# 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のインスタンスとは何ですか？

最大接続数、CPS、QPS などのパフォーマンスメトリックは、パフォーマンス専有型インスタンスの SLA に含まれます。それに対して、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ていません。Server Load Balancer リソースは、パフォーマンス共有型インスタンスの間で共有されます。

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。

**最大接続数**

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、新しい接続ができなくなります。

**毎秒の接続数（CPS）**

1 秒間に新しい接続が確立される数。CPS が仕様の限界に達すると、新しい接続ができなくなります。

**毎秒のクエリ（QPS）**

1 秒あたりに処理できる HTTP/HTTPS クエリまたはリクエストの数。これはレイヤー 7 のリスナーに依存します。QPS が仕様の限界に達すると、新しい接続ができなくなります。

Server Load Balancer は、次の仕様のパフォーマンス専有型インスタンスを用意しています。

| 仕様 | | 最大接続数 | 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インスタンスを公開する前に、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共有型インスタンスになります。パフォーマンス専有型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。


# パフォーマンス専有型インスタンスのリリース計画

2017 年 5 月中旬から Alibaba Cloud は、米国東部1（バージニア）、中国南部1（深セン）および中国東部2（上海）で、パフォーマンス共有型インスタンスをパフォーマンス専有型インスタンスにアップグレードしました。その他のリージョンのリリース計画は次のとおりです。

- 中国北部2（北京）と中国東部1（杭州）：8月中旬
- 中国北部1（青島）：8月下旬
- 中国北部3（張家口）とその他のリージョン：8月下旬

# パフォーマンス専有型インスタンスの仕様を選択する方法は？

サービスの種類に応じて仕様を選択する必要があります。全体的な原則は次のとおりです：

レイヤー 4 リスナーの主な要素は、TCP keep-alive 接続の同時接続数です。最大接続数はキーメトリックとみなされます。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、適切な仕様を選択する必要があります。

レイヤー 4 リスナーの主な要素は、QPS の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。仕様の最初の選択後、圧力テストと実際の運用によって仕様を調整できます。

パフォーマンス専有型インスタンスを導入後、監視メトリックの結果により、実際のビジネス動向、ピーク帯域幅などを把握できます。詳細は、モニタリングデータを参照してください。

請求方法：

合計料金 = インスタンス料金 + パブリックトラフィック料金

パブリックトラフィックはアウトバウンドトラフィック（ダウンストリーム）で、インバウンドトラフィック（アップストリーム）は料金かかりません。

トラフィック課金は 1 時間単位で、リアルタイムで計算されます。1 時間未満の場合でも 1 時間として課金されます。

お支払いは毎月末に行われます。請求書が生成された後、料金はデフォルトのお支払い方法に請求されます。

## パフォーマンス専有型インスタンスが、仕様で定義されているパフォーマンスに達成できないことがありますか？

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仕様で定義されているパフォーマンスに達成することを保証していません。一つのメトリックが制限値に到達した時点で、制限が開始します。

例えば、Higher I（slb.s3.small）のパフォーマンス専有型インスタンスの場合、インスタンスの QPS が 20,000 に到達した場合、最大接続数が 200,000 に到達しなくても、QPS の制限が開始したため、新しい接続は出来なくなります。

## 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、パフォーマンス共有型インスタンスより低下することがありますか？

パフォーマンス共有型インスタンスはすべてのリソースを共有します。トラフィックが少ない場合、パフォーマンスはパフォーマンス専有型インスタンスより優れている可能性があります。しかし、トラフィックが多い場合、パフォーマンス専有型インスタンスのパフォーマンスは保証しますが、パフォーマンス共有型インスタンスは保証しません。

## パフォーマンス専有型インスタンスの作成に API を使用できますか？

現在、パフォーマンス専有型インスタンスの作成には、Server Load Balancer API がサポートされていませ

ん。

## パフォーマンス共有型インスタンスをまだ購入できますか？

はい。ただし、パフォーマンス共有型インスタンスは将来的に使用できなくなります。事前に通知しますので、登録されたメールアカウントまたは、ホームページの通知を確認してください。

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要求を同じバックエンドサーバーに配布できます。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** メソッドを選択した場合は、バックエンドサーバーで Cookie を設定する必要があります。

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、次の行をコメントにします。

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

仮想ホストで設定します。

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

# Nginx

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

# Lighttpd

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

このチュートリアルでは、Nginxを使用して静的Webページとともにデプロイされ たECSインスタンスを例として使用し、SSHおよびWebポートを介したアクセスを許可するセキュリティルールをECSインスタンスに追加します。

# タスク1 ECSインスタンスのクローン作成

システムディスクのスナップショットを作成します。

インスタンスのシステムディスクIDを照会します。

リクエスト：

https://ecs.aliyuncs.com/?Action=DescribeInstanceDisks&InstanceId=id5ab1760-3498-4d95-9687-a91545ef90b3

レスポンス：

```
{
"RequestId" : "9F2188AC-AFAC-4F43-B452-C88463B9F069",
"Disks" : {
"Disk" : [
{
"DiskId" : "1008-27930",
"Size" : 20,
"Type" : "system"}]
}
}
```

システムディスクのスナップショットを作成します。

リクエスト：

https://ecs.aliyuncs.com/?Action=CreateSnapshot&InstanceId=id5ab1760-3498-4d95-9687-a91545ef90b3&DiskId=1008-27930&SnapshotName=mytesthost1-init

レスポンス：

```
{
"RequestId"： "5CA4F9E6-81D2-42E1-A317-4C25284C6939"、
"SnapshotId"： "1008-27930-1097358"
}
```

スナップショット作成プロセスを照会します。進行状況が100の場合、スナップショットが作成されたことを示します。

リクエスト：

https://ecs.aliyuncs.com/?Action=DescribeSnapshotAttribute&RegionId=cn-hangzhou-dg-a01&SnapshotId=1008-27930-1097358

リスポンス：

```
{
"RequestId"： "8307863A-1415-40EF-9520-8974871E651C"、
"SnapshotId"： "1008-27930-1097358"、
"SnapshotName"： "mytesthost1-snp-init"、
"進捗状況"： "100"、
"CreationTime"： "2013-05-19T03：19Z"
}
```

新しく作成したスナップショットでカスタムイメージを作成します。

リクエスト：

https://ecs.aliyuncs.com/?Action=CreateImage&RegionId=cn-hangzhou-dg-a01&SnapshotId=1008-27930-1097358&Description=for creating test instances

レスポンス：

```
{
"RequestId"： "38C930E9-5CE9-4E24-A392-8538FC20D503"、
"ImageId"： "m8a1f80fe-ed9d-4156-a7a8-432f66305c36"
}
```

ECSインスタンスをクローンします。

カスタムイメージでは、同じ構成のECSインスタンスを複製することができ、2番目のECSインスタンスはこのImageIDで作成されます： ImageId = m8a1f80fe- ed9d-4156-a7a8-432f66305c36。

この例では、ECSインスタンスの構成は次のとおりです。

```
{
```

```
"RequestId" : "850ED7ED-A4D5-40A1-A7EF-C33B74B1296B",
"InstanceId" : "i6b47cd72-843f-4558-b911-2776acae06fb",
"ImageId" : "m8a1f80fe-ed9d-4156-a7a8-432f66305c36",
"RegionId" : "cn-hangzhou-dg-a01",
"ZoneId" : "cn-hangzhou-gy002-a",
"InstanceType" : "ecs.t1.small",
"HostName" : "mytesthost2",
"Status" : "Stopped",
"SecurityGroupIds" : {
"SecurityGroupId" : [
"g1f91e6e8-3c4b-4923-98dd-78aacbd09d17"
]
},
"PublicIpAddress" : {
"IpAddress" : [
"10.10.10.173"
]
},
"InnerIpAddress" : {
"IpAddress" : [
"10.32.148.152"
]
},
"InternetMaxBandwidthIn" : 2,
"InternetMaxBandwidthOut" : 2,
"SerialNumber" : "1fec6c01-7186-2c3e-fa10-a672b8c300ec"
}
```


この新しいECSインスタンスを区別するために、ファイル / usr / share / nginx / www / default / index.htmlの本文のサンプル文を変更してください。次に例を示します。 Welcome to nginx on mytesthost2！。

# タスク2 Server Load Balancerインスタンスを作成する

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" : "13ebb82ceaa-cn-hangzhou-dg-a01",
"Address" : "10.10.10.77"
}
```


ID 13ebb82ceaa-cn-hangzhou-dg-a01のサーバー負荷分散装置が作成されます。次のように、

同じ方法を使用してレイヤ4インスタンスを作成できます。

https://slb.aliyuncs.com/?Action=CreateLoadBalancerHttpListener&LoadBalancerId=13ebb82ceaa-cn-hangzhou-dg-a01&ListenerPort=80&BackendServerPort=80&ListenerStatus=active

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=13ebb82ceaa-cn-hangzhou-dg-a01&LoadBalancerStatus=active

# タスク3バックエンドサーバーを追加する

AddBackendServersインターフェースを通してバックエンドサーバーを追加してください。

リクエスト：

https://slb.aliyuncs.com/?Action=AddBackendServers&LoadBalancerId=13ebb82ceaa-cn-hangzhou-dg-a01&BackendServers=[{"ServerId":"i6b47cd72-843f-4558-b911-2776acae06fb"}]

レスポンス：

```
{
"RequestId" : "FA2F2172-63F2-409D-927C-86BD1D536F13",
"LoadBalancerId" : "13ebb82ceaa-cn-hangzhou-dg-a01",
"BackendServers" : {
"BackendServer" : [
{
"ServerId" : "id5ab1760-3498-4d95-9687-a91545ef90b3",
"Weight" : 100
}
]
}
}
```

別のバックエンドサーバーを追加します。

リクエスト：

https://slb.aliyuncs.com/?Action=AddBackendServers&LoadBalancerId=13ebb82ceaa-cn-hangzhou-dg-a01&BackendServers=[{"ServerId":"id5ab1760-3498-4d95-9687-a91545ef90b3"}]

レスポンス：

```
{
"RequestId" : "FA2F2172-63F2-409D-927C-86BD1D536F13",
"LoadBalancerId" : "13ebb82ceaa-cn-hangzhou-dg-a01",
"BackendServers" : {
"BackendServer" : [
{
"ServerId" : "id5ab1760-3498-4d95-9687-a91545ef90b3",
"Weight" : 100
}
]
}
}
```

Server Load Balancerインスタンスの構成の詳細を表示します。

リクエスト：

https://slb.aliyuncs.com/ ? Action = DescribeLoadBalancerAttribute＆LoadBalancerId = 13ebb82ceaa-cn-hangzhou-dg-a01

レスポンス：

```
 {
"RequestId" : "4747E9AE-ADFD-412D-B523-C1CBD45A2154",
"LoadBalancerId" : "13ebb82ceaa-cn-hangzhou-dg-a01",
"Address" : "10.10.10.77",
"IsPublicAddress" : "true",
"ListenerPorts" : {
"ListenerPort" : [
80
]
},
"BackendServers" : {
"BackendServer" : [
{
"ServerId" : "id5ab1760-3498-4d95-9687-a91545ef90b3",
"Weight" : 100
},
{
"ServerId" : "i6b47cd72-843f-4558-b911-2776acae06fb",
"Weight" : 100
}
]
}
}
```

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。

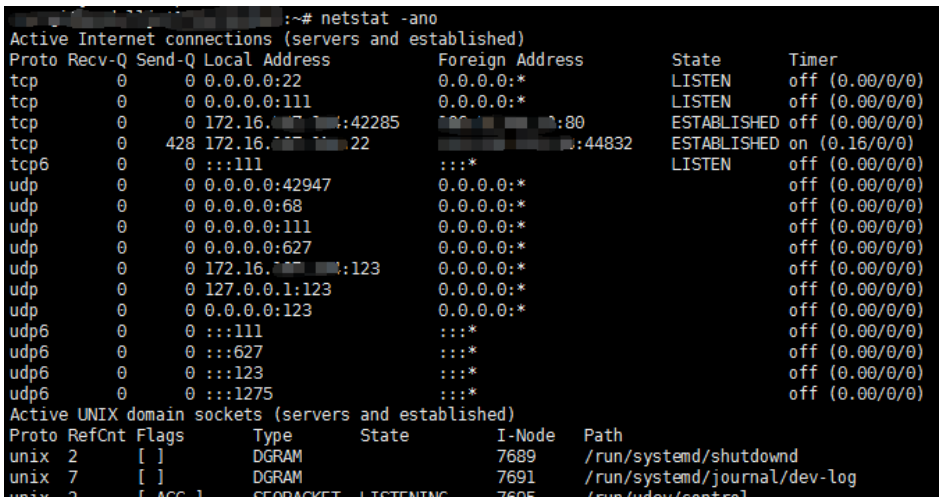
# トラブルシューティング

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ー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。




# 高可用性

## 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 プロトコルで同じです。


# Web アプリケーションの設定

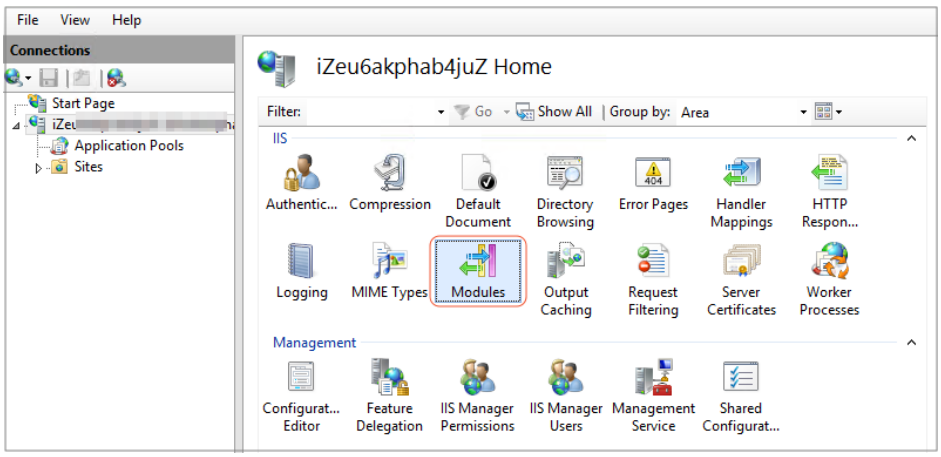
このセクションでは、Web アプリケーションの設定に使用される一般的な方法を紹介します。

## IIS7/IIS8 の構成

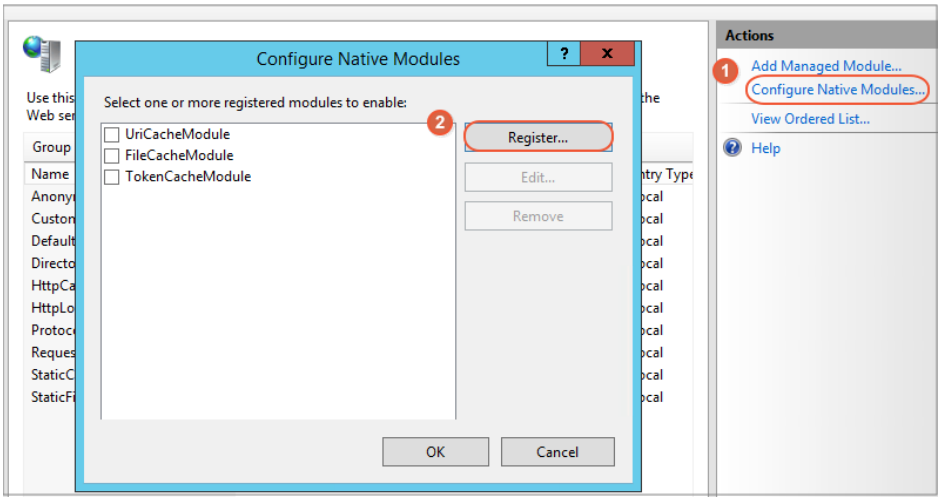
F5XForwardedFor をダウンロードして解凍します。

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。

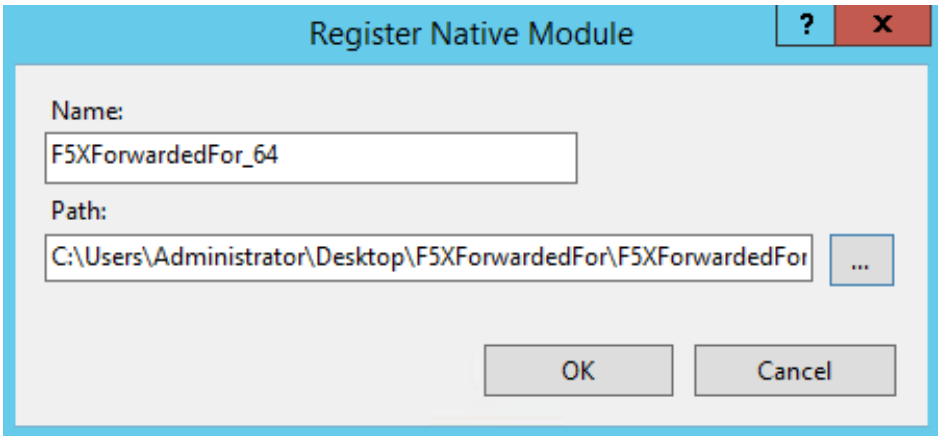
IIS マネージャーを開き、**モジュール** 機能をダブルクリックします。




**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。

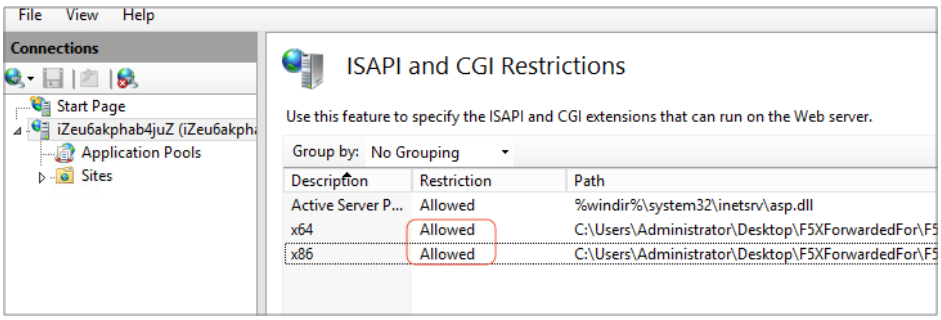



コピーした .dllファイルを追加します。




.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、それ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

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 |


# 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。


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。

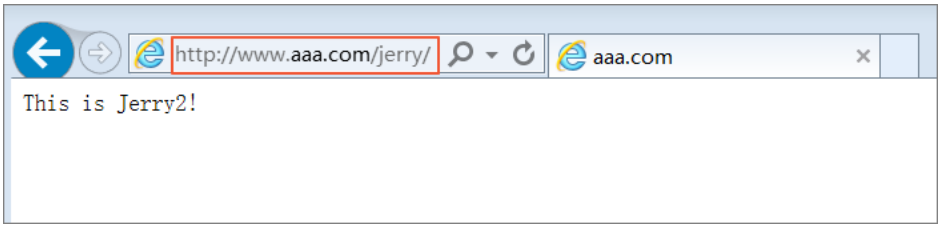
**リスナー**ページで、**その他> フォワーディングルールの追加**をクリックします。


**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。

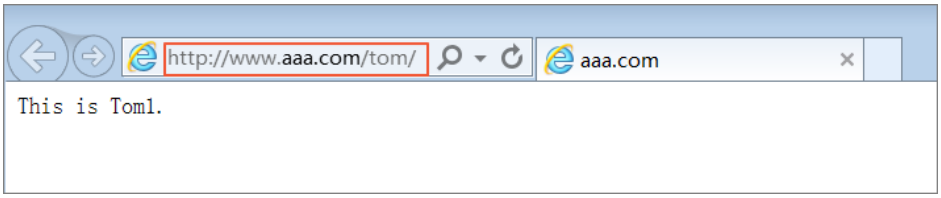
3つの転送ルールを設定します。


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。


http://www.aaa.com/
Welcome to nginx!
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.