値を小さくすると応答速度が速くなりますが、サービスがその値よりも短い時間内に処理されるようにしてください。推奨されるECSヘルスチェック失敗および成功応答時間は、次のとおりです。

- ECSヘルスチェック失敗応答時間（ネットワーク例外の場合）：(2 + 5)× 3 = 21秒
- ECSヘルスチェック成功応答時間：2 × 3 = 6秒

# HTTPSリスナーの設定

## アプリケーションのシナリオ

HTTPSプロトコルは、クライアントとサーバー間で機密情報が送信されるシナリオに適用されます。
HTTPSは、許可されていないアクセス要求をブロックすることにより、データ伝送のセキュリティを保証します。

## 推奨構成

ユーザサービスとアプリケーションステータスの迅速なコンバージェンスを促進するため、推奨されるパラメータ設定は次のとおりです。

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

レスポンスのタイムアウト値を小さくすると応答速度が速くなりますが、サービスがその値よりも短い時間内に処理されるようにしてください。推奨されるECSヘルスチェック失敗および成功応答時間は、次のとおりです。

- ネットワーク例外の場合のECSヘルスチェック失敗応答時間：(2 + 5)× 3 = 21秒
- ECSヘルスチェック成功応答時間：2 × 3 = 6秒

HTTPS監視ポート選択の指定はありませんが、ポート443を推奨します。

# バックエンドサーバーの追加

## アプリケーションシナリオ

バックエンドサーバーの重みを追加、削除、および設定し、Server Load Balancer インスタンスのリスナーのトラフィック転送を制御することができます。

注意: バックエンドサーバーの IP アドレスと関連ポートにアクセスし、それらの実行ステータスを確認することができます。

## 考慮事項

- デフォルトでは、各 ECS インスタンスの “重み” は、平均的な転送ルールに基づいて “100” に設定されます。
- [ランニング] ステータスの ECS インスタンスのみを追加できます。
- プライベートネットワークタイプのバックエンド ECS インスタンスと共に Virtual Private Cloud (VPC) タイプまたはクラシックパブリックネットワークタイプの Server Load Balancer インスタンスを追加できますが、この ECS インスタンスは同じプライベートネットワーク上にある必要があります。

## 操作手順

Server Load Balancer コンソールにログインします。

リージョンのインスタンスを選択して、[**管理**] をクリックし、[インスタンスの詳細] ページを表示します。

左側のナビゲーションツリーから [バックエンドサーバー] を選択し、[ロードバランサーサーバーのプール] ページを表示します。

[追加されていないサーバー] タブページをクリックし、バックエンドサーバーとして追加する ECS インスタンスを選択します。

[追加] をクリックします。重みを設定するための [バックエンドサーバーを追加] ダイアログボックスが表示されます。

重みを設定し、[**確認**] をクリックします。バックエンドサーバーが正常に追加されます。

デフォルト値「100」は、平均転送ルールが使用されていることを示しています。

# VServerグループを追加する

デフォルトでは、インスタンスディメンション内のバックエンドサーバーグループが作成され、システムによってメンテナンスされます。Server Load Balancerインスタンス内のすべてのリスナーは、同じバックエンドサーバーグループを使用します。VServerグループでは、リスナーディメンション内のサーバーグループをパーソナライズできます。つまり、異なるリスナーのインスタンスは、バックエンドサーバー、ドメイン名、およびURL転送の異なるセットを使用して個々のニーズを満たすことができます。

## 検討事項

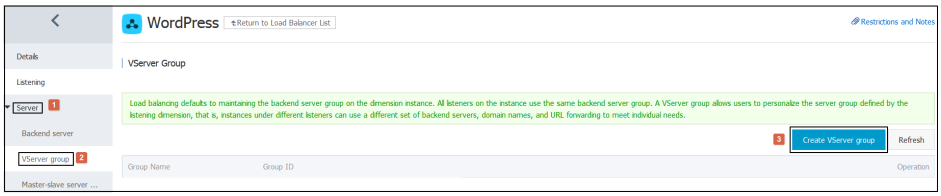
- VServerのグループに追加したバックエンドサーバーは、リスナーのリージョンと同じでなければなりません。
- バックエンドサーバーを複数のVServerグループに追加できます。
- VServerグループは、Server Load Balancerインスタンス内の複数のリスナーに追加できます。
- VServerグループは、指定されたポート番号を持つECSサーバーインスタンスのグループです。

## 手順

Server Load Balancerコンソールにログオンする

**インスタンス管理**ページで、 リージョン選択してから、VServerグループを追加するサーバーロードバランサのインスタンスIDをクリックします。

**Server** > **VServer Group** をクリックし、 **Create VServer group** をクリックします。




**Create VServer Group** ウィンドウで、次の情報を入力します。

i. グループ名を入力します。
ii. ネットワークタイプを選択します。

iii. 追加するECSサーバーをクリックします。
iv. 「**選択したサーバー・リスト**」パネルで、追加された各サーバーのポート番号と重みを入力します。
v. **確認**をクリックします。

追加されたVServerグループが **VServer Group** ページに表示されます。

- **表示** をクリックして、VServerグループの設定を表示します。
- **編集** をクリックして、VServerグループの設定を変更します。
- **削除** をクリックして、VServerグループを削除します。

# 設定の準備

## アプリケーションシナリオ

- Alibaba Cloud アカウントがない場合は、登録する必要があります。
- Alibaba アカウントがある場合は、そのまま Alibaba Cloud の Web サイトにログオンできます。

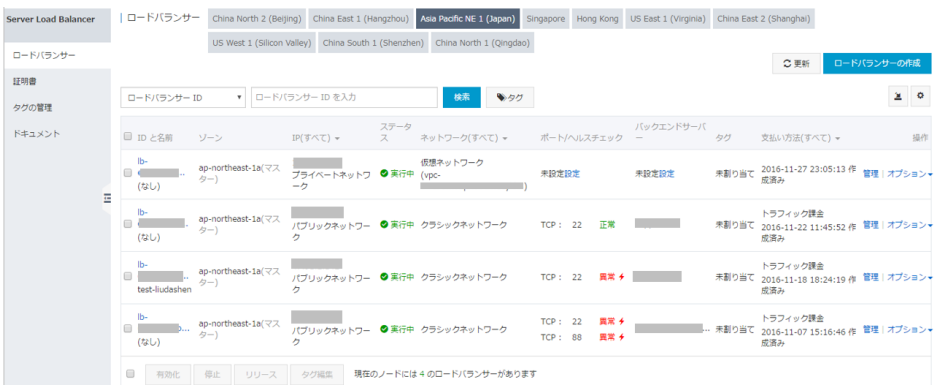
## 操作手順

1. Alibaba Cloud アカウントがあるかどうかを確認します。

- ある場合は、そのまま次のステップを実行します。
- ない場合は、「アカウント登録」の手順に従ってアカウントを登録します。

コンソールからログインします。

[Server Load Balancer] をクリックし、Server Load Balancer コンソールにアクセスします。



Alibaba Cloud ECS を使用したことがない場合は、「クイックスタート」を参照し、ECS 操作の説明に基づいてサービスを購入してください。

# Server Load Balancer マルチゾーン機能

信頼性が高く、いっそう安定した Server Load Balancer サービスをユーザーに提供するため、Alibaba Cloud Server Load Balancer は既に各リージョンにマルチゾーンをデプロイして、マシンルーム間のディザスタリカバリを実現しています。この方式には次のような目的があります。プライマリゾーンのマシンルームで障害が発生して使用できないとき、Server Load Balancer は迅速に (約 30 秒の中断で) バックアップゾーンのマシンルームに切り替えて、サービス機能を復元できます。プライマリゾーンが復元すると、Server Load Balancer サービスはサービス提供をプライマリゾーンのマシンルームに自動的に戻します。

## よくある質問

Q: ゾーンとは何ですか。

A: クラウドプロダクトゾーンは独立したインフラストラクチャセットであり、通常はインターネットデータセンター (IDC) と呼ばれます。インフラストラクチャ (ネットワーク、電源、空調など) はゾーンごとに独立しています。したがって、あるゾーンのインフラストラクチャで障害が発生しても、他のゾーンに影響はありません。

Q: 一般に、マルチゾーン機能はどのようなディメンションに基づいていますか。

A: ゾーンは特定のリージョンに属します。1 つのリージョンが 1 つ以上のゾーンを持つことができます。ほとんどのリージョンでは、SLB サービスは 2 つのゾーンにデプロイされます。

Q: 各リージョンの Server Load Balancer ゾーンの具体的な詳細を教えてください。

A: 各リージョンのゾーンの詳細は次のとおりです。

| リージョン | ゾーンタイプ | 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 | ゾーン D | ゾーン B |
| | マルチゾーン | ゾーン B | ゾーン D |
| 中国北部 2 (北京) | シングルゾーン | ゾーン B | - |
| 中国南部 1 (深圳) | マルチゾーン | ゾーン A | ゾーン B |
| | マルチゾーン | ゾーン B | ゾーン A |
| 中国北部 1 (青島) | マルチゾーン | ゾーン A | ゾーン B |
| | マルチゾーン | ゾーン B | ゾーン A |
| 中国東部 2 (上海) | マルチゾーン | ゾーン A | ゾーン B |
| | マルチゾーン | ゾーン B | ゾーン A |

| | | | |
|---|---|---|---|
| 香港 | シングルゾーン | ゾーン B | - |
| 米国西部 (シリコンバレー) | マルチゾーン | ゾーン 1A | ゾーン 1B |
| | マルチゾーン | ゾーン 1B | ゾーン 1A |
| 米国東部 (バージニア) | シングルゾーン | ゾーン 1A | - |
| アジア太平洋 (シンガポール) | シングルゾーン | ゾーン A | - |
| アジア太平洋 (日本) | シングルゾーン | ゾーン A | - |

説明:

現在、各セールスリージョンの Server Load Balancer サービスには固有のゾーン属性があります。各リージョンは “マルチゾーン” か “シングルゾーン” です。

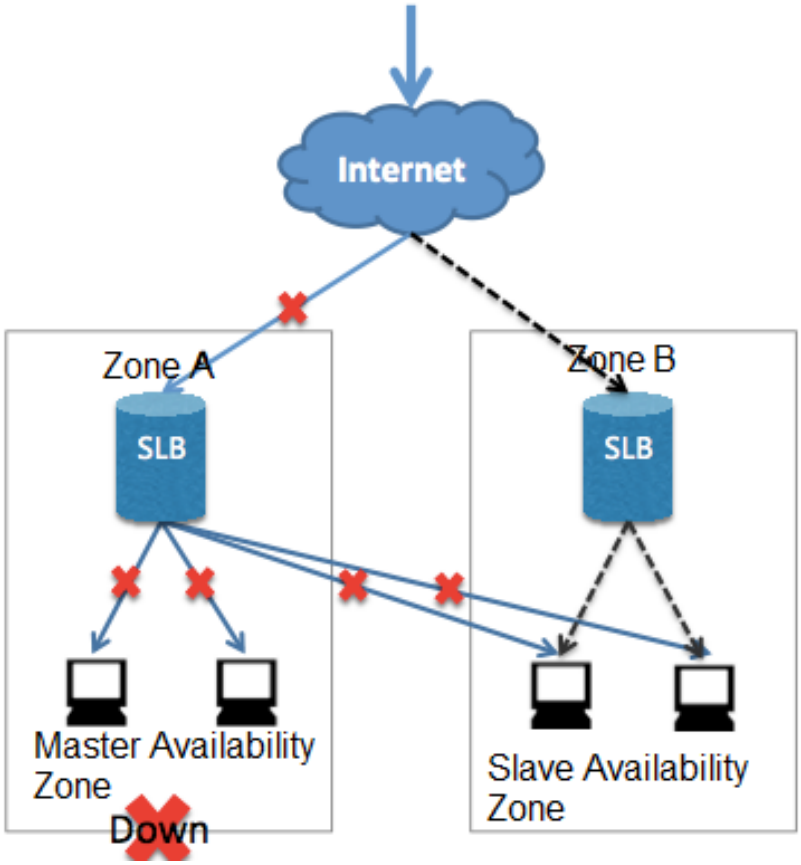
将来的に、北京リージョンはマルチゾーンをサポートするようになります。

Q: シングルゾーンリージョンとマルチゾーンリージョンは何が違うのですか。

A: シングルゾーンリージョンでは、ユーザーが作成するインスタンスはそのリージョンの 1 つのゾーンにのみ格納できます。ユーザーがマルチゾーンリージョンで作成したインスタンスは、2 つのゾーンに同時に格納できます。デフォルトでは、インスタンスはプライマリゾーンに格納できます。ただし、プライマリゾーンに障害がある場合は、インスタンスはバックアップゾーンに自動的に切り替わります。これによりローカル可用性が大きく向上します。

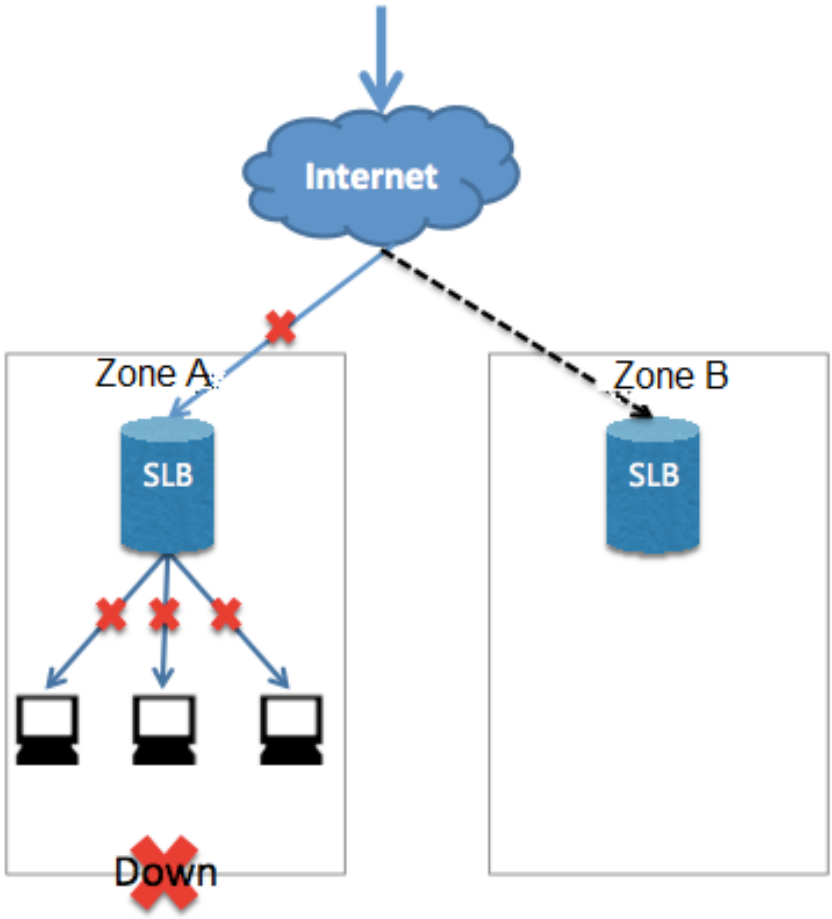
Q: Server Load Balancer マルチゾーン機能は他のプロダクトとどのように連携して高可用性ソリューションまたは低レイテンシソリューションをサポートするのですか。

A: リージョンの信頼性と可用性を高めるには、ローカルディザスタリカバリをサポートするマルチゾーンリージョンの選択に加えて、バックエンドサーバーのデプロイを検討することをお勧めします。例:




- ECS のインスタンスは、単一の Server Load Balancer インスタンスで異なるゾーンにバインドされます。これにより、ゾーン A が正常に動作しているときは、ユーザーのアクセストラフィックは上図の青い線のパスを通ります。ゾーン A で障害が発生すると、黒い線のパスに分散されます。これにより、1 つのゾーンの障害によってサービスが利用できなくなるのを防ぎます。

異なるプロダクトのゾーンを選択することにより、レイテンシを抑えることができます。例:




- ECS のインスタンスは、単一の Server Load Balancer インスタンスでプライマリゾーン (ゾーン A) にバインドされます。これにより、ゾーン A が正常に動作しているときは、ユーザーのアクセストラフィックは上図の青い線のパスを通ります。ゾーン A で障害が発生すると、黒い線のパスを通ります。このデプロイメント方法では、高可用性は低下しますがレイテンシは大きく減少します。

Q: マルチゾーン機能に料金はかかりますか。

A: 現在、マルチゾーン機能は無料です。

## プロダクトの制限と機能向上プラン

Server Load Balancer マルチゾーン機能には現在、次の制限があります。このプロダクトの改善が続けられています。

1) 各リージョンの属性は、マルチゾーンかシングルゾーンかの 1 つだけです。

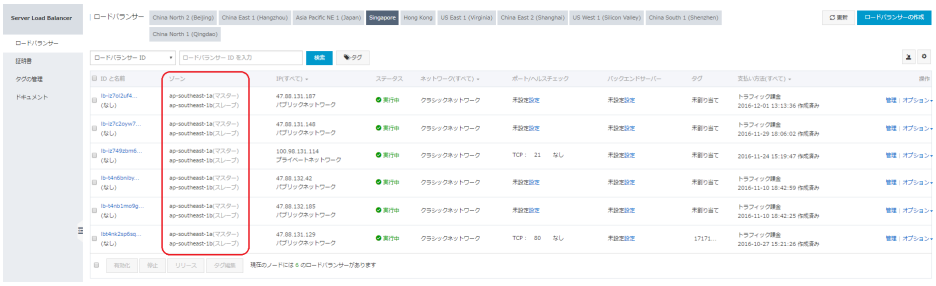
2) ユーザーは設定を変更できません。作成時にプライマリゾーンまたはバックアップゾーンを選択すると、後では変更できません。

## コンソールユーザーマニュアル

## 購入ページ

ログインして特定のリージョンを選択すると、ゾーンのタイプ (シングルゾーン/マルチゾーン) が表示されます。マルチゾーンの場合、プライマリゾーンだけを選択する必要があり、バックアップゾーンは自動的に表示されます。シングルゾーンの場合は、使用できるプライマリゾーンだけが表示されます。

## インスタンスリストページ



インスタンスリストページにはユーザーのプライマリ/バックアップゾーンが表示されます。

## API

マルチゾーン機能には全部で 4 つの API が関連付けられています。

CreateLoadBalancer: 指定したゾーンにインスタンスを作成します

DescribeLoadBalancers: インスタンスの情報を照会します

DescribeLoadBalancerAttribute: Server Load Balancer インスタンスの属性を照会します

DescribeZones: 特定のリージョンのゾーンの情報を照会します

詳細については、ドキュメントセンターの『API ユーザーマニュアル』で LoadBalancer API のセクションを参照してください。

Server Load Balancer コンソールで、Server Load Balancer インスタンスの Anti-DDoS Basic サービスの閾値を表示できます。

**注意：**この機能は現在、次のリージョンで利用可能です：

中国北部 1 (青島)

中国北部 2 (北京)

中国東部 1 (杭州)

中国東部 2 (上海)

中国南部 1 (深セン)

香港

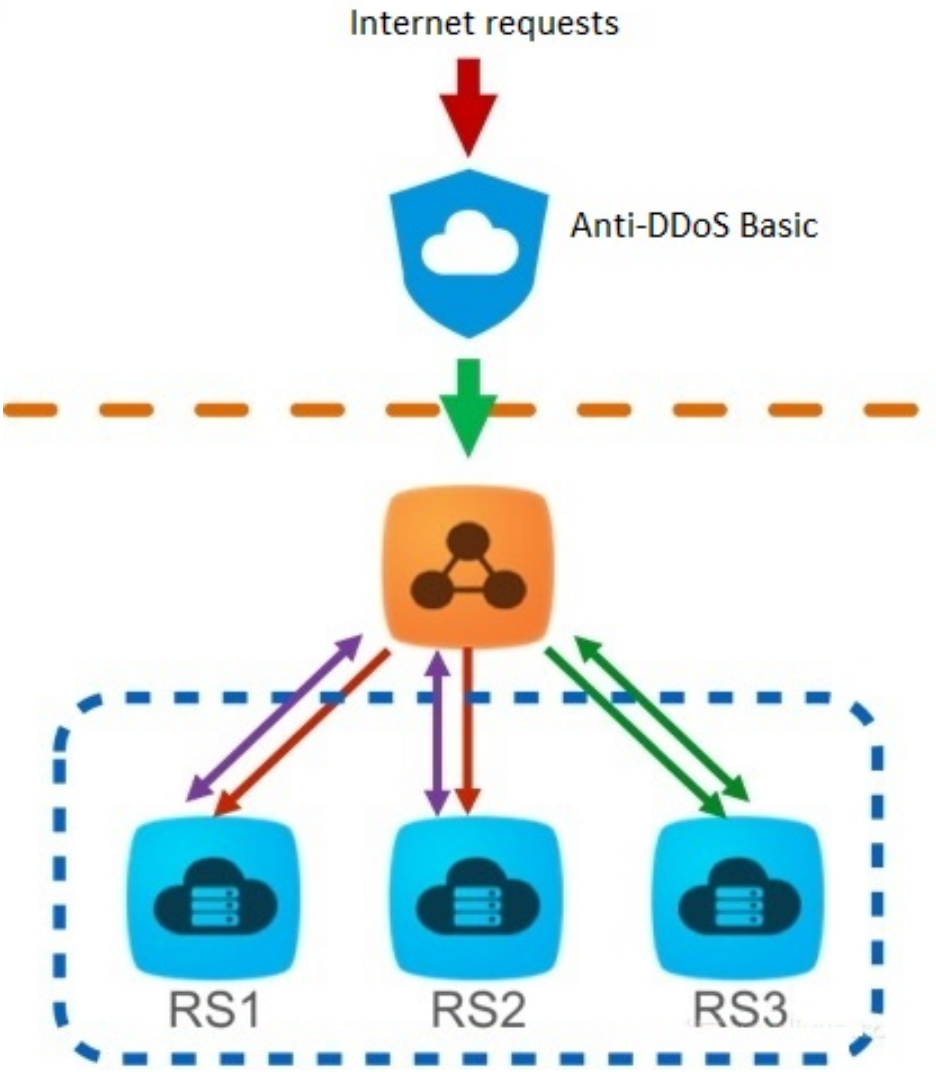
シンガポール

米国東部 1 (バージニア)

米国西部 1 (シリコンバレー)

# DDoS 保護

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 Anti-DDoS Basic を経由して Server Load Balancer に到達し、Anti-DDoS Basic は DDoS 攻撃を検出して防御します。Anti-DDoS Basic は、SYN Flood 攻撃、UDP Flood 攻撃、ACK Flood 攻撃、ICMP Flood 攻撃と DNS Flood 攻撃などの DDoS 攻撃を防御します。




Anti-DDoS Basic は、パブリックネットワーク 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が設定されます。受信トラフィックがしきい値に達すると、DDoS 保護がトリガーされます。

> フローのクリーニング：インターネットの受信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、フロークリーニングがトリガーされ、攻撃トラフィックが遮断されます。
>
> ブラックホール：インターネットのトラフィックがデフォルトのトラフィックの閾値を超えると、ブラックホールがトリガーされます。クラスタのセキュリティのために、すべてのトラフィックは破棄されます。

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

DDoS 閾値は、Server Load Balancer コンソールで確認できます。

SLB コンソールにログインします。

リージョンとターゲット SLB インスタンスを見つけます。

マウスポインタを DDoS アイコンの上に移動して、閾値を表示します。

BPS閾値：受信トラフィックが閾値を超えると、フロークリーニングがトリガーされます。

PPS閾値：受信トラフィックが閾値を超えると、フロークリーニングがトリガーされます。

ブラックホール閾値：受信トラフィックが閾値を超えると、ブラックホールがトリガーされます。