

Server Load Balancer

ベストプラクティス

ベストプラクティス

パフォーマンス専有型インスタンスの使用方法

- 1. パフォーマンス専有型インスタンスとは何ですか
- 2. パフォーマンス専有型インスタンスはどのように請求されますか
- 3. パフォーマンス専有型インスタンスのスペックに関する料金
- 4. パフォーマンス専有型インスタンスのスペックを選択する方法
- 5. パフォーマンス専有型インスタンスの構成変更の制限
- 6. Alibaba Cloud は、いつからパフォーマンス専有型インスタンスの課金を始めますか
- 7. Alibaba Cloud は、パフォーマンス専有型インスタンスでスペック料金を請求し始めた後、共有パフォーマンス型インスタンスに対して追加料金を請求しますか
- 8. 場合によっては、パフォーマンス専有型インスタンスがスペックで定義されている性能限界に達しないことがあるのはなぜですか
- 9. 低スペックのパフォーマンス専有型インスタンスのパフォーマンスが、共有パフォーマンス型インスタンスのパフォーマンスよりも悪い場合があるのはなぜですか
- 10. 今後も共有パフォーマンス型インスタンスを購入できますか
- 11. イントラネットの SLB インスタンスにはスペック料金が請求されますか

1. パフォーマンス専有型インスタンスとは何ですか？

パフォーマンス専有型インスタンスは、保証されたパフォーマンスメトリックを提供します。

対照的に、共有パフォーマンス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。

Alibaba がパフォーマンス専有型インスタンスを起動する前に、すべてのインスタンスが共有パフォーマンスのインスタンスになります。コンソールでインスタンスの種類を表示できます。

次の図に示すように、ターゲットパフォーマンス専有型インスタンスの緑色のアイコンにマウス・ポインターを移動すると、パフォーマンス・メトリックを表示できます。

インスタンス管理

China North 1 (Qingdao) China North 2 (Beijing) China North 3 (Zhangjiakou)
 China East 1 (Hangzhou) China East 2 (Shanghai) China South 1 (Shenzhen) Hong Kong
 Asia Pacific NE 1 (Japan) Singapore Australia 1 (Sydney) US East 1 (Virginia)
 US West 1 (Silicon Valley) Dubai

更新 ロードバランサの作成

ロードバランサ名 ロードバランサ名で検索 検索 タグ

ID と名前	ゾーン	IP(すべて)	ステータス	ネットワーク	ポート/ヘルスチェック	バックエンドサーバー	インスタンスタイプ	帯域幅の支払い方法(すべて)	インスタンスの支払い方法(すべて)	操作
lb-7okdzt5y1... (なし)	cn-qingdao-b(マスター) cn-qingdao-c(スレーブ)	139.129.107.188(パブリックネットワーク)	実行中	クラシックネットワーク	未設定	未設定	パフォーマンス専有型 slb.s1.small	トラフィック課金	従量課金 2017-11-15 09:51:01 作成済み	管理 オプション

有効化 停止 リリース タグ編集 現在のリージョンはロードバランサがあります

合計: 1 項目, ページあたり: 10 項目

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。接続の最大数がスペックの限界に達すると、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 がスペックの限界に達すると、新しい接続が切断されます。

クエリ/秒（QPS）

1 秒間に処理できる 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 がスペックの限界に達すると、新しい接続が切断されます。

Alibaba Cloud Server Load Balancer は、パフォーマンス専有型インスタンスでは次のスペックを提供しています。

スペック		最大接続	CPS	QPS
スペック1	Small I (slb.s1.small)	5000	3000	1000
スペック2	Standard I (slb.s2.small)	50000	5000	5000
スペック3	Standard II (slb.s2.medium)	100000	10000	10000
スペック4	Higher I (slb.s3.small)	200000	20000	20000

スペック5	Higher II (slb.s3.medium)	500000	50000	30000
スペック6	Super I (slb.s3.large)	1000000	100000	50000

より大きなスペックが必要な場合は、サポートセンターに連絡してください。

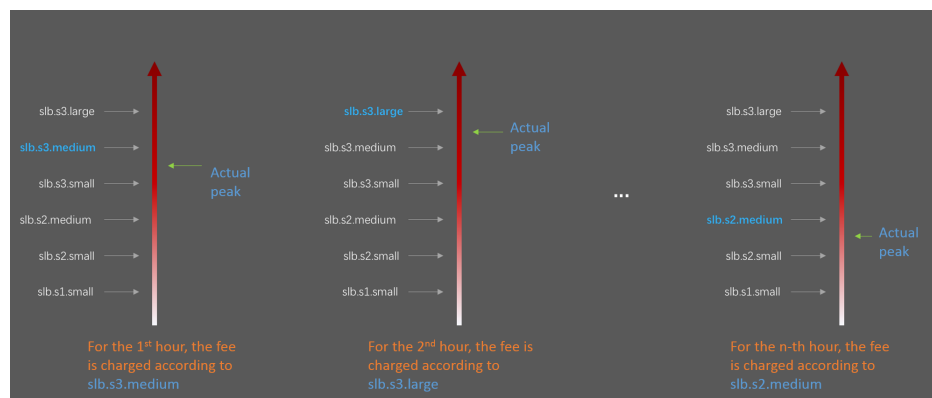
2. パフォーマンス専有型インスタンスはどのように請求されますか？

スペック保証料は、パフォーマンス専有型インスタンスに次のように請求されます。

パフォーマンス専有型インスタンスの料金=インスタンス料金+トラフィック料金+スペック料金

共有パフォーマンス型インスタンスの料金=インスタンス料金+トラフィック料金

指定料金は、実際の使用量に基づいて請求されます。つまり、指定したスペックにかかわらず、実際に使用されたスペック（パフォーマンス）に従ってスペック料金が請求されます。スーパー I（slb.s3.large）スペック（Max Connection 1,000,000、CPS 500,000、QPS 50,000）を選択し、最初の 1 時間のインスタンスの実際の使用量は次のとおりです。最大接続 90,000、CPS 9,000、QPS 8,000。この時間のインスタンスのサービスインデックスは、Super I（slb.s3.large）スペックの上限に達していませんが、slb.s3.small と slb.s3.medium の間にあります。したがって、スペック手数料は slb.s3.medium（整数に切り上げる原則）に従って請求されます。それ以降の各時間の指定料金は、次の図に示すようにこの方法で計算されます。



したがって、パフォーマンス専有型インスタンスは自動スケーリング（課金）が可能です。購入時にお客様が選択したスペックはスケーリングの上限です。たとえば、slb.s3.medium を選択すると、インスタンスが最大で slb.s3.medium のスペックに到達できることを意味します。

注：イントラネット SLB インスタンスを共有パフォーマンス型インスタンスまたはパフォーマンス専有型インスタンスに設定することもできます。その料金はイントラネット SLB インスタンスに対しても同様に課金

されます。

3. パフォーマンス専有型インスタンスのスペックに関する料金

異なるスペックのパフォーマンス専有型インスタンスには、異なるスペックの料金が課金されます。指定料金に加えて、インスタンスの設定料金とトラフィック料金も支払う必要があります。詳細については、請求を参照してください。

4. パフォーマンス専有型インスタンスのスペックを選択するにはどうすればよいですか？

スペック料金は実際の使用量に基づいて計算されるため、ほとんどのユーザーにとって最高のスペック、つまり slb.s3.large を選択することをお勧めします。このようにして、サービスの柔軟性（スケーラビリティ）が確保され、追加コストを支払う必要はありません。使用量が slb.s3.large のスペックに達しないと思う場合は、slb.s3.medium などのより合理的なスペックに設定することもできます。

レイヤー 4 リスナーの重要な要素は、TCP キープアライブ接続の同時接続数です。最大接続数はキーメトリックと見なされます。ビジネスシナリオに応じて、同時接続の最大数を見積もり、適切なスペックを選択します。

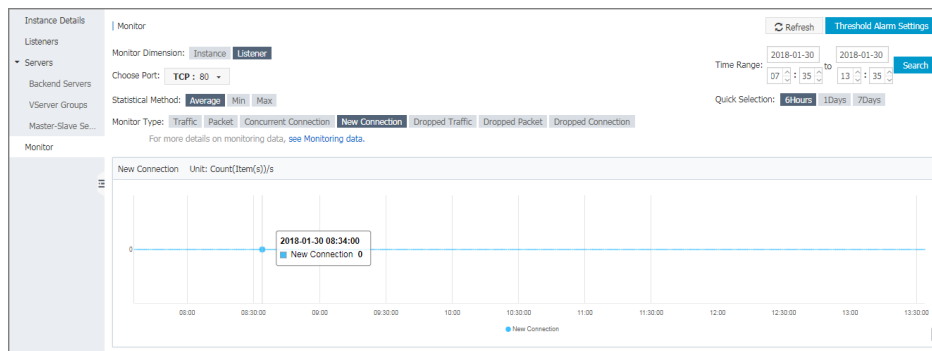
レイヤー 4 リスナーの主要な要素は QPS パフォーマンスです。QPS は、レイヤ7アプリケーションシステムのスループットを決定します。同様に、経験に基づいて QPS を見積もる必要もあります。スペックの最初の選択後、ビジネスストレステストと実際のテストの間にスペックを調整できます。

パフォーマンス専有型インスタンスによって導入された他の監視メトリックを使用して、トラフィック傾向、ピークトラフィックなどをチェックして、より正確な選択を行います。詳細は、監視データを参照してください。

同時接続の監視



新しい接続を監視する



モニター QPS

5. パフォーマンス専有型インスタンスの構成変更の制限

次の図に示すように、パフォーマンス専有型インスタンスの構成を変更できます。

Server Load Balancer

インスタンス管理

証明書

タグの管理

ドキュメント

ロードバランサ

ロードバランサで検索

検索

タグ

ID と名前	ゾーン	IP(すべて)	ステータス	ネットワーク(すべて)	ポート/ヘルムスタック	バックエンドサーバー	インスタンスタイプ	監視	変更	操作
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...

現在の設定

インスタンス名: ...

課金項目: 設定料金

プライマリ: 日本ゾーン A

課金サイクル: 1時間

課金タイプ: トラフィック

インスタンス: インターネット

リージョン: 東京 (日本)

SLB維持料金: はい

Anti-DDoS: 有効

ゾーンタイプ: シングルゾーン

設定のアップグレード

注意：

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。

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。

インスタンスタイプまたは容量を変更した後で、SLB インスタンスの IP は変更されません。

6. Alibaba Cloud はいつからパフォーマンス専用型インスタンスに対して課金を始めますか？

Alibaba Cloud は、2018 年 4 月以降、パフォーマンス専用型インスタンスについての料金を請求し始め、引き続き共有パフォーマンス型インスタンスを販売します。詳細な情報については、公式サイト の発表、メッセージ、電子メールに注意してください。

7. その他のセキュリティ設定で Alibaba Cloud は、パフォー

マンス専有型インスタンスで料金を請求し始めた後、共有パフォーマンス型インスタンスに対して追加料金が課金されますか？

オリジナルの共有パフォーマンス型インスタンス（パフォーマンス専有型インスタンスに変更しない場合）は同じままであり、料金については請求されません。それらをパフォーマンス専有型インスタンスに手動でアップグレードすることができます。アップグレード後、料金が請求されます。

設定のアップグレード



8. 場合によってはパフォーマンス専有型インスタンスがスペックで定義されている性能限界に達しないことがあるのはなぜですか？

これは、ショート・ボードの原則によるものです。

パフォーマンス専有型インスタンスは、3つのメトリックが同時にスペックの制限に達することを保証するものではありません。つまり、あるメトリックが最初にその限界に達すると、その制限がトリガーされます。

たとえば、より高いスペックの I (slb.s3.small) のパフォーマンス専有型インスタンスを購入したとします。インスタンスの QPS が 20,000 に達したにもかかわらず、最大接続数が 200,000 に達しない場合、QPS が制限に達しているため、新しい接続は引き続きドロップされます。

注： Anti-DDoS Pro を使用してソース IP がほとんどない場合、HTTPS トラフィックは不均衡になる可能性があります。チケットを起票して問題を解決することができます。

9. 低スペックのパフォーマンス専有型インスタンスのパフォーマンスが、共有型パフォーマンスインスタンスのパフォー

マンスよりも悪い場合があるのはなぜですか？

共有型パフォーマンスインスタンスでは、すべてのリソースが共有されます。トラフィック負荷が高い場合、その性能を保証することはできません。ただし、パフォーマンス専有型インスタンスは、いつでもパフォーマンスを保証できます。スペック料金は実際の使用量に基づいて計算されるため、最高のスペックを選択することをお勧めします。

10. 今後も、共有型パフォーマンス型インスタンスを購入できますか？

はい。ただし、共有型パフォーマンス型インスタンスは将来使用できなくなります。登録された電子メールアカウントまたは Alibaba Cloud Web サイトで今後の通知を確認してください。

11. イントラネットのSLBインスタンスにはスペック料金が請求されますか？

イントラネットの共有型パフォーマンス・インスタンスを選択すると、指定料金は課金されません。イントラネットパフォーマンス専有型インスタンスを選択した場合、そのスペック料金はインターネット SLB インスタンスと同じ方法で課金されます。イントラネット SLB インスタンスには、インスタンス料金とトラフィック料金は掛かりません。

クライアントの実際 IP アドレスの取得

実際 IP アドレス取得の概要

Alibaba Cloud Server Load Balancer は、クライアントの実際の IP アドレスを取得する機能を提供します。この機能は、デフォルトで有効になっています。

レイヤー 4 ロードバランシングサービス（TCP プロトコル）の場合、リスナーは、要求ヘッダーを変更せずに、クライアント要求をバックエンド ECS サーバーに配布します。したがって、追加の設定を行うことなく、バックエンドの ECS サーバから実際の IP アドレスを取得することができます。

レイヤ 7 ロードバランシングサービス（HTTP/HTTPS プロトコル）の場合は、アプリケーションサーバを設定してから、X-Forwarded-Forヘッダを使用してクライアントの実際の IP アドレスを

取得する必要があります。

注意：HTTPS ロードバランシングサービスの場合、SSL 証明書はフロントエンドリスナーで設定され、バックエンドは引き続き HTTP プロトコルを使用します。したがって、アプリケーションサーバー上の構成は、HTTP および HTTPS プロトコルと同じです。

リスナーの追加

×

1.モニター配置
2.ヘルスチェック
3.成功

フロントエンドプロトコル [ポート]*

TCP

:

ポートの入力範囲は 1 ~ 65535 です。

バックエンドプロトコル [ポート]*

TCP

:

ポートの入力範囲は 1 ~ 65535 です。

帯域幅:

制限なし 設定

帯域幅ピークは、トラフィック量に応じて課金されるインスタンスに対しては制限されません。入力範囲は 1 ~ 5000 M です

転送ルール

重み付きラウンド

利用サーバグループ:

作成後に自動的に有効化する:

有効化

高度な設定

クライアントの送信元 IP の取得:

有効化(デフォルト)

セッションの保持:

無効化

TCP プロトコルのセッションは IP アドレスに基づいて維持され、同じ IP アドレスからのリクエストは同じバックエンドクラウドサーバーが受け取ります

Web アプリケーションの設定

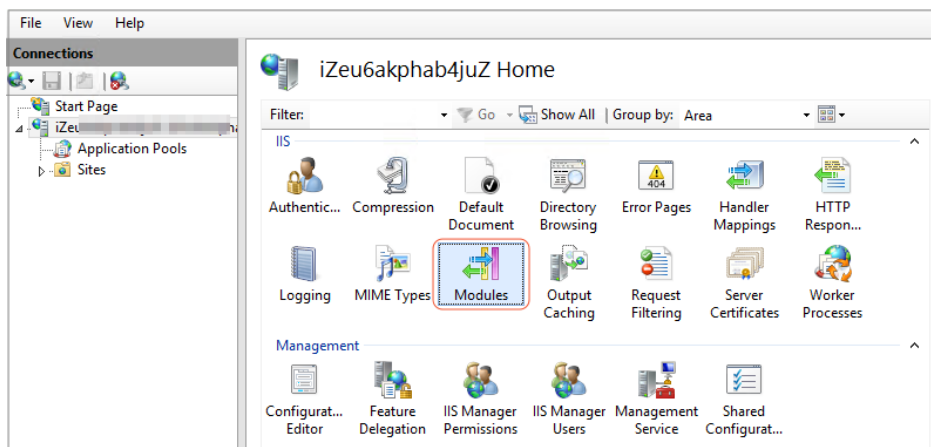
このセクションでは、Web アプリケーションの設定に使用される一般的な方法を紹介します。

IIS7/IIS8 の構成

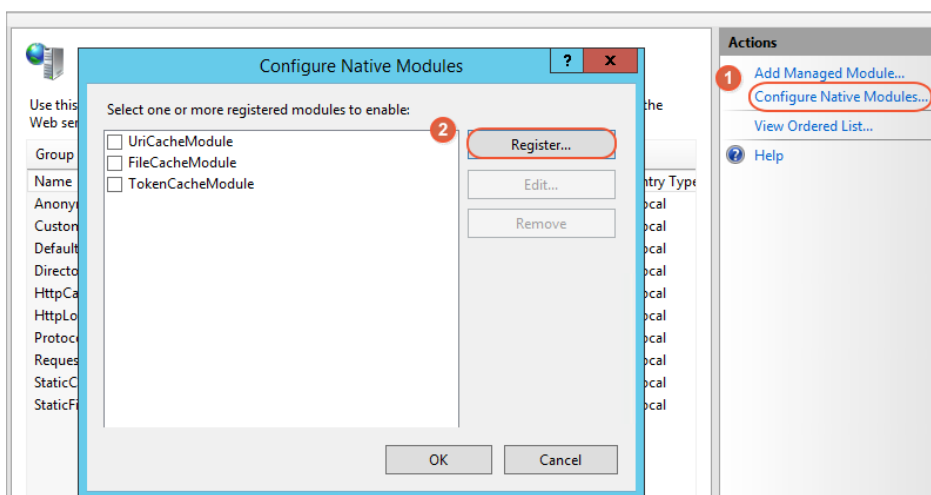
F5XForwardedFor をダウンロードして解凍します。

F5XFFHttpModule.dll と F5XFFHttpModule.ini ファイルを解凍したフォルダから C : \F5XForwardedFor\ のようなフォルダにコピーしてください。IIS プロセスにこのフォルダへの書き込み権限があることを確認します。

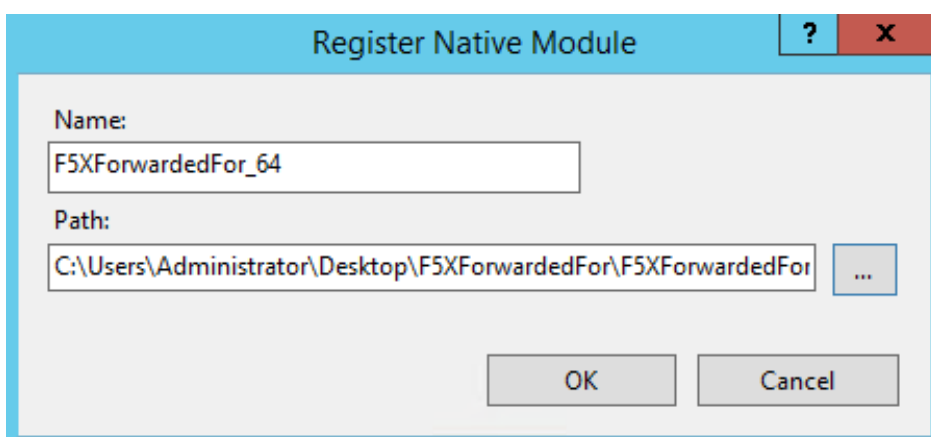
IIS マネージャーを開き、**モジュール** 機能をダブルクリックします。



ネイティブモジュールの設定 をクリックし、**Register** をクリックします。

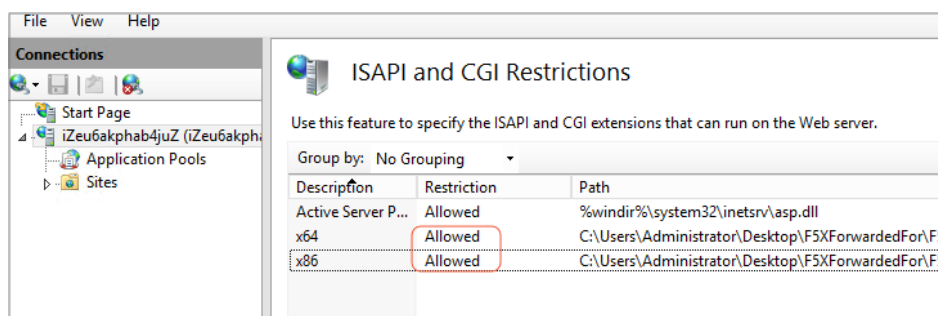


コピーした .dll ファイルを追加します。



.dll ファイルに ISAPI と CGI の制限を加え、**Allowed** に制限を設定します。

ISAPIおよびCGIアプリケーションがインストールされていることを確認します。



IIS マネージャを再起動します。

Apache の設定

以下のコマンドを実行して mod_rpafモジュールをインストールします。

```
wget http://stderr.net/apache/rpaf/download/mod_rpaf-0.6.tar.gz
tar zxvf mod_rpaf-0.6.tar.gz
cd mod_rpaf-0.6
/alidata/server/httpd/bin/apxs -i -c -n mod_rpaf-2.0.so mod_rpaf-2.0.c
```

/alidata/server/httpd/conf/ httpd.confファイルを開き、次の情報をコンテンツの最後に追加します。

```
LoadModule rpaf_module modules / mod_rpaf-2.0.so
RPAFenable On
RPAFsethostname On
RPAFproxy_ips IP_address
RPAFheader X-Forwarded-For
```

RPAFproxy_ips : IPアドレスはServer Load BalancerインスタンスのIPアドレスではありません。IPアドレスを見つけるためにApacheログをチェックしてください。通常、2つのIPアドレスが入力されます。

次のコマンドを実行して、Apacheサーバーを再起動します。

```
/alidata/server/httpd/bin/apachectl restart
```

Nginx の設定

次のコマンドを実行して http_realip_module をインストールします。

```
wget http://soft.phpwind.me/top/nginx-1.0.12.tar.gz
tar zxvf nginx-1.0.12.tar.gz
cd nginx-1.0.12
./configure --user=www --group=www --prefix=/alidata/server/nginx --with-http_stub_status_module --
with-http-cache --with-http_ssl_module --with-http_realip_module
make
make install
kill -USR2 `cat /alidata/server/nginx/logs/nginx.pid`
kill -QUIT `cat /alidata/server/nginx/logs/nginx.pid.oldbin`
```

次のコマンドを実行して nginx.conf ファイルを開きます。

```
vi /alidata/server/nginx/conf/nginx.conf
```

次のコンテンツを見つけ、その後に必要な情報を追加します。

```
fastcgi connect_timeout 300;
fastcgi send_timeout 300;
fastcgi read_timeout 300;
fastcgi buffer_size 64k;
fastcgi buffers 4 64k;
fastcgi busy_buffers_size 128k;
fastcgi temp_file_write_size 128k;
```

追加される情報：

```
set_real_ip_from IP_address
real_ip_header X-Forwarded-For;
```

set_real_ip_from IP : IP アドレスは Server Load Balancer インスタンスの IP アドレスではありません。IP アドレスを見つけるために Nginx のログをチェックしてください。通常、2 つの IP アドレスが入力されます。

次のコマンドを実行して、Nginx サーバーを起動します。

```
/alidata/server/nginx/sbin/ nginx -s reload
```

異なるセッション維持ルールの設定

Server Load Balancer はセッション永続性機能を提供します。セッション永続性を有効にすると、Server Load Balancer はセッション期間中に同じクライアントからのリクエストを同じバックエンドサーバーに配布できます。

レイヤー 4 リスナーの場合、セッション持続性は IP アドレスに基づいています。Server Load Balancer のリスナーは、同じ IP アドレスのリクエストを同じバックエンドサーバーに転送します。

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。**Rewrite Cookie** メソッドを選択すると、**cookie Name** を name に設定し、vip.a.com のクッキーのキーを name としてバックエンドサーバーに設定することができます。



高度な設定

クライアントの送信元IPの取得: 有効化(デフォルト)

セッションの保持: ☒ 有効化
HTTP プロトコルのセッションはクッキーに基づいて維持されます

クッキーの処理モード: クッキーの再書 ▼

クッキー名:

\$ を先頭の文字として使用することはできず、';', '\', '' (空白) の文字は使用できません

このセクションの手順に従って、バックエンドサーバーに Cookie を設定します。

Apache

httpd.conf ファイルを開き、次の行にコメントがないことを確認してください。

```
LoadModule usertrack_module modules/mod_usertrack.so
```

VirtualHost ファイルに以下の設定を追加してください。

```
CookieName name
CookieExpires "1 days"
CookieStyle Cookie
CookieTracking on
```

Nginx

構成ファイルを次のように構成します。

```
server {
    listen 8080;
    server_name wqwq.example.com;
    location / {
        add_header Set-Cookie name=xxxx;
        root html;
        index index.html index.htm;
    }
}
```

Lighttpd

構成ファイルを次のように構成します。

```
server.modules = ( "mod_setenv" )
$HTTP["host"] == "test.example.com" {
    server.document-root = "/var/www/html/"
    setenv.add-response-header = ( "Set-Cookie" => "name=XXXXXX" )
}
```

同じドメイン要求を別のサーバーに転送する方法

この例では、ドメイン名とURLで指定された転送ルールをどのように設定して、次の表に示すようにトラフィック転送を実行するかをデモンストレーションするために、Nginxサーバーとともに展開された4つのECSを例として使用します。

フロントエンドリクエスト	トラフィックを転送する
www.aaa.com/tom	サーバー-SLB_tom1とサーバー-SBL_tom2
www.aaa.com/jerry	サーバー-SLB_jerry1とサーバー-SBL_jerry2

ID と名前	ゾーン	IP(すべて) ▼	ステータス	ネットワーク(すべて) ▼
  (なし)	ap-northeast-1a(マスター)	 (パブリックネットワーク)	● 実行中	クラシックネットワーク
  (なし)	ap-northeast-1a(マスター)	 (パブリックネットワーク)	● 実行中	クラシックネットワーク

手順

インターネットに面したSLBインスタンスを作成します。

詳細は、サーバーロードバランサの作成を参照してください。

DNSを使用して、ドメイン名をSLBインスタンスのパブリックIPに解決します。

便宜上、この場合、SLBインスタンスのパブリックIPはホストファイルのドメイン名
www.aaa.comにバインドされています。

2つのVServerグループを作成します。

Server Load Balancerコンソールで新しく作成したインスタンスを探し、インスタンスIDをクリックして、「インスタンスの詳細」ページに移動します。

左側のナビゲーションペインで、**Server > VServer Group** をクリックします。

VServerグループの作成をクリックします。

表示されるダイアログボックスで、追加するバックエンドサーバーを選択し、それぞれのポートと重みを設定します。VServerグループ内のECSのポートは異なる場合があります。

この場合、サーバーグループ名として **TOM** を入力し、サーバーSLB_tom1とサーバーSBL_tom2をグループに追加し、ポート番号を80に設定して、デフォルトの重み値(100)を維持します。

注意: ネットワークタイプ:クラシックネットワーク, インスタンスタイプ:パブリックネットワーク, VServer GroupにはクラシックECSもしくはVPC ECSの追加が可能です。

*グループ名: TOM

*サーバーネットワークタイプ ☐ クラシックネットワーク ☒ 仮想ネットワーク

インスタンス: インスタンス名を入力してください 検索

すべて

ID と名前	IP	ゾーンとネットワーク
i- kangwei-testforR...	(パブリック) (プライベート)	ap-northeast-1a 仮想ネットワーク
i- サポート用共通	(パブリック) (プライベート)	ap-northeast-1a 仮想ネットワーク

選択済み 追加 (1/20)

ID と名前	IP	ゾーンとネットワーク	*ポート	*重み
i- kangwei-testforR...	(パブリック) (プライベート)	ap-northeast-1a 仮想ネットワーク	80	100

上記の手順を繰り返して、JERRYという別のVServerグループを追加します。このグループには、サーバーSLB_jerry1とサーバーSBL_jerry2が含まれています。

リスナーを追加します。

左側のナビゲーションペインで、**リスナー**をクリックし、**リスナーを追加**をクリックします。

リスナーを構成します。この場合、リスナーは次のように構成されます。

- i. フロントエンドプロトコル[Port]：HTTP：80
- ii. バックエンドプロトコル[Port]：HTTP：80
- iii. スケジューリングアルゴリズム：ラウンドロビン。
- iv. 他の設定項目のデフォルト値を保持する。

リスナーページで、**その他> フォワーディングルールの追加**をクリックします。



転送ルールページで、**転送ルールの追加**をクリックします。

3つの転送ルールを設定します。

転送ルールの作成 ×

ルール名	ドメイン	URL	VServer Group	操作
rule1	www.aaa.com	/tom	aaa ▼	削除
rule2	www.aaa.com		aaa ▼	削除

さらに追加

* ドメイン名：
アルファベット、数字、'-','.',のみ使用できます。下記いずれかの形式のドメインのみサポートしています。
- FQDN : www.test.com;
- ワイルドカードドメイン: *.test.com

* URL :
長さは 1 ～ 80 文字で指定できます。字母、数字、'-','/','.',','%','?','&','#'を使用できます。
URLは'/'で開始する必要がありますが、'/'のみの入力はできません。

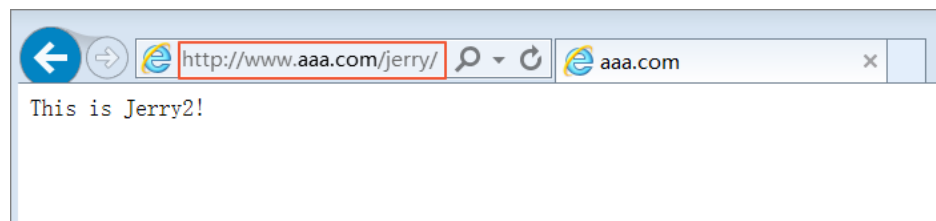
* ドメイン名とURLを少なくとも1つ入力してください。

確認

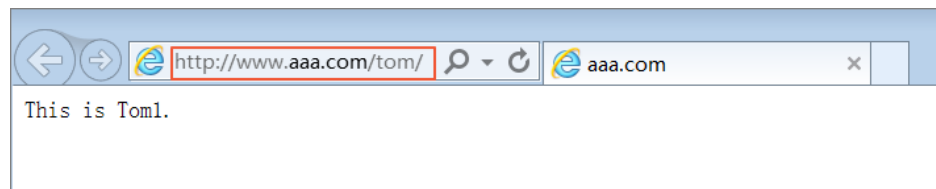
キャンセル

テスト：

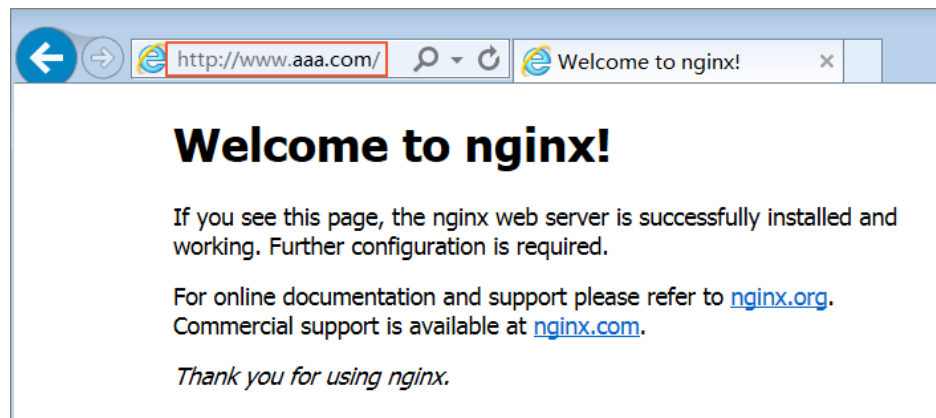
ブラウザにwww.aaa.com/jerryと入力すると、次の結果が返されます。



ブラウザにwww.aaa.com/tomと入力すると、次の結果が返されます。



ブラウザにwww.aaa.comと入力すると、次の結果が返されます。



バックエンド ECS の削除

バックエンド ECS インスタンスを Server Load Balancer インスタンスから直接削除すると、サービスが中断する可能性があります。ECS インスタンスの重みを最初にゼロに設定し、トラフィックが配信されていないときに ECS インスタンスを削除することをお勧めします。

Server Load Balancer コンソールにログインします。

選択リージョンと、ターゲットの 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** をクリックします。

ECS インスタンスがサーバーグループに追加されている場合は、**仮想サーバーグループ** または **マスタースレーブサーバーグループ** をクリックします。

ターゲット ECS インスタンスの重みの値を **0** に設定します。



トラフィックが ECS インスタンスに配信されていないことを確認し、**削除** をクリックしてバックエンドサーバープールから削除します。

トラブルシューティング

バックエンドサーバプールから ECS インスタンスを削除した後、ECS インスタンスに送信されている進行中のビジネスリクエストがある場合は、次の点を確認してください。

ECS インスタンスを他の Server Load Balancer のバックエンドサーバプールにあるかどうかを確認します。

ECS インスタンス ID を使用して、ECS インスタンスが追加される Server Load Balancer をフィルタリングできます。



ECS インスタンスにログインし、netstatコマンドを実行して、ECS インスタンスがサービスを展開したかどうかを確認します。

Windows: netstat -anoを実行して、インスタンス上の開いているすべてのポートを表示します。

Linux: このコマンドを実行すると、インスタンス上の開いているポートをすべて表示するか、netstatコマンドの他のパラメータを使用します。

Windows : netstat -ano を実行して、インスタンス上の開いているすべてのポートを表示します。

Linux : このコマンドを実行すると、インスタンス上の開いているポートをすべて表示するか、netstat コマンドの他のパラメータを使用します。

```

~# netstat -ano
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       Timer
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 0.0.0.0:111            0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 172.16.1.1:42285       172.16.1.1:80          ESTABLISHED off (0.00/0/0)
tcp        0      0 172.16.1.1:22          172.16.1.1:44832       ESTABLISHED on (0.16/0/0)
tcp6       0      0 :::111                 :::*                     LISTEN      off (0.00/0/0)
udp        0      0 0.0.0.0:42947          0.0.0.0:*               off (0.00/0/0)
udp        0      0 0.0.0.0:68             0.0.0.0:*               off (0.00/0/0)
udp        0      0 0.0.0.0:111            0.0.0.0:*               off (0.00/0/0)
udp        0      0 0.0.0.0:627            0.0.0.0:*               off (0.00/0/0)
udp        0      0 172.16.1.1:123         0.0.0.0:*               off (0.00/0/0)
udp        0      0 127.0.0.1:123          0.0.0.0:*               off (0.00/0/0)
udp        0      0 0.0.0.0:123            0.0.0.0:*               off (0.00/0/0)
udp6       0      0 :::111                 :::*                     off (0.00/0/0)
udp6       0      0 :::627                 :::*                     off (0.00/0/0)
udp6       0      0 :::123                 :::*                     off (0.00/0/0)
udp6       0      0 :::1275                :::*                     off (0.00/0/0)
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags     Type       State      I-Node    Path
unix    2      [ ]       DGRAM      off        7689      /run/systemd/shutdown
unix    7      [ ]       DGRAM      off        7691      /run/systemd/journal/dev-log
unix    2      [ ]       DGRAM      off        7695      /run/udev/control

```

API を使用して Server Load Balancer を設定する

このチュートリアルでは、リクエストパラメータはリクエストURLに含まれており、URL には共通パラメータは含まれていません。詳細については、API の概要を参照してください。

注：この例のリクエスト URL のパラメータ値は、読みやすくするために URL エンコードされていません。

前提条件

2 つの ECS インスタンスを作成し、SSH および Web ポートへのアクセスを許可してあること。

手順

CreateLoadBalancer インターフェイスを呼び出して、Server Load Balancer インスタンスを作成します。

リクエスト：

<https://slb.aliyuncs.com/?Action=CreateLoadBalancer&RegionId=cn-hangzhou-dg-a01>

レスポンス：

```
{
  "RequestId": "3DE96B24-E2AB-4DFA-9910-1AADD60E13A5",
  "LoadBalancerId": "LoadBalancerId",
  "Address": "SLBIPAddress"
}
```

CreateLoadBalancerHttpListener インターフェイスを呼び出して、Server Load Balancer インスタンスの HTTP リスナー（ポートは 80）を作成します。

リクエスト：

<https://slb.aliyuncs.com/?Action=CreateLoadBalancerHttpListener&LoadBalancerId=LoadBalancerId&ListenerPort=80&BackendServerPort=80&ListenerStatus=active>

SetLoadBalancerStatus インターフェイスを呼び出して、Server Load Balancer インスタンスをアクティブにします。

リクエスト：

<https://slb.aliyuncs.com/?Action=SetLoadBalancerStatus&LoadBalancerId=LoadBalancerId&LoadBalancerStatus=active>

AddBackendServers インターフェイスを呼び出して、バックエンドサーバーに ECS インスタンスを追加します。

リクエスト：

[https://slb.aliyuncs.com/?Action=AddBackendServers&LoadBalancerId=LoadBalancerId&BackendServers=\[{"ServerId":"ECS1InstanceID"}\]](https://slb.aliyuncs.com/?Action=AddBackendServers&LoadBalancerId=LoadBalancerId&BackendServers=[{)

レスポンス：

```
{
  "RequestId": "FA2F2172-63F2-409D-927C-86BD1D536F13",
  "LoadBalancerId": "LoadBalancerId",
  "BackendServers": [
    "BackendServer": [
      {
        "ServerId": "ECS1InstanceID",
        "Weight": 100
      }
    ]
  ]
}
```

AddBackendServers インターフェイスを再度呼び出して、バックエンドサーバーに ECS イン

スタンスを追加します。

リクエスト：

```
https://slb.aliyuncs.com/?Action=AddBackendServers&LoadBalancerId=LoadBalancerId
&BackendServers=[{"ServerId":"ECS2InstanceID"}]
```

レスポンス：

```
{
  "RequestId": "C61FAD0A-2E87-4D0C-80B0-95AB758FCA70",
  "LoadBalancerId": "LoadBalancerId",
  "BackendServers": {
    "BackendServer": [
      {
        "ServerId": "ECS1InstanceID",
        "Weight": 100
      },
      {
        "ServerId": "ECS2InstanceID",
        "Weight": 100
      }
    ]
  }
}
```

DescribeLoadBalancerAttribute インターフェイスを呼び出して、Server Load Balancer インスタンスの構成を表示します。

リクエスト：

```
https://slb.aliyuncs.com/?Action=DescribeLoadBalancerAttribute&LoadBalancerId=LoadBalancerId
```

レスポンス：

```
{
  "RequestId": "4747E9AE-ADFD-412D-B523-C1CBD45A2154",
  "LoadBalancerId": "LoadBalancerId",
  "Address": "SLBIPAddress",
  "IsPublicAddress": "true",
  "ListenerPorts": {
    "ListenerPort": [
      80
    ]
  },
  "BackendServers": {
    "BackendServer": [
      {
        "ServerId": "ECS1InstanceID",

```

```
"Weight" : 100
},
{
  "ServerId" : "ECS2Instanceid",
  "Weight" : 100
}
]
}
```

ブラウザを使用して Server Load Balancer インスタンスの IP アドレスにアクセスし、サービスが動作しているかどうかを確認します。