

2022 年 10 月

お客様各位

SCSK 株式会社
産業事業グループ
メディア事業本部
IP ネットワークテクノロジー部

Ivanti Pulse Secure Virtual Traffic Manager User's Guide 22.1 - 22.2 **日本語参考訳のお取り扱いについて**

平素は格別のご高配を賜り、厚くお礼申し上げます。

本書のご利用に際し、以下の内容にご同意いただくことが前提になりますこと、予めご了承くださいますようお願い申し上げます。

敬具

記

1. ご利用条件

- (1) SCSK 株式会社（以下、SCSK）の商流にて Virtual Traffic Manager（以下 vTM）ライセンスをご購入いただいたお客様のみ本書をご利用いただけます。
- (2) 本書の内容は Ivanti, Inc. 社が作成した資料（原文）の日本語参考和訳で、齟齬がある場合は原文の内容が優先されます。
- (3) 本書に関するご質問には、SCSK でお答え出来ない場合がございます。
- (4) 本書によって貴社、或いはお客様に損害が生じた場合、SCSK は、損害の発生を予見し得たか否かに関わらず、如何なる損害につきましても一切責任を負わないものとさせていただきます。
- (5) 本書は予告なく内容を変更することがあります。変更により貴社に損害が生じたとしても SCSK は一切責任を負わないものとします。

2. 著作権

- (1) 本書の著作権は、SCSK に帰属します。貴社は vTM を使用する目的以外において SCSK の事前の承諾無く本書を転載、複製、変更、配布、販売、貸与等の利用をすることはできません。
- (2) 本書は、SCSK の事前の承諾無く第三者に提供することはできません。
- (3) 貴社にて本書を修正される場合は、SCSK に事前に通知をお願いいたします。また修正した内容の著作権も変わらず SCSK に帰属するものと致します。

以上



Pulse Secure Virtual Traffic Manager: User's Guide

22.1 - 22.2

著作権情報

本ドキュメントは、ガイドとしてのみ提供されます。いかなる保証も、提供または想定されません。本ドキュメントには、Ivanti, Inc.およびその関連会社(以下「Ivanti」と総称します)の秘密情報および/または専有情報が含まれており、Ivantiから事前に書面による同意を得ずに、開示または複製してはなりません。

Ivantiは、通知なしでいつでも本ドキュメントまたは関連製品の使用および記述を変更する権利を留保します。Ivantiは、本ドキュメントの使用に対していかなる保証も提供せず、本ドキュメントに含まれることがあるいかなる誤りに対しても責任を負わず、本ドキュメントに含まれる情報を更新することを誓約しません。最新の製品情報については、www.lvanti.comをご覧ください。

Copyright © 2022 Ivanti, Inc. 著作権所有。

特許による保護については、<https://www.ivanti.com/patents>を参照してください。

目次

前書き	10
表記上の慣例	10
技術サポートの依頼	12
Traffic Managerの概要	14
Traffic Managerの紹介	14
Traffic Manager製品のバリエーション	15
各章の概要	18
ネットワークレイアウト	23
ネットワーク構成の基本コンポーネント	23
専用管理ネットワーク	24
クラスタのサイジング	25
IPトランスペアレンシ	27
トラフィックIPアドレスとトラフィックIPグループ	29
IPv6について	41
初期設定	44
アーキテクチャコンセプト	44
最初のサービスの管理	46
クラスタの作成	48
仮想サーバ	53
はじめに	53
ルールの適用	55
SSL復号化	56
サービスプロテクションクラス	58
帯域制御クラス	58
サービスレベルモニタリングクラス	59
HTTPコンテンツキャッシング	59
Web Accelerator	60
HTTPコンテンツ圧縮	60
コネクション分析	61
リクエストロギング	65
プロトコル設定	66
プール	74
はじめに	74
負荷分散	75
セッション維持	78
帯域制御	78
ヘルスモニタリング	79
SSL暗号化	79
プロトコル設定	80
プールへのリクエストの制限	80
バックエンドのフォールトトレランス	84

ノードのドレイン、無効化、削除	87
オートスケーリング	89
DNSを用いたオートスケーリング	98
サービスディスカバリ	99
トラフィックIPグループとフォールトトレランス	105
フォールトトレランス	105
トラフィックIPアドレスの選択	106
トラフィックIPグループの作成	107
インターフェースとサブネットのマッピング(トラフィックIPネットワーク)	110
フォールトトレランスの設定	111
フェールオーバー	115
フォールトトレランスアクティビティのデバッグとモニタリング	117
BGP接続性の設定	118
OSPFv2の接続性の設定	120
Traffic Manager管理インターフェースの主要機能	123
ホームページ	123
Services (サービス) > Configuration Summary (設定概要)	124
カタログ	125
ライセンス管理	126
System (システム) > Global Settings (グローバル設定)	130
System (システム) > Backups (バックアップ)	132
アクティビティモニタリング	139
リモートサービスに対する分析のエクスポート	147
クラウドクレデンシャル	157
Pulse Secure Virtual Webアプリケーションファイアウォール	159
TrafficScriptルール	167
概要	167
Traffic Managerでのルールの使用	171
ルールの例	176
サービスに対する認証の提供	181
概要	181
リモートLDAP認証によるTrafficScriptルールの使用	181
SAMLサービスプロバイダとしてのTraffic Managerの設定	185
Java Extension	193
Javaについて	193
Java Extensionの呼び出し	193
Traffic ManagerでJavaを使用するための設定	194
プロトコルのサポート	197
概要	197
基本的なTCPプロトコル	197
HTTP	200
SSL	202
SMTP (Simple Mail Transport Protocol)	205

FTP	206
Real Time Streaming Protocol	210
Session Initiation Protocol	212
プロキシプロトコル	221
セッション維持	222
セッション維持とは	222
セッション維持の設定	223
UDPプロトコルでのセッション維持	234
例	235
SSL暗号化	236
SSLの概要	236
Traffic ManagerのSSL機能	238
SSL:復号化ウィザード	239
SSL証明書の設定	239
認証局の証明書とCRLファイルの管理	245
SSL復号化	246
SSL暗号化	269
SSL転送におけるIPアドレスの維持	270
SSL暗号処理専用デバイスの使用	271
ヘルスマニタリング	279
どのノードをモニタリングするか	279
パッシブヘルスマニタリング	280
ヘルスマニタの概要	282
モニタカタログ	283
ヘルスマニタの使用	288
外部プログラムモニタ	288
サービスプロテクション	291
リスクの分類	291
保護機能	292
サービスプロテクションの有効化	294
サービスプロテクションクラスの追加	294
仮想サーバへのサービスプロテクションクラスの適用	298
サービスプロテクション性能	299
帯域制御	300
帯域制御とは	300
帯域制御の設定	301
リクエストレートシェイピング	304
リクエストレートシェイピングとは	304
リクエストレートシェイピングクラス(レートクラス)の設定	305
レートクラスの使用	305
さらにきめ細かいレートシェイピング	307
リクエストレートシェイピングのグラフ化	308
サービスレベルモニタリング	310

サービスレベルモニタリングについて	310
サービスレベルモニタリングクラス(SLMクラス)の設定	310
仮想サーバへのSLMクラスの適用	312
TrafficScriptによるSLMクラスの適用	312
SLMクラスのTrafficScriptの例	312
SLMクラス適合レートのグラフ化	313
コンテンツキャッシング	315
はじめに	315
コンテンツキャッシングの設定	315
キャッシングポリシー	320
TrafficScriptを使用したコンテンツキャッシングのコントロール	322
キャッシュ内の古いコンテンツの削除	324
Pulse Secure Web Acceleratorの使用	326
はじめに	326
動作モード	326
サービス用のWeb Content Acceleratorの設定	327
Web Content Acceleratorによる変化の測定	334
カスタムアクセラレーションプロファイルについて	336
最適化手法について	344
Web Content Acceleratorのトラブルシューティング	352
イベント処理およびアラート	362
概要	362
イベントタイプ	363
アクション	365
イベントハンドラの設定	366
カスタムアクション	367
TrafficScriptまたはJava Extensionsからのイベント報告	370
システムレベルの設定	373
ネットワーク設定	373
時間と日付の設定	384
Traffic Managerへのリモートログイン	385
カスタムカーネルパラメータの入力	385
カーネルページテーブルの隔離	386
システムセキュリティ	388
ファイアウォールとOSの設定	388
ネットワーク設計	390
UNIXユーザパーミッション	391
ファイルシステムのセキュリティ	391
OSの設定	392
管理システムのセキュリティ	393
管理サーバの基本設定	393
ユーザ管理	400
ログインセキュリティとログイン行動	410

ログイン情報バナー	411
イベントおよび監査ログ	411
一般的なセキュリティに関する配慮事項	412
Traffic ManagerのコントロールAPI	414
Traffic ManagerのコントロールAPIの概要	414
例: 実行中の仮想サーバのリスト表示	415
コマンドラインインターフェース	418
はじめに	418
CLIへのアクセス	418
コマンド	420
CLIのスク립ティング	428
zconfを使用したきめ細かい設定のインポート/エクスポート	430
はじめに	430
zconfの使用	430
完全バックアップのエクスポート	431
設定のリスト表示	432
部分インポート	432
マルチサイトクラスタの管理	433
はじめに	433
アクティブ化と非アクティブ化	434
主なコンセプト	434
デプロイシナリオ	435
設定	438
Traffic ManagerのDNSサーバ	443
DNSの概要	443
サポートされるDNS機能	448
DNSサーバの設定	452
グローバル負荷分散	456
概要と前提条件	456
グローバルサーバ負荷分散について	456
デプロイ計画	458
Traffic Manager GLBデプロイのコンポーネント	463
GLBの設定	466
トラフィックの可視化	476
DigによるDNSのテスト	477
GeoIPデータベースの使用	478
Traffic ManagerのFIPS認証	482
FIPSについて	482
FIPSモード	482
FIPSモードのデプロイ	485
Kerberos制約付き委任のサポート	493
Kerberosプロトコル	493
Kerberosプロトコルの遷移と制約付き委任	493

Kerberosを使用する論理的根拠	494
Traffic ManagerでのKerberos設定	494
Pulse接続セキュアサービスの負荷分散	499
PCS負荷分散の設定の概要	499
PCS負荷分散の設定のトポロジ	500
Load-balance Pulse Connect Secure (Pulse Connect Secureの負荷分散)ウィザードの実行	501
Traffic Managerを使用した最適なゲートウェイの選択	509
トラブルシューティング	518
ツールおよび手法	518
診断とイベントログ	518
リクエストとレスポンスのモニタリング	519
コネクションの詳細なデバッグ	521
個々のノードのテスト	523
設定の確認	524
トラブルシューティングのヒント	524
起こりやすい問題	526
サポートを利用するには	529
用語集	531
ソフトウェア使用許諾契約書	539
Berkeley DBコード(バージョン1.85)の使用許諾条件	539
RSA PKCS11	540
OpenLDAPの使用許諾条件	540
PCRE2ライセンス	541
Yahoo!の使用許諾条件UIライブラリ	544
ssleay暗号化ライブラリの使用許諾条件	544
libxml2およびlibxsltの使用許諾条件	545
Java Servlet APIの使用許諾条件	547
expat XMLパーサの使用許諾条件	548
MooToolsの使用許諾条件	548
OpenLayersの使用許諾条件	549
rsyncの使用許諾条件	550
Antlrおよびlibantlrの使用許諾条件	552
es3-grammarの使用許諾条件	553
jsoncppの使用許諾条件	554
libjpegの使用許諾条件	555
libunwindの使用許諾条件	556
Perl JSONライブラリの使用許諾条件	556
OpenSSLの使用許諾条件	559
WebPの使用許諾条件	562
Flexの使用許諾条件	563
zlibの使用許諾条件	564
XML::Twigの使用許諾条件	565
MIT Kerberosの使用許諾条件	565

Libeditの使用許諾条件	594
ZebOSの使用許諾条件	595
Curlの使用許諾条件	596
Janssonの使用許諾条件	596
Digest::SHAの使用許諾条件	597
Sys::SysLogの使用許諾条件	597
libpcapの使用許諾条件	598
elfutilsの使用許諾条件	598
libpngの使用許諾条件	605
Perlの使用許諾条件	608
xmlsecの使用許諾条件	611
ltraceの使用許諾条件	614
YAML::XSの使用許諾条件	621
GeoNamesの使用許諾条件	623
LibYAMLの使用許諾条件	630
giflibの使用許諾条件	631
Gorilla Websocketの使用許諾条件	631
Goの使用許諾条件	632
オープンソースライセンスの注意点	634

前書き

表記上の慣例

この表記上の慣例では、書式設定とコマンド構文の慣例、Ivanti技術文書で使用される重要な注意事項の書式について説明します。

書式の慣例

太字、イタリック体、またはCourierフォントなどの書式が、文書中で特定の用語またはフレーズを強調するために使用されることがあります。

書式	説明
太字	コマンド名を示します。
	キーワードおよび演算子を示します。
	ユーザが操作するGUI要素の名前を示します。
	GUIで入力するテキストを示します。
イタリック体	強調するときに使用します。
	変数を示します。
	ドキュメントのタイトルを示します。
Courierフォント	コマンド出力を示します。
	コマンド構文の例を示します。

コマンド構文の慣例

太字とイタリック体は、コマンド構文の構成要素を示します。区切り文字と演算子は、パラメータのグループ化とそれぞれの論理的な関係を指定します。

慣例	説明
太字	コマンド名、キーワード、およびコマンドオプションを示します。

慣例	説明
イタリック体	変数を示します。
[]	角括弧内に表示される構文の構成要素は、オプションです。 システムプロンプトに対するデフォルトの操作は、角括弧で囲まれています。
{ x y z }	必須パラメータの選択肢は、縦棒で区切って、中括弧内に記載されます。 いずれかのオプションを選択する必要があります。
x y	縦棒は、相互に排他的な要素を区切ります。
< >	パスワードなどの出力されない文字は、山括弧内に記載されます。
...	member[member...]のように、前の要素を繰り返します。
\	コマンド例で、表示行末を示します。コマンド入力の2行がバックスラッシュで区切られている場合、プロンプトではバックスラッシュなしでコマンド全体を入力します。

注意および警告

本ドキュメントでは、注、注記、および注意という表記が使用されることがあります。



「注」は、ヒント、指針、または助言を示し、重要な情報を強調し、または関連情報の参照先を示します。

注記

「注記」という表記は、例えばトラフィックが中断され、またはデバイスが再起動される可能性がある場合について注意を促すために、重要性の高い注意事項を示しています。

注意

「注意」という表記では、ユーザにとって危険が生じる、またはハードウェア、ファームウェア、ソフトウェア、データに損害が生じる可能性がある状況について注意を促します。

技術サポートの依頼

技術的製品サポートに関しては、Ivantiグローバルサポートセンター(PSGSC)にご連絡ください。サポート契約を締結されている場合は、チケットをPSGSCに送付してください。

- 製品保証 - 製品保証情報については、<https://support.pulsesecure.net/product-service-policies/>にアクセスしてください。

セルフヘルプオンラインツールおよびリソース

問題をすばやく簡単に解決できるよう、Ivantiではカスタマーサポートセンター(CSC)と呼ばれるオンラインセルフサービスポータルを運営し、以下の機能をご提供しています。

- CSCの内容の表示: <https://support.pulsesecure.net>
- 既知のバグの検索: <https://support.pulsesecure.net>
- 製品ドキュメントの表示: <https://www.ivanti.com/support/product-documentation>
- ソフトウェアの最新バージョンのダウンロードと、リリースノートの確認:
<https://support.pulsesecure.net>
- CSCケース管理ツールによりオンラインで問題提起: <https://support.pulsesecure.net>
- 製品シリアルナンバーでサービスエンタイトルメントを確認するには、シリアルナンバーエンタイトルメント(SNE)ツールを使用: <https://support.pulsesecure.net>

製品に関する重要な通知、技術記事、および提案について:

- Pulse Secure Knowledge Centerで技術情報およびセキュリティ勧告について検索:
<https://kb.pulsesecure.net>
- Pulseコミュニティのオンラインフォーラムで質問をして解決策を見つける:
<https://community.pulsesecure.net>

PSGSCでの問題提起

PSGSCでの問題提起は、Webまたは電話で行えます。

- PSGSCの問題管理ツール(<https://support.pulsesecure.net>)を使用してください。
- 1-844 751 7629(米国の無料通話番号)まで電話でお問い合わせください。

無料通話番号のない国からの国際電話または直通ダイヤルについては、以下を参照してください:
<https://support.pulsesecure.net/support/support-contacts/>

Traffic Managerの概要

本章では、Pulse Secure Virtual Traffic Manager(Traffic Manager)を紹介し、利用可能な製品バリエーションを説明し、本マニュアルの内容の概要を示します。

Traffic Managerの紹介

Traffic Manager製品ファミリは、さまざまなソフトウェア、アプライアンス対応製品、仮想アプライアンス、およびクラウドコンピューティング製品で、高可用性と、アプリケーションベースのトラフィック管理および負荷分散を実現するソリューションを提供しています。これらのソリューションは、あらゆるアプリケーショントラフィックをコントロールするとともに、トラフィックにインテリジェンス、セキュリティ、そして高いフォールトトレランスをもたらします。

Traffic Managerは、HTTP (Web)やメディア送信などのTCP/UDPベースのサービス、WebサービスなどのXMLベースのサービスなど、ビジネスクリティカルなサービスをホストする企業を対象とした製品です。

Traffic Manager独自のプロセスアーキテクチャでは、大容量のネットワークトラフィックも効率よく処理することができます。また、このアーキテクチャが備える拡張性により、ニーズに応じてクラスタにフロントエンドTraffic Managerやバックエンドサーバを増設することが可能です。クラスタのサイズに制限はなく、Traffic Managerの性能は、使用プラットフォームの性能が高まるほど向上します。

Traffic Managerファミリは、新たなニーズにも適応して拡張することのできる非常に優れたソリューションです。独自のTrafficScript言語と組み込み式Java Extensionsを使用すれば、ニーズに応じた高度なトラフィック管理ルールを作成し、リクエストとレスポンスをチェック、変換、管理、ルーティングすることができます。

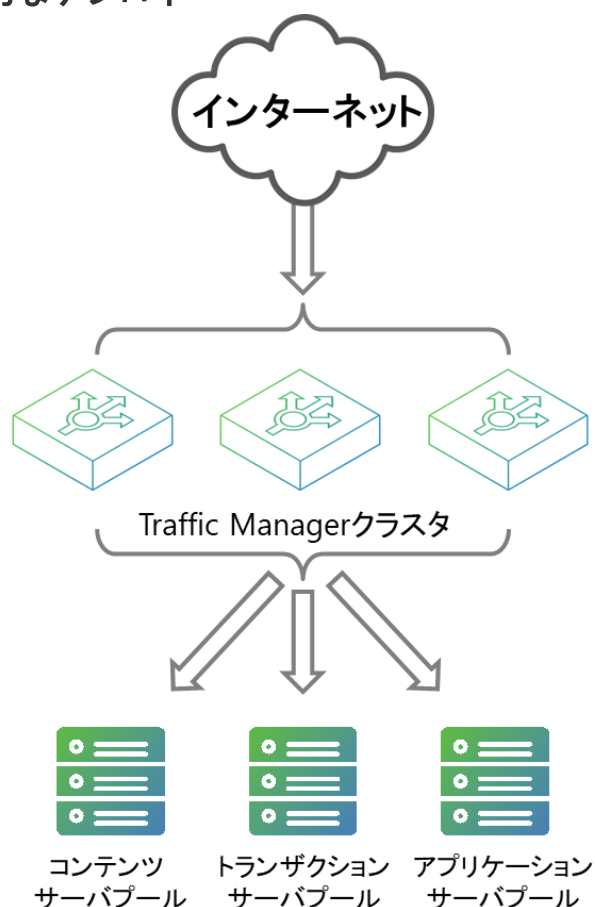
TrafficScriptルールでは、任意のTCPまたはUDPベースのプロトコルでのコネクションを管理することが可能です。

Traffic Manager製品は、導入直後から優れたセキュリティを発揮し、不正侵入やDoS攻撃を阻止します。また、非常に高速で強力な¹ (SSL)暗号化技術を採用しているため、多数のセキュアなコネクションを効率よく復号化および再暗号化することができます。エンドツーエンドのセキュリティを確保しながら、TrafficScriptルール、セキュリティポリシー、およびその他のコンテンツベースの計算を暗号化されたリクエストに適用することができます。

¹ Traffic Managerは、SSLとその後継アルゴリズムのトランスポート層セキュリティ(TLS)をサポートしています。特に指定のない限りは、本ガイドでSSLという用語は、両方のプロトコルを指します。

Traffic Managerは、クリティカルな高可用性ソリューションとして、クラスタ冗長機能を提供しています。この機能により、アクティブ/パッシブ(スタンバイ)構成のフロントエンドサーバを無制限にデプロイできるようになります。アクティブサーバに障害が発生した場合は、スタンバイサーバが自動的に起動します。さらにそのサーバにも障害が発生した場合は、別のスタンバイサーバが負荷を引き継ぐことができます。このため、システム内に単一障害点が発生することはありません。

一般的なデプロイ



Traffic Manager製品のバリエーション

Traffic Manager製品ファミリは、ソフトウェア、ハードウェアアプライアンス、仮想アプライアンス、クラウドインスタンスなど様々な構成で提供されています。

どの構成でもコアとなるTraffic Managerソフトウェアは同じですが、提供される機能レベルは、バージョンごとにホストプラットフォームまたはライセンスキーに応じて異なる場合があります。ライセンスキーが提供されない所では、Traffic Managerは開発者モード(以下のセクション参照)で動きます。

本マニュアルでは、全てのオプションを有効化したTraffic Managerソフトウェアの全機能について説明します。ご使用の製品バージョンによっては、ここに記載されていても、存在しない、または表示されない機能がある場合があります。存在していても、有効にならない機能は、管理UIに淡色表示され、選択できません。

適切なライセンスキーのアップグレードによってのみ利用可能になる、Traffic Managerのオプションコンポーネントがいくつか用意されています。

- Pulse Secure Virtual Webアプリケーションファイアウォール(vWAF): Webアプリケーションに対して、高度な攻撃の検出と保護を実現します。Traffic Managerのインフラストラクチャにおけるこのコンポーネントの役割については、「[Pulse Secure Virtual Webアプリケーションファイアウォール](#)」を参照してください。製品の詳細と詳しい手順については、『*Pulse Secure Virtual Web Application Firewall: User's Guide*』を参照してください。このドキュメントは、IvantiのWebサイト(www.ivanti.com)からダウンロードできます。
- Pulse Secure Web Accelerator (Web Content Accelerator): Webアプリケーションに対してコンテンツ最適化の機能を果たすコンポーネントです。Traffic Managerに完全統合されたコンポーネントとしても、スタンドアロンのプロキシモードでも使用でき、後者の場合はTraffic Managerの負荷分散機能が無効になります。どちらの形態が適しているかは、お近くの販売代理店でご相談ください。「[Pulse Secure Web Acceleratorの使用](#)」では、インフラストラクチャでWeb Content Acceleratorを有効化および設定する詳しい方法について説明しています。

利用可能な機能については、ご使用の製品バージョンの仕様に記載されています。該当する『*Pulse Secure Virtual Traffic Manager: Installation and Getting Started Guide*』も参照してください。このドキュメントは、IvantiのWebサイト(www.ivanti.com)からダウンロードできます。

Traffic Managerコミュニティエディション

ライセンス未適用の状態では、Traffic Managerはデフォルトでコミュニティエディションとして動作します。この状態では、Traffic Managerは通常どおり全ての機能が作動しますが、帯域幅が10Mb/秒に、クラスタサイズが4つまでに制限されます。コミュニティエディションは、本番環境に対応したTraffic Managerの無償版となっており、本番環境で高度なvADC (仮想アプリケーション配信コントローラ)機能を試してみたい、システム管理者やアプリケーション開発者にとって便利なものとなっています。

希望する用途に合った、適切なライセンスキーを購入する方法については、サポート窓口にお問い合わせください。

対応プラットフォーム

Traffic Managerソフトウェアは、様々なプラットフォーム、物理または仮想サーバ、およびクラウドインフラストラクチャにデプロイできます。最新のプラットフォームおよびバージョン番号の必要条件については、www.ivanti.comでリリースノートおよびドキュメントを参照してください。

ソフトウェア

Traffic Managerにはソフトウェアパッケージがあり、これは既存のサポートされるLinuxサーバとUNIXサーバでのデプロイに適しています。対応ディストリビューションのリストは、前述したリリースノートに掲載されています。

アプライアンス

Ivantiは、アプライアンスディスクイメージとして、Traffic Managerを提供しています。これは、承認されたサーバハードウェアプラットフォームでのデプロイに適しています。

仮想アプライアンス

VMware vSphere、Microsoft Hyper-V、およびQEMU/KVM向けに、Traffic Manager仮想アプライアンスパッケージが用意されています。

クラウドコンピューティングサービス

Traffic Managerは、AmazonのElastic Compute Cloud (EC2)、Rackspace、Microsoft Azure、およびGoogle Compute Engine (GCE)でサポートされています。また、EC2およびGCE上で実行されているサポート対象のLinuxおよびUNIX仮想マシンインスタンスでのTraffic Managerソフトウェア製品のインストールもサポートされています。

サポートされるクラスタの組み合わせ

前のセクションで述べたように、Traffic Managerは選択したプラットフォームに応じて各種の製品形態で利用できます。コアとなるTraffic Managerソフトウェアに加え、適切なライセンスをアップグレードすることによって様々な製品オプションと機能が追加されます。

複数のTraffic Managerをクラスタ化すると高可用性と冗長性を得ることができ、これによって同じクラスタ内の異なるプラットフォーム上でTraffic Managerの柔軟なアーキテクチャを利用してインスタンスをホストできるようになります。ただし、クラスタの各メンバが同じ製品設定で実行されているか確認が必要です。

クラスタを設計する際には、以下の点に留意してください。

- ソフトウェア製品に対して別々のサポートされているホストOSが混在していてもサポートされます。クラスタの動作に悪影響を及ぼすことはありません。
- クラスタ内でライセンスキーのタイプを混在することは技術的には可能ですが、お奨めしません。クラスタのサブセットでのみ許諾されている機能を使用する場合、管理UIで警告やエラーが表示される可能性があります。これは、設定の自動複製機能が、ライセンスのない機能をクラスタ全体に適用しようとするためです。
- クラウドインスタンスを、非クラウドインスタンス(ソフトウェア、仮想アプライアンス、ハードウェアアプライアンス)とともに直接クラスタ化することはできません。ただし、マルチサイトクラスタ管理機能を使用することにより¹、ロケーションに関係なくTraffic Managerの複数のクラスタを集中管理することができます。
- vWAFの機能は、プラットフォーム混在のクラスタ(ソフトウェアと仮想アプライアンスなど)では使用できません。
- クラスタで複数の製品バージョンを併用することはできませんが、クラスタ全体のアップグレード(またはダウングレード)を実行する場合は例外です。このような移行状態の間、クラスタで異なるバージョンが実行されているというメッセージが一部のTraffic Managerで表示されます。これは想定される現象であり、クラスタの全てのメンバがアップグレードされれば、このエラーは表示されなくなります。

推奨されるTraffic Managerインフラストラクチャの実行によって生じる影響について不明な点がある場合は、サポート窓口にお問い合わせで指示を受けるようにしてください。

各章の概要

- [ネットワーク構成の基本コンポーネント](#)

この章では、代表的なトラフィック管理されたサーバファームの運用に必要なネットワーク構成について説明します。ハードウェアの必要条件、ネットワークのレイアウト、IPアドレス、およびDNSエントリについて取り上げます。

- [初期設定](#)

この章では、仮想サーバ、プール、およびルールの仕組みなど、Traffic Managerアーキテクチャのコンセプトを説明します。また、最初のサービスの管理を開始する方法について説明します。

- [仮想サーバ](#)

仮想サーバは、Traffic Managerソフトウェアを構成する2つの基本コンセプトのうちの1つです。この章では、仮想サーバについて詳しく説明します。

¹ [「マルチサイトクラスタの管理」](#)を参照してください。

- [プール](#)

プールは、Traffic Managerソフトウェアを構成するもう1つの基本コンセプトです。この章では、プールについて詳しく説明します。

- [トラフィックIPグループとフォールトトレランス](#)

この章では、フォールトトレランス、すなわちバックエンドサーバまたは個々のTraffic Managerシステムの耐障害性について説明します。

- [Traffic Manager管理インターフェースの主要機能](#)

これまでに、基本的なTraffic Managerシステムの動作について説明してきました。この章では、Traffic Managerの性能モニタリングや診断機能など、Traffic Manager管理インターフェースで提供されるツールについて説明します。

- [TrafficScriptルール](#)

この章では、トラフィックを管理するためのルールの作成に使用するTrafficScript言語を紹介します。ルールの作成方法を説明し、例をいくつか掲載します。Traffic ManagerのTrafficScript機能については、別冊の『*Pulse Secure Virtual Traffic Manager: TrafficScript Guide*』で詳しく説明しています。

- [サービスに対する認証の提供](#)

この章では、リモート認証のサポートについて詳しく説明します。この章で説明しているメカニズムを使用すると、サービスにリモートユーザの検証機能を付加することができます。

- [Java Extensions](#)

この章では、Traffic ManagerのJava Extensionsについて説明します。Java Extensionsを使用すると、TrafficScriptからJavaコードを呼び出すことができます。Traffic ManagerのJava Extensions機能については、別冊の『*Pulse Secure Virtual Traffic Manager: Java Development Guide*』で詳しく説明しています。

- [プロトコルのサポート](#)

一部のプロトコルには、プロトコル固有のサポートによりメリットがもたらされます。この章では、Traffic Manager製品の特殊なプロトコルハンドラについて説明します。

- [セッション維持](#)

この章では、Traffic Managerのセッション維持機能について説明します。役立つセッション維持ストラテジを紹介し、本製品での設定方法について説明します。

- [SSL暗号化](#)

この章では、受信トラフィックのSSL復号化、バックエンドサーバに送信されるリクエストの暗号化について取り上げます。また、本製品でのSSL証明書、認証局、および失効リストの設定方法と、サービスへの適用方法について説明します。

- [ヘルスモニタリング](#)

この章では、Traffic Managerのヘルスモニタリング機能について説明します。ヘルスマニタでは、バックエンドサーバノードが正常に動作しているかどうかテストされます。障害が発生すると、Traffic Managerからアラートが報告され、障害が発生したノードを経由しないようトラフィックがルーティングされます。

- [サービスプロテクション](#)

この章では、本製品のサービスプロテクション機能について説明します。カタログにサービスプロテクション設定のクラスを定義する方法と、1つ以上のサービスでそのクラスを使用する方法を説明します。

- [帯域制御](#)

この章では、本製品の帯域制御機能について説明します。カタログに帯域制御設定のクラスを定義する方法と、1つ以上のサービスでそのクラスを使用する方法を説明します。

- [リクエストレートシェイピング](#)

この章では、Traffic Managerを使用した、サーバおよびアプリケーションに対するリクエストのレートシェイピング方法について説明します。レートシェイピングを行うと、ユーザのアクティビティを制限し、リクエストによりアプリケーションが過負荷状態になるのを防ぐことができます。

- [サービスレベルモニタリング](#)

この章では、本製品のサービスレベルモニタリングについて説明します。カタログにサービスレベルモニタリング設定のクラスを定義する方法と、1つ以上のサービスでそのクラスを使用する方法を説明します。

- [コンテンツキャッシング](#)

この章では、Traffic ManagerのHTTPコンテンツキャッシング機能について説明します。頻繁にリクエストされるWebリソースをキャッシングすることで、以降のリクエストに対して、バックエンドサーバノードに同じリクエストを何度も転送せずに、キャッシュから直接レスポンスすることができます。

- [Pulse Secure Web Acceleratorの使用](#)

この章では、Web Content Acceleratorと、WebサービスにWebコンテンツの最適化を適用するための方法について説明します。

- イベント処理およびアラート

この章では、特定のイベントが発生した場合のTraffic Managerの動作をコントロールする方法と、外部管理システムに対する通知または制御の方法を説明します。

- システムレベルの設定

この章では、ハードウェアアプライアンス、仮想アプライアンス、クラウドインスタンス、および一部のTraffic Managerソフトウェア製品に固有の補足的なシステムレベル設定について説明します。

- システムセキュリティ

この章では、Traffic Managerをセキュアに運用するためのヒントを紹介します。ファイアウォールの設定、ネットワーク設計、およびTraffic Managerソフトウェアに関連するUnixソフトウェアのパーミッションについて取り上げます。

- 管理システムのセキュリティ

この章では、外部データベースに対するユーザ認証など、管理インターフェースへのアクセスをコントロールするためのセキュリティ対策について説明します。

- Traffic ManagerのコントロールAPI

この章では、Traffic ManagerのコントロールAPIについて簡単に紹介します。詳細については、IvantiのWebサイトから入手できる別冊の『*Pulse Secure Virtual Traffic Manager: Control API Guide*』を参照してください。

- コマンドラインインターフェース

この章では、コマンドラインインターフェースの詳細と、『*Pulse Secure Virtual Traffic Manager: Control API Guide*』で定義されたコントロールAPIメソッドの参照先について取り上げます。

- zconfを使用したきめ細かい設定のインポート/エクスポート

この章では、zconfコマンドラインユーティリティについて説明します。このユーティリティを使用すると、個々の設定オブジェクトについてきめ細かいインポート/エクスポートを実行することができます。

- マルチサイトクラスタの管理

この章では、Traffic Managerのクラスタのコンセプトを応用し、複数のクラスタを、集中管理される1つのマルチサイトスーパークラスタにまとめるアイデアを紹介します。

- [Traffic ManagerのDNSサーバ](#)

Traffic Managerには、外部にソースを持たせるDNSの代替手段として、内部DNSサーバ機能が用意されています。この章では、Traffic ManagerのDNSサーバの実装および構成について説明します。

- [グローバル負荷分散](#)

この章では、Traffic Managerへのグローバルサーバ負荷分散(GSLB)の実装について説明します。

- [Traffic ManagerのFIPS認証](#)

この章では、FIPSがTraffic Managerにどのように実装されているかを理解しておきたい将来的なユーザのために、その点について詳細に説明します。

- [Kerberos制約付き委任のサポート](#)

この章では、Traffic ManagerへのKerberosプロトコルの実装について説明します。

- [トラブルシューティング](#)

この章では、負荷分散システムでの問題または異常動作を調査し、診断する方法について説明します。

- [用語集](#)

「用語集」では、本マニュアルで使用されている用語について説明します。用語によっては、コンピューティング関連の文献で様々な意味で使用されているものがありますので、必要に応じてこのセクションを参照してください。

ネットワークレイアウト

本章では、ネットワークの構成について説明します。サーバファームのトラフィックを効果的に管理するために必要なハードウェアと、DNSおよびIPアドレスレイアウトについて取り上げます。

ネットワーク構成の基本コンポーネント

トラフィック管理される基本的なサーバファームは、以下のコンポーネントで構成されます。

- Traffic Managerソフトウェアが稼働する1台以上のフロントエンドマシン
- 多数のバックエンドサーバ(Webサーバ、メールサーバなど)

フロントエンドマシンは、インターネット(またはリモートクライアントが配備された場所)からトラフィックを受信できなければなりません。また、バックエンドマシンとも通信できなければなりません。

通常、バックエンドサーバは、内部ネットワークからのみ認識できます。インターネットとバックエンドマシンとの間でやり取りされるトラフィックは、フロントエンドマシンがルーティングする必要はありません。

一般に、Traffic Managerソフトウェアは、マルチホームマシンにデプロイします。インターネット側から認識できるのは1枚のネットワークインターフェースカードのみです。また、バックエンドサーバがデプロイされている内部プライベートネットワークからは、1枚以上のネットワークインターフェースカードを認識できます。ネットワークカードが1枚のみ搭載されたマシンにTraffic Managerを設定することもできます(評価またはテスト環境では一般的な構成)。この場合、Traffic Managerはクライアントとサーバの両方と通信することができます。

完全なフォールトトレランスを備えた構成では、2台以上のフロントエンドマシンと複数台のバックエンドサーバを使用します。この構成では、いずれかのマシンで障害が発生しても、Traffic Managerのフェールオーバー機能によってリクエストが他のマシンにルーティングされるため、システム内に単一障害点が発生することがありません。



製品バージョンによっては、クラスタサイズが2台のTraffic Managerマシンに制限されます。このような製品バージョンでは、ソフトウェアキーをアップグレードすることにより、クラスタを拡張することができます。

利用できるハードウェアが限られている場合、少ない台数のサーバでネットワークを構成することもできます。最小構成として、トラフィック管理ソフトウェアとインターネットサービスを同一マシンにインストールすることも可能です。このような構成では、製品の有用性とハードウェア障害発生時のフォールトトレランス機能が損なわれるためお奨めしませんが、評価またはデモンストレーション用として役立つこともあります。

Traffic Managerは、スタンドアロン型のファイアウォールと組み合わせて使用することができます。このシナリオでは、Traffic Managerがインターネットと内部ネットワークの両方から認識できる必要があります。Traffic Managerを適切なDMZに配置することをお奨めします。



システムセキュリティでは、ネットワーク環境におけるセキュリティについて詳しく説明しています。商用サービスを設定する前に、その章をお読みいただくことをお奨めします。

専用管理ネットワーク

Traffic Managerを設定するときに、単一の管理IPアドレスを指定することができます。これにより、Traffic Managerは、そのIPアドレスでのみ管理トラフィックを受け入れるようになります。専用管理IPアドレスを設定するために必要な設定手順については、ご使用の製品タイプに適合する『Pulse Secure Virtual Traffic Manager: Installation and Getting Started Guide』を参照してください。

Traffic Managerに関するドキュメントは、IvantiのWebサイト(www.ivanti.com)を参照してください。

管理トラフィックには、WebベースのTraffic Manager管理UI、コントロールAPI、REST API、およびTraffic Managerクラスタ内で共有されるあらゆる設定や状態に対する全てのアクセスが含まれます。

管理IPアドレスを使用することで、ユーザは信頼性の高い専用の管理ネットワークを実現できます。一般に、Traffic Managerを運用する各ホストに、管理ネットワークに接続された専用のネットワークカードを搭載します。



ネットワーク環境におけるセキュリティの詳細については、**システムセキュリティ**を参照してください。商用サービスをセットアップする前にこの章をよくお読みいただくことをお奨めします。

完全に構成されたTraffic Managerで管理IPアドレスを変更するには、Traffic Manager管理UIにログインし、**System** (システム) > **Security** (セキュリティ) > **Management IP Address and Admin Server Port** (管理IPアドレスと管理サーバポート)をクリックします。必要な管理IPアドレスに対して「bindip」を設定し、**Update** (更新)をクリックして、変更を保存します。ソフトウェア製品の場合は、「configure」のスクリプトを再実行することもできます。この場合、ソフトウェアのリスタートが必要になることに注意してください。

各管理IPアドレスは、Traffic Managerクラスタ内の単一障害点になり得ます。管理ネットワークで障害が発生すると、マシン間の通信が全て失われ、管理UI、コントロールAPI、またはREST APIを使用したリモート設定が行えなくなります。

耐障害性を実現するために、各Traffic Managerが送信するフォールトトレランスメッセージは、全てのネットワークカードにブロードキャストされます。管理UIでは、このトラフィックを管理ネットワークに制限することができます。

クラスタのサイジング

バックエンドサーバ

Traffic Managerは、無制限の台数のバックエンドサーバをサポートできます。バックエンドサーバには、Webサーバ、メールサーバ、FTPサーバなどを含めることができます。これらのサーバでは、任意のプラットフォームで任意のソフトウェアを実行してサービスを提供でき、そのサービスはTraffic Managerによって管理されます。

バックエンドサーバは、それぞれが処理するコンテンツのタイプ別にプールと呼ばれるグループに整理できます。例えば、IIS™が稼働するWindowsマシンやZeus Web ServerまたはApache™が稼働するUnixマシンなどのWebサーバを複数のグループに分け、グループごとに異なるタイプの静的および動的Webコンテンツを処理することができます。Traffic Managerのプールおよびルールの仕組みを利用すれば、リクエストを分類して複数のサーバグループに振り分けることができます。プールとルールについては、[初期設定](#)で説明します。

特定機能を実行するために専用使用するサーバの台数は、状況によって異なります。例えば、静的Webページを処理する場合、リクエストの負荷が小さければ、2台のLinuxマシンで十分なこともあります。一方、より複雑なSun JSP™やMicrosoft ASPページを処理する場合は、より多くのマシンが必要になります。

Traffic Managerでは、任意の台数のマシンを使用できます。また、必要条件が変化したときは、サービスを中断することなく、バックエンドサーバを動的に追加または削除することができます。

バックエンドサーバでのコンテンツ管理

多数のバックエンドサーバで同一サイトの処理を行う場合、サイトのコンテンツが変更されたときに、それを各バックエンドサーバに伝搬する必要があります。これは、以下のいずれかの方法で行うことができます。

- 各バックエンドサーバにコンテンツのローカルコピーを保存し、データを定期的に同期する。
- NFSファイルサーバなどの共有ファイルシステムを使用し、そこにコンテンツを保存して、各バックエンドサーバが必要に応じてデータを読み込む。

静的Webコンテンツを処理する場合、どちらのコンテンツ管理方法も有効です。SMTPまたはPOP3サーバの場合は、サービスによってファイルが変更されるため、共有ファイルシステムを使用する必要があります。

フロントエンドサーバ

Traffic Managerの単純なデプロイでは、1台のフロントエンドマシンを使用します。この場合、フロントエンドマシンのフォールトトレランス機能を除き、Traffic Managerソフトウェアの全機能を利用できます。すなわち、Traffic Managerマシンが単一障害点になる可能性があります。

完全なフォールトトレランスを備えた構成には、2台以上のTraffic Managerが必要です。これらのTraffic ManagerのIPアドレスを1つのトラフィックIPグループにまとめることで、いずれかのマシンで障害が発生した場合にフォールトトレランスを提供することができます。トラフィックIPグループについては、[トラフィックIPグループとフォールトトレランス](#)で説明します。

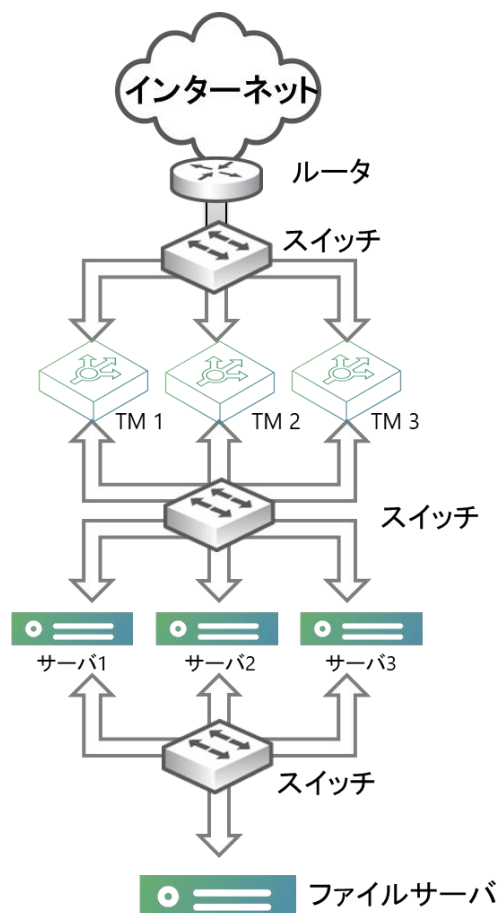
一般的なネットワークの構成では、フロントエンドマシンを、インターネットとの通信用のネットワークとバックエンドとの通信用のネットワーク(通常はプライベートネットワーク)の2つのネットワークに1つずつデプロイします。ただし、フロントエンドマシンとバックエンドマシンを同一ネットワークにデプロイすることも可能です。インターネットからルーティング可能な全てのマシンには、適切にファイアウォールを設定してください。ファイアウォールの設定およびネットワークセキュリティについては、[システムセキュリティ](#)を参照してください。

スループットを高めるために、Traffic Managerにネットワークカードを追加することもできます。必ずしも必要ではありませんが、ネットワークカードを2枚使用した場合、2つの独立したネットワークにTraffic Managerを簡単にデプロイできるというメリットがあります。

Traffic Managerは、非常に優れた拡張性を備えています。サーバを増設したり、既存のハードウェアをアップグレードしたりすると、それに応じてトラフィック管理の性能も向上します。後からTraffic Managerマシンを増設するときも簡単で、既存の構成に手を加えたりサービスを中断したりする必要がありません。

 ソフトウェアバージョンのTraffic Manager製品ファミリを使用している場合、他のアプリケーションおよびサービスによってホストマシンが過負荷状態にならないよう注意してください。例えば、Webブラウザやその他のGUIツールを同一サーバで実行すると、サーバ上のトラフィックを管理するソフトウェアの処理能力が低下します。

下図は、トラフィック管理される一般的なサーバファームを示しています。フロントエンドマシンには、前述のようにフロントエンド用とバックエンド用に個別のネットワークカードが使用されています。



バックエンドサーバが処理するコンテンツは、NFSファイルサーバに保持されます。各バックエンドサーバでは、スループットを最大限に引き上げるために、Traffic Managerマシンとの通信用と、このファイルサーバとの通信用に2枚のネットワークカードを使用します。

IPトランスペアレンシ

Traffic Managerは完全なアプリケーションプロキシとして機能し、リモートクライアントからのネットワークコネクションをいったん終端し、選択されたバックエンドサーバへのコネクションを新たに確立します。単純なレイヤ4ロードバランサで採用されているパケット指向型のNATベースの負荷分散手法は使用しません。

このアーキテクチャでは、バックエンドサーバはクライアントのリクエストを、リモートクライアントではなくTraffic Managerから出されたものとして認識します。このことは、バックエンドサーバがクライアントIPアドレスに基づいてアクセスコントロールを行ったり、リモートIPアドレスを記憶しようとする場合に、問題になる可能性があります。この状況は、通常、アクセスコントロールまたはロギング機能を自体で実行するか、Traffic ManagerがクライアントのIPアドレスを識別するために全てのHTTPコネクションに挿入する「X-Forwarded-For」または「X-Cluster-Client-Ip」ヘッダを利用することにより対処できます。

これらの対処方法が適切でない場合、Traffic Managerは、サーバ側コネクションの送信元IPアドレスをスプーフィングし、リクエストがクライアントのリモートIPアドレスから出されたかのように見せかけることができます。この機能をIPTランスペアレンシと言います。

IPTランスペアレンシは選択的に使用することができます。例えば、Traffic ManagerがWebファームおよびメールファームへのトラフィックの負荷分散を行っている場合、SMTPトラフィックに「IPTランスペアレンシ」を適用し、Webトラフィックには適用しないようにすることが可能です。

トランスペアレンシは、プール単位で有効化します。すなわち、前述の例では、Webプールにはトランスペアレンシを適用せず、SMTPプールにはトランスペアレンシを適用するよう設定します。この場合、WebプールおよびSMTPプールでは、トラフィックを同じバックエンドノードに負荷分散することも、異なるノードに負荷分散することもできます。

デフォルトでは、Traffic Managerのアプライアンスイメージ、仮想アプライアンス、またはクラウドサービスの全てのバージョンで、IPTランスペアレンシを利用できます。Traffic Managerソフトウェア製品では、LinuxホストまたはUNIXホストで、以下の条件でネイティブIPTランスペアレンシ機能を使用できます。

- Traffic Managerソフトウェアがインストールされ、rootユーザとして実行されている。
- ホストオペレーティングシステムで、バージョン2.6.24以降のカーネルが使用されている。
- ホストオペレーティングシステムで、バージョン1.4.11以降のiptablesが使用されている(--transparent オプションを使用可能な場合は、バージョン1.4.11よりも前のバージョンのiptablesもサポートされます)。

ルーティング設定

IPTランスペアレンシが適用されたトラフィックを受信した各サーバノードは、そのトラフィックの送信元であるTraffic Managerを経由してレスポンスをルートバックする必要があります。これを行うには、そのノードのデフォルトルートをTraffic ManagerのいずれかのバックエンドIPアドレスに設定します。デフォルトルートの設定の詳細については、OSのドキュメントを参照してください。

サーバノードは、リクエストに対するレスポンスを返すときに、リモートクライアントのIPにアドレス指定します。ルーティングが正しく設定されていれば、送信元のTraffic Managerはこのトラフィックを傍受し、コネクションを終端して通常どおり処理します。Traffic Managerは、クライアントコネクションを通して(場合によっては変更された)レスポンスをクライアントに転送します。

通常は、Traffic Managerシステムにアドレス指定されていないその他全てのパケットについても、Traffic Managerによって単純に転送されるよう設定します。これにより、Traffic Managerシステムはローカルルータとして機能し、Traffic Managerをデフォルトルートとして使用するサーバが近隣またはリモートネットワーク上の他のシステムと通信することができます。Traffic Managerでは、プライベートネットワーク上のバックエンドノードから転送する全てのパケットをNAT変換する必要があります。



ご使用の種類のTraffic Managerソフトウェアが稼働するホストシステムでこの「IP転送」機能を設定する場合は、OSのドキュメントを参照してください。Traffic Managerの仮想アプライアンスまたはクラウドインスタンスでこの動作を設定するには、システムレベルの設定を参照してください。

ローカルルーティングの制限事項

IPトランスペアレンシを使用する場合、バックエンドノードと同一ネットワーク上にあるクライアントは、Traffic Managerを通してこれらのノードにトラフィックを負荷分散することができません。

これは、バックエンドサーバノードが常に接続の送信元IPアドレスにレスポンスを返そうとするためです。送信元IPアドレス(クライアントのIP)がローカルネットワーク上に存在する場合、サーバノードは、接続元であるTraffic Managerシステムを経由してルーティングせずに、クライアントと直接通信しようとするため、接続がハングアップしたかのように見えます。

この場合、バックエンドネットワークをセグメント化し、サーバノード側から見てクライアントが別のサブネット上に存在し、デフォルトゲートウェイ(Traffic Managerシステム)を経由しなければアクセスできないかのように見せかけることで対処できることがあります。または、TrafficScriptルールにより、クライアントがサーバノードと同一ネットワーク上に存在する場合に、サーバ接続の送信元IPアドレスをTraffic Managerシステム上のIPに選択的に設定することも可能です。

IPトランスペアレンシとTraffic Managerクラスタ

各サーバノードは、1つのIPアドレスを経由してレスポンスをルートバックしなければならないため、トラフィックマネージャのクラスタを使用する場合は、ルーティングが複雑になります。

このような状況での推奨事項については、[トラフィックIPアドレスとトラフィックIPグループ](#)を参照してください。

トラフィックIPアドレスとトラフィックIPグループ

一般的なネットワークでは、バックエンドサーバは、ローカルIPアドレスが割り当てられてローカルネットワーク内に配置され、インターネットからは直接アクセスできません。一方、フロントエンドのTraffic Managerマシンは、外部からアクセス可能なIPアドレスが割り当てられ、ローカルネットワークを介してバックエンドマシンに接続することができます。

フロントエンドマシンには、各ネットワークで固定IPアドレスが割り当てられ、フロントエンドアドレスはインターネットから認識、ルーティングできます。これらのIPアドレスはOSによって設定され、マシンで障害が発生すると、そのIPアドレスは失われます。

このような理由から、これらの固定IPアドレスは、サービスを公開する場合には適していません。ハードウェアやシステムに障害が発生した場合に、サービスの一部または全部が利用できなくなるためです。

Traffic Managerのフォールトトレランス機能を利用すると、トラフィックIPアドレスを設定することができます。トラフィックIPアドレスは特定のマシンに紐付けられないため、Traffic Managerクラスタでは、クラスタ化された一部のTraffic Managerマシンで障害が発生しても、各IPアドレスは完全に利用可能な状態に保たれます。

フォールトトレランスを備えた一般的な構成では、サービスを公開するために使用するDNS名がTraffic Managerクラスタ内のトラフィックIPアドレスに名前解決されるよう設定します。これにより、サービスを完全に利用可能な状態に維持できます。

トラフィックIPはトラフィックIPグループに整理できます。このグループは、一部または全てのTraffic Managerマシンにまたがることができます。これらのマシン間でトラフィックIPアドレスを共有するための判別が行われ、各Traffic Managerでは、割り当てられたIPアドレス (1つまたは複数) が有効になります。

トラフィックIPグループの設定については、[トラフィックIPグループとフォールトトレランス](#)で説明します。

いずれかのTraffic Managerマシンで障害が発生すると、グループ内の他のTraffic Managerがこれを検出し、いずれかのTraffic Managerが、障害が発生したマシンのトラフィックIPアドレスを引き継いで、サービスが中断しないようにします。



デフォルトでは、フォールトトレランス機能ではユニキャストトラフィックによってヘルス状態が配信され、マルチホストトラフィックIPアドレスへのトラフィックが管理されます。マルチキャストトラフィックによる配信に設定を変更する場合、Traffic Managerを接続するスイッチは、IGMPスヌーピング、およびクラスタ内の全てのTraffic Managerに転送されなければならないTraffic Managerによって使用されるマルチキャストアドレスにブロードキャストされるメッセージをサポートしている必要があります。

Traffic Managerを既存のクラスタに組み込むと、様々なテストが自動的に実行され、ブロードキャストメッセージが正常に転送されるかどうかを確認されます。

トラフィックIPアドレスのモード

Traffic Managerは、以下の複数のモードをサポートしています。

- **シングルホスト:** トラフィックIPアドレスは、フォールトトレランスクラスタ内の単一のTraffic Managerで有効になります。このTraffic Managerで障害が発生した場合、別のTraffic ManagerがそのIPアドレスを引き継ぎ、トラフィックを受け入れ始めます。

- **マルチホスト:** トラフィックIPアドレスは、クラスタ内の全てのTraffic Managerで有効になり、マルチキャストMACアドレスを使用することで、受信トラフィックが全てのマシンに送信されます。トラフィックは、カスタムLinuxカーネルモジュールにより、稼働しているTraffic Manager間で均等に分散されます。
- **ルートヘルスインジェクション:** トラフィックIPアドレスは、加えられている全てのTraffic Managerによって「privateに」(ループバックで)有効になり、ルーティングプロトコルとしてOpen Shortest Path Firstバージョン2 (OSPFv2)またはBorder Gateway Protocol (BGP)のいずれかを使用して、隣接したルーティングドメインに動的にアドバタイズされます。レスポンスでは、ルータはアクティブなTraffic Managerにトラフィックを送信します。[ルートヘルスインジェクションおよびネットワーク](#)を参照してください。

マルチホストトラフィックIPアドレスを有効化すると、クラスタ内のTraffic Managerに要求されるパケット処理量が増加するため、性能が低下します。テストによる実測では、中程度のトラフィックレベル(毎秒10,000リクエスト)でCPU使用率が25～30%増加し、それに従って最大処理能力も制限されるという結果が示されています。



マルチホストIP機能は、Traffic Managerのハードウェアアプライアンスと仮想アプライアンスの全てのバージョンで利用できます。Traffic Managerソフトウェア製品にはデフォルトでは含まれていませんが、追加カーネルモジュールとしてダウンロードしてインストールすることができます。この機能はLinuxカーネルのバージョン2.6.18以上でサポートされます。サポートされているバージョンについての詳細は、PulseコミュニティのWebサイト(<https://community.pulsesecure.net>)を参照してください。

構成例

以下の構成は、クラスタに2台のTraffic Managerがあることを前提としていますが、これらの構成を、3台以上のTraffic Managerから成るクラスタに拡張することもできます。

アクティブ-パッシブ構成 – シングルホストモードとルートヘルスインジェクションモード

Webサイトの外部DNS名がIPアドレス162.34.64.29にマッピングされ、2台のTraffic Managerマシンによって多数のバックエンドWebサーバへのトラフィックが処理されているとします。

- シングルホストモードでは、上記の単一のIPアドレスを使用する、両方のTraffic ManagerマシンにまたがるシングルホストトラフィックIPグループを設定できます。2台のTraffic Manager間で判別が行われた後、いずれかのTraffic ManagerでこのIPアドレスが有効になり、全ての受信リクエストが処理されます。もう一方のTraffic Managerマシンはスタンバイ状態で待機します。最初のマシンで障害が発生した場合、2台目のマシンがIPアドレスを引き継ぎ、トラフィックの管理を開始します。

- ルートヘルスインジェクション(RHI)モードでは、上記の単一のIPアドレスを使用する、両方のTraffic ManagerマシンにまたがるRHIトラフィックIPグループを設定できます。一方のTraffic Managerをアクティブに、もう一方のTraffic Managerをパッシブに設定します。両方のTraffic ManagerがIPアドレスをアドバタイズし、(選択したルーティングプロトコルに応じて)OSPFv2やBGPメトリクスを使用して優先順を示します(アクティブなTraffic Managerが低い方のメトリクスを使用します)。アップストリームのルータ(一般的にデフォルトゲートウェイ)は、メトリクスが最も低いルータを選択して全てのトラフィックをアクティブなTraffic Managerに送信します。Traffic Managerは、障害を検出するとアドバタイズメントをキャンセルします。ルータは、障害を検出するとそのルートが無視します。いずれの場合も、ルータは2番目に最適なアドバタイズメントメトリクスに従った送信トラフィックに切り換わります(この場合はパッシブなTraffic Manager)。

この構成のメリットは、いずれかのTraffic Managerで障害が発生しても、トラフィックを処理する十分なリソースを確保できることです。また、トラフィックを処理しているTraffic Managerが1台のみのため、デバッグや障害解決が容易です。

アクティブ-アクティブ構成 – シングルホストモードとマルチホストモード

アクティブ-アクティブ構成では、両方のTraffic Managerがトラフィックを管理します。2台のTraffic Manager間でトラフィックをどのように分配するかは、分散モード(シングルホストIPまたはマルチホストIP)によってコントロールします。

シングルホストモードでは、トラフィックIPグループに2つのトラフィックIPアドレスを設定し、DNS名を162.34.64.29と162.34.64.30など2つのアドレスにマッピングするよう設定できます。2台のTraffic Manager間でネゴシエーションが行われた後、各Traffic Managerで1つずつトラフィックIPアドレスが有効になります。DNSサーバは、各IPアドレスに交互にリクエストを割り当てることができ(ラウンドロビンDNS)、Traffic Managerはそれぞれが受信したリクエストを処理します。

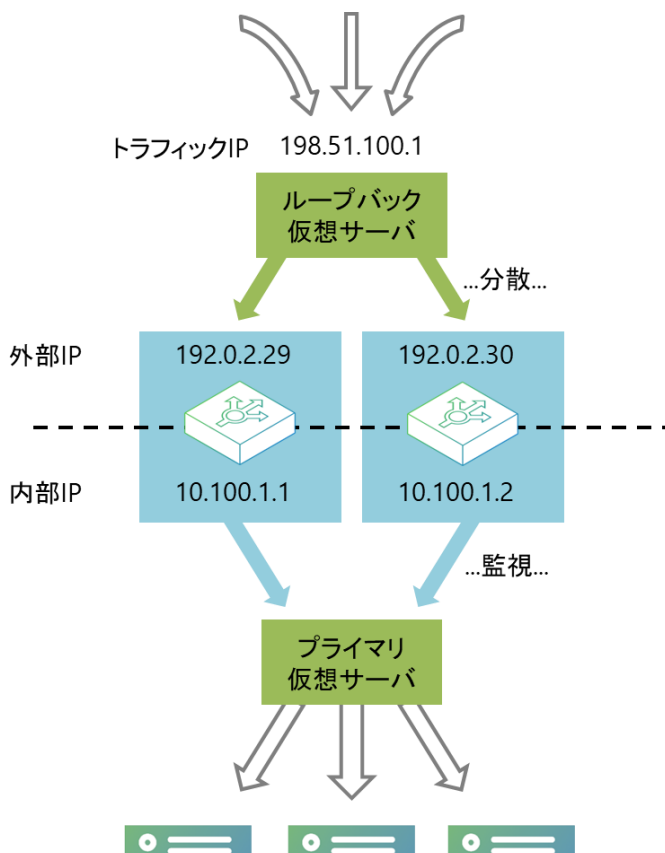
いずれかのマシンで障害が発生すると、もう一方のマシンがそれを検出します。このマシンは、元々使用していたトラフィックIPアドレスに加えて、障害が発生したマシンのトラフィックIPアドレスを引き継ぎ、両方のアドレスに送信された全てのトラフィックを処理します。

マルチホストモードでは、1つのトラフィックIPアドレスで継続的に運用できるため、DNSを単純化し、外部に公開する必要のあるIPアドレスの数を減らすことができます。トラフィックIPアドレスは、トラフィックIPグループに含まれる全てのTraffic Managerで有効になり、このIPアドレス宛ての受信トラフィックはTraffic Manager間で均等に分配されます。

ループバック付きアクティブ-アクティブ構成(単一の外部アドレス、シングルホストモード)

マルチホストモードが使用できない、またはインフラで禁止されている場合でも、単一の外部トラフィックIPアドレスを使用しながら、トラフィック負荷をクラスタ内の全てのTraffic Managerに分散することができます。

この構成では、追加の「ループバック」仮想サーバが外部トラフィックIPアドレスでトラフィックを監視し、Traffic Managerの固定IPアドレスを使用してクラスタ内のTraffic Managerにリクエストを負荷分散します。



まず、トラフィックの処理とバックエンドサーバへの負荷分散を行うプライマリ仮想サーバを作成します。この仮想サーバは、外部からアクセス可能なIPアドレスではなく、各Traffic Managerの内部IPアドレスとポートでトラフィックを監視するよう設定します。例えば、仮想サーバが、内部からのみ認識可能なIPアドレスである192.0.2.1:80と192.0.2.2:80でトラフィックを監視するようにします。

次に、セカンダリとして「ループバック」仮想サーバを作成します。この仮想サーバでは、単一の外部IPアドレスでトラフィックを監視し、クラスタ内の様々なTraffic Managerマシンにまたがるプライマリ仮想サーバにトラフィックをただちに送信します。

アクティブ-パッシブ構成の例と同様に、いずれかのTraffic Managerでのみ有効になる単一のトラフィックIPアドレスを1つ設定します。このアドレスで受信する全てのトラフィックは、そのトラフィックIPアドレスで監視を行っている単純なループバック仮想サーバで処理する必要があります。ループバック仮想サーバは、クラスタ内のTraffic Managerマシンの内部IPアドレスが含まれるループバックプールを即座に選択する必要があります。ループバックプールでは、ラウンドロビンまたは最小接続負荷分散によって、クラスタ内のTraffic Managerマシンにトラフィックを均等に分配する必要があります。レスポンス時間の影響を受ける負荷分散手法ではリクエストの分配が非常に不均等になるため、使用しないでください。

ループバック仮想サーバが使用する処理能力はわずかです。SSL復号化、ルール、コンテンツ圧縮などCPU負荷の大きい処理は全てプライマリ仮想サーバで実行するようにしてください。

この方法では、負荷の大きいタスクの負荷が2台のTraffic Managerに分割されます。いずれかのTraffic Managerで障害が発生しても、サービスは継続されます(一部トラフィックで瞬間的に中断する可能性もあります)。例えば、外部トラフィックIPアドレスをホストしているTraffic Managerで障害が発生した場合、そのトラフィックIPアドレスがもう一方のTraffic Managerに引き渡されます。また、ループバックプールでは、一方のノードが使用不可になっていることを検出し、残りのTraffic Managerで稼働しているプライマリ仮想サーバに全てのトラフィックを送信します。

ターゲット仮想サーバでは、リモートクライアントではなく、ループバック仮想サーバからの接続を監視します。一般に、「X-Forwarded-For」または「X-Client-Cluster-Ip」ヘッダから接続元を正しく特定できますが、通常、SSLリクエストはループバック仮想サーバによって転送されるため、[SSL暗号化](#)の「SSL転送におけるIPアドレスの維持」セクションで説明するssl_enhanceおよびssl_trust_magic設定を使用する必要があります。

多重冗長(N+M)構成

前述の例では2台のTraffic Managerマシンを使用していますが、いずれかのマシンで障害が発生した場合、システムに単一障害点が発生する可能性があります。ミッションクリティカルなサービスを実行している場合は、複数のTraffic Managerマシンを追加して多重冗長クラスタを構成することで、さらに強力な保護レベルを実装できます。

通常の運用で、N台のアクティブなTraffic Managerを使用する場合、M台のパッシブなTraffic Managerを組み込みます。Mは、障害が発生してもアクティブなTraffic Managerの台数を減らせずに済む台数です。

これを実現するには、Traffic Managerが稼働するN+M台のフロントエンドマシンが必要です。N個のトラフィックIPアドレスから成り、しかもN+M台のマシン全てで共有されるトラフィックIPグループを作成することができます。アクティブなグループのマシンで障害が発生した場合には、パッシブグループのバックアップマシンがアクティブになって負荷を引き継ぎます。さらに別のマシンで障害が発生すると、次のパッシブTraffic Managerがアクティブになり、というように続きます。これは、複数レイヤの冗長性を組み込んだ一般的なクラスタ編成です。

クラスタでのIPTランスペアレンシの使用

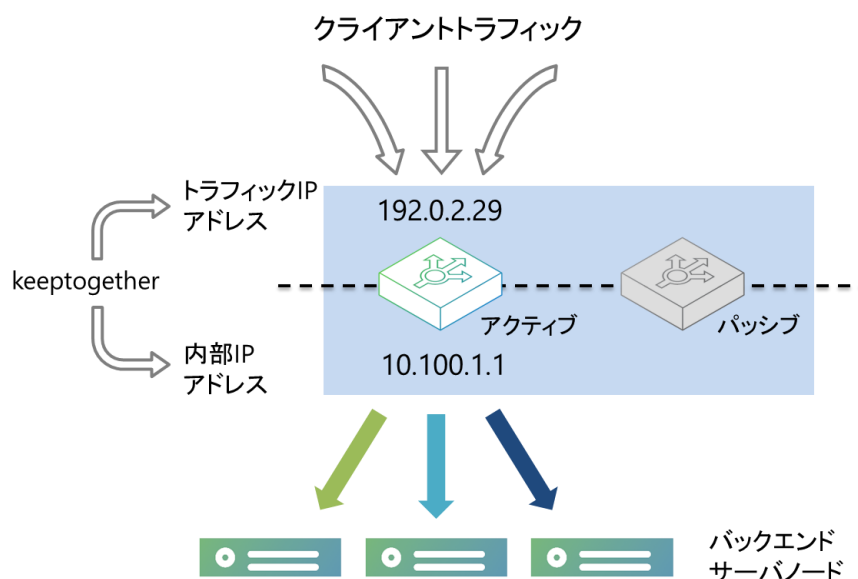
Traffic Managerマシンのクラスタでは、IPTランスペアレンシの使用が複雑になります。これは、各サーバノードが単一のTraffic Manager IPアドレスにルーティングするよう設定される一方、サーバノードに対してクラスタ内のどのTraffic Managerマシンもトランスペアレントに接続を確立する可能性があり、ノードは、その接続元のTraffic Managerを経由して各レスポンスをルートバックしなければならないからです。

アクティブ-パッシブ構成

アクティブなTraffic Managerが1つのみの構成では、これを、シングルホストIPアドレスを使用するトラフィックIPグループに**keep-together**を設定することにより実現できます。

受信トラフィック用のフロントエンドIPアドレスとサーバ側ネットワークに存在するバックエンドIPアドレスの2つのIPアドレスで構成されるトラフィックIPグループを作成します。トラフィックIPグループで**keep-together**オプションを選択します。

トラフィックIPグループに設定したバックエンドIPアドレスを経由して全てのトラフィックをルーティングするよう、各バックエンドサーバを設定します。

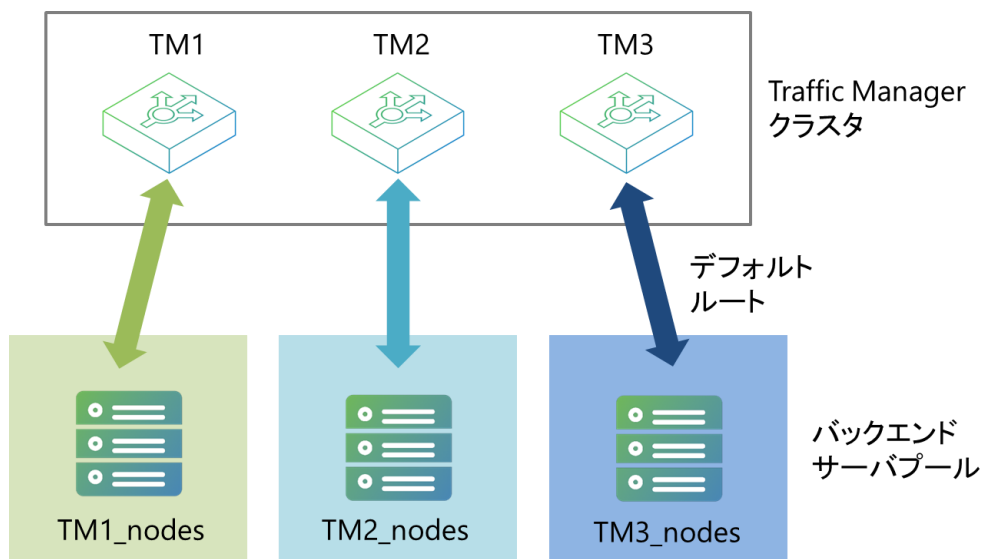


この設定により、両方のIPアドレスが同一のTraffic Managerマシンでホストされるようになります。そのマシンで障害が発生した場合、両方のIPアドレスが同じパッシブマシンに移行され、このマシンが新しいアクティブマシンになります。以降、バックエンドサーバは、新しいアクティブマシンを経由してトラフィックをルートバックします。

アクティブ-アクティブ構成

複数のアクティブなTraffic Managerを伴う構成では、バックエンドサーバをグループに分割し、各グループにアクティブなTraffic Managerマシンを1台ずつ割り当てる必要があります。次に、これらのグループをTraffic Manager管理UI内のプールとして定義し、それぞれにバックエンドサーバをノードとして追加します。

同じプールのサーバは全て、そのデフォルトルートに対応するTraffic ManagerのバックエンドIPアドレスに設定する必要があります。サーバのルート設定の操作方法については、ご使用のOSのドキュメントを参照してください。



次に、TrafficScriptルールを使用して、以下のいずれかの項目に基づいてトラフィックのルーティング先プールを正しく選択することができます。

- コネクションを管理しているTraffic Managerの名前
- ローカルクライアント側IPアドレス(シングルホストモードで複数の「keep-together」トラフィックIPグループを使用する場合)

以下のコードの断片に、この動作の実例を示します(Traffic Manager名を選択条件に使用します)。

```
$hostname = sys.hostname();

if( $hostname == "TM1" ) {
    pool.use( "TM1_nodes" );
}

if( $hostname == "TM2" ) {
    pool.use( "TM2_nodes" );
}

if( $hostname == "TM3" ) {
    pool.use( "TM3_nodes" );
}
```

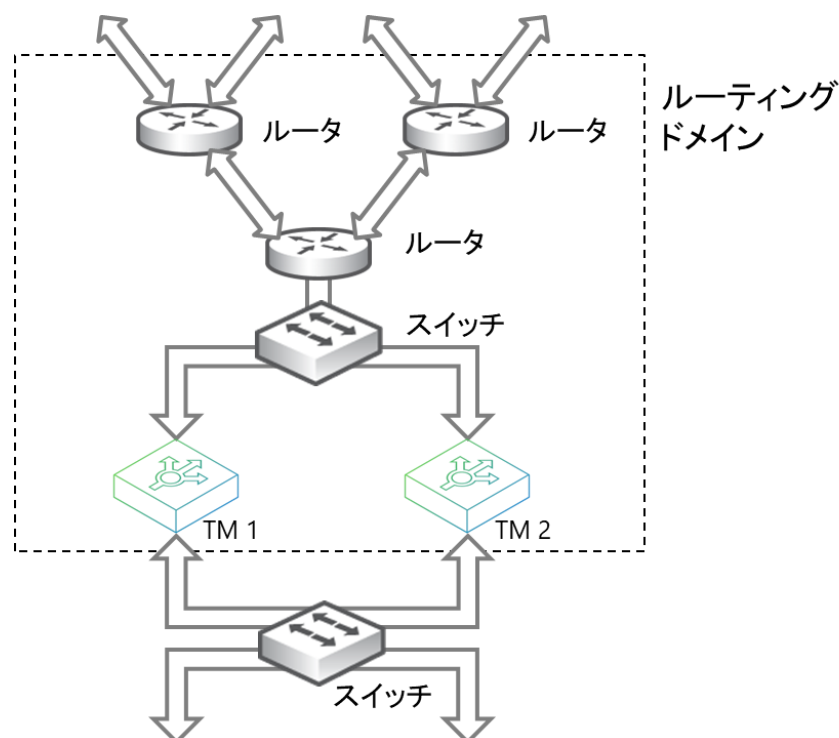
この設定には、Traffic Managerで障害が発生すると対応するプールが冗長になるという制限があります。また、セッション維持を使用するときの信頼性が下がります(特にマルチホストIPアドレスを使用する場合)。

IPトランスペアレンシは選択的に使用することができます。例えば、Traffic Managerクラスタがwww.mysite.comへの大容量Webトラフィックと、mail.mysite.comへの小容量SMTPトラフィックを管理しているとします。この場合、トランスペアレンシの適用が必要になるのはSMTPのみです。この場合は、以下のようになります。

- IPトランスペアレンシを適用しないアクティブ-アクティブTrafficCluster構成では、www.mysite.comを複数のIPアドレスに名前解決できます。
- 前述のアクティブ-パッシブ**keep-together**構成では、mail.mysite.comを単一のIPアドレスに名前解決できます。

ルートヘルスインジェクションおよびネットワーク

ルートヘルスインジェクション(RHI)を使用する場合、Traffic Managerはルータとの通信を隣接したルーティングドメインで行います。通信が確立されると、Traffic Managerはルーティングドメインに加わり、そのドメインにRHITraフィックIPアドレスをアドバタイズします。



このようなアドバタイズメントは動的に行われ、Traffic Managerの構成に対してRHITraフィックIPアドレスを作成、廃止または移動するような変更を加えた場合でも、通知は自動的にそれらの変更に応答を返します。さらに、Traffic Managerが検出した障害にもアドバタイズメントは自動的にレスポンスを返すほか、ルーティングドメインの動的なルーティングプロトコルを介して、Traffic Managerに固有ではないネットワーク障害にも自動的にレスポンスを返します。

これにより、幅広い種類のデプロイにおいてRHIを利用することが可能になります。例えば、ローカル接続のシンプルなフェールオーバーを管理するだけの小規模なデプロイ(単独で存在するデータセンタのラックの中など)はもちろん、企業内の複数のデータセンタ間やインターネット全体にわたってトラフィックの分配やフェールオーバーを管理する必要があるような、大規模なデプロイでも利用することができます。

RHIは、RHI専用のIPv4トラフィックIPグループを使用して機能します。単一のロケーションでは、単一のアクティブなTraffic Manager、またはアクティブ-パッシブのペアになったTraffic Managerのいずれかによって提供されるRHIトラフィックIPグループを使用することができます(43ページの「アクティブ-パッシブ構成 - シングルホストモードとルートヘルスインジェクションモード」を参照してください)。

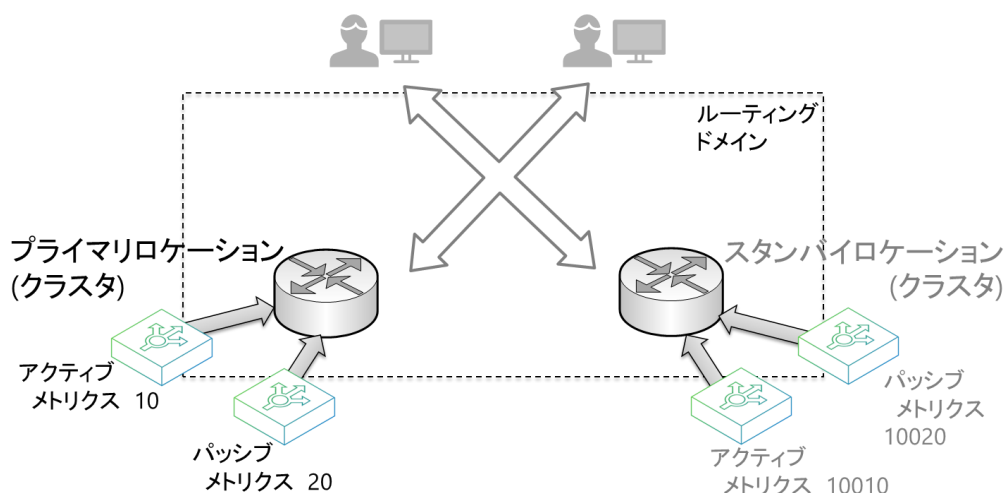
 RHIは、IPv6アドレスに基づくトラフィックIPグループをサポートしていません。

スケールを拡大するには、Traffic Managerのデータセンタのロケーションにおいて、このパターンを必要なだけ繰り返します。各グループのトラフィックIPアドレスが同一である場合、ロケーションごとに異なるRHIトラフィックIPグループが使用されていますが、通常、使用されているOSPFv2またはBGPメトリクスが異なります。

 OSPFv2の場合、このシナリオでは、全てのデータセンタが同じルーティングドメインにあることが要求されます。BGPの場合、データセンタは、ルーティングドメインの内部または外部に配置することができます。

各ロケーションには、異なる優先順位を設定することができます。例えば、プライマリ/スタンバイのロケーションが2点存在するデータセンタのデプロイでは、以下のように構成します。

1. プライマリデータセンタでは、アクティブ-パッシブのペアのTraffic Managerから提供されるプライマリRHIトラフィックIPグループを構成します。
2. スタンバイデータセンタでは、アクティブ-パッシブのペアのTraffic Managerから提供されるスタンバイRHIトラフィックIPグループを構成します。
3. プライマリデータセンタが利用可能である限りプライマリデータセンタが常に優先されるように、極端に高いメトリクスを用いてスタンバイRHIトラフィックIPグループを構成します。



別の方法として、データセンタのロケーションに同等の優先順位を設定することもできます。その場合は、アクティブなロケーションが複数存在することになり、ネットワークポロジに応じた最適なルートに基づいてルーティング経路が決定されます。一般的に、この方法はエニーキャスト構成と呼ばれています。各ロケーションのRHIトラフィックIPグループを同じメトリクスを用いて設定することによって、この構成を実現することができます。



複数のロケーションにわたるBGPを使用して実装されたRHIでは、RHIをサポートするためにインフラストラクチャの他の部分を設定する必要があります。言い換えれば、データセンタのロケーション間のルータが、提供されたメトリクスに従い、ロケーションの選択に影響を及ぼす可能性がある他のポリシーが設定されないようにする必要があります。

RHI通信で使用するクレデンシャル

各Traffic Managerは、OSPFv2を使用するか、またはBGPとのセッションを確立して、隣接ルータと通信することによりRHIを実行しています。Traffic Managerが、OSPFv2またはBGPに対応した隣接ルータとの通信を確立するためには、適切なクレデンシャルを用いてTraffic Managerを設定する必要があります。この手法は、ルータとの「ピアリング」と呼ばれています。



OSPFv2通信にはマルチキャストが必要です。

同じクレデンシャルを使用するTraffic Managerのクラスタを作成することができます。各クラスタ(またはマルチサイトマネージャクラスタ内の各ロケーション設定、404ページの「マルチサイトクラスタの管理」を参照)は1セットのクレデンシャルを使用するため、クラスタに含まれる全てのTraffic Manager(またはロケーション設定)が同一のエリア(OSPFv2の場合)または自律システム(AS)(BGPの場合)に加わります。

別のクレデンシャルを必要とする複数のロケーションがある場合は、異なるロケーションのTraffic Managerをクラスタ化する必要はありません。ただし、それらのTraffic Managerをクラスタ化したい場合には、Traffic Managerのマルチサイトマネージャ機能を使用する必要があります。

OSPFv2またはBGPのクレデンシャルを入力するには、**System** (システム) > **Fault Tolerance** (フォールトトレランス) ページを使用します。

OSPFv2/BGPの設定およびRHIトラフィックIPグループの設定の詳細については、[トラフィックIPグループとフォールトトレランス](#)を参照してください。

RHIに関する問題のトラブルシューティング

Traffic Managerは、サードパーティのルーティングソフトウェアを使用してRHIを運用しています。このルーティングソフトウェアは、ご使用のファイルシステムの\$ZEUSHOME/log/routing_swにある特定ファイルにRHIイベントログを記録します。

このファイルの内容はテクニカルサポートレポートに含まれており(「[サポートを利用するには](#)」を参照)、サポート窓口はこの情報を活用して、ご使用のTraffic Managerのデプロイで生じたRHI通信問題のトラブルシューティングを行います。

このログファイルには、以下のローテーションポリシーが適用されます。

- 仮想アプライアンスおよびクラウドインスタンスの場合、Traffic Managerは固定のファイルサイズ(デフォルトは50 MB)を使用して、ログローテーションを自動的に実施します。アーカイブされたログは「gzip」圧縮法で圧縮され、日付と連番を含むファイル名で保存されます。
- ソフトウェアインスタンスの場合は、独自のローテーションポリシーを制定する必要があります。ログファイルがローテーション(移動)されていることをTraffic Managerに伝えるために、\$ZEUSHOME/zxtm/zebos/bin/reload_logfileポストローテーションのスクリプトを使用して、新規ファイルでロギングプロセスを再開させます。

OSPFv2とBGPについて

Traffic Managerは、RHIのルーティングプロトコルとして、Open Shortest Path Firstバージョン2 (OSPFv2)とBorder Gateway Protocol (BGP)をサポートしています。

OSPFv2は「内部ゲートウェイプロトコル」の1つで、通常、例えばISPや大企業のネットワークなどにおいて、あるネットワーク上の単一の自律システム(AS)内でルートを分散するために使用されます。OSPFv2の働きにより、ルータは同一AS内にあるOSPFv2が設定された他のルータを自動検出し、同期することができます。

OSPFv2は、存続時間(TTL)値に1を使用し、TCPやUDPを介してではなく生のIPパケットを使用してレイヤ3で動作します。OSPFv2は、マルチキャストアドレス指定を使用して大量のルーティング情報をチェーン内の次のルータに送信すると同時に、OSPFv2自体のエラー検出およびエラー補正に対処することができます。通常、OSPFv2はISPなどの組織内部に存在するため、速やかに収束させることができます(あるルートが変更された場合、ネットワークの全てのルータが新しい最短ルートに合意した時点で収束が実現します)。

OSPFの詳細については、http://en.wikipedia.org/wiki/Open_Shortest_Path_Firstを参照してください。

一方、BGPは一般に、個々のASの外部でルーティングと到達可能性に関する情報を配信するために使用される外部ゲートウェイプロトコルです。



BGPは、内部ゲートウェイプロトコルとしても使用できます。そのためには、Traffic Managerで情報交換のためにInternal BGP (iBGPと呼ばれる)が使用されます。異なるASのルータ間の通信では、External BGP (eBGPと呼ばれる)が使用されます。

OSPFv2とは異なり、BGP対応ルータでは、ピアが自動検出されず、セッションが確立されるネイバーの明示的な設定を定義する必要があります。

BGPの詳細については、www.bgp4.asを参照してください。

上記のプロトコルのいずれかまたは両方を使用して、IPアドレスを通知し、ネットワークを通じてこの通知を伝搬させるようにTraffic Managerを設定できます。サービスでルートヘルスインジェクションを有効化し、使用するには、[トラフィックIPグループの作成](#)を参照してください。

IPv6について

IPv6は、パケット交換ネットワークで使用されるネットワークレイヤプロトコルです。IPv6の最大の特徴は利用可能なアドレスが莫大なことです。これは、IPv4のアドレスのビット長が32ビットなのに対し、IPv6では128ビットになっているためです。また、ネットワーク管理が単純化され、複雑なサブネットスキームを使用する必要がないというメリットもあります。

IPv6のメリットを以下に挙げます。

- IPセキュリティ
- モバイルIPアドレス
- 単純なヘッダ構造
- アドレスの自動設定
- エニーキャスト(1つのアドレスを多数のインターフェースで共有)と必須マルチキャストアドレス

Traffic ManagerでのIPv6の主な機能

Traffic Managerは以下のようなIPv6サポートを提供します。

- Traffic ManagerがIPv6のゲートウェイとして機能
- Traffic Managerが、フロントエンドサーバおよびバックエンドサーバで複数のバージョンのIPアドレスを処理可能
- Traffic ManagerはIPv4とIPv6の混在ネットワークでも、IPv4のみのネットワークでも動作可能

IPv4アドレスが使用可能であれば、Traffic Managerを設定するときに、トラフィックIPアドレス(RHIおよびマルチホストトラフィックIPアドレスを除く)、仮想サーバが監視するノードまたはアドレスの指定、TrafficScriptルールなどで、IPv6ユニキャストアドレスを使用できます。

Traffic Managerは、IPv4からIPv6またはIPv6からIPv4、あるいは同時にその両方のゲートウェイとして機能することもできます。

技術的制約

Traffic Manager環境でIPv6を使用する場合、いくつかの制約があります。これらの制約がソフトウェアの通常の動作に影響を与えることはありませんが、常に留意しておいてください。

IPv6の使用に関する主な制約は以下のとおりです。

- Traffic Manager間の内部通信はIPv4プロトコルで行われます。
- ハートビートメッセージはIPv4でのみ機能します。管理サーバも同様です。
- バックエンドノードの設定でホスト名を使用する場合、Traffic ManagerはまずIPv4アドレスを検索します。DNSがIPv4とIPv6の両方のアドレスを持っている場合にマシンでIPv6アドレスを使用するには、IPv6アドレスを直接入力する必要があります。
- ホストのDNSにIPv6アドレスのみが割り当てられている場合は、そのアドレスが使用されます。

重複アドレス検出の調整



このセクションは、Traffic Managerソフトウェア製品のみを対象としています。Traffic Managerのハードウェアアプライアンス、仮想アプライアンス、およびクラウド製品では、自動的に重複アドレス検出が調整され、設定されます。

多くのOSの重複アドレス検出(DAD)機能は、2台のマシンで同時に同じアドレスが有効にならないようにすることを目的としています。この機能は、Traffic Managerのフォールトトレランス機能と競合する可能性があります。IPがTraffic Managerシステムから別のTraffic Managerシステムに引き渡されるときに、タイミングによっては、そのアドレスを引き継ぐTraffic Managerに対してDADが実行されることがあります。DAD機能は以下のように調整することができます。

- Linux: sysctlを呼び出してnet.ipv6.conf.default.dad_transmitsおよびnet.ipv6.conf.all.dad_transmitsの値を確認します。/etc/sysctl.confファイルに、これらの行を以下のように追加(または変更)します。

```
net.ipv6.conf.default.dad_transmits = 0
```

および

```
net.ipv6.conf.all.dad_transmits = 0
```

この変更をすぐに有効にするには、関連する各インターフェース(この例ではeth1)と、「default」および「all」に対して以下のコマンドを使用します。

```
# sysctl -w net.ipv6.conf.eth1.dad_transmits=0
# sysctl -w net.ipv6.conf.default.dad_transmits=0
# sysctl -w net.ipv6.conf.all.dad_transmits=0
```


初期設定

本章では、トラフィック管理される基本的なサイトをセットアップする方法について説明します。

アーキテクチャコンセプト

Traffic Managerは、Webサーバやアプリケーションサーバ(HTTP、HTTPS)のようなネットワークサービス、Webサービス(SOAP)、メール(SMTP、POP、IMAP)、およびDNSやストリーミングメディアなどその他プロトコルのトラフィックを管理します。

「サービス」は、IPアドレスおよびポートとして公開され、適切なプロトコルを使用してトラフィックを受け入れます。

ウェブサイトはwww.example.comでホストされます。このWebサイト名はその後IPアドレスを解決するためDNSでマッピングされ、対応するポート上の特定のトラフィックのタイプ向けに、そのIPアドレスを監視するようにTraffic Managerサービスを設定します(ポート80のHTTPトラフィックなど)。

例えば、インターネットサービスプロバイダ(ISP)は、POP3およびSMTPメールサーバをpop.example.comおよびsmtp.example.comとして公開します。これらの各サーバのDNSエントリは以下のIPアドレスにリンクされています。

- pop.example.comのアドレスは192.0.2.150
- smtp.example.comのアドレスは192.0.2.151

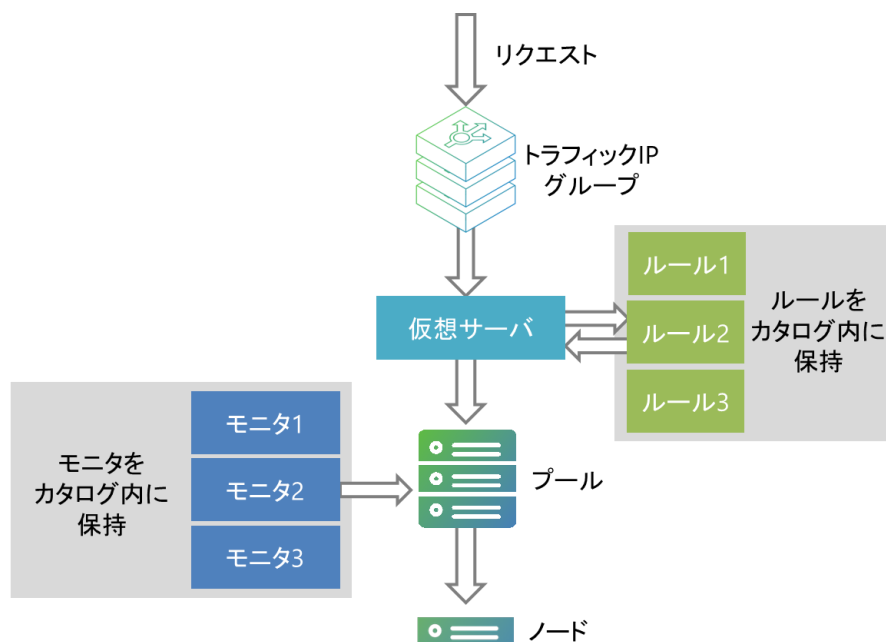
pop.example.comはポート110でPOP3トラフィックを監視し、smtp.example.comはポート25でSMTPトラフィックを監視します。

Traffic Manager内では、特定のサービスに対する全てのトラフィックが「仮想サーバ」で処理されます。仮想サーバは、Traffic Managerとインターネット間のインターフェースとして機能します。指定のポートとプロトコルが設定され、通常、そのプロトコルのトラフィックを全て管理します。

仮想サーバは、リクエストを受信すると、それを「プール」に割り当てます。プールはノードの集合であり、各ノードはserver1.example.com:80のように特定のバックエンドサーバとポートに対応しています。トラフィックはプールによってノードに負荷分散されます。複数のプール間で同じノードを設定することも可能です。

仮想サーバは、リクエストに対して使用するプールを決定するために、「ルール」のリストを適用できます。ルールでは、受信したリクエストを調べ、それに対して実行するアクションを決定します。ルールによって、リクエストを処理するプールを選択したり、リクエストを終了したり、またはリクエストをリスト内の次のルールに渡したりすることも可能です。どのルールでも明確なルーティングが決定されなかったリクエストは、仮想サーバの「デフォルトプール」に割り当てられます。

プール内のノードで障害が発生した場合、Traffic Managerの「モニタ」がこれを自動的に検出します。その結果、Traffic Managerはそのノードへのリクエストの送信を停止します。トラフィックは、プール内の他のノードに分配されるため、サービスで目に見えるような中断は起こりません。



Traffic Managerマシンでは、管理するサービスごとに1つずつ、多数の仮想サーバを実行することができます。各仮想サーバは複数のルールとプールを使用でき、様々なモニタによってプールを監視できます。

ルールとモニタは、「カタログ」として区分されます。例えば、ルールカタログには、そのシステム用に作成された全てのルールが保持されます。カタログ内のルールは、任意の仮想サーバに簡単に適用することができます。SSL証明書やサービスプロテクションクラスなどのその他の共有項目もカタログに保持されます。

複数のTraffic Managerマシンを「クラスタ」として扱うことができます。クラスタ内の全てのTraffic Managerマシンでは設定が共有され、各マシンで実行される仮想サーバ、使用されるルールなどは同じです。

サービスで使用するパブリックトラフィックIPアドレスは、「トラフィックIPグループ」の中に割り当てることができます。1つのトラフィックIPグループは、クラスタ内の一部または全てのTraffic Managerにまたがり、これらのTraffic Manager間でそのグループのIPアドレスがホストされます。特定のTraffic Managerマシンで障害が発生した場合、同じトラフィックIPグループに含まれる他のマシンのいずれかがそのIPアドレスを引き継ぎます。この動作とプールのフェールオーバーシステムにより、フロントエンドとバックエンドの両方に完全なフォールトレランスが実現されます。

Traffic Managerの実行中、ホスト名は、通常5分以下の間隔で定期的に名前解決されます。これにより、DNSが変更された場合も、ノードを効率よく名前解決することができます。

最初のサービスの管理

最初のサービスの管理を開始するには、仮想サーバとデフォルトプールを作成する必要があります。これは、「Manage a New Service (新規サービスを管理)」ウィザードまたは「Configure (設定)」ページから行えます。

設定済みのTraffic Managerの管理UIのWebアドレスでブラウジングします。ユーザ名とパスワードを使用してログインします。



Traffic Managerのインストール方法と設定の詳細については、ご利用の製品形態の『Pulse Secure Virtual Traffic Manager: Installation and Getting Started Guide』をご覧ください。

ウィザードを使用した仮想サーバとプールの作成

「Wizards (ウィザード)」ドロップダウンメニューをクリックして、**Manage a New Service** (新規サービスを管理) ウィザードを選択します。その後表示される指示に従います。

1. 仮想サーバを識別するための名前を指定します。仮想サーバのプロトコルとポート(例: HTTP、ポート 80)を選択します。

Manage a new Service, step 2 of 4

2. Specify the service

Please enter a brief name to identify the service you would like to balance.

Name:

Please select the protocol that the service uses.

Protocol:

Please specify the port that the protocol listens on.

Port:

Cancel

◀ Back

Next ▶

2. バックエンドノードのリストを作成します。これらのノードによって、仮想サーバのデフォルトプールが形成されます。ノードは、ホスト名とポートで識別されます。これらのパラメータは、後で**Pools (プール) > Edit (編集)**ページから変更できます([プール](#)を参照)。指定するホスト名とポート番号の組み合わせによってコンテンツを直接提供できるようにしてください。

Manage a new Service, step 3 of 4

3. Specify the back-end nodes

Please enter the hostname and port of each node:

Hostname: Port:

Nodes:

```
server1.mysite.com:80
server2.mysite.com:80
server3.mysite.com:80
```

To remove a node from the list, select it and press 'Remove node':

i 評価のために、Traffic ManagerをWebサーバ(またはその他のテストサーバ)と同じマシンで実行している場合、2つのサービスが異なるポートでトラフィックを監視するようにしてください。

例えば、Webサーバへのトラフィックを管理しているとします。デフォルトHTTPポートは80です。Webサーバがポート8080で、またTraffic Managerがポート80でトラフィックを監視するよう設定します。この場合、ブラウザにマシン名(例: server1.mysite.com)を入力すると、トラフィックはTraffic Manager経由で送信されます。

3. 選択した設定を見直してから、**Finish** (完了)をクリックします。
4. ここで、新しいサービスに設定したマシンとポートをブラウズして、Traffic Managerの設定をテストすることができます。

i 仮想サーバ、プール、エラーファイルなどの名前の先頭文字にはピリオド(.)、アンダースコア(_)、またはシャープ(#)を使用できません。また、最後の文字にはチルダ(~)を使用できません。

手動でのプールと仮想サーバの作成

管理UIのトップバーにある**Services** (サービス)ボタンをクリックし、Services (サービス)ページを使用して、仮想サーバ、プール、およびTraffic Managerを管理します。

プールの作成

1. まず、バックエンドサーバのプールを作成する必要があります。**Pools** (プール)タブをクリックし、「Create a New Pool (新規プールの作成)」セクションに詳細を入力します。

プールの名前を選択し、ノードのリストを入力します(オートスケーリングを有効にする場合、ノードは不要)。各ノードは、「server1.mysite.com:80」の形式で指定する必要があります。ここで、80は「server1」がトラフィックを監視しているポートです。ノードはスペースで区切ってください。

```
s1.mysite.com:80 s2.mysite.com:8080 s3.mysite.com:80
```

2. **Create Pool** (プールの作成)をクリックします。新しいプールの**Pools** (プール) > **Edit** (編集)ページに移動します。このページでは、基本設定や詳細設定を編集できます。

仮想サーバの作成

1. **Virtual Servers** (仮想サーバ)タブをクリックして、新しい仮想サーバを作成します。名前、プロトコル、および仮想サーバがトラフィックを監視するポートを指定する必要があります。Traffic Managerとサーバソフトウェアを同じマシンにインストールする場合は、前のセクションのポート番号に関する注記に留意してください。

仮想サーバのデフォルトプールを選択する必要があります。ルールによって別のプールまたはアクションが指定されない場合、リクエストはこのプールに割り当てられます。

2. **Create Virtual Server** (仮想サーバを作成)をクリックすると、新しい仮想サーバの**Virtual Servers** (仮想サーバ) > **Edit** (編集)ページに移動します。このページでは、基本設定や詳細設定を編集することができます。これには、仮想サーバがトラフィックを監視するIPアドレス(ドメイン名またはアドレスで指定)も含まれます。

クラスタの作成

一般に、Traffic Managerは、フォールトトレランス([トラフィックIPグループとフォールトトレランス](#)を参照)および管理上の理由から、複数のインスタンスから成るクラスタとしてデプロイします。同じクラスタ内の全てのTraffic Managerは、同じ設定を共有し、単一のエンティティとして管理されます。



ただし、ネットワーク設定などのシステム固有の設定はクラスタメンバーごとに固有のまま保たれ、個別に管理する必要があります。

複数のTraffic Managerシステムをまとめてクラスタを形成するには、以下のいずれかの方法を用います。

- 新たにTraffic Managerクラスタを作成する場合、いずれかのTraffic Managerを最初のクラスタメンバーとして選択します。次に、他の各Traffic Managerの管理UIに順々にログインし、Join a cluster (クラスタに組み込む)ウィザードを使用して、最初のTraffic Managerに組み込みます。
- 既存のTraffic Managerクラスタに組み込む場合は、新しい各インスタンスの管理UIにログインし、Join a cluster (クラスタに組み込む)ウィザードを使用して、そのインスタンスを既存のクラスタに組み込みます。

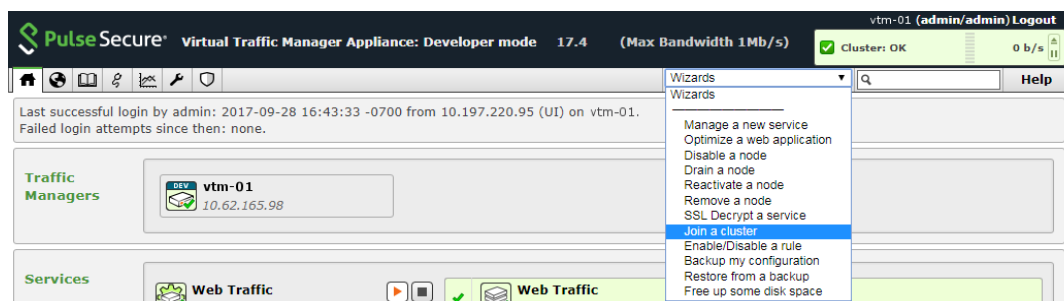
クラスタ内では、全てのTraffic Managerが同等に扱われます。任意のTraffic Managerの管理UIにアクセスでき、設定を変更すると、変更内容が自動的にクラスタ全体に反映されます。全てのTraffic Managerが連携することで、フォールトトレランスと簡潔な管理が実現されます。



ソフトウェアのアップグレード中に、一度にクラスタ全体を更新するには、1つのTraffic Managerのみを使用する必要があります。これにより、複数の同時更新が実行され、自動設定複製メカニズムが破損する可能性を防ぐことができます。アップグレードの詳細については、ご使用の製品の『Pulse Secure Virtual Traffic Manager: **Installation and Getting Started Guide**』を参照してください。

クラスタへの組み込み

いずれかのTraffic Managerの管理UIにログインし、ツールバーの「Wizards:(ウィザード:)」ドロップダウンボックスから**Join a cluster** (クラスタに組み込む)ウィザードを選択します。



Cluster Joining (クラスタに組み込む)ウィザードは、別のポップアップウィンドウで開かれます。

Cluster Joining wizard, step 1 of 5

1. Getting Started

This wizard joins your current traffic manager to an existing cluster so that it can share the cluster's configuration and traffic.

Joining a new cluster will remove this traffic manager from its current cluster.

Would you like to select an existing cluster from a list of available clusters on your network, or enter the Administration Server address and port of a specific traffic manager to join?

- ☒ Select existing cluster
☐ Manually specify host/port

Cancel ◀ Back Next ▶

ネットワークで通信可能なTraffic Managerを自動的にスキャンするようにTraffic Managerに指示するには、「select existing cluster (既存のクラスタの選択)」をクリックします。または、組み込む特定のホスト名とポートを入力するには、「Manually specify host/port (ホスト/ポートの任意指定)」をクリックします。**Next** (次へ)をクリックして操作を続けます。

次のステップには、**ステップ1**の選択が反映されます。「select existing cluster (既存のクラスタの選択)」をクリックした場合は、検出されたTraffic Managerインスタンスおよびクラスタのリストが表示されます。

Cluster Joining wizard, step 2 of 5

2. Cluster selection

Please select the cluster you wish to join:

- ☐ Cluster 1: aknox-02.cam.zeus.com:9092
- ☐ Cluster 2: apritchard-12.cam.zeus.com:9090
- ☐ Cluster 3: coeus.cam.zeus.com:9090
- ☐ Cluster 4: fry:9090
- ☐ Cluster 5: rkistruck-2b:9090 rkistruck-2d.cam.zeus.com:9090
- ☐ Cluster 6: jmoore-01:9090
- ☐ Cluster 7: jsteele-00.cam.zeus.com:9090 jsteele-04.cam.zeus.com:9090

Cancel ◀ Back Next ▶

「Manually specify host/port (ホスト/ポートの任意指定)」をクリックした場合は、表示されるボックスにホスト名とポート番号を入力します。

Cluster Joining wizard, step 2 of 5

2. Cluster selection

Please provide the admin server host and port of one of the machines in the cluster you wish to join:

Hostname:

Port:

Cancel ◀ Back Next ▶

Next (次へ)をクリックして操作を続けます。

指定したインスタンスまたはクラスタに接続するには、クラスタ内のTraffic ManagerのIDを検証し、クラスタによって使用される管理クレデンシャルを提供する必要があります。

Cluster Joining wizard, step 3 of 5

3. Authentication

The admin server you are clustering with is using an SSL certificate with the following SHA-1 fingerprint:

10.62.165.97:9090 ☒ B6:35:68:29:76:56:15:C0:FF:76
69:89:DA:30:7A:DB:02:60:2A:89
▶ **Unfold to view full certificate details ...**

Please check the box beside the fingerprint above to indicate that you have verified it or that you trust the network between it and this system.

If you do not already have this fingerprint on record you can get it by logging into the target admin server and visiting the **System > Security** page. (Refer to the product documentation for further information on cluster security.)

Enter the username and password of a user in the target cluster with permission to add and remove traffic managers.

Username:

Password:

Cancel ◀ Back Next ▶

組み込むクラスタで各Traffic ManagerのIDを検証することをお奨めします。Traffic ManagerのIDを検証するには、表示されたSHA-1フィンガープリントを、対象Traffic Managerの管理UIで**System** (システム) > **Security** (セキュリティ)に表示されるフィンガープリントと照合してチェックします。

各Traffic Managerの横にあるチェックボックスを選択して、そのIDを信頼することを確認し、クラスタの管理ユーザ名とパスワードを入力します。**Next** (次へ)をクリックして操作を続けます。

設定を確認し、**Finish** (完了)をクリックしてクラスタに組み込みます。Traffic Managerソフトウェアが自動的に再設定され、新しいホームページが表示されて、接続された全てのTraffic ManagerインスタンスがTraffic Managerリストに示されます。

さらにTraffic Managerをクラスタに追加するには、追加する各Traffic Managerの管理UIでJoin a cluster (クラスタに組み込む)ウィザードを実行します。

Traffic Managerを組み込もうとしている既存のクラスタがウィザードのリストに表示されない場合は、ネットワーク設定とケーブル接続を確認し、ネットワークでマルチキャストおよびブロードキャストパケットが許可されていることを確認してください。

トラフィックIPグループが設定されたクラスタへの組み込み

クラスタに既に1つ以上のトラフィックIPグループが設定されている場合、ウィザードでは、新しいTraffic ManagerをこれらのトラフィックIPグループに追加し、ただちにトラフィックの処理を開始できるようにすることができます。

ただし、この場合、トラフィックの割り当てを新しいTraffic Managerに引き渡す必要があるため、新しいTraffic ManagerをトラフィックIPグループに追加する瞬間に、多数のコネクションが失われる可能性があります。このような状況に備えて、新しいTraffic Managerを、トラフィックIPグループの「パッシブ」メンバとして追加することができます。こうすることで、グループ内の別のメンバで障害が発生するまで、そのTraffic Managerがトラフィックを受け入れることはありません。

仮想サーバ

本章では、Traffic Managerとインターネット間のインターフェースを表現するために使用される設定オブジェクトである、仮想サーバの概念について説明します。

はじめに

仮想サーバは、指定されたポートとプロトコルの全てのトラフィックを管理します。リクエストを処理するプールを決定するためにルールを適用したり、必要に応じてSSLトラフィックを復号化したりすることもできます。

Traffic Managerは、幅広いプロトコルをサポートしています。HTTP、RTSP、SIPなど複雑なプロトコル用に専用のハンドラも用意されており、ソフトウェアは、「一般的な」バージョンのプロトコルを使用してTCPコネクションやUDPセッションのトラフィックを負荷分散することができます。プロトコルのサポートの詳細については、[プロトコルのサポート](#)で説明します

仮想サーバのリストを表示するには、**Services** (サービス) > **Virtual Servers** (仮想サーバ)をクリックします。既存の仮想サーバの名前をクリックすると設定を変更でき、「Create a new Virtual Server (新しい仮想サーバの作成)」セクションを使用すると新しい仮想サーバを作成できます。

仮想サーバの使用

一般に、実行するサービスごとに1つの仮想サーバを実行するよう計画してください。すなわち、トラフィックを受け入れるTCPポートまたはUDPポートごとに実行するよう計画します。

例えば、ポート80で実行しているWebサイトの数には関係なく、HTTPに対して1つの仮想サーバを実行します。これは、Webサイトごとに仮想サーバを1つずつ設定するWebサーバの設定方法とは異なります。Webサイトごとに必要なトラフィック管理設定がわずかに異なる場合は、TrafficScriptを使用してこれらの設定を実装し、トラフィックの管理方法をコントロールすることができます。

一般的なTraffic Managerのインストールでは、5または10の仮想サーバのみが設定されています。仮想サーバ数が100を超えることが見込まれる場合は、Webベースの管理UIではなく、コントロールAPI([Traffic ManagerのコントロールAPI](#)を参照)またはREST API([Pulse Secure Virtual Traffic Manager: REST API Guide](#)を参照)を使用してプログラムによってTraffic Managerシステムを管理するよう計画してください。

プロトコルの選択

仮想サーバを設定するときに、その仮想サーバが処理するトラフィックのプロトコルを指定する必要があります。指定したプロトコルの値は内部プロトコルとして、Traffic Manager内でトラフィックを解析および解釈するために使用されます。管理するトラフィックに最適なプロトコル設定を選択してください。

「Generic Server First」、「Generic Client First」、および「Generic Streaming」プロトコルは、TCPコネクションを処理する単純なTCPプロトコルハンドラで、クライアントとサーバ間でデータを転送します。この3つのプロトコルの違いは以下のとおりです。

- Traffic Managerは「Generic Server First」仮想サーバでコネクションを受け取ると、ただちにTrafficScriptルールを実行し、負荷分散に関する決定を行ってバックエンドサーバに接続します。その後、クライアントとサーバとの間でデータを双方向に渡します。

このプロトコルは、クライアントアプリケーションの接続時に、クライアントがデータをサーバに書き込む前にサーバアプリケーションが「handshake」または「banner」メッセージを書き込むプロトコルに適しています。

- Traffic Managerは「Generic Client First」仮想サーバでコネクションを受け取ると、クライアントから何らかのデータを受信するまでコネクションを処理しません。その後、Traffic ManagerはTrafficScriptルールを実行して、負荷分散に関する決定を行い、選択されたバックエンドサーバに接続し、クライアントから受信したデータを書き込みます。以降、Traffic Managerはクライアントとサーバとの間でデータを双方向に渡します。

このプロトコルは、サーバにおいて、クライアントがまずデータを書き込んでダイアログを開始することが想定されているプロトコルに適しています。

- 「Generic Streaming」プロトコルはGeneric Server Firstに似ていますが、クライアントまたはバックエンドサーバから送信されたリクエスト/レスポンス以外のデータを仮想サーバが処理することが想定されている場合に適しています。Traffic Managerは、関連付けられたTrafficScriptルールを毎回実行して、負荷分散に関する決定を行い、その後クライアントとサーバとの間でデータを双方向に渡します。

このプロトコルは、サーバまたはクライアントがダイアログを開始でき、受信したデータに対するレスポンスを自動的に想定しないプロトコルに適しています。

これらの一般プロトコルは、他のTCPプロトコルハンドラのベースとなります。一般プロトコルでは、リクエストおよびレスポンスデータをチェックして書き換え、クライアントとサーバ間の通信を同期することができますが、これらの処理は非常に低いレベルに留まります。

ソフトウェア内のその他のTCPプロトコルハンドラは、FTP、HTTPなどの特定のプロトコル用に特化されています。HTTPなどのより特化されたプロトコルを選択すると、以下のようになります。

- 仮想サーバおよびプールの設定において、HTTPの「keepalive」設定やFTPのセキュリティ設定など、より多くのプロトコル固有のオプションを利用できるようになります。
- Traffic Managerによってそのプロトコルのメッセージが解析されるため、例えばHTTPトランザクションでクッキーを読み取って設定するヘルパー関数など、追加のTrafficScript機能を使用してリクエストおよびレスポンスデータを容易に管理できるようになります。

HTTPなど高レベルプロトコルを処理する場合は、適切な基本の「Generic」プロトコルハンドラを使用できます。この場合、そのプロトコル用に特化された処理を利用することはできません。

仮想サーバではUDPによるトラフィックもサポートされます。基本的なUDP (DNSなどの単純なUDPリクエスト-レスポンスプロトコル用)とUDPストリーミング(サーバがリクエストに対するレスポンスとしてUDPパケットを無制限に送信可能)のいずれかを選択できます。

特定のプロトコルのサポートに関する詳細については、[プロトコルのサポート](#)を参照してください。



SSLで暗号化されたトラフィックをTraffic Managerが復号化する場合、復号化されたトラフィックのプロトコルがプロトコル値として使用されます。例えば、Traffic Managerがポート443でHTTPストラフィックを受信し、SSL復号化によってそのトラフィックを復号化する場合、Traffic Manager内で使用されるプロトコルはHTTPになります。

ルールの適用

仮想サーバは、実行するアクションを選択するために、TrafficScriptルールを使用してリクエストをチェックします。各ルールによってリクエストがチェックされ、場合によっては変更され、最終アクションが実行されます。

- リクエストを処理するプールを選択する。
- リクエストを終了する。
- トランザクションを記録する。
- 何も行わず、リクエストをリスト内の次のルールに渡す。

各仮想サーバには、「デフォルトプール」が関連付けられます。どのルールでも明確なルーティングが決定されなかったリクエストは、このプールに割り当てられます。

ルールでは、特定の接続について、Traffic Managerの機能を選択的に有効化または無効化することもできます。例えば、ルールによって、リクエストが特定のセッション維持クラスを使用するようにしたり、レスポンスをキャッシュしないように指定することが可能です。

ルールは、IvantiのTrafficScript言語を使用して作成します。この言語では、受信するリクエストについて、送信元および宛先ポートやIPからトラフィックのタイプや実際のコンテンツまで、あらゆる側面をチェックすることができます。TrafficScriptでは、XPath (XMLドキュメントのクエリに使用される言語)、XSLT、およびその他のXML固有の機能がサポートされます。これらのXML機能は、Webサービスで採用されているSOAPベースのプロトコルで広く使用されており、ユーザが介入しなくても複雑なデータを自動的に交換したり解釈したりすることができます。

ルールは、指定した順序で適用されます。ルールは、トランザクションの完了時に、受信する各リクエストおよび外部に送信されるレスポンスに適用することができます。

ルールの作成はルールカタログで行い(管理UIの**Catalogs** (カタログ) > **Rules** (ルール)ページを参照)、仮想サーバを介して適用します。全ての仮想サーバは、カタログの任意のルールを使用することができます。

仮想サーバにルールを追加するには、該当する**Virtual Servers** (仮想サーバ) > **Edit** (編集)ページに移動し、**Rules** (ルール)をクリックします。リクエストルールの場合は、「Request Rules (リクエストルール)」セクションを使用します。同様に、レスポンスルールおよびトランザクション完了ルールの場合は、それぞれ「Response Rules (レスポンスルール)」および「Transaction Completion Rules (トランザクション完了ルール)」セクションを使用します。目的のセクションで、ドロップダウンボックスから追加する新しいルールを選択し、**Add Rule** (ルールを追加)をクリックします。

仮想サーバのルールはリストに表示され、表示されている順序で適用されます。リスト内のルールは、各ルールバーの左にあるグラブハンドルを使用して上下に移動できます。マウスカーソルを使用して、新たな位置にルールをドラッグアンドドロップすると、Traffic Managerは自動的にリストの並び替えを行います。ルールが1つしかない場合、グラブハンドルは表示されず、セクション間でルールをドラッグすることはできません。

リスト内のルールは、一時的に実行されないよう無効化したり、再び有効化して再度アクティブにしたりすることができます。

多数のリクエストとレスポンスによる長いダイアログを1つのコネクションで実行する非HTTPプロトコルの場合、ルールを「一度のみ」(最初のリクエストに対してのみ)実行するか、「全て」のリクエストに対して実行するかを選択できます。[TrafficScriptルール](#)では、ルールの作成と適用について詳しく説明します。

TrafficScript言語については、次のIvantiのWebサイト(www.ivanti.com)にある専用のリファレンスガイドに記載されています。

SSL復号化

仮想サーバは、SSLトラフィックを復号化することができます。この機能は、以下の2つの理由から有益です。

1. 復号化の後に、ルールによってリクエストのヘッダとコンテンツを解析し、これらの情報に基づいてルーティングを決定することができます。復号化しない場合、パケットから得られる情報はほんのわずかです。

2. リクエストを復号化するには、処理能力が必要です。Traffic Managerがノードに渡す前にリクエストを復号化すれば、バックエンドサーバの負荷が軽減されるため、効率の向上を図ることができます。

ルールを適用するためにトラフィックを復号化した場合、バックエンドサーバへの送信前にトラフィックを再暗号化することもできます。再暗号化は、プールで処理されます([SSL暗号化](#)を参照)。

仮想サーバでSSLトラフィックを復号化するように設定するには、その仮想サーバの**Virtual Servers** (仮想サーバ) > **Edit** (編集) ページに移動し、**SSL Decryption** (SSL復号) をクリックします。トラフィックを復号化するかどうか、SSL証明書カタログのどの証明書を使用するかを選択できます。各仮想サーバに対して許可する暗号スイートとSSLまたはTLS(トランスポート層セキュリティ)のバージョンを設定することもできます。

SSLクライアント証明書をリクエストするかどうかを選択することもできます。クライアント証明書はクライアントの識別に役立ち、選択した個人にのみアクセスを限定するために使用することができます。

Traffic Managerでは、OCSP (オンライン証明書ステータスプロトコル)を使用したクライアント証明書の確認も可能です。OCSPは証明書失効リスト (CRL)に代わるもので、Traffic ManagerはOCSPを使用してクライアント証明書の失効ステータスを取得することができます。TLSコネクションを行っているクライアントは、TLSハンドシェイクの一部としてサーバの証明書のためのステータス情報を提供するように、仮想サーバにリクエストすることができます。OCSP Staplingを有効にすると、Traffic Managerに必要なOCSPレスポンスを読み込ませ、そのレスポンスをハンドシェイクメッセージに含めるように指示することができます。

Traffic ManagerのSSL機能については、[SSL暗号化](#)で詳しく説明します。

SSLパススルートラフィックの復号化

これまでに説明したとおり、仮想サーバはプロトコル値として、Traffic Managerが管理するSSL復号化などの変換実行後の内部プロトコルを使用します。

仮想サーバのプロトコル値が「SSL」に設定されている場合、仮想サーバがSSLパススルーモードでSSLトラフィックの転送のみを行うことを示します。SSL復号化を設定するには、まずプロトコル値を、内部プロトコルを表す正しい値(例: HTTP)に変更する必要があります。この場合、プールから、SSL暗号化トラフィックを想定しているノードにトラフィックが送信される可能性が高いため、プールでSSL暗号化を設定することも必要です。

「SSL Decrypt a Service (サービスのSSL復号)」ウィザードを使用すると、SSLパススルーサービスを設定し、トラフィックを仮想サーバで復号化してプールで再暗号化することができます。このウィザードについては、[SSL復号化ウィザード](#)で説明します。

SSL復号化がサポートされるのは、一部のプロトコルのみです。UDPベースのプロトコル、すなわちSIPなどSSLで自動的にラップできないプロトコルにSSL復号化を使用することはできません。

サービスプロテクションクラス

「サービスプロテクションクラス」は、サービスをDoS (サービス妨害)やDDoS (分散型サービス妨害)など悪意のある攻撃から保護するために指定する一連の設定です。サービスプロテクションクラスはサービスプロテクションカタログ内に作成でき、以下のような設定を行います。

- 禁止IPと承認IPのリスト。禁止IPアドレスからのコネクションが許可されることはなく、また承認IPアドレスからのコネクションは常に許可されます。
- 1台のマシンまたはマシングループからの接続数に関する制限。
- 任意の1つのIPアドレスからの接続レートに関する制限。
- RFC2396を厳密に遵守しなければならないかなど、HTTPリクエストに関する制限。

サービスプロテクションクラスを仮想サーバに適用するには、該当する**Virtual Servers** (仮想サーバ) > **Edit** (編集) ページの**Classes** (クラス) リンクをクリックします。Service Protection (サービスプロテクション) セクションで、リストからクラスを選択し、**Update** (更新) をクリックします。

サービスプロテクションについては、[サービスプロテクション](#)で詳しく説明します。

帯域制御クラス

「帯域制御クラス」では、仮想サーバがクライアントへの送信データに適用できる帯域制限を定義します。例えば、複数の大容量ダウンロードコネクションに250Kbitsの共有制限帯域クラスを割り当てた場合、これらのコネクションは、この制限を超えて帯域を使用することはできません。

帯域制御クラスを仮想サーバに適用するには、該当する**Virtual Servers** (仮想サーバ) > **Edit** (編集) ページの**Classes** (クラス) リンクをクリックします。Bandwidth Management (帯域制御) セクションで、リストからクラスを選択し、**Update** (更新) をクリックします。これにより、指定された制限共有タイプ(コネクション単位、マシン単位、またはクラス全体)に従って、仮想サーバが管理する全てのトラフィックを、そのクラスに定義された値に制限する設定が有効になります。

帯域制御については、[帯域制御](#)で詳しく説明します。



Traffic Managerの設定によっては、帯域制御を利用できない場合があります。この機能が必要な場合は、ソフトウェアまたはライセンスキーをアップグレードすることにより入手できます。

サービスレベルモニタリングクラス

「サービスレベルモニタリング(SLM)クラス」は、サービスのエンドユーザが受けるサービスレベル(レスポンス時間)をモニタリングするために使用します。SLMクラスでは、目標とするレスポンス時間を定義します。また、許容限度のパーセンテージを定義することもできます。目標レスポンス時間を満たしたリクエストのパーセンテージがこの限度を下回ると、アラートまたはログメッセージが報告されます。

サービスレベルモニタリングクラスを仮想サーバに適用するには、該当する**Virtual Servers** (仮想サーバ) > **Edit** (編集) ページの **Classes** (クラス) リンクをクリックします。Service Level Monitoring (サービスレベルモニタリング) セクションで、リストからクラスを選択し、**Update** (更新) をクリックします。これにより、仮想サーバは、そのサーバが管理する全てのコネクションのレスポンス時間をモニタリングして記録するようになります。また、サービスレベルに問題が生じた場合は、ログメッセージまたはアラートを報告します。

サービスレベルモニタリングについては、[サービスレベルモニタリング](#)で詳しく説明します。

 Traffic Managerの設定によっては、サービスレベルモニタリングを利用できない場合があります。この機能が必要な場合は、ソフトウェアまたはライセンスキーをアップグレードすることにより入手できます。

HTTPコンテンツキャッシング

Traffic ManagerのHTTP仮想サーバは、頻繁にリクエストされるHTTPリソースを検出し、そのリソースがリクエストのたびに変わらない場合はそのコンテンツを記憶することができます。この機能を「コンテンツキャッシング」と言います。

仮想サーバは、以降、同じリソースに対するリクエストを検出すると、そのリクエストを(過負荷状態にある可能性のある)バックエンドサーバに転送せずに、リソースのコンテンツをローカルキャッシュから直接返すことができます。この機能により、バックエンドサーバの負荷が軽減され、HTTPサービスの性能と処理能力が向上します。

Virtual Servers (仮想サーバ) > **Edit** (編集) ページの **Content Caching** (コンテンツキャッシング) リンクをクリックすると、コンテンツキャッシングの設定が表示されます。ここでは、コンテンツキャッシング機能を有効化したり、様々なタイプのコンテンツをキャッシュする時間を指定することができます。コンテンツキャッシングは、HTTP仮想サーバと、HTTPストラフィックを受け入れて復号化する仮想サーバでのみ利用できます。

コンテンツキャッシングについては、[コンテンツキャッシング](#)で詳しく説明します。

 Traffic Managerの設定によっては、HTTPコンテンツキャッシングを利用できない場合があります。この機能が必要な場合は、ソフトウェアまたはライセンスキーをアップグレードすることにより入手できます。

Web Accelerator

Ivanti Web Accelerator (Web Accelerator)は、Webコンテンツの最適化技術を提供します。ロードを高速化するために、HTMLコード、画像、スクリプト、カスタムスタイルシートなどWebページのマークアップと要素を自動的に最適化します。Webページのロード時間を短縮することは、ユーザエクスペリエンスの改善に不可欠です。

可能な場合には、このように最適化された要素の各バージョンが内部的にキャッシュされるので、以降のリクエストへのレスポンスはさらに高速になります。高速化されたWebページは短時間でロードされるため、帯域利用が改善され、データトラフィックとサーバインフラストラクチャについても節約されます。

サービスに対して**Web Accelerator**を有効化するには、**Virtual Servers** (仮想サーバ) > **Edit** (編集)ページのWeb Acceleratorリンクをクリックします。これは、HTTP仮想サーバでのみ使用できる機能です。

Web Acceleratorの設定の詳細については、[Pulse Secure Web Acceleratorの使用](#)を参照してください。



Traffic Managerの設定によっては、Web Accelerator機能を標準として利用できない場合があります。Web Acceleratorが必要な場合は、ソフトウェアまたはライセンスキーのアップグレードについてIvantiサポートにお問い合わせください。

HTTPコンテンツ圧縮

Traffic Managerは、HTTPレスポンスをリモートクライアントに送信するときに圧縮することができます。これにより、帯域使用率を減らし、低速なコネクションを使用したクライアントへの大容量Webページの配信を高速化することができます。

ブラウザによっては、圧縮されたコンテンツを受信できない場合があります。このようなブラウザでは、このことがHTTPリクエストヘッダに指定されます。Traffic Managerは、ブラウザがコンテンツを解凍できる場合のみコンテンツを圧縮します。

また、一部のWebサーバでもコンテンツを圧縮できるため、Traffic Managerでコンテンツを圧縮せずに、この処理をWebサーバに分担させる方が効率的な場合もあります。ただし、Traffic ManagerとWebサーバの両方で圧縮を有効化しても問題は生じません。

Virtual Servers (仮想サーバ) > **Edit** (編集)ページの**Content Compression** (コンテンツ圧縮)リンクをクリックすると、コンテンツ圧縮の設定が表示されます。ここでは、圧縮を有効化するかどうか、圧縮するMIMEファイルタイプ、および圧縮するドキュメントのサイズを選択することができます。

サイトでダウンロード用の大容量ファイルを提供する場合は、リクエストのたびにサーバで圧縮するのではなく、ファイルをあらかじめ圧縮しておくことをお勧めします。

コンテンツ圧縮のコントロール

Traffic Managerは、リクエストに「Accept-Encoding: gzip」ヘッダが含まれている場合に、HTTPレスポンスを圧縮できます。Traffic Managerは、TrafficScriptルールの実行の前後にリクエストを確認し、いずれかのリクエストにこのヘッダが存在している場合にコンテンツを圧縮します。

例えば、リクエストからAccept-Encodingヘッダを削除するようルールを設定できます。この場合、バックエンドサーバがレスポンスを圧縮して送信することはありませんが(このヘッダを検出することがないため)、リモートクライアントが圧縮をサポートしている場合、Traffic Managerはバックエンドサーバからのレスポンスを圧縮します。この手法を使用すると、全ての圧縮タスクをバックエンドサーバからTraffic Managerに任せることができます。

また、TrafficScript関数の`http.compress.enable()`および`http.compress.disable()`を使用して、Traffic Managerがコンテンツを圧縮するかどうかをコントロールすることもできます。

コネクション分析

サービスへのトラフィックのトレーシングおよびデバッグの作業を容易にするために、Traffic Managerでは、仮想サーバによって処理される各コネクションについて十分な詳細情報を表示できます。

仮想サーバでこの機能を有効にするには、**Services** (サービス) > **Virtual Servers** (仮想サーバ) > **Edit** (編集) > **Connection Analytics** (コネクション分析)をクリックします。

以下の表には、仮想サーバで利用できるコネクション分析設定オプションの詳細を示します。

設定オプション	説明
recent_conns!enabled	この設定を有効化すると、この仮想サーバで処理されているコネクションが Activity (アクティビティ) > Connections (コネクション)ページに表示されます。
recent_conns!save_all	この仮想サーバで処理されている全てのコネクションをデフォルトでActivity (アクティビティ) > Connections (コネクション)ページに表示するか、非表示にするかを選択します。さらに、TrafficScriptを使用してこの動作をオーバーライドするかどうかを、個々のコネクションに対して指定します。 デフォルトで全てのコネクションを非表示にするには、「No」(いいえ)に設定します。この場合、TrafficScript関数の<code>recentconns.include()</code>を使用して、選択的にコネクションをルールに含めます。

設定オプション	説明
	デフォルトで全てのコネクションを表示するには、「Yes」(はい)に設定します。この場合、TrafficScript関数のrecentconns.exclude()を使用して、選択的にコネクションをルールから除外します。 これらの関数の詳細については、オンラインヘルプの『TrafficScript Reference』、またはIvantiのWebサイト(www.ivanti.com)からダウンロードできる『Pulse Secure Virtual Traffic Manager: TrafficScript Guide』を参照してください。
request_tracing!enabled	各リクエストおよびレスポンスに関連する内部イベントの詳細データの収集をTraffic Managerに指示するには、この設定を有効にします。このデータには、各リクエストがTraffic Managerによって受信された時間、および各TrafficScriptルールが実行された時間が含まれます。
request_tracing!trace_io	request_tracing!enabledをYes (はい)に設定した場合に、このコネクションに対して個々の読み取りおよび書き込みイベントに関するデータ収集を有効にするには、この設定を使用します。

コネクション分析の表示

Activity (アクティビティ) > **Connections** (コネクション) ページで個々のリクエストの詳細情報を得ることができます。各コネクションの横にある虫眼鏡アイコンをクリックすると、具体的な詳細が表示されます。詳細については、[Activity \(アクティビティ\)](#) > [Connections \(コネクション\)](#) を参照してください。

トランザクションメタデータのエクスポート

Traffic Manager クラスタでは、分析サービスやダッシュボードなどの外部エンドポイントに対するJSONストリームとして、仮想サーバによって処理されるトランザクションのメタデータレコードをエクスポートできます。特定の仮想サーバで分析のエクスポートを無効化または制限して、Traffic Managerによってエクスポートされるデータの全体量を制限するには、以下の設定を使用します。

Traffic Managerでトランザクションメタデータのエクスポートを開始する前に、クラスタで使用する必要がある宛先エンドポイントおよびその他のグローバル設定を事前に設定します。詳細については、[リモートサービスに対する分析のエクスポート](#)を参照してください。

トランザクションメタデータレコードをエクスポートする場合に仮想サーバによって使用されるパラメータを設定するには、**Virtual Server** (仮想サーバ) > **Edit** (編集) > **Connection Analytics** (コネクション分析) > **Transaction Metadata Export** (トランザクションメタデータエクスポート)セクションを使用します。以下の表に、利用可能な設定を示します。

キー	説明
transaction_export!enabled	この仮想サーバによって処理されるトランザクションのメタデータレコードのエクスポートをTraffic Managerに指示します。 トランザクションメタデータをエクスポートするサービスから、エクスポートしないサービスを分離するには、この設定を使用します。スコープ内の全ての仮想サーバで、対応する transaction_export!enabledを確実に「Yes (はい)」に設定します。スコープ外の全ての仮想サーバでは、「No (いいえ)」に設定します。 Traffic Managerには、全ての個々の仮想サーバ設定をオーバーライドするグローバルtransaction_export!enabledキーがあります。クラスタ全体で分析エクスポートを全て無効にするには、このグローバルキーを使用します。詳細については、トランザクションメタデータレコードのエクスポート設定を参照してください。
transaction_export!brief	トランザクションメタデータレコードに対して「brief (概要)」モードを有効にします。 トランザクションメタデータのJSONスキーマ(System (システム) > Analytics Export (分析エクスポート) > Transaction Metadata (トランザクションメタデータ)のリンクを参照)には、Traffic Managerによってエクスポートされる各レコードに含まれるデータの全セットが示されています。レコード全体には、HTTP リクエストおよびHTTPレスポンスのヘッダデータなどの詳細な情報と、ネゴシエーションが行われた場合はTLSセッションに関する詳細情報が含まれます。各メタデータレコードでエクスポートされるデータ量を減らすには、brief (概要)モードを使用して、制限付きの一連のプロパティのみを提供するようにTraffic Managerを設定します。

キー	説明
	transaction_export!briefを無効にした場合、request_tracing!enabledとrequest_tracing!trace_ioがそれぞれ有効になっている限り、エクスポートされるメタデータにはリクエストトレーシングとI/Oトレーシングの情報が含まれます。 transaction_export!briefを有効にした場合、リクエストトレーシング情報は、常にエクスポートされるメタデータから除外されます。
transaction_export!http_header_blacklist	Traffic Managerがエクスポートから除外するHTTPヘッダのリスト。 この設定を使用して、パスワードなどの機密性のある情報を含む可能性があるヘッダを除外するか、分析のためにヘッダが必要ない場合にエクスポートされるデータサイズを減らします。 もっと複雑なフィルタリングまたは条件付きのフィルタリングを行うには、TrafficScriptを使用して、記録する前にヘッダを変更するトランザクション完了ルールを作成します。詳細については、ルールの適用を参照してください。
transaction_export!hi_res	コネクションの時系列に対して高解像度モードを有効にします。 この設定を使用して、Traffic Managerが、高解像度(マイクロ秒単位)または低解像度(ミリ秒単位)のいずれでコネクションの時系列を記録するかを制御します。 高解像度モードでは、処理時間のより正確な情報が示されますが、処理される各コネクションの待ち時間に少し影響が生じることがあります。

トランザクションメタデータレコードをエクスポートする各仮想化サーバで、この設定を行います。分析するサービスのみにしてデータをエクスポートするには、全ての他の仮想サーバに対して**transaction_export!enabled**を無効にします。

各仮想サーバを個々に再設定する必要なく、Traffic Managerクラスタからトランザクションの分析エクスポートを全て無効にするには、**System** (システム) > **Analytics Export** (分析エクスポート) の設定を使用します。詳細については、[トランザクションメタデータレコードのエクスポート設定](#)を参照してください。

Request Logging (リクエストロギング)

このページを使用すると、仮想サーバが処理するリクエストの詳細ログを保存することができます。ログは、RFC 5424で規定されているプロトコルに従って、Traffic Managerのローカルファイル、またはリモートマシンで実行されているsyslogデーモンに書き込むことができます。仮想サーバのリクエストロギングを有効化するには、該当する**Virtual Servers** (仮想サーバ) > **Edit** (編集) ページの**Request Logging** (リクエストロギング) リンクをクリックします。

デフォルトのロギング動作は、**log!save_all**設定を使用して設定することができます。デフォルトでTraffic Managerに全てのコネクションを記録するように指示するには、「Yes (はい)」に設定します。さらに、TrafficScript関数のrequestlog.exclude()をルール内で使用することにより、特定のコネクションをログから除外することができます。デフォルトでTraffic Managerにコネクションを記録しないように指示するには、「No (いいえ)」に設定します。さらに、TrafficScript関数のrequestlog.include()をルール内で使用することにより、特定のコネクションを選択してログに含まれるようにすることができます。**Update** (更新) をクリックすると、設定が適用されます。

ファイルへのリクエストロギング

ファイル名と、関係するIPアドレス、関係するプールとノード、送受信されたバイト数などの詳細を記述するログエントリフォーマットを指定できます。リクエストされたURL、指定されたヘッダの値、HTTPメソッドなど、HTTP固有のオプションも設定できます。



他のログとは異なり、リクエストログはクラスタ全体で同期されず、各Traffic Managerがそれぞれ独自のログファイルを保持します。

ファイルにリクエストを記録するには、**log!enabled**を「Yes (はい)」に設定し、ログ用のファイル名を指定します。**Update** (更新) をクリックすると、設定が適用されます。

各ログが大きくなり過ぎないように、ログを自動的にローテーションさせることができます。そのためには、ファイル名で%{Time-String}tマクロを使用します。このマクロは、現在の時刻または日付に展開されます。

例えば、%{Y%m%d}tマクロは20161020 (2016年10月20日の場合)のような値に展開されます。これにより、ログファイルは、マクロの値が変化する午前0時に自動的にローテーションするようになります。

Traffic Managerの仮想アプライアンスおよびクラウドインスタンスでは、ログファイルの命名とローテーションは、以下のように自動的に処理されます。

- ログファイルは1時間ごとにチェックされます。
- 1 GBに達したファイルはアーカイブされます。
- 現在の全ログファイルの累積サイズが6 GBに達すると、サイズが最も大きいログから順に、累積サイズが6 GBを下回るまでアーカイブされます。

- 残りのログは毎日アーカイブされます。

ログディレクトリが容量の85%に達すると、Traffic Managerは、最も大きいディレクトリにある最も古いファイルから順に削除を開始します。

リクエストログの表示

View Request Logs (リクエストログ表示)リンクをクリックすると、リクエストログをリアルタイムに表示することができます。この操作により、**Activity** (アクティビティ) > **View Logs** (ログ表示)ページにリダイレクトされます。このページについては、[Activity \(アクティビティ\)](#) > [View Logs \(ログ表示\)](#)で詳しく説明します。

リモートリクエストロギング

Traffic Managerは、ローカルディスク上のファイルではなく、リモートsyslogサーバにリクエストを記録することができます。同一クラスタ内の全てのTraffic Managerは、同じsyslogサーバにトラフィックを記録します。

リモートロギングを設定するには、**syslog!enabled**を「Yes (はい)」に設定し、エンドポイント(IPアドレスまたはホストと、オプションでポート)を指定し、使用するログファイルフォーマットを選択します。**Update** (更新)をクリックすると、設定が適用されます。

syslogプロトコルではUDPが使用されますが、その信頼性は保証されていません。通信量の多いネットワークやデータ損失を伴うネットワークでトラフィックレベルが特に高いときは、ログレコードが破棄される可能性があります。

リクエストロギングのコントロール

リクエストは、選択的に記録させることができます。TrafficScript関数のrequest.setLogEnabled()を使用すると、個々のリクエストのロギングを有効化または無効化することができます。ロギングを有効化すると、**Request Logging** (リクエストロギング)ページの設定に基づいて、リクエストをどのように記録するか(ローカルログファイルまたはリモートsyslogサーバ)が決定されます。

プロトコル設定

Virtual Servers (仮想サーバ) > **Edit** (編集) > **Protocol Settings** (プロトコル設定) ページでは、リモートクライアントと仮想サーバ間の接続を管理するための詳細設定を行うことができます。これには以下のような設定があります。

タイプ	説明
HTTP設定	keepalive設定、およびTraffic Managerが「X-Forwarded-For」、「X-Forwarded-Proto」、「X-Cluster-Client-Ip」などのヘッダをリクエストに追加するかどうかを設定します。

タイプ	説明
クッキー設定	Traffic Managerがどのようにクッキーを処理するかを指定します。
RTSP固有の設定	RTSPで使用する特定のポート範囲を定義できます(RTSP仮想サーバのみ)。詳細については、 Real Time Streaming Protocol を参照してください。
SIP設定	Traffic ManagerがどのようにSIPトラフィックを処理するかを指定します(SIP仮想サーバのみ)。詳細については、 Session Initiation Protocol を参照してください。
ロケーションヘッダ設定	不正なプロトコル、ホスト名、またはポートを使用してノードが301 (永続的移動)または302 (一時的リダイレクト)ステータスコードを返した場合に、URLをリダイレクトすることができます(HTTP仮想サーバのみ)。
FTP設定	FTPデータ接続のセキュリティ設定とポート範囲を設定します(FTP仮想サーバのみ)。
UDP設定	タイムアウトと、1つの接続に対して想定される最大レスポンスデータグラム数を設定します(UDP仮想サーバのみ)。
タイムアウト設定	新しい接続と確立済みの接続のタイムアウトを設定します。
エラー処理	仮想サーバが、処理している接続上のエラーに対応する方法を定義できます。
メモリ制限	クライアントが送信するデータとサーバが返すデータの最大バッファサイズを指定します。
TCP接続設定	仮想サーバによって許可される同時TCP接続の最大数を設定するには、max_concurrent_connectionsを使用します。設定した上限に達した場合、Traffic Managerでは、同時TCP接続の数が指定された上限より少なくなるまで、仮想サーバに対する新しい接続が受け入れられなくなります。 Traffic Managerが、接続に接頭辞としてPROXYプロトコルヘッダが付くことを予期している場合は、proxy_protocolを有効化します。TrafficScriptを使用して、PROXYヘッダに含まれる情報を検査します。有効なPROXYプロトコルヘッダが接頭辞として付けられていない接続は破棄されます。 Traffic Managerは、バージョン1および2のPROXYプロトコルをサポートしています。

タイプ	説明
	close_with_rstを有効化すると、FIN (終了)パケットではなくRST (リセット)パケットによってクライアントからのコネクションを切断するようにTraffic Managerに指示することができます。これにより、TIME_WAIT状態を回避して、まれな状況で、行き場を失った重複パケットを安全に無視することができます。また、ピアによるデータ受信も完全に確保することができます。一般に、TCPコネクションはピアへのFINパケットおよび返答としての確認応答FINパケットによって終了するため、TCPの状態遷移の意味を十分に理解している場合に限り、このオプションを有効化してください。このメカニズムによってデータ送信者は、キューイングされた全てのデータがピアによって読み込まれたことを確認することができます。これは、RSTパケットによって切断するコネクションでは起こりえません。信頼性のないネットワークでは、送信されたデータが送信中に失われることがあります。TIME_WAIT状態の目的は、<A>:<port P>から:<port Q>へのアドレスを使用する1つのTCPコネクションからの行き場を失った遅延パケットが、同じアドレスを使用するその後のTCPコネクションによって受け入れられないようにすることです。すなわち、1つ目のコネクションからの重複パケットがネットワーク内で遅延し、そのシーケンス番号が2つ目のコネクションのウィンドウに存在するときに、そのパケットが2つ目のアドレスにたどり着いた場合、カーネル内のネットワークスタックは、遅延パケットに1つ目のコネクションからのデータが含まれていることを特定することができず、データの破損を引き起こす可能性があります。カーネルは使用中のポート番号を増加させるため、このようなことはほとんど起こりませんが、その可能性はゼロではありません。 proxy_closeを有効化すると、クライアントFINをバックエンドサーバに送信し、ただちにコネクションを切断せずにサーバのレスポンスを待機させることができます。 so_nagleを有効化すると、ネットワーク経由で送信する必要のあるパケット数を減らすことにより、TCPコネクションの効率を向上させることができます。
透過プロキシ	この仮想サーバでバインドされたソケットに対して透過プロキシを有効化するには、「transparent (透過)」をYes (はい)に設定します。この設定では、ファイアウォールルールを使用して、宛先アドレスを変更せずに、選択したトラフィックを仮想サーバに送信できます。

タイプ	説明
	TrafficScriptリクエストルールでは、「request.getDestIp」または「request.getDestPort」を使用してリクエストの宛先IPアドレスまたはポートを確認することにより、実行される処理を判断できます。次に、「pool.select」を使用して元の宛先に対して、フォワードプロキシを実行できます。

特定のプロトコルに固有の設定は、仮想サーバがそのプロトコルを管理している場合のみ表示されます。

通常、これらの設定を変更する必要はありません。デフォルト値は、幅広いシステムで良好な性能が得られるよう選択されています。

MIMEタイプ自動検出

MIMEタイプはHTTPレスポンスのコンポーネントで、画像、HTMLページなど、レスポンス内のデータのタイプを記述します。これにより、クライアントはそのレスポンスをどのようにレンダリングまたは処理するかを判断できます。

WebサーバやWebアプリケーションの中には、MIMEタイプレスポンスヘッダを正しく設定しないものがありますが、多くのクライアントソフトウェアは、この間違いを検出し、MIMEタイプ自体を自動検出することができます。

状況によっては、Traffic ManagerでもレスポンスのMIMEタイプを認識する必要があります。例えば、Traffic Managerは、特定タイプのHTTPレスポンスデータのみ圧縮しようとしますが、MIMEタイプが欠落しているか正しく指定されていないと、Traffic Managerのコンテンツ圧縮は正しく機能しません。

この状況では、**Virtual Servers** (仮想サーバ) > **Edit** (編集) > **Protocol Settings** (プロトコル設定) > **HTTP Specific Settings** (HTTP固有の設定)をクリックし、**mime!detect**を有効にして、MIMEタイプの自動検出を実行します。Traffic Managerは、一連の経験則を適用してレスポンスデータのMIMEタイプを推測します。自動検出は、以下の全ての条件に当てはまる場合に実行されます。

- レスポンスボディが存在する。
- Content-Typeヘッダが欠落している、またはヘッダと**mime!default**の値が一致する。
- レスポンスが圧縮されていない。

チャンク転送エンコーディングが使用されている場合、Traffic Managerは、必要に応じてデータをデチャンクします。

MIMEタイプ自動検出を使用すると、Traffic Managerシステムの性能が低下するため、絶対に必要な場合のみ設定してください。

ロケーションヘッダ設定

この設定では、301 (オブジェクトが永続的に移動)および302 (オブジェクトが一時的に移動)という、Webサーバから送信される最も一般的なURLリダイレクションレスポンスを書き換えることができます。

- **location!regex:** 想定されるノードからのレスポンスに一致する正規表現を入力します(301/302リダイレクション)。(.)などの汎用文字を使用して、表現の一部に一致させることができます。
- **location!replace:** \$1～\$9を使用して正規表現を入力し、レスポンスを別のオブジェクトにリダイレクトします。
- **location!rewrite:** ノードからのレスポンスが**location!regex**に指定した表現と一致しなかった場合の希望レスポンスを選択します。



正規表現を使用する前に、[正規表現の使用](#)で説明するセキュリティに関する注意事項を確認しておいてください。

例えば、以下の値を使用するとします。

- location!regex= (.*)www.example.com/products/(.*)
- location!replace= \$1products.example.com/\$2

ここで、Webサーバが以下のようなレスポンスを返したとします。

```
HTTP/1.1 302 Moved Temporarily
Location: http://www.example.com/products/hacksaw.htm
```

Traffic Managerは、これを以下のように書き換えます。

```
HTTP/1.1 302 Moved Temporarily
Location: http://products.example.com/hacksaw.htm
```

エラーの処理

仮想サーバで生じたコネクションエラーのログギングを有効にするには、**Virtual Servers** (仮想サーバ) > **Edit** (編集) > **Error Logging** (エラーログギング)をクリックします。

コネクションエラーは、以下のような様々な理由で発生します。

- 選択されたプール内のどのノードも使用できない。

- バックエンドサーバとのコネクションがタイムアウトになり、再試行できなかった([失敗したリクエストの再試行](#)を参照)。
- サーバ側またはクライアント側のコネクションでSSLプロトコルエラーが発生した。
- バックエンドサーバから有効なレスポンスが返されなかった(例: 不正形式のHTTPレスポンス)。

クライアントコネクションエラーは、クライアントのアクションや動作が原因である可能性が高く、その多くはTraffic Managerがコントロールすることができません。サーバコネクションエラーは、Traffic Managerからバックエンドサーバへのコネクションに関連しているため、Traffic Managerとバックエンドサーバの設定の影響を受けている可能性が高く、これらの設定によって対処することができます。

一般に、コネクションエラーは、パッシブモニタリングによって検出されるバックエンドサーバでのエラーに関連していない限り、記録されません。

以下のエラーを記録させると、デバッグに非常に役立つことがあります。

- **log!client_connection_failures:** この設定により、エラーログ、またはTraffic Managerのイベント処理設定で指定されている場合は別の場所へのクライアントコネクションエラーの詳細ロギングが有効化されます。
- **log!server_connection_errors:** この設定により、サーバコネクションエラーと、Traffic Managerのあらゆる内部障害の詳細ロギングが有効化されます(例: **max_instr**制限を超えたことによるTrafficScriptルールの中断)。
- **log!ssl_failures:** この設定により、クライアントおよびサーバとのSSLネゴシエーション中に発生した障害のロギングが有効になります。

詳細については、[トラブルシューティング](#)を参照してください。

カスタムエラーメッセージの返信

Traffic Managerがリクエストに対する有効なレスポンスを得られず、クライアントコネクションがまだ存続している場合、Traffic Managerはそのクライアントコネクションを切断するか、(HTTPのみ)以下のエラーをクライアントに返します。

```
Service Unavailable
The service is temporarily unavailable.Please try again later.
```

このエラーメッセージは、「conf/extra」リソースフォルダにアップロードされているファイルのコンテンツを使用して置き換えることができます。エラーファイルを選択するには、**Virtual Servers** (仮想サーバ) > **Edit** (編集) > **Protocol Settings** (プロトコル設定) > **Error Handling** (エラー処理) ページで **error_file** 設定を使用します。

ファイルを「conf/extra」リソースフォルダにアップロードするには、**Catalog** (カタログ) > **Extra Files** (追加ファイル) > **Miscellaneous Files** (その他のファイル)設定ページを使用します。

このファイルの処理方法について明確な指示を設定したい場合、Traffic Managerは、実際のファイルタイプに関わらず、エラーファイル内のHTTPヘッダを処理できます。例えば、GIFまたはJPG画像ファイルなどの特定タイプのファイルを表示する場合は、「Content-Type」ヘッダを設定できます。この場合、バイナリファイルの先頭に以下の行を挿入します。

```
Content-Type: image/gif
```

この機能を拡張して、以下の例のようにHTTPレスポンスコードをオーバーライドすることもできます。

```
HTTP/1.1 200 OK
```

この場合、Traffic Managerにより、目的のコードがレスポンスで強制的にクライアントへ返されます。

 ヘッダとファイルの実際の内容との間には、必ず空白行を挿入してください。

コネクション用メモリ制限

Traffic Managerには、仮想サーバが各コネクションに対して使用できるメモリ量を制限できる設定があります。

- **max_client_buffer**: Traffic Managerが(サーバに送信するために)クライアントから受信するデータ量のうち、バッファ処理される量を制限します。設定された制限値に達すると、コネクションの状態に応じて、Traffic Managerは様々な方法でレスポンスを返します。
- Traffic ManagerがHTTPヘッダで読み込みを行っている場合、413 HTTPエラーコードがクライアントに送信され、警告が記録されます。
- Traffic Managerがリクエストルールを実行している場合、バッファ処理が継続され、警告が記録されます。
- Traffic Managerがサーバへのデータストリーミング段階に達している(リクエストルールの処理が終了している)場合、フロー制御が適用されます。この場合、クライアント側のコネクションが一時停止されるため、Traffic Managerは、バッファ処理されたデータの一部をサーバにプッシュできるようにするまでは、クライアントからのデータをそれ以上読み込みません。
- **max_server_buffer**: Traffic Managerが(クライアントに送信するために)サーバから受信するデータ量のうち、バッファ処理される量を制限します。設定された制限値に達すると、コネクションの状態に応じて、Traffic Managerは様々な方法でレスポンスを返します。

- Traffic Managerがレスポンスヘッダで読み込みを行っている場合、500 HTTPエラーコードがクライアントに送信され、警告が記録されます。
- Traffic Managerがレスポンスルールを実行している場合、バッファ処理が継続され、警告が記録されます。
- Traffic Managerがクライアントへのレスポンスのストリーミングを開始している(レスポンスルールの処理が完了している)場合、フロー制御が適用されます。この場合、サーバ側のコネクションが一時停止されるため、Traffic Managerは、バッファ処理されたデータの一部をクライアントにプッシュできるようになるまでは、サーバからのデータをそれ以上読み込みません。



両方の制限が同時にアクティブになることがあります。Traffic Managerがサーバへのリクエストボディのストリーミング段階に達しており、かつサーバが、完全なリクエストボディを読み込む前にレスポンスボディのストリーミングを開始した場合、クライアントとサーバ両方の送信が速いにもかかわらず読み込みが遅い場合、Traffic Managerは両方のフロー制御メカニズムを同時にアクティブにすることがあります。

プール

本章では、バックエンドノードの論理グループを表すために使用される設定オブジェクトであるプールの概念について説明します。

はじめに

仮想サーバがプールにリクエストを割り当てると、プールはそれらのリクエストを多数の定義済みのサーバノードに負荷分散します。プール内の各ノードは、指定されたポートで、仮想サーバのプロトコルを使用してリクエストを受信できなければなりません。

各プールは負荷分散を行うほか、セッション維持、トラフィックのSSL暗号化、およびヘルスマonitoringに関する設定をそれぞれ保持しています。

プール設定にアクセスするには、**Services** (サービス) > **Pools** (プール)をクリックします。このページを使用すると、既存のプールの設定を変更したり、新しいプールを追加したりすることができます。

プールを作成するときは、以下のカテゴリの1つを選択します。

- **Static** (静的): 管理UIまたはTraffic Manager APIの1つを使用して、管理者がノードのリストを宣言および管理します。このプールのノードリストは、管理者が変更しない限り変化しません。
- **Dynamic** (動的): オートスケーリングまたはプラグインサービスディスカバリスクリプトの使用により、ノードのリストが自動的に生成されて管理されます。オートスケーリングの詳細については、[オートスケーリング](#)または[DNSを用いたオートスケーリング](#)を参照してください。サービスディスカバリプラグインの詳細については、[サービスディスカバリ](#)を参照してください。

各ノードは、サーバ名またはIPアドレスとポートの組み合わせで識別されます。有効なノードエントリの例は以下のとおりです。

- server1.mysite.com:80
- 192.0.2.1:1234
- [2001:DB8:1b6c:6402:250:56ff:fea6:3f43]:4444

IPv6アドレスとしてノードを指定するには、URIの角括弧表記法の[<IPv6 address>]:<port>を使用します。



Traffic Managerは、ノードに対するIPv6リンクローカルアドレスの使用をサポートしていません。

負荷分散

Traffic Managerでは、リクエストをプール内のノードに分配するための負荷分散アルゴリズムを選択することができます。以下のアルゴリズムがあります。

アルゴリズム	説明
ラウンドロビン	各バックエンドサーバに順々に接続をルーティングします。
加重ラウンドロビン	ラウンドロビンと同様ですが、各ノードには割合の異なるトラフィックが送信されます。各ノードの加重は、入力ボックスで指定する必要があります。
パーセプティブ	各ノードの負荷とレスポンス時間をモニタリングし、トラフィックをどのように分配するのが最善かを予測します。これにより、レスポンス時間を最適化し、どのサーバも過負荷状態にならないようにします。
最小接続	現在の接続数が最小のバックエンドサーバを選択します。
加重最小接続	各サーバの重みを考慮して、現在の接続数が最小のバックエンドサーバを選択します。重みは、入力ボックスで指定できます。
最速レスポンス時間	現在のレスポンス時間が最速のバックエンドサーバにトラフィックを送信します。
ランダムノード	バックエンドサーバをランダムに選択します。

 (Traffic Managerマルチサイトモードのみ)Traffic Manager UIの他のエリアと同様に、ノードの加重を単一ロケーションベースの入力と複数ロケーションベースの入力で切り替えることはできません。ノードをロケーションごとに設定するよう選択した場合は、関連付けられた加重が使用ロケーションごとに自動的に表示されます。

最適な負荷分散手法の選択

一般に、トラフィックが均一の場合は、「最小接続」負荷分散アルゴリズムが最適です。この場合、全てのバックエンドサーバの性能が同じであれば、リクエストによってバックエンドサーバにかかる負荷は同じになります。多くのHTTPサービスはこの状況に該当します。最小接続手法では、一部のリクエストによって生成される負荷が他のリクエストより大きい場合でも(画像検索に対してデータベースの検索など)、リクエストがマシンに均等に分配されますが、各タイプのリクエストが十分にあれば、負荷は非常に効率的に分配されます。加重最小接続はこれを改善したもので、サーバによって処理能力が異なる場合に有効です。重みの大きいサーバは、その重みに比例してより多くの接続を受け取ります。

個々の高負荷のリクエストによって速度が著しく低下し、このようなリクエストが頻繁でない場合、最小接続は適していません。また、サーバごとに処理能力が異なる場合も同様です。「最速レスポンス時間」アルゴリズムでは、現在の性能が最高(レスポンスが最速)のサーバにリクエストが送信されますが、リアクティブアルゴリズム(イベント後に初めて速度低下を認識)であるため、高速なサーバが過負荷状態に陥りやすく、これによってサーバの性能が大きく上下します。

「パーセプティブ」は、「最小接続」と「最速レスポンス時間」の特長を取り入れた設計となっており、トラフィックの性質とサーバの性能に応じて負荷分散方法を変化させます。トラフィックが均一なときは最小接続の特性が強くなり、負荷が大きく増減するときは最速レスポンス時間の特性が強くなります。現在の接続数と最近のレスポンス時間から各サーバの性能の傾向を捉えて予測します。

このアルゴリズムでは、新しいサーバ(または障害状態から復旧したサーバ)に割り当てられるトラフィックは、最初は少なく、その後完全な運用レベルまで徐々に増加します。新しいサーバがプールに追加されると、そのサーバに単一のリクエストを送信して試します。レスポンスを受信すると、新しいサーバの負荷がプール内の他の同等ノードと同じ割合になるまで、そのサーバに送信するリクエスト数を徐々に増やします。このリクエスト数の増加は、サーバのレスポンス速度を考慮しながら行われます。そのため、例えば、少量の静的コンテンツを処理する新しいWebサーバの場合は、最大速度まで急速に増加し、初めて使用するJSPをコンパイルするJavaアプリケーションサーバ(これによって開始時のレスポンスが遅くなる)の場合は、ゆっくり増加します。

最小接続は「パーセプティブ」より単純で決定的であるため、状況が適しているときはこのアルゴリズムを優先的に使用してください。

負荷分散アルゴリズムに関する注意事項

最小接続、最速レスポンス時間、およびパーセプティブは、トラフィックレベルが非常に低いときに予測外の動作を示すことがあります。処理する接続が一度に1つのみの場合、最小接続ではリクエストが分配されません。また、パーセプティブおよび最速レスポンス時間では、レスポンス時間が良好であることが分かっているノードに優先的にリクエストが送信される傾向にあり、テストされていないノードは無視されます。

負荷分散のメトリクスは、クラスタ内のTraffic Manager間で共有されません。例えば、2台のTraffic Managerがラウンドロビンアルゴリズムでリクエストを分配する場合、それぞれのサーバが独立して、ノードに順々にリクエストを送信します。

トラフィック管理ソフトウェアをマルチコアサーバで運用する場合、多くの負荷分散メトリクスがプロセス(CPUコア)間で共有されます。ただし、レスポンス時間情報は例外で、コア間で共有されません。

バックエンドサーバをテストし、トラフィックが全てのバックエンドサーバに送信されていることを確認する(負荷分散よりもリクエストの分配を重視する)場合は、テストトラフィックに対して「ラウンドロビン」または「ランダム」を使用してください。

局所認識リクエスト分散(LARD)

パーセプティブ、最小接続、および最速レスポンス時間の各アルゴリズムでは、LARD (局所認識リクエスト分散)と呼ばれる手法によって、同じリクエストを同じバックエンドサーバに送信しようとします。この手法ではサーバのキャッシングが利用されます。すなわち、サーバが特定項目のコンテンツを返す場合、そのコンテンツは内部キャッシュ(メモリまたはディスク)に保存されるため、同じコンテンツであれば、以降のリクエストに対してコンテンツを迅速に提供できる可能性が高くなります。

各アルゴリズムでは、負荷分散に関する決定を行うときに、以前に同じリクエストを処理したノードに関する情報が考慮されます。

- 最小接続では、接続数が最小のノードが複数ある場合、これらの候補ノードのうち、以前同じリクエストを処理したことのあるノードが選択されます。
- パーセプティブおよび最速レスポンス時間アルゴリズムでは、内部選択時に、以前同じリクエストを処理したことのあるノードにさらにわずかな重みが付加されます。

局所認識リクエスト分散は、バックエンドノードへのリクエストのルーティング方法についてTraffic Managerにアドバイスする軽量の方法です。同じURLに対するリクエストが同じバックエンドサーバに送信されるようにするには、ユニバーサルセッション維持(URLでキー設定) ([セッション維持](#))を使用して、さらに強制的な方法でリクエスト送信を行ってください。または、TrafficScriptとネームドノードセッション維持、フォワードプロキシ、またはプール選択手法を使用すれば、リクエストのルーティングを完全にコントロールできます。

セッション維持

セッション維持は、同じクライアントセッションによる全てのリクエストを同じバックエンドサーバに送信するプロセスです。任意のTCPまたはUDPプロトコルで使用できます。

静的Webコンテンツを処理するプールは、通常、セッション維持を必要としません。これは、特定のクライアントに各ページまたは画像を異なるマシンから提供しても、悪影響はないからです。一方、オンラインショッピングサイトを処理するプールでは、セッション維持を使用すると、ユーザのリクエストを、そのユーザの買い物かごの詳細が保持されているノードに必ず送信するようにできます。

Traffic Managerでは、いくつかの方法で同じセッションに属するリクエストを識別することができます。様々なクッキーを使用したり、一定のルールに基づいてセッションを維持したり、またはクライアントのIPアドレスによってセッションを識別したりすることもできます。受信トラフィックがSSLで暗号化されている場合は、SSLセッションIDを利用できます。

セッション維持が失われたときの対応を選択することができます。セッション維持が失われる原因には、無効なセッションデータや、処理を行っているノードでの障害発生などがあります。この場合、コネクションを切断するか、リクエストを新しいノードに送信するか、ユーザをエラーページなどの指定URLにリダイレクトするかを選択できます。

セッション維持をプールに適用するには、そのプールの**Pools** (プール) > **Edit** (編集) ページの **Session Persistence** (セッション維持) リンクをクリックします。セッション維持クラスを選択し、**Update** (更新) ボタンをクリックします。

他のプールで既に使用されているセッション維持クラスを使用するときは注意が必要です。セッション維持クラスは、このプール内のノードが、そのクラスを既に使用しているプールのノードと一致している場合のみ再利用してください。それぞれノードが異なる複数のプールで同じセッション維持クラスを使用した場合、セッションアフィニティを保証できません。

セッション維持については、[セッション維持](#)で詳しく説明します。

帯域制御

Traffic Managerは、バックエンドノードへのコネクションに使用する帯域に制限を適用することができます。これらの制限は、バックエンドノードがリモートデータセンタに存在し、そのリモートサイトへの帯域を管理する必要がある場合に役立ちます。

帯域制限をプールに適用するには、そのプールの**Pools** (プール) > **Edit** (編集) ページの **Bandwidth Management** (帯域制御) リンクをクリックします。帯域制御クラスを選択し、**Update** (更新) ボタンをクリックします。



帯域制限は、Traffic Managerシステムからリモートサーバノードに送信されるデータにのみ適用されます。リモートノードからTraffic Managerに戻される帯域には適用されません。帯域制御については、[帯域制御](#)で詳しく説明します。



Traffic Managerの設定によっては、帯域制御を利用できない場合があります。この機能が必要な場合は、ソフトウェアまたはライセンスキーをアップグレードすることにより入手できます。

ヘルスモニタリング

Traffic Managerは、ノードに送信したリクエストに対するレスポンスを受信しない場合、そのノードで障害が発生したものと見なします。この場合、そのノードへのリクエスト送信を停止し、プール内の残りのノードにトラフィックを負荷分散します。

Traffic Managerの「ヘルスモニタ」では、バックエンドサーバノードに対して高度なヘルスチェックを実行し、ノードが正常に動作しているかどうかを判断することができます。

ヘルスモニタがプールに割り当てられている場合、ヘルスモニタによってプール内の各ノードがチェックされます。特定数の障害が検出された場合、Traffic Managerは、そのノードまたはプールが使用不可であると見なします。これらのヘルステストは、Traffic Managerがノードにリクエストを送信し、ノードからのレスポンスを読み取るに行われる接続テストに加えて実行されます。

ヘルスモニタでは、各ノードに対するPingの実行などの単純なテストから、該当するポートが開いていてノードがホームページなど指定されたデータを処理できるかどうかをチェックする高度なテストまで、幅広いテストを実行できます。特定のタスク(POP3ログインの検証など)を実行するための設定済みのモニタが多数用意されているほか、必要な任意のテストを実行するためのカスタムモニタを作成することもできます。

モニタはノード単位とプール全体の2つのカテゴリに分かれます。ノード単位のモニタは、プール内の各ノードのヘルスをテストします。プール全体のモニタは、プール全体のヘルスに影響を与える特定のマシンに対してテストを実行します。例えば、メールサーバプールがデータをNFSサーバに保存し、各バックエンドサーバがそのNFSサーバにアクセスしている場合、プール全体のモニタではこのNFSサーバをテストできます。このサーバで障害が発生すると、どのバックエンドサーバもデータを読み込むことができないため、プール全体で障害が発生したものと見なされます。

ヘルスモニタをプールに割り当てるには、**Pools** (プール) > **Edit** (編集) ページの **Health Monitors** (ヘルスモニタ) リンクをクリックします。設定を選択したら、**Update** (更新) をクリックして設定を適用します。

ヘルスモニタについては、[ヘルスモニタリング](#)で詳しく説明します。

SSL暗号化

データをTraffic Managerからバックエンドサーバに送信する前に暗号化したい場合があります。

SSL暗号化設定を指定するには、**Pools** (プール) > **Edit** (編集) ページの **SSL Settings** (SSL設定) リンクをクリックします。設定を選択したら、**Update** (更新) をクリックして設定を適用します。

SSLの設定については、[SSL暗号化](#)で詳しく説明します。

プロトコル設定

Traffic Managerとバックエンドノード間のコネクションを管理するための設定を行うには、**Pools** (プール) > **Edit** (編集) > **Protocol Settings** (プロトコル設定)をクリックします。これには以下のような設定があります。

- Traffic Managerからノードへの接続時に、接続元として強制的に特定のIPアドレスを使用するかどうか
- ノードへのHTTP keepaliveコネクションを維持するかどうか
- コネクションを確立するまでの最大回数、またはノードからのレスポンスを待機する最大回数
- 断念してクライアントにエラーを返す前に試行する最大ノード数
- Traffic ManagerからノードへのコネクションをデフォルトのFINパケットではなくRSTパケットによって切断するかどうかこの設定の効果についての詳細は、[プロトコル設定](#)を参照してください。

通常、これらの設定を変更する必要はありません。デフォルト値は、幅広いシステムで良好な性能が得られるよう選択されています。

プールへのリクエストの制限

はじめに

全てのサーバまたはアプリケーションは、特定量のシステムリソースを使用してリクエストを処理します。これらのリソースには、スレッド、ファイル記述子、一定量のメモリなどが含まれます。各バックエンドサーバは、システムパフォーマンスに影響を及ぼす前に、特定の数の同時接続数を処理できる場合があります。さらに、接続内で行われるリクエスト/レスポンスのトランザクションもまた、過度にリソースを消費するタスクを同時処理する際に、パフォーマンスに悪影響を及ぼす場合があります。極端なケースでは、過負荷状態のシステムがリクエストに応え続けようとして、サーバが新しいコネクションをキューイングしたか、破棄したか、または拒否したように見えることがあります。

これらのような場合、サーバノードが受信する同時接続数、またはノードが同時処理できるトランザクション数にハードリミットを設けることで、プールノードが過負荷にならないことを保証しながら、最適なレベルで実行し続けることができます。

接続とトランザクションの制限により、他のTraffic Manager機能に影響が出る場合があります。制限の使用による影響を理解するには、実装前にこちらのセクションを全てお読みください。

制限の適用

Traffic Managerでは、同時接続数やプールノードが同時に処理できるトランザクション数に制限を設けることができます。コネクションベースの制限は、コネクション単位でリソースを消費するアプリケーションを実行する、バックエンドサーバノードの保護に役立ちます。また、Traffic Managerでは、選択したプール内の全てのノードが既に前のトランザクションの処理でビジー状態となっている場合に、受信するHTTPリクエストをキューするためのトランザクション制限を設定することもできます。

コネクションとトランザクションの制限はどちらもプール単位で設定し、そのプール内の各ノードに適用されます。例えば、コネクション制限を50に設定すると、そのプール内の各ノードが同時に受け取る最大接続数は50になります。制限の使用時には、プール内のノードが全て同じ仕様で、同じアプリケーションを提供しているものと見なされます。同一ノードが複数のプールに存在する場合は、そのノードが同時に受け取る可能性のあるコネクション数、またはそのノードにより処理されるトランザクションは、対応する各プールに設定されているレート制限の合計になります。



コネクション制限はTCP接続のみに適用されます。トランザクション制限はHTTPプロトコルを使用しているサービスのみに適用されます。

Traffic Managerクラスタ内のコネクション制限

バックエンドサーバノードへのコネクションは、クラスタ内のTraffic Manager間で共有されません。アクティブ-パッシブTraffic Managerペアでは、全てのコネクションがアクティブTraffic Managerで使用され、パッシブTraffic Managerでは全く使用されません。アクティブ-アクティブ構成では、コネクション制限は事実上2倍になり、アクティブ-アクティブ-アクティブTraffic Managerクラスタでは3倍になります。

リクエストのキュー

負荷分散アルゴリズムによりノードが選択された際、Traffic Managerは同時接続数が設定された制限に到達したかどうかを見つけ出すためにチェックします。同様に、Traffic Managerは、ノードによって処理中のトランザクション数が定義されたトランザクション制限に到達したかどうかをチェックします。どちらも当てはまる場合、Traffic Managerは後続のリクエストをキューに格納し始めます。

リクエストが許容制限を超えた状態が長時間続くと、(例えばHTTPの場合)クライアントへのレスポンスがタイムアウトになる可能性があります。このリスクを軽減するために、TrafficScript関数の`connection.checkLimits()`を使用して、プールでリクエストのキューイングが開始されたときにそれを検出することができます。Traffic Managerはこの情報を使用して、例えばレスポンスがタイムアウトになるまで待たずに、ただちにクライアントに対して別のページをレスポンスとして返すことができます。詳細については、『*Pulse Secure Virtual Traffic Manager: TrafficScript Guide*』を参照してください。

その他のTraffic Manager機能の考慮事項

リクエスト制限は強力な機能で、Traffic Managerの他の機能と相互に影響し合います。このセクションではそれらの影響について説明し、それに基づき実装を計画できるようにします。

キュー管理とSIP

RFC 3261 - SIP: Session Initiation Protocol Section 17.2.1では、SIPサーバのレスポンス時間が100msを超える場合に必要な処理について、以下のように具体的な動作を定義しています。Traffic Managerは、SIP規格のこの部分に従って開発されています。

「サーバトランザクションでは、TUが200ms以内に暫定的または最終的なレスポンスを生成することが分かっている場合を除き、100 (試行中)レスポンスを生成する必要があります。TUが200ms以内に暫定的または最終的なレスポンスを生成することが分かっている場合も、100 (試行中)レスポンスを生成することができます。」
[\[http://www.faqs.org/rfcs/rfc3261.html\]](http://www.faqs.org/rfcs/rfc3261.html)

リクエストレートシェイピング

Traffic Managerのリクエストレートシェイピング機能では、トランザクションとコネクションの制限機能とは異なるキューメカニズムが使用されます。制限を超えると、レートシェイピングクラスからキューを外した後にTraffic Managerがリクエストをキューイングする原因となるため、ノードに送信されるリクエストのレートが、レートシェイピングクラスで指定されたレートと異なる可能性があります。

サービスレベルモニタリング

トランザクションやコネクション制限を超えたためにノードに送信されたリクエストがキューイングされると、サービスレベルモニタリング(SLM)クラスにより測定されるレスポンス時間が影響を受けます。SLMの目的では、リクエストキューに使用された時間は、レスポンス時間の一部としてカウントされます。

プールリクエスト制限の設定

キューの開始前にノードによって処理されるコネクションやトランザクションの最大数の制限を設定するには、以下の手順に従います。

1. **Services (サービス) > Pools (プール) > Edit (編集) > Protocol Settings (プロトコル設定)**をクリックします。
2. コネクション制限は、**max_connections_per_node**を希望の数値に設定します。トランザクション制限は、**max_transactions_per_node**を希望の数値に設定します。
3. ページの下部までスクロールし、**Update (更新)**をクリックします。

制限のオプション

リクエストキューに適用される次の追加のオプションを設定できます。

設定オプション	説明
queue_timeout	キューイングされたリクエストが破棄されるまでの秒数。
max_queue_size	ノードが利用可能となるまでの待機中、キューイングされた状態を維持できるリクエストの数。値を0に指定するとキューのサイズ制限がなくなります。

制限のトラッキング

コネクションとトランザクションの制限ではキューイングが行われるため、リクエストがタイムアウトとなる場合があります。現在のアクティビティグラフを使用すると、キューイングされているリクエスト数と、キューイング中にタイムアウトになったリクエスト数をトラッキングできます。

1. **Activity** (アクティビティ) > **Current Activity** (現在のアクティビティ)ページに移動します。
2. Change data (データ変更)ボタンをクリックします。
3. 選択されている全てのボックスの選択を解除します。
4. Values (値)ツリーのPools (プール)ブランチを展開します。
5. **ConnsQueued**および**QueueTimeouts**に対応するボックスを選択します。
6. リクエスト制限が機能しているかどうかを確認する場合は、Values (値)ツリーのNodes (ノード)ブランチを展開します。
7. **CurrentConn**ボックスを選択します。
8. ページの下部までスクロールします。
9. **New Settings** (新規設定)フィールドに、新しいチャートの名前(Request Limitsなど)を入力します。
10. **New Settings** (新規設定)の横のラジオボタンをクリックします。
11. **Apply** (適用)ボタンをクリックします。

設定した制限のテスト

「zeusbench」ユーティリティを使用すると、設定した制限が正しく機能するかどうかを確認することができます。zeusbenchユーティリティは\$ZEUSHOME/admin/binにあります。同時実行モードでzeusbenchを実行すると、バックエンドノードへの空きコネクションを使い果たすのに十分なトラフィックを生成できます。(例えば) 2つのノードから成るプールがあり、各ノードのコネクション制限を10に設定した場合、以下のzeusbenchコマンドを実行すると、コネクションがキューイングされます。

```
./zeusbench -c 100 -t 60 http://<your site>/index.html
```

この場合、zeusbenchは、(URLで)指定された仮想サーバに対して100のコネクションを確立して、できる限り早く各ノードをダウンさせるために十分なリクエストを送信しようとします。これにより、テスト対象となっているプールに対してこれとほぼ同数のアクティブなコネクションが確立されます。ノード数が多いほど、zeusbenchユーティリティを呼び出すときに指定する必要がある同時接続数が多くなることに注意してください。

バックエンドのフォールトトレランス

Traffic Managerでは、フォールトトレランスを実現するための強力なアプローチが採用されています。トラフィックをできる限り効率よく管理できるよう、様々な方法でプールを設定することができます。

プール内のノードで発生した障害は、Traffic Managerのヘルスマニタによって検出されます。また、負荷分散中には、ノードがリクエストに対するレスポンスを返さないことから、障害を検出することもできます。それ以降、リクエストはそのノードに送信されず、プール内の他のノードに分配されます。障害から復旧したノードは、Traffic Managerが信頼性に問題がないと判断するまで、徐々に元の運用レベルに戻されます。

プールには「障害プール」を関連付けることができます。元のプールの全ノードで障害が発生した場合、リクエストはそのプールを迂回して障害プールに送信されます。

障害プールを設定するには、**Services** (サービス) > **Pools** (プール)をクリックして、変更するプールの名前をクリックします。「Basic Settings (基本設定)」セクションで、ドロップダウンリストから障害プールを選択します。

HTTPサービスの一般的な障害プールは、単一の「sorry server」で構成することができます。このサーバは、受信した全てのリクエストを、「使用不可」メッセージを表示する単純なWebページのリクエストに書き換えることができます。



Traffic Managerでは、プールで障害が発生した場合に、単純なエラーページを直接提供することもできます。カスタムエラーページを設定する方法については、[エラーの処理](#)を参照してください。

プライオリティリスト

プール内にプライオリティリストを設定することができます。プライオリティリストでは、プライオリティに従ってノードをグループ化することができます。トラフィックを常時受信できるようにするマシンの最小数を指定できます。

通常のトラフィックに使用する3台のサーバ(「primary1」、「primary2」および「primary3」と、2台のオフサイトバックアップサーバ(「backup1」および「backup2」)があるとします。要件は、常時少なくとも2台のサーバを利用可能にすることです。

2台のバックアップサーバを3台のプライマリサーバの障害プールとして設定することができます。ただしこの場合、障害プールが使用されるのは、プライマリプールの全てのノードで障害が発生した場合のみです。「primary1」および「primary3」で障害が発生した場合は、「primary2」のみで全てのトラフィックを受信することになります。

この方法の代わりに、グループ化されたプライオリティリストを設定できます。これには、{primary1, primary2, primary3}とその下の{backup1, backup2}の、合わせて2つのプライオリティグループがあります。また、常時少なくとも2台以上のノードを使用可能にするよう指定します。

全てのサーバが稼働している状態では、プライマリサーバ3台が全て使用され、バックアップマシンにトラフィックは送信されません。

primary1で障害が発生した場合、上位グループの2つのノードでリクエストを受信できるため、この2つのノードで全てのトラフィックが処理されます。

ここで、さらにprimary3で障害が発生すると、上位グループで残されたノードは1つのみになります。2台のサーバを使用可能にするよう指定したため、Traffic Managerは、バックアップグループの両方のマシンにリクエストを送信し始め、リクエストはprimary2、backup1、およびbackup2で処理されるようになります。

プライオリティレベルはこれ以上ありません。さらにノードで障害が発生した場合、トラフィックは残されたノードに負荷分散されます。全てのノードで障害が発生した場合は、プールに障害プールが設定されていれば、障害プールにトラフィックが渡されます。

▼ Priority Lists

Priority Lists control the order of preference in which the nodes in a pool are used.

To alter the priorities of the nodes use the up and down buttons.

Enable priority lists

priority!enabled: ☐ Yes ☒ No

Minimum number of highest-priority active nodes

priority!nodes:

Highest Priority Group

Node	Up	Down
primary1:80	▲	▼
primary2:80	▲	▼
primary3:80	▲	▼

Lowest Priority Group

Node	Up	Down
backup1:80	▲	▼
backup2:80	▲	▼

プールのプライオリティグループを設定するには、**Pools** (プール) > **Edit** (編集) ページからそのプールを編集します。**Load Balancing** (負荷分散) をクリックし、「Priority Lists (プライオリティリスト)」セクションを展開します。

プール内のノードが **Highest Priority Group** (最高プライオリティグループ) リストに表示されます。このプールでプライオリティリストを有効化するには、**priority!enabled** を「Yes (はい)」に設定し、**priority!nodes** を継続的に使用可能にするノードの最小数に設定します。次に、各ノードの横にある上下矢印ボタンを使用して、最初のグループの上位または下位に別のグループを作成します。



最上位のグループ内で使用可能なノード数が指定の最小ノード数を下回るまで、下位のプライオリティグループ内のノードは使用されません。使用可能なノード数が最小ノード数を下回ると、その下位にあるグループの全てのノードで処理が開始されます。これらの上位2グループの健全なサーバ数が指定の最小ノード数を下回るまで、これらの2つのグループのサーバが使用され、最小ノード数を下回った時点で、さらにその下位にあるグループ全体が使用されるようになります。これは、全てのプライオリティグループが使用されるまで続きます。

プール内の全てのノードで障害が発生した場合、障害プールが設定されていれば、このプールにリクエストが送信されます。

ノードのドレイン、無効化、削除

負荷分散が行われているサーバファームからいずれかのバックエンドノードを除外したい場合があります。これは、恒久的な措置の場合もあれば、メンテナンスやアップグレードのための一時的な措置である場合もあります。

Traffic Managerが新たなリクエストを送信しないように、ノードを「disable (無効化)」か「drain (ドレイン)」にするかもしくは、プールからノードを完全に削除します。

- **ノードの無効化:** Traffic Managerからそのノードに対してこれ以上コネクションが送信されなくなります。Traffic Managerがノードのモニタリングを停止しても、ノードはプール設定に残ります。詳細については、[ノードの無効化](#)を参照してください。
- **ノードのドレイン:** Traffic Managerからそのノードに対してこれ以上新たなコネクションが送信されなくなりますが、既に確立されているセッションは維持されます。Traffic Managerでは、そのノードが引き続きアクティブにモニタリングされます。詳細については、[ノードのドレイン](#)を参照してください。
- **ノードの削除:** プール設定からノードを完全に削除します。詳細については、[プールからのノードの削除](#)を参照してください。

ノードの無効化、ドレイン、プールからの削除を行うには、**Services** (サービス) > **Pools** (プール) > **Edit** (編集)ページを使用します。目的のノードが含まれるプールを編集し、必要に応じてノードのステータスを「Disabled」(無効)または「Draining」(ドレイン)に変更します。ノードを削除するには、横にある**Delete** (削除)チェックボックスをオンにします。変更を保存するには、**Update** (更新)をクリックします。



この処理では、その特定のプールに対して、ノードがドレイン、無効化、または削除されます。同じノードを使用する他のプールは影響を受けません。

Node (ノード)ウィザードの使用

様々なサービス全体でノードを無効化、ドレイン、再アクティブ化、または削除する処理を簡単に実行できるように、Traffic Managerには様々なウィザードが用意されています。ウィザードにアクセスするには、上部のツールバーにある**Wizards** (ウィザード)メニューを使用します。

- ノードの無効化
- ノードのドレイン
- ノードの再アクティブ化
- ノードの削除

ドレイン/無効化するノードを指定し(選択したウィザードによって異なります)、全てのサービスに対して実行するか、指定したサービスに対してのみ実行するかを選択します。ドレイン/無効化を実施したノードは、後から *Reactivate a node* (ノードの再アクティブ化)ウィザードを使用してアクティブな状態に戻すことができます。サービスのいずれかまたは全てからノードを完全に削除するには、*Remove a node* (ノードの削除)ウィザードを使用します。

ノードの無効化

サービスからノードを一時的に除外する場合は、ノードを無効化します。ノードへの既存の接続は、[ノードの無効化または削除時に既存の接続の動作を変更する](#)で説明している設定に応じて処理されます。この場合、(例えば、サーバを停止したために)ノードの動作が停止しても、Traffic Managerはエラーを報告しません。ノードを構成から除去せずに無効化すると、そのノードが使用可能な状態であれば、無効化したノードを簡単に処理に復帰させることができるというメリットがあります。

ノードのドレイン

既存のセッションを考慮する必要がある場合は、ノードをドレインします。ただし、ノードの動作が停止した場合、Traffic Managerからエラーが報告されることに留意してください。Traffic Managerがリクエストを受信し、セッションを維持するためにそのノードを使用する必要がある場合、Traffic Managerは引き続きそのノードを使用します。ノードを停止する場合は、まずそのノードをドレインし、全てのセッションが完了するまで待ってから無効化してください。

ノードがまだコネクションを処理しているかどうかを判断するには、**Activity** (アクティビティ) > **Draining Nodes** (ノードのドレイン) をクリックします。このページには、ノードがまだ処理している接続数と、最後のコネクションからの経過時間が表示されます。詳細については、[コネクションのドレイン](#)を参照してください。ノードの既存のコネクションが終了した時点で、そのノードを無効化するか、*Remove a Node* (ノード削除)ウィザードを使用してクラスタから完全に削除することができます。

プールからのノードの削除

プールから1つ以上のサーバノードを廃止し、対応する設定を残す必要がない予定の場合は、プールからノードを削除します。ノードへの既存の接続は、[ノードの無効化または削除時に既存の接続の動作を変更する](#)で説明している設定に応じて処理されます。

プールから全てのノードを削除しようとする、管理UIで、プールによって新たなトラフィックが受信できなくなり、この操作を完了するには明示的な同意が必要であるという警告が表示されます。警告に同意するには、**Update** (更新)ボタンの横にある確認チェックボックスをオンにしてもう一度試します。

ノードの無効化または削除時に既存の接続の動作を変更する

無効化または削除したノードへの既存の接続の動作は、**node_delete_behavior**の値に依存します(**Services** (サービス) > **Pools** (プール) > **Edit** (編集) > **Protocol Settings** (プロトコル設定)にあります)。この設定には2つのオプションがあります。

- **All connections to the node are closed immediately** (ノードへの全ての接続を即座に閉じる): Traffic Managerはノードへの接続を即座に終了します。
- **Allow existing connections to the node to finish before deletion** (削除前にノードへの既存の接続の完了を待つ): 既存の接続がある状態で無効化または削除されたノードは、以下のイベントの初回発生時に自動的に無効化または削除されるまで、「ドレイン」に似た状態のままになります。
 - 接続カウントがゼロになる
 - **node_drain_to_delete_timeout** (**Services** (サービス) > **Pools** (プール) > **Edit** (編集) > **Protocol Settings** (プロトコル設定)にあります) の値に達する(ゼロ以外の場合)。

そのようなドレイン状態のノードが、**Activity** (アクティビティ) > **Draining Nodes** (ドレイン状態のノード)リストの「Autodelete (自動削除)」列で「Yes (はい)」の状態で含まれている。

オートスケーリング



オートスケーリングはライセンスキーによって管理される機能であり、Traffic Manager製品によっては利用できない場合があります。サポートされている設定の詳細については、IvantiのWebサイトを参照してください。

はじめに

「アプリケーションオートスケーリング」オプションは、サポートされている仮想またはクラウドプラットフォームで実行中のサービスの性能をモニタリングします。性能が目標サービスレベルを下回った場合、Traffic Managerはオートスケーリングアクションを開始し、プラットフォームに追加のサービスインスタンスをデプロイするよう要求することができます。また、新しいインスタンスが使用可能になった直後から、自動的にトラフィックをそのインスタンスに負荷分散します。オートスケーリング機能は、モニタリング/決定エンジンと、関連するプラットフォームと連携するドライバスクリプトの集合で構成されます。

オートスケーリングはプールのプロパティの1つです。有効化されている場合、プールの設定時に具体的なノードリストを指定する必要がありません。代わりに、性能が低下し始めると、必要とされる追加の処理能力を提供するために、自動的に追加ノードをプールに組み込むことができます。逆に、追加ノードが不要になったときには、プールを元の規模にスケールダウンしてノードを解放できます。このように、この機能を利用すると、トラフィックの一時的な急増や長期的な負荷の増加に動的に対応することができます。

内蔵サービスモニタは、プールの自動スケールアップまたはスケールダウンがいつ必要か判断するために使用されます。プールによって提供されるサービスレベル(レスポンス時間)は綿密にモニタリングされます。レスポンス時間が目標レベルを下回ると、最小のコストでサービスレベルを満たすために、オートスケーリングによってプールのノードが追加または削除され、リソースが増加または減少します。

オートスケーリングの動作

オートスケーリングメカニズムは決定エンジンと、プラットフォーム依存のドライバスクリプトの集合で構成されます。

決定エンジン

決定エンジンは、プールからのレスポンス時間をモニタリングし、スケールアップ/スケールダウンしきい値を提供します。その他のパラメータは、プール内の最大ノード数および最小ノード数と、スケールアップまたはスケールダウン完了後にレスポンス時間が安定するまでTraffic Managerが待機する時間をコントロールします。

例えば、250msというSLAを確保する場合について考えます。このSLA内に完了するトランザクションが50%を下回った場合に、最大10台のバックエンドノードまでスケールアップ(ノードを追加)するようTraffic Managerに指示することができます。または、このSLA内に完了したトランザクションが95%を超えた場合は、最小1台のノードまで段階的にスケールダウン(ノードを削除)するようにします。



プール内の最大ノード数と最小ノード数を編集して、手動でノードをプロビジョニングすることができます。この場合、Traffic Managerは、最大/最小ノード数と実際のアクティブなノード数の不一致を検出すると、一連のスケールアップまたはスケールダウンアクションを開始します。

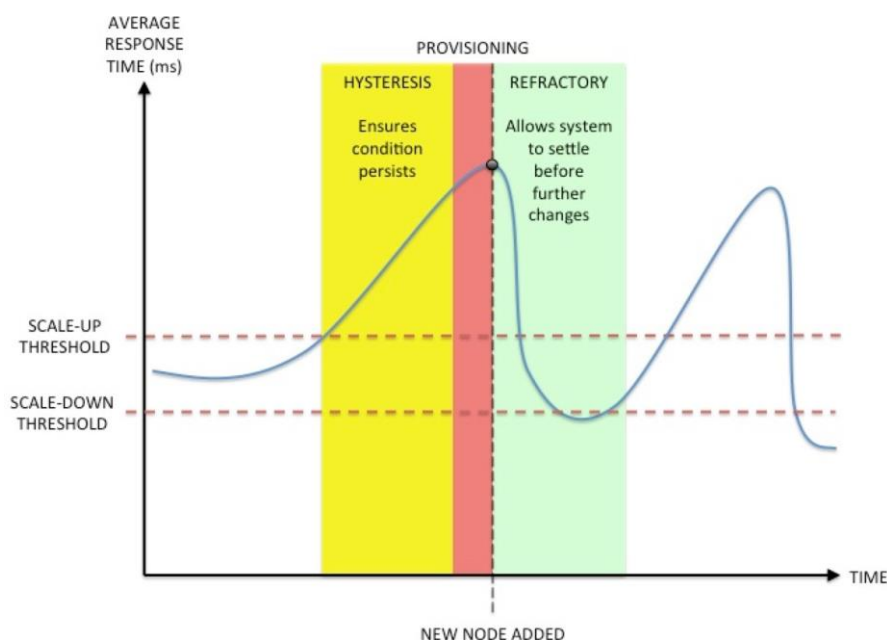
クラウドAPIドライバ

Traffic Managerには、Amazon EC2、Rackspace、およびVMware vSphereクラウド環境用のAPIドライバスクリプトが用意されています。オートスケーリングプールを作成する前に、使用するクラウドAPIに関連する一連のクラウドクレデンシャルを作成する必要があります。これらのクレデンシャルには、Traffic Managerが前述のクラウドプロバイダと通信するために必要な情報が保持されます。使用される具体的なクレデンシャルは、指定するクラウドプロバイダによって異なります。詳細については、[クラウドクレデンシャル](#)を参照してください。

決定エンジンは、設定されたクレデンシャルとパラメータを使用してドライバを呼び出し、スケールアップまたはスケールダウンアクションを開始します。ドライバは、仮想マシンをデプロイまたは削除するよう仮想化レイヤに指示します。このアクションが完了すると、ドライバはプールの新しいノードリストを返し、決定エンジンがプールの設定を更新します。

オートスケーリングイベントの時系列の例

以下の図には、Traffic Managerのオートスケーリングイベントの時系列を示します。「スケールアップしきい値」を超えた時点で、その状況が持続し、一時的な急増ではないことを確認するために、観察期間が始まります。この期間中にプールのパフォーマンスが想定レベルに戻った場合は、Traffic Managerによってオートスケーリングイベントが破棄されます。



観察期間にわたってしきい値を超える状態が持続した後に、Traffic Managerのオートスケーリングメカニズムにより、選択したクラウドまたはハイパーバイザのAPIを使用して、新しいバックエンドサーバノードインスタンスが作成されます(または、図でスケールダウンイベントが示されている場合は、削除されます)。このプロセスでは、最終的に、ノードが正常に作成され、負荷分散プールに追加されます。

プールに新しいサーバノードが追加されると、無反応期間が始まります。この期間中に、Traffic Managerは、スケールしきい値で再評価される前に、追加のオートスケーリングが発生せず、サービスのパフォーマンスが新しいレベルで安定するようにします。

ここで説明したしきい値と期間を設定するには、[オートスケーリングの設定](#)を参照してください。

オートスケーリングの設定

プールにオートスケーリングを実装するためのメカニズムでは、Traffic Managerがプール設定ファイルを自動的に変更(必要に応じてノードを追加または削除するなど)することができます。このようなプールがアクティブに使用されている状況では、この自動更新プロセスと同時にユーザが管理UIまたはTraffic Manager APIから設定を変更した場合、自動更新プロセスによってユーザによる変更が妨げられる可能性がわずかにあります。そのため、変更を行った場合は、プールページを更新して、行った更新が有効になったことを再確認することをお奨めします。

選択したクラウドAPI用に一連のクレデンシャルを作成したら、選択したプールのオートスケーリングプロパティの設定に進むことができます。新しいプールでは、**Dynamic...** (動的)のプールタイプを使用してから、使用可能なオプションのリストから **vTM Autoscaler** (vTMアクセラレータ)を選択します。既存のプールでは、**Pools** (プール) > **Edit** (編集) > **Autoscaling** (オートスケーリング)をクリックすると使用可能な設定オプションが表示されます。

基本設定

設定	説明
autoscale!enabled	オートスケーリングを有効化または無効化します。この機能が有効化されている場合、ノードは、 autoscale!external に指定したメカニズムに従ってプールに追加またはプールから削除されます。
autoscale!external	一部のクラウドプロバイダは、オートスケーリングを実行するための独自のメカニズムを持っています。この場合、クラウドプロバイダはマシングループ(プール内のマシン)の性能をモニタリングし、必要に応じてマシンの追加または削除を行います。その後、Traffic Managerに対して、そのクラウドプロバイダのAPIを使用してマシンを追加または削除したことを通知します。Traffic Managerは、この変更に関する通知を受け取ると、それに従ってオートスケーリングプールのノードを追加または削除します。 クラウドプロバイダのオートスケーリングメカニズムを使用する場合は、このパラメータを「Yes (はい)」に設定してください。Traffic Managerのオートスケーリングメカニズムを使用する場合は、「No」に設定してください。
クラウドクレデンシャル	クラウドのAPIが必要とする一連のクレデンシャルです。一連のクラウドクレデンシャルは、 Catalogs (カタログ) > Cloud Credentials (クラウドクレデンシャル)ページであらかじめ設定することができます。

Traffic Managerは、クラウドプロバイダの必要条件に適用できる一連の設定キーを提示するために、ここで指定されるAPIタイプのクレデンシャルを使用します。

Amazon EC2クラウドの設定

設定	説明
EC2 AMI	新しいノードインスタンスの作成元となる仮想マシンイメージの固有の識別子です。Traffic Managerは、処理能力を上げる必要があると判断すると、新たに作成された各ノードをプールに追加します。Traffic Managerは、クラウド内でこのAMIを使用するインスタンスを検出すると、それらのインスタンスが対応するプールに属するものと見なします。 ここで使用される固有の識別子によって、ノードで実行されるOSとサービスが決まります。EC2クラウドでは、このパラメータはImageIdと呼ばれています。
EC2 Machine Type	プールに追加されるノードのインスタンスに必要とされる、仮想マシンのリソースに関するサイズを示す識別子です。EC2では、このパラメータはInstanceTypeと呼ばれています。 通常、強力なマシンほど高いコストが伴います。
Name Prefix	オートスケーリングイベント中に作成される新規ノードの名前に追加される、オプションのプレフィックスです。 Traffic Managerは、クラウド内でこの名前で始まるインスタンスを検出すると、それらのインスタンスが対応するプールに属するものと見なします。
EC2 Security Group IDs	1つのセキュリティグループで、1つ以上のノードインスタンスに対するトラフィックをコントロールするファイアウォールとして機能します。あるインスタンスが起動されるときに、EC2はそのインスタンスを1つ以上のセキュリティグループに関連付けることができます。この関連付けを行うためには、セキュリティグループIDのリストをExtra Arguments (余剰引数)テキストボックスで指定します(グループ名は使用しません)。EC2-Classicのデプロイでは、EC2-Classic専用で作成したセキュリティグループを使用する必要があります。同様に、EC2-VPCでは、VPC専用で作成したセキュリティグループを使用する必要があります。ここでセキュリティグループが指定されていない場合、EC2はデフォルトのセキュリティグループを適用します。

設定	説明
EC2-VPC Subnet IDs	新たなEC2-VPCインスタンスが起動された場合の、VPCサブネットのIDのリストです。インスタンスは、Extra Arguments (余剰引数)テキストボックスで指定したサブネットに均等に分配されます。EC2-Classic内部でインスタンスを起動するには、このリストを空の状態のままにします。 オートスケーリングされたEC2-VPCノードを可用性の異なるゾーン全体で均等に分配する場合は、可用性の異なるゾーンに存在するサブネットのリストを作成することをお奨めします。
Extra RunInstances Arguments	AWS API機能のRunInstancesに対するキー/値の引数をカンマで区切ったリストで、ノード作成時に使用されます。 RunInstancesについてはAWSに関するドキュメントを参照し、利用可能な引数のリストを確認してください。

Rackspaceクラウドの設定

設定	説明
Image ID	新しいノードインスタンスの作成元となる仮想マシンイメージの固有の内部識別子です。Traffic Managerは、処理能力を上げる必要があると判断すると、新たに作成された各ノードをプールに追加します。ここで使用される固有の識別子によって、ノードで実行されるOSとサービスが決まります。Rackspaceクラウドでは、このパラメータはimageIdと呼ばれています。 ここでは、内部イメージ識別子を使用する必要があります。 特定のクラウドクレデンシャルセットで使用可能なイメージ識別子のリストを取得するには、以下のように組み込みのrackspace.plスクリプトを手動でコマンドラインから実行します。 <pre>\$ZEUSHOME/zxtm/bin/rackspace.pl \ listimageids \ --cloudcreds=<credentials></pre> <credentials>の部分をRackspaceクラウドのクレデンシャルに置き換えます。

設定	説明
Flavor ID	プールに追加されるノードのインスタンスに必要とされる、仮想マシンのリソースに関するサイズを示す内部識別子です。Rackspaceでは、このパラメータはFlavorIdと呼ばれています。 通常、強力なマシンほど高いコストが伴います。 特定のクラウドクレデンシャルセットで使用可能なサイズ識別子のリストを取得するには、以下のように組み込みのrackspace.plスクリプトを手動でコマンドラインから実行します。 <pre>\$ZEUSHOME/zxtm/bin/rackspace.pl \ listsizeids \ --cloudcreds=<credentials> <credentials>の部分をRackspaceクラウドのクレデンシャルに置き換えます。</pre>
Name Prefix	オプションのネームプレフィックスで、オートスケーリングイベントで作成されたクラウドでホストされる各仮想マシンに適用できます。Traffic Managerは、クラウド内でこの名前で始まるインスタンスを検出すると、それらのインスタンスが対応するプールに属するものと見なします。

VMware vSphereの設定

設定	説明
VMware Template	組み込み式vSphereドライバスクリプトは、既存の仮想マシン(VM)テンプレートを基にVMをデプロイして、新しいVMを作成します。これらのテンプレートはあらかじめ設定されているVMで、新しいVMのひな形として使用できるテンプレートであることを示すマークが付けられています。一般に、vCenter Server上のテンプレートのパスは以下の形式です。 <pre><DatacenterName>/vm/<TemplateName></pre>
Name Prefix	vCenter Serverでは、新たに作成されるノードにはこのName Prefix (ネームプレフィックス)が付けられます。同一のvSphereインフラストラクチャ上で複数のプールがオートスケーリングされる場合、各プールのName Prefix (ネームプレフィックス)がユニークになるよう注意してください。

設定	説明
Data Center	vCenter Serverで管理される全てのESXホストは、論理的なData Center (データセンタ)に整理されます。オートスケーリングは、このData Center (データセンタ)内のホストまたはクラスタに対して実行されます。
Data Store	新たに作成された仮想マシンで使用されるData Store (データストア)です。空白のままにすると、(vCenterの設定に従って)そのData Center (データセンタ)のデフォルトデータストアが使用されます。 Data Store (データストア)をここで指定する場合、VMテンプレートで同じData Store (データストア)が使用されていることを確認してください。
VMware ESX Cluster	通常、vCenter Server上のインフラストラクチャの最上位にData Center (データセンタ)があり、ここにESXホストまたはESXホストのクラスタが保持されます。クラスタにDynamic Resource Scheduling (DRS)が設定されている場合、新しい仮想マシンの配置先ホストはvCenterによって決められます。クラスタが構成されていない場合、この設定で、新しいVMを作成するESXホスト名またはIPを指定する必要があります。

ノードの設定

設定	説明
autoscale!ipstouse	(使用IPアドレス)一般に、クラウド環境内のインスタンスでは、同じクラウド内の他のノードと通信するためのプライベートネットワークのIPアドレスと、インターネットと通信するためのパブリックネットワークのIPアドレスの、少なくとも2つのIPアドレスが使用されます。「private」または「public」を選択し、トラフィックを新たなノードのプライベートアドレスに送信するか、パブリックアドレスに送信するかを定義します。 デフォルトでは、Amazon EC2-VPC内のインスタンスにパブリックIPアドレスはありません。作成時にIPアドレスを割り当てるには、「public」を選択してください。
autoscale!port	ノードをプールに追加するときに使用するポートを定義します。このポートは、クラウド環境内に作成されたバックエンドマシンでサービスを監視するポートに対応します。

設定	説明
autoscale!min_nodes	オートスケーリングプールに存在できる最小ノード数を定義します。このノード数には、現在障害発生としてマークされているノードも含まれます。このような状況におけるノードの障害は、レスポンス時間の低下につながる可能性があり、その結果、オートスケーリングによって新しいノードが組み込まれます。ただし、この動作は保証されません。障害が発生したノードが含まれるオートスケーリングプールは、 autoscale!max_nodes で指定した数を超えてオートスケーリングされません。
autoscale!max_nodes	オートスケーリングプールに存在できる最大ノード数を定義します。これにより、プールは莫大な数のノードにオートスケーリングできなくなります。

スケーリングパラメータ

設定	説明
autoscale!response_time	Traffic Managerは、オートスケーリングプール内のノードのレスポンス時間(ミリ秒単位)をモニタリングします。この時間は、Traffic Managerがそのノードへのコネクションを開始してからレスポンスの最初のバイトを受信するまでの時間として測定されます。 autoscale!response_time 未満のレスポンス時間は適合していると見なされ、これより長い時間は不適合と見なされます。Traffic Managerは、適合しているコネクションの割合を継続的に測定します。
autoscale!scaledown_level	適合しているレスポンスの割合が autoscale!hysteresis 秒以上に渡ってこの値を上回ると、プールはスケールダウン(すなわち、プールからノードが削除)されます。
autoscale!scaleup_level	適合しているレスポンスの割合が autoscale!hysteresis 秒以上にわたってこの値を下回ると、プールはスケールアップ(すなわち、プールにノードが追加)されます。

設定	説明
autoscale!refractory	プールサイズの変更(増加または減少)後、Traffic Managerは、 autoscale!refractory 秒待機してから、次の変更を行います。これは、新しいノードがプール全体のレスポンス時間に反映されるまで時間がかかるためです。一度に多数のノードが作成(または廃止)されないようにするために、この時間を増加させることが必要な場合があります。
autoscale!hysteresis	Traffic Managerが変更を開始する前に、オートスケーリング条件が満たされなければならない時間(秒単位)です。これは、(例えば)スケールアップイベントを発生させるために、プールの適合しているコネクションの割合が autoscale!scaleup_level を下回った状態が継続しなければならない時間に相当します。
autoscale!lastnode_idletime	オートスケーリングされたプール内の最後のノードが、廃止される前にアイドル状態でなければならない時間(秒単位)です。 これは、 autoscale!min_nodes が0に設定されている場合にのみ該当します。

DNSを用いたオートスケーリング



この機能はサービスディスカバリプラグインとしても使用可能で、追加のカスタマイズオプションが用意されています。詳細については、[サービスディスカバリ](#)を参照してください。

Traffic Managerは、DNSレコードを使用してノード数を自動的に増減することにより、Traffic Managerによって処理されるトラフィックのリクエストに応えることができます。このメカニズムをDNSを用いたオートスケーリングと言います。

ホスト名に対するDNSレコードが複数のIPアドレスに名前解決され、アドレス数が有効な名前解決に応じて変化する場合、その変化をプール内で使用されるノードに反映させることができます。例えば、backend.example.comは最初に4つのIPv4レコードに名前解決され、続いて6つのIPv4レコードに名前解決されます。Traffic Managerは、新たなIPアドレスが含まれるように、プール内のノードリストを更新します。

Traffic Managerは30秒間隔で更新を実行します。

DNSを用いたオートスケーリングを使用するようにプールを設定するには、**Pools (プール) > Edit (編集) > DNS-derived autoscaling** (DNSを用いたオートスケーリング)をクリックします。以下の設定が適用されます。

設定	説明
dns_autoscale!enabled	プールに対するDNSを用いたオートスケーリングを有効化するには、このキーをYesに設定します。 デフォルトでは、プールはオートスケーリングされません。
dns_autoscale!hostnames	Traffic Managerが定期的にクエリするDNSホスト名のリストです。IPv4 AまたはIPv6 AAAAのレコードは全て、このプールのノードリストに対して追加または削除されます。
dns_autoscale!port	DNSを用いたオートスケーリングを使用して追加された全てのノードにTraffic Managerが追加したポート番号です。

サービスディスカバリ



サービスディスカバリはライセンスキーによって管理される機能であり、Traffic Manager製品によっては利用できない場合があります。サポートされている設定の詳細については、IvantiのWebサイトを参照してください。

Traffic Managerプールには、サービスディスカバリプラグインスクリプトにより入力および管理される、動的ノードリストを含めることができます。この機能を使用すると、外部またはリモートサービスにより、適切なスクリプトまたはプログラムを使用して、プールがオートスケーリングされるように設定できます。



この機能は、Traffic Managerの内蔵オートスケーリング機能に関連しています。ただし、サービスディスカバリによって、プールのノードリストが内部ではなく外部で管理されます。プールのオートスケーリングの詳細については、[オートスケーリング](#)を参照してください。

サービスディスカバリを使用するよう新しいプールを設定するには、プールのタイプを**Dynamic...** (動的)に設定し、ドロップダウンリストから適切なサービスディスカバリプラグインを選択します。

Create a new Pool

Pool Name:

Pool Type: ☐ Static ...
☒ Dynamic ...

☐ Pulse Secure vTM Autoscaler
☒ Service Discovery Plugin ...

Specify any arguments which should be passed to the script:

Monitor:

Create Pool

また、サービスディスカバリプラグインを使用するよう既存の静的プールを変換するには、**Services** (サービス) > **Pools** (プール) > **Edit** (編集) > **Service Discovery** (サービスディスカバリ) をクリックし、**service_discovery!enabled**を「Yes (はい)」に設定します。

▼ Service Discovery Settings

These settings control how the Service Discovery plugin is called.

Are the nodes of this pool determined by a Service Discovery plugin? If yes, nodes will be automatically added and removed from the pool by the traffic manager.

service_discovery!enabled: ☒ Yes ☐ No

The plugin script a Service Discovery autoscaled pool should use to retrieve the list of nodes.

service_discovery!plugin:

The arguments for the script specified in "service_discovery!plugin", e.g. a common instance tag, or name of a managed group of cloud instances.

service_discovery!plugin_args:

The minimum time before rerunning the Service Discovery plugin

service_discovery!interval: seconds

The maximum time a plugin should be allowed to run before timing out. Set to 0 for no timeout.

service_discovery!timeout: seconds

以下の設定を使用すると、プールのサービスディスカバリをさらに詳細に設定できます。

設定	説明
service_discovery!enabled	プールのサービスディスカバリのオン/オフを切り替えます。 このプールが既にオートスケーリングまたはDNSを用いたオートスケーリングのいずれかを使用している場合は、サービスディスカバリを有効にできません。このプールに現在これらの機能のいずれかを使用している場合は、それを無効にしてからサービスディスカバリを有効にする必要があります。
service_discovery!plugin	このプールに使用するサービスディスカバリプラグイン。
service_discovery!plugin_args	プラグインに渡す単一の文字列の引数。 内蔵プラグインでは全て、「--key1=value1 --key2=value2」形式の引数が期待されています。各プラグインに必要な引数の詳細については、UIヘルプページのCatalogs (カタログ) > Service Discovery Plugins catalog (サービスディスカバリプラグインカタログ) > Built-in Plugins (内蔵プラグイン)を参照してください。 ユーザ提供のプラグインでは、この設定はオプションです。 全てのプラグインタイプで、この設定は技術サポートレポートに含まれていません。引数文字列でセキュリティクレデンシャルが指定される可能性があるからです。
service_discovery!interval	1秒単位でのプラグインへのポーリング間隔。 この設定では、プラグインの実行ごとに経過する時間間隔が指定されます。前の実行が完了していない場合、または間隔が経過する前にタイムアウトした場合は、その実行が完了するまでプラグインは再実行されません。 このデフォルトの間隔を一時的により短い間隔で上書きするには、プラグイン応答の「interval_override」フィールドを使用します。詳細については、カスタムユーザプラグインを参照してください。
service_discovery!timeout	プラグインの実行ごとのタイムアウト値。 この設定では、プラグインの実行サイクルが失敗または停止したと見なされるまで、Traffic Managerが待機すべき時間が指定されます。

設定	説明
	タイムアウト期間内にプラグインの実行が完了しなかった場合、プールは直前に取得したノードのリストを使用し続けます。

サービスディスカバリプラグインは、スタンドアロンスクリプトまたは実行可能バイナリであり、リモートサービスからノード(アクティブ、ドレイン中、または無効化済み)のリストを取得します。これらのプラグインの1つを使用するように設定されたプールは、プラグインからの応答に基づいて、そのノードのリストを更新できます。

プラグインは**built-in**または**user-provided**のいずれかとなり、「Service Discovery Plugins」(サービスディスカバリプラグイン)カタログに保存されます。プラグインのリストを表示するには、**Catalogs** (カタログ) > **Service Discovery** (サービスディスカバリ)をクリックします。

内蔵プラグイン

Traffic Managerには、名前付き外部サービスを使用する動的に生成されたノードリストを提供するよう設計された、多数の定義済みプラグインスクリプトが含まれています。各プラグインでは、サービスへの接続を確立するために、一連の入力引数を提供する必要があります。これらの引数はプール設定で提供され、プールごとに一意です。このため、それぞれが外部サービスの異なるインスタンスからノードリストを取得して、複数のプールでプラグインを再利用できます。

Traffic Managerには、プラグインカタログページで接続をテストする機能があります。必要な引数は、`--argument=value`式のリストの形式でプラグインの説明に一覧が示されています。「Test Plugin (プラグインのテスト)」セクションで、引数文字列を**Arguments** (引数)キストボックスに入力してから、**Test** (テスト)をクリックすると接続がテストされます。

テストの結果は、受信されるとすぐに表示されます。



一部のプラグインでは、正常に動作するために、追加のソフトウェアまたはツールが必要です。詳細については、プラグインに付属するUIヘルプページを参照してください。

カスタムユーザプラグイン

Traffic Managerには、カスタムサービスディスカバリプラグインスクリプトおよびプログラムの機能が含まれています。プラグインは、ターミナルから直接実行可能で(例えば、「#!」行で開始するスクリプト)、STDOUTに特定のフォーマットでノードリストを提供する必要があります。Traffic Managerは、このセクションで説明しているスクリプト出力フォーマットでエラーレポートが可能ですが、終了コードまたはSTDERRを使用せず、反応しません。

プール設定内の入力引数インターフェース(**service_discovery!plugin_args**)を使用して、ユーザクレデンシャルまたはその他の機密情報を要求しないことをお奨めします。機密の引数値を渡す必要がある場合は、環境変数を使用するか、個別の入力ファイルにしてディスクから読み取ることを検討してください。ただし、プラグイン引数は技術サポートレポートには含まれません。

Traffic Managerは、カスタムサービスディスカバリプラグイン用の仕様バージョンシステムを運用します。Traffic Managerの新しいリリースごとに、新機能を利用し、パフォーマンスを向上する新しい仕様バージョンが導入されます。古い仕様バージョンは、技術上またはサポート上の理由で廃止が必要になるまでサポートが継続されます。Ivantiは、現在サポートされている仕様バージョンを、UIヘルプページと製品リリースノートで配布します。詳細については、Pulse Secureテクニカルサポートにお問い合わせください。

スクリプト出力はJSON構造に準拠し、トップレベルに以下のプロパティを含んでいる必要があります。

プロパティ	タイプ	説明
version	整数	必須。このスクリプトが適合するプラグイン仕様のバージョン。
code	整数	必須。実行ステータスを表すHTTPステータスコード。正常終了には「200」を使用します。
error	文字列	オプション。何らかの原因でプラグインが失敗した場合に、Traffic Managerイベントログに記録されるエラーメッセージ。
interval_override	整数	オプション。プラグインが次回実行されるまで、プール設定で指定されたポーリング間隔を、より短い間隔で上書きします。この値は、設定されたプール間隔よりも大きい場合に無視されます。
nodes	ハッシュ配列	必須。ノードのリスト ノードのリストには以下のものが含まれます。 ip 必須の文字列。ノードのアドレス。 port 必須の整数。このノードのポート番号。

プロパティ	タイプ	説明
		weight オプションの整数。他全てのノードに対する相対値として、このノードに適用される加重を表す数字。サービスで <i>weighted round robin</i> などの加重負荷分散アルゴリズムを使用している場合にこのフィールドを使用します。詳細については、負荷分散を参照してください。 draining オプションのブール値。このノードを「ドレイン」状態にするには、true (真)に設定します。 disabled オプションのブール値。このノードを無効にし、トラフィックを受け入れないようにするには、true (真)に設定します。これは、スクリプト内でデフォルト(またはフォールバック)位置として役立ち、エラーが発生した場合に、存在するノードを抽象化したリストがプールに入力されるようになります。

以下のJSON構造は、カスタムプラグイン出力の例です。

```
{
  "version":1,
  "nodes":[
    { "ip":"192.0.2.0", "port":80 },
    { "ip":"192.0.2.1", "port":81, "draining":true }
  ],
  "code":200
}
```


トラフィックIPグループとフォールトトレランス

Traffic Managerのクラスタでは、受信したネットワークトラフィックをTraffic Managerに分配し、いずれかのTraffic Managerで障害が発生した場合は、そのTraffic Managerに割り当てられたトラフィックを別のTraffic Managerに転送することができます。トラフィックの分配は、トラフィックIPグループを使用して設定します。

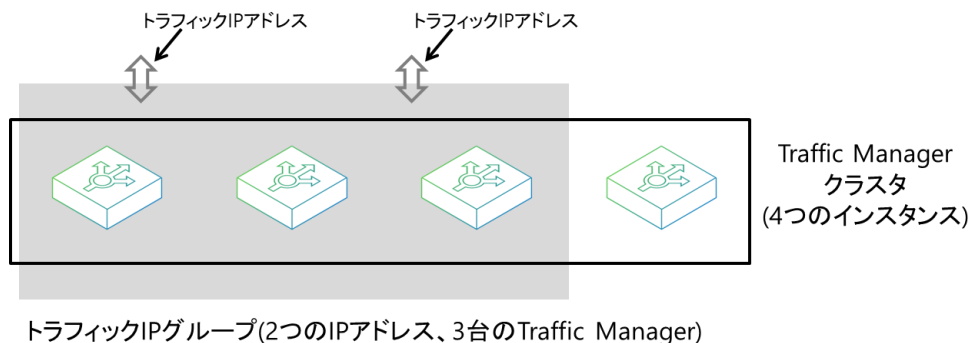
本章で説明する機能は、ご使用の製品プラットフォームによってわずかに異なる場合があります。詳細および製品の仕様については、IvantiのWebサイト(www.ivanti.com)を参照してください。

フォールトトレランス

トラフィックIPアドレスとトラフィックIPグループ

「トラフィックIPアドレス」は、高い可用性を維持しなければならないIPアドレスです。トラフィックIPアドレスは、「トラフィックIPグループ」と呼ばれるグループに割り当てられます。

各トラフィックIPグループは、クラスタ内の一部(または全て)のTraffic Managerで管理されます。Traffic Managerは連携してトラフィックをTraffic Manager間で分配し、トラフィックIPアドレスに送信された全てのトラフィックがクラスタ内のいずれかのTraffic Managerに送信され、管理されるようにします。



上図では、クラスタ内の4台のTraffic Managerのうち3台にまたがるようトラフィックIPグループが設定されています。このトラフィックIPグループはトラフィックIPアドレスを管理し、3台のTraffic Managerによって、これらのトラフィックIPアドレスが利用可能な状態に保たれます。

例えば、WebサービスをIPアドレス34.56.78.90で公開しているとします。この場合、そのIPアドレスをトラフィックIPグループに追加することにより、サービスを常に利用可能な状態に保つことができます。IPアドレスを追加すると、Traffic ManagerクラスタでそのIPアドレスが有効になり、このIPアドレスに送信される全てのトラフィックが管理されます。通常、1つ以上のトラフィックIPアドレスに名前解決されるよう、サービスにDNSエントリを設定します。

トラフィックIPグループ内でのトラフィックの分配

トラフィックは、以下のいずれかの分散ポリシーに従ってクラスタ内のTraffic Manager間で分配されます。

- **シングルホストモード:** 各トラフィックIPアドレスは、グループ内のいずれかのTraffic Managerで有効になります。グループが複数のIPアドレスで構成されている場合、トラフィックができる限り均等に分配されるように各IPアドレスが別々のTraffic Managerで有効になります。
- **マルチホストモード:** 各トラフィックIPアドレスは、グループ内の全てのTraffic Managerで有効になります。各IPアドレス宛てのトラフィックは、全てのTraffic Managerに均等に分配されます。
- **ルートヘルスインジェクション:** (一部の種類の製品は該当せず)各RHITraffic IPグループにアクティブなTraffic Managerが1つずつ、オプションでパッシブなTraffic Managerが1つずつ含まれます。グループ内の全てのトラフィックIPアドレスは、加えられている両方のTraffic Managerによってprivately(ループバックで)有効になり、ルーティングプロトコルとしてOSPFv2またはBGPを使用して、隣接したルーティングドメインに動的にアドバタイズされます。レスポンスでは、ルータはアクティブなTraffic Managerに全てのトラフィックを送信します。

各Traffic Managerは定期的に他のTraffic Managerにレポートを送信し、いずれかのTraffic Managerで障害が発生した場合は、そのTraffic Managerに割り当てられたトラフィックを残りのTraffic Managerが引き継ぎます。これにより、トラフィックIPアドレスを使用するサービスは常に利用可能な状態に保たれます。

トラフィックIPグループの設定例については、[設定例](#)を参照してください。Traffic Managerクラスタの作成方法については、[クラスタの作成](#)を参照してください。

トラフィックIPアドレスの選択

通常、Traffic Managerには、受信トラフィック用にはフロントエンドネットワークが、バックエンドサーバ用にはバックエンドネットワークが追加されます。IPレベルでは、これらのネットワークはIPサブネットとして定義されます。例えば、192.0.2.0/24、198.51.100.0/16などになります。

シングルホストおよびマルチホストトラフィックIPグループでは、これらの直接追加されたサブネット内にあるトラフィックIPアドレスを選択する必要があります。適切な物理インターフェースにあるこれらのIPアドレスがTraffic Managerで有効になることによって、前述のネットワークにあるルータやその他のデバイスがそれらのIPアドレスに直接アクセスできるようになります。

RHITraffic IPグループでは、これらの直接追加されたサブネット内にはないトラフィックIPアドレスを選択する必要があります。RHITraffic IPアドレスがTraffic Managerで有効になるときは、必ずTraffic Managerの内部ループバック(lo)インターフェースにおいて、RHITraffic IPアドレス自体の/32サブネットでprivatelyに有効になります。ルータやその他のマシンは、常にOSPFv2ルーティングプロトコルを使用して確立されたルーティング情報に従って、Traffic ManagerのフロントエンドIPアドレスを経由してそれらのIPアドレスに間接的にトラフィックを送信します。

直接追加されたサブネットとは、以下の条件に該当する場合を指します。

- 特定のインターフェースで静的に有効になるIPアドレスを設定するときに指定する。
- 「トラフィックIPネットワーク」テーブルにエントリを追加することによって指定する。詳細については、[インターフェースとサブネットのマッピング\(トラフィックIPネットワーク\)](#)を参照してください。

トラフィックIPグループの作成

トラフィックIPグループを作成するには、**Services** (サービス) > **Traffic IP Groups** (トラフィックIPグループ)をクリックします。このページでは、トラフィックIPグループを新規作成したり、既存のトラフィックIPグループを編集したりすることができます。

グループのメンバにするTraffic Managerの名前を入力し、選択します。リストに、グループのトラフィックIPアドレスをスペースまたはカンマで区切って入力し、必要に応じてIP分散モードを選択したうえで、**Create Traffic IP Group** (トラフィックIPグループの作成)をクリックします。

トラフィックの分配

同じトラフィックIPグループ内のTraffic Manager間でどのようにトラフィックを分配するかは、IP分散モードで決まります。

トラフィックIPグループは、トラフィックIPアドレスを一度に1台のTraffic Managerで有効にするか(シングルホストモード)、グループ内の全てのTraffic Managerで同時に有効にする(マルチホストモード)ように設定できます。一部の製品では、分散モードとしてルートヘルスインジェクションも選択できます。

シングルホストモード

グループ内の各IPアドレスは、一度に1台のTraffic Managerでのみ有効になります。グループに複数のIPアドレスが含まれている場合、加えられているTraffic Manager全体で分散されます。1つのTraffic Managerで障害が発生すると、そのトラフィックIPアドレスが、グループ内で稼働しているその他のTraffic Managerに引き渡されます。

トラフィックIPグループに**keeptogether**オプションを設定することにより、そのグループ内でのIPアドレスの分散を防止することができます。その場合、全てのトラフィックIPアドレスが同じTraffic Managerで有効になります。この機能は、トラフィックIPグループとIPトランスペアレンシを組み合わせる場合に便利です。詳細については、[クラスタでのIPトランスペアレンシの使用](#)を参照してください。

マルチホストモード

各IPアドレスが同時に全てのマシンで有効になります。各IPに送信されるデータは、グループ内の全てのTraffic Managerで受信され、各Traffic Managerは、受信したコネクションの等しい配分を担当します。マシンで障害が発生した場合、そのマシンで担当していた配分が残りのアクティブなマシンに分配されます。

マルチホストIPアドレスは、ローカルネットワーク上でマルチキャストパケットを使用して動作します。Traffic Managerは、ローカルサブネットのマルチキャストMACアドレスを使用してトラフィックIPアドレスをアドバタイズします。これにより、スイッチおよびルータに、そのIPへのトラフィックを全てのTraffic Managerに送信しなければならないことが示されます。使用されるMACアドレスは、別のマルチキャストグループ(239.0.0.0～239.255.255.255の範囲のIPとして表される)を提供することにより計算されます。このマルチキャストグループは、ネットワーク上でユニークである必要があります。

各Traffic Managerは、受信した全てのコネクションを受け取り、コネクションの送信元IPアドレスに基づく計算を行い、そのコネクションを受け入れるか、(グループ内の別のTraffic Managerが受け入れるため)そのまま破棄するかを決定します。

この方法では、送信元IPアドレスのみによって判断が行われるため、(クライアントがIPアドレスを変更しない限り)同じクライアントは同じTraffic Managerで処理されることになります。これには、Traffic Managerに保存されているセッション維持データとSSLセッションデータを利用できるというメリットがあります。

ただし、特にクライアントの送信元IPアドレスが少数である場合に、Traffic Managerに分配されるトラフィックが不均等になる可能性があります。不均等に分配されると、負荷分散が行われるサービスの性能に影響を与えることがあります。このような状況が懸念される場合は、**consider source port** (ソースポート考慮)を設定し、送信元IPアドレスとポートに基づいてトラフィックを分配するようTraffic Managerに指示します。これにより、トラフィックがクラスタ全体に容易に分配されるようになります。

セッション維持データをクラスタ全体で共有するときに多少の遅延が生じることがあるため、ソースポートを使用すると、一部のセッション維持手法の妨げになることがあります([セッション維持とマルチホストトラフィックIPアドレスの併用](#)を参照)。同様に、**consider source port** (ソースポート考慮)は、FTP仮想サーバが使用しているトラフィックIPアドレスでは使用しないでください([FTP](#)を参照)。



Traffic ManagerがマルチホストIPに接続しようとする、そのコネクションは常にローカルTraffic Manager自体が受け取ります。

ルートヘルスインジェクション

Traffic Managerは、グループで管理されるトラフィックIPアドレスを広範なルーティングネットワーク全体に知らせることにより、フロントエンドマシンのフォールトトレランスを使用します。この手法を使用するためには、組織内のルータが次のいずれかのルーティングプロトコルを使用するように設定されている必要があります。

- **OSPFv2**: Open Shortest Path Firstバージョン2
- **BGP**: Border Gateway Protocol

RHIはIPv4アドレスにのみ使用できます。IPv6はサポートされていません。

ルートヘルスインジェクション(RHI)に基づくトラフィックIPグループには、いずれも単一のアクティブなTraffic Manager、またはアクティブ-パッシブ構成を使用したペアのTraffic Managerで構成される、最大2台のTraffic Managerが含まれます。

さらに、分散モードにRHIを選択する場合は、アクティブおよびパッシブなTraffic Managerに対してルーティングメトリクスを指定します。Traffic Managerがアップストリームのルーティングネットワークにルートのインジェクションを行う場合、これらのメトリクスによってマルチクラスタ(またはマルチデータセンタ)シナリオ内におけるTraffic Managerのインスタンスの優先順位が決定します。

詳細については、[ルートヘルスインジェクションおよびネットワーク](#)を参照してください。

パッシブマシン

Traffic Managerに「パッシブ」のマークを付けると、そのTraffic Managerでは、パッシブでないいずれかのTraffic Managerで障害が発生するまで、トラフィックIPグループ内のどのIPアドレスも有効にならないか(シングルホストモード)、何の負荷も処理されません(マルチホストモード)。

Traffic Managerを既存のトラフィックIPグループに追加すると、トラフィックの一部が新しいTraffic Managerに転送されることがあります。これにより、転送の瞬間に必然的に一部のコネクションが切断されます。このような状況は、Traffic Managerを「パッシブ」として追加することで対処できます。この場合、いずれかのピアで障害が発生するまで、そのTraffic Managerはトラフィックを全く受信しません。

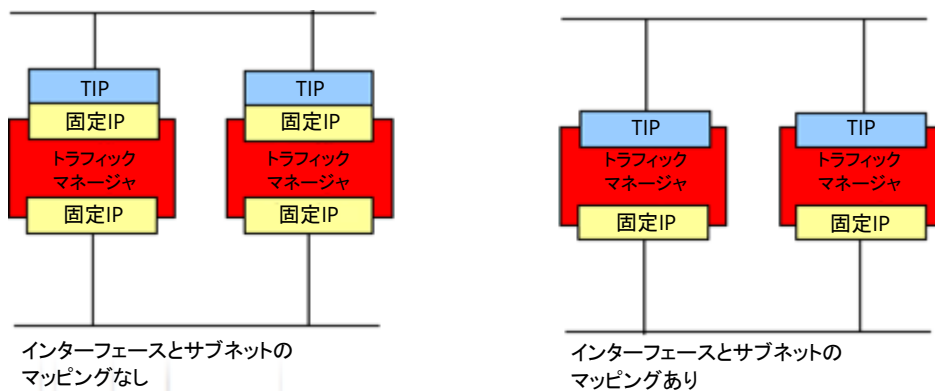
トラフィックIPグループの無効化

トラフィックIPグループを無効化するには、**enabled** (有効化)チェックボックスの選択を解除します。このオプションは、グループを一時的に無効化しつつも、その後元の状態に戻すために設定を維持する場合に使用します。

グループが無効化されているときは、そのグループのどのIPアドレスも有効になりません。RHIトラフィックIPグループに対するルートアドバタイズメントは取り消されます。

インターフェースとサブネットのマッピング(トラフィックIPネットワーク)

Traffic Managerのバージョン7.0までは、トラフィックIPアドレスのホストについて、インターフェースに同じサブネットのIPアドレスが設定されていることが前提条件となっていました。この前提条件には、Traffic Managerが(ルーティングテーブルとインターフェースの「ヘルス」に基づいて)どのデバイスがトラフィックIPアドレスをホストできるかを自動的に判断できるというメリットがあります。その一方、最終的に、実際には何のトラフィックも受信しないのに容量を占有するIPアドレスになるという欠点があります。トラフィックを受信させたいネットワークが小規模な場合(/29など)、このことがストレスの原因になります。インターフェースとサブネットのマッピング機能を使用すれば、サブネットとインターフェースの明示的なマッピングを定義することができます。この機能を使用すると、インターフェースに特定サブネットのIPアドレスを設定しなくても、そのサブネットでトラフィックIPアドレスをホストすることができます。



トラフィックIPネットワークを設定するには、以下の手順に従います。

1. **Services** (サービス) > **Traffic IP Groups** (トラフィックIPグループ)ページに移動します。
2. 「Traffic IP Networks (トラフィックIPネットワーク)」セクションで、**Network Settings** (ネットワーク設定)リンクをクリックします。
3. **Add Network** (ネットワークの追加)フィールドに、IPアドレスをホストするIPサブネットをCIDR表記で入力します(192.168.50.0/24など)。
4. **Default Interface** (デフォルトインターフェース)ドロップダウンボックスから、このサブネットをホストするネットワークインターフェースを選択します。



Traffic Managerによって、トラフィックIPアドレスをホストするためのDHCP対応ネットワークインターフェースの使用が妨げられます。トラフィックIPサブネットに対して静的ネットワーキングで設定されたインターフェースのみを使用してください。

5. **Update** (更新)をクリックして変更を保存します。

目的のインターフェースにトラフィックIPサブネットを設定すると、クラスタ内のその他のTraffic Managerで、同じサブネットに使用するインターフェースを設定することのできるダイアログボックスが表示されます。

フォールトトレランスの設定

クラスタ内のTraffic Managerは、アクティブな各ネットワークインターフェースを通してICMP Pingを使用し、ネットワークと通信できることを定期的に確認します。この確認後、それぞれのヘルス状態(good (良好)またはfailed (障害発生))を記述したメッセージをブロードキャストします。

クラスタ内のいずれかのTraffic Managerで障害が発生すると、その他のTraffic Managerが、障害が発生したTraffic Managerで管理されていた全てのトラフィックIPアドレスを引き継ぎます。

フォールトトレランスの設定

フォールトトレランスは、**System** (システム) > **Fault Tolerance** (フォールトトレランス) > **General** (一般)で設定することができます。

- **flipper!autofailback**: トラフィックIPアドレスをホストしているTraffic Managerで障害が発生すると、そのTraffic Managerが復旧するまで、破棄されたアドレスはクラスタ内の別のTraffic Managerで有効になります。

元のTraffic Managerのコネクションが復旧した時点で、トラフィックIPアドレスが自動的にそのTraffic Managerへ返されるようにするには、「Yes」に設定します。

「No」に設定すると、復旧に合わせて元のTraffic Managerは強制的に保留モードになります。このモードでは、トラフィックIPアドレスをいつ有効にするかについて指示されるまで、Traffic Managerは待機します。

Traffic Managerが保留状態にあると、管理UIに警告が表示されます。保留モードのTraffic Managerを完全にアクティブな状態に戻すには、**Diagnose** (診断) > **Cluster Diagnosis** (クラスタ診断)ページ、または**Services** (サービス) > **Traffic IP Groups** (トラフィックIPグループ)ページを使用します。



クラスタ内のその他全てのTraffic Managerで障害が発生すると、保留モードのTraffic Managerは自動的にアクティブな状態に戻ります。この場合、アクティブな状態に戻ったTraffic Managerは、それ自体が管理している全てのトラフィックIPアドレスを引き継ぎます。また、既に実行中のサービスを中断させることがないTraffic Managerも、自動的にアクティブな状態に戻ります。

- **flipper!autofailback_delay**: 回復したTraffic Managerが、破棄したトラフィックIPアドレスを再び有効化するまでの秒単位の時間。

flipper!autofailbackを「Yes (はい)」に設定した場合、以前に障害が発生したクラスタメンバは、フェイルバックを実行できるようになるまで、この期間、正常に動作し続ける必要があります。このメカニズムは、回復したTraffic Managerが、トラフィックIPアドレスの有効化を要求される前に、接続をブロックしている外部要因の影響を引き続き受けなていないことを確認するために役立つことがあります。待機期間中に、追加の障害が検出された場合は、遅延タイマーが再開されます。

遅延オートフェイルバック中には、いつでも手動のフェイルバックを行うことができます。手動のフェイルバックを実行するには、**Diagnose** (診断) ページで、保留モードのTraffic Managerを再びアクティブにします。

flipper!autofailback_delayの値の変更は、後続の障害またはフェイルバックイベントの成功のいずれかによって遅延タイマーが強制的に再開されるまで、すでに動作中の遅延タイマーに影響しません。ただちに変更を適用するか、現在のタイマーをキャンセルするには、**flipper!autofailback**を「No (いいえ)」に設定してオートフェイルバックを無効にします。次に、**flipper!autofailback**を「Yes (はい)」に再設定すると、新しい遅延期間が適用されます。この状況で**flipper!autofailback_delay**を0(ゼロ)に設定すると、動作中のタイマーが停止し、ただちにフェイルバックが発生します。

遅延タイマーのカウントダウン中にオートフェイルバックをオフにすると、システムがすでに正常な状態になり、遅延タイマーが動作していた場合でも、常にTraffic Managerの使用が妨げられます。

- **flipper!monitor_interval**: Traffic Managerが、ネットワーク上のその他のデバイスと通信できるかどうかを定期的に確認するために使用する時間間隔(ミリ秒単位)です。確認結果は、マルチキャストアドレスに存在するその他のTraffic Managerに通知されます。間隔を短くすると、ネットワーク内の障害がより早く検出されるようになりますが、それによってシステムへの負荷は高まります。
- **flipper!monitor_timeout**: Traffic Managerが、コネクションを確認するために使用するデバイスの1つと通信を試みているときに、Traffic Managerにレスポンスを待機させる時間の長さ(秒単位)です。この時間内にレスポンスが受信されなかった場合、Traffic Managerは、そのデバイスとの通信は不可能であると見なします。この時間を短くすると、Traffic Managerによる障害の検出が早くなりますが、低速のコネクションが障害であると誤って判断されることがあるため、システムの信頼性は低下します。
- **flipper!heartbeat_method**: クラスタ内では、各Traffic Managerは定期的なハートビートメッセージを送信し、全てのソフトウェアおよびネットワークサービスが稼働していることを通知しなければなりません。

Traffic Managerに対し、送信相手を指定したUDPメッセージをクラスタ内の各Traffic Managerに送信するように指示するには、ユニキャストを使用します。ネットワーク内のスイッチが、より効率的になり得るマルチキャストパケットの手法を処理できない場合は、この設定を使用してください。ユニキャストメッセージは、全く異なるデータセンタにあるTraffic Managerなど、別のサブネットに存在するマシンにも到達します。Traffic Managerがユニキャストメッセージを監視する必要があるネットワークポートを指定するには、**Communication Port**: (通信ポート)を使用します。このポートは、クラスタ内のその他全てのTraffic Managerから通信できなければなりません。

Traffic Managerに対し、Traffic Managerが接続されている全てのスイッチの全てのネットワークポートをトラバースするマルチキャストネットワークパケットを使用してこれらのメッセージを送信するように指示するには、マルチキャストを使用します。マルチキャストメッセージは、同じサブネットに存在するマシンにのみ到達します。**Multicast address and port:** (マルチキャストアドレスおよびポート)フィールドでは、全てのTraffic Managerに監視させるマルチキャストアドレスを設定してください(デフォルトでは239.100.1.1:9090)。

- **flipper!use_bindip:** マルチキャストメッセージの送受信を管理インターフェースだけで行うかどうかを設定します。

Traffic Managerが、メイントラフィックに使用される別々の専用ネットワーク上で相互に通信する場合は、Yesに設定します。ただし、結果としてシステムの信頼性が損なわれる可能性があることに注意してください。

Traffic Managerに対し、全てのネットワークインターフェースを経由して接続性を通知するために使用しているマルチキャストメッセージを送信するように指示するには、「No (いいえ)」に設定します(デフォルト)。Noに設定すると、1つのネットワークインターフェースで障害が発生しても、クラスタ内のその他のTraffic Managerはそのマシンで障害が発生していると思なすことはないため、システムの信頼性が高まります。

- **flipper!arp_count:** ネットワークインターフェースが有効なときにTraffic Managerが送信するARPパケットの数です。Traffic Managerの新たなインターフェースを認識するために、ネットワーク上のその他のデバイスはARPリクエストを1つだけ必要としますが、ARPパケットが行方不明になったり、失われたりすることがあります。インターフェースが有効な場合、Traffic Managerはここで定義された数のARPパケットを送信し、ブロードキャストの信頼性と速度を向上させます。
- **flipper!igmp_interval:** Traffic Managerが、マルチホストトラフィックIPグループに対するIGMPメンバーシップレポートパケットを一方向的に送信するために使用する時間間隔(秒単位)です。この間隔を使用すると、スイッチおよびルータにTraffic Managerのマルチキャストサブスクリプションを忘れずに実行させることができます。この機能を無効化するには、間隔を0に設定してください。
- **flipper!verbose:** Traffic Managerに全ての接続テストを記録するかどうかを設定します。システムログに詳細情報を出力するには、Yes (はい)に設定します。



この設定は、診断目的のみに使用することをお奨めします。

- **flipper!frontend_check_addrs:** フロントエンドのコネクションを確認するために使用するIPアドレスです。Traffic Managerがこれらのアドレスに対するPingの実行を適切に完了できない場合、Traffic Managerはそのネットワークのコネクションが切断されていると判断し、クラスタ内のその他のTraffic Managerにフェールオーバーします。

Traffic Managerがイントラネットの一部である場合など、Traffic Managerが外部接続を必要としないように設定されている場合、このフィールドを空白にすることで、外部接続の確認を無効化することができます。

Traffic Managerのフォールトトレランスチェック

各Traffic Managerは、それ自体のネットワークインターフェースが正常に動作しているかどうかをチェックします。このチェックは以下に行われます。

- デフォルトゲートウェイに対して定期的にPingを実行し、フロントエンドネットワークインターフェースが機能していることを確認します。
- 使用中のプール内の各バックエンドノードに対して定期的にPingを実行し、バックエンドネットワークインターフェースが機能していることを確認します。

Traffic Managerは、デフォルトゲートウェイに対するPingが失敗した場合、または現在使用しているどのプールのどのノードに対するPingも失敗した場合、そのTraffic Manager自体に障害が発生していると判断します。

フロントエンドチェックのオーバーライド

例えば、ゲートウェイがICMP Pingに対してレスポンスを返さない場合などに、フロントエンドチェックのオーバーライドが必要になることがあります。フロントエンドチェックをオーバーライドするには、**flipper!frontend_check_addrs**に空の文字列を設定します。

ヘルスブロードキャスト

各Traffic Managerは、ローカルヘルスチェックの結果を、そのヘルス状態に関わらず定期的にブロードキャストします。

デフォルトでは、各マシンは、1秒間に2回ハートビートメッセージをブロードキャストします。この動作は、**flipper!monitor_interval**で設定することができます。

ブロードキャストメッセージの送信方法は、マルチキャストとユニキャストから選択できます。**flipper!heartbeat_method**設定でいずれかの方法を選択してください。

ユニキャストを使用する場合、「State Sharing (状態共有)」を有効化する必要があります。これを行うには、**System (システム) > Global Settings (グローバル設定)**ページで**state_sync_time**をゼロ以外の値に設定します。

多くの状況では、ユニキャストが適しています。ただし、Traffic Managerクラスタが単一のネットワークセグメント内にある場合など、全てのTraffic Managerがマルチキャストメッセージを受信できるようにネットワークが設定されている場合は、代わりにマルチキャストを使用してください。両方の手法を試しても問題が解消されない場合は、サポート窓口にお問い合わせして指示を受けるようにしてください。



Hyper-VベースのTraffic Managerを使用している場合はデフォルトのユニキャスト手法を使用することをお奨めします。複数のインスタンスがまとめてクラスタ化されている場合、状況によってはマルチキャストを使用することでパケット損失につながる可能性があります。

ハートビート通信の問題をデバッグする場合は、tcpdumpプログラムを使用してネットワーク上の全てのハートビートメッセージをキャプチャすることをお奨めします。

```
# tcpdump -i eth0 ip multicast
```

ご使用のTraffic Managerから送信されたハートビートメッセージのみをキャプチャするには、以下のプログラムを使用してください。

```
# tcpdump -i eth0 dst host 239.100.1.1 and dst port 9090
```



この例では、Traffic Manager管理通信のデフォルトであるポート9090を使用しています。異なる場合は、実際のTraffic Managerで使用されているポート番号に置き換えてください。Traffic Managerのネットワークポートの要件の詳細については、[Traffic Managerが使用するデフォルトポート](#)を参照してください。

クラスタのヘルス判定

各Traffic Managerは、クラスタ内のその他全てのTraffic Managerから送信されたヘルスメッセージを監視し、以下のいずれかの条件を満たした場合にいずれかのピアで障害が発生したと判断します。

- ピアから「I have failed」ヘルスメッセージを受信した場合
- **flipper!monitor_timeout**で設定された値以内に、ピアから何のメッセージも受信しなかった場合
- ピアが、その子プロセスのいずれかでトラフィックが処理されなくなったことをレポートした場合。これは、いずれかの子プロセスが**flipper!child_timeout**で設定した値の時間内にレスポンスを返さない場合に発生します。

Traffic Managerは、ピアから「I am healthy」メッセージを受信し始めると、そのピアが復旧したと判断します。

フェールオーバー

クラスタ内の各Traffic Managerは、ピアのいずれかで障害が発生したと判断した場合、障害が発生したシステムに割り当てられていた一部または全てのトラフィックを引き継ぐことができます。これをどのように行うかは、トラフィック分散方法によって決まります。

トラフィックIPアドレstransファ(シングルホストモード)

クラスタ内の各Traffic Managerは、どのマシンがアクティブかについて保持している情報に基づき、使用するトラフィックIPアドレスを判断します。クラスタは完全な決定的アルゴリズムによって、以下のようにIPアドレスをマシンに分配します。

- このアルゴリズムは決定的であるため、いずれかのピアで障害が発生するか復旧したときに、Traffic Manager相互でネゴシエーションを行う必要がありません。
- このアルゴリズムは、クラスタ内のアクティブなTraffic ManagerにトラフィックIPアドレスを広く分散させ、Traffic Managerの障害または復旧時にIPアドレスの転送件数を最小限に抑えるよう最適化されています。

Traffic ManagerでトラフィックIPアドレスが有効になると、そのTraffic Managerは、隣接するネットワークデバイスにそのIPアドレスに対応するMACアドレスが変更された可能性があることを通知するARPメッセージをいくつか送信します。Traffic Managerは最大10件のARPメッセージを送信し(**flipper!arp_count**設定で調整可能)、これらのメッセージの送信頻度は**flipper!monitor_interval**設定でコントロールできます(デフォルトでは0.5秒間隔でメッセージを送信)。

Traffic Managerは、それ自体のネットワークコネクションで障害が発生したことを検出すると、そのTraffic Managerで有効になっていたトラフィックIPアドレスを即座に無効にし、ピアにI have failedヘルスメッセージをブロードキャストします。これは、クラスタ内のその他のTraffic ManagerがこのTraffic Managerでの障害発生を認識し、そのインターフェースを有効にすることを期待して行われる動作です。

トラフィックIPアドレ스트ランスファ(シングルホストモード)

トラフィックIPグループ内の各Traffic Managerは、パケットの送信元IP (およびオプションでソースポート。詳細は[トラフィックの分配](#)を参照)に基づいて、各パケットを処理すべきかどうかを確定的に選択します。

Traffic Managerで障害が発生した場合、そのTraffic Managerに割り当てられていた負荷が残りのTraffic Managerに均等に分配されます。Traffic Managerは障害から復旧すると、負荷が均等になるようピアから負荷を受け入れるため、トラフィックIPグループ内の稼働しているマシンにトラフィックが常に均等に分散されます。

 Traffic Managerソフトウェアには、マルチホストIP機能がデフォルトでは含まれていません。この機能は、追加カーネルモジュールとしてダウンロードしてインストールすることができ、Linuxカーネルのバージョン2.6.18以上でサポートされます。サポートされているバージョンについての詳細は、IvantiのWebサイト(www.ivanti.com)からダウンロードできるTraffic Managerのドキュメントを参照してください。

トラフィックIPアドレストランスファ(RHIモード)

Traffic Managerのフォールトトレランスチェックが失敗すると、Traffic ManagerはRHIトラフィックIPグループで使用するアドレスを下げ、ネットワークのルートアドバタイズメントを取り下げます。

RHIトラフィックIPグループ内の専用のアクティブなTraffic Managerと通信不能であることがネットワークにより検出されると、専用のパッシブなTraffic Managerや、別のデータセンタ内で同じトラフィックIPアドレスをホストしているTraffic Managerなど、最も低いメトリクスを使用する、次に可用性の高いルートを代用してトラフィックIPアドレスに対するルーティングが決定されます。

障害からの復旧

障害の発生したTraffic Managerが復旧すると、そのTraffic Managerの割り当て分のトラフィックが他のTraffic Managerから転送されます。

分配されたトラフィックがTraffic Manager間で転送されるたびに、そのトラフィックに使用されていたコネクションは切断されます。この動作は、Traffic Managerで障害が発生したために転送が行われる場合には避けられません。Traffic Managerの復旧時には望ましくない場合があります。

この場合、管理UIの **System** (システム) > **Fault Tolerance** (フォールトトレランス) ページで **flipper!autofailback** 設定を無効化することができます。この設定を無効化すると、Traffic Managerは復旧時にトラフィックを受け入れません。代わりに、Traffic Managerが復旧し、IPアドレスを再び受け入れることができることを示すメッセージがユーザインターフェースに表示されます。

Traffic Managerを再びアクティブにするには、**Diagnose** (診断) ページに移動し、**Reactivate this Traffic Manager** (このTraffic Managerの再アクティブ化) リンクを選択します。

または、各トラフィックIPグループを編集し、復旧したTraffic Managerをパッシブに設定することもできます。全てのグループでパッシブに設定すると、トラフィックを分配する必要がなくなります。その後、再度自動的にアクティブになり、エラー状態が解消されます。また、分配されたトラフィックが転送されないため、トラフィックが失われることもありません。

フォールトトレランスのデバッグとモニタリング

全ての状態変化とIPアドレスの転送は、関連する各Traffic Managerのイベントログに記録されます。電子メールアラートが正しく設定されていれば([イベント処理およびアラート](#)を参照)、Traffic Managerは、状態変化と、その結果行われたIPアドレスの転送について記述した電子メールメッセージをシステム管理者に送信します。

詳細なデバッグを行うには、**flipper!verbose**設定を有効化します。これにより、各Traffic Managerでは、接続テスト、送信したブロードキャストメッセージ、受信したブロードキャストメッセージが全て記録されるため、フォールトトレランスが期待どおりに動作しない原因を判断するときに役立ちます。

管理UIの**Diagnose** (診断)セクションにある**Cluster Diagnosis** (クラスタ診断)ページも確認してください。このページには、正しく受信されていないブロードキャストメッセージがあるかどうかが表示され、エラーが発生している場合はシステムステータスの概要が表示されます。

ルートヘルスインジェクションの状態に関する詳細は、管理UIの**Diagnosis** (診断) > **Routing** (ルーティング) ページを参照してください。

BGP接続性の設定

トラフィックIPグループは、ルーティングプロトコルとしてBGPを使用して、IP分散モードにRHIを使用するように設定することができます。トラフィックIPアドレスに対してRHIを有効化するには、**Services** (サービス) > **Traffic IP Groups** (トラフィックIPグループ)ページを使用します。

System (システム) > **Fault Tolerance** (フォールトトレランス) > **BGP Route Health Injection** (BGPルートヘルスインジェクション)の設定によって、クラスタ内のTraffic Managerと、隣接するルーティングインフラストラクチャとの間のBGP接続性が制御されます。

- **bgp!enabled**: BGPルートヘルスインジェクションを有効化するかどうかを設定します。
- **bgp!as_number**: Traffic Managerクラスタが動作する自律システム(AS)です。10進値を使用して、ASを指定します。この数値は、クラスタ内の全てのTraffic Manager(またはマルチサイトマネージャクラスタ内の各ロケーション設定。[マルチサイトクラスタの管理](#)を参照してください)に適用されます。

BGPルータIDの設定

デフォルトでは、クラスタ内の各Traffic Managerは、それ自体のIPv4デフォルトゲートウェイへの接続に使用されているアドレスおよびサブネットを使用して、BGPのルーティングドメインに組み込まれます。この動作をオーバーライドするには、**flipper!bgp_router_id**を使用して、クラスタメンバごとに別のIDを設定します。

BGPネイバーの管理

OSPFv2とは異なり、Traffic ManagerはそのBGPネイバーを自動的に検出できません。クラスタでBGPセッションを確立できるようにするには、初めに隣接するルータを定義してから、各クラスタメンバと対象ネイバー間に適用されるマッピングを作成する必要があります。

このマッピングで使用できるBGPネイバーを管理するには、**Manage BGP Neighbors (BGPネイバーの管理)**をクリックします。このページを使用して、新しい隣接するBGP対応ルータを定義し、既存のネイバー定義のパラメータを変更します。

新しい各ルータに対して、以下のBGPコネクションパラメータを入力します。

フィールド	説明
BGP Neighbor name (BGPネイバー名)	識別名
IP Address (IPアドレス)	ネイバーのIPv4アドレス
AS Number (AS番号)	このネイバーのAS番号。これは、推奨される対応Traffic Managerと同じASにするか(セッションでiBGPが使用されます)、まったく異なるASすることができます(セッションでeBGPが使用されます)。
Keepalive (キープアライブ)	ヘルスメッセージ送信の間隔(秒単位)
Holdtime (ホールド時間)	最後にkeepaliveを受信してから、このネイバーで、マッピングされたTraffic Managerがアクティブではなくなったと見なされるまでの最小間隔(秒単位)
Advertisement Interval (アドバタイズメント間隔)	このネイバーに対して実行するアドバタイズメントの間隔(秒単位)
Authentication Password (認証パスワード)	(オプション)マッピングされたTraffic Managerが、このネイバーで使用する必要があるBGPごとのセッションパスワード 2.6.20よりも前のLinuxカーネルバージョンまたは関連オプション (CONFIG_TCP_MD5SIG)を有効化せずに構築されたカーネルでは、この設定がサポートされないことがあります。詳細については、システム管理者またはサポート窓口にお問い合わせください。

BGPネイバーを定義した後に、**System (システム) > Fault Tolerance (フォールトトレランス)**をクリックしてこのページに戻り、個々のTraffic Managerに対するマッピングを作成します。

BGPを使用する前に、隣接するBGPルータを設定して、選択したTraffic Managerとのセッションを確立する必要があります。これは、実質的に、コネクションのTraffic Manager側で最後に作成したネイバー定義に対応します。

Update (更新)をクリックしてマッピングと設定を適用します。このページの設定は、Traffic Managerクラスタの全てのメンバに複製されます。

Traffic Managerによって、全てのルーティングと接続失敗がイベントログにレポートされます。管理UIのアプレットとDiagnose (診断) ページの両方に、現在のステータスが反映されます。

OSPFv2の接続性の設定

トラフィックIPグループは、ルーティングプロトコルとしてOSPFv2を使用して、IP分散モードにRHIを使用するように設定することができます。トラフィックIPアドレスに対してRHIを有効化するには、**Services** (サービス) > **Traffic IP Groups** (トラフィックIPグループ) ページを使用します。

System (システム) > **Fault Tolerance** (フォールトトレランス) > **OSPF Route Health Injection** (OSPFルートヘルスインジェクション) の各設定により、Traffic Manager クラスタ内のOSPFv2の接続性をコントロールできます。

- **ospfv2!enabled:** OSPFv2ルートヘルスインジェクションを有効化するかどうかを設定します。
- **ospfv2!area:** Traffic Manager クラスタが動作するOSPFv2エリアです。10進フォーマットまたはIPv4アドレスフォーマットを使用できます。
- **ospfv2!area_type:** Traffic Manager クラスタが動作するOSPFv2エリアのタイプです。OSPFv2では、特定エリア内の全てのルータが同じタイプを使用する必要があります。
- **ospfv2!hello_interval:** Traffic Manager がOSPFv2の「Hello」パケットをネットワークに送信する間隔です。
- **ospfv2!router_dead_interval:** Traffic Manager がサイレントなルータダウンを宣言する前に経過しなければならない秒数です。



OSPFv2を無効化したり、Traffic Managerの動作を停止させたりすると、隣接する全てのルータにMaxAgeリンクステートアドバタイズメント(LSA)が送信されることによって、まずトラフィックIPアドレスのアドバタイズメントが停止されます。

Traffic Managerは、2つの別個のOSPFv2認証キー/シークレットのペア(「a」および「b」)の使用をサポートしています。このメカニズムによって、キー/シークレットの1つのペアを交換しなくなるときに、ネットワークインフラストラクチャ全体で認証を保持することができます。使用中の認証キーを更新するときは、新しいキーのプロビジョニングがネットワーク全体で完了するまで、古いキーをアクティブのまま保つことができます。

Traffic Managerは、0 (何もしない) と2 (MD5) のタイプのOSPFv2認証をサポートしています。MD5は、アクティブなキーが1つ以上設定されているときに使用されます。

- **ospfv2!authentication_key_id_a:** OSPFv2認証のキーIDです。キーを無効化するには、0に設定します。
- **ospfv2!authentication_shared_secret_a:** OSPFv2認証の共有シークレット(MD5)です。キーを無効化するには、空白に設定します。

- **ospfv2!authentication_key_id_b**: OSPFv2認証のキーIDです。キーを無効化するには、0に設定します。
- **ospfv2!authentication_shared_secret_b**: OSPFv2認証の共有シークレット(MD5)です。キーを無効化するには、空白に設定します。



両方のキーペアを0 (無効)に設定すると、Traffic ManagerはOSPFv2認証を使用しません。

Traffic Managerは、汎用OSPFv2ルータとしては機能しなくなります。

OSPFv2のIPアドレスの設定

デフォルトでは、クラスタ内の各Traffic Managerは、それ自体のIPv4デフォルトゲートウェイへの接続に使用されているアドレスおよびサブネットを使用して、OSPFv2のルーティングドメインに組み込まれます。この動作をオーバーライドするには、管理UIの**flipper!ospfv2_ip**を使用して、個々のクラスタメンバに対して代替のIPv4アドレスを設定します。

ここで指定する個々のIPアドレスは、各Traffic Managerによって、Traffic Managerが通知しているトラフィックIP用の隣接するルータにインストールされているOSPFv2ルートに対し、ネクストホップアドレスとして使用されます。

Traffic Managerのルーティングソフトウェアを無効化するには、**flipper!ospfv2_ip**を「0.0.0.0」に設定します。

隣接モニタリングの設定

Traffic Managerはルーティングネットワークのポーリングを10秒ごとに行い、隣接データを収集しています。このデータには、Traffic Managerがピアとして設定されているルータのリストが含まれています。OSPFの場合、このリストは「ネイバーリスト」と言います。

隣接モニタリングを実行するためには、Traffic Managerがトラフィックを受信することが想定される、隣接するルータのIPアドレスリストを使用して、クラスタ内の各Traffic Managerを設定します。このリストと発見されたネイバーリストが比較されることにより、ルーティングネットワークのヘルス判定が行われます。

想定されるネイバーを設定するには、スペースまたはカンマで区切られたIPv4アドレスのリストを**flipper!ospfv2_neighbor_addrs**に入力します。各Traffic Managerが接続されているデフォルトのIPv4ゲートウェイを指すには、デフォルト値の%gateway%を使用します。

特定のTraffic Managerに対する隣接モニタリングを無効化するには、**flipper!ospfv2_neighbor_addrs**を空白のリストに設定します。

想定されるルータの一部がピアに設定されていない場合、Traffic Managerはイベントログに警告を報告します。また、想定されるルータが1つもピアに設定されていない場合、Traffic Managerはエラーを報告します。管理UIおよびDiagnose (診断) ページのアプレットには、どちらも最後のポーリング時の状態が反映されます。



OSPFの設定を変更すると、ピアの確立に伴う遅延により、警告またはエラーが一時的に表示されることがあります。

Traffic Manager管理インターフェースの 主要機能

本章では、Traffic Managerの管理UIの主な機能について説明します。

ホームページ

Traffic Manager管理インターフェースのホームページには、主な設定、トラフィック、およびステータスの概要が動的に更新されて表示されます。

- 「Status Applet (ステータスアプレット)」には、クラスタのヘルスステータス、トラフィックチャート、および概要がリアルタイムに表示されます。何らかの問題が検出されると、このアプレット内のステータスライトが緑色からオレンジ(重要性の低い問題)または赤(深刻な問題)に変化します。ステータスライトをクリックすると、**Diagnosis** (診断)ページに直接移動します。
- 「Login Information Banner (ログイン情報バナー)」には、これらのクレデンシャルを使用して前回正常にログインしたときのレコードが表示されます。ここには、これまでに失敗したログイン試行も示されます。
- 「Traffic Managers」セクションには、クラスタ内のトラフィック管理デバイスのステータスが表示されます。いずれかのデバイスで障害が発生すると、そのデバイスがハイライトされ、エラーメッセージが表示されます。
- 「Services (サービス)」セクションには、管理しているサービスのリストと、各サービスで使用しているプロトコル、ポート、仮想サーバ名、および全プールのリストが表示されます。サービスは、**Stop/Start** (停止/開始)ボタンで停止または開始することができます。仮想サーバまたはプールで何らかの問題が検出されると、該当する仮想サーバまたはプールがハイライトされます。
- 「Event Log」(イベントログ)セクションには、クラスタ内のTraffic Managerシステムから送信された最近のイベントログメッセージが表示されます。**Examine Logs** (ログ確認)をクリックすると、イベントログ全体が表示されます。詳細については、[イベントおよび監査ログ](#)を参照してください。

ホームページは自動的に更新され、新しいイベントログメッセージや、Traffic Manager、仮想サーバ、またはプールのステータスの変化が表示されます。



マルチクラスタ管理モードでは、設定されているロケーションが表示されるLocationsセクションなどが追加されるため、それに合わせてTraffic Managerのホームページのレイアウトが変わります。詳細については、[マルチサイトクラスタの管理](#)を参照してください。

Services (サービス) > Configuration Summary (設定概要)

多数の仮想サーバ、プール、およびルールを設定すると、設定はすぐに複雑になってしまいます。「Config Summary (設定概要)」には、設定した項目とその相互関係の概要が1ページにまとめて表示されます。

設定概要を表示するには、**Services (サービス) > Config Summary (設定概要)**をクリックします。このページには、サービスがテーブル形式で表示されます。初期状態では、サービスがポートおよび仮想サーバごとにグループ化され、各サービスの横にそのサービスで使用されている各項目が表示されます。

このテーブルは、特定の列や設定オブジェクトをキーにして並べ替え、その項目に依存する設定オブジェクトを分かりやすく表示することができます。例えば、**Pool (プール)**列をクリックすると、各プール名と、その横にそのプールに含まれるノード、そのプールを使用する仮想サーバおよびルールが表示されます。

これにより、設定が複雑であっても、設定されている各項目がどこで使用されているかを正確にトラッキングできます。

設定アーカイブのエクスポート

Export Configuration (設定のエクスポート)オプションを使用して、現在実行中の設定を含むファイルをローカルマシンにダウンロードします。以前に保存した設定のアーカイブファイルを復元するには、**System (システム) > Backups (バックアップ)**ページを使用します。詳細については、[System \(システム\) > Backups \(バックアップ\)](#)を参照してください。

注記

このバックアップは安全に保管してください。このバックアップにはSSL証明書などの機密情報が含まれています。

設定ドキュメントのエクスポート

Traffic Managerには設定インポータと呼ばれるツールが含まれ、システム設定全体を以前作成された定義 (一般的にはDockerやKubernetesなどのオーケストレーションツール内)からTraffic Managerインスタンスにインポートするために使用します。そのような定義は、*configuration documents* (設定ドキュメント)というテキストファイル内に格納され、設定インポータがそれらのドキュメントを使用して前に実行中の設定を入れ替えながらローカルのTraffic Managerに定義を構築します。

現在実行中の設定に基づいて設定ドキュメントを作成およびエクスポートするには、「設定ドキュメントのエクスポート」セクションを使用します。

この機能の詳細な説明については、*Pulse Secure Virtual Traffic Manager: 設定インポータガイド*を参照してください。

カタログ

カタログは、トラフィックを管理するために使用できるオブジェクトのセントラルリポジトリです。以下のタイプのオブジェクトを保存できます。



一部のカタログタイプは、ライセンスに応じて異なることがあります。詳細については、IvantiのWebサイト(www.ivanti.com)を参照してください。

- **ロケーションカタログ:** グローバル負荷分散(GLB)、およびTraffic Managerのマルチサイトクラスタ管理機能に使用される、ロケーションベースの設定オブジェクトが保持されます。
- **DNSサーバ:** Traffic Managerの権威ドメインネームサーバ(DNS)機能で使用されるゾーンファイルおよびゾーン設定が保持されます。
- **GLBサービスカタログ:** 地理的位置が異なるクライアントの負荷分散を行うためにTraffic Managerが使用するGLBサービス設定が保持されます。
- **サービス検出プラグインカタログ:** バックエンドノードのリモートオートスケールされたリストを取得するために、プールが使用するスクリプトが保持されます。
- **Web Acceleratorカタログ:** Ivanti Web Acceleratorによって使用されるアプリケーションのスコープとプロファイルが保持されます。
- **ルールカタログ:** TrafficScriptおよびRuleBuilderルールが保持されます。
- **Java Extensionsカタログ:** Java Extensionsと、Java Extensionsが必要とする全てのサポートJavaクラスファイルが保持されます。これらのJava ExtensionsはTrafficScriptルールから呼び出すことができます。
- **モニタカタログ:** プール内のノードが正常に動作していることを確認するためのヘルスマニタが保存されます。
- **SSLカタログ:** サーバおよびクライアント証明書、認証局、証明書失効リストなどのSSLリソースが保持されます。
- **認証コードカタログ:** リモートLDAP認証サービスの定義が保持されます。これらのサービスにTrafficScriptからアクセスして、ユーザに関する情報を検索したり、ユーザのパスワードを検証することができます。
- **Kerberosカタログ:** Kerberosのレルムに加わるためにTraffic Managerが必要とする設定リソースが保持されます。
- **SAMLカタログ:** SAMLベースのシングルサインオン認証に加わるためにTraffic Managerが使用するリソースが保持されます。

- **サービスプロテクションクラス:** Traffic Managerが不要なトラフィックをフィルタリングするためのポリシーおよびセキュリティ対策を定義するクラスが保持されます。
- **セッション維持クラス:** クライアント接続のセッション維持情報を管理するクラスが保持されます。
- **帯域クラス:** 帯域の使用を管理するための帯域割り当てが保持されます。
- **サービスレベルモニタリングクラス:** ノードのレスポンス時間と合意されたサービスレベルへの適合状況をモニタリングするクラスが保持されます。
- **レートシェイピングクラス:** 最大リクエストレートを実現するために、リクエストをキューイングおよびレートシェイピングするためのクラスが保持されます。
- **クラウドクレデンシャルクラス:** Traffic ManagerがクラウドプロバイダAPIと通信するために必要な情報が保持されます。
- **追加ファイル:** TrafficScriptルールからアクセスできる(resource.get())を使用)、またはプール設定でエラーメッセージを提供するために使用する追加のファイルとリソースが保持されます。

仮想サーバ、プール、およびルールはそれぞれカタログ内のオブジェクトを参照できます。カタログ内の項目を編集すると、その項目を使用している全てのサービスに変更内容が伝搬されます。

カタログにアクセスするには、トップナビゲーションバーにある**Catalogs** (カタログ)をクリックします。カタログ内の項目を編集または追加するには、該当する**Edit** (編集)リンクをクリックします。

各カタログタイプについては、後で詳しく説明します。

ライセンス管理

Traffic Managerのドキュメントと製品インターフェースは、ライセンスキーの使用を参照して、製品の機能を有効化または無効化することがあります。Ivanti(または指定されたサポート窓口)は、個別にライセンスキーを生成し、通常、ライセンスキーは特定の製品インスタンスまたはサービス契約に紐付けられます。

複数のTraffic Managerインスタンスからなるクラスタでは、クラスタ全体のライセンス要件を対象にした適切なキー(または一連のキー)を入手する必要があります。ライセンスのないTraffic Managerは、コミュニティエディションと同様にパフォーマンスが制限された状態で動作します。詳細については、[Traffic Managerコミュニティエディション](#)を参照してください。

注記

1つのクラスタでライセンスの異なるTraffic Managerを使用することは可能ですが、推奨はされません。クラスタの1つ以上のメンバに対してライセンスがない機能を更新しようとして、クラスタの自動同期に失敗する可能性があるためです。

現在インストールされているライセンスキーを表示するには、**System** (システム) > **Licenses** (ライセンス)をクリックします。このページには、シリアル番号によって識別できる各ライセンスキーと、そのライセンスキーが有効なTraffic Managerが表示されます。製品、有効期限、ライセンスされる一連の機能など、特定のライセンスの全ての詳細を表示するには、折りたたみ表示の矢印をクリックします。

1つのTraffic Managerインスタンスに複数のライセンスがインストールされている場合、そのTraffic Managerでは、以下の順序の基準に基づいて使用するライセンスが選択されます。

- ・ インストール済みのあらゆるライセンスは、デフォルトのライセンスのないコミュニティエディションよりも優先されます。
- ・ 認可されたライセンスは、認可されていないライセンスよりも優先されます。
- ・ リモートに認可されたライセンス(Flexible License Architecture (FLA)タイプのライセンスなど)は、非リモートのライセンス(永続ライセンス)よりも優先されます。
- ・ より機能の豊富なライセンスは、機能の少ないライセンスよりも優先されます。

ライセンスキーの追加と削除

新たにインストールおよび設定されたTraffic Managerの管理UIに初めてアクセスすると、「Unlicensed (未ライセンス)」ページが表示されます。このページで、ライセンスキーをアップロードするか、コミュニティエディションを引き続き使用するかを選択できます。Traffic Managerのハードウェアアプライアンス、仮想アプライアンス、および特定のクラウドインスタンスでは、Initial Configuration (初期設定)ウィザードを使用してライセンスをアップロードすることもできます。詳細については、ご使用の製品の『Pulse Secure Virtual Traffic Manager: Installation and Getting Started Guide』を参照してください。

初回以降は、**System** (システム) > **Licenses** (ライセンス)ページで個々のキーを追加または削除できます。新しいライセンスは、ページ下部の「Install new License Key (新しいライセンスキーのインストール)」セクションを使用してアップロードできます。**Choose File** (ファイルの選択)をクリックして、ライセンスデータが含まれている標準テキストファイルの場所を指定し、**Install Key** (キーのインストール)をクリックします。この操作が正常に完了すると、リストに新しいライセンスが表示されます。

クラスタに存在する不要または無効なキーは、そのライセンスの右にあるチェックボックスを選択して**Remove Selected Keys** (選択したキーの削除)ボタンをクリックすると削除できます。キーを削除するとTraffic Managerのライセンスが無効になるような場合には、それが警告され、警告を無視すると削除が確定されます。確定すると、該当するTraffic Managerがコミュニティエディションとなります。

Pulse Secure Services Directorへの登録

Traffic Managerは、遠隔管理された柔軟なライセンス用に、Services Directorインスタンスへ登録リクエストを送信できます。この機能は自己登録と呼ばれ、一般的には初期設定で有効化されます。このとき、新たに設定したTraffic Managerを登録するServices Directorインスタンスでの詳細を入力します。その後、Services DirectorはTraffic ManagerのREST APIを使用して、設定の更新をクエリおよび発行します。



Pulse Secure Services Directorの詳細については、IvantiのWebサイト(www.ivanti.com)を参照してください。

Services DirectorでTraffic Managerを登録するには、**System** (システム) > **Licenses** (ライセンス) > **Services Director registration** (Services Directorへの登録)をクリックします。このページのフィールドを使用して、Services Directorインスタンスに登録リクエストを手動で発行します。

登録リクエストを開始すると、Traffic ManagerはServices Director管理者がリクエストを承認または拒否するまで待ちます。リクエストが承認されると、Services DirectorはTraffic Managerに適切なライセンスをプッシュします。登録リクエストに関するステータスメッセージを表示するには、イベントログを確認します。

以下の要件に従って、提供されたフォームに登録の詳細を入力します。

フィールド	説明
remote_licensing!registration_server	このTraffic Managerの登録に使用するServices Directorの、ポート番号を含むアドレスです。 Services Director VAについては、サービスエンドポイントアドレスを使用します。
remote_licensing!server_certificate	PEMでエンコードされたフォーマットの、Services Directorのサーバ証明書です。これは、証明書の更新時にServices Directorによって自動更新されます。
remote_licensing!owner	自動受諾を使用している場合に、このインスタンスに関連付けるオーナーです。
remote_licensing!owner_secret	remote_licensing!owner で指定したオーナーのIDを検証するために使用されるオーナーのシークレットです。 このフィールドは、登録リクエストの自動受諾を使用し、Services Directorのvalidate_owners設定がtrueの場合のみ使用されます。

フィールド	説明
remote_licensing!policy_id	自動受諾の使用の試行のための、自動受諾/登録ポリシーIDです。
remote_licensing!comm_channel_enabled	Services Directorとの通信を、通信チャネルエージェントによってチャネル設定するかどうか指定します。 通信チャネルエージェントは、長期的なTLS暗号化ウェブソケット接続を使用し、Traffic ManagerとServices Directorの間の双方向の通信を処理します。これは、Services DirectorによるTraffic ManagerのREST APIとの直接通信機能の妨げとなる、NATゲートウェイやその他のネットワーク境界の背後のTraffic Managerのインスタンスに必要です。例えば、Traffic Managerをコンテナ内にデプロイする場合などです。 Traffic Managerの登録リクエストがServices Directorにより正常に承認されるまで、Traffic Managerイベントログに、通信チャネルエージェントの失敗の警告が表示される場合があります。Traffic Managerは1秒あたり1回の頻度で接続を試行しますが、ログを見やすくするために、接続失敗メッセージは15秒あたり1回の頻度で表示されます。 通信チャネルエージェントの機能をサポートしていないServices Directorへの自己登録リクエストでは、この値は自動で「No」(いいえ)に設定されます。 この機能は、Services Directorとの自己登録を実行するTraffic Managerでのみ利用可能です。Services Directorに手動で追加されたTraffic Managerはこの機能を使用できません。
remote_licensing!comm_channel_port	Services Director上で実行中の通信チャネルエージェントのポートです。デフォルト値は8102です。
remote_licensing!email_address	(オプション)Services Director管理者が連絡を取ることができる電子メールアドレスです。

フィールド	説明
remote_licensing!message	(オプション)登録リクエストに含める自由形式のメッセージで、Services Director管理者に表示されます。 必要に応じて、このフィールドは登録リクエストを識別する追加情報に使用できます。

登録リクエストを送信するには、**Save and Register** (保存して登録)をクリックします。

このセクションの全てのフィールドの更新を保存すると、指定したServices Directorで再登録が自動的に開始されます。設定を変更せずに手動で再登録をリクエストするには、**Re-register** (再登録)をクリックします。



Services Directorに、以前の試行に関わらずに自己登録を最初の登録として扱うよう指示するには、**Force** (強制)チェックボックスをオンにします。

System (システム) > Global Settings (グローバル設定)

Traffic Managerソフトウェアでは、幅広いグローバル設定が可能です。設定にアクセスするには**System** (システム) > **Global Settings** (グローバル設定)をクリックします。グローバル設定は、個々のサービスまたはトラフィックには関連しない、Traffic Manager全体の設定が対象となっており、以下のカテゴリに分かれています。

カテゴリ	説明
System Settings (システム設定)	各Traffic Managerのネットワークおよびシステムの設定を定義します。
EC2 Account Settings (EC2アカウント設定)	Traffic ManagerとAmazon EC2 APIの相互関係をコントロールします。
Cache Settings (キャッシュ設定)	各Traffic Managerの共有キャッシュの動作をコントロールします。
FIPS 140-2 Configuration (FIPS 140-2設定)	「FIPSモード」を有効化するかどうかを決定します。このモードでは、Traffic ManagerはFIPS 140-2準拠の暗号モジュールを使用してトラフィックを処理することができます。
SSL Configuration (SSL設定)	SSLのサポートの動作をコントロールします。
SSL Hardware Support (SSLハードウェアサポート)	Traffic Managerが使用する追加のSSLハードウェアに適用されます。

カテゴリ	説明
Logging (ロギング)	システムイベントおよび仮想サーバのアクセスに対してTraffic Managerがロギングを実行する方法をコントロールします。
State Synchronization Settings (状態同期設定)	クラスタメンバがそれぞれの状態データを共有する方法をコントロールします。
Idle Connection Settings (アイドルコネクション設定)	ノードに対するアイドルHTTPコネクションの管理方法をコントロールします。
Java Extension Settings (Java Extension設定)	Java Extensionプロセスの開始および管理方法をコントロールします。
Login and Security Settings (ログインおよびセキュリティ設定)	ユーザのログイン動作をコントロールします(ログインセキュリティとログイン行動 を参照)。
Web Accelerator	Ivanti Web Acceleratorのグローバルプロパティをコントロールします。
Enterprise Authentication (エンタープライズ認証)	サービスのSAMLベースのエンタープライズ認証の動作をコントロールします(サービスに対する認証の提供 を参照)。
Telemetry (テレメトリ)	オプションの、Ivantiへの匿名による使用状況データのレポートをコントロールします。このデータは製品の改善と、Ivantiの技術サポート能力の向上に役立ちます。
Other Settings (その他の設定)	Traffic Managerのその他の設定をコントロールします。

各カテゴリの横にある矢印をクリックすると、そのカテゴリが展開され、オプションが表示されます。変更を完了したら、**Update** (更新)ボタンをクリックします。**Restore Defaults** (デフォルトに復元)ボタンを使用すると、全てのグローバル設定をシステムのデフォルト値に戻すことができます。

注記

これらの設定を変更するときは注意が必要です。デフォルト値は、幅広いシステムで良好な性能が得られ、ソフトウェアの安定性とセキュリティが維持されるよう選択されています。これらの設定を不適切な値に設定すると、システムの安定性とセキュリティが損なわれ、性能に影響を与える可能性があります。

System (システム) > Backups (バックアップ)

Traffic Managerのバックアップ管理機能を使用すると、設定をバックアップしたり、バックアップを相互に、また現在の設定と比較したり、以前のバックアップから設定を復元したりできます。

設定のバックアップを管理するには、**System** (システム) > **Backups** (バックアップ)をクリックします。

また、「zconf」コマンドラインユーティリティを使用すると、個々の設定オブジェクトについてさらにきめ細かいバックアップ/インポート/エクスポートを実行できます。詳細については、[zconfを使用したきめ細かい設定のインポート/エクスポート](#)を参照してください。

設定を大幅に変更する前に設定をバックアップすることをお奨めします。そうすることにより、以下のような操作が可能になります。

- 設定を比較し、変更した内容を正確に把握する。
- 変更を破棄して以前のバックアップを復元する。

この機能を使用すると、旧式の仮想マシンまたはクラウドインスタンスを新しいバージョンに置き換える場合などに、Traffic Managerマシン間で設定のバックアップをコピーすることもできます。

注記

設定バックアップファイルは、そのファイルが作成されたTraffic Managerに固有であるため、クラスタ内のTraffic Manager間で自動的に複製される設定には含まれません。特定のクラスタメンバを終了または削除した場合、そのメンバに保存した全ての設定バックアップは失われます。

バックアップの作成

設定のバックアップを作成するには、「Create a Backup (バックアップ作成)」フォームを使用します。作成したバックアップは、ローカルのTraffic Managerマシンに保存されます。

ご使用のTraffic ManagerのデプロイでWeb Application Firewallのライセンス認証および有効化が行われている場合、**App Firewall**ボックスを選択することにより、バックアップにオプションでApplication Firewallの設定を含めることができます。

バックアップの復元

Backup Management (バックアップ管理)ページのテーブルでバックアップ名をクリックします。**Restore Configuration** (設定復元)オプションを使用して、現在の設定をバックアップの設定内容に置き換えます。

設定のバックアップには、トラフィックIPグループなど、バックアップ元となったTraffic Managerのマシン固有の情報が含まれています。そのため、この時点で、現在のTraffic Managerのローカルマシン設定を置き換えるかどうかを決定する必要があります。以下の2つの方法があります。

- バックアップに含まれるマシン固有の設定を使用する。

復元するバックアップの作成元Traffic Managerが同じ場合、ローカル設定をバックアップの設定で単純に上書きします。バックアップ元のTraffic Managerが異なる場合は、マッピングコントロールが提示されます。ここで、バックアップに保存されているマシン設定のマッピングを指定することができます。

This backup contains machine specific information, such as networking configuration and Traffic IP groups. Do you want to:

- ☒ Restore backup with machine specific information according to the following mapping...

Original Traffic Manager		New Traffic Manager
dev-vm08-0	→	dev-vm08-2
dev-vm08-4	→	Do not restore this machine's config

- ☐ Restore backup without any machine specific information.

この例のバックアップには、2台のTraffic Managerから成るクラスタの設定が保持されています。このツールでは、どのTraffic Managerの設定を復元し、どのTraffic Managerの設定を無視するかを決定できます。マシン設定間に作成できるのは1対1のマッピングのみであることに注意してください。

- マシン固有の情報を使用せずにバックアップを復元する。

バックアップに保存されているマシン固有の設定を無視し、現在のTraffic Managerのローカルマシン設定を維持します。

ご使用のTraffic ManagerのデプロイでWeb Application Firewallのライセンス認証および有効化が行われている場合、Include Application Firewall configuration (Application Firewallの設定を含める)をクリックすることにより、バックアップからオプションでWeb Application Firewallの設定も復元することができます(設定のバックアップが存在する場合)。

バックアップのエクスポート

設定のバックアップは、ローカルのTraffic Managerマシンに保存されます。設定のバックアップをエクスポートすると、保管しておくことができます。

Backup Management (バックアップ管理)ページのテーブルでバックアップ名をクリックします。**Export Configuration** (設定のエクスポート)オプションを使用して、設定のバックアップをローカルマシンにダウンロードします。

注記

このバックアップは安全に保管してください。このバックアップにはSSL証明書などの機密情報が含まれています。

設定ドキュメントとしてのバックアップのエクスポート

Traffic Managerには設定インポータと呼ばれるツールが含まれ、システム設定全体を以前作成された定義 (一般的にはDockerやKubernetesなどのオーケストレーションツール内)からTraffic Managerインスタンスにインポートするために使用します。そのような定義は、*configuration documents* (設定ドキュメント)というテキストファイル内に格納され、設定インポータがそれらのドキュメントを使用して前に実行中の設定を入れ替えながらローカルのTraffic Managerに定義を構築します。

Backup Management (バックアップ管理)ページのテーブルでバックアップ名をクリックし、「Export Configuration document (設定ドキュメントのエクスポート)」セクションを使用して、バックアップに含まれた設定に基づいて設定ドキュメントを作成およびエクスポートします。

 現在実行中の設定に基づいて設定ドキュメントを作成するには、**Services** (サービス) > **Configuration Summary** (設定概要)ページを使用します。

この機能の詳細な説明については、*Pulse Secure Virtual Traffic Manager: 設定インポータガイド*を参照してください。

バックアップのインポート

以前エクスポートした設定のバックアップをインポートすることができます。設定のバックアップをローカルマシンからアップロードするには、「Import a Backup (バックアップのインポート)」セクションを使用します。

インポートしたバックアップは、Traffic Managerマシンの設定のバックアップリストに追加されます。バックアップをインポートするだけでは、現在の設定は置き換わりません。必要に応じて、アップロードしたバックアップを復元することにより、現在の設定を置き換えることができます。

System (システム) > Backups (バックアップ) > Partial Backups (部分バックアップ)

このセクションで利用できるバックアップ/復元機能は完全バックアップのものと同じですが、バックアップに設定全体のサブセットのみが含まれるよう調整することができます。このセクションには、**System** (システム) > **Backups** (バックアップ)ページのリンクからアクセスできます。

 部分バックアップは、Traffic Managerソフトウェアの異なるバージョン間で転送できません。つまり、あるバージョンのTraffic Managerで作成した部分バックアップを他のバージョンにインポートすることはできません。

この機能には、[zconfを使用したきめ細かい設定のインポート/エクスポート](#)で説明するzconfコマンドラインユーティリティと同様の機能があり、特定のサービスを他のTraffic Managerクラスタにコピーする場合に役立ちます。通常のインポートとは異なり、インポートした設定は、既存の設定を置き換えるのではなく、既存の設定にマージされます。

このページは、インポートおよびマージプロセスを処理するためのセクションと、部分バックアップをエクスポートするためのセクションに分かれています。

Importing (インポート)

このセクションでは、部分バックアップまたは完全バックアップをインポートすることができます。表示される **Include Only** (指定の値のみを含む) テキストボックスに適切なフィルタを指定することにより、アップロードされたバックアップファイルから簡単に部分インポートを行うことができます。システムでは、このフィルタに一致するオブジェクトと、選択されたオブジェクトが依存するオブジェクトのみがインポートされます。例えば、選択された仮想サーバが使用しているプールもインポートされます。使用されるフォーマットの詳細については、[Filter formats \(フィルタフォーマット\)](#)を参照してください。

バックアップファイルをアップロードし、オプションでフィルタを指定すると、システムに対してどのような変更が行われるかを示す「Diff」(差異のリスト)が提示されます。このDiffの内容を見直し、間違いがないことを確認してください。**Apply Partial Backup** (部分バックアップの適用)をクリックして変更内容を適用します。

注記

部分バックアップをインポートするときに適切に処理を行わないと、設定に矛盾が生じる可能性があります。部分バックアップをインポートするときは、提示されるDiffの内容を必ず見直してください。全てのバックアップおよび復元操作と同様に、部分バックアップを適用する前に、完全バックアップポイントを作成することを強くお勧めします。

Exporting (エクスポート)

このセクションは、設定の部分バックアップまたは完全バックアップをダウンロードするときに使用します。表示される **Include Only** (指定の値のみを含む) テキストボックスに適切なフィルタを指定することにより、部分バックアップを簡単に行うことができます。システムでは、このフィルタに一致する設定オブジェクトと、そのオブジェクトが依存するオブジェクトのみがエクスポートされます。使用されるフォーマットの詳細については、[Filter formats \(フィルタフォーマット\)](#)を参照してください。

Filter formats (フィルタフォーマット)

Include Only (指定の値のみを含む) フィルタは、以下のいずれかのフォーマットに従って記述された、スペースで区切られたエントリのリストです。

[CONFIGURATION TYPE]

例: vservers

または

[CONFIGURATION TYPE] / [NAME]

例: vservers/Intranet

または

[CONFIGURATION FILE]

例: users



設定名にスペースが含まれる場合は、各スペース文字の前にフォワードスラッシュ(\)を挿入してください。これにより、スペースが区切り文字としてではなく名前の一部として扱われます。例えば、Intranet Master Serviceの場合はIntranet\ Master\ Serviceと入力します。



特殊な設定ファイルsettings.cfgを含めることにより、グローバル設定(**System** (システム) > **Global Settings** (グローバル設定))をバックアップできます。また、設定ファイルusersを含めると、ローカルユーザ設定(**System** (システム) > **Users** (ユーザ))をバックアップすることができます。

例えば、全ての仮想サーバ、「Primary Database Monitor」モニタ、およびグローバル設定をインポートする場合は、以下のようにフィルタを入力します。

Backup:

Choose File ztm_config_...02_1738.tar

Include Only:

vservers
monitors/Primary\ Database\ Monitor
settings.cfg

Upload and review partial backup

以下の表に、フィルタに含めることのできる設定識別子を示します。

識別子	説明
actionprogs	アクションプログラム
actions	アラートアクション
activitymonitor	現在のアクティビティグラフデータ設定
appliance	アプライアンス管理
optimizer/profiles	Web Acceleratorプロファイル
optimizer/scopes	Web Acceleratorスコープ

識別子	説明
auth	管理インターフェース/管理サーバ認証コード
authenticators	TrafficScriptリモート認証コード
ApplicationFirewallConfig	Web Application Firewallの設定
auth	認証コード
bandwidth	帯域クラス
cloudcredentials	クラウドクレデンシャル
custom	カスタム設定のセット
dnsserver/zonefiles	DNSサーバゾーンファイル
dnsserver/zones	DNSサーバゾーン
events	アラートイベントタイプ
extra	その他のファイル
flipper	トラフィックIPグループ
groups	ユーザグループ
jars	TrafficScriptのJava Extensions
licensekeys	ライセンスキー
locations	マルチサイト管理およびグローバル負荷分散の地理的位置
monitors	モニタ
persistence	セッション維持クラス
pools	プール
protection	サービスプロテクションクラス
rate	レートシェイピングクラス
rules	TrafficScriptルール
scripts	モニタスクリプト

識別子	説明
services	グローバル負荷分散(GLB)サービス
servlets	Java Extension Servlet
slm	サービスレベルモニタリングクラス
ssl/cas	SSL認証局
ssl/client_keys	SSLクライアント証明書/キー
ssl/dnssec_keys	DNSSECキー
ssl/server_keys	SSLサーバ証明書/キー
supplementarykeys	追加ライセンスキー
vservers	仮想サーバ
zxtms	Traffic Manager

部分バックアップへのWeb Application Firewallの取り込み

Web Application Firewall (WAF)を部分バックアップに含めるには、グローバル設定ファイルも含める必要があります。さらに、バックアップを作成する前に、Traffic Managerの設定でWAFを有効化する必要もあります。

すなわち、WAFを有効化した状態で、以下を「Export (エクスポート)」セクションの**Include Only** (指定の値のみを含む)フィルタに追加します。

- settings.cfg
- ApplicationFirewallConfig

同様に、WAF設定をPartial Backups (部分バックアップ)ページからインポートする場合も、「Import (インポート)」セクションの**Include Only** (指定の値のみを含む)フィルタに両ファイルを追加してください。

注記

この方法でアクティブなTraffic Managerの設定に「settings.cfg」ファイルを復元すると、WAF固有のキー値だけでなく、全てのグローバル設定がオーバーライドされます。部分バックアップでこのファイルのコピーを復元する前に、まずファイルのバックアップを作成することをお奨めします。

アクティビティモニタリング

システム上の現在および過去のアクティビティをトラッキングすることのできるビジュアルトラフィックモニタリングツールがいくつか用意されています。これらのツールにアクセスするには、管理UIのトップバーにある**Activity** (アクティビティ) ボタンをクリックします。

Activity (アクティビティ) > Current Activity (現在のアクティビティ)

「Current Activity (現在のアクティビティ)」ページには、システム上のトラフィックアクティビティを示すグラフがリアルタイムに表示されます。このページに表示されるデータは幅広くカスタマイズできます。

標準データセットがいくつか用意されており、これらを選択すると、現在の帯域幅などのトラフィック値、Traffic Managerが処理している接続数、空きメモリや予備CPU時間などのハードウェア値などがプロットされます。この情報の多くは、仮想サーバ、プール、またはノードごとに分割することができます。また、グラフのサイズやタイムスケールを指定したり、軸タイプを線形軸と対数軸から選択したり、パイチャートを選択したりすることも可能です。さらに、グラフの対象を全てのマシンにすることも、一度に1台のマシンにすることもできます。

設定を選択したら、**Plot Data** (データをプロット) をクリックするとグラフが表示されます。データを .tsv (タブ区切り変数) ファイルとしてダウンロードして、解析に利用することもできます。

グラフのデータをさらにカスタマイズするには、**Change Data** (データ変更) ボタンをクリックします。これにより **Current Activity** (現在のアクティビティ) > **Edit** (編集) ページが表示され、このページでプロットするデータを指定することができます。

「Global Values (グローバル値)」セクションでは、モニタリングする統計値を選択し、システム全体のデータをプロットすることができます。また、「Virtual Server Values (仮想サーバ値)」、「Pool Values (プール値)」、および「Node Values (ノード値)」セクションでは、データをこれらの各インスタンスに分割してプロットすることができます。データ転送および接続カウントのほか、DNSおよびSNMPリクエストデータ、コンテンツ圧縮、およびHTTP書き換えを確認することができます。

「Service Protection Class Values (サービスプロテクションクラス値)」セクションでは、サービスプロテクションクラスによって拒否されたリクエストに関する情報をプロットできます。サービスプロテクションクラスの全てまたはサブセットのデータを表示できます。また、単一のIPアドレスによる接続カウントが高すぎる、リクエストに許可されていないコンテンツが含まれるなど、拒否理由ごとにデータを分割することもできます。

プロットの設定を選択したら、後で使用するために保存することができます。完了したら、**Apply** (適用) をクリックします。

SNMPによる性能のモニタリング

簡易ネットワーク管理プロトコル(SNMP)は、ネットワークのモニタリングおよび管理に関するオープン規格です。SNMPを使用すると、ネットワーク上のデバイスをモニタリングし、性能データを収集することができます。

Traffic Managerは複数のSNMPバージョン(v1、v2c、およびv3)をサポートしています。

Traffic Managerでは、HP OpenView™などのSNMPツールを使用して性能をモニタリングできます。また、トラフィックフローに関する情報を抽出して、解析に利用することもできます。



Traffic Managerは、SNMPトラップと呼ばれる非同期SNMP通知を送信することもできます。詳細については、[イベント処理およびアラート](#)を参照してください。

SNMPを設定するには、**System** (システム) > **SNMP**をクリックします。ここでは、選択したSNMPバージョンに関連するTraffic Manager MIB (マネジメントインフォメーションベース)をダウンロードできます。MIBはSNMPが使用する情報のデータベースで、以下の2つのSMI (管理情報構造)バージョンがあります。

- **SMIv1:** SNMPv1クライアントのみ
- **SMIv2:** SNMPv2cおよびv3クライアントのみ

いずれかのリンクをクリックしてブラウザにMIBを表示するか、右クリックしてコンテンツをコンピュータ上のファイルに保存します。

一般的なSNMP設定

SNMPを有効化するには、**snmp!enabled**を「Yes (はい)」に設定し、**Update** (更新)をクリックします。

SNMPサービスで監視するポートおよびバインドIPアドレス、サービスへのアクセスを許可するクライアントなど、一般的な設定を行うことができます。

SNMPの標準ポートは161ですが、非特権ポートを使用する場合は、1161を使用することをお奨めします。

SNMPv1およびSNMPv2cの設定

以下の設定を行うことができます。

設定	説明
snmp!community	SNMPサーバが正常に動作するためには、適切なコミュニティストリングが必要です。コミュニティストリングはパスワードに相当し、これによって、認可されたSNMPリクエストかどうかを識別します。ただし、コミュニティストリングはSNMPリクエストでプレーンテキストとして送信され、セキュアではないため、機密情報をコミュニティストリングとして使用しないでください。多くのSNMPクライアントでは、デフォルト値として「public」が使用されます。Traffic ManagerはSNMPサービスへのアクセスを制限できるため、この秘密情報の強度は重要ではありません。

設定	説明
	コミュニティストリングを空白/空のままにすると、全てのSNMPv1およびSNMPv2cコマンドが拒否されます。

SNMPv3の設定

このセクションでは、SNMPv3コマンドを受け入れて、それに対するレスポンスを返すための認証およびプライバシー設定を指定できます。



パスワードから暗号化キーを生成するために使用される要素の1つに「エンジンID」があります。Traffic Managerは、Zeus¹エンタープライズOID (7146)と、Traffic Managerが稼働しているマシンの最初のMACアドレスを組み合わせることでこの値を生成します。このセクションには、接続しているTraffic Managerの正しいエンジンID値が表示されます。

全てのSNMPv3通信は、以下に示すユーザ設定と照合され、セキュリティパラメータが正確に一致していないコマンドは拒否されます(ただし、認証されることがなく、空のユーザ名で送信される可能性のあるsnmpEngineID検出メッセージは許可され、それに対するレスポンスが返されます)。

以下の設定を行うことができます。

設定	説明
snmp!username	全てのSNMPv3コマンドおよび通信に必要となるSNMPv3ユーザのユーザ名です。空白/空に設定すると、全てのSNMPv3コマンドが拒否されます。 Traffic Managerでは、単一のSNMPv3ユーザアカウントが定義されていることが要求され、このアカウントが許可されます。このユーザの認証およびプライバシー設定は、以下に示すフィールドによって定義され、RFC3414 (http://www.ietf.org/rfc/rfc3414.txt)で規定される「User-based Security Model」に基づきます。
snmp!security_level	SNMPv3ユーザによる、認証を使用するかどうか、またプライバシーを使用するかどうかの選択です。SNMPv3コマンドはこの選択に従う必要があります。従わないコマンドは拒否されます。SNMPv3レスポンスはこの選択に従います。

¹ Zeusは、Traffic Managerの基盤となった製品の名前です。

設定	説明
	ここでAuthentication (認証)を選択すると、コマンドおよびレスポンスは、snmp!auth_passwordから導き出されたキーを使用して暗号化された署名がなされ、この署名がメッセージに組み込まれます。不正に署名されたコマンドまたはレスポンスは検出され、拒否されます。これは、パスワードの機密性が維持されている限り、信頼されたパーティへのなりすましを防ぐことを目的としています。 ここでさらにPrivacy (プライバシー)を選択すると、コマンドおよびレスポンスの一部が、snmp!priv_passwordから導き出されたキーを使用してDESで暗号化されます。これは、パスワードの機密性が維持されている限り、盗聴者によるメッセージコンテンツの読み取りを防ぐことを目的としています。
snmp!hash_alg	SNMPv3ユーザによる、暗号ハッシュアルゴリズムの選択です。暗号ハッシュアルゴリズムはSNMPv3コマンドで要求され、レスポンスに使用されます。 サポートされているハッシュアルゴリズムはMD5およびSHA-1です。詳細については、RFC 3414 (http://www.ietf.org/rfc/rfc3414.txt)を参照してください。 snmp!security_levelを「No Authentication, No Privacy (認証なし、プライバシーなし)」に設定した場合、この設定は使用されません。
snmp!auth_password	SNMPv3ユーザの認証パスワードです。 これは共有秘密情報です。SNMPクライアントユーザアカウントに同じ認証パスワードを設定する必要があります。 SNMPv3通信の認証キーは、このパスワードから導き出されます。 snmp!security_levelを「No Authentication, No Privacy (認証なし、プライバシーなし)」に設定した場合、この設定は使用されません。
snmp!priv_password	SNMPv3ユーザのプライバシー(メッセージ暗号化)パスワードです。 これは共有秘密情報です。SNMPクライアントユーザアカウントに同じプライバシーパスワードを設定する必要があります。

設定	説明
	SNMPv3通信のプライバシキーは、このパスワードから導き出されます。セキュリティ上の理由から、認証パスワードと同じパスワードを使用しないことをお奨めします。 snmp!security_levelを「No Authentication, No Privacy (認証なし、プライバシなし)」または「Authentication only (認証のみ)」に設定した場合、このパスワードは使用されません。

性能およびイベントモニタリングのカスタマイズ

TrafficScript関数のcounter.increment()を使用すると、10個の内蔵カウンタのいずれかを増加させることができます。

SNMPを使用してこれらのカウンタの値を読み込み、**アクティビティモニタ**にグラフとして表示することができます。

これらのカウンタは、TrafficScriptで検出したカスタムイベントを記録するために使用できます。以下に例を挙げます。

- TrafficScriptルールで認証を実行すると、認証の失敗件数をカウントできます。
- 特定のリクエストクラスのリソース使用量が特に多い場合に、リクエストが行われた回数のチャートを作成することができます。
- 例えば既知のWebワームシグネチャについてセキュリティチェックを実行すると、そのシグネチャが検出された回数をカウントし、チャートとして表示することができます。

内蔵SNMPカウンタの値を増加させるには、**System (システム) > Global Settings (グローバル設定) > Other Settings (その他の設定)**にある**snmp_user_counters**値を変更します。

Activity (アクティビティ) > Historical Activity (ヒストリカルアクティビティ)

デフォルトでは、Traffic Managerは、過去90日間のシステムのアクティビティを記録します。データが保存される日数を変更するには、**System (システム) > Global Settings (グローバル設定) > Other Settings (その他の設定)**をクリックし、**std!days**に格納された数値を変更します。0の値は、データを永続的に保管することをTraffic Managerに指示します。

ヒストリカルシステムアクティビティを表示するには、**Activity (アクティビティ) > Historical Activity (ヒストリカルアクティビティ)**をクリックします。

仮想サーバ、プール、またはノードごとに表示される1分あたりのヒット数または1秒あたりの処理ビット数のグラフを選択します。Traffic Managerには、1時間から1年まで、プロットするデータに対するタイムスケールの範囲が用意されています。ただし、利用できるオプションは、**std!days**で設定した値によって影響を受けます。

システム内の全ての仮想サーバ、プール、ノード、または選択したサブセットに対してデータをプロットすることを選択します。また、チャートサイズと軸タイプも選択します。

Plot Data (データのプロット)をクリックし、現在の設定を使用してグラフを表示します。データを「.tsv」(タブ区切り変数)ファイルとしてダウンロードして、解析に利用することもできます。

Activity (アクティビティ) > Map (マップ)

このマップには、トラフィック管理システムが処理しているリクエストの送信元の地理的位置が表示されます。

マップはインタラクティブに設計されており、マップ上を移動したり、インフラストラクチャに必要な詳細レベルにズームインしたりするためのツールが用意されています。マウスの左ボタンを押しながらマップを所要の目的地にドラッグすることにより、任意の地理的位置を中央に表示することができます。

マップ上に表示される小さいドットは、特定のサービスにリクエストを送信したユーザを表します。

 ユーザの位置に対応するアクティビティをマップに表示できるようにするために、Traffic ManagerではGeolPデータベースへのアクセスが要求されます。詳細については、[GeolPデータベースの使用](#)を参照してください。

Activity (アクティビティ) > Connections (コネクション)

 このセクションで説明する機能は、それぞれライセンスによって管理されています。一部のライセンスでは基本機能モードのみが提供されます。このモードでは、単純なコネクション情報のみが表示され、カスタマイズオプションが少なく、利用できるフィルタは1つのみです。フルモードは、ここで説明する全ての機能をサポートし、「advanced real-time analytics」ライセンスオプションで提供されます。このオプションのご購入については、サポート窓口にお問い合わせください。

基本モード

Connections (コネクション)ページでは、Traffic Managerが処理している現在および最近のコネクションを、日付順で(最も新しいものから順に)モニタリングできます。これは、特定時点におけるコネクションのスナップショットです。**Refresh Snapshot** (スナップショットを更新)ボタンを押すまで、それ以降のコネクションは表示されません。

コネクションテーブルの左上にある[+]ボタンを使用すると、フィールド列の表示/非表示を切り替えることができます。このボタンをクリックすると、現在テーブルに表示されているフィールドのドロップダウンリストが表示されます。このリストで各フィールドをクリックすると、そのフィールドの表示/非表示が切り替わります。

表示されているデータに単純なフィルタを適用して、特定ノードへのコネクションのみを表示することができます(「To」列)。これは、フルモードで利用可能なフィルタ機能の単一フィールドバージョンです。次のセクションで説明する使用手順を参照してください。

フルモード

このモードでは、基本モードで提供される全ての機能に加え、コネクションの詳細およびカスタマイズを利用できます。ソート基準にする列のヘッダをクリックすると、表示されているデータをソートすることができます。ソートされた列の横にある小さい矢印は、ソート順を示します。この矢印をクリックして、昇順と降順を切り替えることができます。

ノードフィルタに加え、テーブル内の任意のフィールドまたはデータでコネクションをフィルタすることができます。これを行うには、ページの上部にあるフィルタセクションのドロップダウンボックスを使用するか、テーブル内のいずれかのデータ値を直接クリックします。後者の方法では、インラインの「add filter」ダイアログが表示され、選択したフィールドの値を共有していない全てのコネクションを除外することができます。

例えば、特定のIPアドレスとポートから送信され、結果的にHTTPレスポンスコード200が生成されたリクエストを表示するとします。この場合、まず「From」列で、使用するIPアドレスをクリックします。選択したIPが入力された状態でフィルタポップアップが表示されます。

	Traffic Manager	From	To	VS	Pool	Resp. Code	Request
	coeus.cam.zeus.com	10.100.1.14:37077	92.52.65.222:80	website	website	200	/favicon.ico
	coeus.cam.zeus.com	10.100.1.14:37074	92.52.65.213:80	website	website	200	/assets/img/bullet.gif
	coeus.cam.zeus.com	10.100.1.14:37076	92.52.65.213:80	website	website	200	/assets/default/Site/en/images/tri/triboo.gif
	coeus.cam.zeus.com	10.100.1.14:37072	92.52.65.222:80	website	website	200	/assets/default/Site/en/images/gil/gilt.gif
	coeus.cam.zeus.com	10.100.1.14:37074	92.52.65.213:80	website	website	200	/assets/default/Site/en/images/dom/dominos.gif
	coeus.cam.zeus.com	10.100.1.14:37072	92.52.65.222:80	website	website	200	/assets/default/Site/en/images/vir/virgin.gif
	coeus.cam.zeus.com	10.100.1.18:3636	212.58.246.93:80	intranet	test	301	/
	coeus.cam.zeus.com	10.100.1.18:3620	212.58.246.93:80	intranet	test	301	/
	coeus.cam.zeus.com	10.100.1.18:3499	212.58.246.92:80	test	test	301	/
	coeus.cam.zeus.com	10.100.1.18:3438	212.58.246.92:80	test	test	301	/

Filter (フィルタ)ボタンをクリックして、表示されたロジックを適用します。このフィルタが、このページの上部に表示される「Active Filters」に追加されます。次に、「Resp.Code」(レスポンスコード)列で値「200」をクリックして、この値以外を除外するフィルタを適用します。これにより、目的のコネクションのみが表示されます。

Traffic Manager	From	To	VS	Pool	Resp. Code	Request
coeus.cam.zeus.com	10.100.1.14:37074	92.52.65.213:80	website	website	200	/assets/img/bullet.gif
coeus.cam.zeus.com	10.100.1.14:37074	92.52.65.213:80	website	website	200	/assets/default/Site/en/images/vir/virgin.gif
coeus.cam.zeus.com	10.100.1.14:37074	92.52.65.213:80	website	website	200	/assets/img/divider.gif
coeus.cam.zeus.com	10.100.1.14:37074	92.52.65.222:80	website	website	200	/assets/img/logo.gif
coeus.cam.zeus.com	10.100.1.14:37074	92.52.65.213:80	website	website	200	/assets/default/Site/en/css/main.css

アクティブなフィルタを調整するには、変更を行った後に**Update Filters** (フィルタの更新)ボタンをクリックします。これにより、コネクションデータが再表示されます。フィルタを削除するには、削除する条件の横のチェックボックスを選択し、**Update Filters** (フィルタの更新)をクリックします。**Clear Filters** (フィルタの消去)をクリックすると、全てのフィルタを削除できます。

Refresh (更新)ボタンを使用すると、結果のスナップショットを更新できます。また、**Download** (ダウンロード)ボタンを使用すると、結果の.tsv (タブ区切り値)ファイルを作成し、他のアプリケーションでの解析に利用できます。

さらに、各行の左側にある虫眼鏡アイコンをクリックすると、特定のコネクションについて以下の追加の詳細が表示されます。

セクション	説明
Request tracing (リクエストトレーシング)	このセクションには、内部コネクション処理イベントの時系列が表示されます。リクエストの全所要時間、そのうちTrafficScriptルールの処理時間、およびその他の役立つ情報を表示できます。 この情報は、そのコネクションを処理した仮想サーバで request_tracing!enabled キーが有効化されている場合のみ記録されます。詳細については、コネクション分析を参照してください。
Request details (リクエストの詳細)	このセクションには、リクエストに含まれていた全てのリクエストヘッダが表示されます。
Response details (レスポンスの詳細)	このセクションには、全てのレスポンスヘッダ(TrafficScriptまたは詳細キャッシュロギングによって追加されたものも含む)が表示されます。



コネクションスナップショットのサイズは、**System** (システム) > **Global Settings** (グローバル設定) ページで拡張することができます。**recent_conns**キーの「Logging」(ロギング)セクションを確認してください。

Activity (アクティビティ) > Draining Nodes (ノードのドレイン)

例えばメンテナンスやアップグレードのために、ノードをシステムから削除したい場合、そのノードを「ドレイン」するよう設定できます。ドレインされたノードは引き続き既存のセッションを処理しますが、Traffic Managerはそのノードにコネクションを送信しなくなります。「Draining Nodes (ノードのドレイン)」セクションには、現在ドレインしているノードに関する情報が表示されます。

このページには、ドレイン中の各ノードについて、最後のコネクションからの経過時間および現在のアクティブな接続数が表示されます。**Reload This Page** (このページのリロード)をクリックすると、この情報を更新できます。

アクティブなセッションが全て終了した時点で、ノードを安全に削除することができます。これを行うには、**Pools (プール) > Edit (編集)**ページを使用するか、**Remove a Node (ノードの削除)**ウィザードを使用します。

詳細については、[ノードのドレイン](#)、[無効化](#)、[削除](#)で、ノードのドレインに関する説明を参照してください。

Activity (アクティビティ) > View Logs (ログ表示)

全てのトランザクションをリクエストログに記録するよう仮想サーバを設定することができます。**Virtual Server (仮想サーバ) > Request Logging (リクエストロギング)**ページ([リクエストロギング](#)を参照)で、記録する情報およびログファイルの書き込み先を設定できます。

Activity (アクティビティ) > View Logs (ログ表示)ページでは、リクエストログにレコードが追加されていく状況をリアルタイムに確認できます。



Traffic Manager仮想インスタンスおよびクラウドインスタンスは自己完結型であり、自己管理される専用のログパーティションを備えています。ログファイルはこのパーティションに保存され、自動的にアーカイブおよび削除されます。そのため、ログファイルの保存場所を設定する必要はありません。



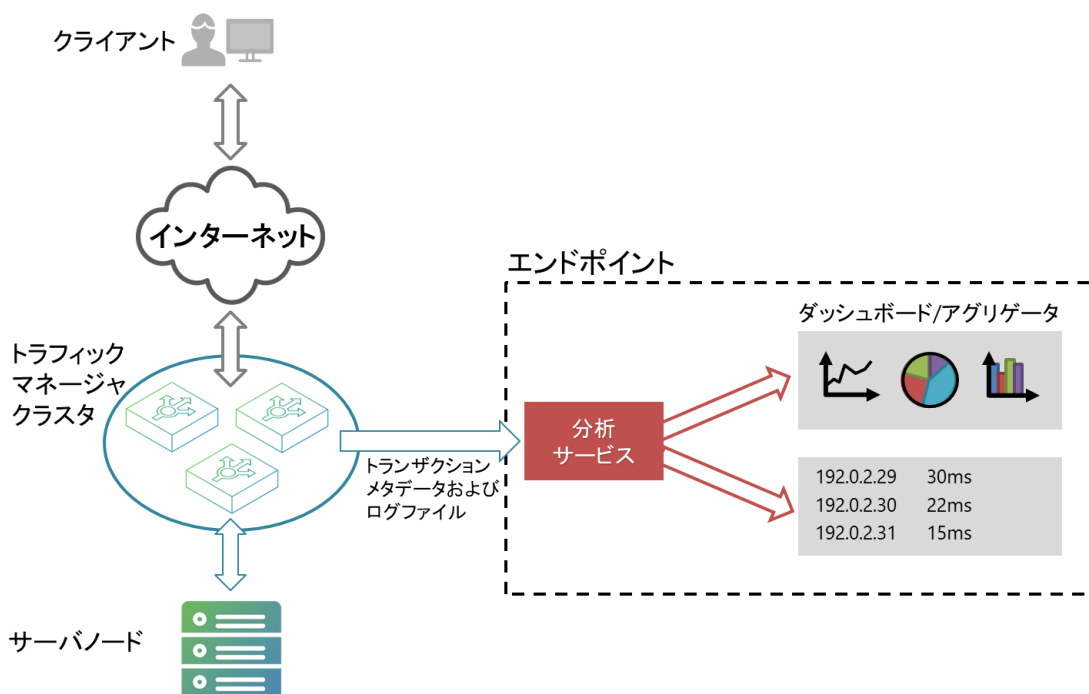
ログファイルは、**Activity (アクティビティ) > View Logs (ログ表示)**ページで表示でき、**Activity (アクティビティ) > Download Logs (ダウンロードログ)**ページでダウンロードできます。また、Traffic ManagerコントロールAPIを使用して、Traffic Manager仮想/クラウドインスタンス上のログファイルをダウンロードしたり管理したりすることもできます。

リモートサービスに対する分析のエクスポート

Traffic Managerクラスタでは、仮想サーバによって処理されるトランザクションのメタデータおよび保存されたログデータをオフライン分析のために外部ソースにエクスポートできます。

トランザクションメタデータとは、汎用TCPコネクションを通じたクライアントとサーバ間のHTTPリクエストまたは通信など、Traffic Managerによって処理される個々のトランザクションに関する情報です。

この情報は、クラスターで動作しているサービスのヘルスおよび効率性に関する詳細な可視性を提供し、予期しないサービスの動作が生じた場合にデバッグを容易にすることができます。



Traffic Managerでは、以下のデータタイプをリモートにストリーミングすることができます。

- 仮想サーバによって処理され、**Activity** (アクティビティ) > **Connections** (コネクション)ページに記録される全てのトランザクションのメタデータ。
- Traffic Managerクラスターに保存される設定可能なログファイルのセット。
- SNMPカウンタ



SNMPカウンタは、Traffic Managerのアクティビティをモニタリングする標準の方法として、SNMP対応サービスで利用できます。これは、厳格には分析エクスポート機能の設定可能な部分ではありません。SNMPの詳細については、[アクティビティモニタリング](#)を参照してください。

Traffic Managerからエクスポートされた分析データを受信するには、エンドポイントで、TCPストリーム(トランザクションメタデータの場合)およびPOSTリクエスト(ログデータの場合)を通じて、JSONフォーマットでデータの取り込みをサポートする分析サービスを利用する必要があります。分析サービスが監視しているIPアドレスおよびポートとエクスポートされたログデータの送信先URLでTraffic Managerを設定します。

トランザクション分析とログデータのエクスポートを設定するには、**System** (システム) > **Analytics Export** (分析エクスポート)をクリックします。

トランザクションメタデータのエクスポートを制御する設定では、「Transaction Metadata (トランザクションメタデータ)」セクションを使用します。ログデータのエクスポートを制御する設定では、「Log Files (ログファイル)」セクションを使用します。



分析エクスポートの設定は、同じクラスタ内の全てのTraffic Managerで共有されます。この結果、1つのクラスタメンバに対して加えた変更は、全ての他のメンバに自動的に複製されます。

トランザクションメタデータレコードのエクスポート設定

Traffic Managerは、エンドポイントに対して一連の長期的なTCPコネクションを維持し、各コネクションは内部データプレーンプロセスによって管理されます。データプレーンプロセスは、Traffic Managerで受信データを処理し、改行文字で区切られたJSONフォーマットで、対応するメタデータレコードをエンドポイントにストリーミングします。

エンドポイントに対するコネクションに何らかの理由のために障害が発生した場合、全ての送信されなかったメタデータレコードは、一般に破棄されます。Traffic Managerは、エクスポートできなかったレコードの数をトラッキングするために、SNMPカウンタを備えています。

Traffic Managerで定義済みのエンドポイントにトランザクションメタデータレコードをエクスポートする方法を設定するには、トランザクションメタデータの設定を使用します。

▼ Transaction Metadata

These settings control how metadata about transactions processed by the traffic manager is exported.

 **View JSON Schema for transaction metadata**

Export metadata about transactions processed by the traffic manager to an external location.

transaction_export!enabled: ☐ Yes ☒ No Default: No

The endpoint to which transaction metadata should be exported. The endpoint is specified as a hostname or IP address with a port.

transaction_export!endpoint:

Whether the connection to the specified endpoint should be encrypted.

transaction_export!tls: ☒ Yes ☐ No Default: Yes

Whether the server certificate presented by the endpoint should be verified, preventing a connection from being established if the certificate does not match the server name, is self-signed, is expired, is revoked, or has an unknown CA.

transaction_export!tls_verify: ☒ Yes ☐ No Default: Yes

トランザクションメタデータをストリーミングする場合にTraffic Managerによって使用されるJSONスキーマを表示するには、**View JSON Schema for transaction metadata** (トランザクションメタデータのJSONスキーマの表示) をクリックします。このスキーマを使用して、予期されるメタデータレコードの構造で分析サービスを設定します。

Transaction Metadata (トランザクションメタデータ)セクションには、以下の設定キーが含まれます。

キー	説明
transaction_export!enabled	transaction_export!endpointで設定される外部エンドポイントに対して、クラスタ全体のコネクションを有効にします。 これを有効にした場合、Traffic Managerでは、同等の設定が有効な仮想サーバによって処理される全てのトランザクションに対してメタデータがエクスポートされます。これを無効にした場合、Traffic Managerでは、個々の仮想サーバの設定にかかわらず、いかなるトランザクションメタデータもエクスポートされません。 個々の仮想サーバのエクスポート設定を表示および変更するには、トランザクションメタデータのエクスポートを参照してください。
transaction_export!endpoint	Traffic Managerによって分析データが送信されるエンドポイント。 フォーマット<hostname>:<port>を使用します。ここで、<hostname>はホスト名またはIPv4/IPv6アドレスであり、<port>はポート番号です。リテラルIPv6アドレスは、角括弧で囲む必要があります。例えば、「[fe80::1]:5000」のように指定します。ホスト名を指定する場合、これはTLSハンドシェイクでのサーバ名の指定(SNI)として使用されます。

キー	説明
	エンドポイントは、TCPソケットを通じてJSONイベントを受信するように設定する必要があります。これはHTTPエンドポイントではありません。
transaction_export!tls	エンドポイントへのコネクションで、TLS暗号化を有効にします。 これを有効にした場合、System (システム) > Global Settings (グローバル設定) > SSL Configuration (SSL設定)に含まれる設定が、コネクションに適用されます。 トランザクションメタデータレコードには、アプリケーションによって、機密情報またはユーザの識別情報が含まれることがあります。商用システムで機密性を保持するために、分析サービスへのコネクションは確実に暗号化し、検証します(transaction_export!tls_verifyも参照してください)。TLS暗号化を無効にするのは、開発環境でのテストの場合に限定することを強くお奨めします。
transaction_export!tls_verify	Traffic Managerの「Certificate Authorities and Certificate Revocation Lists (認証局および証明書失効リスト)」カタログ内にある認証局(CA)の証明書に照らしたエンドポイントのTLSサーバ証明書の検証を有効にします。 この設定より、Traffic Managerでは、サーバ名と一致しない証明書、自己署名された証明書、期限切れの証明書、失効した証明書、またはCAが未確認の証明書が許可されなくなります。

追加のサービス別の設定を行うには、対象の仮想サーバの**Edit** (編集)ページでConnection Analytics (コネクション分析)セクションを使用します。特定の仮想サーバで分析のエクスポートを無効化または制限して、Traffic Managerによってエクスポートされるデータの全体量を制限するには、以下の設定を使用します。詳細については、[トランザクションメタデータのエクスポート](#)を参照してください。

ログファイルのエクスポート設定

Traffic Managerが定義済みのエンドポイントにログデータをエクスポートする方法を設定するには、Log Files (ログファイル)セクションを使用します。

▼ Log Files

These settings control which log files should be exported, and how entries from those log files should be exported.

Monitor log files and export entries to the configured endpoint.

log_export!enabled: ☐ Yes ☒ No Default: No

The URL to which log entries should be sent. Entries are sent using HTTP(S) POST requests.

log_export!endpoint:

Log file categories currently configured to be exported

Admin Server Access	✕
Application Firewall	✓
Audit Log	✓
Event Log	✓
Process Monitor	✕
Routing Software	✕
System - authentication log	✕
System - syslog	✕

 **Choose Log Files to export**

The HTTP authentication method to use when exporting log entries.

☒ None

☐ Basic (Username and Password) ...

The username to use for HTTP basic authentication.

log_export!auth!http:

The password to use for HTTP basic authentication.

☐ Splunk (HEC token) ...

The HTTP Event Collector token to use for HTTP authentication with a Splunk server.

Whether the server certificate should be verified when connecting to the endpoint. If enabled, server certificates that do not match the server name, are self-signed, have expired, have been revoked, or that are signed by an unknown CA will be rejected.

log_export!tls_verify: ☒ Yes ☐ No Default: Yes

The number of seconds after which HTTP requests sent to the configured endpoint will be considered to have failed if no response is received. A value of 0 means that HTTP requests will not time out.

log_export!request_timeout: seconds Default: 30

Traffic Managerによって、現在エクスポートセットに含まれている、または除外されているログファイルカテゴリの概要リストが提供されます。ログファイルカテゴリのリストを設定するには、**Choose Log Files to export** (エクスポートするログファイルの選択)をクリックします。

Log Files (ログファイル)セクションには、以下の設定キーが含まれます。

キー	説明
log_export!enabled	ログファイルのエクスポートのために設定されたエンドポイントへのコネクションを有効にします。 これを有効にした場合、ログデータは、HTTPコネクションを通じて、log_export!endpoint!に含まれるエンドポイントURLに送信されます。 ログファイルのエクスポートを有効にする前に、Traffic Managerからエクスポートするログファイルカテゴリのリストを見直します。詳細については、エクスポートするログファイルカテゴリの選択を参照してください。
log_export!endpoint	Traffic Managerによってログデータが送信されるエンドポイントのURL。 エンドポイントは、1つ以上のJSON符号化イベントを含むHTTP Postリクエストを受信するように設定されたhttpまたはhttpsベースのURLにする必要があります。
log_export!auth!http	Traffic Managerがエンドポイントへのコネクションで使用する必要があるHTTP認証のタイプ。 以下のいずれかの方法を選択します。 「None (なし)」: 認証方法を使用しません。 「Basic (基本)」: HTTP basic-auth (基本認証)を使用します。 log_export!auth!usernameとlog_export!auth!passwordで必要なクレデンシャルを指定します。 「Splunk」: Splunk® サーバで、HTTP認証にHECトークンを使用します。 log_export!auth!hec_tokenでトークンを指定します。
log_export!tls_verify	Traffic Managerの「Certificate Authorities and Certificate Revocation Lists (認証局および証明書失効リスト)」カタログ内にある認証局(CA)の証明書に照らした設定済みエンドポイントのTLSサーバ証明書の検証を有効にします。 この設定より、Traffic Managerでは、サーバ名と一致しない証明書、自己署名された証明書、期限切れの証明書、失効した証明書、またはCAが未確認の証明書が許可されなくなります。

キー	説明
log_export!request_timeout	Traffic Managerによってコネクションに障害が発生したと見なされるまでの秒単位の時間制限。 Traffic Managerで各POSTリクエストを処理するための時間制限を指定するには、この設定を使用します。これは、エンドポイントで動作している分析サービスの機能不全を検出するために役立つことがあります。 制限なしに指定するには、値0を使用します。

エクスポートするログファイルカテゴリの選択

エクスポートセットの一部として、Traffic Managerに含まれるログのカテゴリを追加、除外、または変更するには、Log File Export Categories (ログファイルエクスポートカテゴリ)ページを使用します。

ログファイルカテゴリは、Traffic Managerシステムに保存される一連の関連ログファイルに対応しています。標準として、Traffic Managerでは、全ての製品タイプに適用される複数の内蔵ログカテゴリが用意されています。

Log File
Export

Log File Export Categories

Unfold All / Fold All

A log file export category manages a set of related log files that can be exported by the traffic manager. Some categories of log file are exported only on appliance platforms.

▶  Admin Server Access (Built-in)	 Edit
The Administration Server access log.	
▶  Application Firewall (Built-in)	 Edit
Log files from all application firewall processes.	
▶  Audit Log (Built-in)	 Edit
The traffic manager audit log.	
▶  Event Log (Built-in)	 Edit
The traffic manager event log.	
▶  Process Monitor (Built-in)	 Edit
Log files maintained by the traffic manager's internal health monitor.	
▶  Routing Software (Built-in, Appliance)	 Edit
Log files generated by the traffic manager's OSPF and BGP routing software.	
▶  System - authentication log (Built-in, Appliance)	 Edit
The operating system's authentication log.	
▶  System - syslog (Built-in, Appliance)	 Edit
The operating system's syslog.	

Create new log file export category

Name:

「Built-in (内蔵)」と表示されているリスト内の各項目は、標準のTraffic Managerのログエクスポートカテゴリを示しています。チェックアイコンは、エクスポートセットに含めるためにマークが付けられたログを示しています。バツ印のアイコンは、現在除外されているログを示しています。

ログ名の後ろの「Appliance (アプライアンス)」タグは、Traffic Managerアプライアンス製品のみでエクスポートされるログカテゴリを示しています。

Traffic Managerと同じシステムで動作している他のプロセスに属するログカテゴリなど、追加のログカテゴリをエクスポートするには、「Create new log file export category (ログファイルエクスポートカテゴリの新規作成)」セクションに新しいログタイプを追加します。次に、Log Export Category edit (ログエクスポートカテゴリの編集) ページでログの場所を設定します。

ログエクスポートカテゴリの編集

エクスポートするログカテゴリを追加または除外するか、その他の設定可能な詳細を設定するには、ログ名の横にある**Edit** (編集) をクリックして、ログエクスポートカテゴリのプロパティを編集します。

編集ページには、以下の設定キーが含まれます。

キー	説明
enabled	エクスポートセットにこのログカテゴリを含めます。
appliance_only	Traffic Managerアプライアンス製品のみに対してログカテゴリをエクスポートします。
files	Traffic Managerがエクスポートセットに含める必要があるログファイルとローテーション(履歴)バージョンを選択するためのワイルドカードパターンのセット。
note	このログカテゴリに対するフリーテキストの説明。
history	このログに対してどの程度の履歴アクティビティをTraffic Managerでエクスポートするかを示します。以下のいずれかの値を使用します。 「Do not export any historic entries」: エクスポートで、全ての履歴アクティビティログを無視し、現在のアクティビティログのみを使用します。 「Export all historic entries」: 全ての現在のアクティビティログと履歴アクティビティログをエクスポートに含めます。 「Export recent historic entries, according to the history_period setting」: history_periodで設定した日数までの履歴アクティビティを含めます。

ログレコードのメタデータタグの編集

メタデータセクションを使用して、このログカテゴリのエンドポイントに送信されたレコードに追加された、機械が読み取れる名前/値のペアのセットを定義します。

リモート分析サービスの要件に合わせて、メタデータタグを編集します。各ログカテゴリの出力が個々に識別可能になるように、各ログカテゴリに対して、一意のメタデータタグのセットを確実に定義します。これを定義しなければ、分析サービスは、各レコードが属するログファイルのグループを判断できません。お使いの分析サービスの資料を参照して、どのレコードタグが必要となるのかを判断してください。

Traffic Managerはネストされたメタデータタグをサポートし、関連するデータオブジェクトのグループを実装できます。タグをネストするには、「<first part>.<second part>.<third part>.<etc.>」の形式で、ピリオド(.)を区切り文字として使用します。例えば、「fields.format」となります。

Traffic Managerでは、2つの特殊なマクロもサポートしています。これは、メタデータタグ値に指定すると、次のように、ログレコードで対応するIDで置き換えられます。

- **%Cluster_id%:** Traffic Managerクラスタの ID です。
- **%instance_id%:** ログエントリを記録したTraffic ManagerインスタンスのIDです。

内蔵ログカテゴリには、デフォルトで多くの定義済みのメタデータタグが含まれています。これらのタグはオプションであり、各タグの横にある**Remove** (削除)チェックボックスを選択することで削除できます。**Complete** (完了)をクリックしてアクションを完了します。

定義済みのタグには次のものがあります。

- **source:** 分析サービスでレコードの検索を容易にするために使用されるログのソースを説明するテキスト名。例えば、「eventlog」は、Traffic Managerのイベントログからエクスポートされたエントリを識別します。
- **sourcetype:** フィールドの解析と抽出の指針とするために、分析サービスによって使用されるログフォーマットのテキストによる定義。例えば、「zxtm_event_log」は、エクスポートされたレコードの主要な「event」パラメータに含まれる情報が、Traffic Managerのイベントログのフォーマットであることを示します。

クラウドクレデンシャル

Traffic Managerソフトウェアをクラウド環境で使用する場合、Traffic Managerがクラウドプロバイダに対してAPI呼び出しを行うために、クレデンシャルが必要になる場合があります。まず、オートスケーリングを使用するよう設定されたプールには、クラウド内でスケーリングメカニズムを使用するためにAPIクレデンシャルが必要です。このセクションでは、選択したクラウドプロバイダAPIの必要条件に基づいて、そのようなクレデンシャルをカタログオブジェクトに記録することができます。プールのオートスケーリングの詳細については、[オートスケーリング](#)を参照してください。

Traffic Managerには、クレデンシャルセットのベースになる多数の内蔵クラウドAPIが用意されています。

- VMware vSphere
- Amazon EC2
- Rackspaceクラウド

また、**Catalogs** (カタログ) > **Extra Files** (追加ファイル)で適切な実行可能スクリプトをアップロードすることにより、カスタムクラウドAPIを追加することもできます。詳細については、PulseコミュニティのWebサイト (<https://community.pulsesecure.net>) のTraffic Managerセクションを参照してください。

新しいクラウドクレデンシャルセットを作成するには、以下の手順に従います。

1. **Catalogs** (カタログ) > **Cloud Credentials** (クラウドクレデンシャル)をクリックします。
2. **Name** (名前)フィールドに名前を入力します。
3. **Cloud API** (クラウドAPI)で、使用するクラウドAPIを選択します。
4. 残りのフィールドは、選択したAPIによって異なります。通常は、ID、パスワード/パスコード、および追加の認証情報を入力するためのフィールドが表示されます。
5. **Create Cloud Credentials** (クラウドクレデンシャルの作成)をクリックします。

以下の表に、新しいクレデンシャルセットを作成するときに表示される各フィールドの詳細を示します。

設定	説明
Name	Traffic Managerの設定内でこのクレデンシャルセットを識別するための名前です。
Cloud API	選択したクラウドプロバイダAPIです。
ID / Name (cred1)	API呼び出しに使用するクラウドプロバイダアカウントのユーザ名またはIDです。
Auth Key / Password (cred2)	ユーザ名/IDに関連付けられたパスワードです。
Token / Server (cred3)	一部のクラウドプロバイダでは、API呼び出しを行うためにさらに認証トークンまたは追加項目が必要になります。VMware vSphereでは、API呼び出しを受け入れるためにvCenterホスト名/IPを必要とします。このような追加情報はここで指定できます。
update_interval	Traffic Managerはユーザに代わって、定期的にクラウドプロバイダに対して実行中の全インスタンスのステータスをクエリします。この動作は、外部システムによってノードが追加または削除されたときに最新の状態に保つために必要になります。 update_interval は、これらのステータス呼び出しを行う頻度(秒単位)を決定します。一部のクラウドでは、このような呼び出しを行える1分あたりの回数が制限されることに注意してください。
change_process_timeout	変更プロセスの最大所要時間(秒単位)です。例えば、クラウドAPIに対してノードの作成/削除リクエストが行われるときに、この設定により、そのリクエストが完了するまで待機する時間が指定されます。

Amazon EC2クレデンシャルのIAMロール

Amazon EC2のインスタンスは、ローカルに保存されているクレデンシャルの代わりにアイデンティティおよびアクセス管理(IAM、Identity and Access Management)ロールを使用します。IAMロールは、トラフィックIPアドレス、オートスケーリング、またはアプライアンスネットワーク管理を使用するデプロイのために必要です。

Amazon EC2で新しい仮想マシンのインスタンスを開始するときは、そのインスタンスに割り当てるロールを指定します。

ロールの作成および管理方法など、IAMロールに関する詳細については、AWSに関するドキュメント(<http://aws.amazon.com/documentation/>)を参照してください。

Pulse Secure Virtual Webアプリケーションファイアウォール

Pulse Secure Virtual Web Application Firewall(vWAF)は、最新世代のミッションクリティカルなWebアプリケーションに対する攻撃を検出し、これらのアプリケーションを保護するエンタープライズレベルのWeb Application Firewallです。

中央からセキュリティのモニタリング、レポートの送信、およびアラートの報告を行うことができ、外部攻撃に対するWebアプリケーションおよびインフラストラクチャの保護をカスタマイズして適用することも可能です。



これは、適切なライセンスキーのアップグレードによってのみアクティブになる、Traffic Managerのオプション機能です。詳細については、サポート窓口にお問い合わせください。

概要

vWAFは、オプションで提供されるTraffic Managerのコンポーネントです。ライセンスを入手し、クラスターで有効化すると、Traffic Managerソフトウェアの総合的な機能に完全に統合されます。

Traffic Managerは、Webサービスのインフラストラクチャと総合的なコントロールを提供し、指示があった場合にトラフィックをvWAFに渡します。

本章では、vWAFの概要とTraffic Managerとの相互関係について説明します。vWAFの詳しい設定および使用手順については、『Pulse Secure Virtual Web Application Firewall: User Guide』およびオンラインヘルプを参照してください。

vWAFの有効化

注記

Traffic Managerのソフトウェアインスタンスと非ソフトウェアインスタンスが混在するクラスターでvWAFコンポーネントを有効化しないでください。この構成はサポートされていません。

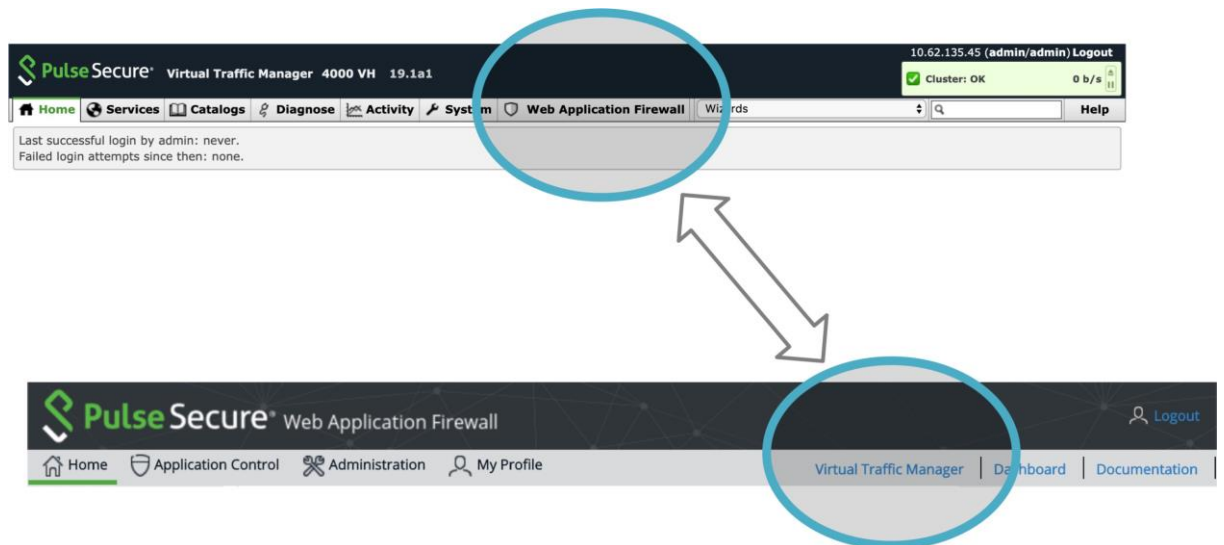
vWAFを使用するためには、サポート窓口から適切なライセンスキーを取得します。新しいライセンスをインストールすると、Traffic Managerの管理UIの**System** (システム) > **Web Application Firewall**ページがアクティブになります。このページを使用して、vWAFコンポーネントの有効化、無効化、および設定を行います。

他の設定変更と同様に、クラスター内の1台のTraffic ManagerでvWAFの有効化または無効化を行うと、その他全てのクラスターメンバーの設定にも同等の変更が自動的に反映されます。

Traffic Managerの管理UIで利用できるvWAF機能

vWAFのコントロールおよび設定は、Traffic Managerのナビゲーションバーにあるボタンを使用して行います。このボタンを使用すると、Traffic ManagerとvWAFのユーザインターフェースが切り替わります。

 このボタンを表示させるには、vWAFコンポーネントを有効化する必要があります。



メインのTraffic Manager管理UIには、vWAFの通信や動作を管理するための機能も多数用意されています。

- **System** (システム) > **Web Application Firewall**ページには、vWAFコンポーネントを有効化および無効化するためのセクションがあります。また、vWAFが使用するネットワークポートや実行するプロセス数など、低レベルのvWAF設定も含まれています。
- **System** (システム) > **Traffic Managers**ページのSoftware Restart (ソフトウェアリスタート)セクションに「Restart Web Application Firewall... (Application Firewallのリスタート)」ボタンが追加されます。一部の設定変更には、Traffic Manager自体のリスタートは不要でも、vWAFのリスタートが必要になる場合があります。
- **Services** (サービス) > **Virtual Servers** (仮想サーバ) > **Edit** (編集)ページの「Basic Settings (基本設定)」セクションには、各HTTPサービスについてvWAFを有効化または無効化するためのコントロールがあります。
- ルールカタログには、新しい組み込み式TrafficScriptルールである「Application Firewall Enforcer」があります。Traffic Managerはこのルールを使用して、vWAFサービスとの通信を維持します。
- Traffic Managerにより、vWAFサービス、特に現在の実行状態に関連するステータスメッセージがTraffic Managerのイベントログおよび監査ログに記録されます。特定の設定キーの変更およびモニタエラーに加え、ソフトウェアの停止、起動、リスタートが全て記録されます。vWAFでも、サービスのセキュリティに関連する攻撃およびその他のイベント情報を記録する独自のローカルイベントログが保持されます。このログは、vWAFの管理UIから表示できます。詳細については、『*Pulse Secure Virtual Web Application Firewall: User Guide*』を参照してください。
- vWAF設定セクションが**Diagnose** (診断)ページに表示されます。このセクションには、クラスタ内で発生したファイアウォールの問題に関する詳細が表示されます。Traffic ManagerとvWAF間の通信が何らかの理由で中断した場合、エラーが発生したことがStatus Applet (ステータスアプレット)に表示されます。このエラーインジケータをクリックすると、Diagnose (診断)ページに移動します。このページで、具体的にどのような問題が発生したのかを確認し、この状況に対処するための適切なアクションを決定することができます。

System (システム) > Web Application Firewallページ

このページには、vWAFの性能および通信に関する設定オプションがあります。



通常の運用では、このセクションの設定を変更する必要はありません。ここで説明する内容は、参考のための情報です。

vWAFには、DeciderプロセスとEnforcer TrafficScriptルール間の通信専用のベースポートが必要です。

▼ Application Firewall Decider Settings

This section allows you to configure the application firewall Decider processes.

The number of Application Firewall decider process to run.

Decider Processes:

The port to which the Enforcer rule should send traffic so it can be distributed between the decider processes.

Decider Port:

工場出荷時のデフォルトポートはほとんどのデプロイに適していますが、システムでこのポートを利用できない場合は、別の値を指定できます。このページで設定を変更すると、vWAFが自動的にリスタートされます。

「Decider Processes (Deciderプロセス)」設定を使用すると、システムで利用可能なCPUコア数に応じてvWAFの性能を調整することができます。より多くのトラフィックを処理してシステムの性能を向上させるには、この値を最大64に増やしてさらなるDeciderプロセスをインスタンス化します。ただし、空きリソース以上のプロセスを実行すると、性能が抑制され、サービスの低下につながります。このような状況では、プロセス数を減らすことによって、システムの全体的な性能を高めることができます。

Deciderプロセスの数は、Traffic Managerクラスタ内で搭載CPUコアの数が最も少ないサーバのCPU数と同じ数に設定することをお奨めします。例えば、クラスタ内に2台のTraffic Managerがあり、一方のTraffic Managerには4個のCPUコアが、もう一方には2個のCPUコアが搭載されている場合、Deciderプロセスの数を2に設定してください。

▼ Local Application Firewall Ports

This section allows you to configure additional ports used by the application firewall on this machine, this might be necessary to avoid port conflicts with existing services. To change these settings on another cluster member, you must log in to that cluster member's Administration Server.

The base port from which the Application Firewall decider processes should run. Ports will be used sequentially above this for each additional decider process that runs.

Internal Decider Base Port:

The Application Firewall Administration Server port, this port is only open on localhost.

Admin Server Port:

The Application Firewall XML Master port, this port is used on all IP addresses.

Admin Master Port:

The Application Firewall XML Slave port, this port is used on all IP addresses.

Admin Slave Port:

The Application Firewall Updater GUI Server Port, this port is used on localhost only.

Updater Admin Server port:

The Application Firewall Updater External Control Center Port, this port is used on localhost only.

External Updater Control Center Port:

The Application Firewall Updater GUI Backend Port, this port is used on localhost only.

Internal Updater Control Center Port:

The Application Firewall REST Internal API port, this port should not be accessed directly

Internal REST API Server Port:

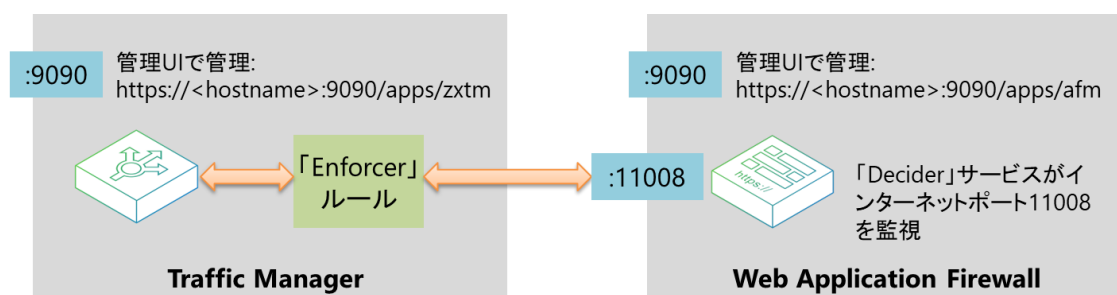
The Application Firewall internal communication base port. The Application Firewall will use ports sequentially above this for internal communication. These ports are bound only to localhost.

Internal Communication Base Port:

これ以外に、vWAFには内部通信用にいくつかのポートが必要です。通常の場合は、デフォルト値を変更する必要はありませんが、ポート番号の競合を解決する場合は、このページで必要なポート設定を別の値に変更してください。

EnforcerとDecider

Traffic ManagerのEnforcer TrafficScriptルールにより、Deciderサービスによるチェックのためリクエストがデフォルトローカルポート11008に転送されます。Deciderから「許可」または「拒否」の決定が返されると、Enforcerルールによってこの決定が適用されます。



「Enforcer」は、Traffic ManagerとvWAF間の通信を可能にする、組み込み式のTrafficScriptルール(名前は「Application Firewall Enforcer」)です。全てのHTTPリクエストとレスポンスをキャプチャし、詳細なチェックのためvWAFで実行中の「Decider」サービスに転送します。

Deciderは、設定データベースに保存されている一連のルールを使用してHTTPリクエストを評価し、実行するアクションを決定します。その後、Enforcerがその決定を実行し、各リクエストまたはレスポンスは、その決定に従って受け入れられるか、変更されるか、拒否されます。

Enforcerルールについての詳細は、[Enforcerルール](#)を参照してください。

Enforcerルール

Traffic Managerは、TrafficScriptルールシステムを介してvWAFと通信します。TrafficScriptルールシステムでは、Webアプリケーションとの間で送受信されるトラフィックによって生成されるリクエストとレスポンスに特定のロジックが適用されます。vWAFを有効化すると、「Application Firewall Enforcer」(Enforcer)という名前の特殊なルールが作成され、Traffic ManagerがHTTPトラフィックをDeciderサービスに転送するよう指示されます。このルールを仮想サーバに追加するには、**Virtual Server** (仮想サーバ) > **Edit** (編集) ページで**Web Application Firewall**設定を有効化します。

Enforcerルールは、他のルールと同様に有効化、無効化、削除、または編集することはできませんが、移動してTrafficScriptルールの実行順序を変更することは可能です。これは、リクエストまたはレスポンスを先に処理する必要がある場合に便利です。例えば以下のTrafficScriptのように、Enforcerの実行より前に特定のリクエストをホワイトリストに指定するルールを追加する場合などが挙げられます。

```
connection.data.set( "enforcer.whitelist", 1 );
```

他のルールと同様に、Enforcerはルールカタログに保存されます。Enforcerルールは削除できませんが、**Save As**機能を使用してコピーを作成できます。



このようなコピーは、元のEnforcerルールと同じように見え、提供されるオプションも同じですが、従来のルールと同様に使用する必要があります。すなわち、通常の方法で仮想サーバ、プール、およびルールに割り当てる必要があります。仮想サーバの**Web Application Firewall**設定を使用して割り当てるのは、元のEnforcerルールのみです。

Enforcerには、設定可能なデータ項目数を設定することもできます。ルールの設定ページにアクセスするには、そのルール名をクリックするか、**Edit** (編集)リンクをクリックします。そのままでは実際のTrafficScriptテキストは表示されません。**Rule text** (ルールテキスト)リンクをクリックして表示します。TrafficScriptテキストはバイナリフォーマットであるため、人間が読み取れるのは一部のみです。

ルールテキストの一番上に、設定可能なパラメータが多数表示されます。変更が行われると、Traffic Managerは自動的にルールテキストを更新します。

- **allowonerror**: Deciderにアクセスできない場合に全てのトラフィックを許可するには、1に設定します(デフォルト: 0)
- **backend_timeout**: Deciderを待機する秒数です。この時間が経過するとタイムアウトになります(デフォルト: 5)
- **bypass_enforcer_max**: enforcer_max_body_sizeよりも大きいリクエストがvWAFをバイパスすることをブロックするには、0に設定します。(デフォルト: 1)
- **bypass_file_types**: vWAFで免除するファイル拡張子のスペース区切りリストです。(デフォルト: 「css js png jpg gif」)
- **debug**: デバッグモードを有効化するには、1に設定します。このオプションを使用すると、問題の追跡に役立つ詳細なエラーが出力されます(デフォルト: 0)
- **discard_on_reject**: エラーメッセージを送信せずに、拒否されたコネクションを閉じるには、1に設定します。
- **enforcer_max_body_size**: vWAFによって処理されるバイト単位の最大ボディサイズです。この上限よりも大きいリクエストは、**bypass_enforcer_max**の設定に応じてブロックまたはバイパスされます。(デフォルト: 2097152)



通常は、これらの設定を変更する必要はありません。変更すると、ルールが正しく機能しなくなる可能性があるため、どのような結果になるかが完全に分かっている場合のみ変更してください。

ユーザ管理

Traffic ManagerとvWAF間には、ユーザおよびグループアクセスが維持されています。vWAFの新規ユーザには、Traffic Managerの管理UIの**admin** (管理)グループのメンバと同じ認証レベルが設定されます¹。そのため、通常の管理タスクを実行するときに、追加のアクセス権限またはログイン処理は必要ありません。いずれかのUIで**Logout** (ログアウト)をクリックすると、ユーザは両方のアプリケーションからログアウトすることができます。

¹ 厳密には、vWAFのユーザレコードは、そのユーザがTraffic Managerの管理UIからvWAFの管理UIにアクセスするまで作成されません。同じユーザ名が既に存在する場合、新しいユーザは作成されず、代わりに既存のユーザレコードが使用されます。

注記

Traffic Managerからユーザを削除しても、そのユーザはvWAFから自動的に削除されません。この操作は、管理者が手動で行う必要があります。

「Demo (デモ)」や「Monitoring (モニタリング)」など、Traffic Managerの他のユーザグループのメンバは、**System** (システム) > **Users** (ユーザ) > **Groups** (グループ) > **Edit Group** (グループの編集) ページでパーミッションが正しく設定されている場合のみ、vWAFの様々なUI機能にアクセスできます。Traffic Managerの**System** (システム) セクションには、そのための以下の設定があります。

- **Application Firewall**: Traffic Managerの管理UIの**System** (システム) > **Web Application Firewall** ページに対するアクセスをコントロールします。
- **Application Firewall > Administration (管理)**: (前述のツールバーボタンからの) のvWAFの管理UIに対するアクセスをコントロールします。



vWAFの管理UIでユーザアカウントをあらかじめ定義しておくことにより、管理ユーザを自動的に作成するメカニズムをオーバーライドすることができます(制限付きアクセスを指定したり、非管理グループを使用したりする場合など)。これにより、vWAFのUIでは、Traffic Manager内で同じ名前を持つユーザに対して指定したアクセスのみが与えられます。

ソフトウェアの更新

vWAFIにはアップデータと呼ばれる機能があり、vWAFソフトウェアコンポーネントへの更新処理を任せることができます。アップデータへは、vWAFの管理UIの**Administration** (管理) > **Cluster Management** (クラスター管理) タブからアクセスできます。詳細については、『Pulse Secure Virtual Web Application Firewall: User Guide』およびオンラインヘルプを参照してください。

TrafficScriptルール

本章では、TrafficScriptのルールについて説明します。これには、新しいルールを作成してそれらをサービスに適用する手順が含まれます。

概要



一部のTraffic Manager製品では、TrafficScriptルールまたはXML機能がサポートされていないか、サポートが限定的(RuleBuilderのみ)である場合があります。完全なサポートは、ソフトウェアまたはライセンスキーをアップグレードすることにより入手できます。

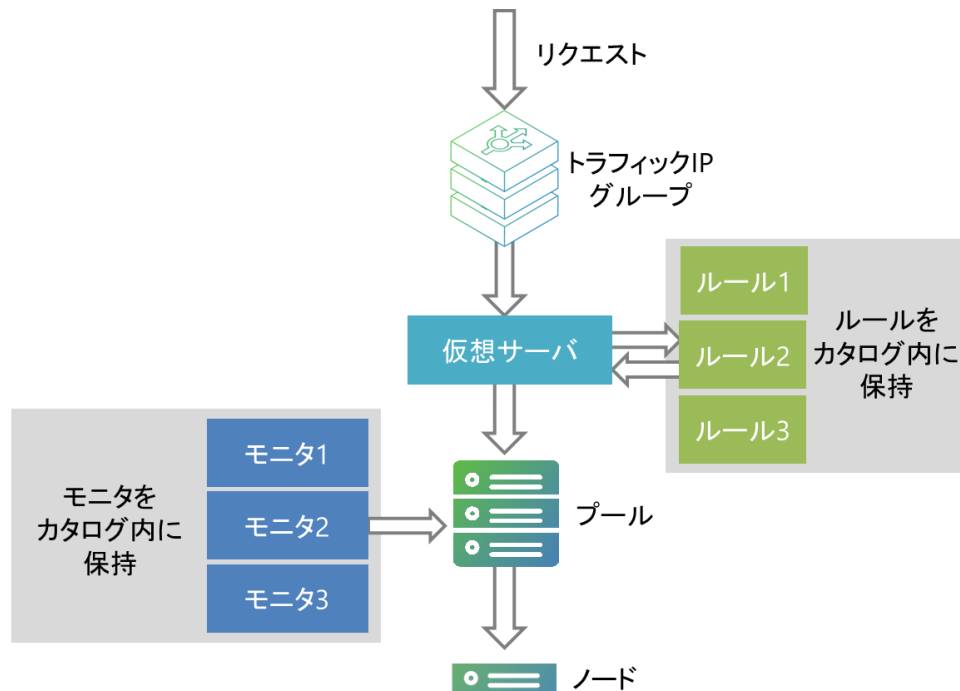
PulseコミュニティのWebサイトのTraffic Managerセクションには、TrafficScriptルールによって様々なネットワークおよびアプリケーションの問題を解決する方法について取り上げた、豊富な例とドキュメントが掲載されています。詳細については、<https://community.pulsesecure.net>を参照してください。

独自のトラフィック管理ルールを使用することにより、Traffic Managerの動作をカスタマイズすることができます。これらのルールは、TrafficScriptと呼ばれるスクリプト言語を使用して作成します。

Traffic Managerは、新しいコネクションまたはネットワークリクエストを受信するたび、ノードからレスポンスを受信するたび、およびトランザクションの論理的完了時にTrafficScriptルールを実行します。これらのルールにより、そのコネクションで受信するデータや外部に送信するデータ、リモートクライアントアドレスなどその他の要素をチェックすることができます。



TrafficScriptルールでは、正規表現を使用してデータを検索、照合、置換できます。ルールで正規表現を使用する前に、[正規表現の使用](#)で説明するセキュリティに関する注意事項を確認してください。



TrafficScriptルールでは、リクエストやレスポンスを変更したり(例: HTTPリクエスト内のURLやヘッダの書き換え)、セッション維持パラメータを設定したり、トランザクションを選択的に記録したり、最適なプールへのリクエストのルーティング方法についてTraffic Managerに通知したりできます。

このように、それぞれのホストの必要条件に応じて設計されたルールを使用することで、Traffic Managerによるトラフィックの管理方法を正確にコントロールすることができます。

TrafficScriptの例

以下のTrafficScriptルールは、HTTPリクエストに対して使用できます。このルールはリクエストを以下のように処理します。

- 「www.example.co.uk」に対するリクエストを「www.example.com」に書き換える。
- .jspページに対するリクエストを一連のアプリケーションサーバ(プール名:「JSPServers」)にルーティングする。
- 「/secure」で始まるURLに対するリクエストは営業時間中のみ許可する。

```
# Rewrite host header if necessary
if( http.getHostHeader() == "www.example.co.uk" ) {
    http.setHeader( "Host", "www.example.com" );
}

$path = http.getPath();
```

```
# Give .jsp requests to the "JSPServers" pool
if( string.endsWith( $path, ".jsp" ) ) {
    pool.use( "JSPServers" );
}

# Deny access to /secure outside office hours
if( string.startsWith( $path, "/secure" ) ) {
    if( sys.time.hour() < 9 || sys.time.hour() >= 18 ) {
        connection.discard();
    }
}
```

以下のTrafficScriptルールは、HTTPレスポンスに対して使用できます。このルールはレスポンスを以下のよう
に処理します。

- ステータスコードが404または5xxの場合、リクエストを最大3回再試行する。
- レスポンスにwww.example.co.ukへの参照が含まれる場合、これらの参照をwww.example.comに変更して書き換える。

```
$code = http.getResponseCode();
if( $code == 404 || $code >= 500 )
{
    if( request.getRetries() < 3 ) {
        # Avoid the current node when we retry,
        # if possible
        request.avoidNode( connection.getNode() );
        request.retry();
    }
}

# We're only going to process text/html responses, so
# break out of the rule if the response is of a
# different type...
if( http.getResponseHeader( "Content-Type" ) != "text/html" ) break;

# getResponseBody automatically de-chunks and uncompresses
# the response if required
$response = http.getResponseBody();

if( string.contains( $response, "www.example.co.uk" ) ) {
    $response = string.regexsub( $response, "www.example.co.uk", "www.example.com", "g"
);
```

```
http.setResponseBody( $response );  
}
```

TrafficScriptに関するドキュメント

TrafficScript言語の構文および関数は、『*Pulse Secure Virtual Traffic Manager: TrafficScript Guide*』を参照してください。IvantiのWebサイト(www.ivanti.com)で、Traffic Manager製品ページからこのガイドにアクセスできます。

TrafficScript編集ページにも、利用可能な関数へのクイックリファレンスリンクがあります。リファレンスを表示するには、ルールの編集時に**TrafficScript Reference** (TrafficScriptリファレンス)リンクをクリックします。

ルールの作成方法については、[カタログでのルールの作成](#)を参照してください。

Traffic Managerでのルールの使用

通常、仮想サーバは、リクエストを処理する適切なプールを選択するためにルールを使用します。ルールにより、リクエストの任意の部分をチェックし、場合によっては変更し、どのプールでリクエストを処理するかを決定できます。

プールに対してセッション維持情報を指定するためにルールを使用できます。ルールでは、リクエストをチェックした後に、`connection.setPersistenceKey()`関数を使用して、セッションを維持するための文字列を設定できます。この文字列は、リクエストが属するセッションを識別するためにユニバーサルセッション維持メソッドにより使用されます。

サーバからのレスポンスをチェックして変更したり、一時的なエラーが検出された場合にリクエストを再試行したり(可能な場合)するために、ルールを使用できます。

ルールでは、仮想サーバまたはプールによってコネクションに割り当てられたクラスをオーバーライドすることにより、各コネクションに対して動作をカスタマイズして指定できます。例えば、低速なリソースへのコネクションには、長い許容レスポンス時間を設定することができます。この方法で割り当てることのできるクラスとして、サービスレベルモニタリング、セッション維持、および帯域制御があります。

サービスプロテクションクラスでルールを使用することができます。仮想サーバにサービスプロテクションクラスが関連付けられている場合、受信パケットがチェックされます。このクラスでルールを使用することにより、パケットが既知のワームやウイルスに一致するかどうかを調べることができます。このルールは、仮想サーバの主要処理を実行する前に実行されます。

セッション維持およびサービスプロテクションでのルールの使用については、それぞれ[セッション維持](#)および[サービスプロテクション](#)で説明します。

Traffic Managerでのルールの使用

TrafficScriptルールは、ルールカタログに保存されます。必要に応じて、ルールカタログを使用してルールを作成したり、外部ソースからルールをアップロードしたり、ルールを変更または複製したり、不要なルールを削除したりできます。

ルールカタログにアクセスするには、**Catalogs** (カタログ) > **Rules** (ルール)をクリックします。

新しいリクエストまたはレスポンスを受信するたび、またはトランザクションの完了時に、カタログに保存されている1つ以上のルールを実行するよう仮想サーバを設定することができます。このようにカタログからルールを使用することで、複数の仮想サーバが同じルールを使用でき、ルールに対する変更は全ての仮想サーバで有効になります。

ルールを使用するには、まず新しいルールを作成してカタログにアップロードします。次に、仮想サーバがそのルールを使用するように設定します(**Services** (サービス) > **Virtual Servers** (仮想サーバ) > **Rules** (ルール)をクリックして、表示される設定を使用してルールを追加します)。

カタログでのルールの作成

新しいルールを作成する

1. **Catalogs** (カタログ) > **Rules Catalog** (ルールカタログ)をクリックします。
2. 「Create new rule (新しいルールの作成)」セクションで、ルールの名前を入力します。
3. TrafficScriptでルールを作成する場合は、**Use TrafficScript Language** (TrafficScript言語を使用)をクリックします。RuleBuilderを使用する場合は、**Use RuleBuilder** (RuleBuilderを使用)をクリックします。
4. **Create Rule** (ルールの作成)をクリックします。

RuleBuilderの使用



ご使用のライセンスキーによっては、RuleBuilderの一部のオプションおよび機能、およびTrafficScriptの使用が制限される場合があります。ここで説明する機能の使用中に問題が生じた場合は、サポート窓口にお問い合わせください。ご使用のソフトウェアまたはライセンスをアップグレードするための最善の方法をご案内します。

RuleBuilderでは、リクエストのチェック条件と、その条件の一部または全てが満たされた場合のアクションを選択することができます。

クライアントのIPアドレスとポート、HTTPパラメータ、クッキー、またはリクエストされたURLに基づく条件を追加できます。条件を作成するときには必ず、その項目のパラメータを入力するための追加のテキストボックスが表示されます。

Rules Catalog

Rule: Content-based routing

Name: [? RuleBuilder Reference](#)

Notes:

Conditions

Any of the conditions must be met before executing the rule's actions:

URL Path starts with

OR

URL Path ends with

Actions

The following actions will be executed:

Choose Pool

Apply Changes

Conditions Actions

Requests and Responses

- Remote IP Address
- Local IP Address
- Remote Port
- ☒ HTTP only
- ☒ SIP only
- ☒ RTSP only

Responses Only

- Response Body
- ☒ HTTP only
- ☒ SIP only
- ☒ RTSP only

次に、ルールで実行する一連のアクションを選択できます。最終アクションの実行前に、リクエストの一部を変更したり、Traffic Managerによりログメッセージを書き込むことができます。最終アクションには、指定プールの選択やコネクションの破棄などがあります。複数の最終アクションを実行することはできません。また、最終アクションを決定するために、必ずしもルールは必要ありません。

例えば、以下のようなルールを作成するとします。

```
IF
    URL Path starts with /servlet
または
    URL Path ends with .jsp
THEN
    Choose Pool: Servlet Runners
```

このルールでは、URLパスが「/servlet」で始まっていないか「.jsp」で終わっていないリクエスト(静的HTMLコンテンツのリクエストなど)は無視されます。このルールを適用する仮想サーバでは、別のルールを使用して、その他のページに対するリクエストを処理したり、リクエストをデフォルトプールに割り当てたりできます。

作成したルールに問題がなければ、**Update** (更新)をクリックします。

ルールをTrafficScriptに変換することもできます。TrafficScript言語では、選択できる条件やアクションの幅および構造が大きく広がります。**Preview Rule as TrafficScript** (ルールをTrafficScriptとして見る)をクリックすると、どのようなTrafficScriptルールになるかを確認できます。**Convert Rule** (ルール変換)ボタンを使用すると、ルールが永続的に変換されます。

Traffic Managerのルールを初めて使用する場合は、まずRuleBuilderを使用して単純なルールを実装することをお奨めします。そのルールに相当するTrafficScriptルールを調べ、TrafficScriptの構文を理解します。複雑なルールを作成する必要がある場合は、最初にRuleBuilderで単純なバージョンを作成し、RuleBuilderの機能の限界に達した時点でルールをTrafficScriptに変換し、さらに編集を行ってください。

特殊変数

RuleBuilderのテキストボックスには、TrafficScript関数に展開されるいくつかの特殊変数を入力することができます。これらの変数を使用すると、より動的なルールを柔軟に作成することができます。

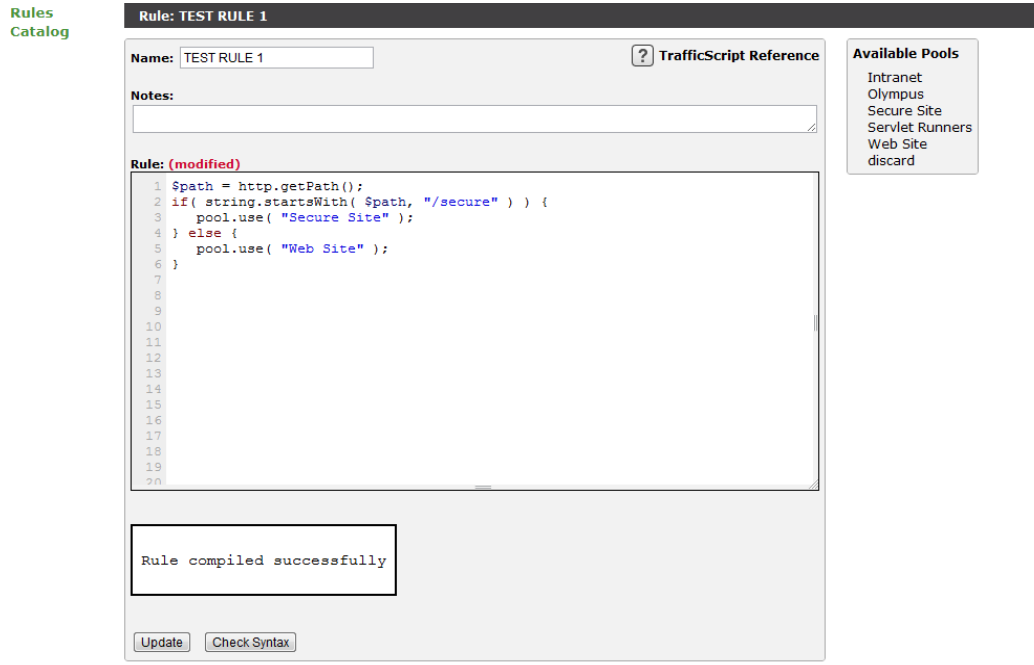
例えば、HTTPヘッダ「X-Forwarded-For」を「%REMOTE_IP%」という値に設定するルールを作成した場合、このヘッダは、Traffic Managerに接続しているクライアントのIPアドレスに設定されます。

特殊変数は以下のとおりです。

- **%REMOTE_IP%**: クライアント接続のリモートIPに置き換わります(TrafficScript関数のrequest.getRemoteIP())。
- **%REMOTE_PORT%**: クライアント接続のリモートポートに置き換わります(TrafficScript関数のrequest.getRemotePort())。

TrafficScriptルールの作成

TrafficScript編集ページには、ルールの説明テキストおよびルール自体を入力し、コンパイル前に構文をチェックすることのできるフォームが表示されます。ルールを別のテキストエディタで作成してから、このページに貼り付けることもできます。



TrafficScript Reference (TrafficScriptリファレンス)リンクをクリックすると、関数のクイックリファレンスが表示されます。また**Check Syntax** (構文チェック)ボタンを使用すると、ルールをチェックできます。完了したら、**Update** (更新)ボタンをクリックします。

一部のTrafficScript関数とRuleBuilderの条件およびアクションは、特定のリクエストルール、レスポンスルール、またはトランザクション完了ルールでのみ有効です。例えば、リクエストのパラメータを変更する関数は、レスポンスルールで使用しても何の効果もありません(リクエストは既にノードに送信されているため)。同様に、コネクションに対して悪影響を伴う関数を使用しようとするトランザクション完了ルール(`http.setResponseHeader()`、`connection.discard()`など)は許可されません。各TrafficScript関数とRuleBuilderの条件およびアクションについては、オンラインヘルプまたは『*Pulse Secure Virtual Traffic Manager: TrafficScript Guide*』を参照してください。

カタログへのルールのアップロード

Traffic Managerでは、あらかじめ作成しておいたTrafficScriptルールをカタログにアップロードできます。ルールはプレーンテキストでなければなりません。また、Traffic Managerはファイル名全体(拡張子も含む)をルール名に使用します。

アップロードされたルールファイルに対し、Traffic Managerは構文チェックを実行します。警告およびエラーは、Rule Catalog (ルールカタログ)ページ、Diagnose (診断)ページ、およびイベントログに表示されます。

カタログヘルールをアップロードする

1. **Catalogs** (カタログ) > **Rules Catalog** (ルールカタログ)をクリックします。
2. 「Upload an existing rule (既存ルールのアップロード)」セクションで**Choose file** (ファイルの選択)をクリックして、ローカルのファイルシステムからルールファイルを選択します。
3. アップロードするファイルと同じファイル名の既存ルールに上書きするかどうかを選択します。
4. **Upload Rule** (ルールのアップロード)をクリックします。

仮想サーバへのルールの適用

新たに追加されたルールは、まだTraffic Managerで使用されていません。ルールを使用するには、そのルールを仮想サーバ設定に追加します。

仮想サーバがルールを使用するように設定する

1. 仮想サーバの**Virtual Servers** (仮想サーバ) > **Edit** (編集)ページにアクセスするには、**Services** (サービス)ボタンをクリックしてから**Virtual Servers** (仮想サーバ)タブをクリックし、仮想サーバの名前をクリックします。
2. **Rules** (ルール)リンクをクリックします。これにより、現在その仮想サーバで使用されているリクエストルール、レスポンスルール、およびトランザクション完了ルールのリストが表示されます。
3. リストに新しいルールを追加するには、各セクションの「Add rule (ルールを追加)」オプションを使用します。ドロップダウンボックスからルールを選択し、**Add rule** (ルールを追加)をクリックします。
4. ルールは指定された順序で実行されます。最初のルールによってトランザクションについての最終決定が行われなかった場合、2番目のルールがテストされ、以下同じように各ルールが順にテストされます。リスト内でルールを移動するには、各ルールバーの左側にあるグラブハンドルを使用します。

多くのプロトコルでは、ルールを一度のみ実行するか(最初のリクエストまたはレスポンスに対して実行)、またはプロトコルダイアログの全てのリクエストとレスポンスに対して実行するかを指定できます。

RTSPおよびSIPでは、Traffic Managerを経由する個々のリクエストまたはレスポンスに対して自動的にルールが実行されます。

HTTPはリクエストとレスポンスが一对一で対応しており、keepaliveコネクションで行われるリクエストは個別に処理されるため、HTTP仮想サーバにはこのオプションは必要ありません。

新しいルールの効果をテストするには、テスト仮想サーバでそのルールの有効化および無効化を切り替えます。

ルールの例

このセクションでは、Traffic Managerがサービスへのトラフィックを検査および操作するために、TrafficScriptルールをどのように使用するかの例を説明します。

コンテンツタイプによるルーティング

この例では、HTTPリクエスト内のURLパスをチェックします。URLパス内のコンテンツタイプに応じて、リクエストを処理するプールを選択します。

ルールの作成前に、「windows」、「java」、「linux」というプールをそれぞれ設定します。

```
$path = http.getPath();
if( string.endsWith( $path, ".jsp" ) ) {
    pool.use( "java" );
} else if( string.endsWith( $path, ".asp" ) ) {
    pool.use( "windows" );
} else {
    pool.use( "linux" );
}
```

時刻に基づくアクセス制限

この例では、特定のサービスへのアクセスを一般的な営業時間中(この例では、月～金曜日の午前9時～午後6時)のみ許可します。それ以外の時間帯に発生したコネクションは全て破棄します。

```
$dayofweek = sys.time.weekDay();
$hourofday = sys.time.hour();

# $dayofweek: Sunday is 1, Saturday is 7
# $hourofday: office hours are between 9am and 5:59pm
if( $dayofweek == 1 || $dayofweek == 7 || $hourofday < 9 || $hourofday >= 18 ) {
    connection.discard();
}
```

実際には、警告せずにそのままコネクションを破棄するよりも、アクセス制限されたトラフィックをサーバの別の「エラープール」に送信する方が適切なことがあります。この場合、コネクションを切断する前に適切なエラーメッセージを返すようエラープール内のサーバを設定します。これを行うための手順は、負荷分散の対象となるプロトコルによって異なります。

カスタマのプライオリティ設定

この例では、HTTPリクエスト内のクッキーをチェックします。クッキーの値を使用して、選択するプールを決定します。一方のプールにはプレミアムユーザ用に予約されたマシンが含まれているため、もう一方のプールより高速です。

ある企業で、カスタマベースを「gold」会員と「silver」会員の2つに分けているとします。この企業では、yellow、green、blue、black、およびpurpleの5台のサーバを運用しており、goldカスタマのリクエストを優先的に処理にしようとしています。

ここで、yellow、green、およびblueマシンで構成される「silver」カスタマ用の「standard」プールと、5台全てのサーバで構成される「gold」カスタマ用の「premium」プールの、2つのプールを作成します。すなわち、blackおよびpurpleは、goldカスタマのみが利用できるサーバです。

このサイトでは、クッキーログインシステムを使用しており、カスタマタイプはクッキーに符号化されています。Traffic Managerは、会員レベルを識別して、トラフィックを適切なプールに送信することができます。

```
$cookie = http.getHeader( "cookie" );
if( string.contains( $cookie, "gold" )) {
    pool.use( "premium" );
} else {
    pool.use( "standard" );
}
```

サービスレベルの管理

この例では、プレミアムカスタマに「premium」サービスレベルモニタリング(SLM)クラスのタグを付け、そのトラフィックを「premium」プールに送信します。

プレミアムカスタマ以外のカスタマは、premium SLMクラスが許容値内で機能しているときはpremiumプールを共有できますが、premium SLMクラスの実行速度が遅い場合は「standard」プールに送信されます。

```
# $isPremium could be based on the presence of a
# login cookie, the contents of the shopping cart,
# or even the remote address of the client.

if( $isPremium ) {
    connection.setServiceLevelClass( "premium" );
    pool.use( "premium" );
} else {
    if( slm.conforming( "premium" ) == 100 ) {
        pool.use( "premium" );
    } else {
```

```

        pool.use( "standard" );
    }
}

```

XMLトラフィックに基づくルーティング

TrafficScriptでは、XMLドキュメントのクエリに使用される業界標準言語であるXPathを使用したXMLドキュメントの解析がサポートされます。

XMLドキュメントは、WebサービスなどSOAPベースのプロトコルで使用され、ユーザが介入しなくても複雑なデータを自動的に交換したり解釈したりすることができます。

XMLドキュメントは、ノードのツリー構造で表され¹、各ノードはデータ、または他のノードを保持することができます。XPathはこれらのノードを検索し、XMLドキュメントから特定のデータを抽出できます。Traffic Managerはこのデータを基に、トラフィックのルーティングに関する決定を行うことができます。

XPath 1.0の仕様は<http://www.w3.org/TR/xpath>で参照できます。

例: Google検索リクエスト

Google™検索エンジンは、検索クエリのSOAPリクエストを受け入れるWebサービスインターフェースを備えています。Ivanti, Inc.の検索リクエストは、以下のXMLボディデータを含むHTTP POSTから成ります。

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <namespl:doGoogleSearch xmlns:namespl="urn:GoogleSearch">
      <key xsi:type="xsd:string">googleUniqueID</key>
      <q xsi:type="xsd:string">Ivanti, Inc.</q>
      <start xsi:type="xsd:int">0</start>
      <maxResults xsi:type="xsd:int">10</maxResults>
      <filter xsi:type="xsd:boolean">false</filter>
      <restrict xsi:type="xsd:string"/>
    </namespl:doGoogleSearch>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

¹ ここでいうノードは、Traffic Managerのバックエンドプールノードとは関係ありません。

```

        <safeSearch xsi:type="xsd:boolean">false</safeSearch>
        <lr xsi:type="xsd:string"/>
        <ie xsi:type="xsd:string">latin1</ie>
        <oe xsi:type="xsd:string">latin1</oe>
    </namespl:doGoogleSearch>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAPボディにはdoGoogleSearchノードが含まれていることに注意してください。このノードには、検索リクエストのパラメータが保持されています。

インターネットサービスはdoGoogleSearchリクエストを実行またはプロキシでき、このサービスへのトラフィックは、Traffic Managerを使用して管理することができます。

例えば、指定された最大結果件数によっては、doGoogleSearchリクエストを分割することが必要になる場合があります。maxResultsが100より大きい場合、リクエストはgoogleLargeプールに送信され、それ以外の場合はgoogleプールに送信されます。

TrafficScriptルールでは、xml.XPath.MatchNodeSet()およびxml.XPath.MatchNodeCount()関数を使用して、SOAPリクエストボディをクエリし、XMLノードをテストすることができます。

```

# Read the entire body of the SOAP/HTTP request
$body = http.getBody();

# XML parameters lie in the "urn:GoogleSearch" XML
# namespace:
$googlens = "xmlns:googlens=\"urn:GoogleSearch\"";

# Test for the presence of a "doGoogleSearch" node.
# If present, get the value of the "maxResults"
# parameter and choose the pool

if( xml.XPath.MatchNodeCount( $body, $googlens, "//googlens:doGoogleSearch" ) ) {
    $maxResults = xml.XPath.MatchNodeSet( $body, $googlens,
                                           "//googlens:doGoogleSearch/maxResults/text()"
    );

    if( $maxResults >= 100 ) {
        pool.use( "googleLarge" )
    } else {
        pool.use( "google" );
    }
}

```

}

サービスに対する認証の提供

本章では、Traffic Managerが外部サービスに対してユーザをリモートで認証する方法について説明します。

概要

ユーザがTraffic Manager経由で提供されるサービスに接続するときに、ユーザのクレデンシャルをTraffic Manager自体ではなく、リモートサーバに保持されているレコードと照合して検証することができます。

Traffic Managerは、ユーザリクエストの認証および承認を提供する多数の一般的な機構をサポートします。

- リモートLDAP認証サービスをクエリするためのTrafficScriptルールの使用
- サードパーティIDプロバイダとの認証ハンドシェイクを実行するSAML (Security Assertion Markup Language)の使用
- KCD (Kerberos Constrained Delegation)用のTraffic Managerの設定



一部のメカニズムは、ライセンスキーによって管理されます。詳細については、サポート窓口にお問い合わせください。

本章では、TrafficScriptベースの認証およびSAML認証のサポートについて説明します。KCDに対するTraffic Managerのサポートの詳細については、[Kerberos制約付き委任のサポート](#)を参照してください。

リモートLDAP認証によるTrafficScriptルールの使用

TrafficScriptルールで認証を使用するには、まず認証コードカタログ(**Catalogs (カタログ) > Authenticators (認証コード)**)でリモートLDAP認証サービス定義を作成します。この方法で作成した認証コードには、TrafficScriptルール内でauth.query()関数を使用することによりアクセスできます。このルールは、認証されるサービス処理する仮想サーバに追加できます。



TrafficScriptはライセンスで管理される機能です製品によっては、ここで説明する機能が使用できない場合があります。詳細については、サポート窓口にお問い合わせください。

認証コードの設定

リモートLDAP認証コードレコードを作成および編集するには管理UIの**Catalogs** (カタログ) > **Authenticators** (認証コード) ページを使用します。新しい認証コードを作成するには、「Create New Authenticator (新しい認証コードの作成)」に名前、ホスト、およびポートを入力してから、**Create Authenticator** (認証コードの作成) をクリックします。Traffic Managerに認証コード編集ページが表示され、そこに新しい認証コードの基本およびLDAP設定が全て表示されます。認証コードを正常動作させるためには、全てのLDAP設定を完了する必要があります。

認証コード編集ページには、以下の設定キーが含まれます。

設定	説明
Name	この認証コードに付ける識別名です。この名前は、認証を行う TrafficScriptルール内のauth.query()関数の呼び出しで使用されます。 ここで認証コードの名前を変更すると、この認証ルールを参照している全てのルールが、新しい名前を使用するよう自動的に更新されます。
Host	接続先のLDAPサーバのホスト名またはIPアドレスです。
Port	接続先のLDAPサーバのポートです。
Note	この認証コードの説明です。
ldap!bind!dn	「バインド」ユーザの識別名(DN)です。バインドユーザは、LDAPサーバへのアクセスおよび認証されるユーザに属するレコードの検索に使用されます。 バインドユーザには、LDAPサーバ上のユーザレコードの検索および読み取り許可が必要です。 バインドユーザが指定されていない場合、Traffic Managerは、認証されるユーザを検索するために匿名ログインを試行します。
ldap!bind!password	ldap!bind!dn で指定されたバインドユーザのパスワードです。
ldap!filter	ベースDNにあるユニークなユーザレコードを抽出するために使用するフィルタです。文字列%uは、認証コードを呼び出すときに指定されたユーザ名に置き換えられます。 一般的なLDAPフィルタの例として、Active Directoryの sAMAccountName=%u、一部のUNIX LDAPスキーマのuid=%uなどがあります。

設定	説明
ldap!filter!basedn	Traffic Managerが、認証されているユーザのレコードを検索するためのベース識別名(DN)です。 このLDAP認証コードで認証される全てのユーザのエントリは、ここで指定するDNの下になければなりません。 一般的なベースDNとして、OU=users、DC=mycompany、DC=localなどがあります。
ldap!attr	Traffic Managerは、認証されるユーザのレコードをLDAPサーバで見つけると、そのレコードから追加情報を取得できます。この情報は、ユーザが属しているグループに基づくアクセス制限など、認証されるユーザに対して追加のチェックを実行するために使用できます。 スペースまたはカンマで区切った属性名のリストをldap!attr設定に指定することにより、ユーザのレコードから特定の属性を取得することができます。 ユーザのレコードから全ての属性を取得するには、ldap!attrを「*」に設定します。 この設定をブランクにすると、サーバから追加属性が読み込まれません。 ユーザのレコードから読み込んだ属性は、認証をリクエストしたTrafficScript関数の戻り値で使用できます。
ldap!ssl	この設定は、LDAPサーバへのコネクションをSSLで暗号化するかどうかを決定します。SSLコネクションの確立方法は、ldap!ssl!type設定で指定できます。
ldap!ssl!type	この設定は、LDAPサーバへのSSLコネクションの確立方法を指定します。ldap!sslが「Yes (はい)」に設定されている場合のみ使用されます。 使用できる方法は以下のとおりです。 LDAPS: Traffic Managerは、LDAPメッセージの送信前に、LDAPサーバに対してセキュアコネクションを確立します。 Start TLS: Traffic Managerは、LDAPv3トランスポートレイヤセキュリティ拡張を使用して、サーバに対してセキュアコネクションを確立します。

設定	説明
ldap!ssl!cert	SSLを通じてリモートLDAPサーバに接続する場合、Traffic Managerでは、サーバの証明書がこの設定で指定された認証局によって署名されていることが確認されます。 サーバが、ここで指定された認証局によって署名されていない証明書を送信した場合、またはHostで与えられるサーバのIDとして指定された値と証明書が一致しない場合、認証コードを使用しているTrafficScript関数にエラーが返されます。 認証局が指定されていない場合、サーバの証明書は無効とされます。

TrafficScriptルールの設定

仮想サーバでリモート認証を使用するには、認証ルールをリクエストルールとして仮想サーバに割り当てます。このルールでは、以下の引数を指定してauth.query()関数を呼び出す必要があります。

```
auth.query( authenticator, user, [password] );
```

この関数は、設定されたauthenticatorに対してユーザに関する情報をクエリし、passwordが指定された場合は、その値がそのユーザのレコードに設定されているパスワードと一致していることを確認します。その後、検証結果に従って設定された「OK」および「Error」の2つの値を含むハッシュを返します。結果には、クエリの対象となったユーザの識別名など、認証コードによって返された追加情報を含めることができます。詳細については、『*TrafficScript Reference*』を参照してください。

TrafficScriptルールの作成の詳細については、[TrafficScriptルール](#)を参照してください。

以下の例は、auth.query()関数を使用して、作成済みの「ldap」という認証コードに基づいてユーザを検証する方法を示しています。

```
# Verify the user's password using an LDAP
# authenticator called 'ldap'
$auth = auth.query( "ldap", $user, $pass );
if( $auth['Error'] ) {
    log.error( "Error with authenticator 'ldap': " . $auth['Error'] );
    connection.discard();
} else if( !$auth['OK'] ) {
```

```
# Unauthorised
http.sendResponse( "403 Permission Denied", "text/html",
                  "Incorrect username or password", "" );
}

# Allow through members of the 'admin' group using
# the 'group' attribute returned by the authenticator
if( $auth['group'] != "admin" ) {
    http.sendResponse( "403 Permission Denied", "text/html",
                      "You do not have permission to view this page", "" );
}
```

仮想サーバの設定

適切な設定を行ってルールを設定したら、認証を有効化する仮想サーバにそのルールを割り当てる必要があります。

1. 管理UIの**Services** (サービス) > **Virtual Servers** (仮想サーバ)セクションに移動し、認証を有効化する仮想サーバを選択します。
2. **Rules** (ルール)セクションをクリックします。
3. 「Request Rules (リクエストルール)」で、認証ルールを選択して**Add Rule** (ルールを追加)をクリックします。

SAMLサービスプロバイダとしてのTraffic Managerの設定



Traffic Managerの設定によっては、SAMLサービスプロバイダ機能を標準で利用できない場合があります。詳細については、またライセンスキーのアップグレードを受けるには、Pulse Secureテクニカルサポートにお問い合わせください。

SAML (Security Assertion Markup Language)は、オープンスタンダードのXMLベースのメッセージフレームワークであり、組織のWebサービスにアクセスを試みるユーザエージェントの認証および承認に関する情報交換に使用されます。

SAMLのデプロイには、一般に以下の2つの参加者タイプが含まれます。

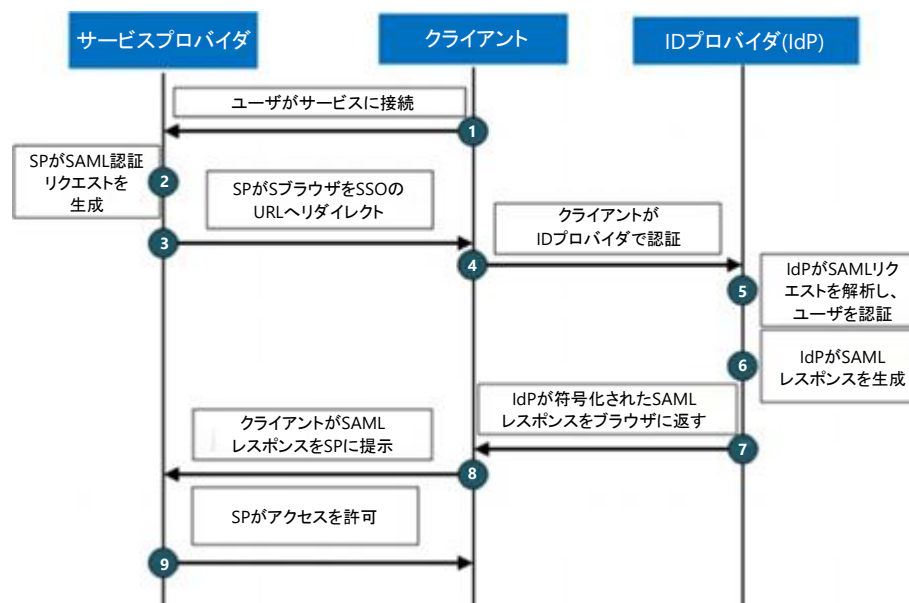
- サービスプロバイダ(SP): 保護対象のリソースを提供します
- IDプロバイダ(IDP): リソースが必要とする認証および承認チェックを実行します。

組織でセキュアなSAMLベースのIDPをデプロイすると、単純なバックエンドLDAPディレクトリサービス認証情報チェックから、1つの企業全体の多くのサービス用の複数要因の認証が関係する、非常に複雑なセキュリティ手法までの、幅広い認証シナリオを処理できるようになります。

Traffic Managerには、SAML SPとして機能して、バックエンドWebアプリケーションへのアクセスを制御する機能があります。アクセスは、リクエストしているユーザエージェントにより有効なSAMLトークンが提示されたときのみ許可され、そのエージェントは定義されたIDPからトークンを取得します。

SAMLメッセージの交換

一般的なシナリオでは、ユーザのブラウザが、特定のサービスへのアクセスを試みて、SPに接続します。SPは以前のバリデーションを取得するよう設定されているため、ブラウザをIDPにリダイレクトします。IDPはユーザの識別子を自分のレコードでチェックし、ユーザが希望するサービスに対する適切な権限を持っていることのバリデーションを取得します。成功した場合、IDPは正当なユーザであるというSAMLアサーション付きでブラウザをSPに戻します。SAML交換がユーザエージェントを通じてSPとIDPの間で行われるため、バックエンドサービスは、発生するセキュリティネゴシエーションから完全に隔離され、それを意識しません。



IDPは、一般にシングルサインオン(SSO)サービスと呼ばれる、複数のバックエンドサービス用の認証を提供できます。同じく、組織はサービスごとに1つずつ、複数のIDPを利用できます。

さらに、IDPは必ずしもSPと同じネットワークに存在する必要がありません。必要なのは、ユーザエージェント（一般にユーザのWebブラウザ）から接続可能なことだけです。共有するSAML設定詳細によって、SPIはIDPを認識し、IDPもSPを認識する必要がありますが、直接通信することはありません。SAMLメッセージはユーザエージェントによって伝達されます。

SAMLとTraffic Manager



Traffic Managerは、SAMLバージョン2.0をサポートし、SPから開始された認証のみを許可します。さらに、Traffic Managerは、暗号化されたSAMLアサーションをサポートしません。

Traffic Managerは、HTTP(S)サービスへのアクセスを制御するSAMLサービスプロバイダ(SAML SP)として設定でき、同じ標準を実装するSAML IDPとの互換性が必要です。

SAML自体には、他の汎用ネットワークプロトコルとのバインディングによる、多様なSAMLプロトコルを必要とする多くの用途(プロファイル)があります。Traffic Managerは、それぞれ認証リクエストとレスポンスに使用されるHTTP-RedirectとHTTP-POSTバインディングを持つ、「Web SSO」プロファイルを実装します。

SAMLリクエストが信頼されるよう、Traffic ManagerはIDP定義の一部として保存された証明書に基づいて暗号署名されたレスポンスを受け付けます。詳細については、[IDプロバイダの定義を参照してください](#)。

Traffic Managerでは、SAMLアサーションをIDPから受信するユーザセッション情報も維持されます。セッションデータには、ログインしているユーザの識別子、認証された時刻(直近のもの)、およびSAML認証トークンが失効する時間が含まれます。これにより、セッションがタイムアウトしたときは、再認証をリクエストできます。

セッション状態はサーバ側に保存されず、セッションクッキーが作成されます。これは、Traffic Managerの既存のセッションチケットメカニズムを使用してTraffic Managerクラスタ間で共有される秘密鍵を使用して暗号化され署名されます。これにより、Traffic Managerでアクティブ-アクティブ設定を利用でき、アクティブなクラスタメンバー間で認証セッションが共有されるようになります。セッションクッキーの動作を設定するには、[IDPを使用するための仮想サーバの設定](#)で説明する設定を使用します。セッションチケットの詳細については、[SSL暗号化](#)を参照してください。

Traffic Managerには、認証されたユーザの名前識別子を返すTrafficScript関数「`http.auth.getAuthenticatedUser()`」が含まれます(フォーマットはIDPで決定されますが、一般的には電子メールアドレスなどになります)。このユーザデータは、例えばHTTPリクエストヘッダの一部として、バックエンドサービスに転送されます。

Traffic Managerの設定

Traffic Managerは、SAMLリクエストおよびレスポンスを互換性のあるIDPと交換できます。Traffic Managerは、特定のIDPから得られた詳細をそのSAML設定の一部としてリクエストし、IDPに特定の設定項目を提供する必要があります。正常に動作するためには、SAML設定が(SPとしての)Traffic ManagerとIDPの両方で一致している必要があります。

IDPの設定は、使用される実際のIDPプラットフォームに応じて異なるため、このドキュメントでは説明しません。認証メッセージをTraffic Managerと交換するためIDPの設定方法の詳細については、IDP管理者にお問い合わせください。

このセクションでは、Traffic Manager側の設定のみを詳しく説明します。

IDプロバイダの定義

Traffic ManagerはIDレコードをSAMLカタログに保存します。その後、各IDPを1つ以上の仮想サーバで使用すると、各サーバが処理するサービスにリモート認証を追加できます。IDP定義にアクセスするには、**Catalogs** (カタログ) > **SAML Catalogs** (SAMLカタログ) > **Trusted Identity Providers Catalog** (信頼されるIDプロバイダカタログ)をクリックします。

以下のテーブルに、IDP定義で必要な設定項目の一覧を示します。

キー	説明
Name	信頼されるIDプロバイダの名前
Entity ID	信頼されるIDプロバイダがSAMLアサーションの「Issuer」フィールドでそれ自体を識別するために送信する識別子。
URL	Traffic ManagerによるSAML認証リクエストの送信先となるURL。
add_zlib_header	SAML標準では、RFC 1951で定義されているDEFLATE圧縮データ形式を使用します。 IDプロバイダがこの標準に従う場合は、「No (いいえ)」に設定します。 IDプロバイダがこの標準に従わず、ZLibフォーマットの使用を必要とする場合は、「Yes (はい)」に設定します。
strict_verify	<i>Conditions</i> 要素に未知の属性が含まれる場合、または要素が不明か欠落している場合に、Traffic ManagerがSAMLレスポンスを拒否するよう強制するには、「Yes (はい)」に設定します。

キー	説明
	互換性の問題が予想される場合は、「No (いいえ)」に設定すると厳密な検証が無効化されます。
Certificate	受信したSAMLアサーションの完全性を検証するためにTraffic Managerが使用するべき公開鍵を含むPEMエンコードされたX.509証明書。

新しいIDPレコードを設定するときは、適切な値についてIDP管理者にお問い合わせください。

IDPを使用するための仮想サーバの設定

認証コントロールをサービスに適用するには、該当する仮想サーバをSAMLサービスプロバイダのエンドポイントとして設定します。**Services** (サービス) > **Virtual Servers** (仮想サーバ) > **Edit** (編集) > **Authentication** (認証)をクリックします。

以下のテーブルに、SAML SPエンドポイントの設定に必要な項目を示します。

キー	説明
auth!type	Traffic Managerがこの仮想サーバにより処理されるリクエストのために使用すべき認証のタイプです。一部の認証タイプはプロトコル固有であり、例えば、HTTP仮想サーバのみに適用されます。 サポートされている以下の認証タイプの1つを選択します。 None: Traffic Managerは追加認証を必要としません。 SAML Service Provider: Traffic Managerはこの仮想サーバへのアクセスで、有効なSAMLアサーションが提示されることを要求します。このオプションは、HTTP仮想サーバでのみ使用できます。
auth!verbose	仮想サーバ認証に関する詳細メッセージをエラーログに記録します。
auth!session!cookie_name	認証セッショントークンを保存するために使用されるクッキーの名前。

キー	説明
auth!session!cookie_attributes	認証セッショントークンを保存するために使用されるクッキーの属性。設定された属性は、そのままクッキーに追加されます。 デフォルト属性では、Javascriptなどのクライアント側APIによるクッキーへのアクセスが防止され、厳密に同じサイトポリシーが適用されます。
auth!session!timeout	セッションの不活動状態が続いた場合に、IDPでユーザの再認証が必要になるまでの秒数。 再認証とは、IDPが新たなトークンを発行することのみを意味する場合があります。ユーザにIDPとの有効なセッションがある場合、設定されたSSOポリシー次第では、この時点で再認証が必要な場合もあれば、不要な場合もあります。
auth!saml!sp_entity_id	認証のためにどのSAML SPがユーザエージェントをリダイレクトしたかを判断するために、IDPが使用する文字列識別子。 仮想サーバに関連しているが、バックエンドサービスの実際のリソースを参照しない一意のURLを使用します。ここで使用されるURLはTraffic Managerの内部で処理され、そこに送信されるリクエストがバックエンドに転送されないため、このURLに存在全てのリソースが使用できなくなります。 SAMLアサーションに「Audience」フィールドが存在する場合は、この値と一致している必要があります。 IDP管理者から、「Audience」という設定フィールドにこの値を指定するよう求められる場合があります。
auth!saml!sp_acs_url	SAMLアサーションコンシューマサービスのURL。すなわち、Traffic ManagerがSAMLアサーションを処理すべきURLです。 バックエンドサービスの実際のリソースを参照しない一意のHTTPS URL(特にライブサービス本番環境のもの)を使用します。ここで使用されるURLはTraffic Managerの内部で処理され、そこに送信されるリクエストがバックエンドに転送されないため、このURLに存在全てのリソースが使用できなくなります。

キー	説明
	SAMLアサーションに「Recipient」フィールドが存在する場合は、この値と一致している必要があります。 IDP管理者から、「Recipient」という設定フィールドにこの値を指定するよう求められる場合があります。
auth!saml!idp	SAML認証リクエストの送信先となり、信頼すべきSAMLアサーションの送信元となる、信頼されるIDPの名前。
auth!saml!time_tolerance	認証チェック時間の許容範囲(秒単位)。 システムの現在時間を使用して、タイムスタンプおよび満了日をチェックするときは、この許容範囲までの時間を許可します。例えば、SAMLレスポンスに含まれる「NotOnOrAfter」が、ローカル時間によると4秒経過したことになっていて、許容範囲が5秒に設定されている場合、このレスポンスは受け入れられます。この設定では、SAMLレスポンスが拒否されてクロック同期が失われることを防止できます。
auth!saml!nameid_format	IDプロバイダにリクエストして得られる名前識別子のフォーマット。以下のオプションを使用できます。 none: SAML認証リクエストでは特定のフォーマット識別子は使用されず、SAMLアサーションの件名にはあらゆる文字列フォーマットを使用できます。 unspecified: フォーマット識別子 「urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified」がSAML認証リクエストで送信され、SAMLアサーションの件名にはあらゆる文字列フォーマットを使用できます。 emailAddress: フォーマット識別子 「urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress」がSAML認証で送信され、SAMLアサーションの件名は電子メールアドレスにする必要があります。

SAML認証のトラブルシューティング

SAML認証は以下の2点の1つで失敗することがよくあります。

- SPからIDPへのハンドオーバー
- IDPからSPへのハンドオーバーの戻り

最初のケースでは、認証試行でIDPからエラーページが送信されます。2番目のケースではエラーページがSPから送信されます。Traffic Managerが監視するメッセージ交換の詳細を取得するには、仮想サーバ設定の **auth!verbose** を有効にします。さらに、問題の診断では以下が役立ちます。

- エラーページ。特にIDPで生成された場合。
- Traffic Managerからの技術サポートレポート(TSR)
- IPDメタデータ。IDPがダウンロードをサポートしている場合。
- SAMLロガーWebブラウザ拡張。使用可能な場合は、SAML交換をキャプチャ。

注記

これらの手法を使用して取得した情報には、暗号シークレットやエンドユーザのパスワードが含まれてはいけません。場合によっては、IDPからSPに渡されたエンドユーザの名前識別子(一般に電子メールアドレス)が、SAMLトレースに含まれることがあります。このようなデータに適用される機密レベルを、組織のセキュリティポリシーで規定しておく必要があります。

Java Extensions

本章では、利用可能なJavaのサポートについて説明します。TrafficScriptからJavaを呼び出すことができるため、さらに強力なルーティングと容易なトラフィック管理が可能になります。

詳細については、『*Pulse Secure Virtual Traffic Manager: Java Development Guide*』を参照してください。

Javaについて

Javaはプラットフォームに依存しないオブジェクト指向の開発言語で、幅広い開発者、ライブラリ、およびアプリケーションに利用されています。Traffic Managerでは、TrafficScriptでJava Extensionsを利用することで、さらに柔軟にトラフィックを操作することができます。これらのJava ExtensionsはTrafficScriptルールから呼び出すことができます。

Traffic Manager Java APIを使用することにより、ユーザおよびインターネットサービスプロバイダは、Traffic Managerの機能を拡張することができます。

TrafficScriptでJava APIを使用すると、以下のような機能を利用できます。

- **コンテンツ圧縮:** 専用のJavaライブラリを使用することにより、XML/HTMLの処理が向上します。
- **追加ライブラリ:** 付加価値ソリューションとして提供され、Traffic Managerの「プラグイン可能な」拡張機能として機能するISVライブラリにより、新たな機能を利用できるようになります。
- **認証:** 外部Webサーバを使用しなくても、RADIUS、TACACS+、LDAPなどを利用できるようになります。

Java Extensionの呼び出し

Java Extensionsは、TrafficScriptまたはRuleBuilderルールから呼び出します。

Java ExtensionをTraffic Managerにインポートすると、そのJava Extensionと同じ名称のRuleBuilder基本ルールが作成されます。このルールによりインポート元のExtensionが呼び出されるため、ルールでトラフィックをチェック、変更、およびルーティングできるようになります。この基本ルールは、仮想サーバが実行するリクエストルールまたはレスポンスルールのリストに追加できます。

注記

全ての新しいTraffic Managerインスタンスで、Javaはデフォルトで無効化されています。Java Extensionを呼び出すには、まずJavaサポートを有効にします。詳細については、[Traffic ManagerのJava Extension Runnerの設定](#)を参照してください。

多くの場合、特定のJava Extensionを全てのリクエストやレスポンスに対して呼び出す必要はありません。例えば、あるJava Extensionは特定タイプのリクエストまたはレスポンスのみを処理します。Java Extensionは、以下のように条件を指定してRuleBuilderまたはTrafficScriptルールから呼び出すことができます。

```
if( http.getPath() == "/serverstatus" ) {  
    java.run( "ServerStatus" );  
}
```

Traffic ManagerでJavaを使用するための設定

このセクションでは、Traffic ManagerでJava Extensionsを作成、設定、および実行するためのプロセスについて説明します。このプロセスの詳細については、『**Java Development Guide**』を参照してください。

必要条件

Traffic ManagerでJava Extensionsを使用するには、Java Run-time Environment (JRE)バージョン1.5以降をインストールします。これより古いバージョンはTraffic Managerでサポートされません。

Traffic Manager仮想/クラウド/アプライアンスインスタンスには、Java Extensionsを実行するために必要なJavaソフトウェアが含まれています。

Java Extensionのコンパイル

Java ExtensionsをTraffic Manager用にコンパイルするには、以下のリソースが必要です。

- Java Development Kit (JDK): Javaコンパイラが含まれています。このコンパイラは<http://www.oracle.com/technetwork/java/index.html>からダウンロードできます。
- Java Servlet APIライブラリ: このライブラリは\$ZEUSHOME/zxtm/lib/servlet.jarにあります。また、**Java Extensions**カタログのリンクからダウンロードすることもできます。
- Traffic Manager Java Extensions APIライブラリ: このライブラリは\$ZEUSHOME/zxtm/lib/zxtm-servlet.jarにあります。また、**Java Extensions**カタログのリンクからダウンロードすることもできます。

Java Extensionをコンパイルするには、以下のコマンドを実行します。

```
javac -cp servlet.jar:zxtm-servlet.jar MyServlet.java
```

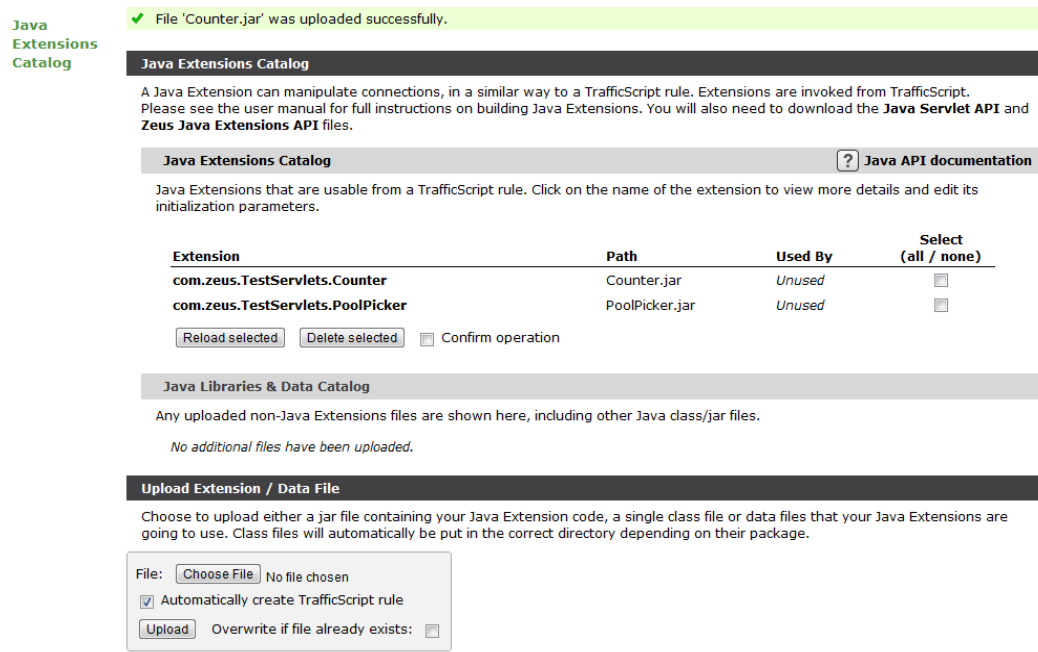
これにより、「MyServlet.class」というクラスファイルが作成されます。上記のコマンドは、servlet.jarおよびzxtm-servlet.jarがコンパイル元のディレクトリにコピーされていることを前提としています。そうでない場合は、コマンドを変更して、それぞれのファイルのフルパスを指定してください。

Java Extensionを、その他の必要なクラスと一緒に単一のjarファイルにパッケージ化することもできます。Traffic Managerは、使用するJava Extensionsのjarファイルを自動的に検索します。

Traffic ManagerへのJava Extensionsのロード

Java ExtensionクラスとjarファイルをTraffic Managerの**Java Extensions**カタログに追加する必要があります。

Extensionをアップロードするには、**Catalog** (カタログ) > **Java Extensions**をクリックして、「Upload (アップロード)」セクションでクラスまたはjarファイルを指定します。Extensionが、Javaの配布パッケージに含まれていない他のJava jarファイルに依存する場合、これらのファイルもカタログにアップロードする必要があります。



Traffic ManagerのJava Extension Runnerの設定

Traffic Managerには、Java ExtensionsをホストするJava Extension Runner(JRE)というJavaヘルパアプリケーションが含まれています。Traffic Managerがこのヘルパアプリケーションを初期化および実行する方法をコントロールするには、管理インターフェースの**System** (システム) > **Global Settings** (グローバル設定) > **Java Extension Settings** (Java Extensionの設定)を使用します。

以下の表に、利用可能な設定を示します。

設定	説明
java!enabled	Traffic ManagerのJavaサポートを有効または無効にします。

設定	説明
	デフォルトでは、全ての新しいTraffic ManagerインスタンスでJavaサポートが無効化されています。Traffic ManagerでJava Extensionsを使用するには、java!enabledを「Yes (はい)」に設定します。
java!command	コマンドライン引数を含むJava実行可能ファイルのパスです。
java!classpath	Javaクラスが保存されているフォルダのコロンで区切ったリストです。
java!lib	Java Extensionsで使用されるライブラリを自動的に検索するフォルダを指定します。Traffic Managerは、このディレクトリからJava Extensionsをロードするわけではありません。
java!max_conns	同時に許容されるJavaリクエストの最大数を定義します。この数を超えるリクエストはキューイングされ、以前のリクエストが完了するまで処理されません。この設定はCPUコアごとに行います。使用できるプロセッサコアの数をこの設定に乗じてください。
java!session_age	Javaセッションを維持するタイムアウトを定義する値(秒単位)です。

Java実行可能ファイルを指定するには、**java!command**に実行可能ファイルの名前を設定します(デフォルトパスでない場合はフルファイルパスを含めます)。JREの実行時に使用するオプションも指定します。

設定を確認するには、**Diagnose**(診断) > **Cluster diagnosis** (クラスタ診断)をクリックし、「Java extensions」セクションにエラーまたは警告が表示されていないことを確認します。これで、Traffic ManagerでJava Extensionsを実行することができます。

プロトコルのサポート

本章では、Traffic Managerでサポートされている特定のプロトコルについて説明します。

概要

Traffic Managerは、TCPコネクションまたはUDPデータグラムに含まれるアプリケーショントラフィックの負荷分散と処理を行います。TrafficScript、負荷分散、セッション維持、帯域制御などのほとんどのTraffic Manager機能、全てのプロトコルタイプに同じように適用することができます。

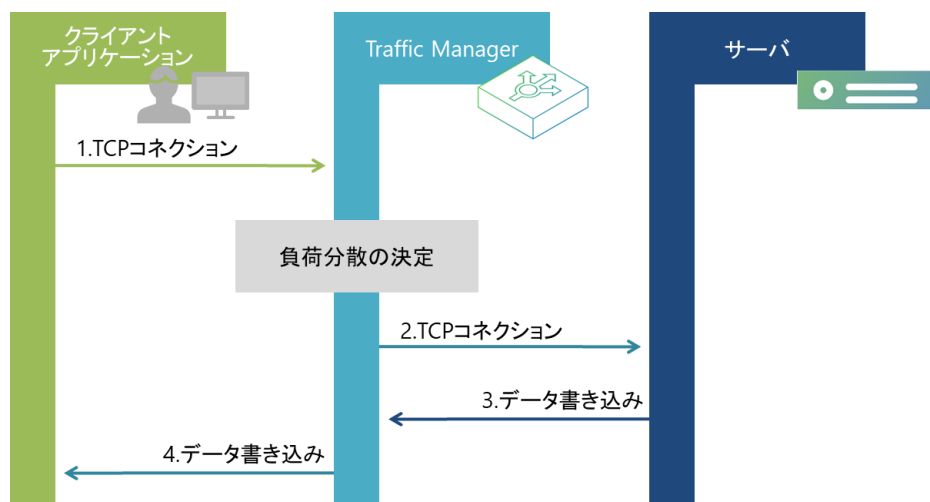
適切な場合は、特定プロトコルに固有のTrafficScript関数を含めることで、そのプロトコルでのトラフィックの管理およびコントロールを容易に行えるようになります。

基本的なTCPプロトコル

「Generic Server First」、「Generic Client First」、および「Generic Streaming」プロトコルは、Traffic Managerで利用できる最も基本的なL7プロトコルタイプです。これらのプロトコルタイプは、トラフィックが特定フォーマットに適合していることが想定されないため、カスタムプロトコルや単純なTCPコネクションを管理する場合に役立ちます。

Server Firstプロトコル

Server Firstは最も単純なTCPプロトコルタイプです。Traffic Managerはクライアントからコネクションを受け取ると、直ちにTrafficScriptリクエストルールを実行し、その後すぐにバックエンドサーバに接続します。以降、Traffic Managerは、両方のコネクション(クライアント側とサーバ側)を監視し、受信したトラフィックを一方のコネクションからもう一方のコネクションへと渡します。

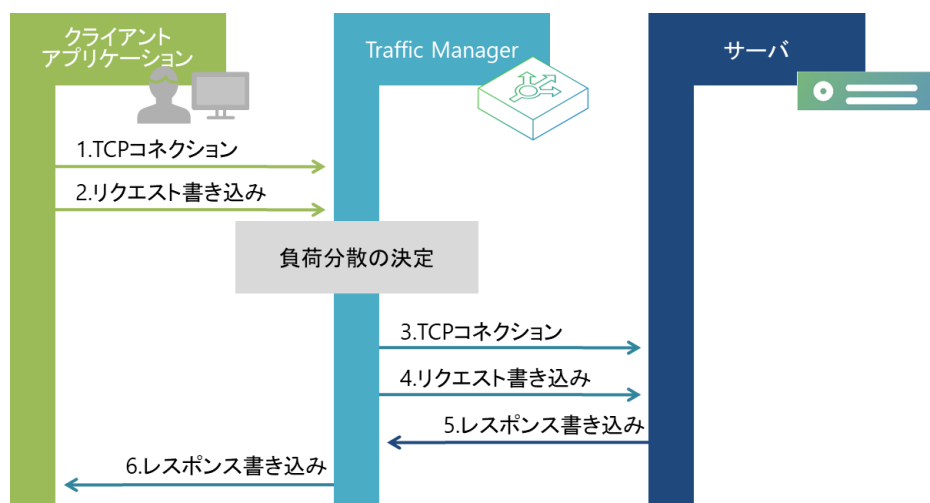


Server Firstプロトコルには、Time (サーバがレスポンスデータを書き込んでから直ちに接続を切断)などのワンショットプロトコルや、MySQL (サーバがパスワード入力のためのダイアログを開く)などの複雑なプロトコルがあります。

「Generic Server First」仮想サーバプロトコルタイプは、接続の確立後にサーバからやり取りを開始する場合に最適です。実際には、多くのプロトコルは設計上「Generic Client First」の方が適しています。

Client Firstプロトコル

Client FirstはServer Firstを改良したもので、接続したクライアントが、サーバからのレスポンスの前にリクエストを送信するプロトコルに適しています。「Generic Client First」プロトコルタイプでは、クライアントのリクエストをチェックして、使用サーバノードの選択に使用するサーバプールを選択することができます。



Traffic Managerには、クライアント接続が確立され、データを受信した時点で初めてアラートが報告されます。その後、Traffic Managerは、Server Firstの方式に従い、TrafficScriptルールを実行し、バックエンドに接続し、データを双方向に渡します。

ルール処理の詳細

Traffic Managerは、TrafficScriptリクエストルールとレスポンスルールを交互に実行します。クライアントから最初のデータを受信したときにリクエストルールを実行します。リクエストルールで`request.getLine()`や`request.get()`などの関数が使用されている場合、クライアントからさらにデータを読み取るためにリクエストルールが中断することがあります。

リクエストルールが完了すると、Traffic Managerは、以降クライアントから受信する全てのデータをサーバに転送します。また、サーバからデータ(すなわち「レスポンス」)を受信すると、Traffic Managerは直ちにレスポンスルールを実行します。これらのレスポンスルールも中断することがあります。

以降、Traffic Managerは、サーバからの全てのデータをクライアントに転送します。その後、クライアントからのデータを待機し、リクエストルールを再度実行します。このように、Traffic Managerは、リクエストデータを待機してリクエストルールを実行し、レスポンスデータを待機してレスポンスルールを実行する処理を交互に繰り返します。

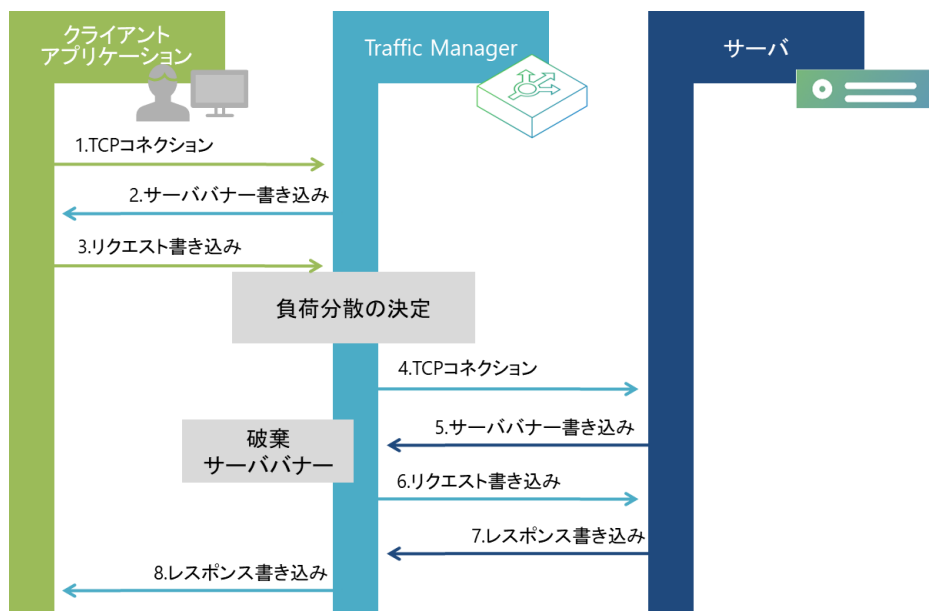
「Run Every (毎回実行)」ではなく「Run Once (一度実行)」として設定されているルールは、最初のサーバデータまたはクライアントデータに対してのみ実行され、以降やり取りされるデータについては実行されません。

「サーババナー」を使ったServer First

サーババナーを使ったServer Firstは、「Server First」を別の形で最適化したもので、SMTPなど、接続時にバナーをブロードキャストするサーバ用に調整されています。「サーババナーを使ったServer First」では、クライアントのリクエストをチェックし、それに基づいて負荷分散の決定を行うことができます。

クライアントが接続すると、Traffic Managerは、直ちに設定されているServer Firstバナーをクライアントに書き込み、その後通常のClient First接続と同じように処理を進めます。また、サーバが送信する最初の行またはデータ(改行で終了)をスラップして破棄します。

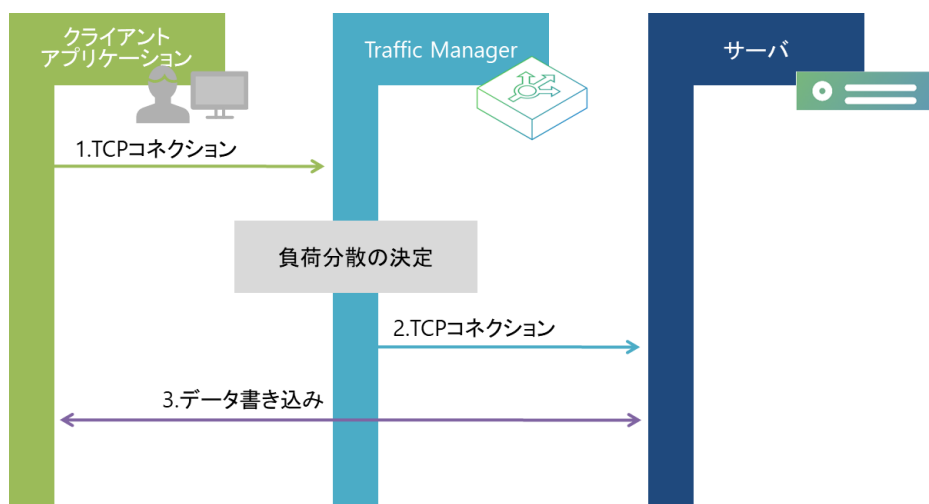
TrafficScriptルールを使用してクライアントのリクエストデータをチェックしてから負荷分散の決定を行います。



この動作を設定するには、「Generic Server First」プロトコルタイプを選択し、仮想サーバの設定のProtocol Settings (プロトコル設定)セクションでバナーメッセージを設定します。

Generic Streamingプロトコル

サーバがクライアントから送信された全てのメッセージに対してレスポンスを返す必要のないプロトコルを使用しているか、TrafficScriptでのデータ処理にさらに柔軟性が必要な場合、「Generic Streaming」オプションを選択できます。このサーバタイプでは、コネクションの確立後、クライアントまたはサーバのどちらからも最初のメッセージを送信することができます。また、そのコネクションでデータを受信するたびにTrafficScriptルールを呼び出すことも可能です。



このオプションは、リクエスト-レスポンスというセマンティックが存在しないプロトコルに適しています。コネクションのどちら側も最初のメッセージを書き込むことができ、レスポンスは必ずしも必要でない、または想定されていません。クライアントがコネクションによってデータを書き込むたびにTrafficScriptリクエストルールが実行され、同様にサーバがコネクションによってデータを書き込むたびにレスポンスルールが実行されます。

HTTP

Traffic ManagerのHTTP仮想サーバプロトコルタイプは、様々な点がHTTPトラフィック用に最適化され、特化されています。

- Traffic Managerは、クライアント側とサーバ側のコネクションを個別に管理します。サーバへの確立済みTCPコネクションおよび新たなTCPコネクションの数を減らすために、可能な場合は、サーバ側でkeepaliveコネクションを再利用します。これにより、サーバが処理しなければならない同時接続数が最小限に抑えられ、性能および処理能力の面で大きなメリットがもたらされます。

HTTP POSTなどの機能を使用する場合やコンテンツ長が欠落している場合は、keepaliveコネクションを安全に使用できません。Traffic Managerはこのような状況を検出し、適宜新たにコネクションを確立します。また、NTLM認証では、keepaliveが明示的に有効化されていることが必要になります。

- Traffic Managerでは、トラフィック管理ルールにおいてHTTPトランザクションのリクエスト-レスポンスという単純な性質に専念できるように、keepaliveの使用やパイプライニングが管理者から見えなくなっています。全てのHTTPトランザクションは、keepaliveコネクションによるものかどうかに関わらず、個別に処理されます。
- Traffic Managerは、gzipおよびdeflateコンテンツ圧縮やチャンク転送エンコーディングなどのHTTP符号化を自動的に処理します。例えば、TrafficScript関数の`http.getResponseBody()`またはJava ExtensionによってHTTPレスポンスをチェックする場合、Traffic Managerは、レスポンスを容易に操作できるよう自動的にデチャンクし、解凍します。
- Traffic Managerには、HTTPリクエストおよびレスポンスの処理を容易に行えるよう特化されたHTTP TrafficScript関数が豊富に用意されています。例えば、基礎となる符号化や、これらの操作を複雑にしている多様なフォーマットおよびプロトコルの詳細を理解しなくても、これらの関数を利用することで、HTTPクッキーの操作、HTTPヘッダ(ホストヘッダなど)の読み取り、URLの処理などを行うことができます。

Traffic Managerでは、HTTP/2プロトコルのクライアント側サポートも提供されます(RFC7540とRFC7541を参照してください)。HTTP/2は、HTTP/1.xに置き変わるもので、特に単一のTCPコネクションでトランザクション多重化をサポートすることにより、待ち時間の長いコネクションでページロード時間を短縮するように設計されています。

Traffic Managerは、HTTP/2をサポートしていますが、以下の注意点があります。

- 一部のブラウザは、暗号化されていないコネクションでHTTP/2をサポートしていません。HTTP/2を使用してサービスにアクセスできるユーザ数を最大にするには、該当する仮想サーバに対してTraffic ManagerのSSL復号化機能を有効化することをお奨めします。
- セキュアなコネクション上のHTTP/2では、TLS 1.2以降、SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256暗号、P256楕円曲線が要求され、コネクションの確立後の再ネゴシエーションはサポートされません。これらのオプションは、デフォルトで有効ですが、以前のリリースからTraffic Managerをアップグレードした後に無効になることがあります。廃止された暗号または旧式のTLSバージョンを使用しているクライアントは、HTTP/2が有効化された仮想サーバに接続するときに、HTTP/1.xリクエストのみを送信できます。
- Traffic Managerは、HTTP/2トランザクションでPulse Secure Web Acceleratorコンテンツ最適化をサポートしていません。

- Traffic Managerでは、クライアントから受信したHTTP/2データが処理前にHTTP/1.1に変換され、仮想サーバのプールノードに転送されます。これにより、バックエンドノード自体にHTTP/2のサポートを要求せずに、Traffic ManagerでHTTP/2機能を引き続き使用できます。
- TrafficScript関数は、HTTP/2コネクションで透過的に動作します。注:
 - クライアントコネクションで使用されているHTTPバージョンを調べるには、TrafficScript関数 `http.getClientVersionNumber()` を使用します。
 - HTTP仮想サーバで `connection.discard()` を使用する場合は、注意することをお奨めします。HTTP/2では単一のコネクションでトランザクションを多重化できるため、コネクションを破棄すると、継続中のトランザクション全てが中断されます。
 - TrafficScript関数 `connection.close()` は、全てのHTTP/1.xサービスとHTTP/2サービスで廃止されました。この関数を使用する全てのTrafficScriptルールを、`http.sendResponse()` と `http.discardClientKeepalive()` を使用するように変更することをお奨めします。詳細については、『*Pulse Secure Virtual Traffic Manager: TrafficScript Guide*』(IvantiのWebサイト: www.ivanti.com から入手可能)を参照してください。

クライアントで使用されているHTTPバージョンを調べるには、**Activity (アクティビティ) > Connections** (コネクション)でコネクションを選択するときに「Request Tracing (リクエストトレーシング)」または「Request Details (リクエストの詳細)」のいずれかを確認します。また、%rまたは%Hロギングマクロを使用して、HTTP/2の使用についてレポートするか、HTTP/2リクエストが実行されるたびに増加するSNMPカウンタ「virtualserverTotalHTTP2Requests」を参照します。

SSL

SSLプロトコルを使用してトラフィックを管理するようTraffic Managerを設定すると、Traffic Managerは、自動的にトラフィックを復号化しません。SSLプロトコルタイプは、SSLトラフィックを変更しないSSLパススルーに使用します。

「SSLセッションID」セッション維持タイプを使用してプール内でSSLコネクションの負荷分散を行うと、同じSSLセッションIDのコネクションを同じバックエンドサーバに送信することができます。一般に、SSLサーバでは相互にSSLセッションIDが共有されないため、使用を検討することが重要です。そのため、クライアントがSSLセッションIDを再利用しようとしても、新しいサーバに送信され、再度SSLセッションについての判別が必要になります。この処理には大きな処理能力が伴い、SSLサーバファームに負荷がかかります。実際、SSLセッションIDセッション維持を使用しない場合、クラスターにSSLサーバを追加すると、SSLセッションのネゴシエーションを頻繁に行わなければならないため、性能が低下します。



SSLトラフィックを復号化し、内部プロトコル(HTTPなど)を使用して処理するには、内部プロトコルを使用するよう仮想サーバを設定し、仮想サーバの設定でSSL復号化を有効化する必要があります(SSLの設定の詳細については、[SSL復号化](#)および[SSL暗号化](#)を参照してください)。

SSLハンドシェイクの保護

DoS攻撃から防御するには、SSLコネクションのネゴシエーションを行うときにクライアントが試行するピアハンドシェイクメッセージのサイズに、制限を設定することができます。**System (システム) > Global Settings (グローバル設定) > SSL Configuration (SSL設定)**の`ssl!max_handshake_message_size`設定を使用します。

この設定の値によって、SSLコネクションで受け入れられるSSLハンドシェイクメッセージの最大サイズが決まります。ハンドシェイクメッセージのサイズがこの設定を超えるとSSLコネクションに失敗し、イベントログにはDoS攻撃の可能性があるとして記録されます。デフォルト値は10KiBで、連続して5つの証明書を受け入れることができるので、通常のSSL設定ではこれで十分です。この値に0(ゼロ)を指定すると、Traffic Managerで処理されるSSLハンドシェイクメッセージのサイズは無制限になります。

SSLコネクションの再ネゴシエーション保護

SSL 3.0とTLS (トランスポートレイヤセキュリティ)のどちらのプロトコルでも、クライアントまたはサーバがセキュアコネクションの再ネゴシエーションを開始できます。これを利用して、コネクションを管理する新しい暗号化パラメータを設定することができます。

再ネゴシエーションの際には、クライアントとサーバとの対話にプレフィックスとしてトラフィックを挿入しようとする攻撃者によって、セキュアコネクションが傍受に対して脆弱になる可能性があります。中間者(MitM)攻撃として知られるこのメカニズムは、攻撃者から送信される初期データを本来のクライアントと同じエンティティから送信されたように見せかけてターゲットサーバを欺くことで、攻撃者側から新しいクライアントコネクションを確立するために使用できます。

また、再ネゴシエーションをサービス妨害(DoS)攻撃に使用することも可能です。クライアントよりサーバに対して高い負荷をかけることになるからです。クライアントは、高い頻度で再ハンドシェイクを繰り返しリクエストすることで、サーバに極端に大きな処理負荷をかけます。

RFC-5746 (<http://tools.ietf.org/html/rfc5746>)では、このような動作を実行しているクライアントに対し、暗号化した再ネゴシエーションを試行することによってMitM攻撃を防ごうとするTLS拡張が導入されました。ただし、この拡張が導入されたからといって必ずしもDoS攻撃を予防できる、あるいは攻撃から身を守れるとは限りません。

RFC-5746 Renegotiation Indication Extensionは、RFC-7627(<http://tools.ietf.org/html/rfc7627>)で説明されているようにトリプルハンドシェイク攻撃に対して脆弱であることが判明しました。このRFCでは、TLSバージョン1.0～1.2に対する「拡張マスタシークレット」の使用によってこの攻撃を防御するための別のTLS拡張についても紹介されています。

Traffic Managerが再ネゴシエーションハンドシェイクを処理する方法を設定するには、**System** (システム) > **Global Settings** (グローバル設定) > **SSL Configuration** (SSL設定) にある`ssl!allow_rehandshake`の必須設定を選択します。使用できるオプションは以下のとおりです。

設定	説明
Always allow	クライアントがRFC-5746またはRFC 7627をサポートしていない場合でも、再ネゴシエーションを常に許可します。ただし、第三者による介入に対しては全く保護されなくなります。この設定は、システムがMitMまたはDoS攻撃を受けやすくなるため、お奨めできません。
Allow safe re-handshakes [デフォルト設定]	クライアントがRFC-5746をサポートする場合と、コネクションを介してデータがまだ送信されていない場合がこれに該当します。前述した攻撃は先頭に付けられたプレーンテキストで構成されているため、データの受信前であれば、再ハンドシェイクは安全です。クライアントの観点から言えば、このオプションの用途は限定的です。再ハンドシェイクは実際には最初のハンドシェイクであるため、コネクションが攻撃を受けているかどうか、クライアントには判断できません。したがって、このオプションは主に、アップグレード前でまだRFC-5746をサポートしていないクライアントに後方互換性を持たせることを目的としています。再ネゴシエーションは、拡張マスタシークレットを使用してハンドシェイク中に元々作成されたものではないセッションの再開後も妨げられます。この設定では再ハンドシェイクが許可されるため、システムはやはり前述のDoS攻撃を受けやすい可能性があります。
Only if client uses RFC 5746 (Secure Renegotiation Extension)	再ネゴシエーションハンドシェイクは、RFC-5746で定義されているセキュアな再ネゴシエーションがクライアントでサポートされている場合にのみ許可されます。再ネゴシエーションは、RFC 7627拡張マスタシークレットを使用してハンドシェイク中に元々作成されたものではないセッションの再開後に妨げられます。この設定では再ハンドシェイクが許可されるため、システムはやはり前述のDoS攻撃を受けやすい可能性があります。

設定	説明
Never allow	再ネゴシエーションハンドシェイクで全ての試行を拒否します。これは、MitMとDoSのどちらの攻撃に対しても保護されるため、最もセキュアなオプションです。ただし、システムの一部がクライアント側の再ネゴシエーションに依存している場合には、サービスエラーが発生する可能性があります。

SSL 3.0/TLS再ハンドシェイクが許可されている場合、**ssl!min_rehandshake_interval**設定を使用して、単一のSSL 3.0/TLSコネクションのハンドシェイク間で許可される最低時間間隔(ミリ秒単位)を定義することができます。この時間間隔は、開始ハンドシェイクが始まると開始され、再ハンドシェイクの開始時点で必ずリセットされます。ハンドシェイクの最低間隔を無効化するには、0 (ゼロ)に設定します。

Traffic ManagerがSSLサーバとして機能している(仮想サーバで**ssl_decrypt**が「Yes (はい)」に設定されている)場合、再ネゴシエーションがTraffic Manager自体によりTrafficScript関数**ssl.requirecert()**を用いて実行されると、その結果生じる再ネゴシエーションはレート制限の対象にはなりません。

SMTP (Simple Mail Transport Protocol)

SMTPは、機能が1つ追加されたServer Firstプロトコルです。セキュリティを強化するために、SSLクライアントは、SMTPコネクションをSSL (TLSとも言う)で暗号化するようリクエストすることができます。

SMTPプロトコルを使用するように仮想サーバを設定した場合、そのプロトコルは通常のGeneric Server Firstプロトコルのように動作します(詳細については、[Server Firstプロトコル](#)を参照してください)。Server Firstパナーで仮想サーバを設定することをお奨めします。詳細については、「[サーバパナー](#)」を使った[Server First](#)を参照してください。

SMTP仮想サーバでSSLを有効化すると、Traffic Managerでは、**smtp!expect_starttls**という設定が**Virtual Server** (仮想サーバ) > **Edit** (編集) > SSLページに追加されます。これは以下のように設定します。

- **No** (いいえ): 仮想サーバは、通常の方法でSSLトラフィックを復号化します。言い換えれば、コネクションの最初に、トラフィックがSSLで暗号化されていると想定します。
- **Yes** (はい): 暗号化されていないかのように、プレーンテキストとしてトラフィックが処理されます。Traffic Managerは、クライアントからの「STARTTLS」SMTPコマンドを監視し、このコマンドを受信すると、SSLハンドシェイクを開始してクライアントのコネクションをSSLにアップグレードします。

Traffic Managerは、プール設定でSSL暗号化が有効化されていない限り、全てのトラフィックを暗号化されていない状態(プレーンテキスト)でバックエンドSMTPサーバに転送します。SSL暗号化を有効化した場合、Traffic Managerは、STARTTLS SMTPコマンドを使用せずに、最初からコネクションを暗号化します。

FTP

FTPプロトコルは、クライアントからサーバにファイルを転送するために使用します。FTPセッションは以下のコネクションから成ります。

- クライアントからリモートサーバのポート21 (一般的な設定)へのコントロールコネクション。このコネクションは、Traffic Managerによって受け取られ、プロキシされます。
- コントロールコネクションによって送信されたコマンドの結果、即座に確立される1つ以上のデータコネクション。これらのコネクションは、サーバからクライアントに対して確立される場合と(「アクティブモード」)、クライアントからサーバに対して確立される場合(「パッシブモード」)があります。

Traffic Managerの「FTP」仮想サーバタイプでは、追加のデータコネクションを確立するためにリクエストを傍受して管理する完全なFTPプロキシが行われます。Traffic Managerは、コントロールコネクションをプロキシするときに同様にデータコネクションをプロキシし、アクティブモードとパッシブモードの両方のデータ転送をサポートします。

このプロキシでは、long (RFC 1639)バージョンおよびextended (RFC 2428)バージョンを含む、一般的なFTPデータコネクションコマンドが全て完全にサポートされます。ただし、RFC 4217暗号化コントロール命令文(「AUTH TLS」、「AUTH SSL」、「CCC」)はサポートされず、コントロールストリームから削除されます。



Consider source port (ソースポート考慮)は、FTP仮想サーバが使用しているトラフィックIPアドレスでは使用しないでください([トラフィックIPグループの作成](#)を参照)。

FTPセキュリティ

FTPはセキュアなプロトコルではありません。パブリックインターネットで一般的な「匿名」ファイル転送には、パスワードが使用されず、パスワードが必要な場合もプレーンテキストで送信されるため、盗聴される可能性があります。このような状況が懸念される場合は、SCP、WebDAV、SSL-wrapped FTP (本章の「SSL-wrapped FTP」セクションを参照)など他のプロトコルを使用してください。

もう1つの懸念事項は、クライアントとサーバのデータコネクション準備を行うダイアログを盗聴者がのぞき見し、コネクションに入り込んでトラフィックの監視側とのコネクションを容易に開始できることです。

例えば、アクティブなFTPコネクションを確立するときに、クライアントはサーバに対して、データコネクションを確立するためにサーバが接続すべきローカルクライアントポートを通知します。このとき、盗聴者がサーバより先にクライアントのローカルポートに入り込んで接続し、クライアントから送信されるデータをキャプチャする(またはクライアントに偽のファイルを返す)可能性があります。

FTPプロトコルには、コネクションを認証してこのような状況を防ぐためのセキュリティ対策が組み込まれていませんが、現代のFTPクライアントおよびサーバの多くは、受信するコネクションの送信元IPアドレスをチェックし、通信相手であるリモートFTPエージェントから送信されたものかどうかを確認します。Traffic Managerでは、この動作がデフォルトで有効化されます。**Virtual Server** (仮想サーバ) > **Edit** (編集) > **Protocol Settings** (プロトコル設定)をクリックし、このページの**FTP-Specific Settings** (FTP固有設定)セクションで以下の設定を確認してください。

設定	説明
ftp_force_client_secure	全てのパッシブデータコネクションの接続元がクライアントのIPアドレスであることを確認します。デフォルト:「Yes (はい)」。
ftp_force_server_secure	全てのアクティブデータコネクションの接続元がサーバのIPアドレスであることを確認します。デフォルト:「Yes (はい)」。
port_range	データコネクションに使用する宛先ポートの範囲を指定します。これらのポートは、例えばクライアントがTraffic Managerに対してパッシブコネクションを確立できるように、介在するファイアウォールによって許可されていない場合があります。

FTPソースポート

FTP仕様では、FTPサーバデータコネクションに、FTPサービスポートより1つ小さい番号のソースポートを使用することが推奨されています。一般的なFTPサービスポートはポート21であることから、推奨されるデータソースポートはポート20です。

ところが実際には、大部分のFTPクライアントおよびサーバでこの推奨内容が無視されています。これは、このようなポートの使用がセキュリティに影響するにも関わらず、保証や認証のための追加措置が提供されていないためです。デフォルトでは、Traffic Managerは、Traffic Managerから開始するFTPデータコネクション用に、高いポート番号をランダムに選択します。

ソースポートの指定

仮想サーバが使用すべきソースポートを指定するには、**ftp_data_source_port** (**Virtual Server** (仮想サーバ) > **Edit** (編集) > **Protocol Settings** (プロトコル設定) > **FTP Settings** (FTP設定))を使用します。この操作は、一般に、アップストリームでファイアウォールを使用して、外部に送信されるコネクションのうち明示的ポリシー(ソースポートとIPなど)によって許可されたもの以外をブロックする場合に必要になります。

Unix/Linux環境のFTPソフトウェアは、ポート20など低い番号のポート(1024未満)からリクエストを送信するための特定のパーミッションを必要とします。デフォルトでは、Traffic Managerは、早い段階でこれらの許可を破棄するため(最小権限の原則)、**ftp_data_source_port**を低い値に設定しようとすると失敗します。低いポート番号を使用するには、**System** (システム) > **Global Setting** (グローバル設定) > **System Settings** (システム設定)で **ftp_bind_data_low**設定を有効化します。これにより、Traffic Managerはパーミッションの制限を放棄するため、低い番号のポートにバインドできるようになります。

現代のLinuxプラットフォームでは、低い番号のポートへのバインド権限のみが維持され、UNIX系製品では、完全なルート権限が維持されます。

SSL-wrapped FTP (FTPS)

Traffic Managerは、RFC 4217以前のFTPS (「Implicit FTPS」または「SSL-wrapped FTP」と呼ばれます)の使用をサポートしています。この実装では、コントロールポートは最初から暗号化されます。また、一般にデータコネクションも暗号化されますが、必要に応じてプレーンテキストで動作させることも可能です。

SSL-wrapped FTP用の仮想サーバプロトコルタイプは「FTP」です。この場合、仮想サーバとプールは、コントロールチャネル通信を復号化または暗号化するよう個別に設定することができます。

コントロールチャネルのコントロール

以下の設定を使用します。

- **ssl_decrypt** (**Virtual Server** (仮想サーバ) > **Edit** (編集) > **SSL Decryption** (SSL復号化)): クライアント側コネクションに対してSSL復号化を有効にします。SSL復号化の設定の詳細については、[SSL復号化の設定](#)を参照してください。
- **ssl_encrypt** (**Pool** (プール) > **Edit** (編集) > **SSL Settings** (SSL設定)): コントロールコネクションの暗号化を有効にしてから、Traffic Managerがそれをバックエンドサーバに送信します。詳細については、[SSL暗号化](#)を参照してください。

Traffic ManagerのFTPプロキシでは、確立済みのSSL-wrapped FTPセッションにRFC 4217の暗号化コントロール命令文(「AUTH TLS」、「AUTH SSL」、「CCC」)が干渉できないよう、これらの命令文が無効化されます。

データチャネルのコントロール

Traffic Managerは、データチャネルを復号化することも、変更せずにそのままにすることもできます。この動作は、**Virtual Server** (仮想サーバ) > **SSL Decryption** (SSL復号)の**ftp!ssl_data**設定でコントロールします。

- **ftp!ssl_data**設定を有効化すると、仮想サーバは、コントロールポートに使用されているものと同じ公開証明書を使用して全てのデータコネクションを復号化します。この場合、**ssl_decrypt**を使用して、SSLを復号化するようFTP仮想サーバも設定しなければならないことに注意してください。

- 無効化した場合、仮想サーバはデータを変更せずにそのまま渡します。バックエンドサーバがデータコネクションでSSLの使用を希望している場合、クライアントは、Traffic Managerを中間プロキシとして使用して、バックエンドサーバと直接ネゴシエーションを行うことができます。

サーバ側では、以下のいずれかの場合に、Traffic Managerがデータコネクションを暗号化することに注意してください。

- 仮想サーバの**ssl_decrypt**設定が無効化され(仮想サーバがSSLを復号化しない)、プールの**ssl_encrypt**設定が有効化されている(プールはデータチャネルのサーバ側を暗号化)場合。この場合、クライアントとのデータ通信はプレーンテキストで行われます。
- 仮想サーバの**ssl_decrypt**および**ftp!ssl_data**設定の両方が有効化され(仮想サーバがSSLを復号化する)、プールの**ssl_encrypt**設定が有効化されている(プールはデータチャネルのサーバ側を暗号化)場合。この場合、クライアントとのデータ通信は暗号化されます。

SSL-wrapped FTPの使用例

使用例1: サーバとクライアントがSSL-wrapped FTPをサポート

この使用例では、関与する全てのパーティがSSL-wrapped FTPをサポートして使用し、Traffic Managerは、クライアントをサーバに接続する負荷分散プロキシとして機能します。Traffic Managerは、コントロールコネクションをチェックできるよう内部的に復号化し、その後再度暗号化します。また、データコネクションについても復号化して再度暗号化することがオプションで可能です。

- SSL復号化(仮想サーバ: **ssl_decrypt**)およびSSL暗号化(プール: **ssl_encrypt**)を有効化します。これにより、Traffic Managerは、コントロールチャネルをチェックしてプロキシし、データコネクションを動的に確立することができます。
- オプション: **ftp!ssl_data**を有効化します。無効化した場合、クライアントとサーバは、データコネクションで暗号化を使用するかどうかを判別することができます。有効化した場合、Traffic Managerは、データ転送にSSLを使用するようクライアントにリクエストします。



ftp!ssl_dataを有効化すると、以下の2つのメリットが得られます。1つ目のメリットとして、全てのコネクションが暗号化され、Traffic Managerの仮想サーバ証明書と照合して認証されるため、証明書の管理が容易になります。2つ目のメリットとして、Traffic Managerは全てのクライアントに対し、データコネクションにSSLコネクションを使用するよう強制します。

ftp!ssl_dataを使用しない場合、Traffic Managerとバックエンドサーバ間で証明書を同期する必要があり、クライアントおよびサーバがプレーンテキストで(暗号化されていない)データコネクションするのを防ぐことができません。ただし、これらのメリットには、Traffic Managerシステムで追加のSSL処理が生じるという代償を伴います。

使用例2: クライアントがSSL-wrapped FTPをサポートし、サーバは非サポート

この設定は、SSLをサポートしていないFTPサーバや、Traffic ManagerとFTPサーバが非常に近接しているためTraffic ManagerおよびFTPサーバ間のネットワークコネクションをセキュアにする必要がない状況で使用できます。

- SSL復号化(仮想サーバ: **ssl_decrypt**)は有効化しますが、SSL暗号化(プール: **ssl_encrypt**)は有効化しません。この場合、Traffic Managerは、クライアントからのSSL-wrapped FTPを復号化し、プール内のFTPサーバにプレーンテキストで渡します。
- 推奨: **ftp!ssl_data**を有効化します。無効化した場合、クライアントとサーバでは常にプレーンテキスト(暗号化されていない)コネクションによってデータ転送が行われます。有効化した場合、Traffic Managerは、データ転送にSSLを使用するようクライアントに強制します。

この使用例の方法では、サーバを変更することなく、FTPサービスをSSL-wrapped FTPに簡単にアップグレードすることができます。

Real Time Streaming Protocol

Real-Time Streaming Protocol (RTSP)を使用すると、クライアントは、オーディオおよびビデオサーバなどの連続メディアの単一または複数の時間同期ストリームを確立し、コントロールすることができます。RTSPに基づくアプリケーションは、マルチメディアサーバにおいてビデオのようなリモートコントロールパネルとして機能します。同一カンファレンス内でSession Initiation Protocol (SIP)とRTSPを組み合わせて使用するサーバもあります。

RTSPは、QuickTime、MPlayer、VLC、Windows Media Player、RealPlayerなどのクライアントアプリケーションで使用されています。

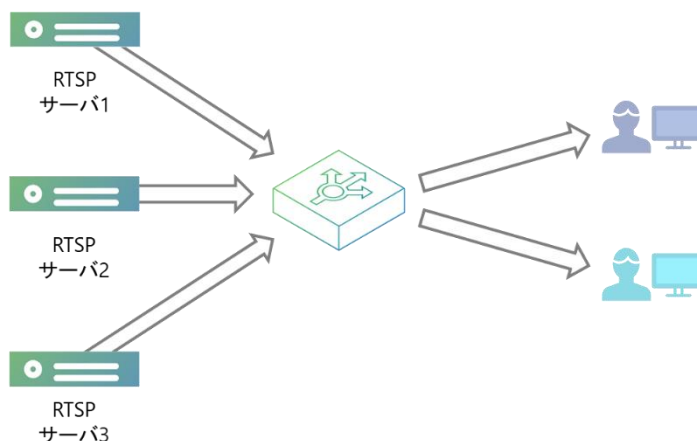
Traffic Managerは、RTSPトラフィックを認識して管理できるため、ユーザは、このタイプのトラフィックを管理するメディアサーバのプールで負荷分散を行うことができます。

下図に、RTSPコネクションの一般的なレイアウトを示します。RTSPクライアントは、Traffic Managerを介してRTSPサーバへのTCPコネクションを確立し、このコネクションによってRTSPリクエストを送信します。これはコントロールチャンネルです。メディアデータ自体は、UDPで個別に送信することも(一般的なシナリオ)、同じコネクションを使用してTCPで多重化することもできます。

または、HTTPを使用してデータ配信することも可能です。基礎となる転送プロトコルとしてUDPまたはTCPを使用する場合、クライアントとサーバがコネクションの確立時に判別を行うことに注意してください。この設定はTraffic Managerからは行いません。

RTSPコンテンツのソース
(オーディオ、ビデオ)

RTSPクライアント(Realplayer、
Windows Mediaなど)



RTSP自体は、メディアの転送のみをコントロールします。メディアトランザクション全体では、以下のようなプロトコルが関与します。

- **RTP**: Real-time Transport Protocol。データを転送するために使用されます。
- **RTCP**: Real Time Control Protocol。RTPストリームに関する追加情報を提供するために使用されます。
- **RDТ**: Real Data Transport。データストリームを転送する、Real Networks社独自のデータ配信フォーマットです。

機能的には、RTSPの処理にTraffic Managerを使用すると、以下のようなメリットが得られます。

- サーバの負担が軽減され、RTSPサーバのプールにわたる負荷分散が可能
- TrafficScriptルールを使用することで、性能のカスタマイズおよび向上が可能
- サーバでの障害発生時にセッション維持を提供

RTSPサービスの設定

デフォルトでは、RTSPはポート554で動作します。RTSPサービスをTraffic Managerで管理するには、ポート554 (または、適切な場合は別のポート)で監視する仮想サーバを作成し、プロトコルタイプ「RTSP」を選択します。

UDPでコンテンツをストリーミングする場合、仮想サーバでは追加のポートを使用する必要があります。ポートの範囲を指定するには、RTSP仮想サーバの**Protocol Settings** (プロトコル設定)ページを使用します。「RTSP-Specific Settings (RTSP固有設定)」セクションで、ポート範囲を変更できます。このとき、可能な限り広い範囲を入力することをお奨めします。

Protocol Settings

Virtual Server: RTSP Service (RTSP, port 554)

Unfold All / Fold All

Settings controlling how the virtual server communicates with the remote client.

▼ RTSP-Specific Settings

How the virtual server handles RTSP traffic.

If non-zero data-streams associated with RTSP connections will timeout if no data is transmitted for this many seconds.

rtsp_streaming_timeout: 30 seconds

The virtual server needs to bind to additional ports for UDP streams between the client and server to allow streaming data delivery. You may need to allow these ports through your firewall.

Specify a port range: ☐ Yes ☒ No

Lowest port: Highest port:

必要なポートの合計数は、転送方法と使用サーバ、クライアントが使用するプレイヤー、ストリーミングするファイルのフォーマットによって異なります。例えば、アクティブなクライアントが100台含まれ、クライアントあたり最大8個のポートが必要な場合、合計800個のポートが必要になります。

Session Initiation Protocol

Session Initiation Protocol (SIP)は、単一または複数の参加者とのセッションを開始、コントロール、変更、または終了することのできるアプリケーションレイヤコントロールプロトコルです。通常、SIPは、電話やビデオ通話、マルチメディア配信やカンファレンスのストリーミングに使用されます。SIPプロトコルでは、以下のような機能が提供されます。

- ユーザロケーション
- ユーザの可用性
- ユーザの機能
- セッションの設定
- セッションの管理

SIPの機能

このセクションでは、SIPの機能について詳しく説明します。

- **ユーザロケーション:** ユーザがSIPデバイス(SIP電話など)をアクティブ化すると、デバイスによりその現在位置がSIPサーバに登録されます。以降、サーバのドメインでそのユーザとコンタクトを取ることができます。各ユーザは、異なる場所にある複数のデバイスを同時にサーバに登録することができます。

誰かがそのユーザとコンタクトを取ろうとすると、サーバにメッセージが届きます。サーバは、そのユーザについて登録されている場所から、メッセージの転送先として1つまたは複数の場所を選択します。

- **ユーザの可用性:** SIPデバイスはメッセージを受信した場合に、いくつかの方法でレスポンスを返すことができます。レスポンスは、例えばユーザが着呼に応答するなど、ユーザ入力によって決めることができます。また、例えばデバイスが使用中の場合に「ビジー」レスポンスを返すなど、レスポンスを自動的に生成することも可能です。このレスポンスをどのように処理するかはサーバが決定し、例えばレスポンスを呼び出し元に返したり、元のメッセージを別の場所に送信したりできます。

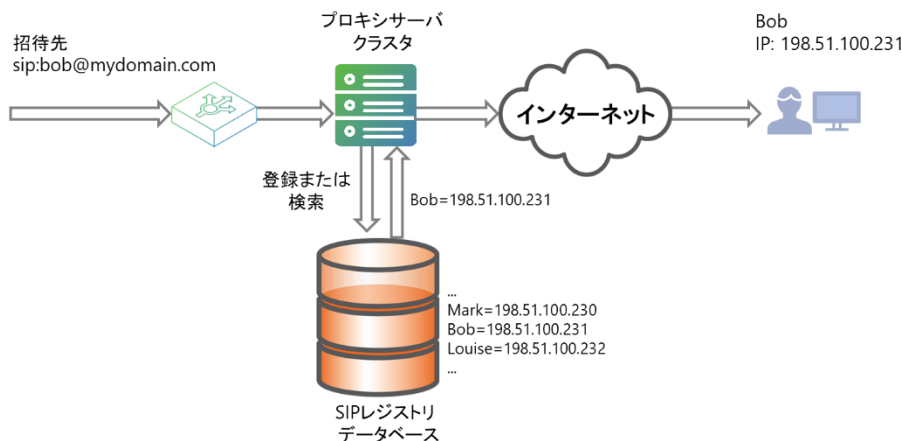
ユーザがサーバにデバイスを登録していない場合は、使用不可と見なされます。

- **ユーザの機能:** SIPデバイスは、他のSIPデバイスをクエリし、そのデバイスが持つ機能を学習することができます。その後、両方のデバイスでサポートされている機能を使用して通信を行うことができます。
- **セッションの設定:** SIPをSession Description Protocolと組み合わせて使用し、2台のデバイス間に直接通信チャネルを確立することができます。この通信チャネルでは、通話の場合は音声データなどセッションデータが送信されます。このとき、SIPチャネルは、通話を切断するリクエストなどのコントロール情報を送信するために使用されます。通常、セッションデータはReal-Time Transport Protocol (RTP)を使用して送信されます。
- **セッションの管理:** SIPは、デバイス間のセッションの終了およびセッションの転送をコントロールします。確立されたセッションのパラメータを変更することもできます。

SIPIは、IPv4およびIPv6をサポートし、TCPおよびUDPで動作します。

Traffic ManagerとSIPプロトコル

Traffic Managerは、SIPプロキシサーバにSIPトラフィックを負荷分散し、可用性、信頼性、および拡張性に優れたSIPインフラストラクチャを構築します。下図に、Traffic Managerが含まれるSIPネットワークの例を示します。一般に、SIP電話はユーザの個々のロケーションに設置されますが、Traffic Manager、SIPプロキシサーバ、およびロケーションデータベースは、SIPサービスプロバイダが所有します。



Traffic Managerは、SIPトラフィックの処理に関して以下の機能を提供します。

- ・ カスタマイズ可能な負荷分散アルゴリズムを使用して、SIPプロキシサーバにSIPリクエストを負荷分散します。
- ・ セッション維持を維持するため、同じセッションに属する全てのリクエストが同じプロキシサーバに送信されます(デフォルトでは、Traffic ManagerはSIP「Call-ID」ヘッダに残ります)。
- ・ RuleBuilderおよびTrafficScriptルールを使用して、Traffic Managerを通過するSIPリクエストおよびレスポンスパケットを変更します。
- ・ 受信したリクエストをプロキシサーバに送信せずにレスポンスを送信します。
- ・ 使用中のプロキシサーバのヘルスをモニタリングし、障害発生時にはトラフィックを他のプロキシサーバにルーティングします。
- ・ 全てのクライアントがプライベートネットワーク内に存在する場合、全てのSIPデータおよびセッションデータのゲートウェイとして機能します。
- ・ DoS攻撃および悪意のあるデータから保護します。

プロキシサーバでトラフィック管理をサポートするための設定

Traffic Managerは、Oracle/Weblogic SIP ServerやSIP Express Router (SER)など一連のSIPプロキシサーバの前段階で使用するのが最も効果的です。この構成で動作させるには、プロキシサーバは、ドメインクライアントが利用するTraffic Managerを管理対象として認識する必要があります。例えば、sip:user@example.comがTraffic ManagerのIPに名前解決される場合、プロキシサーバはexample.comドメインを管理する必要があります。こうすることで、プロキシサーバは、リクエストをそのドメインに転送せずに処理ようになります。プロキシサーバがリクエストをこのドメインに転送してしまうと、Traffic Managerとプロキシサーバ間でリクエストがループすることになります。

または、プロキシサーバがドメインをサポートしていない場合は、管理UIの**Virtual Servers** (仮想サーバ) > **Edit** (編集) > **Protocol Settings** (プロトコル設定) ページに移動し、**rewrite_request_uri** オプションを有効化します。このオプションを有効化すると、受信する各リクエストのURIを、リクエストの送信先プロキシサーバのURIで置き換えるようTraffic Managerに指示が出されます。

Traffic ManagerでのSIPサービスの設定

新しいSIPサービスを作成するには、**Manage a new service** (新規サービスの管理) ウィザードを使用します。サービスの適切な名前を指定し、SIP over TCPまたはSIP over UDPプロトコルのいずれかを選択します。デフォルトのSIPポートは、TCPおよびUDPのどちらのサービスの場合も5060です。このウィザードにより、指定した詳細に基づいて新しいSIP仮想サーバとプールが設定されます。

SIP仮想サーバを正常に動作させるには、この仮想サーバが、SIPリクエストの送受信に使用される全てのIPアドレスを監視する必要があります。これを行うには、**Virtual Servers** (仮想サーバ) > **Edit** (編集) > **Basic Settings** (基本設定) ページに移動し、**Listening on** (監視) 設定を見つけます。「All IP addresses (全IPアドレス)」が選択されていることを確認します(これがデフォルトです)。

2台の仮想サーバを作成し、それぞれにUDPプロトコルとTCPプロトコルを割り当てることにより、同じポートで両方のプロトコルを使用することもできます。

注記

SIPインフラストラクチャ内でTraffic Managerを使用する場合、クライアントへのSIPリクエストの送信時に、SIPプロキシサーバへの登録時とは異なるコネクションが使用されます。そのため、SIPリクエストが電話まで通過できるようにクライアントのファイアウォールを設定する必要があります。一部のファイアウォールではこの設定が自動的に行われますが、その他のファイアウォールは、ポート5060のリクエストをユーザの電話に転送するよう設定する必要があります。

Traffic ManagerでのSIP動作モード

SIP仮想サーバは、「SIP Routing (SIPルーティング)」と、「SIP Gateway (SIPゲートウェイ)」、および「Full Gateway (フルゲートウェイ)」の3つの動作モードで実行できます。この設定は、コントロール情報(SIPメッセージ)およびセッションデータ(一般には音声およびビデオ通信のRTPデータ)にTraffic Managerがどの程度関与するかに影響します。

SIP動作モードを定義するには、Virtual Servers (仮想サーバ) > Edit (編集) > **Protocol Settings** (プロトコル設定) > **SIP-specific settings** (SIP固有設定) をクリックして、Traffic Manager管理インターフェースで仮想サーバのSIP設定を編集します。

▼ SIP-Specific Settings

How the virtual server handles general SIP traffic.

The virtual server should discard a SIP transaction when no further messages have been seen within this time.

sip_transaction_timeout: seconds

When timing out a SIP transaction, send a 'timed out' response to the client and, in the case of an INVITE transaction, a CANCEL request to the server.

sip_timeout_messages: ☒ Yes ☐ No

The mode that this SIP virtual server should operate in.

sip_mode: ☐ SIP Routing
☒ SIP Gateway
☐ Full Gateway ...

Replace the Request-URI of SIP requests with the address of the selected back-end node.

sip_rewrite_uri: ☐ Yes ☒ No

Should the virtual server follow routing information contained in SIP requests. If set to ☒ requests will be routed to the chosen back-end node regardless of their URI or Route header.

sip_follow_route: ☒ Yes ☐ No

The action to take when a SIP request with body data arrives that should be routed to an external IP.

sip_dangerous_requests: ☒ Send the request to a back-end node
☐ Send a 403 Forbidden response to the client
☐ Forward the request to its target URI (dangerous)

SIP clients can have several pending requests at one time. To protect the traffic manager against DoS attacks, this setting limits the amount of memory each client can use. When the limit is reached new requests will be sent a 413 response. If the value is set to 0 (zero) the memory limit is disabled.

sip_max_connection_mem: bytes

各動作モードについて以下に説明します。

- **SIPルーティング**(SIPルーティング): このオプションは、SIPプロキシサーバにSIPセッションを負荷分散する場合に選択します。最初のSIPリクエストとそれに対する全てのレスポンスはTraffic Managerを経由するため、RuleBuilderおよびTrafficScriptルールでチェックし、操作することができます。

Traffic Managerは、リクエストにRecord-Routeヘッダフィールドを追加しません。その結果、同じセッションに属する以降のリクエストはTraffic Managerを経由せず、最初のリクエストが送信されたサーバに直接送信されます。

すなわち、一般的なSIPセッションでは、INVITEリクエストとそれに対する全てのレスポンスはTraffic Managerを経由し、このセッションを管理するBYEリクエストなどの以降のリクエストはTraffic Managerを経由しません。

- **SIP Gateway** (SIPゲートウェイ): このオプションは、特定のセッションに属する全てのSIPリクエストをチェックする場合に選択します。Traffic Managerは、リクエストがTraffic Managerを経由するときにRecord-Routeヘッダフィールドを追加するため、同じセッションに属する以降のリクエストは、Traffic Managerを経由してルーティングされます。これらの全てのリクエストとそれに対するレスポンスは、RuleBuilderおよびTrafficScriptルールで操作することができます。

一般的なSIPセッションでは、INVITEやBYEなどの全てのコントロールメッセージがTraffic Managerを経由しますが、セッションデータ自体は経由しません。これは、Traffic ManagerでのSIP仮想サーバのデフォルト動作モードです。

- **Full Gateway** (フルゲートウェイ): このオプションは、全てのセッションデータにTraffic Managerを経由させる場合に選択します。このモードは、全てのSIPクライアントがTraffic Managerと同じネットワーク上に存在し、Traffic Managerがインターネットへのゲートウェイとして機能している場合に便利です。

このモードでは、INVITEやBYEなど全てのコントロールメッセージがTraffic Managerを経由します。Traffic Managerは、リクエストまたはレスポンスのボディデータに含まれるSession Description Protocol情報を変更し、セッションデータもTraffic Managerを経由するようにします。

セッションデータに使用するポート範囲を指定できます。また、セッションデータコネクションでデータが送信されないときに、Traffic Managerがこのコネクションを切断するまでの時間を指定することもできます。

SIPのその他の設定

Protocol Settings (プロトコル設定) ページにも、Traffic Managerの動作に影響を与える設定がいくつか表示されます。これには以下の設定があります。

設定	説明
sip_transaction_timeout	SIPリクエストとレスポンスはトランザクションにグループ化されます。SIPのトランザクションには、INVITEトランザクションと非INVITEトランザクションの2つのタイプがあります。INVITEトランザクションは、INVITEリクエストとそれに続く一連のレスポンスから成り、最後のレスポンスが「200 OK」レスポンスでない場合は、その後にACKリクエストが続きます。一方、非INVITEトランザクションは、リクエストとそれに続く0個以上のレスポンスから成ります。クライアントは同じコネクションで複数のトランザクションを送信できます。

設定	説明
	特定のトランザクションに属するメッセージが、sip_transaction_timeoutに指定された時間にわたって検出されないと、Traffic Managerはそのトランザクションに関連付けられたリソースを再利用します。 トランザクションが完了しないままタイムアウトになると、Traffic Managerは、408リクエストタイムアウトレスポンスをクライアントに送信します。
sip_rewrite_uri	このオプションを設定すると、Traffic Managerに対し、受信したリクエストのURIを、Traffic Managerが選択したバックエンドノードのURIに置き換えるよう指示が出されます。例えば、受信したリクエストのURIが「sip:bob@example.com」で、Traffic Managerがexample.comドメインを管理しているとしします。また、バックエンドノードのIPが192.0.2.1で、そのSIPサーバがポート5070で実行されている場合、バックエンドノードに送信されるリクエストはsip:bob@192.0.2.1:5070になります。 この設定は、SIPプロキシがユーザロケーションの検索時にユーザのドメインをチェックしない場合に便利です。
sip_follow_route	このオプションを設定すると、Traffic Managerに対し、受信したリクエストに含まれるルーティング情報に従うよう指示が出されます。Routeヘッダが存在する場合、リクエストはそれに相当する宛先に送信され、Routeヘッダが存在しない場合は、URIで指定された宛先に送信されます。URIがTraffic Managerを指している場合、リクエストは通常どおりプールに送信されます。 この動作は、sip_dangerous_requests設定でオーバーライドできます。例えば、受信したリクエストにボディデータとRouteヘッダが含まれる場合、危険なリクエストをバックエンドノードに送信するようsip_dangerous_requestsが設定されていれば、Routeは無視され、リクエストはプールに送信されます。 sip_follow_routeをNoに設定すると、全てのリクエストがプールに送信され、選択したノードに対応するRouteヘッダが追加されます。

設定	説明
sip_dangerous_requests	SIPリクエストは、それぞれ独自のルーティング情報を持ちますが、この情報が、中間マシンを介してリモートコンピュータを攻撃するために悪用される可能性があります。Traffic Managerは、次の宛先が外部IPで、リクエストにボディデータが含まれている場合、そのリクエストを危険なものと見なします。 Traffic Managerは、危険性が疑われるリクエストを以下の方法で処理することができます。 リクエストをバックエンドノードに送信できます。このオプションは、SIPプロキシサーバが、Traffic ManagerのIPに名前解決されないドメインを管理している場合に便利です。これは、危険性が疑われるリクエストに対するデフォルト設定です。 403 Forbiddenレスポンスをクライアントに返すことができます。このオプションは、Traffic Managerのドメインにアドレス指定されていない全てのリクエストを拒否する場合に便利です。 リクエストをそのターゲットURIに転送できます。このオプションを選択すると、SIPリクエストは通常どおりにルーティングされます。この設定は、外部に送信されるSIPトラフィックを仮想サーバで処理する場合に使用します。
sip_max_connection_mem	クライアントは、同じコネクションで複数のSIPリクエストを送信し、それに対するレスポンスを任意の順序で受信することができます。クライアントでレスポンス待ちのアクティブなリクエスト数が過剰にならないようにするために、Traffic Managerは、各コネクションに割り当てることのできる最大メモリ量を制限できます。 この設定では、Traffic Managerが各コネクションに許容するメモリ量を指定します。このメモリ量を超えると、新しいリクエストは「413 Request Entity Too Large」レスポンスで拒否されます。この値を0に設定すると、メモリ制限が解除されますが、Traffic ManagerがDoS攻撃を受けやすくなります。

UDPベースのSIPサービスでは、「UDP-Specific Settings」(UDP固有の設定)にある以下の設定も使用されます。

設定	説明
sip_udp_associate_by_source	SIP UDPトランザクション内でクライアントから送信された全てのデータグラムを、同じIPアドレスとポートから送信する必要があるかどうか指定します。 デフォルトでは、この設定は有効化されています。無効にすると、データグラムが属するトランザクションの識別時に、データグラムの送信元のIPアドレスがTraffic Managerで考慮されません。 Traffic Managerがレスポンスのデータグラムを送信するクライアントのIPアドレスは、トランザクション内の最初のデータグラムが受信されたIPアドレスと異なるIPアドレスからデータグラムが受信されても、未変更のままとなります。

UDPベースのSIPサーバとの通信

一般に、UDPベースのSIPサーバは、TCPベースのSIPサーバと同様にリクエストに対してレスポンスを返します。すなわち、元のリクエストを受信したポートからレスポンスを送信します。この動作が必ずしも必要なわけではありませんが、SIPサーバには、リクエストを受信したものとは異なるポートやIPからレスポンスを返すものがあります。

Traffic Managerでは、**Pools (プール) > Edit (編集) > Protocol Settings (プロトコル設定)** ページで「**udp_accept_from**」を設定することにより、この動作を確実に行うことができます (UDPサービスでプールが使用される場合)。この設定を使用して、Traffic ManagerがUDPリクエストに対するレスポンスをどのように受信するかを以下のオプションから選択できます。

- リクエストの送信先のIPアドレスとポートのみ。
- リクエストの送信先のIPアドレスのみ。ただし任意のポートを使用可能。
- 特定のIPアドレスセット。ただし任意のポートを使用可能。
- 任意のIPアドレスと任意のポート。

特定のIPアドレスセットからのみレスポンスを受け入れるよう選択した場合、表示されるボックスにCIDRマスク (198.51.0.0/16など)を入力します。

これらの設定では、セキュリティと互換性との間に様々なレベルのトレードオフが生じます。複数のIPアドレスからのレスポンスを許容すると、不正なレスポンスを受け入れる可能性が高くなり、セキュリティリスクが増大する可能性があります。特定のSIPでは、適切に通信するためにこの機能が必要になる場合があります。必要条件に最も適したオプションを選択してください。

関連する設定がSIP (over UDP)ヘルスマニタにあります。詳細については、[内蔵ヘルスマニタを参照してください](#)。

クライアントからUDPリクエストを受信した後にサーバノードに許可される最大時間を制限するには、**udp_response_time**を必要な秒数に設定します。値を「0」(デフォルト)に設定すると、タイムアウトカウンタが無効化されます。つまり、レスポンスを送信しないサーバは、障害が発生したと自動的にマークされます。

注記

IPv4アドレスに送信されたリクエストでは、IPv4アドレスからのみレスポンスが返されることが想定されます。一方、IPv6アドレスに送信されたリクエストでは、IPv4またはIPv6形式でのレスポンスが許容されます。

プロキシプロトコル

プロキシプロトコルとは、ネットワーク内のプロキシとその他のトラフィックルーティング/処理デバイス間で、TCPコネクションの送信元と宛先の詳細情報を通信するために使用されるコネクションレベルのプロトコルです。

プロキシプロトコルに対応したクライアントとサーバでは、リクエストの「PROXY」ヘッダに、送信元と宛先のIPアドレスとポートが追加されます。このヘッダは、そのライフサイクル中にTraffic Managerなどのプロキシによってコネクションに加えられた変更にかかわらず、変化しません。

Traffic Managerは、クライアント側のコネクションのみで、プレーンテキストヘッダを使用した仮想サーバのプロキシプロトコルバージョン1の使用および処理をサポートしています。



TrafficScriptを使用して、サーバ側のコネクションにPROXYヘッダを追加できます。ただし、このようなコネクション上では、Traffic Managerによっていかなる処理も実行されません。

仮想サーバでプロキシプロトコルのサポートを有効にするには、**Services** (サービス) > **Virtual Servers** (仮想サーバ) > **Edit** (編集) > **Protocol Settings** (プロトコル設定) > **TCP Connection Settings** (TCPコネクション設定)をクリックし、**proxy_protocol**を「Yes (はい)」に設定します。

PROXYヘッダの解析は、コネクションの設定に基づいて実行されるため、ヘッダの内容は、リクエストルールのTrafficScript(および後の段階)を通じて利用できます。該当するTrafficScript関数については、『*Pulse Secure Virtual Traffic Manager: TrafficScript Guide*』を参照してください。

プロキシプロトコル対応の仮想サーバによって処理される全てのコネクションには、有効なPROXYヘッダが必要です。このヘッダが存在しないか、形式に誤りがある場合、コネクションは破棄されます。または、仮想サーバでクライアントコネクション障害のロギングが有効になっている場合は、Traffic Managerによってイベントログにエラーがレポートされます。

リクエストログにPROXYヘッダデータ項目を記録するか、管理UIの**Activity** (アクティビティ) > **Connections** (コネクション)ページで、PROXYヘッダの内容を確認することもできます。

セッション維持

本章では、Traffic Managerにより、クライアントとバックエンドサーバ間のセッション維持(スティッキーセッションとも言う)がどのように提供されるかについて説明します。

セッション維持とは

クライアントから送信されるリクエストの多くのクラスは、バックエンドサーバのプールで負荷分散することができます。これにより、同じクライアントから送信された複数のリクエストを、サービスを中断することなくバックエンドサーバに分配できます。ただし、ローカルに保存されているクライアントの情報に依存するサーバアプリケーションなどは例外で、この方法では容易に負荷分散できない場合があります。これには以下のようなアプリケーションがあります。

- ファイルをディスクに保存してユーザのログインをトラッキングするWebメールアプリケーション
- クライアントの買い物かごの中身をディスクまたはメモリに保存する買い物かご
- 通常より起動時間が長いプログラムやプロトコル(Javaアプリケーション、SSLコネクションなど)

このような場合、Traffic Managerはクライアントとのセッションの間、以下のようにリクエストごとに同じバックエンドサーバにマッピングすることができます。

Traffic Managerは新しいコネクションを受け取ると、負荷分散ロジックを使用してそのコネクション用のノードを選択します。その後、選択したノードをセッション維持マップに記録します。

Traffic Managerは、同じセッションで別のコネクションを受け取ると、前に選択したノードを使用します。

このように、同じ「セッション」での全てのコネクションが同じバックエンドノードに固定されます。セッションの識別方法は、使用するセッション維持クラスによって決まり、複数のTrafficScriptルールを使用することでその決定を微調整することが可能です。

セッション維持クラスを使用すると、特定クライアントセッションの全てのリクエストを同じノードに送信することができます。これは、1つのアプリケーションセッションが多数のコネクションにわたるような複雑なアプリケーションで必要になる場合があります。例えば、Webベースの買い物かごや多くの複雑なUDPベースのプロトコルがこれに該当します。

セッション維持は、必要な場合のみ使用してください。セッション維持を使用すると、実質的に新しいセッションを除く全てのセッションで負荷分散プロセスが回避されます。同じクライアントからの全てのリクエストを同じバックエンドノードに送信する必要がある場合を除き、セッション維持を使用しない方がレスポンス時間は短くなります。

セッション維持の設定

セッション維持は、カタログを使用して設定します。カタログに追加したセッション維持クラスは、プールに割り当てたり、TrafficScriptルールで選択したりできます。

TrafficScriptを使用しない場合、セッション維持はプール単位で有効化します。複数のプールを定義し、必要なプールでのみセッション維持を有効化することができます。同じバックエンドサーバが複数のプールに存在することに注意してください。

複数のプールで同じセッション維持クラスを参照し、セッション維持マッピングを共有することができます。例えば、いくつかのHTTPノードが含まれるプール(ポート80)と、HTTPSサービスを実行する同じノードが含まれるプール(ポート443)は、ユーザとサービス間のセッション維持マッピングを共有できます。

TrafficScriptを使用する場合、TrafficScriptコマンドでセッション維持をオーバーライドすることができます。このコマンドが使用されていない場合は、Traffic Managerは、リクエストを処理するプールに指定されたセッション維持クラスを使用します。

セッション維持の有効化

セッション維持クラスを表示および編集するには、**Catalogs** (カタログ) > **Persistence** (維持)をクリックします。新しいクラスタイプを作成するには、「Create new Session Persistence class (新規セッション維持クラス作成)」ラベルの付いたボックスに名前を入力し、**Create Class** (クラス作成)をクリックします。

これで管理UIに、新しいセッション維持クラスの設定の概要が表示されます。

Name:

The type of session persistence to use.

type:

- ☒ **IP-based persistence ...**
Send all requests from the same source address or subnet to the same node.
If the subnet prefix length is 0, requests from the same IPv4 or IPv6 source address will be sent to the same node.
If the subnet prefix length is specified, requests from the same IPv4 or IPv6 subnet, based on that prefix length, will be sent to the same node.
IPv4 subnet prefix length:
IPv6 subnet prefix length:
- ☐ **Universal session persistence**
Use session persistence data supplied by a TrafficScript rule.
- ☐ **Named Node session persistence**
Use a node specified by a TrafficScript rule.
- ☐ **Transparent session affinity**
Insert cookies into the response to track sessions.
- ☐ **Monitor application cookies ...**
Monitor a specified application cookie to identify sessions.
- ☐ **J2EE session persistence**
Monitor Java's JSESSIONID cookie and URLs
- ☐ **ASP and ASP.NET session persistence**
Monitor ASP session cookies and ASP.NET session cookies and cookieless URLs.
- ☐ **X-Zeus-Backend cookies**
Inspect an application cookie named 'X-Zeus-Backend' which names the destination node.
- ☐ **SSL Session ID persistence**
Use the SSL Session ID to identify sessions (SSL pass-through only).

The action the pool should take if the session data is invalid or it cannot contact the node specified by the session.

failuremode:

- ☒ Choose a new node to use
- ☐ Redirect the user to a given URL ...
- ☐ Close the connection (using error_file on Virtual Servers > Edit > Protocol Settings)

Whether or not the session should be deleted when a session failure occurs. (Note, setting a failure mode of 'choose a new node' implicitly deletes the session.)

delete: ☒ Yes ☐ No

A description of the session persistence class.

note:

セッション維持手法の選択

デフォルトでは、新しいセッション維持クラスは、「IP-based Persistence (IPベースセッション維持)」が選択された状態で設定されます。この他、必要に応じて、以下のセッション維持手法を選択できます。

手法	対応プロトコル	説明
IP-based persistence (IPベースセッション維持)	全て	同じ送信元IPアドレス(またはオプションのサブネットの範囲)からの全てのリクエストを同じノードに送信します。
Universal session persistence (ユニバーサルセッション維持)	全て	このクラスでは、TrafficScriptルールにユニークなセッション維持キーを設定することにより、リクエスト内の任意の情報に基づいてセッションを維持することができます。

手法	対応プロトコル	説明
Named Node persistence (ネームドノードセッション維持)	全て	この手法では、TrafficScriptルールによりコネクションを特定ノードに送信することができます。
Transparent session affinity (トランスペアレントセッションアフィニティ)	HTTP、HTTPS	セッションをトラッキングするために、レスポンスにクッキーを挿入します。
Monitor application cookies (アプリケーションクッキーのモニタリング)	HTTP、HTTPS	セッションを識別するために、指定されたアプリケーションクッキーをモニタリングします。
J2EE	HTTP、HTTPS	JavaのJSESSIONIDクッキーとそのURL書き換えフォールバックをモニタリングします。
ASP and ASP.net session persistence (ASPおよびASP.netセッション維持)	HTTP、HTTPS	この手法では、セッション維持を使用して両方のタイプのASPセッションを処理することができます。
X-Zeus-Backend cookies (X-Zeus-Backendクッキー)	HTTP、HTTPS	宛先ノードを設定する「X-Zeus-Backend」というアプリケーションクッキーをチェックします。
SSL Session ID persistence (SSLセッションIDによる維持)	SSLプロトコル	SSLセッションIDでセッションを識別します。

IPベースセッション維持

IP-based persistence (IPベースセッション維持)を選択すると、Traffic Managerは、このプールに送信された各リクエストの送信元IPアドレスをトラッキングし、格納します。同じIPアドレスから追加のトラフィックを受信した場合、Traffic Managerは、格納されたセッションマップ情報を使用して、以前に使用したバックエンドサーバノードにリクエストを送信します。

同じIPサブネットからのリクエストが確実に同じノードに送信されるように、オプションとしてIPv4またはIPv6のサブネットプレフィックス長を指定できます。これは、Traffic Managerクラスタによって維持管理されるIPセッションマップのサイズを制限するために役立つことがあります。

セッション中にIPアドレスを変更する特殊なモバイルクライアントや、IPを共有するクライアントがあるため(プロキシサーバを経由する場合など)、環境によってはこの維持タイプが適さない場合もあります。

ただし、IPベースセッション維持は完全に透過的でプロトコルに依存しないため、複雑なプロトコルやアプリケーションでのセッション維持コネクションの確保に役立ちます。

IPセッションマップは、クラスタ内の全てのTraffic Managerインスタンスで共有されます。そのため、リクエストが別のTraffic Managerで受信された場合も適切なノードに送信されます。また、特定のTraffic Managerで障害が発生した場合も、そのTraffic Managerが管理していたIPセッションマップは他のTraffic Managerにより認識されます。

ユニバーサルセッション維持

ユニバーサルセッション維持では、TrafficScriptルールで定義されたセッション維持キーを使用して、同じセッション維持キーが設定された全ての接続を同じバックエンドノードに送信しようとします。

セッション維持キーは、ルールでTrafficScript関数の`connection.setPersistenceKey()`によって指定されます。これにより、リクエスト内の任意の情報に基づくセッション維持が可能になります。例については、[ユニバーサルPHPセッション維持](#)を参照してください。

ユニバーサルセッション維持マップは、クラスタ内の全てのTraffic Managerマシンで共有されます。そのため、リクエストが別のTraffic Managerで受信された場合も適切なノードに送信されます。また、特定のTraffic Managerで障害が発生した場合も、そのTraffic Managerが管理していたユニバーサルセッションマップは他のTraffic Managerにより認識されます。

注記

ユニバーサルセッション維持をSIP over UDPで使用する場合、セッション維持は、クライアントから送信された最初のリクエストにのみ適用されることに注意してください。同じクライアントが送信する以降の全てのリクエストは、別のセッション維持キーが割り当てられていても、最初のリクエストと同じノードに送信されます。そのため、ユニバーサルセッション維持は、同じクライアントからの全てのリクエストを同じサーバに送信する場合のみSIPで使用してください。

ネームドノードセッション維持

ネームドノードセッション維持を使用すると、接続をプール内の特定ノードに送信できます。この手法では、リクエストのルーティング方法を非常にきめ細かくコントロールすることができます。

TrafficScriptルールで`connection.setPersistenceNode()`関数を使用することにより、セッション維持の決定時にどのノードを使用するかを指定できます。例えば、ルールでは、リモートデスクトップサービスへの新しい接続を受け取ると、外部データベースをクエリして、ユーザのデスクトップをホストするノードを決定します。次に、`connection.setPersistenceNode()`関数を使用して、接続をそのノードに送信することができます。

トランスペアレントセッションアフィニティ

トランスペアレントセッションアフィニティでは、セッションをトラッキングするためにクッキーをHTTPレスポンスに挿入します。この手法は、ノードがレスポンスにクッキーを設定する必要がないため、一般にHTTPトラフィックやSSL復号化HTTPストラフィックに最適です。

トランスペアレントセッションのマップは、「X-Mapping-xxxxxx」というクライアント側のクッキーに保存されます。ここで、「xxxxxx」は固有のIDです。クラスタ内の全てのTraffic Managerは、このセッションクッキーの値をチェックし、セッションを同じサーバノードに送信します。

デフォルトでは、セッションクッキーが挿入されるのは、対応するリクエストに既に一致するクッキーが含まれない場合のみです。Traffic Managerが、クライアントに送信された全てのレスポンスにセッションクッキーを挿入するよう強制するには、「Set the cookie in every response (全てのレスポンスでクッキーを設定)」を有効化します。セッションクッキーをクライアントへの全てのレスポンスに含めることで、TrafficScriptのレスポンスルールでセッションクッキーやクッキーが持つ指示を確認または修正できます。

Traffic Managerがセッションクッキーに含めるはずのHTTPのクッキーの指示を設定するには、「Directives to include in the cookie (クッキーに含める指示)」設定を使用します。例えば、指示「Secure」を追加すると、ほとんどのクライアントは、セッションクッキーを安全な(HTTPS)接続で送信されたリクエストのみに含めます。これにより、「X-Mapping-」セッションクッキーを、同じ宛先への安全ではない接続のコンテンツを見る可能性がある、第三者に表示しないようにできます。

複数の指示を追加するには、セミコロンを区切り文字として使用します。例えば、「Secure」と「Max-Age」指示を使用する場合、指示リストを「Secure; Max-Age=3600」として指定します。

アプリケーションクッキーのモニタリング

アプリケーションクッキーによるセッション維持では、ノードから送信されたHTTPレスポンス内の特定のクッキーをモニタリングします。例えば、PHPアプリケーションでは、クラスタ化されたPHPアプリケーションがセッションの状態をトラッキングしてインデックス化できるよう「PHPSESSID」というセッションクッキーが生成されます。このセッション維持手法では、特定のアプリケーションクッキーが保持されているリクエストを同じサーバノードに送信します。モニタリングするアプリケーションクッキーの名前を指定する必要があります。

アプリケーションクッキーセッションマップは、クライアント側の「K-CookieName-xxxxxx」クッキーに保存されます。ここで、「CookieName」はモニタリング対象のクッキー、「xxxxxx」はセッション維持クラスを識別する内部文字列で、このクッキーの値は適切なノードを識別するための内部文字列になります。クラスタ内の全てのTraffic Managerマシンは、このセッションクッキーの値をチェックし、セッションを同じサーバノードに送信します。

バックエンドサーバがアプリケーションクッキーの値を変更しても、セッションは有効で、クライアントは引き続き同じサーバノードに送信されます。

J2EE JSESSIONIDクッキー/URL

J2EE JSESSIONIDクッキー/URLセッション維持では、JSESSIONIDクッキー(アプリケーションクッキーセッション維持と同様)とjsessionidパスパラメータの両方をモニタリングします。これらは、Java Extension仕様(v2.4)で定義されており、BEA WebLogic、IBM WebSphere Application Server、JBoss/TomcatなどのJava Extensionコンテナで使用されています。

J2EEセッションマップは、クラスタ内の全てのTraffic Managerで共有されるため、リクエストが別のTraffic Managerで受信された場合も同じセッションが適用されます。

ASP.netセッション維持

ASP (Active Server Pages)は、動的に生成されるWebページを処理するためにMicrosoftが開発したサーバー側スクリプティングプロトコルです。サーバークラスタでASPセッション管理を使用するには、セッションの存続期間にわたって特定ユーザから受信した全てのリクエストを同じWebサーバで処理する必要があります。

ASPセッションでは、ブラウザでクッキーがサポートされているかどうか、ユーザが自分の意思でクッキーを無効化しているかどうかなど様々な要因に応じて、クッキーを使用することも使用しないこともできます。

- **クッキーベースのセッション:** Traffic ManagerのASPセッション維持クラスは、クッキーを検出し、それに基づいてクライアントのセッションを識別します。
- **クッキーレスセッション:** Traffic ManagerのASPセッション維持クラスは、ASPアプリケーションが生成したURLに埋め込まれたASP識別子を検出して使用します。

ASPセッションマップは、クラスタ内の全てのTraffic Managerで共有されるため、リクエストが別のTraffic Managerで受信された場合も同じセッションが適用されます。

単一のホスト名で複数のASPアプリケーションをホストしている場合、各アプリケーションによって独自のクッキーが生成されることに注意してください。そのため、アプリケーションごとに個別にクッキーとノード間のマッピングを管理するためには、各アプリケーションに別々のASP.Netセッション維持クラスが必要になります。例えば、リクエストURLをチェックしてどのアプリケーションにアクセスするかを決定し、TrafficScript関数のconnection.setPersistence()を使用してセッション維持クラスを選択することができます。

X-Zeus-Backendクッキー

X-Zeus-Backendクッキーセッション維持では、各アプリケーションリクエスト内でX-Zeus-Backendというクッキーを探します。このクッキーが存在し、現在のプールに含まれるノードの名前が保持されている場合、リクエストはそのノードに送信されます。このクッキーは、バックエンドサーバまたはTrafficScriptルールによって挿入できます。



このセッション維持手法は廃止される予定です。Zeus Load Balancerセッション維持クッキーとの後方互換性を維持する目的でのみ提供されているものです。

SSLセッションIDセッション維持(SSLパススルーのみ)

SSLセッションIDセッション維持手法では、同じSSLセッションIDの全てのSSLトラフィックを同じサーバノードに送信します。SSLパススルートラフィックにのみ使用でき、SSL復号化トラフィックには使用できません。



TLSv1.3プロトコルを使用した接続はセッション回復にセッションIDを使用しないため、そのケースではこの方法が適用されません。

SSLハンドシェイクには、多くのCPU時間、ネットワーク帯域、および待ち時間が伴います。SSLセッションIDセッション維持手法を使用すると、再開された接続がセッションを作成(および保存)したノードに向けることで、ノードが実行するSSLハンドシェイク処理の回数が減少します。

SSLセッションIDセッションマップは、クラスタ内の全てのTraffic Managerマシンで共有されます。そのため、リクエストが別のTraffic Managerで受信された場合も適切なノードに送信されます。また、特定のTraffic Managerで障害が発生した場合も、そのTraffic Managerが管理していたSSLセッションIDのセッションマップは他のTraffic Managerにより認識されます。

多くのSSLクライアントは定期的にSSLセッションIDについてのネゴシエーションを行うため、SSLセッションIDセッション維持は、アプリケーションレベルのセッション維持には適していません。アプリケーションレベルのセッション維持を行うには、IPベースセッション維持を使用するか、トラフィックを復号化してユニバーサルセッション維持または適切であればHTTP手法を使用する必要があります。

ノードへのセッション維持マップの名前解決

Traffic Managerは、新しいコネクションを受け取ったときにセッション維持情報がない場合、負荷分散ロジックを使用してそのコネクションに使用するノードを選択します。選択されたノード(IPアドレスとポート)は、内部クッキー(IP、ユニバーサル、またはSSLセッション維持の場合)または外部クッキー(トランスペアレント、アプリケーションクッキー、またはX-Zeus-Backendクッキーの場合)のいずれかのセッション維持マッピングに記録されます。

Traffic Managerは、同じセッションで別のコネクションを受け取った場合に、そのコネクションをプール内のノードに転送できる状態にあれば、使用中のセッション維持クラスをチェックし、特定のノードを使用するかどうかを決定します。

コネクション用のプールに同じIPアドレスとポートのノードが存在する場合、Traffic Managerはコネクションをそのノードに送信します。

正確に一致するものがない場合、Traffic Managerはプール内で同じIPアドレスの(ただしポートが異なる)ノードを検索します。このようなノードが1つのみ存在する場合、Traffic Managerはコネクションをそのノードに送信します。

これにより、ノードタイプの異なる複数のプール間でセッション維持情報を共有することができます。

例えば、Webベースのアプリケーションにおいて、買い物かごの中身の管理にはHTTPインターフェースを使用し、支払いの管理にはセキュアHTTPSインターフェースを使用する場合があります。このとき、セッション維持の必要条件によっては、ユーザのHTTPトラフィックとHTTPSTraフィックの両方を同じ物理マシンに送信しなければならない場合があります。

両方のサービス(HTTPおよびHTTPS)で同じセッション維持クラスを参照することもできます。例えば、ユーザが最初にHTTPサービスに接続したときに、Traffic ManagerはHTTPプールを使用し、そのプール内の特定のノード(IPアドレス、ポート80)とのセッションが確立されます。

その後、ユーザがHTTPSサービスにアクセスしたときに、Traffic Managerが、HTTPSノードが含まれる別のプールを使用することがあります。この場合、セッション維持クラスにより、ユーザのリクエストはHTTPSプール内の同じ物理マシン(IPアドレス、ポート443)に送信されます。

ノード障害発生時のオプション

セッション維持マッピングで必要とされるノードが使用できなくなる場合があります。例えば、モニタによって「障害発生」のマークが付けられたり、Traffic Managerが接続できなくなった場合などです。

このような場合に備え、Traffic Managerには、そのノードを現在使用しているセッションに対する代替アクションが用意されています。**failuremode**設定を使用して、以下のアクションの1つを選択します。

アクション	説明
Choose a new node to use (使用する新しいノードの選択)	プールはセッションマップを破棄し、現在の負荷分散アルゴリズムを使用して新しいノードを選択します。
Redirect the user to a given URL... (指定されたURLにユーザをリダイレクト)	プールは、リクエストに対するレスポンスとして、設定されているロケーション(URL 設定)にHTTP 302リダイレクトを送信します。そのリダイレクトロケーションにあるリソースでは、メッセージを表示するか、ユーザに再度ログインして新しいセッションを確立するよう促すことができます。
Close the connection (コネクション終了)	プールは即座にコネクションを終了します。

「Redirect the user to a given URL... (指定されたURLにユーザをリダイレクト)」または「Close the connection (コネクション終了)」を選択した場合は、**delete** (削除)設定を使用して、セッションマップを破棄するか、保持するかを選択します。**delete** (削除)を「No (いいえ)」(セッションマップを保持)に設定した場合、クライアントが元に戻り、セッションのノードが再び使用可能になると、Traffic Managerがリクエストを同じノードに送信します。

「Choose a new node to use (使用する新しいノードの選択)」を選択した場合は、Traffic Managerが暗黙的にセッションマップを削除します。

コネクションのドレイン

例えばソフトウェアのアップグレード、ハードウェアのメンテナンス、撤去や再利用のために、バックエンドサーバをサービスから除外することが必要になる場合があります。

プールの「コネクションのドレイン」機能は、この作業を容易に行えるよう設計されています。ノードに「ドレイン」のマークを付けると、Traffic Managerはそのノードに新しいコネクションを送信しなくなります。ただし、そのノードに対して既に確立されているセッションでのコネクションは引き続きそのノードに送信されます。

これにより、継続中のコネクションや長時間確立されているセッションを中断することなく、ノードをプールから安全に除外することができます。管理UIの**Activity** (アクティビティ)セクションを使用すると、ドレイン中のノードがアイドル状態にある時間の詳細を取得してから、全てのセッションが完了したかどうかを判断することができます。

例えば、eコマースサービスを運営し、セッション維持によって個々のセッションを特定のバックエンドノードに紐付けているとします。ここで、ノードの1つにドレインのマークを付けると、以下ようになります。

そのノードとの間で新しいセッションは確立されません。

確立済みの既存のセッションでは、引き続きそのノードを使用できます。

60分後、管理UIの**Activity** (アクティビティ) > **Draining Nodes** (ノードのドレイン)ページを調べます。ノードは35分間アイドル状態のままです。このことから、確立されている全てのセッションが完了していると判断し、ノードを安全に除外することができます。

実際には、ノードがドレインを完了したかどうかを判断するための時間は、アプリケーションとユーザの動作によって大きく異なってきます。システムの設定時に、管理者は、目的とする用途を考慮してこれらの値を決定する必要があります。

コネクションのドレインの設定方法については、[ノードのドレイン](#)、[無効化](#)、[削除](#)を参照してください。

セッション維持キャッシュの設定

セッション維持手法には、クライアント側クッキーを使用してセッション維持データを保存するものと、クラスター内で自動的に共有されるTraffic Managerのキャッシュを使用してセッション維持マッピングを保存するものがあります。

キャッシュを使用する維持手法ごとに、各キャッシュが保持できるマップエントリの最大数を設定できます。また、特定の手法に対して、各セッションマップのエントリの有効期限を設定できます。

キャッシュされたセッションマップに有効期限を設定すると、新たなノードが追加された場合やノードが障害から復旧した場合などに負荷をプールノード間で再分配するために役立ちます。特に、クライアントのセットが固定された企業環境がそれに該当します。制限された期間でセッションキャッシュを設定することは、ノードをドレイン状態にする場合にも役立ちます。詳細については、[ノードのドレイン](#)、[無効化](#)、[削除](#)を参照してください。

定義済みの有効期限になった際に、Traffic Managerがアクティブにキャッシュエントリを削除することはありません。代わりに、エントリが参照された場合のみ、キャッシュエントリ時間が有効期限値と比較チェックされます。セッションがまだ実行中の場合、Traffic Managerはセッションを返しそのセッションマッピングのキャッシュエントリ時間をリセットします。セッションが期限切れであると見なされた場合、Traffic Managerはキャッシュからそれを削除し新たなセッションを確立します。

 Traffic Managerが期限切れセッションエントリを削除するよりも早くキャッシュがいっぱいになった場合は、新たなエントリが追加された際に、Traffic Managerが最も古い(使用日時が最古のもの)エントリを破棄します。

キャッシュの保持手法を設定するには、**System** (システム) > **Global Settings** (グローバル設定) > **Cache Settings** (キャッシュ設定)をクリックします。以下の表は、維持クラスのタイプ、対応するキャッシュのサイズ設定、および(該当する場合)対応する有効期限設定を示しています。

維持タイプ	キャッシュサイズ設定 (エントリ数)	キャッシュの有効期限設定 (秒単位の期間)
IP session persistence (IPセッション維持)	ip_cache_size	ip_cache_expiry
Universal session persistence (ユニバーサルセッション維持)	universal_cache_size	universal_cache_expiry
SSL session-id session persistence (SSLセッションIDセッション維持)	ssl_cache_size	(対象外)
J2EE session persistence (J2EEセッション維持)	j2ee_cache_size	j2ee_cache_expiry
ASP session persistence (ASPセッション維持)	asp_cache_size	(対象外)

注記

キャッシュのサイズと有効期限設定はグローバルに適用されます。設定した値は、特定の維持手法に作成した全てのセッション維持クラスのインスタンスへ平等に適用されます。

キャッシュの動作をモニタリングするには、アクティビティモニタでセッション維持キャッシュデータ値をプロットします。まず、**Activity** (アクティビティ)をクリックしてアクティビティモニタを使用します。次に、**Change Data** (データ変更)をクリックして、「Values to monitor (モニタリングする値)」ツリーで**Cache Values** (キャッシュ値)」を展開します。このセクションには、維持キャッシュタイプのサブツリーが含まれます。リストからモニタリングする維持タイプを選択/展開し、グラフに追加する関連データアイテムのチェックボックスをオンにします。選択した値をプロットするには、**Apply** (適用)をクリックします。

キャッシュの適切なモニタリングおよびサイジングに役立つ主な値を、以下の表に示します。

設定	説明
Entries (エントリ)	キャッシュ内にあるエントリの数
EntriesMax (エントリ最大値)	設定されているキャッシュの最大サイズ
Oldest (最も古いエントリ)	使用されていない期間が最も長いエントリの最終使用時点からの経過時間

Traffic Managerが多数のセッションを処理すると、通常、キャッシュはいっぱいになります。**Oldest** (最も古い)値は、未使用または有効期限切れの状態でログエントリがキャッシュで保持される長さを示します。

キャッシュのサイジングを適切に行うには、新しいエントリがキャッシュに追加される頻度(1秒あたりのエントリ数)と、エントリが最後に使用されてから破棄されるまでの時間(秒単位)を考慮する必要があります。これらの2つの値を乗じて必要なキャッシュサイズを見積ります。また、**Oldest** (最も古いエントリ)値をモニタリングして、キャッシュがいっぱいになってエントリが破棄されるのが早過ぎないかを確認します。

セッション維持レコードをきめ細かくコントロールする必要がある場合、最も効果的な方法は、特定の有効期限が設定されたクライアント側クッキーを使用し、セッション維持をクッキーの存在に紐付けることです。

セッション維持とマルチホストトラフィックIPアドレスの併用

「consider source port (ソースポート考慮)」設定を有効にしてマルチホストトラフィックIPアドレスを使用する場合、特定の送信元IPアドレス(すなわち特定のクライアント)からのリクエストは、クラスタ内の全てのTraffic Managerで処理されます([トラフィックIPグループの作成](#)を参照)。

この状況では、状態の共有(Traffic Manager間でセッションマップを共有)に依存するセッション維持手法が確実に機能しない場合があります。セッション情報がクラスタ全体に伝搬されるまでに短い待ち時間が生じ、その間にクライアントが複数のTraffic Managerにアクセスすると、クライアントのセッション維持マッピングが壊れる可能性があるからです。

状態の共有に依存するセッション維持アルゴリズムには以下のものがあります。

- IPベースセッション維持
- ユニバーサルセッション維持
- SSLセッションIDセッション維持
- J2EEおよびASP.NETセッション維持

上記のセッション維持手法を使用するサービスで使用されるマルチホストトラフィックIPでは、**consider source port** (ソースポート考慮)設定を使用しないでください。

UDPプロトコルでのセッション維持

UDPプロトコルはコネクション指向ではありませんが、同じセッションのUDPデータグラムを同じバックエンドサーバにルーティングするのが望ましい場合がよくあります。

UDPセッションをどのように扱うかをTraffic Managerに通知するには、**Virtual Server** (仮想サーバ) > **Protocol Settings** (プロトコル設定)ページにある2つの設定**udp_timeout**および**udp_response_datagrams_expected**を使用します。

UDPセッションは、設定**udp_response_datagrams_expected**に等しい数のレスポンスデータグラムが検出されるまで存続します。例えば、サーバが1つのデータグラムをクライアントに送信したらセッションを完了する場合は、**udp_response_datagrams_expected**を「1」に設定します。

セッション長時間存続させる場合は、**udp_response_datagrams_expected**を「-1」に設定します。UDPセッションは、**udp_timeout**秒間に渡ってUDPTraフィックが確認されない場合にタイムアウトになります。

長時間に渡ってセッションをトラッキングする場合、UDPで様々なタイプのセッション維持(一般にはIPベースまたはユニバーサル)を使用できます。

例

ユニバーサルPHPセッション維持

TrafficScriptルールでは、受信したHTTPリクエストをチェックし、リクエストに基づいてバックエンドノードを選択することができます。

[コンテンツタイプによるルーティング](#)には、ファイル拡張子に基づいてプールを選択するTrafficScriptルールの例が示されています。このルールは、選択したセッション維持クラスにセッション維持データを渡すよう変更することができます。

PHPというセッション維持クラスを作成し、ユニバーサルセッション維持([セッション維持の設定](#)を参照)を使用するよう設定します。次に、カタログで以下のようなTrafficScriptルールを作成します。

```
$path = http.getPath();
if( string.endsWith( $path, ".php" )) {

    # Persist on the PHPSESSID cookie, IP address and
    # user agent
    $phpCookie = http.cookie( "PHPSESSID" ) .
        connection.getRemoteIP() .
        http.getHeader( "User-Agent" );

    # Set the persistence key to our unique value
    connection.setPersistenceKey( $phpCookie );

    # Select the PHP Session Persistence Class
    connection.setPersistence( "PHP" );
}
```

このルールを、トラフィックを処理する仮想サーバに適用します。pool.use()関数に渡す文字列(\$phpCookie)は、プールが使用するセッション維持データです。

上記のルールでは、その他の拡張子(.htmlや.jpgなど)のファイルに対するリクエストは無視されます。これらのリクエストは、仮想サーバが使用する他のルール、またはデフォルトプールで設定されているデフォルトのセッション維持クラスに渡されます。

また、必要なプールは1つのみで、セッション維持が2つのセッション維持クラスによって全て管理されることに注意してください。

SSL暗号化

本章では、Traffic ManagerでのSecure Sockets Layer (SSL)暗号化の使用方法について説明します。SSLの概要、Traffic Managerでの認証および暗号化の管理方法、および証明書の保存について扱います。



SSLプロトコルのバージョン3は、トランスポート層セキュリティ(TLS)プロトコルのバージョン1.0、1.1、1.2、および1.3で置き換えられました。特定のSSLバージョンが指定されている場合を除き、Traffic Managerでは、SSLとTLSの両方のプロトコルを総称して、SSLという用語を一貫して使用します。

SSLの概要

SSLは、インターネット上でトラフィックをセキュアに送信するためのプロトコルです。トラフィックは、サーバマシンとクライアントマシン間で合意された鍵を使用して暗号化されます。

SSLには、以下のような効果があります。

- サーバ認証
- クライアント認証
- 暗号化データ転送

SSLは、ほぼ全てのTCP/IPプロトコルで使用できますが、一般には、HTTP (Web)トラフィックのセキュリティを強化してHTTPSプロトコルを形成するために使用されています。

サーバ認証

サーバは、SSL通信において「SSL証明書」を使用してそれ自体の身元を明らかにします。この証明書には、組織の名前と場所およびそのDNS名が含まれ、これによってクライアントはアクセス先が正しいサイトであることを確認できます。

SSL証明書は、その証明書を所有している組織が「自己署名」できます。ただし、第三者による検証がなければ、自動的にクライアントから信頼されません。信頼されるためには、SymantecやThawteなど広く認められた独立系認証局(CA)の署名が必要です。組織が証明書署名リクエスト(CSR)をCAに送信すると、CAにおいて証明書の詳細が綿密にチェックされます。認証局によるこのサービスには料金がかかる場合があることに注意してください。

CAは、自ら発行する証明書内の鍵が正しく生成され、十分な暗号強度を持つことに責任を負う場合があります。

クライアントは、オンライン証明書ステータスプロトコル(OCSP)を使用して、サーバ証明書が取り消されているか調べようとする場合があります。TLS「status_request」拡張を使用することにより、このチェックをTLSハンドシェイクに含めることができ(「OCSP Stapling」と呼ばれます)、クライアントからOCSPレスポンスへ別途接続する必要がなくなります。OCSP Staplingを有効化するには、[OCSPの設定](#)を参照してください。

OCSPとは

OCSPは、使用時点におけるSSL証明書の有効性を取得するためのインターネットプロトコルです。CRLには、リストを最新の状態に保つために更新版を頻繁にダウンロードしなければならないなどの制限がありますが、OCSPはこのようなCRL固有の欠点を解決するために証明書失効リスト(CRL) ([クライアント認証におけるCRL](#)を参照)に代わる手段として作成されました。

ユーザは、セキュアサービスにアクセスしようとするとき、OCSPサーバ(レスポンスと言う)に証明書のステータス情報をリクエストするHTTPリクエストを送信します。このリクエストは、ASN.1メッセージの形式にパッケージ化され、オプションで証明書によって署名され、レスポンスに送信されます。このリクエストに対し、レスポンスは「good (良好)」、「revoked (取り消し済み)」、または「unknown (不明)」のレスポンスを返します。

OCSPの詳細については、<http://tools.ietf.org/html/rfc6960>を参照してください。

クライアント認証

企業のイントラネットやエクストラネットなど、承認された特定のユーザにのみサービスへのアクセスを許可したい場合もあります。これを行うために、クライアントに対して、信頼された認証局によって署名されたSSL証明書を提示するようリクエストすることができます。

Traffic Managerは、CRLおよびOCSPを使用してこれらの証明書の有効性を確認します。

クライアント認証におけるCRL

Traffic Managerでは、信頼された認証局はカタログに保持されます。各認証局は、認証失効リスト(CRL)を配信することができ、これもこのカタログに保持されます。通常、証明書には12か月など一定の有効期間がありますが、この期間が期限切れになる前に取り消される場合もあります。この場合、認証局は、その証明書が以降信頼されないよう証明書失効リストに追加します。

クライアント認証におけるOCSP

OCSPを使用すると、クライアント証明書の現在のステータスを確認することができます。証明書が良好として報告されない限り、SSLコネクションは切断されます。セキュアサービスにOCSPを設定するには、[OCSPの設定](#)を参照してください。

暗号化データ転送

サーバとクライアントは、相互の身元を確認すると、データ転送に使用する暗号化鍵について合意します。これには、効率上の理由から、識別鍵とは異なる鍵が使用されます。データは、第三者が読み取れないよう転送前に暗号化されます。また、SSLには、データストリームの損傷を検出するための信頼性機能があります。これらの機能により、クライアントとサーバは、相互の通信の秘密が守られ、データに損傷がないことに確信を持つことができます。

Traffic ManagerのSSL機能

復号化と暗号化

Traffic Managerでは、仮想サーバ内でSSLトラフィックを復号化することができます。この機能は、以下の2つの理由から有益です。

- 復号化後、Traffic Managerのトラフィック分析機能をリクエスト全体に使用することができます。サービスプロテクション手法によって悪意のあるコンテンツ、ウイルス、Webワームをフィルタしたり、ルールによってリクエストのヘッダおよびボディをチェックし、その情報に基づいてルーティングを決定したりできます。復号化しない場合、パケットから得られる情報はほんのわずかです。
- 復号化には処理能力が必要です。Traffic Managerがノードに渡す前にリクエストを復号化すれば、バックエンドサーバの負荷が軽減されるため、効率の向上を図ることができます。

TrafficScriptルールを使用するために仮想サーバでSSLトラフィックを復号化する場合は、一般に、そのトラフィックをノードに送信する前に暗号化します。この暗号化はTraffic Managerの設定で作成するプールによって処理され、システムにおいて完全なエンドツーエンドのセキュリティが実現されます。

SSL証明書カタログ

Traffic Managerでは、SSLサービスで利用できるSSLサーバ証明書、クライアント証明書、認証局、および証明書失効リストが一元管理される中央ストアとして、SSL証明書カタログが使用されます。

仮想サーバは、デフォルトサービスへのトラフィック、およびオプションとして仮想サーバの設定で定義される「SSLサイト」へのトラフィックを管理します。各SSLサイトでは、カタログからの最大2つのSSL証明書と、IPアドレスまたはホスト名のいずれかが関連付けられます。詳細については、[1台の仮想サーバを使用した複数のサイトへの対応](#)を参照してください。

SSL復号化ウィザード

SSL Decrypt a service (サービスのSSL復号)ウィザードを使用すると、画面の簡単な指示に従うことで、SSL復号化を適切に設定することができます。このウィザードでは、以下の操作を実行します。

- SSL仮想サーバでSSL復号化を有効化する
- カタログのSSL証明書を仮想サーバに割り当てる
- 仮想サーバが使用するデフォルトプールでSSL暗号化を有効化する
- 全てのプロトコルタイプを基礎となるプロトコルタイプ(非SSL)に変更する



SSL Decrypt a service (サービスのSSL復号)ウィザードを使用するには、1台以上のSSL仮想サーバ(タイプは問わない)が必要です。必要なSSL証明書がない場合、Traffic Managerでウィザードを使用して証明書を作成することができます。

SSL Decrypt a service (サービスのSSL復号)ウィザードを使用する

1. ツールバーの**Help** (ヘルプ)リンクの横にある「Wizards (ウィザード)」ドロップダウンリストをクリックし、**SSL Decrypt a service** (サービスのSSL復号化)を選択します。新しいウィンドウが開き、ウィザードにより、受信時に復号化し、プールへの引き渡し前に再度トラフィックを暗号化しようサービスが設定されることを示す説明が表示されます。**Next** (次へ)をクリックします。
2. 復号化するサービスを選択します。リストには、SSLプロトコルを使用している仮想サーバのみが表示されることに注意してください。**Next** (次へ)をクリックします。
3. 受信したリクエストの復号化に使用する証明書を選択(またはCreate New (新規作成)をクリックして証明書を追加)します。**Next** (次へ)をクリックします。
4. プロトコルタイプを選択します。これは、基礎となる復号化後のプロトコルです。例えば、HTTPSリクエストは、内部的にHTTPに復号化されます。この場合は、基礎となるプロトコルとして単純にHTTPを選択します。これが、SSLでラップされていない状態で受け入れるプロトコルタイプだからです。
5. **Finish** (終了)をクリックしてウィザードを終了します。

SSL証明書の設定

SSL証明書を設定するには、**Catalogs** (カタログ) > SSLをクリックし、要件に応じて**SSL Server Certificates catalog** (SSLサーバ証明書カタログ)または**SSL Client Certificates catalog** (SSLクライアント証明書カタログ)をクリックします。

- 「サーバ証明書」は、Traffic Managerがホストする、SSLで暗号化されたサービスを識別するために使用されます。

- 「クライアント証明書」は、Traffic Managerがそれ自体をSSLノードに対して認証する必要がある場合に使用されます。



以降のセクションでは、クライアントおよびサーバの両方のSSL証明書に適用できる機能について取り上げます。

自己署名SSL証明書の新規作成

自己署名SSL証明書を新規作成する

1. 管理UIのツールバーでCatalogs (カタログ)をクリックし、SSLタブをクリックします。そこから、必要に応じてSSL Server Certificates Catalog (SSLサーバ証明書カタログ) (サーバ証明書の場合)またはSSL Client Certificates catalog (SSLクライアント証明書カタログ)のEdit (編集)ボタンをクリックします。
2. Traffic Managerに設定済みの既存の証明書のリストが表示され、証明書の作成またはインポートが可能になります。テストや内部利用が目的の場合には、SSL証明書を自己署名することができます。これは、証明書が信頼できる第三者によって署名されていないことを表し、サーバを認証する手段として信頼して利用すべきでないということを意味します。

Enter a short name to identify your certificate. If you leave this blank, the 'Common Name' field or the first 'Subject Alternative Name' will be used.

Name:

List DNS names and IP addresses to include them in the certificate's Subject Alternative Name extension.

Subject Alternative Name(s):

Subject Alternative Name:

Subject Alternative Name:

The public DNS address of your server, such as 'secure.yourcompany.com':

Common Name (CN):

The name of your organization, such as 'Your Company':

Organization (O):

The unit within your organization, such as 'Sales':

Organizational Unit (OU): (optional)

Your location (town or city), such as 'Anytown':

Location (L):

Your state or province, such as 'Somestate':

State (S): (required for US only)

Your two-letter country code, such as 'US', 'GB' or 'FR':

Country (C):

How long should this certificate be valid for:

Expires in:

Private key type (2048 bit RSA or P-256 ECDSA recommended):

Key type:

3. **Create Self-Signed Certificate** (自己署名証明書作成)/**Certificate Signing Request** (証明書署名リクエスト)をクリックします。
4. 証明書に分かりやすい**Name** (名前)を付けます。
5. この証明書に対して1つ以上の**Subject Alternative Name**を追加するには、「+」アイコンをクリックします。各エントリに対して、この証明書で保護する有効なDNS名またはIPアドレスを使用します。
6. **Common Name** (CN、一般名称)を入力します。これにはこの証明書を使用するサーバのDNS名、例えば「secure.yourcompany.com」などを指定する必要があります。ほとんどのクライアントは、証明書のCNとサーバのDNSアドレスが一致しない場合に警告メッセージを表示します。1つ以上のSubject Alternative Nameを指定した場合、RFC6125準拠クライアントソフトウェアでCommon Name (CN、一般名称)は使用されません。
7. **Organization** (組織)、**Organizational Unit** (組織単位)、**Address** (住所)に、それぞれ入力します。これらは証明書に表示される情報です。

8. 2文字のISO国コード標準を使用して、**Country** (国)コードを入力します¹。例えば、英国は「GB」、ドイツは「DE」です。
9. 証明書の**Expires In** (有効期限) (デフォルトは1年)を設定します。
10. 証明書の秘密鍵サイズ(**Key type** (鍵タイプ))を選択します。P256、P384、またはP521を選択してECDSA鍵を含む証明書を作成するか、1024、2048、3072または4096を選択して、RSA鍵を含む証明書を作成します。
11. **Create Certificate** (証明書の作成)をクリックします。エラーがなければ、Traffic ManagerによってSSL証明書が作成され、SSL証明書カタログに追加されます。

Catalogs (カタログ)ページで**SSL Certificates Catalog** (SSL証明書カタログ)の横にある拡張タブをクリックすると、証明書が追加されたことを確認できます。各証明書とその有効期限がテーブルに表示されます。

証明書データの管理

既存の自己署名証明書を変更するには、Edit Certificate (証明書編集)ページで値を更新し、修正したフィールドに基づいて新しい自己署名証明書を生成します。

それには、**Catalogs** (カタログ) > **SSL** > **SSL [Client | Server] Certificates Catalog** (SSL [クライアント|サーバ]証明書カタログ)をクリックします。次に、既存の自己署名SSL証明書の名前をクリックして、Edit Certificate (証明書編集)ページにアクセスします。Edit Certificate (証明書編集)セクションで、証明書の既存のフィールドを変更し、**Update Certificate** (証明書更新)をクリックします。



認証局によって署名された証明書を編集することはできません。編集すると、代替の証明書が自己署名されます。署名証明書を更新するには、新しい自己署名証明書を作成し、それに基づいて新しい証明書の署名リクエストを行います。

SSL証明書のコピーを新たな名前で作成するには、「Copy Certificate (証明書コピー)」セクションを使用します。コピーの新しい名前を入力し、**Copy Certificate** (証明書コピー)をクリックします。新しいSSL証明書が関連するカタログに直ちに追加されます。

証明書署名リクエストの作成

証明書署名リクエスト(CSR)は、身元を明らかにするデジタル証明書を取得するために、個人が認証局(CA)に対して行う正式なリクエストです。認証局とは、第三者が使用する証明書の発行を担う機関です。

¹ ISO-3166標準で定義されています。

自己署名証明書の作成後、以下の手順に従って、自己署名証明書に含まれる情報に基づいて証明書署名リクエストを作成することができます。

1. **Catalogs** (カタログ) > **SSL** > **[Server|Client] Certs** ([サーバ|クライアント] 証明書)画面に移動します。
2. 必要な証明書をクリックして編集します。
3. 「Certificate Signing (証明書署名)」セクションで、**Export CSR** (CSRのエクスポート)/**Update certificate** (証明書更新)をクリックします。

Traffic Managerに証明書リクエストが表示されます。

SSL
Certificates
Catalog

SSL Certificate: SSL Cert 1
This form helps you to sign your certificate.

Certificate Signing Request (CSR)
Your Certificate Authority will use this Certificate Request text to create and issue a trusted certificate, based on this certificate.

```

-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBqDCCARECAQAwDELMAkGA1UEBhMCRC0xEjAQBgNVBAcTCUNhbWJyaWRnZTER
MA8GA1UEChMIUm12ZXJiZWQxY29tMIGfMA0GC8qGSIb3DQEBAQUAA4GNADCBiQKB
gQDfmczkzdumadLvx19PLHBpykMfrZOOSY4n+1zuYhvKJqLbmesFQv5+41v+Jw3Z
zhTd/S/GgzieXl+3z9PMo9x15HQD3Ey7fEB7rum6iAFnccf7nV55hJWthpMEPMSo
tTCrBODoX5t/i1V90FuFoyPKGyYBcjstqREFbPvneN0ukwIDAQABoAAwDQYJKoZI
hvcNAQEFBQADgYEAiuV66PRzBYv1NJ2qYlE2OlHxbEyXaqBORG+Ysb++UcXgMwS3
x/+Kr9i/eOSPLSO77sB7+Wvx2IIBrzzG2hoaGCOOLUKCyLPWdlm03BpTK2CYTbyK
Y72PF5BPVDo6eoFj fMNMA02RuzA8POEA1uUh2V8BMDmJmVVy3mXBtVH6v4E=

```

Replace certificate
Once you have received a new certificate, paste it here to replace your current certificate.

Update Certificate

Note: This will completely replace your current certificate. You may wish to copy your current certificate before doing this.

このウィンドウの上部に表示されるテキストは、CAによる署名に適した、保存されている証明書内の情報を符号化したものです。このテキストをコピーしてリクエストに貼り付けることができます。リクエストにその他のクレデンシャルや追加情報を設定するよう求められる場合があります。

認証局から、認証局がいずれかの公開証明書(または公開証明書チェーン内の証明書)でデジタル署名した代替証明書が返されます。返された証明書のテキストをウィンドウの下部に貼り付けて**Update certificate** (証明書更新)をクリックすると、このプロセスは完了します。

新規SSL証明書のインポート

既存の証明書をインポートするには、以下のものがが必要です。

- 証明書ファイル
- 秘密鍵ファイル

これらのファイルは、アップロード用のテキストファイルにカットアンドペースト可能な標準のPEM符号化フォーマットになっています。PEM符号化証明書のフォーマットは以下のとおりです。

```
-----BEGIN CERTIFICATE-----
MIIBPLMAkGA1UEBhMCWkExFTATBgNVBAGTDfDlc3Rlcm4gQ2Fw
ZTESMwZSBUb3duMRQwEgYDVQQKEwtPcHBvcnR1bml0aTEYMBYG
A1UEC1cnZpY2VzMRowGAYDVQQDExF3d3cuZm9yd2FyZC5jb256
YTBAmbAQUAA0kAMEYCCQDDT5oxxeBWu5WLHD/G4BJ+PobiC9d7S
6pDvAtXdm2j190D1kgDoSp5ZyGSgwJh2V7diuuPlHDAgEDoAAw
DQYJKDQQBf8ZHIu4H8ik2vZQngXh8v+iGnAXD1AvUjuDPCWzFu
pReiq7UR8Z0wiJBeaqiuvTDnTFMz6oCq6htdH7/tvKhh==
-----END CERTIFICATE-----
```

アップロードするには、関連するClient/Server SSL Certificate Catalog (クライアント/サーバSSL証明書カタログ)の**Import Certificate** (証明書インポート)リンクをクリックします。インポート後の証明書の名前、証明書ファイルおよび秘密鍵ファイルの名前を指定します。最後に、**Import Certificate** (証明書のインポート)をクリックして、プロセスを完了します。Traffic Managerは、証明書を適切なカタログに追加し、自動的に鍵データを全てのTraffic Managerにセキュアに配布します。

注記

インポートプロセス中に作成したSSL秘密鍵の一時コピーは必ず削除してください。秘密鍵は大切なものであるため、セキュリティが確保されていない場所に保存しないでください。

認証局によって署名された証明書が「ルート証明書」で直接署名されていない場合、署名された証明書と認証局の中間証明書の両方をクライアントに送信することが必要になることがあります。

中間証明書の使用方法

Webブラウザおよびその他のSSLクライアントには、信頼する認証局の一連のルート証明書があらかじめ設定されています。ユーザはこれらのルート証明書により、信頼するいずれかの認証局によって署名されたSSLサーバ証明書を使用しているSSLサービスに接続することができます。同様に、バックエンドSSLノードは、Traffic Managerが提示するクライアントSSL証明書を認証するようあらかじめ設定されている場合があります。

ただし、管理を容易にし、セキュリティを強化するために、多くの認証局は「証明書チェーン」を使用しています。これらの証明書チェーンから配信されるSSL証明書は、ルート証明書で直接署名されていません。代わりに、ルートによって署名された中間証明書によって署名されており、これによってSSL証明書から、信頼されたルート証明書までの信頼チェーンが形成されています。場合によっては、SSL証明書からルートまでの間で複数の中間証明書が使用されることもあります。

一般に、WebクライアントおよびSSLノードには中間証明書が設定されていません。例えば、Webクライアントにはサーバ証明書が提示されますが、その証明書が信頼されたルート証明書によって(間接的であっても)署名されたものであることを検証する手段がありません。この場合、SSLサーバ(Traffic Managerなど)は、クライアントがSSL証明書を検証できるようSSL証明書チェーンの全部または一部を提示する必要があります。証明書チェーンには、SSL証明書および全ての中間証明書が含まれますが、ルート証明書は必ずしも含まれている必要はありません。

証明書チェーンが使用されているかどうかは、認証局から通知されます。関連するサーバまたはクライアントSSL証明書カタログに全ての中間証明書をアップロードする必要があります。

Certificate signing

This certificate has been signed by an external 'Certificate Authority'. You can replace it with a new certificate, for example if it has nearly expired, your details have changed, or you have a new certificate signed by another Certificate Authority.

⬇ Export CSR / Update Certificate

⬆ Add Intermediate Certificate

Traffic Managerは、アップロードした中間証明書が現在のSSL証明書(またはチェーン内の1つ前の中間証明書)で署名されていることを検証します。

認証局の証明書とCRLファイルの管理

Traffic Managerでは、SSLコネクションするリモートクライアントに対し、そのクライアントを認証するためにクライアント証明書を提示するようリクエストすることができます。この動作を有効化するには、信頼する認証局(CA)からの公開証明書と、オプションで認証局から配信される証明書失効リスト(CRL)をTraffic Managerに設定する必要があります。

1つ以上のCA証明書をインストールすると、ssl_encryptとssl_strict_verifyを有効化したTraffic Manager設定の各プールでは、コネクションが試行されたときにCA証明書のオーナーによって発行された証明書の認証が要求されます。

CA証明書とCRLを管理するには、**Catalogs (カタログ) > SSL > Certificate Authorities and Certificate Revocation Lists Catalog** (認証局および証明書失効リストカタログ)をクリックします。

Traffic Managerに新しいCAまたはCRLファイルをインポートするには、**Import certificate or CRL** (証明書またはCRLのインポート)をクリックし、以下のいずれかの方法を使用します。

- **Choose File** (ファイルの選択)をクリックして、ローカルワークステーションからTraffic Managerに証明書またはCRLをアップロードします。
- Traffic Managerの**File URL** (ファイルのURL)ボックスにHTTP URLまたはHTTPS URLを入力して、直接ダウンロードします。
- **File Contents** (ファイルコンテンツ)ボックスにファイルのコンテンツを入力またはコピーします(PEM符号化)。

Import File (ファイルのインポート)をクリックして、プロセスを完了します。Traffic ManagerがCAまたはCRLファイルをインポートし、新しい情報をクラスタ内の全てのTraffic Managerに伝搬します。



ノードに対して暗号化コネクションが実行されたとき、またはクライアント証明書を認証するために仮想サーバが設定されたときに、暗号化セキュリティを保証するために、CAが自ら発行する証明書の公開鍵有効性チェックを確実に実行するようにします。例えば、FIPSモードが有効化されている場合、NIST SP 800-89「Recommendation for Obtaining Assurances for Digital Signature Applications」(セクション5.2)によって要求されるチェックを実行したことが確認されているCAのみをCAカタログに含める必要があります。

SSL復号化

Traffic Managerは、クライアントとバックエンドプールとの間でリクエストをプロキシする際に、SSLカタログに保存されているサーバ証明書やクライアント証明書を使用して、SSLトラフィックを復号化し、再度動的に暗号化することができます。

例えば、Traffic Managerは、TrafficScriptルールを適用するためにトラフィックを復号化した後、データをノードに渡す前に必要に応じてデータを再度暗号化できます。これにより、信頼されていない可能性があるネットワークでも必要なセキュリティを損なうことなく、柔軟にリクエストを管理することができます。このように、Traffic Managerは、完全に透過的なSSLゲートウェイとして機能します。

暗号化に対する攻撃者が利用できるコンピューティングリソースが増加するほど、SSL/TLSサイトを保護するために強度の高い鍵が必要になります。しかし、DSA鍵やRSA鍵を使用するために必要な処理時間は、鍵が長くなるほど急速に増大します。この問題に対処するために、Traffic Managerには、強度の高い鍵に対する効率の高い楕円曲線暗号化(ECC)オプションがあります。具体的には、Traffic Managerは楕円曲線DSA(ECDSA)に基づく鍵をサポートしています。

SSL復号化の設定

SSL復号化を有効にするには、**Services** (サービス) > **Virtual Servers** (仮想サーバ)をクリックし、SSLをサポートしている仮想サーバの**Edit** (編集)ボタンをクリックします。

仮想サーバの設定オプションのリストで、「SSL Decryption (SSL復号)」の**Edit** (編集)ボタンをクリックします。

この仮想サーバのSSL復号化に関する設定オプションのリストが表示されます。

SSL Decryption

These settings control how SSL connections are decrypted.

Whether or not the virtual server should decrypt incoming SSL traffic.

ssl_decrypt: ☒ Yes ☐ No

Which SSL certificate(s) should this virtual server use?

Additional certificates can be supplied to match different sites hosted by this virtual server. You can specify a different certificate for any hostname or IP address. The wildcard character "*" can be used to match multiple hostnames. If none of the addresses or hostnames match the default certificate will be used.

Note: Hostname mappings require support of the TLS 1.0 'Server Name Indication' extension, which is not supported by all browsers.

certificate: Default Certificates: RSA-Cert (secure.pulsesecure.net, Expires 04 Oct 2018, RSA) ECDSA-Cert (secure.pulsesecure.net, Expires 04 Oct 2018, RSA)

IP Address / Host Name	Certificate	Remove
example.pulsesecure.net	RSA-Site-Cert (secure.pulsesecure.net, Expires 04 Oct 2018, RSA)	☐
	ECDSA-Site-Cert (secure.pulsesecure.net, Expires 04 Oct 2018, RSA)	☐

Add certificate mapping:

IP Address / Host Name:

Certificates: Select a certificate... Select a certificate...

Manage SSL Certificates

Whether or not the virtual server should add HTTP headers to each request to show the SSL connection parameters.

ssl_headers: ☐ Yes ☒ No

If the traffic manager is receiving traffic sent from another traffic manager, then enabling this option will allow it to decode extra information on the true origin of the SSL connection. This information is supplied by the first traffic manager.

ssl_trust_magic: ☐ Yes ☒ No

Whether or not to send an SSL/TLS "close alert" when the traffic manager is initiating an SSL socket disconnection.

ssl_send_close_alerts: ☒ Yes ☐ No

If OCSP URIs are present in certificates used by this virtual server, then enabling this option will allow the traffic manager to provide OCSP responses for these certificates as part of the handshake, if the client sends a TLS status_request extension in the ClientHello.

ssl_ocsp_stapling: ☐ Yes ☒ No

SSL復号化を有効にするには、**ssl_decrypt**をYes (はい)に設定し、ページの下部で**Update** (更新)をクリックします。

デフォルトのSSLサイトで使用する証明書を選択するには、**certificate** (証明書)設定を使用します。この仮想サーバによって処理される追加のSSLサイトに証明書のマッピングを追加するには、「Add certificate mapping (証明書のマッピングの追加)」セクションを使用します。

いずれの場合にも、表示される1組のドロップダウンリストを使用して、1つまたは2つの証明書を選択します。1つの証明書を指定する場合は、任意の該当する証明書の鍵タイプを使用できます。しかし、2つの証明書を指定する場合は、各証明書に対して異なる鍵タイプを選択する必要があります。詳細については、[複数の証明書を使用した他のクライアントとの互換性の実現](#)を参照してください。

以下の設定を行うこともできます。

- **ssl_headers:** Traffic Managerには、実行した復号化の詳細(HTTPS復号化のみ)が記述されたHTTPヘッダを追加するためのオプションがあります。HTTPヘッダには、クライアントの暗号、セッションID、および有効なクライアント証明書が提示されたかどうか記述され、バックエンドアプリケーションは、これらのヘッダをチェックして使用された暗号化パラメータを特定することができます。

ヘッダ名	値の例
SSLClientCipher	SSL_RSA_WITH_AES_128_CBC_SHA256, version=TLSv1.2, bits=128
SSLClientCertStatus	NoClientCertまたはOK
SSLSessionID	8723367551317ABE5443278C6E... (64文字の16進数文字列)

- **ssl_trust_magic:** Traffic Managerの2つの独立したクラスタを含む環境がある場合、あるクラスタから他のクラスタに転送されるSSLトラフィックでは、通常は最初のクラスタが接続元と見なされます。コネクションの実際の接続元IPアドレスを相互に渡すようにTraffic Managerに指示するには、この設定を使用します。



この設定を使用するには、Traffic Managerが適切に設定されている必要があります。詳細については、プールのSSL設定の**ssl_enhance**を参照してください。

- **ssl_send_close_alerts:** トランケーション攻撃を防ぐために、SSLの仕様では、切断アラートの送信によってコネクションが終了しているという情報を、クライアントとサーバが共有することが要求されています。一部のクライアント、特にFTPクライアントは切断アラートに厳密なため、この設定はデフォルトで有効化されており、アラートが省略されると正常に機能しません。Microsoft Internet Explorerの特定のバージョンを含む一部の旧式のSSL実装では、切断アラートが正しく動作しなくなる可能性があることに注意してください。使用しているサービスがこれらに該当する場合は、この設定を無効化することをお奨めします。TrafficScriptルールを使用することにより、この動作を動的に変更することもできます。

- **ssl_ocsp_stapling:** status_request拡張がクライアントによって送信される場合、このオプションを有効化することにより、仮想サーバがOCSP証明書ステータスをTLSハンドシェイクに含められるようになります。またこれにより、各証明書の「AIA」拡張フィールドのOCSP URIで指定されたレスポンスに対し、バックグラウンドで仮想サーバの各証明書に対するOCSPリクエストが実行されます。デフォルトでは、このオプションは無効化されています。

Update (更新)をクリックし、変更内容を仮想サーバに適用します。

1台の仮想サーバを使用した複数のサイトへの対応

Traffic Managerは、以下の2通りの方法で複数のサービスを区別することができます。

- **宛先IPアドレス:** 例えば、「secure.site1.com」と「secure.site2.com」を異なるIPアドレスに名前解決することができます。Traffic Managerは、両方のIPアドレスを監視し、コネクションを受け取ったIPアドレスに基づいて使用する証明書を選択します。
- **TLSサーバ名の指定¹:** TLSには、TLSハンドシェイクの開始時にクライアントがアクセスしようとしているサービスの名前(プレーンテキスト形式)を指定できるオプション機能があります。Traffic Managerはこの名前をチェックし、使用する証明書を選択します。

注記

各SSLサービスを専用IPアドレスで実行する必要がないため、TLSサーバ名の指定は拡張性が高く、管理が簡単です。しかし、Internet Explorer 7(Windows XP)などの一部の旧式のWebクライアントではこの機能がサポートされず、TLSサーバ名のサポートが必要なサービスにアクセスしようとすると、正しくない証明書を受信することがあります。

仮想サーバが複数のサイトでトラフィックを管理している場合は、クライアントがアクセスしようとしている各サイトと一致するように、Subject Alternative Name (サブジェクトの別名)を含む証明書で仮想サーバを設定する必要があります。クライアントは、サービスのDNS名と一致するSubject Alternative Nameを見つけることができない場合、代わりにCommon Name(CN、一般名)フィールドをチェックすることがあります。一致が見つからない場合、クライアントはエンドユーザに警告します。

複数の証明書の設定

単一の証明書には、仮想サーバによって提供される各SSLサイトと一致するように、複数のSubject Alternative Nameが含まれることがあります。この場合は、デフォルトとしてその証明書を使用するように仮想サーバを設定するだけで十分です。

¹ http://en.wikipedia.org/wiki/Server_Name_Indication

または、1つのSSLサイトと一致する単一のSubject Alternative Nameを含む複数の証明書で、仮想サーバを設定することもできます。次に、「Add certificate mapping (証明書のマッピングの追加)」セクションを使用して、特定の証明書のマッピングを使用するように仮想サーバを設定します。

最初のテキストボックスでIPアドレスまたはホスト名を指定し、ドロップダウンリストから証明書を選択して、新しい証明書のマッピングを作成します。ページの下部で**Update** (更新)をクリックして、このマッピングを保存します。

Traffic Managerはリクエストを受信すると、証明書の設定を調べ、送信する証明書を決定します。接続しているクライアントがSSLハンドシェイクで、マッピング内のホスト名と一致するサーバ名を提供するか、クライアントがマッピングで指定されたIPアドレスに接続する場合は、デフォルトの証明書ではなく、指定した証明書が提供されます。

Traffic Managerによって、最初に一致する証明書が選択されます。サーバ名の正確な一致、サーバ名のワイルドカードの一致、IPアドレスの正確な一致の順に確認されます。一致するものがない場合は、デフォルト証明書が使用されます。

ワイルドカードを使用した複数の証明書マッピングがサーバ名と一致する場合、Traffic Managerでは、ワイルドカード文字の比率が最も少ないマッピングが最初に使用されます。例えば、クライアントによってサーバ名「www.foo.example.com」が送信された場合、Traffic Managerでは、「*.example.com」のマッピングよりも「*.foo.example.com」のマッピングに関連付けられた証明書が使用されます。

複数の証明書を使用した他のクライアントとの互換性の実現

仮想サーバでデフォルトの証明書を設定する際や、IPアドレス/ホスト名用に証明書のマッピングを設定する際は、Traffic Managerで2種類の証明書を設定できます。SSLハンドシェイク中に、Traffic Managerは暗号スイート、署名アルゴリズム、およびクライアントがサポートを示したサーバ証明書の組み合わせを選択します。すると、次のことを行えるようになります。

- 一方の証明書チェーンではECDSAを使用し、他方ではECDSA暗号をサポートしていないTLSv1.3よりも前のクライアントが使用するRSA鍵を使用する
- 一方の証明書チェーンではRSA-PSS署名を使用し、他方ではPKCS1-v1_5署名のみを使用する



クライアントが複数の証明書チェーンを許容する場合、Traffic Managerは暗号スイートと署名アルゴリズムの最適な組み合わせと互換性がある最初のチェーンを送信します。詳細については、[暗号およびTLSバージョンの設定](#)を参照してください。

暗号およびTLSバージョンの設定

暗号化アルゴリズムは時間が経過すると脆弱になる性質があるため、SSLとTLSはサポートされる複数のアルゴリズムをネゴシエーションでき、次のようなメリットがあります。

- エンドポイントの認証
- 鍵交換
- 一括暗号化

Traffic Managerではサービスを通してネゴシエーション可能なアルゴリズムを設定でき、状況によっては利用可能なアルゴリズムの優先度を設定できます。

Traffic Managerにはセキュリティ設定のグローバルなデフォルト設定が含まれますが、それを特定のサービスの専用コントロール用として、仮想サーバ単位またはプール単位で上書きできます。また、Traffic Managerには管理UIとREST APIに使用する設定をコントロールする設定も含まれます。次のセクションで利用可能なコントロールについて説明します。

プロトコルバージョン

Traffic Managerは全てのSSL接続で、SSLv3、TLSv1.0、TLSv1.1、TLSv1.2、およびTLSv1.3をサポートしています。

クライアントがサーバに接続する際、最初に行われるアクションの1つはプロトコルバージョンのネゴシエーションです。サーバとして動作する場合、Traffic Managerはクライアントがサポートをアドバタイズしている中で、サポートされる最も高いバージョンを常に変更します。暗号選択におけるその後の要素の中には、プロトコルバージョンごとに影響がわずかに異なるものがあります。

どのプロトコルバージョンを有効にするかを設定するには、管理UIの次のページを表示します。

- 管理インターフェースおよびコントロールプレーンの接続は、**System** (システム) > **Security** (セキュリティ) > **SSL Settings for Admin Server and Internal Connections** (管理サーバおよび内部接続のSSL設定)をクリックします。「admin!support_<version>」設定を使用してプロトコルのサポートを設定します。
- サービス接続のグローバルデフォルトは、**System** (システム) > **Global Settings** (グローバル設定) > **SSL Configuration** (SSL 設定)をクリックします。「ssl!support_<version>」設定を使用して、サービス向けのプロトコルのサポートを設定します。
- 個々の仮想サーバのグローバルデフォルトを上書きするには、**Services** (サービス) > **Virtual Servers** (仮想サーバ) > **Edit** (編集) > **SSL Decryption** (SSL復号化)をクリックして、「ssl_support_<version>」を設定します。
- 個々のプールのグローバルデフォルトを上書きするには、**Services** (サービス) > **Pools** (プール) > **Edit** (編集) > **SSL Settings** (SSL設定)をクリックして、「ssl_support_<version>」を設定します。

暗号スイート

Traffic Managerでは、SSL接続で使用可能な暗号スイートのカスタムのリスト(優先順位)を設定できます。Traffic Managerで現在サポートされている暗号スイートの全リストを表示するには、**System** (システム) > **Global Settings** (グローバル設定) > **SSL Configuration** (SSL設定) ページでUIヘルプを確認します。



UIヘルプに表示される一部の暗号スイートはサポート対象のマークが付いていますが、デフォルトで無効化されています。Traffic Managerでそれらの暗号スイートを使用できるようにするには、暗号スイートの優先順位リストに明示的に追加する必要があります。または、特殊な値「all」を使用して、Traffic Managerで暗号スイートの全リストを強制的に有効化します。

Traffic Managerは接続タイプごとに個別の暗号スイートの優先順位リストを使用します。暗号スイートの優先順位リストを設定するには、管理UIの次のページを表示します。

- 管理インターフェースおよびコントロールプレーンの接続は、**System** (システム) > **Security** (セキュリティ) > **SSL Settings for Admin Server and Internal Connections** (管理サーバおよび内部コネクションのSSL設定)をクリックし、「admin!ssl3_ciphers」を設定します。このフィールドを空白のままにすると、このTraffic Managerで有効化およびサポートされている全ての暗号スイートの使用が許可されます。
- その他全ての接続のグローバル優先順位リストを設定するには、**System** (システム) > **Global Settings** (グローバル設定) > **SSL Configuration** (SSL設定)をクリックします。「ssl!cipher_suites」設定を使用して、仮想サーバおよびプール用の暗号スイートの優先順位リストを設定します。このフィールドを空白のままにすると、このTraffic Managerで有効化およびサポートされている全ての暗号スイートの使用が許可されます。
- 仮想サーバごとのグローバル優先順位リストを上書きするには、**Services** (サービス) > **Virtual Servers** (仮想サーバ) > **Edit** (編集) > **SSL Decryption** (SSL復号化)をクリックし、「ssl_cipher_suites」に専用の暗号スイートリストを設定します。このフィールドを空白のままにするとグローバルリストが使用されます。
- プールごとのグローバル優先順位リストを上書きするには、**Services** (サービス) > **Pools** (プール) > **Edit** (編集) > **SSL Settings** (SSL設定)をクリックし、「ssl_cipher_suites」に専用の暗号スイートリストを設定します。このフィールドを空白のままにするとグローバルリストが使用されます。

TLSv1.3より前の場合は、特定の暗号スイートの選択によって以下が決定されます。

- サーバが認証用に提示する必要がある公開鍵のタイプ。サポートされている鍵タイプはECDSA、RSA、DSAです。
- 鍵交換に使用されるアルゴリズム。サポートされているアルゴリズムはECDHE、DHE、non-Diffie-Hellman RSAです。
- 使用する一括暗号化アルゴリズム。サポートされているアルゴリズムは、AES-GCM、AES-CBC (HMAC)、3DES (HMAC)、RC4 (HMAC)、DES (HMAC)です。

TLSv1.3よりも前の暗号スイート用にTraffic Managerが使用する名前は、Internet Assigned Numbers Authority (IANA) TLS暗号スイートレジストリ(<https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml>)の「TLS_」プレフィックスを、「SSL_」プレフィックスに置き換えて形成されます。

例えば、Traffic Manager暗号SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHAは、IANA暗号スイートTLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHAを表します。この暗号スイートを使用する際、サーバはECDSA公開鍵をサーバ証明書内に認証として示す必要があります。使用される鍵交換アルゴリズムはECDHEで、一括暗号化には256ビットのAES-CBCが使用されます。

TLSv1.3以降では、暗号スイートの意味が変化しています。特定の暗号スイートの選択が公開鍵や鍵交換アルゴリズムに影響することはありません、一括暗号化に使用するアルゴリズムのみが決定されます。TLSv1.3暗号スイート用にTraffic Managerが使用する名前は、IANA TLS暗号スイートレジストリ内の名前と同じです。

サーバとして動作し、暗号スイートを選択する際に、Traffic Managerは適用される暗号スイートリストで最初に表示された利用可能なスイートを選択します。選択したスイートには設定した証明書との互換性が必要で、利用可能な互換性のある署名アルゴリズムが必要となります。この場合、クライアントの優先順位は考慮されません。Traffic Managerは署名アルゴリズムを選択する前に暗号スイートを選択します。

署名アルゴリズム

署名の検証に公開鍵を使用することは、TLSの重要なコンセプトです。このアプローチは、既知のコンテンツで提供されている署名を確認することで、証明書チェーンの検証と、TLSハンドシェイク内のリモートのエンティティの認証に使用されます。一般的に、Traffic ManagerはECDSA、DSA、RSA鍵に基づいた署名アルゴリズムをサポートしています。RSAについては、PKCS1標準(<https://tools.ietf.org/html/rfc8017>)で次の2つの署名スキームが定義されています。

- PKCS1-v1_5
- PSS

Traffic Managerは、証明書チェーンの検証時に両方のスキームをサポートしています。ハンドシェイク署名の署名/検証において、PKCS1-v1_5はTLSv1.2よりも前でサポートされ、PSSはTLSv1.2以降でサポートされています。



PSSは、TLSv1.2証明書確認ハンドシェイクメッセージでサポートされていません。

Traffic Managerは、証明書署名を検証するために、およびハンドシェイク署名を署名または検証する際 (TLSv1.2以降)に署名アルゴリズムを使用します。この目的で使用可能なアルゴリズムを制限するために、グローバル設定内のシステム全体の優先順位リストを設定するか、仮想サーバおよびプールのSSL設定内のローカルの上書きリストを設定します。使用する署名アルゴリズムを選択する必要がある場合(ハンドシェイク署名の生成時など)、Traffic Managerは暗号スイートと互換性のある、リスト上の最初の署名アルゴリズムを使用します。

Traffic Managerで現在サポートされている署名アルゴリズムの全リストを表示するには、**System** (システム) > **Global Settings** (グローバル設定) > **SSL Configuration** (SSL設定) ページでUIヘルプを確認します。

署名アルゴリズムの優先順位リストを設定するには、管理UIの次のページを表示します。

- 管理インターフェースおよびコントロールプレーンの接続は、**System** (システム) > **Security** (セキュリティ) > **SSL Settings for Admin Server and Internal Connections** (管理サーバおよび内部接続のSSL設定)をクリックし、「admin!ssl_signature_algorithms」を設定します。このフィールドを空白のままにすると、このTraffic Managerでサポートされている全ての署名アルゴリズムの使用が許可されます。
- その他全ての接続のグローバル優先順位リストを設定するには、**System** (システム) > **Global Settings** (グローバル設定) > **SSL Configuration** (SSL設定)をクリックします。
「ssl!signature_algorithms」設定を使用して、仮想サーバおよびプール用の署名アルゴリズムの優先順位リストを設定します。このフィールドを空白のままにすると、このTraffic Managerでサポートされている全ての署名アルゴリズムの使用が許可されます。
- 仮想サーバごとのグローバル優先順位リストを上書きするには、**Services** (サービス) > **Pools** (プール) > **Edit** (編集) > **SSL Settings** (SSL設定)をクリックし、「ssl_signature_algorithms」に専用の署名アルゴリズムリストを設定します。このフィールドを空白のままにするとグローバルリストが使用されます。
- プールごとのグローバル優先順位リストを上書きするには、**Services** (サービス) > **Pools** (プール) > **Edit** (編集) > **SSL Settings** (SSL設定)をクリックし、「ssl_signature_algorithms」に専用の署名アルゴリズムリストを設定します。このフィールドを空白のままにするとグローバルリストが使用されます。



仮想サーバまたはプールの「ssl_signature_algorithms」設定に特殊な値「default」を入力すると、対応する仮想サーバまたはプールがグローバル優先順位リストで設定された全ての値を無視し、代わりにUIヘルプでサポート対象として宣言された(かつ無効化されていない)署名アルゴリズムのデフォルトリストが強制的に使用されます。

サーバとして動作し、署名アルゴリズムを選択する際に、Traffic Managerは選択した暗号スイートと設定した証明書と互換性がある、アプリケーション署名アルゴリズムリストの最初の利用可能なアルゴリズムを選択します。Traffic Managerは暗号スイートを選択後に署名アルゴリズムを選択します。

サーバとして動作し、署名アルゴリズムを選択する際に、Traffic Managerは適用される署名アルゴリズムリストで最初に表示された利用可能なアルゴリズムを選択します。選択したアルゴリズムには、選択した暗号スイートおよび設定した証明書との互換性が必要です。Traffic Managerは暗号スイートを選択後に署名アルゴリズムを選択します。

楕円曲線

Traffic ManagerはECDHE鍵交換およびECDSA認証用に楕円曲線を使用します。次の曲線がサポートされています。

- NIST P-256 (別名SECG secp256r1またはANSI x9.62 prime256v1)
- NIST P-384 (別名SECG secp384r1)
- NIST P-521 (別名SECG secp521r1)

Traffic ManagerがECDHE鍵交換に使用するため、またはECDSA署名を署名/検証する際に、利用できる楕円曲線を制限および優先順位付けするには、グローバル設定内のシステム全体の優先順位リストを設定するか、仮想サーバおよびプールのSSL設定内のローカルの上書きリストを設定します。

楕円曲線の優先順位リストを設定するには、管理UIの次のページを表示します。

- 管理インターフェースおよびコントロールプレーンの接続は、**System** (システム) > **Security** (セキュリティ) > **SSL Settings for Admin Server and Internal Connections** (管理サーバおよび内部コネクションのSSL設定)をクリックし、「admin!ssl_elliptic_curves」を設定します。このフィールドを空白のままにすると、このTraffic Managerでサポートされている全ての楕円曲線の使用が許可されます。
- その他全ての接続のグローバル優先順位リストを設定するには、**System** (システム) > **Global Settings** (グローバル設定) > **SSL Configuration** (SSL設定)をクリックします。「ssl!elliptic_curves」設定を使用して、仮想サーバおよびプール用の楕円曲線の優先順位リストを設定します。このフィールドを空白のままにすると、このTraffic Managerでサポートされている全ての楕円曲線の使用が許可されます。
- 仮想サーバごとのグローバル優先順位リストを上書きするには、**Services** (サービス) > **Pools** (プール) > **Edit** (編集) > **SSL Settings** (SSL設定)をクリックし、「ssl_elliptic_curves」に専用の楕円曲線リストを設定します。このフィールドを空白のままにするとグローバルリストが使用されます。
- プールごとのグローバル優先順位リストを上書きするには、**Services** (サービス) > **Pools** (プール) > **Edit** (編集) > **SSL Settings** (SSL設定)をクリックし、「ssl_elliptic_curves」に専用の楕円曲線リストを設定します。このフィールドを空白のままにするとグローバルリストが使用されます。



仮想サーバまたはプールの「ssl_elliptic_curves」設定に特殊な値「default」を入力すると、対応する仮想サーバまたはプールがグローバル優先順位リストで設定された全ての値を無視し、代わりにサポートされている楕円曲線の全リストが強制的に使用されます。

クライアント証明書

SSLサーバ証明書は、SSLトランザクションにおいてサーバを識別するために使用されますが、SSLではクライアント証明書を使用することもできるため、クライアントは必要に応じてそれ自体の身元を明らかにすることができます。

デフォルトでは、Traffic Managerは、仮想サーバに接続するクライアントに対し、クライアント証明書をリクエストしません。必要な場合は、クライアントにクライアント証明書をリクエストするように仮想サーバを設定できます。

SSLトランザクションでクライアントを認証する方法を制御するには、以下の設定を使用します。

- **request_client_cert**: 仮想サーバが各クライアントに身元確認のための証明書をリクエストするかどうか、また証明書の提示をオプションにするか必須(すなわち、全てのクライアントが有効なクライアント証明書を提示する必要がある)にするかを指定することができます。**request_client_cert**を「Do not request a certificate (証明書をリクエストしない)」に設定した場合(デフォルト設定)、クライアントは、それ自体が証明書を提示しようとしているかどうかに関わらず、証明書を送信することができません。
- **ssl_client_cert_headers**: リクエストの復号化後、Traffic Managerは、オプションでHTTPヘッダを設定して、使用されているSSL証明書の証明書フィールドを記録することができます。

ヘッダ名	値の例
SSLClientCertVersion	3
SSLClientCertSerialNumber	ED41A8
SSLClientCertIssuer	C=US, ST=CA, L=San Jose, O=MyOrg
SSLClientCertSubject	C=US, ST=CA, L=San Jose, O=MyOrg
SSLClientCertNotValidBefore	1155304575 (unixtime)
SSLClientCertNotValidAfter	1186840575 (unixtime)
SSLClientCertSubjectPublicKey	RSA (2048 bit)
SSLClientCertSignatureAlgorithm	sha256withRSAEncryption
SSLClientCertHash	873FECCB50E0C1D34711ABE65BA70BA7
SSLClientCertSHA1Fingerprint	7D90A8EB6081183C0560970A7DA77306C40E5DD2

ヘッダ名	値の例
SSLClientCert (有効な場合)	PEMでエンコードされた証明書のテキストの全体。

- **client_cas**: クライアント証明書が使用されている場合、Traffic Managerは、提示された証明書が正当なものかどうかを検証する必要があります。そのため、Traffic Managerは信頼しているCAをクライアントに通知し、クライアントはそのいずれかのCAによって署名された証明書を送信します(利用可能な場合)。この設定を使用するには、初めに「Certificate Authorities and Certificate Revocation Lists Catalog (認証局および証明書失効リストカタログ)」で信頼されたCA証明書を設定し、必要に応じてここでCAを選択します。指定したCAによって認識されていない証明書をクライアントが提示した場合、Traffic Managerによってコネクションが即座に切断され、クライアントにSSLエラーが送信されます。

request_client_cert、**client_cas**、**ssl_client_cert_headers**の設定は、仮想サーバに対して確立されているTLSコネクションで、指定した宛先IPアドレスが使用されるか、指定したDNS名の *サーバ名指定拡張* が含まれる場合、オーバーライドすることができます。

サーバ名指定(SNI)とは、TLSハンドシェイク用に定義された拡張であり、TLSクライアントが接続先のサーバをTLSサーバに指定できます。これは、複数の異なるアプリケーションが同じIPアドレスを利用できるように、複数の異なるサーバに対するDNSエントリが同じIPアドレスに解決される場合に役立つことがあります。

▼ SSL Client Authentication

These settings control how clients are authenticated in SSL transactions.

Whether or not the virtual server should request an identifying certificate from each client.
request_client_cert: Do not request a client certificate

What HTTP headers the virtual server should add to each request to show the data in the client certificate.
ssl_client_cert_headers: No data

The certificate authorities that this virtual server should trust to validate client certificates. If no certificate authorities are selected, then all client certificates will be accepted.

client_cas: No certificate authorities selected, all certificates will be accepted.

[Add new Certificate Authority to the list](#)



Manage Certificate Authorities

Specific CA certificates can be used when requesting client certificates for different sets of addresses. You can specify a different set of CA certificates for any DNS hostname or IP address. The wildcard character * can be used to match any address. If none of the addresses or hostnames match the **request_client_cert**, **ssl_client_cert_headers**, or **client_cas** settings, the default settings will be used.

Note: Hostname mappings require support of the TLS 1.0 'Server Name Indication' extension.

Host or IP	Request Certificate	Insert HTTP Headers	CAs
www.example.com	Yes, allow if absent ▼	Fields ▼	10.62.165.195-CA
www2.example.com	Yes, deny if absent ▼	Fields ▼	10.62.165.195-CA

Add hostname/IP address mapping for client authentication

Host or IP:

Request Certificate: No ▼

Insert HTTP Headers: None ▼

CAs: 10.62.165.195-CA ☐

SNI拡張機能がサポートされ、使用される場合、Traffic Managerでは、受信コネクションに適用するために特定のホスト名またはIPアドレスマッピングを追加できます。マッピングを追加するには、「Add hostname/IP address mapping for client authentication」セクションを使用します。

次の設定を使用してマップパラメータを指定し、**Update** (更新)をクリックしてマッピングを追加します。

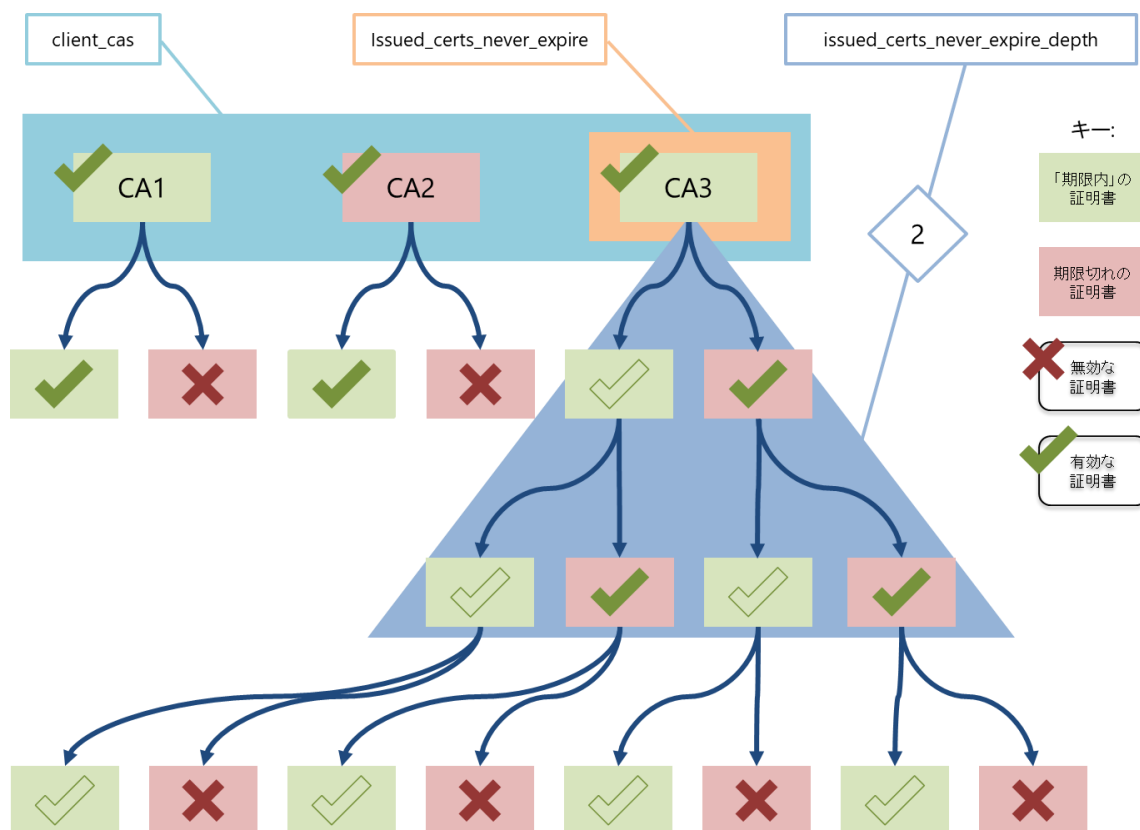
設定	説明
Host or IP (ホストまたはIP)	これは、受信コネクションを照合するために使用する文字列であり、SNIのコンテンツを照合する必要がある場合はDNS名として、またはコネクションの宛先IPアドレスを照合する必要がある場合はIPアドレスとして指定することができます。いずれの場合にも、検査するSNIまたはIPアドレスで複数の文字の代わりに「*」文字を指定できます。
Request Certificate (証明書要求)	初回のTLSハンドシェイクがクライアント証明書を要求するか否か。 以下のプロファイルを選択できます。 No (いいえ): いかなる証明書も要求しません。 Yes, allow if absent (はい、ただし証明書がない場合は許可): 証明書を要求しますが、クライアントが証明書の提供を拒否した場合は、コネクション処理の続行を許可します。 Yes, deny if absent (はい、ただし証明書がない場合は拒否): 証明書を要求しますが、クライアント証明書が提供されない場合はコネクションが中断されます。 証明書を要求しないオプションは、TrafficScript関数 <code>ssl.requestCert()</code> および <code>ssl.requireCert()</code> とともに再ハンドシェイクをサポートするTLSバージョン(v1.3より前のTLSバージョン)を使用して確立された接続で使用できます。
HTTPヘッダの挿入 (HTTPのみ)	プールノードに送信する前にリクエストに追加する必要があるHTTPヘッダフィールド(存在する場合)。 以下のプロファイルを選択できます。 None: いかなるヘッダフィールドも挿入されません。

設定	説明
	Fields Only (フィールドのみ): <code>ssl_client_cert_headers</code>の記述に従ってヘッダフィールドを挿入します。 Fields and PEM (フィールドおよびPEM): 上記のように、証明書全体を含む「SSLClientCert」ヘッダ。
CA (認証局)	証明書要求で使用する必要がある認証局。クライアントが、その認証局によって発行された1つ以上の証明書があるか否かを判断できるようにします。チェックされた各認証局は、証明書要求に含まれます。 いずれの認証局もチェックしない場合、クライアントが提供する全ての証明書は、アクセスとコネクションの処理を許可するとみなされます。

マッピングを削除するには、**Remove** (削除)列の横のチェックボックスを選択し、**Update** (更新)をクリックします。

次の設定では、クライアント証明書チェーンの検証において、選択したクライアント認証局の有効期限を無視することができます。

- issued_certs_never_expire**および**issued_certs_never_expire_depth**: 公開鍵基盤の問題が生じている組織では、クライアント証明書の設定に対するエキスパートレベルの制御を有効にするために、これらの設定を使用します。**client_cas**の一部として設定されるCAのサブセットを定義するには、**issued_certs_never_expire**を使用します。通常、チェーン内のいずれかの証明書が期限切れになった場合、Traffic Managerはそのチェーンを検証することができません。つまり、クライアント証明書が要求される場合、TLSハンドシェイクは失敗します。この設定に基づいてCAを設定した場合、Traffic Managerでは、トラストアンカーとして特定のCAまで構築される証明書チェーン内の証明書の期限切れが無視されます。このプロパティは、**issued_certs_never_expire_depth**の値を設定して、さらに制限することができます。この設定は、トラストアンカーからどの程度離れた証明書の期限切れの無視を許可するかを示します。デフォルト設定は「1」です。これは、検証済みのクライアント証明書チェーン内で、CAによって直接発行された証明書のみの期限切れが許可されることを示します。



OCSPの設定

オンライン証明書ステータスプロトコル(OCSP)を使用すると、OCSPレスポンドと呼ばれる中央のサーバで証明書の取り消しステータスを確認することができます。OCSPは、証明書失効リスト(CRL)の代替手段として広く利用されています。詳細については、[クライアント認証](#)を参照してください。

Traffic Managerの**Virtual Servers** (仮想サーバ) > **SSL Decryption** (SSL復号)ページには、OCSPを設定するための専用のセクションがあります。OCSPを設定するには、以下の設定を使用します。

設定	説明
<code>ssl_use_ocsp</code>	yes (はい)に設定すると、この仮想サーバはOCSPを使用して、クライアント証明書が取り消されていないことを確認します。

設定	説明
ssl_ocsp_timeout	Traffic Managerは、OCSPレスポンスへの問い合わせ時に、この設定に指定した時間だけレスポンスを待機します。この時間制限内にOCSPレスポンスがレスポンスを返さない場合、クライアントの証明書は拒否されます。
ssl_ocsp_max_response_age	OCSPレスポンスの「thisUpdate」フィールドが、この設定に指定した秒数より古い場合、レスポンスは無効と見なされます。0 (ゼロ)に設定すると、レスポンスが期限切れかどうかを判断するためにレスポンスの「nextUpdate」フィールドが使用されます。 レスポンスの「nextUpdate」フィールドに指定された時間が経過すると、この設定に関わらず、レスポンスは無効と見なされます。
ssl_ocsp_time_tolerance	OCSPレスポンスとTraffic Managerのクロックにわずかなずれがある場合、レスポンスの「thisUpdate」および「nextUpdate」フィールドに指定された時間によっては、Traffic Managerがそのレスポンスを無効と見なすことがあります。例えばレスポンスのクロックがTraffic Managerのクロックより数秒進んでいる場合、Traffic Managerがレスポンスの「thisUpdate」フィールドを未来の時間として捉える可能性があります。 この設定では、クロックの差の許容秒数を指定できます。

以下のOCSPレスポンス設定は、発行者単位で指定します。そのため、認証局カタログに保持されている発行者ごとに、その発行者が署名した証明書に異なる設定を指定することができます。

特定の発行者の設定を追加するには、**Add Issuer Specific Settings** (発行者固有設定の追加)セクションで認証局を選択します。Traffic Managerが、明示的に設定されていない発行者によるクライアント証明書を受信した場合は、**Default Settings** (デフォルト設定)が使用されます。

設定	説明
OCSP check (OCSPチェック)	この設定は、発行者のOCSPの確認を有効化/無効化するときに使用します。「Always (常時)」を選択すると、適切なOCSPレスポンスURLを特定できない場合にリモートコネクションは切断されます。

設定	説明
Use AIA for URL (URLにAIAを使用)	発行者は、X.509のAIA拡張フィールドを使用してクライアント証明書にOCSPレスポンドURLを埋め込むことができます。この設定を有効化すると、この拡張が利用可能な場合、Traffic ManagerはAIAを使用します。それ以外の場合、 Fallback URL (フォールバックURL)設定のURLを使用します。
Fallback URL (フォールバックURL)	レスポンドURLを他の方法で特定できない場合、このURLが使用されます。
Request signing certificate (証明書署名要求)	レスポンドがリクエストの送信元を識別し、認証できるように、OCSPリクエストに署名することができます。SSL証明書カタログにアップロードされた証明書からリクエストへの署名に使用する証明書を選択するか、Default Settings (デフォルト設定)に設定されている証明書を使用するか、またはリクエストへの署名を無効化することができます。
Use nonce extension (nonce拡張を使用)	クライアント証明書の取り消し後に攻撃者が「良好な」OCSPレスポンスを返せないようにするために、ユニークなデータ配列をOCSPリクエストおよびレスポンスに追加することができます。nonce拡張を使用すると、常に「strict」モードを使用してnonce (ランダムな文字列)を返すようレスポンドにリクエストできます。それ以外の場合、Traffic Managerは、nonceが含まれていないレスポンスを受け入れます。 この拡張を使用すると、レスポンドは、性能を向上させるために実行するレスポンスの事前生成を行わなくなります。また、この拡張は、全てのOCSPレスポンドがサポートしているわけではありません。サポートされていない場合、レスポンドは「unauthorized」レスポンスを返す可能性があります。
Responder certificate (レスポンド証明書)	OCSPレスポンスは、その正当性を証明するためにレスポンドによって署名されます。レスポンスへの署名に使用する特定の証明書を、SSL Certificate Authorities (SSL認証局カタログ)にリストされている証明書から選択することができます。 または、Issuing Certificate (証明書の発行)を選択して、レスポンスが発行者の証明書によって署名されていることを想定するようシステムを設定することもできます。 あるいは、Signed by issuing certificate (証明書の発行による署名)を選択する方法もあります。この場合、証明書が発行者の証明書そのものであるか、発行者によって署名されている必要があります。また、extendedKeyUsage拡張にid-kp-OCSPSigningマーカおよびid-pkix-ocsp-nocheck拡張がなければなりません。

SSLセッションの回復

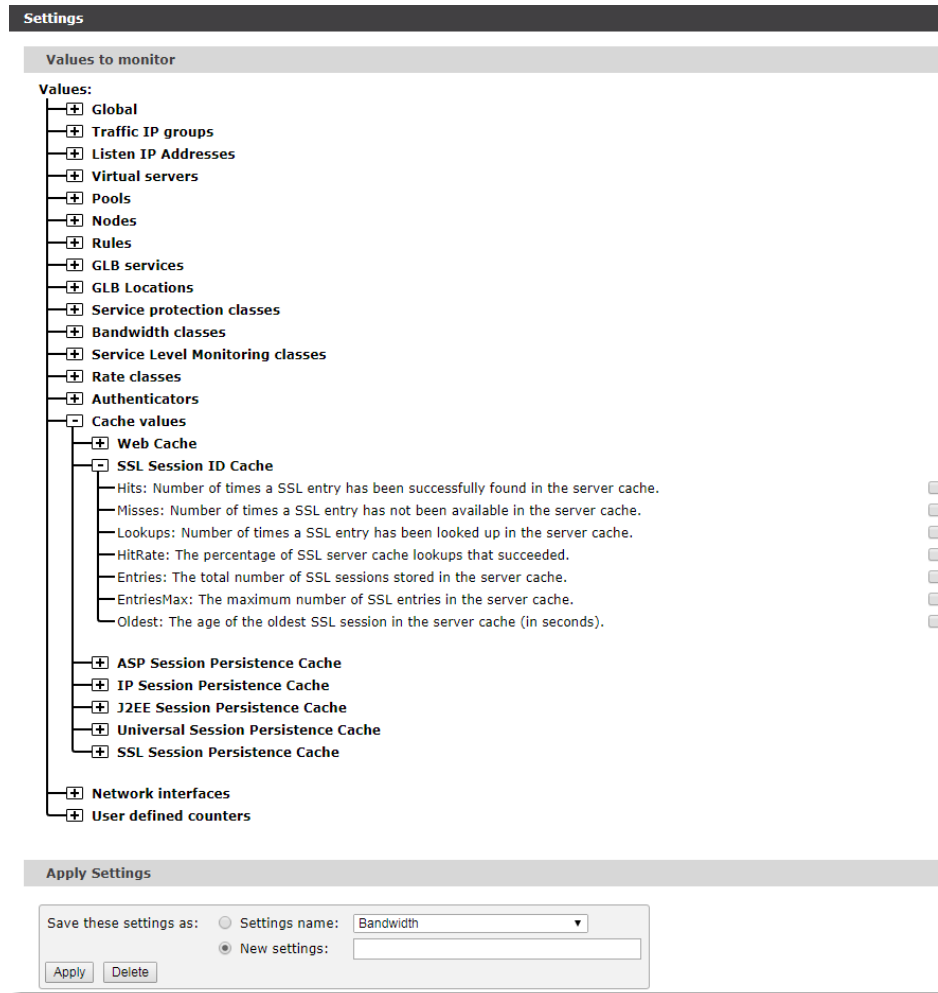
SSLセッションは、2つの当事者間で(再)確立されるセキュアチャネルを許可する詳細として定義できます。これは、コネクションにより使用される全ての暗号鍵がそこから派生する秘密を含めて、そのチャネルのために必要なセキュリティプロパティを確立するために設計されたプロセス(ハンドシェイク)により生成されます。クライアントとサーバの両方がそれらの詳細を保存し、サーバは識別子(セッションID)を、その特定のセッションを参照するときに両当事者が使用するその詳細と見なします。SSL TCPコネクションが終了し、クライアントがその後再接続する場合、クライアントはこのセッションIDを提示し、キャッシングされた暗号パラメータを使用して、以前のSSLセッションを再開することをサーバに伝えます。

Traffic Managerでは、サポートされる全てのSSL/TLSバージョンの再開が可能です。Traffic Managerは以下の場合に、2つのSSLセッションIDキャッシュを管理します。

- SSLサーバとして動作している(すなわち、「SSL復号化」が有効な状態で仮想サーバを使用している)
- SSLクライアントとして動作している(すなわち、「SSL暗号化」が有効な状態でプールを使用している、またはTrafficScriptルールで発信httpsリクエストを作成している)

これらのキャッシュのサイズおよび有効期限を変更するには**System** (システム) > **Global Settings** (グローバル設定) > **SSL Configuration** (SSL設定) をクリックします。サーバおよびクライアントのキャッシュ用として、それぞれ`ssl!cache!size`、`ssl!cache!expiry`、`ssl!client_cache!size`および`ssl!client_cache!expiry`の設定を更新します。

SSLセッションIDキャッシュの動作をモニタリングするには、キャッシュデータ項目を現在のアクティビティグラフにプロットするか、SNMPを使用してこれらの値をモニタリングします。現在のアクティビティグラフを使用するには、**Activity** (アクティビティ) > **Current Activity** (現在のアクティビティ) をクリックします。次に、**Change data** (データの変更) をクリックし、キャッシュデータ項目を選択して、グラフにプロットします。値のリストから、**Cache values** (キャッシュ値) > **SSL Session ID Cache** (SSLセッションIDキャッシュ) を展開し、必要なキャッシュデータ項目を選択します。グラフを更新するには、**Update** (更新) をクリックします。



キャッシュがいっぱいになったときに(**Entries** (エントリ)と**EntriesMax** (エントリ最大値)を比較)、設定されている有効期限より**Oldest** (最も古いエントリ)値が大幅に低い場合、キャッシュが小さすぎるために、Traffic Managerで早すぎる時期にキャッシュエントリが期限切れになります。**System** (システム) > **Global Settings** (グローバル設定) > **SSL Configuration** (SSL設定)でキャッシュサイズを増加してみてください。

RFC 5077(サーバ側ステートなしのトランスポート層セキュリティ(TLS)セッション回復)が、1.3よりも前のバージョンのTLSの「セッションチケット」に導入されました。TLSv1.3以降では、Traffic Managerは、RFC 8446 (TLSv1.3)のセッションチケットのより全般的な定義を追加でサポートします。これは、以前のセッションチケットとセッションIDの両方の機能を果たします。

セッションチケットは、サーバがステートを保存する必要なく、SSLセッションを再開できるメカニズムです。代わりにサーバは(対称)秘密鍵を使用して、セッション情報をセッションチケットに暗号化します。この秘密鍵の当事者でなく媒介者である全てのクライアントおよびネットワークは、チケットの内容を復号化できません。

クライアントに短いセッションIDを与える代わりに、サーバはクライアントに長いセッションチケットを送信します。クライアントはセッションIDではなく、このセッションチケットを保存して、セッションを再開する必要があるときに提示します。チケットを復号化するために使用された秘密鍵がまだ使用可能な場合、サーバはチケットを復号化し、セッション情報を抽出して、クライアントとのセッションを再開します。

このシナリオには以下の利点があります。

- SSLサーバがセッション再開をサポートするためのメモリ要件が軽減されます。
- 複数のSSLサーバがクラスタで連携し、それらがチケット鍵を共有する場合、全てのクラスタメンバー間で全てのセッション情報を共有することなく、1つのサーバで作成されたセッションを、クラスタ内の他のサーバに対して再開できます。

Traffic Managerは、サーバとして動作する場合と、クライアントとして動作する場合の両方で、TLSセッションチケットをサポートします。クライアントとしての動作では、セッションチケットがセッションIDの代わりにクライアントセッションキャッシュに入れられ、そのチケットは、セッションIDの代わりにSSLセッションを再開するために使用されます。

サーバとしての動作では、Traffic Managerがチケットの暗号化に使用されるキーを十分頻繁に変更することで、鍵が解読される可能性を最小限に抑え、鍵の1つが解読された際の損害を軽減する必要があります。そのためには、チケット暗号化で使用される鍵を、Traffic Managerが自動的に生成してローテーションします。

Traffic Managerはそれぞれに一意の識別子を付けて、チケット鍵の履歴データベースを維持します。各セッションチケットはチケット鍵識別子で始まっているため、Traffic Managerがセッションチケットを受信すると、そのセッションチケットの暗号化に使用された鍵を見つけることができます。Traffic Managerがデータベースでチケット鍵を見つけると、セッションチケットを復号化し、セッションを再開します(セッションが有効な場合)。

複数のTraffic Managerがクラスタを構成している場合は、チケット鍵のデータベースが全てのクラスタメンバー間で複製されます。クラスタ内のTraffic Managerはそれぞれ別の鍵を使用して、自分が生成するセッションチケットを暗号化します。また、現在および過去の全ての鍵が、クラスタメンバー間で共有される中央データベースに追加されます。このようにして、クライアントからのSSLセッションチケットを、クラスタ内の任意のTraffic Managerが復号化して再開できます。

Traffic ManagerがSSLサーバとして機能する場合のセッションチケットの動作を設定するには、**System** (システム) > **Global Settings** (グローバル設定) > **SSL Configuration** (SSL設定)をクリックします。以下の表に、利用可能な設定を示します。

設定	説明
ssl!tickets!ticket_key_rotation	チケット鍵が更新されてからの経過時間(秒単位)。以前に使用されていた鍵はデータベースに追加されます。
ssl!tickets!ticket_key_expiry	チケット鍵がデータベースから削除されてからの経過時間(秒単位)。
ssl!tickets!ticket_expiry	セッションチケットが期限切れと見なされてからの合計経過時間(秒単位)。この値よりも古いチケットを使用するセッション再開リクエストがTraffic Managerに提示された場合は拒否されます。

チケット鍵の自動鍵生成およびローテーションを完全に無効にするには、Traffic ManagerのREST APIの「ssl/ticket_keys」リソースを使用します。詳細については、『*Pulse Secure Virtual Traffic Manager: REST API Guide*』を参照してください。以下の点を考慮して、この手順は上級ユーザのみが使用することをお奨めします。

- チケット鍵は、暗号強度の高い乱数発生器で生成する必要があります。サーバのMACアドレスやシリアルナンバーなどの静的情報に由来するものにしないでください。
- チケット鍵の保存は可能な限り短時間で行う必要があります。チケット鍵が解読された場合、その鍵で暗号化されたチケットを使用して行われたコネクションのForward Secrecyも解読されます。特に、チケット鍵は、ログファイルなどに記録すべきではありません。
- チケットキーをネットワークリンク上で送信するときは、ディフィーヘルマンまたは楕円曲線ディフィーヘルマン鍵交換アルゴリズムを使用する暗号など、完全なForward Secrecyを実現する暗号を使用して送信する必要があります。

TLSv1.3では、セッションIDとセッションチケットの両方が、新たなセッション回復のスタイルに置き換えられています(以前の形式よりもセキュリティのプロパティが改善されています)。TLSv1.3セッションチケットは、「データベース検索鍵」(セッションIDに似たもの)または「自己暗号化および自己認証の値」(TLSv1.3よりも前のセッションチケットに似たもの)のいずれかにすることが明示的に許可されています。

このセクションに記載された設定を適用し、セッションを保存またはセッションチケットを送信するか、セッションを再開するかを決めることで、Traffic Managerでは以下のアクションがTLSバージョンに基づいて考慮されます。

TLSv1.2以前	TLSv1.3以降
TLSv1.3よりも前のセッションをセッションキャッシュに保存し、セッションIDをクライアントに送信する	TLSv1.3のセッションをセッションキャッシュに保存し、検索鍵を含むチケットを送信する

TLSv1.2以前	TLSv1.3以降
TLSv1.3よりも前のセッションIDからセッションを検索する	検索鍵を含むチケットからセッションを検索する
TLSv1.3よりも前のセッションチケットを生成/解析する	TLSv1.3の暗号化チケットを生成/解析する TLSv1.3よりも前、および暗号化されたTLSv1.3のセッションチケットに対して、同じチケットの暗号鍵が使用されます

OCSP Staplingキャッシュ

Traffic Managerは、**ssl_ocsp_stapling**が有効化されている仮想サーバによって使用されるサーバ証明書のために、OCSPレスポンスのキャッシュを保持します。

管理UIの**System (システム) > Global Settings (グローバル設定) > SSL Configuration (SSL設定)** セクションで、Traffic ManagerのOCSPレスポンスのキャッシング動作を設定することができます。

- **ssl!ocsp_stapling!mem_size**でキャッシュのサイズを設定します。
- **ssl!ocsp_stapling!default_refresh_interval**は、その先の「nextUpdate」時間を伴うレスポンスがキャッシュ内で利用できない場合に、新たなOCSPリクエストが行われる前にTraffic Managerにどれだけ待機させるかをコントロールするために使用します。
- **ssl!ocsp_stapling!maximum_refresh_interval**は、OCSPリクエスト間の最大経過間隔を定義するために使用します。有効なOCSPレスポンスを受信すると、OCSPレスポンスの「nextUpdate」時間の方に早く到達しない限り、ここで定義された秒数後に次の更新が行われます。
- Traffic Managerには、**ssl!ocsp_stapling!verify_response**の署名検証オプションがあります。このオプションをYes (はい)に設定すると、Traffic Managerは、OCSPレスポンスを使用してStaplingを行う前に、強制的にOCSPレスポンスの署名および有効期間の認証を行います。認証に失敗すると、Traffic ManagerはそのレスポンスをOCSP Staplingに決して使用しません。

- OCSPレスポンスには、OCSPレスポンス内の2つのフィールド「thisUpdate」および「nextUpdate」で定義された有効期間があります。**ssl!ocsp_stapling!verify_response**が有効化されているとき、この期間外のレスポンスはキャッシュされないため、Staplingされません。環境によっては、クロックスキューが関係していることがあります。そのためにローカルの現在時刻がOCSPレスポンスの「thisUpdate」フィールドよりも早く表示されることがあり、誤って有効性の認証に失敗する原因になります。Traffic Managerに、レスポンスが有効であると見なし続けさせるためには、**ssl!ocsp_stapling!time_tolerance**設定を使用して、この間隔外の許容時間マージン(秒単位)を定義します。

SSL暗号化

プールは、バックエンドSSLサーバにトラフィックを送信する前に、SSL暗号化を実行することができます。SSL暗号化が有効化されている場合、暗号化には、それを実行するTraffic ManagerのCPUリソースが追加で使用されます。

プールに対してSSL暗号化を有効化するには、**Pools (プール) > Edit (編集) > SSL Settings (SSL設定)**に移動します。

以下のオプションを設定することができます。

設定	説明
ssl_encrypt	バックエンドノードへのSSL暗号化を有効化するには、「Yes (はい)」に設定します。
ssl_send_close_alerts	ソケット切断を開始するときにSSL/TLS「切断アラート」を送信するには、「Yes (はい)」に設定します。
ssl_support_<version>	このプールがバックエンドノードへのコネクションにSSLまたはTLSの専用バージョンを使用できるようにするには、「Enabled (有効)」に設定します。Traffic Managerに、 System (システム) > Global Settings (グローバル設定) > SSL Configuration (SSL設定) で設定されている同等のグローバル値を強制的に使用させるには、 Use the global setting (グローバル設定を使用) を選択します。
ssl_cipher_suites	バックエンドノードとのセキュア通信に利用できる暗号のリストを指定します。スペース、カンマ、またはコロンで区切られたリストを(望ましい順に)使用します。Traffic Managerに、 System (システム) > Global Settings (グローバル設定) > SSL Configuration (SSL設定) で設定されている同等のグローバル値を強制的に使用させるには、リストを空白のままにします。
ssl_client_auth	バックエンドサーバがクライアント証明書の認証をリクエストする場合、Traffic Managerがローカルのクライアント証明書カタログから適切な証明書を選択するよう設定するには、この設定を有効化します。Traffic Managerにクライアント証明書を設定するには、 Catalogs (カタログ) > SSL Catalogs (SSL カタログ) > SSL Client Certificates catalog (SSLクライアント証明書カタログ) を表示します。

設定	説明
	サーバは、サーバが信頼する認証局のリストをクライアントに送信します。デフォルトでは、Traffic Managerはサーバが信頼する認証局の1つによって署名されたと判別した最初のクライアント証明書を選択します。どの証明書を使用するかを明示的にコントロールするには、「ssl_fixed_client_certificate」を設定します。 サーバが空のCAリストを送信した場合、Traffic Managerは「ssl_fixed_client_certificate」が設定されていない限りクライアント証明書を選択しません。
ssl_fixed_client_certificate	このプール内のサーバがクライアント証明書をリクエストした際に毎回はTraffic Managerによって使用される、SSLクライアント証明書カタログ内のエントリ名です。この場合、リクエスト内の信頼済みのCAの全てのリストは無視されます。 この設定は、「ssl_client_auth」が有効である場合にのみ適用されます。
ssl_server_name	Traffic ManagerにTLS 1.0「server_name」拡張を試行させるには、「Yes (はい)」に設定します。これにより、バックエンドノードが適切な証明書を提示しやすくなります。この設定を有効化すると、Traffic Managerは強制的にTLS 1.0以上を使用します。
ssl_strict_verify	中間者攻撃やサーバスプーフィングから保護するために、Traffic Managerは、バックエンドサーバが使用するSSL証明書をカタログ内のCAと照合して検証することができます。証明書の有効期限が切れている場合、証明書の一般名称がノードのIPアドレス、ホスト名、または「ssl_common_name_match」にリストされているいずれかの名前と一致しない場合、カタログ内のCAがその証明書に署名していない場合に、サーバへのコネクションを拒否するようにTraffic Managerに指示するには、「Yes (はい)」に設定します。
ssl_common_name_match	サーバによって提示される証明書で、Traffic Managerが一般名称として受け入れるホスト名またはIPアドレスのリストです。

Update (更新)をクリックして変更を保存します。

SSL転送におけるIPアドレスの維持

Traffic Managerは、HTTPリクエストを転送するときに、ダウンストリームのサーバがリクエストの正しい送信元IPアドレスを特定できるように、オプションでX-Forwarded-Forというヘッダを挿入することができます。Traffic ManagerがHTTPSリクエストを復号化する場合は、後でリクエストを再度暗号化するときでも、X-Forwarded-Forヘッダをリクエストに挿入できます。

ただし、復号化と再暗号化を行わずにSSLリクエストを転送する場合、Traffic Managerは、リクエスト内部のデータを変更することができません。この設定は、復号化のためにコネクションをTraffic Managerシステムのクラスタに負荷分散するループバック仮想サーバで使用されます。この場合、プールで`ssl_enhance`設定を使用することにより、独自のヘッダをSSLコネクションに追加し、ここに送信元IPおよびポート、宛先IPおよびポートなど維持されない重要なコネクションデータを保持することができます。

`ssl_enhance`が設定された「拡張」SSLコネクションを受信するTraffic Managerシステムには、仮想サーバのSSL復号化設定の`ssl_trust_magic`を設定する必要があります。この設定により、Traffic Managerは、独自のヘッダを取り出し、送信元IPおよびポート、宛先IPおよびポートなどの正しいコネクションデータを認識できるようになります。

SSL暗号処理専用デバイスの使用

Traffic Managerソフトウェア製品は、Thales e-Security社のnShield Connectなど、RSA Security Inc.のPKCS #11 Cryptographic Token Interface (Cryptoki)に準拠したSSLハードウェアをサポートしています。

全てのTraffic Managerは、Microsoft Azure Key Vaultサービスをサポートしています。Azure Key Vaultは、従来のハードウェアセキュリティモジュールに代わるクラウドベースサービスとして提供されています。Azure Key Vaultsでは、以下の条件に従ってください。

- Traffic Managerは、それ自体によって作成された鍵、またはAzure Key Vaultに移行された有効な既存のRSA鍵の使用をサポートしています。
- Traffic Managerは、「RSA-HSM」タイプの鍵をサポートしています。このタイプの鍵は、Azure Key Vaultサービスでセキュアなハードウェアに保存されます。
- RSA-HSM鍵を作成し、使用するには、Azure Key Vaultの「Premium」サービスレベルを使用する必要があります。利用可能なサービスレベルの詳細と価格については、<http://azure.microsoft.com/>を参照してください。
- 他のセキュアハードウェアのサポートと同様に、Traffic Managerは、SSL復号化(仮想サーバ)またはSSL暗号化(プール)用としてAzure Key Vaultに保存された鍵を使用できます。Traffic Managerは、SSHまたはDNSSECなどの他の目的のために、保存された鍵を使用することはできません。
- Traffic ManagerがFIPSモードを使用するように設定されている場合([Traffic ManagerのFIPS認証](#)を参照)、Azure Key Vault REST APIとの通信もFIPSモードになります。
- Azure Key Vaultを使用するには、Traffic ManagerのDNSで、Azure Key VaultサーバとMicrosoftのログインサーバの両方のホスト名を解決する必要があります。Traffic Managerは、ポート443で送信TCPコネクションを確立できる必要もあります。Traffic Managerシステムがファイアウォールによる制限を受けている場合は、この通信チャンネルが許可されるようにファイアウォールを設定する必要があります。



Azure Key Vaultの設定と管理については、
<http://azure.microsoft.com/en-us/documentation/services/key-vault/>を参照してください。

HSMデバイスまたはAzure Key Vaultサービスを使用すると、Traffic ManagerシステムのCPUから接続先のSSL暗号処理専用デバイスに、SSL演算(RSA秘密鍵の復号化)がオフロードされます。ハードウェアで鍵管理を行うPKCS#11デバイスもあり、この場合、秘密鍵はハードウェアデバイスにセキュアに保存され、適切な認証がなければ直接アクセスすることはできません。

SSL暗号処理専用デバイスまたはサービスで実行されているRSA暗号動作はTraffic ManagerのFIPS 140-2 Cryptographic Boundaryの対象ではないため、暗号処理専用デバイスまたはサービスがFIPS 140-2に十分に適合していることを、必要に応じて個別に確認する必要があります。



仮想アプライアンスなどTraffic Manager製品によっては、サポートされるハードウェアとサービスが異なるため、以下のオプションの一部のみしか表示されない場合があります。



SSLデバイスを使用しても、Traffic Managerシステムの全体的な性能が向上しない場合があります。RSA演算はメインCPUコアからオフロードされますが、デバイスとの通信オーバーヘッドも無視できません。Traffic Managerは、は、数千または数万のSSL演算を汎用CPUで実行することができます。多くのSSLデバイスを使用する最大のメリットは、秘密鍵をセキュアに保存できることです。

SSLデバイスを使用するためのTraffic Managerの設定

SSLデバイスまたはサービスを使用するようにTraffic Managerを設定するには、**System (システム) > Global settings (グローバル設定)**をクリックし、**SSL Hardware Support (SSLハードウェアサポート)**をクリックします。**ssld!library**で必要なSSLデバイスを選択します。

▼ SSL Hardware Support

These settings control how cryptographic hardware devices (such as PCI cards or network devices) are used.

The type of SSL hardware to use. The drivers for the SSL hardware should be installed and accessible to the traffic manager software.

ssld!library:

- ☒ None
- ☐ Microsoft Azure Key Vault
- ☐ PKCS#11 (e.g. nCipher NetHSM) ...

Location of PKCS#11 Library:

ssld!driver!pkcs11_lib:

The User PIN for the PKCS#11 token:

ssld!driver!pkcs11_user_pin:

Security token type:

ssld!driver!pkcs11_slot_type: Operator Card Set

Security token label (required when multiple tokens available):

ssld!driver!pkcs11_slot_desc:

The Microsoft Azure Key Vault can be used to provide cryptographic services.



Connect to Microsoft Azure Key Vault

Whether or not the SSL hardware is an "accelerator" (faster than software). By default the traffic manager will only use the SSL hardware if a key requires it (i.e. the key is stored on secure hardware and the traffic manager only has a placeholder/identifier key). With this option enabled, your traffic manager will instead try to use hardware for all SSL decrypts.

ssld!accel: ☐ Yes ☒ No Default: No

The number of consecutive failures from the SSL hardware that will be tolerated before the traffic manager assumes its session with the device is invalid and tries to log in again. This is necessary when the device reboots following a power failure.

ssld!failure_count: 5 Default: 5

以下の表に、利用可能な設定を示します。

設定	説明
ssld!library	Azure Key Vaultサービスを使用することをTraffic Managerに指示するには、「Microsoft Azure Key Vault」に設定します。Azure Key Vaultに接続するには、「Connect to Microsoft Azure Key Vault (Microsoft Azure Key Vaultに接続)」をクリックします。 Thales e-Security nShield Connectなどの汎用PKCS準拠ハードウェアの使用をTraffic Managerに指示するには、「PKCS#11」に設定します。 PKCS#11の追加設定は以下のとおりです。 ssld!driver!pkcs11_lib: Traffic ManagerでPKCS#11インターフェースライブラリを検索する必要があるシステムロケーション。Traffic Managerで標準のシステムロケーションを検索するには、このフィールドを空白のままにします。

設定	説明
	ssld!driver!pkcs11_user_pin: SSLハードウェアでユーザセッションを作成するために必要なPINです。 ssld!driver!pkcs11_slot_type: Traffic ManagerでSSL秘密鍵の保護に使用されるセキュリティトークンの形態です。 Operator Card Set: Traffic Managerに接続される適切なカードリーダー付きの物理的なスマートカードです。このスマートカードには、オプションのユーザPINが必要になる場合があります。 Soft Card: 物理的なスマートカードと同様に動作するディスク上のファイルです。このファイルにはユーザPINが必要です。 Module Protected: スマートカードやユーザPINを使用せずに単独で鍵を保護する暗号モジュールです。 ssld!driver!pkcs11_slot_desc: (「Operator Card Set」と「Soft Card」のトークンタイプのみ)使用するカードの識別ラベルまたは識別名を指定するには、この設定を使用します。
ssld!accel	全ての復号化動作で、このページで定義したSSLデバイスを使用するかどうかを指定します。全ての動作でSSLデバイスを使用するようにTraffic Managerに指定するには、「Yes (はい)」に設定します。「No (いいえ)」に設定すると、Traffic Managerは、秘密鍵がSSLハードウェアにセキュアに保存されている場合のみそのハードウェアを使用します。
ssld!failure_count	(リブート、停電などの場合に)Traffic Managerがセッションを無効と見なして再度ログインを試行するまでに実行できる連続試行回数です。

SSLデバイスの正常な動作の検証

Diagnose (診断)ページには、Traffic ManagerがSSLデバイスと通信可能かどうかについてのチェック結果が表示されます。全てのエラーまたは警告が示されます。

ハードウェアによる鍵管理をサポートしているSSLデバイスを使用するようにTraffic Managerを設定した場合は、以下の操作を実行できます。

- Traffic Managerを使用して、そのデバイスに新しい秘密鍵(およびそれに対応する公開証明書)を作成できます。新しい自己署名サーバまたはクライアントSSL証明書を作成するには、[自己署名SSL証明書の新規作成](#)の手順に従います。この手順では、鍵管理デバイスが設定されているときに表示される**Create private key on hardware** (ハードウェアで秘密鍵作成)オプションを選択してください。
- ハードウェアからTraffic Managerに手動で秘密鍵と公開証明書をインストールできます。ハードウェアデバイスの手順に従って、符号化された秘密鍵と公開証明書を抽出します。次に、関連するSSL Server Certificates (SSLサーバ証明書)カタログとSSL Client Certificates (SSLクライアント証明書)カタログの**Import Certificate** (証明書インポート)ツールを使用して、抽出した秘密鍵と公開証明書をTraffic Managerにインストールします。
- Traffic Managerから新たに接続したSSLデバイスに、既存の秘密鍵を移行します。鍵を移行するには、関連する証明書編集ページの「Migrate certificate to Hardware Security Module (ハードウェアセキュリティモジュールへの証明書の移行)」セクションを使用します。

注記

攻撃者がすでに秘密鍵のコピーを入手している場合、Traffic ManagerからSSLデバイスに秘密鍵を移行しても、その保護は保証されません。セキュリティを最大限にするために、SSLデバイス自体で生成した鍵を使用することをお奨めします。

Traffic Managerによって、ハードウェアで管理される鍵にアクセスして使用できることが検証されます。問題が発生した場合は、エラーが**System** (システム) > **Diagnose** (診断)ページと**Event log** (イベントログ)に表示されます。



Diagnose (診断)ページと**Event log** (イベントログ)で、クライアント証明書とそれに対応する秘密鍵に使用不可のマークが付けられることがあります。これらの証明書および鍵を使用するプールに問題があることを示すマークが付くことはありません。これは、Traffic Managerが、クライアント証明書をリクエストするバックエンドSSLノードに接続するまで、どのクライアント証明書を求められるかが分からないためです。

Connect to Microsoft Azure Key Vault (Microsoft Azure Key Vaultに接続)ウィザードの使用

Azure Key Vaultに接続するには、System (システム) > Global Settings (グローバル設定) > SSL Hardware Support (SSLハードウェアサポート)で、**Connect to Microsoft Azure Key Vault** (Microsoft Azure Key Vaultに接続)をクリックします。

ただし、このウィザードをアクティブにする前に、以下の情報を入手してください。

- **Azure Key VaultのURL:** 指定されたAzure Key VaultのURL(「https://dev.vault.azure.net」など)。
- **クライアントID:** Azure Key Vaultでのユーザアカウントの識別子(ID)。
- **クライアントのパスワード:** ユーザアカウントのパスワードを表す英数字の文字列。

- **信頼された認証局:** (オプション)「中間者」攻撃からAzure Key Vaultへの接続を保護するために、Azure Key Vault REST APIによって使用されるSSL証明書のIDを検証するようにTraffic Managerに指示することができます。MicrosoftのSSL証明書に署名した信頼された認証局の情報を入手し、ウィザードを実行する前にTraffic ManagerのCAカタログに保存します。

この情報の入手については、サポート窓口にお問い合わせるか、以下の場所にあるMicrosoft Azure Key Vaultのドキュメントを参照してください。<http://azure.microsoft.com/en-us/documentation/services/key-vault/>

他のTraffic Managerウィザードと同様に、「Connect to Microsoft Azure Key Vault (Microsoft Azure Key Vaultに接続)」ウィザードでは、一連のステップが実行されます。各ステップ間を移動するための、**Back** (戻る) ボタンと**Next** (次へ) ボタンが表示されます。

1. 開始します。
2. Azure Key VaultのURLを入力します。

Connect to a Microsoft Azure Key Vault, step 2 of 5

2. Enter Vault URL

Please enter the URL of the vault to use.

Vault URL:

3. クライアントのクレデンシャルを入力します。

Connect to a Microsoft Azure Key Vault, step 3 of 5

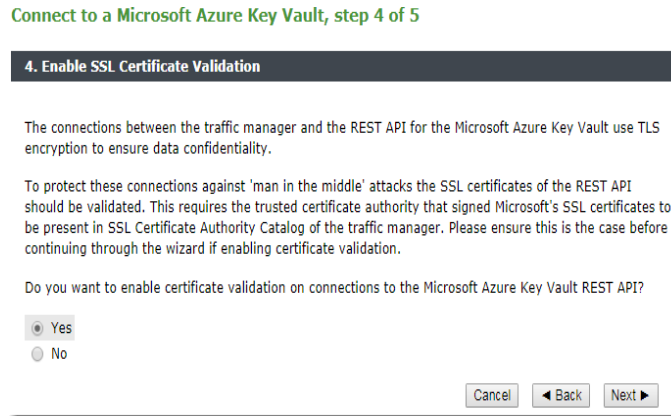
3. Enter Vault Credentials

Please enter the new client authentication settings for the vault.

Client ID:

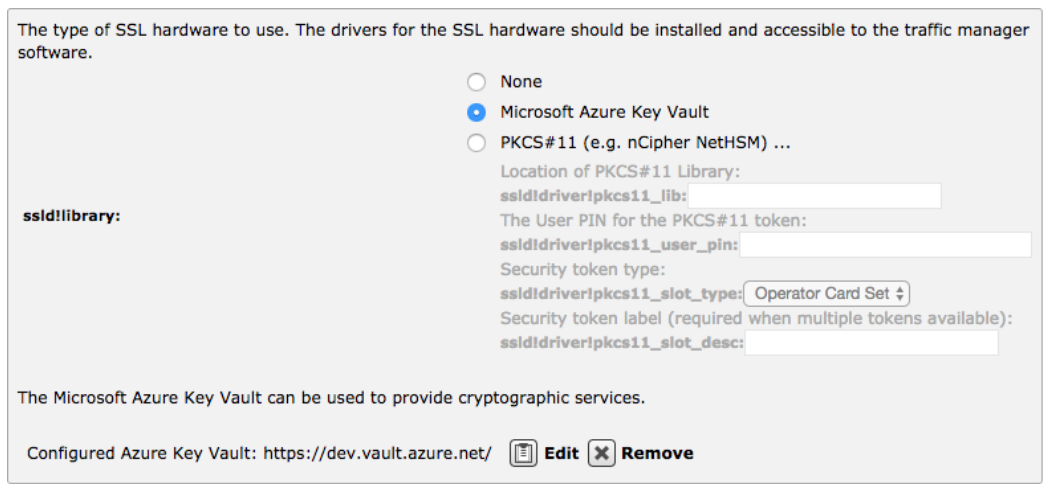
Client Secret:

4. 「Yes (はい)」を選択して、Azure Key Vault REST APIのSSL証明書を検証するようにTraffic Managerに指示するか、「No (いいえ)」を選択して検証を無効にします。



5. **Finish** (完了)をクリックし、表示されている設定を使用してAzure Key Vaultに接続します。

接続に成功した場合、SSL Hardware Support (SSLハードウェアサポート)セクションが更新され、アクティブなAzure Key Vaultコネクションが表示されます。



ウィザードに戻って別の設定を行うには、**Edit** (編集)をクリックします。アクティブなAzure Key VaultからTraffic Managerを切断するには、**Remove** (削除)をクリックします。

Azure Key Vaultに関連する接続失敗、警告、およびエラーは、**Diagnose** (診断ページ)と**イベントログ**でレポートされます。

セキュアなデバイスに保存された鍵と証明書の識別

セキュアなハードウェアまたはAzure Key Vaultに正常に接続した後、他の鍵ペアと同様に、デバイスの鍵および証明書ファイルを使用できます(証明書の編集を含みます)。デバイスが正常に動作している場合、Traffic Managerは秘密鍵が必要な全ての暗号化演算をデバイスに委任します。管理UIで、鍵がセキュアなハードウェアまたはAzure Key Vaultに保存されたことが以下のように示されます。

Which SSL certificate should this virtual server use?

certificate:	Name	Site (CN)	Signed by (Issuer)	Expires
	 example	example.cam.zeus.com	(self-signed)	02 May 2007
	 (no certificate specified)			

 = private key stored on secure hardware

ヘルスマニタリング

本章では、Traffic Managerのヘルスマニタリング機能を説明します。この機能を使用すると、正しい動作に関してサーバノードを監視し、エラーを検出した際には、アラートを発生させ、故障ノードを迂回することができます。

どのノードをモニタリングするか

Traffic Managerは、パッシブモニタリング(使用されているノードのステータスを確認)とヘルスマニタリング(定期的にノードに対して実行される追加テスト)の2通りの方法でノードが正常に動作しているかどうかをモニタリングします。

パッシブモニタリングテストは、アクティブに使用されているノードに対してのみ実行されます。

ヘルスマニタは、プール内の全てのノードに対して実行されます。実行対象は、以下の方法で使用されているプールのみです。

- 仮想サーバのデフォルトプールとして設定されているプール。
- TrafficScriptまたはRuleBuilderルールで`pool.select()`または`pool.use()`関数によって明示的に名前が参照されているプール
- 使用中のプールの障害プールとして設定されているプール。

Traffic Managerの設定によりこの方法で参照されていないプールは、「使用されていない」と見なされ、ヘルスマニタでモニタリングされず、ノードの障害は検出されません。



プールをTrafficScriptルールで使用するときは、明示的に名前が参照する必要があります。デフォルトでは、Traffic Managerで変数(`pool.use( $name );`)またはその他の方法によって参照することはできません。

trafficscript!variable_pool_use設定(**System** (システム) > **Global Settings**(グローバル設定))を有効化すると、プール名を変数で指定することができます。この設定が有効化されている場合、Traffic Managerは、明らかに使用中のプールだけでなく、設定されている全てのプールに対してヘルスマニタを実行します。

複数プールでのノードの使用

複数のプールで同じノード(IPアドレスおよびポート)を参照することができます。1つのプールで障害が発生したノードは、そのプールでは負荷分散またはセッション維持の決定に使用されません。ただし、他のプールに割り当てられたパッシブモニタおよびヘルスマニタが障害を報告するまで、これらのプールでは使用される場合があります。

例

Webアプリケーションが4台のバックエンドサーバ(ノード)で構成されているとします。各ノードでは、同じ動的コンテンツと静的コンテンツがホストされています。あるノードの動的コンテンツで障害が発生したときに(Java Servletがクラッシュするなど)、そのノードの他の静的コンテンツは引き続き使用したい場合があります。

- それぞれに4台の全てのノードが含まれる「Dynamic」および「Static」という2つのプールを作成します。
- 動的コンテンツ(Java Servlet、PHP、ASPファイルなど)には「Dynamic」プールを使用し、その他全てのコンテンツには「Static」プールを使用するルールを作成します。
- 「Dynamic」プール内のあるノードで障害が発生し、有効なHTTPレスポンスを送信できなくなった場合、そのプールで使用されているパッシブモニタリングにより、そのノードで障害が発生したと判断されます。以降、そのノードには「Dynamic」プールからトラフィックが送信されなくなります。ただし、そのノードが「Static」プールからのリクエストに対して有効なHTTPレスポンスを返している限り、「Static」プールからは、そのノードに引き続きトラフィックが送信されます。
- さらにきめ細かくエラーを検出するために、各プールに異なるヘルスマニタを割り当てることができます。例えば、「Dynamic」プールのヘルスマニタから擬似的なPHPまたはASPリクエストを送信できます。これらのモニタが有効なレスポンスを受信できなかった場合、そのノードは「Dynamic」プールで障害が発生したと見なされますが、他のプールでは障害はないと見なされます。

パッシブヘルスモニタリング

プールは、リクエストをノードに送信するたびに一連のチェックを実行します。このプロセスを「パッシブモニタリング」と言います。

- Traffic Managerがノードに接続しようとするときに、コネクションが拒否されるか、**max_connect_time**設定の時間内(デフォルトでは4秒)にコネクションが確立されなかった場合、そのリクエストは失敗したと見なされます。
- Traffic Managerはコネクションによってデータを書き込むときに、処理の完了前にコネクションが切断されるか、**max_reply_time**設定の時間内(デフォルトでは30秒)にレスポンスの最初のデータが受信されない場合、リクエストは失敗したと見なされます。
- SSLのみ: ノードとのSSLハンドシェイクに失敗した場合、リクエストは失敗したと見なされます。



max_connect_timeおよび**max_reply_time**設定は、プール設定の**Protocol Settings** (プロトコル設定)ページのプロパティです。詳細については、[プロトコル設定](#)を参照してください。

失敗したリクエストの再試行

これらのチェックが失敗した場合、プールは、稼働中の別のノードに対してリクエストを試行し、リクエストを破棄する前にプール内の全てのノードを試すことがあります。実際の動作は、リクエストの「べき等性 (idempotent)」の状況によって決まります。

デフォルトでは、リクエストはべき等であると見なされます。すなわち、悪影響が生じることなく複数回安全に再試行することができます。例外は、genericタイプのプロトコル(「Generic Client First」、「Generic Server First」、「Generic Streaming」)のいずれかを使用して仮想サーバを通じて受信されるリクエストです。デフォルトでべき等になるためには、障害を測定して再試行を実行できるように、リクエストへのエンドポイントをまず定義¹する必要があります。

RFC 2616では、一部のHTTPリクエストが非べき等²として定義されています(トランザクションが発生し、またはサーバのステータスが変更される可能性がある場合など)。Traffic Managerはこれらの勧告に従い、HTTP GET、HEAD、PUT、DELETE、OPTIONS、およびTRACEメソッドをべき等として扱い、その他全てのリクエストを非べき等と見なします。

要約:

- **べき等(悪影響なし):** Traffic Managerは、動作していると考えられる他のノードに対してこれらのリクエストを再試行し、リクエストを破棄する前に、プール内の全てのノードに対してリクエストを試行することができます。
- **非べき等(悪影響あり):** Traffic Managerは、障害が発生したノードに対するTCPコネクションの確立に失敗した場合のみ、非べき等リクエストを再試行します。



Traffic Managerは、TCPコネクションを確立すると、直ちにそのコネクションによってリクエストデータを書き込みます。送信先のノードがリクエストデータを受信し、その処理を開始したかどうかを特定することはできません。そのため、非べき等リクエストは、最初の段階でコネクションを確立できなかった場合のみ再試行されます。

Traffic Managerによるべき等/非べき等の判断を変更するには、TrafficScript関数の`request.setIdempotent()`および`http.setIdempotent()`を使用して、特定のリクエストを安全に再試行できると見なすようTraffic Managerに指示を出します。

¹ これは、`request.endsWith()`または`request.endsAt()`のような関数を使用して、TrafficScriptで実現されます。

² <http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>.

ノード障害

Traffic Managerは、ノードへのコネクションが、中間トランザクションが成功することなく

node_connection_attempts設定の回数(デフォルトでは3回)だけ連続して失敗すると、そのノードで障害が発生したと推測します。

障害が発生したと見なされたノードは、少なくとも**node_fail_time**設定の時間(デフォルトでは60秒)は使用されません。その後、復旧したかどうかを判断するため、定期的にそのノードにライブサービストラフィックからべき等リクエストの送信を試みることで、暫定的に使用されます。リクエストが失敗する場合は、他の動作ノードに対してリトライできます。プール内の全てのノードで障害が発生した場合は、**node_fail_time**設定に関わらず、Traffic Managerは復旧したノードに直ちにトラフィックを送信します。



node_connection_attemptsおよび**node_fail_time**を設定するには、**Pools** (プール) > **Edit** (編集) > **Protocol Settings** (プロトコル設定) ページを使用します。

ノード障害の原因の詳細を知るには、**Virtual Servers** (仮想サーバ) > **Edit** (編集) > **Error Logging** (エラーロギング) の **log!server_connection_failures** を有効にします。この機能を使用すると、ノード障害の実際の理由に関してより有益なログ情報が得られます。詳細については、[エラーの処理](#) を参照してください。



パッシブモニタリングにより障害ありとマークされていたノードは、ライブサービストラフィックに反応する場合のみ、復旧できます。これはアクティブなヘルスマニタで復旧できません。

パッシブモニタリングの有効化と無効化

パッシブモニタリングはデフォルトで使用されますが、プール設定の **Monitors** (モニタ) セクションの **passive_monitoring** 設定を使用することによりプール単位で無効化することができます。この設定が無効化されている場合、適切なヘルスマニタが設定されていることを確認してください。設定されていないと、リクエストの失敗が検出されず、再試行されません。

ヘルスマニタの概要

パッシブモニタリングによる推測に加え、定期的にテストを実行し、各ノードが正常に動作しているかどうかを検証する明示的なヘルスマニタをプールに設定することができます。Traffic Managerはヘルスマニタを使用することで、トラフィックが処理されていないときでも障害を検出できます。

テストは、設定された間隔でプール内の各ノードに対して実行されます。頻繁にテストで不合格になったノードには使用不可のマークが付けられ、Traffic Managerからトラフィックが送信されなくなります。プール全体のモニタでは単一のマシンのみがテストされ、その結果からプール全体が使用不可であることを示すことができます。

i 障害ありとマークされていたノードを復旧できるのは、それを障害ありとマークしたモニタだけです。

モニタはモニタカタログに保持され、任意のプールに適用することができます。各モニタでは、各ノードに対するPingの実行などの単純なテストから、該当するポートが開いていてノードがホームページなど指定されたデータを処理できるかどうかを確認する高度なテストまで、特定のテストが実行されます。

モニタカタログ

i 一部のモニタタイプでは正規表現が使用されます。続行する前に、[正規表現の使用](#)のセキュリティに関する注意事項を確認しておいてください。

モニタの設定

モニタカタログにアクセスするには、管理UIで**Catalogs** (カタログ) > **Monitors** (モニタ)をクリックします。

Traffic Managerには、多数のモニタがプリセットされています。これには、一般的なTCPのClient FirstまたはServer Firstプロトコルを処理するモニタと、プロトコル固有のモニタがあります。例えば、内蔵のFull HTTPモニタは、指定されたページをWebサーバにリクエストし、レスポンス内で適切なステータスコードを検索します。

全てのモニタについて、以下の設定を行うことができます。

設定	説明
delay	モニタは、各ノードに対して繰り返し実行されます。モニタリングが完了するかタイムアウトになると、Traffic Managerは、ここに設定された時間だけ一時停止してから再度モニタリングを試行します。
timeout	モニタがこのタイムアウト時間内に完了できなかった場合、中断され、失敗したと判断されます。
failures	一時的なネットワーク異常など繰り返されない予測外の原因により、モニタが失敗することがあります。この設定に指定された回数だけ連続してモニタリングに失敗すると、モニタリング対象ノードで障害が発生したと判断されます。
verbose	この設定により、エラーログへの詳細モニタロギングが有効化されます。モニタのテストには役立ちますが、エラーログファイルがいっぱいになる可能性があるため、商用システムでの使用や長時間の使用は避けてください。

内蔵モニタを編集するには、モニタの名前をクリックし、タイミング、失敗条件、およびロギングと、その他のモニタ固有の設定(リクエストするWebページなど)を変更します。**Update** (更新)をクリックして変更内容を適用します。

新たな名前で内蔵モニタのコピーを作成するには、**Copy Monitor** (モニタのコピー)をクリックします。

内蔵ヘルスモニタ

Traffic Managerのモニタカタログには、多数のモニタがあらかじめ定義されています。これらの内蔵モニタは、様々なプロトコルについて標準テストを実行します。

基本モニタ

基本モニタは、ノードに対して単純なテストを実行します。以下に例を挙げます。

- **Ping:** Pingモニタは、各ノードに対してPingリクエストを実行します。レスポンスが受信されない場合、失敗と判断されます。
- **Connect:** ノードへのTCPコネクションを実行します。サーバがポートで監視してトラフィックを受け入れていることの確認を目的としています。このモニタでは、一部のプログラム(eximなど)でエラーが出力されることがあります。この場合は、よりプロトコル固有のモニタを使用してください。

高機能モニタ

高機能モニタは、ノードに対してより複雑なリクエストとレスポンスのテストを実行します。他のモニタのツールとして使用され、SSLを使用して接続してリクエストおよびレスポンスを転送することができます。以下に例を挙げます。

- **Client First:** これは、TCPトランザクションモニタに基づくモニタです。ノードにデータを送信し、レスポンスを指定の正規表現と照合します。このモニタを使用するには、送信データ(**write_string**)を設定する必要があります。
- **Server First:** このモニタでは、最初にサーバがデータを書き込むサービスをチェックすることができます。ノードに接続し、サーバが何らかのデータを返すことを確認します。サーバが特定データを返していることを確認するよう簡単に変更できます。
- **Full HTTP / HTTPS:** このモニタでは、HTTPまたはHTTPSリクエストを実行できます。ホストヘッダ、URL、および認可パラメータを全て設定できます。

レスポンスの戻りコードは、**status_regex**の値と照合されます。例えば、200の戻りコードが予期される場合は、**status_regex**を「^200\$」に設定します。または、「^[234][0-9][0-9]\$」を使用して、200～499のエラーを受け入れます。

レスポンスボディはオプション設定の**body_regex**と照合されます。

SSLモニタが割り当てられたプールで**ssl_server_name**が有効化されている場合、モニタは、ノード名とともにTLS拡張のサーバ名の指定(SNI)をバックエンドに送信します。

プロトコル固有のモニタ

プロトコル固有のモニタは、ノードに対して特定のテストを実行します。これには以下のようなアプリケーションがあります。

- **DNS:** これはプログラムモニタの例です。ノードに対してDNSリクエストを行い、返されたアドレスを検証するプログラム(ZEUSHOME/zxtm/conf/scripts/dns.pl)を実行します。
- **FTP:** このモニタは、TCPトランザクションモニタに基づきます。単純なFTPコネクションを確立し、サーバが正常にレスポンスを返していることを確認します。このとき、サーバからバナーが返されることのみを確認し、認証は行いません。
- **POP:** このモニタは、TCPトランザクションモニタに基づきます。ノードに対して初期コネクションを確立し、正しいPOPバナーが返されることを確認します。
- **SMTP:** これはPOPモニタと同じですが、SMTPサーババナーが返されることを確認します。
- **シンプルHTTPおよびHTTPS:** これらのモニタは、各ノードにホームページをリクエストします。レスポンスコードが2xx、3xx、または4xxであることを確認し、これらのレスポンスコードが返されない場合は、失敗を報告します。
- **RTSP:** このモニタは、サーバにRTSP DESCRIBEリクエストを送信します。レスポンスコードが2xx、3xx、または4xxであることを確認し、これらのコードが返されない場合は、失敗を報告します。一部のRTSPサーバでは、このようなDESCRIBEリクエストにパスが存在していることが必要になります。この場合は、**rtsp_path**設定にパスを入力する必要があります。RTSPのプロトコル固有の追加設定は以下のとおりです。
 - **rtsp_path:** DESCRIBEリクエストで使われるパスです。これは、ストリーミングされるマルチメディアファイルのパスです。
 - **rtsp_status_regex:** RTSPレスポンスが一致する必要があるステータスコードの正規表現です。
 - **rtsp_body_regex:** RTSPレスポンスボディが一致する必要がある正規表現です。

- **SIP (UDP):** このモニタは、Max-Forwardsを0に設定してサーバにSIP OPTIONSリクエストを送信します。レスポンスコードが2xx、3xx、または4xxであることを確認し、これらのコードが返されない場合は、失敗を報告します。
- **sip_status_regex:** レスポンスが一致する必要があるステータスコードの正規表現です。
- **sip_body_regex:** SIPレスポンスボディが一致する必要がある正規表現です。
- **udp_accept_all:** このモニタが任意のIPおよびポートからレスポンスを受け入れるかどうかを指定します。
- **SIP (TCP):** このモニタは、SIP-UDPモニタと同じリクエストを使用しますが、そのリクエストをTCPコネクションで送信します。

カスタムヘルスモニタ

Monitors Catalog (モニタカタログ)メインページには、新たなモニタを作成することのできる「Create New Monitor」(新モニタ作成)セクションがあります。基礎となるモニタタイプとして、以下のオプションがあります。

モニタタイプ	説明
TCP Transaction	ターゲットマシンとの間でTCPトランザクションを実行し、オプションのデータ文字列をコネクションに書き込みます。指定した正規表現をレスポンス内で検索することができます。
Ping	指定した間隔でターゲットマシンに対してPingを実行します。
External Program	指定したTraffic Managerマシン上のファイルパスにある外部プログラムを実行します。2つの内部引数 port および ipaddr と、その他指定したデータを送信します。終了コード0は成功に分類され、0以外のコードまたはタイムアウトは失敗と見なされます。 ZEUSHOME/zxtm/conf/scriptsで、カスタムモニタの実行可能ファイルの例を参照することができます。
HTTP	オプションでSSLを使用して、ホストヘッダや使用URLパスなど指定されたパラメータとともに、HTTPリクエストをターゲットサーバに送信します。レスポンスコード内でステータスコードの正規表現を検索し、正規表現をレスポンスボディデータと照合します。

モニタタイプ	説明
TCP Connect	ターゲットマシンとのTCPコネクションを確立し、適切なポートが開いていることを確認します。
RTSP	指定された <code>rtsp_path</code> パラメータを使用して、ターゲットサーバにRTSP DESCRIBEリクエストを送信します。レスポンス内でステータスの正規表現を検索し、正規表現をレスポンスボディデータと照合します。
SIP	指定された転送プロトコルを使用して、ターゲットサーバにSIP OPTIONS リクエストを送信します。受信したレスポンスコード内でステータスの正規表現を、またボディデータ内でボディの正規表現を検索します。これらのいずれかが一致しない場合、失敗が報告されます。

モニタの基本設定を選択したら、**Add Monitor** (モニタ追加)をクリックします。モニタの名前をクリックすると、そのモニタを編集できます。2

「TCPトランザクション」または「HTTP」モニタテンプレートを使用して、高度なカスタムモニタを作成することができます。例えば、2つのHTTPベースのモニタを組み合わせて、Webサイトのセキュアな部分をテストし、以下のことを検証できます。

- ユーザ名とパスワードが有効なリクエストに対して、有効な200レスポンスを受信すること
- ユーザ名とパスワードが不正なリクエストに対して、401または403エラーレスポンスを受信すること

「TCPトランザクション」モニタは、他のプロトコルを使用するノードでも構成要素として役立ちます。

より高度なテストを実行するために、モニタで外部プログラムやスクリプトを実行することができます。「外部プログラムモニタ」については、[外部プログラムモニタ](#)に説明があります。

ノード単位のモニタとプール全体のモニタ

モニタはノード単位とプール全体の2つのカテゴリに分かれます。ノード単位のモニタは、プール内の各ノードのヘルスをテストします。プール全体のモニタは、プール全体のヘルスに影響を与える特定のマシンに対してテストを実行します。例えば、メールサーバプールがデータをNFSサーバに保存し、各バックエンドサーバがそのNFSサーバにアクセスしている場合、プール全体のモニタではこのNFSサーバをテストできます。このサーバで障害が発生すると、どのバックエンドサーバもデータを読み込むことができないため、プール全体で障害が発生したものと見なされます。

モニタをノード単位にするかプール全体にするかを選択できます。プール全体にする場合、プールが使用するNFSサーバなど、モニタでテストするマシン(および場合によってはポート)を指定する必要があります。

ヘルスモニタの使用

プールへのモニタの適用

モニタをプールに割り当てるには、**Services** (サービス) > **Pools** (プール)をクリックします。その後、プールの名前をクリックして編集します。

Health Monitoring (ヘルスモニタリング)をクリックして、プールで使用するモニタを選択します。ドロップダウンからモニタを選択して**Add Monitor** (モニタ追加)をクリックし、新しいモニタを追加することができます。プールに適用されているモニタの設定を変更するには、そのモニタの**Edit In Catalog** (カタログ内編集)リンクをクリックします。



テストリクエストを送信するモニタを使用すると、多少ですがプールへの負荷が増加します。DoS障害を防ぐために、バックエンドサーバがモニタで生成されたトラフィックに対応できることを確認してください。

外部プログラムモニタ

外部プログラムモニタは、任意の言語で作成できます。Traffic Managerは、実行可能ファイルに以下のコマンドライン引数を渡します。

```
domonitor --ipaddr=<machine_to_monitor> \  
--node=<nodename> \  
--port=<port_to_monitor>
```



ipaddr、node、およびport引数は常に外部プログラムモニタに渡されます。node引数はプールに設定されているノードのホスト名部分、ipaddrはそれに対応するIPアドレスです。

モニタにその他の引数を指定することもできます。これは、同じ実行可能ファイルで複数のモニタリングタスクを実行する場合に役立つことがあります。該当するモニタの**Edit** (編集)ページの「Program Arguments」(プログラム引数)セクションを使用します。

例えば、2つの追加の引数(「regex」および「interface」)を設定し、**Edit**(編集)ページでこれらの引数に適切な値を指定すると、モニタの実行可能ファイルは以下のように呼び出されます。

```
domonitor --ipaddr=<machine_to_monitor> \  
  --node=<nodename> \  
  --port=<port_to_monitor> \  
  --interface=<interface_value> \  
  --regex=<regex_value>
```

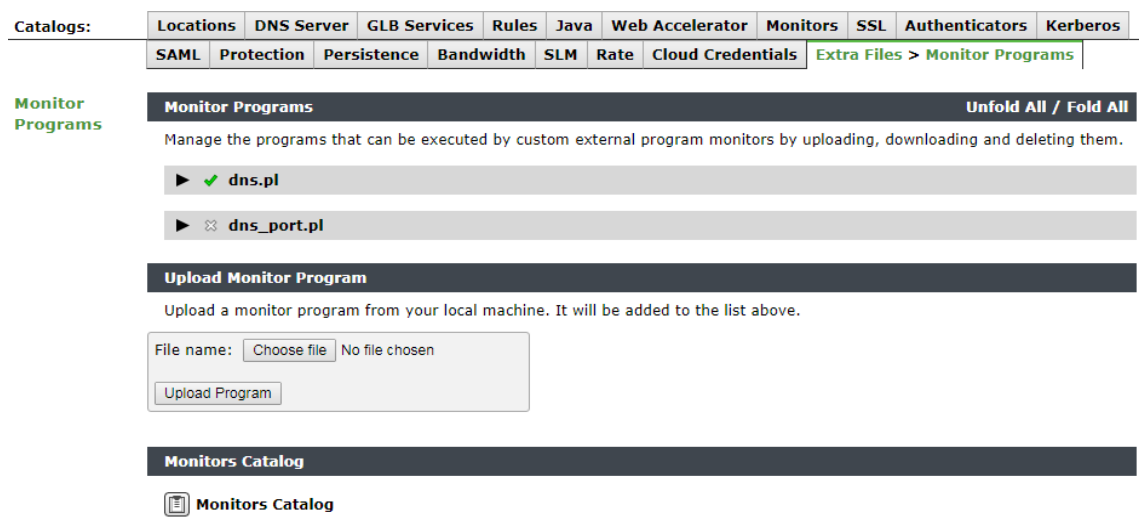
モニタが正常に終了すると、終了コード0が返されます。タイムアウトまたは0以外の終了コードは、失敗と見なされます。この場合、STDERRの内容がイベントログに書き込まれます。

モニタの実行可能ファイルまたはスクリプトを作成してテストする場合、上記のフォーマットで引数を指定してコマンドラインから簡単に実行することができます。モニタにより、デバッグ情報をSTDOUTに出力することができます。

Traffic Managerでモニタを使用する場合、詳細モードを有効化することができます。モニタが詳細モードで呼び出された場合、実行可能ファイルによってSTDOUTに出力された全ての情報がイベントログに書き込まれます。詳細モードの有効化/無効化は、モニタのEdit (編集)ページで切り替えることができます。

Traffic Managerへのモニタのアップロード

Traffic Managerでモニタをアップロード、管理、および削除するには、**Catalog** (カタログ) > **Extra Files** (外部ファイル) > **Monitor Programs** (モニタプログラム)をクリックします。



Perlでのモニタの作成

Perlでモニタを作成する場合、Traffic Managerに含まれているヘルパモジュールMonitor.pmを使用します。このモジュールには、**ParseArguments()**関数と、成功または失敗で終了する関数およびログを書き出す関数が含まれています。

```
use Zeus::ZXTM::Monitor qw( ParseArguments
                             MonitorWorked
                             MonitorFailed
                             Log );

%ARGS = ParseArguments();

Log( "Running test" );
if( do_test( $ARGS{ipaddr}, $ARGS{port} ) ) {
    MonitorWorked();
} else {
    MonitorFailed( $error_message );
}
```

Monitor.pmの詳細を表示するには、以下のコマンドを使用します。

```
$ perldoc ZEUSHOME/zxtm/lib/perl/Zeus/ZXTM/Monitor.pm
```

サービスプロテクション

本章では、インターネットホスティングに伴ういくつかのリスクと、Traffic Managerを使用してサービスに対するこれらのリスクを減らす方法について説明します。

リスクの分類

サービス妨害

「DoS」(サービス妨害)攻撃は、サービスの正当な利用を妨げる悪意のある試みを特徴とし、ネットワークへの過剰なデータの送信やマシン間の接続の妨害といった形を取ります。サーバまたはサーバクラスタの使用可能なCPU、帯域、またはストレージを枯渇させるなど、特定のユーザやアプリケーションが通常利用できるリソースを消費することを狙ったものもあります。

DoS攻撃の根本にあるのは、設計上の弱点やOSの欠陥、CPU時間、ディスク、メモリなどのリソースへのアクセスが制限されていないサービスの悪用です。例えば、悪意のある攻撃者がデータベース駆動型のWebサイトに対し、完了までに極度に集中的なSQL処理を必要とするリクエストを作り上げる方法を見つける可能性もあります。このようなリクエストが行われると、そのクエリが実行されている間はデータベースへの正当なアクセスが実質的に拒否されかねません。

Webサイトなど公開されたサービスを通して仕掛けられるDoS攻撃は、最も一般的なリスクに分類されます。Traffic Managerは、このような攻撃の影響を軽減、または全て阻止できるよう設計されています。

Webワームとウイルス

OSの脆弱性は、弱点から侵入し、自己インストールして増殖する多くのインターネットワームが生まれるきっかけになりました。例えば、ワームには、それ自体をさらに増殖させるために十分なペイロードをインストールできるものもあります。ワームの影響はデスクトップPCにも及んでいます。このリスク分類の攻撃頻度は増加しつつあり、その多くは、感染したマシンが攻撃の一端を担う計画的な「分散型サービス妨害」(DDoS)攻撃へと発展します。

ワームに加え、デスクトップPCのユーザは、電子メールやセキュリティの甘いWebサイトから侵入するウイルスのリスクにもさらされています。これらのウイルスにより、PCにリモートコントロール可能なサービスがインストールされる可能性もあります。

DDoS攻撃

分散型サービス妨害(DDoS)攻撃では、通常はワームやウイルスに感染した、「戦力として集められた」多数のクライアントマシンが利用されます。悪意のある攻撃者は、これらのクライアントをリモートコントロールできるため、特定のサービスプロバイダや企業に対して集中攻撃を行い、従来のDoS攻撃よりはるかに大規模な影響を与えることが可能です。この問題を完全に解決するためには、他のISPとの協力(外部に送信されるパケットのフィルタなど)が必要になるため、分散型攻撃を減らすことはDoS攻撃に対処するより大幅に困難です。

Traffic Managerは、ネットワーク帯域がデータであふれた状態ではなく、受信する接続を管理する十分な処理能力があれば、基本的なDDoS攻撃およびDoS攻撃を阻止することができます。

不正形式のHTTP攻撃

適切にファイアウォールで保護されたWebサーバであっても、公開サービスをポート80でインターネットに提供する必要があります。HTTPのクライアントサーバアーキテクチャでは、クライアントがポート80を通してこれらのWebサーバに自由にデータを送信することができます。言い換えれば、悪意をもってサーバを過負荷状態または正常に機能しない状態へと追い込むようこの情報を作り上げることも可能です。バグを悪用してWebサーバを脆弱にし、コントロールをリモート攻撃者に引き渡すような不正形式のリクエストが作成される状況で攻撃は発生します。

ファイアウォールおよびその他のセキュリティ対策

Traffic Managerはファイアウォールではなく、専用のファイアウォールと組み合わせて使用するための製品です。セキュアな運用の詳細については、[システムセキュリティ](#)を参照してください。

保護機能

Traffic Managerは、セキュリティ上の弱点を突いた既知の攻撃の影響を受けない堅牢なソリューションです。例えば、既知のあらゆるワームおよび不正形式のHTTP攻撃に屈することはありません。また、Traffic Managerはクラスタに送信されるトラフィックの入口に配置され、ルーティングをインテリジェントに決定することができるため、プラットフォームを明らかなネットワーク攻撃から防御したり、悪意のあるコンテンツを透過的にフィルタすることが可能です。これは、バックエンドサーバがインターネットIPアドレスを持たず、プライベートネットワークを使用している場合に特に有効です。

Traffic Managerは、各仮想サーバに個別に設定可能な「サービスプロテクション」機能も備えています。これらの機能では、様々な方法で動的にリクエストを管理し、フィルタすることができます。

ネットワークアクセス制限

クライアントの送信元IPアドレスを基に、リクエストを受け入れるかどうかを決定することができます。特定のIPアドレスまたはネットワークブロックで悪意のあるリクエストが生成されている場合、これらのアドレスからのコネクションを全て効率的に破棄するようTraffic Managerを設定することで、バックエンドプールを攻撃から保護することができます。

コネクション制限

システムを乱用しようとするクライアントが異常に多くのコネクションを生成することがよくあります。Traffic Managerは、アクセス頻度の高い上位10のIPアドレスや、各IPアドレスから行われた1分あたりの接続数に基づいて、過剰なアクティビティを検出してフィルタすることができます。これにより、Traffic Managerは、システムに対して過度に集中的なリクエストを行うクライアントのアクセスを拒否することができます。

不正形式のHTTPのフィルタ

Traffic Managerは、特定の不正形式のHTTPリクエストを検出して拒否したり、規格に準拠したリクエストを送信するようクライアントにリクエストしたりすることができます。例えば、リクエストからバイナリデータをフィルタし、非常に大きいヘッダがDoS攻撃の誘導手段として使用されないようにすることができます。

ルールベースの保護

Traffic Managerのサービスプロテクションシステムは、受信する全てのリクエストをチェックし、指定条件に一致するものを全て破棄するよう設定できます。この機能を利用すると、Webサイトで実行されているサードパーティアプリケーションの既知の脆弱性からシステムを保護することができます。例えば、買い物かごなどの一部のビジネスクリティカルなアプリケーションには、不正形式のパラメータを含むURLのリクエストを契機として生じる脆弱性が存在することが分かっています。

一般に、これらのリクエストの形式は、セキュリティメーリングリストで公開されています。Traffic Managerのサービスプロテクションシステムでは、この情報を使用して、不正形式のパラメータを含む全てのリクエストをフィルタして除外できるため、全てのWebサイトにわたってこのような脆弱性に素早く簡単に対処することができます。

受信する全てのリクエストをフィルタするためのルールのリストを作成することにより、リクエストをブロックするようTraffic Managerを設定することができます。また、リクエストのサイズやリクエストに含めることのできる情報のフォーマットを制限することも可能です。

サービスプロテクションの有効化

サービスに保護を適用するには、まず「サービスプロテクション」カタログに保護クラスを作成します。次に、**Services** (サービス) > **Virtual Servers** (仮想サーバ) > **Edit** (編集) > **Classes** (クラス) ページで仮想サーバにそのクラスを適用します。

サービスプロテクションクラスは必要な数だけ設定でき、それぞれに独自の設定およびルールを指定できます。また、1つのサービスプロテクションクラスを複数の仮想サーバに適用できます。

サービスプロテクションクラスの追加

サービスプロテクションクラスを追加するには、**Catalogs** (カタログ) > **Service Protection** (サービスプロテクション) をクリックします。

Traffic Managerに、既存のサービスプロテクションクラスのリストが表示されます。名前を入力して**Create Class** (クラス作成) をクリックすることにより、このリストに新しいサービスプロテクションクラスを追加できます。

新しいサービスプロテクションクラスを追加するときに、5つの主要設定グループで設定を行います。

基本設定

これらの設定では、サービスプロテクションの有効化/無効化を切り替えたり、どのモードでサービスプロテクションクラスを機能させるかを決定したりします。

このサービスプロテクションクラスが一時的に適用されないようにするには、そのクラスのサービスプロテクションの有効化/無効化を切り替えます。これは、特にこのサービスプロテクションクラスを使用するよう多数の仮想サーバが設定されている場合に便利です。

ログバッファ時間、すなわちディスクへの書き込みを減らすために(またDoS攻撃中に大量の結果が記録される場合の波及的な影響を抑えるために)Traffic Managerがロギングをバッファする時間を設定することができます。Traffic Managerには、詳細な出力を記録するデバッグオプションもあります。

注記

デバッグオプションを使用すると、Traffic Managerにかかる負荷が大きくなる可能性があるため、商用システムでは使用しないでください。

Traffic Managerにはテストモードもあります。このモードでは、このサービスプロテクションクラスに従ってTraffic Managerが行った決定が記録されますが、それに対するアクションは実行されません。このオプションは、サービスプロテクションクラスのパラメータをサービスに適した値に調整することを目的としています。商用サービスへの移行前のテスト期間に使用してください。

同時接続

これらの設定では、Traffic Managerが個々のIPアドレスに許容する接続数の制限を指定します。

設定	説明
max_1_connections	Traffic Managerが各接続IPアドレスに許容する最大同時接続数です。この制限を無効化するには、0に設定します。
per_process_connection_count	Traffic Managerがプロセス単位の同時接続カウントモデルを使用するかどうかを決定します(通常、Traffic Managerには、有効なCPUコアに1プロセスずつ、複数のプロセスがあります)。 「Yes (はい)」をクリックすると、その接続IPアドレスはTraffic Manager内の各プロセスに最大max_1_connectionsのコネクションを行えるようになります。 「No (いいえ)」をクリックすると、その接続IPアドレスはTraffic Manager全体で最大max_1_connectionsのコネクションを行えるようになります。 per_process_connection_countが「Yes (はい)」に設定されている場合、以下の制限が追加で適用されます。 max_10_connections: アクセス頻度の高い上位10を合わせた接続IPアドレスからの最大同時接続数です。通常、Traffic Managerに対して何らかのDoS攻撃が進行中か、その他の不正なコネクションが行われている場合、特定仮想サーバの負荷のほとんどは上位10のクライアントによって占められます。そのため、この特定グループのユーザを制限しながら、その他のリクエストは通常どおり処理するのが適切です。通常の利用においては、max_1_connections制限の1～10倍の値をお奨めします。この制限の効果をなくすには、max_1_connections制限の10倍以上の値に設定してください。per_process_connection_countが「No (いいえ)」に設定されている場合、またはmax_1_connectionsが0、またはmin_connectionsが0の場合、Traffic Managerはこの設定を無視します。

設定	説明
	min_connections: max_10_connections制限に対するエントリのしきい値です。この値以下の同時接続を行う接続IPアドレスには、max_10_connections制限は適用されません。 max_1_connectionsおよびmax_10_connectionsの両方の制限を無効化するには、0に設定します。 per_process_connection_countが「No (いいえ)」に設定されている場合、Traffic Managerはこの設定を無視します。

接続レート

注記

接続レートの制限はプロセス単位で行われ、そのレートは、プロセスが処理しているコネクションに対して独立して適用されます。

これらの設定では、各接続IPアドレスが新たなコネクションまたはリクエストを行うレートに対して制限をかけることができます。

設定	説明
max_connection_rate	個々のIPアドレスが、rate_timerで指定された間隔の間にTraffic Managerと行うことができる最大接続数です。制限を無効化するには、0に設定します。 HTTPの場合、単一のkeepaliveコネクション内でトンネルされている場合でも、各リクエストが1つのコネクションとしてカウントされます。
rate_timer	Traffic Managerが、max_connection_rateが評価される頻度を判定するために使用する間隔(秒単位)です。例えば、この値を60に設定すると、毎分max_connection_rateコネクションの制限が課されます。この制限を上回って試行されたコネクションは、rate_timer間隔の残りの間、全て拒否されます。

[リクエストレートシェイピング](#)で説明するリクエストレートシェイピング機能を利用すると、接続レートをさらにきめ細かくコントロールすることができます。

アクセス制限

Allowed (許可)リストでは、許可するIPアドレスまたはネットワークを指定できます。これらのIPアドレスまたはネットワークは、このサービスプロテクションクラスでの通常のサービスプロテクションチェックが免除されます。つまり、このリストに含まれるIPアドレスは、同時接続や接続レートセクションで設定されている制限の対象になりません。

Banned (禁止)リストは、Traffic Managerに対して常にコネクションを破棄させるIPアドレスまたはネットワークを宣言するために使用します。

送信元IPをチェックするとき、この両方のリストに含まれるサブネットがロングストマッチ優先のアルゴリズムに従って評価されます。すなわち、より具体的なサブネットが優先されます。そのため、許可された具体性の低いサブネットに含まれる具体性の高いサブネットを禁止することも、その逆のこともできます。例えば、192.168.0.0/16を許可し、192.168.128.0/24を禁止すると、192.168.0.137のIPからのリクエストは許可しますが、192.168.128.137のIPからのリクエストは許可しません。**Allowed** (許可)および**Banned** (禁止)の両方のリストに同じサブネットが存在する場合、そのサブネットからのリクエストは許可されます。

いずれかのリストに追加するには、適切なボックスにIPアドレスまたはネットワークアドレスを入力します。以下のいずれかのフォーマットを使用します。

- 192.0.2.0/255.255.255.0
- 192.0.2.0/24
- 192.0.2.1
- 192.0.2.

HTTP固有の設定

これらの設定では、HTTPトラフィックに対して実行できる追加のチェックを指定します。

Traffic Managerでは、一般に認められているガイドラインに従ってURLの形式が正しいかどうかをチェックすることにより、RFC2396への適合を確保できます。標準的なWebブラウザはRFC2396に適合しているため、このテストでは、異常なデータや悪意のあるデータまたはフォーマットを含むリクエストは拒否され、正当なリクエストは通過することができます。

HTTPボディとHTTPヘッダの長さを制限することができます。これにより、悪意のある攻撃者からボディやヘッダが大量または際限なく送信されるのを防ぎます。通常、Webサーバはリクエストを受信しながらメモリにバッファするため、メモリ枯渇攻撃に対して脆弱になる可能性があります。クライアントが大量のリクエストデータを送信できなくすることで、このような攻撃から保護できます。

全てのHTTPヘッダの合計サイズ制限を指定することもできます。これにより、攻撃者が多数のHTTPヘッダを使用して非常に大きいリクエストを作成するのを阻止できます。

バッファオーバーフロー攻撃や大量のデータやワームの誘導手段としてURLを使用するその他の攻撃を防ぐために、リクエストされるURLのサイズを制限できます。

復号後、Traffic Managerはバイナリデータを含むHTTPヘッダを拒否できます。これは、一般にバッファオーバーラン攻撃が試行されたことを示しますが、一部の低品質のアプリケーションがバイナリデータを使用する場合もあります。ご使用のアプリケーションが影響を受ける可能性が疑われる場合は、まずテストモードでこの機能を使用して確認してください。

最後に、Traffic Managerでは、上記のいずれかのテストで不合格になったリクエストをそのまま拒否するか、リクエストを行ったクライアントにHTTPエラーメッセージを送信するかを指定できます。エラーメッセージを送信すると、オーバーヘッドがわずかに増加し、DoS攻撃が進行中の場合は非常に大きいオーバーヘッドが生じる可能性があります。



HTTPトラフィックを処理しない仮想サーバにこのサービスプロテクションクラスを関連付けた場合、これらの設定は無視されます。

サービスプロテクションルール

このセクションには、サービスプロテクションクラスが使用している現在のルールが表示されます。ルールを変更するには、ドロップダウンメニューから使用するルールを選択します。TrafficScriptの詳細については、[TrafficScriptルール](#)を参照してください。

設定内容を適用してサービスプロテクションクラスを作成するには、**Update** (更新)をクリックします。

仮想サーバへのサービスプロテクションクラスの適用

サービスプロテクションクラスを有効にするには、まず仮想サーバに割り当てる必要があります。それには、**Services** (サービス) > **Virtual Servers** (仮想サーバ)をクリックし、既存の仮想サーバを選択します。**Classes** (クラス)を選択し、「Service Protection (サービスプロテクション)」というセクションを探します。この管理UIには、カタログ内の使用可能なサービスプロテクションクラスのドロップダウンボックスがあります。

サービスプロテクションクラスを選択したら、**Update** (更新)をクリックします。

Traffic Managerにより、新しいサービスプロテクションクラスが即座に全てのTraffic Managerに適用されます。サービスプロテクションクラスがテストモードかデバッグモードかを示す概要テーブルが表示されます。サービスプロテクションクラスが有効化されていない場合は([基本設定](#)を参照)、そのことも示されます。

サービスプロテクション性能

Traffic Managerのサービスプロテクションシステムでは、リクエストごとに追加のメモリとCPU時間が必要になります。多数のリクエストを管理し、サービスプロテクションクラスで高度なルールを使用する場合は、このサービスプロテクションクラスを使用する仮想サーバに対する全てのリクエストについて、処理に要する時間がわずかに長くなります。大規模なシステムでは、処理能力を強化するために、フロントエンドネットワークにTraffic Managerを追加することが必要になる場合があります。

帯域制御

本章では、帯域制御の概要と、ネットワークで帯域をどのように使用するかをコントロールするようTraffic Managerを設定する方法について説明します。



Traffic Manager製品によっては、帯域制御を利用できない場合があります。この機能が必要な場合は、ソフトウェアまたはライセンスキーをアップグレードすることにより入手できます。

帯域制御とは

帯域制御により、Traffic Managerは、外部から受信するトラフィックまたは外部へと送信されるトラフィックで使用される1秒あたりのバイト数を仮想サーバまたはリクエストのタイプごとに制限することができます。

通常、ネットワーク帯域は、全ての接続に対して最高の転送速度で提供されます。そのため、二次的なサービスで多くの帯域が使用され、最もクリティカルなサービスが犠牲になるなど、ネットワークの使用が均等でなくなることがあります。Traffic Managerでは、帯域制御を使用して、この不均衡を明示的にコントロールすることができます。

例えば、20Mbps/sのネットワーク接続があり、この接続がFTPダウンロードに過剰に使用されているために、主要なHTTPサービスのレスポンス速度に悪影響が及んでいるとします。この場合、FTP仮想サーバの帯域を2Mbps/sに制限できます。

帯域制限は、クラスタ内の全てのTraffic Managerマシンで自動的に共有され、適用されます。個々のTraffic Managerは、それぞれの負荷に応じて、合計制限から異なる比率の帯域を受け取ります。未使用の帯域は、各Traffic Managerのニーズに応じて、クラスタ全体に均等に割り当てられます。



帯域制御は、Traffic Managerから送信されるトラフィックにのみ適用されます。すなわち、Traffic Managerは、リクエストをサーバノードに書き込むときやレスポンスをクライアントに書き込むときに使用される帯域をコントロールできますが、クライアントがリクエストをTraffic Managerに書き込む速度やサーバがレスポンスをTraffic Managerに書き込む速度をコントロールすることはできません。

帯域制御は、外部アドレスに送信されるか、Traffic Managerの内部のみで転送されるかにかかわらず、全てのデータに適用されます。内部トラフィックには、例えば、ループバックインターフェースを使用してデータがプールから仮想サーバに戻される場合が含まれます。

帯域制御の設定

Traffic Managerの帯域制御機能は、カタログを通して提供されます。すなわち、それぞれ制限の異なる複数の帯域制御クラスを追加することができます。また、TrafficScriptルールによって帯域制御クラスを選択できるため、同じ仮想サーバに対するリクエストを区別すると有益な状況では、さらに柔軟性が高まります。例えば、Traffic Managerの帯域制御システムを使用して、CGIリクエストに対して、画像リクエストとは異なる制限を適用することも可能です(以下の例を参照)。

複数のTraffic Managerから成るクラスタがある場合、帯域制御システムではこの構成が考慮されます。サーバを増設した場合も調整が自動的に行われるため、ユーザが調整を行う必要はありません。

カタログへの帯域クラスの追加

帯域制御クラスをカタログに追加するには、**Catalogs** (カタログ) > **Bandwidth Classes** (帯域幅クラス)をクリックします。

Traffic Managerに既存の帯域制御クラスのリストが示され、新しい帯域制御クラスを作成するためのボックスが表示されます。

このボックスに名前を入力し、**Create Class** (クラス作成)をクリックします。この新しいクラスの設定ページが表示されます。最大帯域および制限共有オプションを選択できます。

最大帯域制限を選択するには、以下のいずれかの操作を行います。

- ドロップダウンリストからあらかじめ定義されている値を選択する
- ドロップダウンリストからCustom (カスタム)を選択し、表示されるボックスに値をkbits/秒単位で入力する

帯域クラスにより帯域をコネクションに分配する方法として、以下の3つのモードがあります。

- **各コネクションを最大転送速度に制限:** 帯域の共有は適用されず、全てのTraffic Managerにおいて、このクラスが適用された全てのユーザが、設定されている最大帯域制限に固定されます。これは、個々のクライアントにビデオ/オーディオストリームを配信する場合に便利です。
- **Traffic Manager単位で帯域を共有:** 各Traffic Managerにおいて、この帯域クラスが適用された全てのユーザで制限が共有されます。1Mbpsの制限を適用し、3台のTraffic Managerがある場合、各Traffic Managerが使用できる帯域は最大1Mbpsで、合計帯域は3Mbpsになります。

- **全てのTraffic Managerで帯域を共有**: 全てのTraffic Managerにわたり、この帯域クラスが適用されている全てのユーザで制限が共有されます。1Mbpsの制限を適用し、3台のTraffic Managerがある場合、クラス全体で使用できる帯域は合計1Mbpsのみです。この制限は動的に割り当てられます。各Traffic Managerでは、他のTraffic Managerがアイドル状態のときは、1Mbps全てを使用できます。

Update (更新)をクリックして変更内容を適用します。



ストレージまたはメモリ容量とは異なり、ネットワークスループットは慣習的に(1024ではなく)1000の倍数で測定されます。

仮想サーバへの帯域クラスの割り当て

仮想サーバに割り当てられた帯域制御クラスにより、その仮想サーバがレスポンスデータをクライアントに送信するときに使用できる帯域が制限されます。

設定

帯域制御クラスを仮想サーバに割り当てるには、**Services** (サービス) > **Virtual Servers** (仮想サーバ)をクリックします。使用可能なサーバのリストから仮想サーバを選択し、**Edit** (編集)ボタンをクリックします。

次に、**Classes** (クラス)をクリックして「Bandwidth Management (帯域管理)」というタイトルのセクションを探します。使用可能ないずれかの帯域制御クラスをラジオボタンで選択します。

ページの下部にある**Update** (更新)ボタンをクリックして変更内容を適用します。仮想サーバで即座にこの帯域制御クラスの使用が開始されます。



帯域制御クラスがない場合は、「none configured (設定済みなし)」オプションのみが表示されます。帯域制御クラスを追加するには、**Create New Bandwidth Management Class** (新しい帯域制御クラス作成)をクリックします。

プールへの帯域クラスの割り当て

プールに割り当てられた帯域制御クラスにより、そのプールがリクエストデータをノードに送信するときに使用できる帯域が制限されます。

設定

帯域制御クラスをプールに割り当てるには、**Services** (サービス) > **Pools** (プール)をクリックします。使用可能なサーバのリストからプールを選択し、**Edit** (編集)ボタンをクリックします。

次に、**Bandwidth Management** (帯域制御)をクリックします。いずれかの帯域制御クラスをラジオボタンで選択します。

ページの下部にある**Update** (更新)ボタンをクリックして変更内容を適用します。プールで即座にこの帯域制御クラスの使用が開始されます。



帯域制御クラスがない場合は、「none configured (設定済みなし)」オプションのみが表示されます。帯域制御クラスを追加するには、**Create New Bandwidth Management Class** (新しい帯域制御クラス作成)をクリックします。

TrafficScriptを使用した帯域クラスの選択

帯域制御クラスは、TrafficScript関数のrequest.setBandwidthClass()およびresponse.setBandwidthClass()のいずれかを使用してリクエストごとに選択することもできます。これらの関数は、それぞれ「ノード方向」の帯域と「クライアント方向」の帯域に影響を与えます。

帯域制御クラスは、TrafficScriptルールの任意の時点で適用でき、これによって仮想サーバ自体の設定がオーバーライドされます(設定がある場合)。

例えば、以下のTrafficScriptレスポンスルールでは、大きいレスポンスまたはサイトの「/downloads/」部分からのレスポンスに対して「downloads」帯域制御クラスが選択されます。

```
# This must be used as a response rule, because we
# want to determine the response content length...
$url = http.getPath();
$clen = http.getResponseHeader( "Content-Length" );

if( string.startsWith( $url, "/downloads" ) ||
    $clen >= 100*1024 ) {
    response.setBandwidthClass( "downloads" );
}
```

リクエストレートシェイピング

本章では、リクエストレートシェイピングの概要と、アプリケーションに対するリクエストをレート制限するよう Traffic Manager を設定する方法について説明します。



Traffic Manager 製品によっては、リクエストレートシェイピングを利用できない場合があります。この機能が必要な場合は、ソフトウェアまたはライセンスキーをアップグレードすることにより入手できます。

リクエストレートシェイピングとは

特定のユーザによってサービスが独占され、他のユーザによるそのサービスの利用が妨害されることがあります。バックエンドアプリケーションインフラストラクチャの拡張性が限られている場合もあり、アプリケーションに送信されるリクエストが多すぎると、容易に処理が追いつかなくなります。より広範なセキュリティポリシーの一環として、電子メールの送信やサービスへのログインなどの特定のアクティビティの許容レートを制限することが望ましい場合があります。

リクエストレートシェイピングでは、イベントの識別方法を非常にきめ細かくコントロールしながら、幅広いイベントに対して制限を指定することができます。これらのイベントに1秒あたりのレートおよび1分あたりのレートを適用できます。

以下に例を挙げます。

- 個々のWebスパイダをレートシェイピングして、Webサイトの処理が追いつかなくなる状況を阻止できます。それぞれ異なるリモートIPアドレスからアクセスする各Webスパイダに最大リクエストレートを適用することができます。
- 個々のSMTPコネクションまたは同じクライアントによるコネクショングループを抑制し、各コネクションの1分あたりの最大送信電子メール数を制限することができます。
- 新しいSMTPコネクションをレートシェイピングし、リモートクライアントが特定レートでのみ新しいコネクションを確立できるようにすることもできます。
- 特定のアプリケーションに転送される1分あたりの接続数にグローバルレートシェイピングを適用できます。
- 個々のユーザによるサービスへのログイン試行を識別し、各ユーザの1分あたりの試行回数を制限することにより、ログインの辞書攻撃を阻止することができます。

TrafficScript関数の`rate.use()`を使用してリクエストレート制限を適用すると、以下のように処理が行われます。

- 仮想サーバが受信トラフィックを受け入れます。
- 仮想サーバによりリクエストルールが実行されます。

- リクエストルールにより、TrafficScript関数のrate.use()を使用してレートシェイピングが適用されます。



一般に、リクエストレート制限は、リクエストの処理レートをシェイピングするためにTrafficScriptリクエストルールで使用します。必要に応じて、レスポンスルールで使用することも可能です。この場合、その他のイベントを制限できますが、通常は必要ありません。

リクエストレートシェイピングクラス(レートクラス)の設定

リクエストレートシェイピングを使用するには、少なくとも1つのリクエストレートシェイピングクラス(レートクラスとも言う)を追加する必要があります。各レートクラスでは、1秒または1分あたりのイベントのレート制限を定義します。

レートクラスはカタログに保存され、TrafficScript関数のrate.use()を使用して呼び出します。

カタログへのレートクラスの追加

新しいレートクラスを追加するには、**Catalogs** (カタログ) > **Rate Shaping classes** (レートシェイピングクラス)をクリックします。「Create new Rate class (新しいレートクラスの作成)」ボックスに新しいクラスの名前を入力し、**Create Class** (クラス作成)をクリックします。

レートクラスには以下の設定が含まれます。

設定	説明
max_rate_per_minute	1分あたりに許容されるリクエストの最大数です。値「0」(ゼロ)は制限なしを意味します。
max_rate_per_second	1秒あたりに許容されるリクエストの最大数です。値「0」(ゼロ)は制限なしを意味します。

必要に応じて、1秒あたりの制限と1分あたりの制限を両方設定することができます。この場合、両方の制限が適用されます(1分あたりの制限が1秒あたりの制限の60倍を超えている場合、1分あたりの制限に効果がないことに注意)。

レートクラスを設定したら、**Update** (更新)をクリックして変更内容を適用します。

レートクラスの使用

レートクラスはキューとして機能します。TrafficScript関数のrate.use()が呼び出されると、コネクションが一時中断し、レートクラスが管理するキューに追加されます。その後、1秒および1分あたりの制限に従って、コネクションがキューから解放されます。

「shape requests」というレートクラスを作成し、1秒あたりのイベント数(リクエスト数)を10個に制限するよう設定したとします。

以下のTrafficScriptリクエストルールでは、新しいリクエストがシステムに着信する頻度に関わらず、1秒あたり処理されるリクエストは最大10個です。

```
rate.use( "shape requests" );
```

リクエストが着信すると、リクエストルールが実行されます。1秒あたりに着信するリクエストが10個未満の場合、rate.use()関数は即座に戻り、リクエストは遅延なく処理されます。

1秒あたりに着信するリクエストが10個を超えると、rate.use()関数によって一部のリクエストがキューイングされます。TrafficScriptからは、この関数がブロックされているように見えます。その後、リクエストがキューから抜けた時点で、rate.use()関数が戻ります。

レートキュー

キューイングされる接続のバックログのサイズに制限はありません。例えば、1000個のリクエストが短時間に連続して着信すると、そのうち990個のリクエストは即座にキューイングされます。その後、1秒あたり10個のリクエストがキューの前方から解放されていきます。

キューイングされている間に、接続がタイムアウトになったり、リモートクライアントにより切断される場合があります。この場合、接続は即座に破棄されます。

rate.getBacklog()関数を使用して、現在キューイングされているリクエスト数を調べることができます。バックログが大きすぎる場合、接続がタイムアウトになるまで待たずにユーザにエラーページを返すことができます。

```
if( rate.getBacklog( "shape requests" ) > 100 ) {  
    http.redirect( "http://some.other.site/" );  
} else {  
    rate.use( "shape requests" );  
}
```

このルールにより、このレートクラスによってキューイングされるリクエストが100個を超えることがなくなります。

キューイングの回避

TrafficScript関数のrate.use.noqueue()を使用しても、rate.use()と同様にリクエストをレートシェイピングできますが、この場合、リクエストがキューイングされないという大きな違いがあります。

- レート制限を超えていない場合、`rate.use.noQueue()`は即座に値1を返し、接続レートのカウントが増加します。これは`rate.use()`と同じです。
- レート制限を超えている場合、`rate.use.noQueue()`は値0を返します(これに対して`rate.use()`はリクエストをキューイングします)。この場合、TrafficScriptを使用して、503 Too Busyエラーを返すなどリクエストに対する処理を決定することができます。

選択的レートシェイピング

特定タイプのリクエストのみをレート制限したい場合があります。例えば、WebサイトでJava Extensionページとその他のコンテンツ(画像、静的コンテンツなど)が混在している場合、以下のようにJava Extensionページのみをレートシェイピングすることができます。

```
$url = http.getPath();
if( string.endsWith( $url, ".jsp" ) ) {
    rate.use( "shape requests" );
}
```

さらにきめ細かいレートシェイピング

これまでに説明した例は、全てグローバル制限を適用するものです。全てのリクエストが同じレートキューに入れられます。

多くの場合、さらにきめ細かいレートシェイピング制限を適用することが必要になります。例えば、ログインページについて、各ユーザがログイン試行できる頻度を1分あたり2回に制限するとします。

グローバル制限では、ログインページへのアクセス回数を合計で1分あたり2回に制限することになりますが、代わりにユーザごとの制限を適用して、各ユーザのログイン回数を1分あたり2回に制限します。

ユーザごとに固有のレートシェイピングクラスを作成し、TrafficScriptを使用して適切なクラスを選択することもできますが、これよりはるかに良い方法があります。

`rate.use()`関数は、レートクラスの新しいインスタンスを識別するオプションの「キー」を使用できます。このキーを使用して、同じ制限を共有する複数の独立したレートクラスを作成し、これらのクラスをキーごとに個別に適用することができます。

```
$url = http.getPath();
if( string.endsWith( $url, "login.cgi" ) ) {
    $user = http.getFormParam( "username" );
    rate.use( "login limit", $user );
}
```


レートシェイピング制限は、\$userの各値に個別に適用されます。システムに新たにアクセスした各ユーザは、「login limit」レートシェイピングクラスを共有するその他全てのユーザとは関係なく、1分あたりのリクエストが2個に制限されます。

レートシェイピングWebスパイダ

低品質のWebスパイダからWebサイトに大量のリクエストが送信され、過剰な帯域が消費されて、正当なサイトユーザが受けるサービスレベルが低下する場合があります。

以下のTrafficScriptルールは、「spiders」というレートクラスを使用して各スパイダのリクエストを1秒あたり1個に制限します。TrafficScriptは、User-Agentヘッダに「Mozilla/」または「Opera/」のプレフィックスのないユーザをスパイダと見なします。そして、ユーザエージェントおよび送信元IPアドレスでスパイダを識別し、各スパイダインスタンスに制限を適用します。

```
$ua = http.getHeader( "User-Agent" );

if( ! string.startsWith( $ua, "Mozilla/" ) &&
    ! string.startsWith( $ua, "Opera/" ) ) {
    $ip = request.getRemoteIP();
    rate.use( "spiders", $ua.$ip );
}
```

さらに高度なルールでは、スパイダを動的に検出するために、5秒間に渡って1秒あたりのリクエストが10個を超えるなどレートを超えたことのあるユーザのみに対して、1秒あたりのリクエストを1個に制限することもできます。このルールは以下のように機能します。

- TrafficScript関数のdata.set()を使用して、既知の違反者(ユーザエージェントとIPアドレス)のリストを維持します。このリストは、最初は空の状態です。
- リクエストの受信時に、既知の違反者のリストを確認し、リクエストを抑制する場合は1秒あたり1個のリクエストのレートを使用します。
- それ以外の場合は、1秒あたり10個のリクエストのレートを使用しますが、レート制限の適用前にバックログを確認します。バックログが大きすぎる場合(50を超える場合など)、このユーザが送信しているリクエストが多すぎるため、違反者のリストに挿入します。

リクエストレートシェイピングのグラフ化

アクティビティモニタを使用して、リクエストレートシェイピングクラスのアクティビティのチャートを作成し、SNMPを使用して値をモニタリングします。

この方法でグラフ化やモニタリングを行えるのは、グローバルレート制限のみです。特定のキーに適用されたレート制限をグラフ化したりモニタリングしたりすることはできません。

サービスレベルモニタリング

本章では、サービスレベルモニタリングの概要と、サービスレベルモニタリングを使用してネットワークのレスポンス時間をモニタリングするようTraffic Managerを設定する方法について説明します。



Traffic Manager製品によっては、サービスレベルモニタリングを利用できない場合があります。この機能が必要な場合は、ソフトウェアまたはライセンスキーをアップグレードすることにより入手できます。

サービスレベルモニタリングについて

サービスのレスポンス時間は、Webサービスの重要なメトリクスであり、レスポンス時間が短いほど、反応のよい使いやすいサービスであるという印象を与えます。サービスレベルモニタリングでは、Traffic Managerがリクエストを受信してからレスポンスデータの最初のバイトをクライアントに送信するまでにかかった時間(ミリ秒単位)をレスポンス時間として使用します。これには、コンテンツがインターネットを経由してクライアントまで返される時間は含まれず、一般にはリモートネットワークの速度の影響を受けません。

レスポンス時間には、Traffic Managerでの内部処理(TrafficScriptルールやJava Extensionsの実行など)にかかった時間、バックエンドノードへのリクエストの送信にかかった時間、ノードがレスポンスを生成して最初のデータを返すまでにかかった時間、レスポンスルールの実行にかかった時間が含まれます。一般的に、レスポンス時間はクライアントコネクションの速度や待ち時間とは無関係のため、内部システムの性能を表す正確な指標となり、アプリケーション管理者は、これを基に調整や最適化を行うことができます。



ノードがレスポンスを返す前に、クライアントからHTTP POSTボディデータなどの追加データを読み取る必要がある場合、このデータの読み取り時間もレスポンス時間の測定対象になります。

サービスレベルモニタリングを使用すると、Traffic Managerは、実際のレスポンス時間と適合値を比較して、ノードのレスポンス時間の変化を測定し、それに対処することができます。また、レスポンス時間やその他の関連情報のグラフを取得したり、レスポンス時間が適合値を下回った場合にアラートを報告するかログに記録することができます。Traffic Managerでは、TrafficScriptルールを使用して、レスポンス時間に基づいて利用可能なリソースを動的に調整することもできます。

サービスレベルモニタリングクラス(SLMクラス)の設定

サービスレベルモニタリング(SLM)を使用するには、少なくとも1つのSLMクラスをTraffic Managerに追加します。各SLMクラスでは、目標とするレスポンス時間とアラートおよびロギングのしきい値を定義します。

SLMクラスはサービスレベルモニタリングカタログに保存され、各クラスは1台以上の仮想サーバに適用することができます。

カタログへのSLMクラスの追加

新しいSLMクラスを追加するには、**Catalogs** (カタログ) > **SLM**をクリックします。「Create new service level monitoring class (新しいサービスレベルモニタリングクラスの作成)」ボックスに新しいクラスの名前を入力し、**Create Class** (クラス作成)をクリックします。

各SLMクラスには、以下の多数の設定可能な項目が含まれます。

設定	説明
response_time	Traffic Managerが測定するレスポンス時間の時間制限(ミリ秒単位)です。経過した応答時間がこの値以下の場合、Traffic Managerはレスポンスを適合として扱います。この値より長くかかったレスポンスは、不適合と見なされます。 Traffic Managerがレスポンス時間を測定する方法の説明については、 サービスレベルモニタリングについて を参照してください。
serious_threshold	適合レスポンスのパーセンテージがこのレベルを下回ると、SERIOUS (重大)レベルのイベントが有効になり、SERIOUS (重大)レベルのメッセージがイベントログに書き込まれます。
warning_threshold	適合レスポンスのパーセンテージがこのレベルを下回ると、WARNING (警告)レベルのイベントが有効になり、WARNING (警告)レベルのメッセージがイベントログに記録されます。

例えば、レスポンス時間を1000ms、警告しきい値を適合レスポンスの70%、重大しきい値を適合レスポンスの40%に設定したとします。この場合、以下のようになります。

- サーバから戻る時間が1000ms未満のリクエストが70%未満の場合、Warning (警告)レベルのエントリがTraffic Managerのイベントログに書き込まれます。
- 所要時間が1000ms未満のリクエストが40%未満の場合、Serious (重大)レベルのエントリがTraffic Managerのイベントログに書き込まれます。

適合リクエストのパーセンテージは、過去10秒間のリクエストについて、そのSLMクラスでモニタリングしているリクエストのうち**response_time**しきい値を満たしたリクエスト数とリクエストの合計数を比較した平均値から求められます。

SLMクラスにより「Warning」(警告)または「Serious」(重大)レベルのイベントが有効になったときにアクションを呼び出すイベントハンドラを定義することができます。

SLMクラスを設定したら、**Update** (更新)をクリックして変更内容を適用します。

仮想サーバへのSLMクラスの適用

SLMクラスを仮想サーバに割り当てるには、**Services** (サービス) > **Virtual Servers** (仮想サーバ)をクリックします。使用可能なサーバのリストから仮想サーバを選択し、**Edit** (編集)ボタンをクリックします。

次に、**Classes** (クラス)をクリックし、「Service Level Monitoring (サービスレベルモニタリング)」というタイトルのセクションを探します。使用可能ないずれかのSLMクラスをラジオボタンで選択します。

Update (更新)をクリックして変更内容を適用します。

SLMクラスを仮想サーバに関連付けると、そのクラスが全てのリクエストに対して直ちに使用されます。ただし、次のセクションで説明するように、TrafficScriptを使用してリクエストの性質に応じてSLMクラスを変更することも可能です。

TrafficScriptによるSLMクラスの適用

TrafficScriptには、SLMクラスから最新情報を読み取ったり、コネクションのSLMクラスを変更したりするための関数が多数用意されています。

これらの関数については、『**TrafficScript Reference**』(オンラインヘルプから利用可能)に記載されています。

特に、`connection.setServiceLevelClass()`でリクエスト用に特定のSLMクラスを選択すると、仮想サーバのデフォルトSLMクラスをオーバーライドできます。現在の適合率の値を返す`slm.conforming()`も参照してください。

SLMクラスのTrafficScriptの例

以降に示す例では、特定タイプのリクエストについてデフォルトのSLMクラスをオーバーライドする方法と、特定のSLMクラスの性能を基にトラフィックの管理方法をコントロールする方法を示します。

「FrontPage™スクリプトのみ」のサービスレベルモニタリング

特にクライアントが低速なコネクションでコンテンツをアップロードする場合、Microsoft FrontPage™アップロードスクリプトの実行に時間がかかることがあります。これにより、正常に動作しているにも関わらずプールに障害のマークが付けられる可能性もあります。

通常のコンテンツのレスポンス時間とFrontPageバイナリのレスポンス時間を区別するために、FrontPageリクエスト(_vti_bin内)を検出するTrafficScriptルールを使用できます。

```
$path = http.getPath();

if( string.StartsWith( $path, "/vti_bin" ) ) {
    connection.setServiceLevelClass( "FrontPage" );
} else {
    connection.setServiceLevelClass( "Regular" );
}
```

FrontPage SLMクラスでは、仮想サーバが管理するその他全てのリクエストで使用されるRegularクラスよりも大幅に長いレスポンス時間を許容する必要があります。

サービスレベルモニタリングによるリソースのプライオリティ設定

この例では、プレミアムカスタマに「premium」サービスレベルモニタリングクラスのタグを付け、そのトラフィックを「premium」プールに送信します。

プレミアムカスタマ以外のカスタマは、premium SLMクラスが許容値内で機能しているときはpremiumプールを共有できますが、premium SLMクラスの実行速度が遅い場合は「standard」プールに送信されます。

```
if( $IsPremium ) {
    connection.setServiceLevelClass( "premium" );
    pool.use( "premium" );
} else {
    if( slm.conforming( "premium" ) == 100 ) {
        pool.use( "premium" );
    } else {
        pool.use( "standard" );
    }
}
```



pool.use()はTrafficScriptルールを即座に終了し、リクエストを指定プールに送信します。

SLMクラス適合レートのグラフ化

特定SLMクラスのデータのグラフを取得するには、**Activity** (アクティビティ)をクリックします。現在のアクティビティグラフの下にある**Change Data** (データ変更)をクリックします。

Traffic Managerに、プロット可能な様々な値のツリーが表示されます。**Service Level Monitoring Classes** (サービスレベルモニタリングクラス)と示されたノードを展開します。



ここで、特定のSLMクラスのデータをプロットするか、全てのSLMクラスのデータをプロットするかを選択できます。また、特定のSLMクラスの平均レスポンス時間など、SLMクラスが使用する測定データの値をプロットすることもできます。全てのSLMクラスの合計接続数と合計不適合接続数を同じグラフにプロットすると、システムの全体的な性能を視覚的に比較しやすくなります。

全ての接続データとレスポンス時間データをノード別にプロットすることができます。これは、十分に機能していないノードを特定するときに役立ちます。この目的には、SLMクラスが使用する各ノードの平均レスポンス時間をプロットしたグラフが適しています。SLMクラスが使用するノードのデータを選択するには、そのSLMクラス名の付いたセクションを展開し、対象のノードを選択します。

コンテンツキャッシング

本章では、Traffic Managerのコンテンツキャッシング機能の概要と、この機能を設定してバックエンドWebサービスノードの負荷を減らす方法について説明します。



Traffic Manager製品によっては、コンテンツキャッシングを利用できない場合があります。この機能を取得するには、ソフトウェアまたはライセンスキーのアップグレードについてサポート窓口にお問い合わせください。

注記

仮想サーバでPulse Secure Web Accelerator(Web Content Accelerator)機能を有効にしたときのキャッシングへの影響については、[コンテンツキャッシング](#)も参照してください。

はじめに

多くのWebサーバは、幅広いタイプのコンテンツをホストし、これらのコンテンツによってWebサイトと提供サービスが構成されます。例えば、一般的なWebサイトには、標準的なWebページ(HTMLファイル)、画像(GIF、JPEG、PNGファイル)、CSSスタイルシート、JavaScriptソースファイルが含まれます。Webサイトで提供するサービスによっては、Webサーバでホストされているアプリケーションによって生成される、パーソナライズされたカスタムコンテンツがさらに含まれる場合もあります。

Traffic Managerのコンテンツキャッシング機能により、Traffic Managerは、リクエストごとにカスタマイズされないコンテンツを特定し、そのコンテンツを記憶(「キャッシング」)することができます。コンテンツは、Webサーバのディスク上のファイルなど「静的」な場合と、Webサーバで実行されているアプリケーションによって生成される場合があります。

Traffic Managerは、内部Webキャッシュにキャッシングしたコンテンツが別のクライアントからリクエストされた場合、リクエストをバックエンドWebサーバに転送せずに、そのコンテンツをクライアントに直接返すことができます。

これには、特に、Webサーバマシン上のリソースを消費する複雑なアプリケーションによって生成されたコンテンツをキャッシングできることをTraffic Managerが検出した場合に、バックエンドWebサーバの負荷を減らす効果があります。

コンテンツキャッシングの設定

コンテンツキャッシングは、HTTP仮想サーバが実行します。仮想サーバは、頻度の高いリクエストに対するレスポンスをWebキャッシュに保存し、このようなリクエストに対するレスポンスをWebキャッシュから直接返します。

仮想サーバへのコンテンツキャッシングの適用

HTTP仮想サーバのためにコンテンツキャッシングを設定するには、**Services** (サービス) > **Virtual Servers** (仮想サーバ)をクリックします。使用可能なサーバのリストから仮想サーバを選択し、**Edit** (編集)ボタンをクリックします。

「Content Caching (コンテンツキャッシング)」を探して、**Edit** (編集)をクリックします。**webcache!enabled**を使用して、その仮想サーバのコンテンツキャッシングを有効化または無効化します。

変更を適用するには、**Update** (更新)をクリックします。

注記

この仮想サーバに対してWeb Content Accelerator機能が有効化されている場合、**webcache!enabled**の設定に関わらず、Traffic Managerは常に結合された画像などの最適化されたリソースをキャッシュに保存します。ただし、この設定が有効化されない限り、Traffic Managerは、Web Content Acceleratorにより生成された、最適化されたHTMLページのキャッシングを試みません。キャッシングとWeb Content Acceleratorの両方が有効化されている場合、Traffic Managerは、Web Content Acceleratorが取得する最適化されていないリソースもキャッシングします。

ライフタイムの設定

Traffic Managerは、頻繁にリクエストされるコンテンツを自動的にキャッシングして、Webキャッシュを最大限に活用します。

Webキャッシュを使用する各仮想サーバについて、「古くなった」リソースを破棄するまでのキャッシング時間を設定することができます。この目的のために以下の設定を使用します。

設定	説明
webcache!time	リソースをキャッシングしておく時間(秒単位)
webcache!errorpage_time	エラーページをキャッシングしておく時間(秒単位)
webcache!refresh_time	リソースが期限切れになるどのくらい前にTraffic Managerがリソースの更新を開始するか

Traffic Managerは、通常のHTTPレスポンスを、**webcache!time**に設定された時間だけキャッシングしておきます。リソースは、リモートクライアントから初めてリクエストされたときにキャッシュにロードされ、その後、期限切れになるまで同じリソースに対する以降のリクエストはキャッシュから直接提供されます。



バックエンドサーバでリソースが変更された場合、Traffic Managerがその変更を検出してキャッシングされたコピーを更新するまでに**webcache!time**秒かかることがあります。古いコンテンツをキャッシュから強制的に破棄する方法については、[キャッシュ内の古いコンテンツの削除](#)を参照してください。

Traffic Managerは、エラーページ(404 Not Found、410 Gone、501 Not Implementedなど)もキャッシングしますが、バックエンドサーバに対してこれらのページを再度リクエストするまでのキャッシング時間は、**webcache!errorpage_time**設定で決まります。一般に、エラーページは長時間キャッシングせず、より頻繁にコンテンツを再確認して、エラーメッセージを消去して正しいコンテンツに置き換えるようにします。



Traffic Managerは、サーバエラーの結果として返される「503 Too Busy」などのページはキャッシングしません。

コンテンツの期限切れ

Traffic Managerは、キャッシュされたコンテンツが期限切れとなっても自動で削除されません。キャッシュがフルになり、新たなコンテンツ用にスペースが必要となった場合、Traffic Managerは使用されていない期間が最も長いコンテンツを削除することでスペースを確保します。これは、未使用期間が最も長いわけではないが、削除しても良い期限切れのコンテンツがキャッシュに含まれていても関係ありません。

リソースが期限切れになると、完全なリソースが再度キャッシュにロードされるまで、そのリソースに対するリクエストはバックエンドサーバに送信されます。同じリソースに対して多数のリクエストが同時に発生するか、リソースの生成に時間がかかる場合、バックエンドサーバへのトラフィックが急増する可能性があります。

この問題は、**webcache!refresh_time**設定により解決できます。この値は、リソースの期限が近づいたときにTraffic Managerが推測でキャッシュを更新する方法をコントロールします。

例えば、**webcache!refresh_time**が10秒に設定されているとします。この場合、リソースが期限切れになる最後の10秒間にTraffic Managerがそのリソースに対するリクエストを受信すると、Traffic Managerはキャッシングされているレスポンスを使用せずに、リクエストをバックエンドサーバに転送します。バックエンドサーバから有効なレスポンスが返された場合、このレスポンスによってキャッシュが更新され、期限切れカウンタがリセットされます。

webcache!refresh_time期間にTraffic Managerが各リソースについてバックエンドサーバに送信するリクエストは、1秒あたり最大1回のみです。同じリソースに対する他のリクエストは、キャッシングされたレスポンスを使用して満たされます。このように、キャッシュを期限切れ前に更新できるため、バックエンドサーバに大量のリクエストが送信されることはありません。

Webキャッシュのメモリ使用量の設定

Webキャッシュのメモリ使用量を設定するときは、以下の要件が満たされるようにすることを推奨します。

- 最も頻繁にリクエストされるページの全てのバリエーションを保持できる十分な容量のキャッシュにすること
- ホストマシンが実行中のTraffic Managerプロセスの一部をディスクにスワッピングするほどキャッシュが大きすぎないこと
- 全ての仮想サーバが同じグローバルキャッシュメモリを共有し、サイズはTraffic Manager用としてグローバルに設定されます。

Webキャッシュのメモリサイズを設定するには、**System** (システム) > **Global Settings** (グローバル設定) > **Cache Settings** (キャッシュ設定) クリックします。以下の設定が適用されます。

設定	説明
webcache!size	Webキャッシュが使用できるメモリ容量の上限(Mb単位)です。合計システムメモリのパーセンテージで指定することもできます。
webcache!max_file_num	キャッシュに保存できる最大エントリ数を制限します。
webcache!max_file_size	キャッシュに挿入される最大ファイルサイズを指定します。キャッシュの合計サイズのパーセンテージ、または絶対的なファイルサイズで指定することができます。
webcache!avg_path_length	キャッシングされるリソースに対して推定されるパスの平均的な長さ(クエリ文字列を含む)です。この数字に相当するメモリ容量を webcache!max_file_num 倍した容量が、キャッシュエントリ用のパスを保存するために割り当てられます。
webcache!max_path_length	パスをキャッシュに保存するときの、パスの最大長です。1つで必要以上に保存スペースを使い尽してしまうような、非常に長いURLを阻止するための設定です。
webcache!verbose	この設定を有効化すると、リクエスト/レスポンスをキャッシング可能かどうかを示す「X-Cache-Info」というヘッダがHTTPレスポンスに追加されます。
webcache!normalize_query	クエリ文字列の正規化を有効化します。
webcache!disk	ディスクベースのキャッシュの使用を有効化します。 ディスクベースのキャッシングの設定 を参照してください。

設定	説明
webcache!disk_dir	ディスクキャッシングが有効化されている場合に、ディスクキャッシュファイルを保存するディレクトリを設定します。

キャッシュアクティビティのモニタリング

Webキャッシュの活動を監視するには、**Activity** (アクティビティ) > **Content Cache** (コンテンツキャッシング) をクリックします。このページには、Webキャッシュのコンテンツと、保存されている各レスポンスのバリエーション数が表示されます。Content Cache (コンテンツキャッシング) ページには、Webキャッシュで使用されているメモリ容量も示されます。

Virtual Server (仮想サーバ) > **Edit** (編集) > **Content Cache** (コンテンツキャッシング) ページの **View cache contents** (キャッシュコンテンツの表示) をクリックしても、このページを表示できます。

Content Cache (コンテンツキャッシング) のアクティビティページには、割り当てられたエントリのパーセンテージ、現在使用中のメモリ、およびセクタが表示されるため、キャッシュのコンテンツを調べ、必要に応じて手動でキャッシュ項目を期限切れにすることができます。

キャッシュがいっぱいになるか、ほぼいっぱいになると、**Entries** (エントリ) 値が **MaxEntries** 値に近い値になるか、または **MemUsed** 値が **MemMaximum** 値に近い値になります。この状態で新しいエントリがキャッシュに挿入されると、最も使用頻度の低いエントリが期限切れ前に破棄されます。**Oldest** (最も古いエントリ) 値をモニタリングして、キャッシュ内で最も使用頻度の低いエントリの最終使用時点からの経過時間を判断することができます。

リアルタイムでキャッシュ使用のグラフを表示するには、**Activity** (アクティビティ) > **Current Activity** (現在のアクティビティ) をクリックして Traffic Manager のアクティビティ監視ページにアクセスします。「Settings (設定)」セクションで **Chart Data** (データのグラフ) を「HTTP Content Cache (HTTPコンテンツキャッシング)」に設定します。このアクティビティ情報は、SNMPからも利用できます。

ディスクベースのキャッシングの設定

一般に、Traffic Manager のキャッシュはメインメモリ内に存在し、選択するサイズは、OS、アプリケーションメモリ、ネットワークバッファなどへの割り当て後に使用可能な RAM の容量で決まります。

または、より大容量のローカルディスク上のキャッシュを使用するよう Traffic Manager を設定することもできます。これにより、OS は、システムのカーネルディスクバッファを使用して、ディスクベースのキャッシュの最もアクティブな部分をメモリに保存するため、効果的な2段階キャッシュが構成されます。

キャッシュの設定

ディスクベースのキャッシュを使用するには、以下の手順を実行します。

1. **System (システム) > Global Settings (グローバル設定) > Cache Settings (キャッシュ設定)** をクリックします。
2. キャッシュのサイズをMBまたはGB単位で設定します。このサイズは、サーバのメモリ容量の数倍になる場合があります。
3. **webcache!disk**設定を使用して、データベースのキャッシングを有効化します。
4. ディスクキャッシュの場所を選択します。この場所は、十分な空き容量のあるローカルディレクトリでなければなりません。また、より低速なローカルHDD (ハードディスクドライブ)ではなく、高速SSD (ソリッドステートドライブ)ストレージを選択することを強くお奨めします。
5. **Update (更新)**をクリックして設定を保存します。
6. データベースのキャッシュを完全に適用するには、**System (システム) > Traffic Managers**ページからTraffic Managerを再起動します。

ディスクキャッシュは永続的ではなく、Traffic Managerソフトウェアをリスタートするとクリアされ、リセットされます。ディスクキャッシュを複数のTraffic Managerシステムで共有することはできません。

SSDおよびHDDディスクキャッシュ

ディスクキャッシュは、選択した場所に作成される単一のフラットファイルに保存されます。ホストOSは、ファイルの最も頻繁にアクセスされる部分をキャッシングし、書き込みをバックグラウンドで行いますが、キャッシュに対する全ての読み取りおよび書き込みがディスクアクセスにつながる可能性があります。

ローカルSSDドライブの読み取り、書き込み、およびシークは非常に高速なため、キャッシュファイルにはこのドライブを使用することを強くお奨めします。HDDの読み取り、書き込み、およびシークはこれよりはるかに低速です。一般的な機械式HDDのディスク操作は1秒あたり数百回に制限されるため、Traffic Managerがキャッシュによって処理できる1秒あたりのリクエスト数が制限されます。

キャッシングポリシー

全てのWebコンテンツをキャッシングできるわけではありません。Traffic Managerは、Webキャッシュを使用してリクエストを処理するかどうか、レスポンスをキャッシングするかどうかを、HTTPリクエストおよびHTTPレスポンス内の情報を基に決定します。

リクエスト

キャッシングできるのは、HTTP GETおよびHEADリクエストのみです。その他全てのメソッドはキャッシングできません。

HTTPリクエストの「Cache-Control」ヘッダにより、Traffic Managerに強制的にWebキャッシュを無視してバックエンドノードと通信させることができます。

HTTP basic-auth (基本認証)を使用するリクエストはキャッシングできません。

レスポンス

HTTPレスポンスの「Cache-Control」ヘッダにより、HTTPレスポンスをWebキャッシュに保存しないよう指定することができます。

このヘッダで「max-age」値を使用して、キャッシングされたオブジェクトのキャッシング時間を指定することもできます。この場合、レスポンスのキャッシング時間が設定された**webcache!time**設定より短くなる場合があります。

HTTPレスポンスで「Expires」ヘッダを使用して、レスポンスのキャッシング時間をコントロールすることができます。「Expires」ヘッダは、Cache-Controlレスポンスヘッダで「max-age」値を使用する場合より効率が劣ります。

「Vary」HTTPレスポンスヘッダは、リソースのバリエーションのキャッシング方法と、新しいリクエストに対するレスポンスでキャッシュからどのバリエーションを提供するかをコントロールします。

WebアプリケーションでTraffic Managerがレスポンスをキャッシングしないようにするには、レスポンスに「Cache-Control: no-cache」ヘッダを追加します。

キャッシュの動作をデバッグするには、**System** (システム) > **Global Settings** (グローバル設定) > **Cache Settings** (キャッシュ設定)で、**webcache!verbose** を「Yes (はい)」に設定します。この設定を有効化すると、Traffic Managerは、キャッシュポリシーがどのように有効になっているかを示す「X-Cache-Info」というヘッダをHTTPレスポンスに追加します。

HTTPキャッシュポリシーの詳細については、RFC 2616を参照してください。

バリエーション

以下の場合、1つのURLに対して複数のバリエーションが存在することがあります。

- レスポンスが、Accept-EncodingヘッダやAccept-Languageヘッダなど追加のリクエストパラメータによって異なる場合。コンテンツ提供元のWebサーバは、レスポンスに「Vary」ヘッダを含めることによってこのようなケースに該当することを示します。

```
Vary: Accept-Language, Accept-Encoding
```

- TrafficScript関数のhttp.cache.setkey()により、レスポンスのコンテンツが他の演算パラメータに依存することが示されている場合。

バイトレンジ

バイトレンジは、ドキュメントの一部をリクエストするために一部のクライアントソフトウェアで使用されます。例えば、PDFリーダーは、PDFファイルの最初の部分(ページの索引を含む)をダウンロードしてから、バイトレンジリクエストを送信し、ユーザが読もうとしているページを読み込みます。

Traffic Managerでは、バイトレンジはURLのバリエーションとして扱われます(詳細については、[バリエーション](#)を参照)。バイトレンジレスポンスはキャッシングされ、同じバイトレンジに対する別のリクエストを受信した場合、レスポンスはキャッシュから直接提供されます。

Traffic Managerは特定のURLのレスポンス全体をキャッシングした場合、そのレスポンスによって全てのバイトレンジリクエストを処理します。

ETag

ETag (エンティティタグ)は、Webサーバが特定レスポンスの各バージョンをユニークに識別するために使用します。Traffic Managerは、ETagをキャッシングし、これを考慮します。現在キャッシングされているリソースのバージョンのETagがリモートクライアントでキャッシングされているバージョンと同じ場合、304 Not Modified レスポンスを返します。

TrafficScriptを使用したコンテンツキャッシングのコントロール

HTTPリクエストの処理

Traffic Managerは、新しいHTTPリクエストを受信したときに、レスポンスをWebキャッシュから直接提供できるか、リクエストをバックエンドサーバに転送する必要があるかを判断する必要があります。Traffic Managerは、Webキャッシュをチェックする前に、リクエストに対して全てのTrafficScriptリクエストルールを実行します。そのため、TrafficScriptリクエストルールによってリクエストを変更し、Traffic Managerによる決定に影響を与えることができます。

例えば、TrafficScriptリクエストルールにより、リクエストに「Cache-Control: no-cache」ヘッダを追加し、Traffic ManagerがレスポンスをローカルWebキャッシュから提供しないようにすることができます。

HTTPレスポンスの処理

同様に、Traffic Managerは、HTTPレスポンスをバックエンドサーバノードから受信したときに、今後のためにレスポンスをWebキャッシュに挿入するかどうかを決定する必要があります。Traffic Managerは、この決定を行う前に全てのTrafficScriptレスポンスルールを実行するため、TrafficScriptレスポンスルールによってレスポンスを変更し、この決定に影響を与えることができます。

TrafficScriptレスポンスルールにより、レスポンスに「Cache-Control: no-cache」ヘッダを追加してキャッシングされないようにしたり、「Cache-Control: max-age=60」ヘッダを追加してレスポンスを60秒間キャッシングするよう指示したりできます。

レスポンスがWebキャッシュから提供される場合、TrafficScriptレスポンスルールは実行されません。

TrafficScriptキャッシュコントロール関数

Traffic Managerのキャッシング動作をコントロールするためのキャッシュコントロール関数がいくつか用意されています。多くの場合、これらの関数を使用すると、HTTPリクエストおよびレスポンスのヘッダを操作する必要がなくなります。

関数	説明
<code>http.cache.disable()</code>	レスポンスルールで <code>http.cache.disable()</code> を呼び出すと、Traffic Managerはレスポンスをキャッシングしません。
<code>http.cache.enable()</code>	レスポンスルールで <code>http.cache.enable()</code> を呼び出すと、 <code>http.cache.disable()</code> の呼び出し前の状態に戻り、Traffic Managerのデフォルトのキャッシングロジックが有効になります。 Traffic Managerに、キャッシュ不可と見なされているレスポンスを強制的にキャッシングさせることはできません。
<code>http.cache.setkey()</code>	<code>http.cache.setkey()</code> 関数は、Varyレスポンスヘッダ関数とほぼ同様に、同じリクエストの複数のバージョンを区別するために使用します。この関数はリクエストルールで使用しますが、レスポンスルールで使用することも可能です。 送信元アドレスが内部か外部か、クライアントの「User-Agent」ヘッダが示すブラウザがIEかGeckoベースかなどの計算値に基づいてリクエストを区別できるため、RFC2616のVaryのサポートより柔軟性に優れています。

関数	説明
<code>http.cache.exists()</code>	<code>http.cache.exists()</code> 関数は、呼び出し時点でキャッシュ内に一致するレスポンスが存在しているかどうかを確認するために、リクエストルールで使用できます。ルールがリクエストパラメータを変更せずに終了した場合、レスポンスはキャッシュから提供されます(途中で期限切れにならなかった場合)。
<code>http.cache.respondIfCached()</code>	この関数は、リクエストルールから呼び出すことができます。Webキャッシュ内に一致するレスポンスが存在する場合、ルールは即座に終了し、レスポンスがリモートクライアントに返され、これ以上リクエスト(またはレスポンス)ルールは実行されません。 一致するレスポンスが存在しない場合、関数が戻り、ルール処理が続行されます。 この関数は、以下のようにバックエンドノードに送信するリクエストについてはレートシェイピングし、レスポンスをTraffic Managerのキャッシュから直接提供できる場合はレート制限を適用する必要がない場合のレートシェイピングなどのロジックをサポートするために使用できます。 <pre>http.cache.respondIfCached(); rate.use("my rate limit");</pre>

詳細およびコードサンプルについては、『*Pulse Secure Traffic Manager: TrafficScript Guide*』(IvantiのWebサイト: www.ivanti.comから入手可能)を参照してください。

キャッシュ内の古いコンテンツの削除

負荷分散されているWebサイトのコンテンツの一部を更新した場合、Traffic Managerは、キャッシングされている既存のバージョンが期限切れになったときに、新しいコンテンツのみを取り出します。

手動でTraffic Managerに強制的にキャッシュのコンテンツを期限切れにさせ、次回クライアントからそのコンテンツがリクエストされたときに更新後のバージョンを読み込むようにすることができます。または、この処理をプログラムで行うこともできます。

キャッシングされているコンテンツの手動削除

キャッシングされているコンテンツを手動で削除するには、以下の手順に従います。

1. **Activity** (アクティビティ) > **Content Cache** (コンテンツキャッシュ)をクリックして、Traffic Managerクラスタにより現在キャッシュされているファイルを表示します。
2. 必要に応じて、**Display Cache Items** (キャッシュ項目表示)セレクトを使用し、一致させるホストヘッダやパスを指定して選択を絞り込み、期限切れにする項目を見つけます。
3. 期限切れにするエントリを選択し、**Clear Selected** (選択項目をクリア)ボタンをクリックします。

これにより、これらのリソースについてキャッシングされているコピーが即座に無効になり、Traffic Managerが、次回クライアントからリクエストされたときに新しいバージョンを強制的に読み込むようになります。

キャッシングされているコンテンツのプログラムによる削除

Traffic ManagerコントロールAPIには、キャッシュ内のエントリをチェックして無効にするSOAPメソッドが含まれています。これらのメソッドはSystem.Cacheインターフェースにあります。

新しいコンテンツをWebサーバにアップロードするコンテンツプロビジョニングシステムは、Traffic ManagerコントロールAPIを使用したスクリプトを呼び出し、更新された全ての項目をコンテンツキャッシュで無効にする可能性があります。

詳細については、『*Pulse Secure Virtual Traffic Manager: Control API Guide*』を参照してください。

Pulse Secure Web Acceleratorの使用

この章では、Pulse Secure Web Acceleratorを使用してウェブコンテンツを最適化する方法を説明します。

はじめに

Webページのロードを高速化することにより、Webページが表示され、使えるようになるまでの待機時間が短くなるため、満足度の高いユーザエクスペリエンスを実現できます。研究でも明らかになっているように、平均的なWebページのロード時間が6秒の場合でもアプリケーションとWebサーバの内部で要している時間は1秒ほどに過ぎず、ユーザが待たされる残りの4～5秒はクライアントブラウザがWebページをレンダリングしている時間です。Webページのロード時間を短縮することは、ユーザエクスペリエンス全体の改善に不可欠です。

Pulse Secure Web Accelerator (Web Content Accelerator)は、ロードを高速化するためにWebページのマークアップと要素を自動的に最適化します。そのための技術はTraffic Managerインフラストラクチャに既に装備されており、バックエンドのアプリケーションに対する変更は特に必要ありません。

動作モード

Web Content AcceleratorはTraffic Managerのオプション機能であり、特別に設定されたライセンスキーで有効になります。ご使用のライセンスキーに応じて、ExpressとEnterpriseの2つの機能レベルを利用できます。

Web Content Accelerator Expressは、幅広い用途で最適に機能するように設計されており、あらかじめ定義された最適化レベルをWebコンテンツに適用します。このモードでは、Web Content Acceleratorが有効化された仮想サーバは、処理対象のWebページに含まれているHTMLを圧縮します。また、クライアントがページリソースを取得する際に、そのページリソースに対して以下の最適化を実行します。

- JavaScriptファイルの最小化および圧縮
- スタイルシートの最小化および圧縮
- バックグラウンド画像のインライン化またはバージョニング
- Webフォントのバージョニング
- 画像コンテンツの再サンプリング
- 全てのリソースの圧縮

各項目の詳細については、[最適化手法について](#)を参照してください。

注記

Expressモードでは、認証を必要とするサイトを最適化するためにWeb Content Acceleratorを使用することはできません。詳細については、[認証](#)を参照してください。



Web Content Accelerator固有のTrafficScript関数は、Expressモードのデフォルトパラメータで機能するように制限されており、オーバーライドすることはできません。詳細については、『*Pulse Secure Virtual Traffic Manager: TrafficScript Guide*』を参照してください。

Enterpriseライセンスを使用すると、Web Content Acceleratorの全ての最適化手法を実行できるようになり、Web Acceleratorを使用してHTMLページのコンテンツをチェックしたり、変更したりできるようになります。また、Web Content Accelerator Enterpriseでは、ご使用のWebインフラストラクチャに適用される個々の最適化パラメータを完全にコントロールできます。ご使用の仮想サーバが処理する個々のアプリケーションスコープに対して、固有の最適化プロファイルを作成して適用できます。

本章では、Web Content AcceleratorをEnterpriseモードで実行する場合の特徴と機能について詳細に説明します。一部の要素は、Expressモードではご利用いただけません。

サービス用のWeb Content Acceleratorの設定

管理UIの**Virtual Servers** (仮想サーバ) > **Edit** (編集) ページでWeb Content Accelerator リンクをクリックすると、Web Content Acceleratorを仮想サーバのプロパティとして有効化できます。

Basic Settings (基本設定) セクションで**optimizer!enabled** 設定キーを使用して、仮想サーバのWeb Content Accelerator機能を完全に有効化または無効化できます。



Web Content Accelerator Expressを実行している場合、このセクションで以下に説明するUIコントロールを利用することはできません。参考情報としてお読みください。詳細については、サポート窓口にお問い合わせください。

仮想サーバは、指定されたポートとプロトコルのトラフィックを管理します。仮想サーバが処理するリクエストは、リクエストされたURL (<http://www.mywebsite.com/supportsystem> など) に基づいて特定のアプリケーションに向けることができます。このホスト名とパスの組み合わせを利用すると、(仮想サーバを介した) リクエスト先のアプリケーションと、その識別に使用すべき最適化プロファイルとの間にコネクションを作成できます。

Web Content Acceleratorの「Profiles」セクションでは、使用する各種のアクセラレーションプロファイルと、適用される「スコープ」との間のマッピングを定義します。プロファイルとスコープは**カタログオブジェクト**であり、このページで使用する前に作成する必要があります。これらのコンセプトの詳細については、以降のセクションを参照してください。

必要に応じて、複数のスコープを定義して1つのプロファイルにマップすることができますが、1つの仮想サーバ内で一度に使用できるスコープは1つのみです(多対1の関係)。

Web Content Acceleratorウィザード

他のウィザードと同様に、Traffic Managerには、WebサービスでWebコンテンツの最適化を簡単に設定できる**Optimize a Web application** (Webアプリケーション最適化)ウィザードという自動プロセスが用意されています。

このウィザードを実行するには、あらかじめ少なくとも1つの仮想サーバが作成されている必要があります。仮想サーバがない状態でウィザードを実行することはできません。ウィザードを実行できる適切な仮想サーバを作成するには、まず**Manage a new service** (新規サービスを管理)ウィザードに従うことをお勧めします。

ウィザードを実行するには、ナビゲーションバーの「Wizards (ウィザード)」ドロップダウンメニューから**Optimize a web application** (Webアプリケーション最適化)を選択します。

このウィザードには、以下の3つの主なステップがあります。

1. 最適化を適用する仮想サーバを選択します。
2. アプリケーションのスコープを選択します。
3. そのスコープに適用するWeb Content Acceleratorプロファイルを選択します。

アプリケーションスコープ

アプリケーションスコープは、仮想サーバでホストされている特定のWebサイトまたはアプリケーションに一致するURLの集合を定義するためのオブジェクトです。新しいアプリケーションスコープの表示、変更、および追加を行うには、**Catalogs** (カタログ) > **Web Accelerator Catalogs** (Web Acceleratorカタログ) > **Application Scopes catalog** (アプリケーションスコープカタログ)をクリックします。。

仮想サーバを介してリクエストが受信されると、**Virtual Server** (仮想サーバ) > **Edit** (編集) > Web Content Acceleratorで定義されているスコープのマッピング内で最も限定的なURLとの一致が検索されます。Traffic Managerで一致が見つかり、マッピングに定義されている適切な最適化プロファイルが適用されます。

1つのスコープレコードには、複数のホスト名を含めることができます。ただし、これらの名前は全て同じホストのエイリアスであるため、それを使用する同じリソースが取得される可能性があります。別のホスト名を使用してコンテンツ配信ネットワークからのコンテンツを提供している場合は、コンテンツがTraffic Manager内で同じアプリケーションスコープを通過するように、そのCDNが元ホストのプロキシとして機能する必要があります。

「任意」のホスト名と一致させるには、新しいスコープレコードを作成し、**hostnames**リストをブランクのままにします。あるいは、特定の仮想サーバで複数のアプリケーションがホストされ、リクエストのURLで区別されることがあります。この場合、関連するホスト名とルートパスの組み合わせごとに新しいスコープレコードを設定します。

仮想サーバを介して提供されるリソースは、設定されたスコープで宣言されているホスト名と、その仮想サーバが実行しているポートの両方がそのリソースと一致する場合にのみ、最適化の対象であると見なされます。ホスト名の一致は、スコープで指定されているホスト名を使用して、またはスコープのホスト名設定がブランクの場合はページリクエストのホスト名を使用して実行されます。依存リソースのリクエストには、明示的に設定されている場合、必ずプライマリホスト名(リストの先頭)が使用されます。

外部メディアホストのように異なるホスト上にあるリソースを最適化するには、**Applies to externally hosted resources** (外部ホストリソースに適用)オプションを有効にしてURLを含むWeb Content Acceleratorプロファイルを使用します。詳細については、[アクセラレーション設定](#)を参照してください。

ホスト名が指定されているアプリケーションスコープを、その名前前でアプリケーションをホストしていない仮想サーバに適用すると、リクエストが502ステータスで失敗することがあります。このHTTPステータスが表示された場合は、アプリケーションスコープのホスト名が、その名前の表すアプリケーションによって論理的にグループ化されていることを確認してください。



複数のスコープによって、一致するURLが競合しないようにしてください。

新しいスコープエントリを作成するときに、さらにスコープのアプリケーションへのルートパスを指定するには、**root**設定キーを使用します。

内蔵のスコープは、「Any hostname or path」という名前です。自動的に指定され、Webアプリケーション内で特定のホスト名やパスの指定が必要ない場合に、これを使用できます。**hostname** (ホスト名)のフィールドはブランク、**root** (ルート)のフィールドは「/」にそれぞれ設定します。

Web Content Acceleratorプロファイル

Web Content Acceleratorプロファイルカタログには、コンテンツの自動最適化を有効化するためにWebサイトとWebアプリケーションに適用できるプロファイルのリストが含まれています。新しいWeb Content Acceleratorプロファイルの表示、変更、および追加を行うには、**Catalogs** (カテゴリ) > **Web Content Accelerator Catalogs** (Web Content Acceleratorカタログ) > **Web Content Accelerator Profiles catalog** (Web Content Acceleratorプロファイルカタログ)をクリックします。

内蔵プロファイルとカスタムプロファイル

Traffic Managerには、あらかじめ定義された最適化設定をWebアプリケーションに適用する多数の「内蔵」プロファイルが用意されています。これらのプロファイルは変更も削除も不可能ですが、設計されている最適化設定は、ほとんどの状況に適用可能です。以下のプロファイルを選択できます。

Name	説明
Express	幅広いWebサイト用に設計された、最適化の基本的な設定および手法を提供するプロファイルです。CSSの圧縮、CSSバックグラウンド画像のインライン化、JavaScriptの最小化、画像の圧縮、高度なキャッシングなどの最適化手法があります。
SharePoint 2007	Microsoft SharePoint 2007 Internetおよびイントラネットのサイトのアクセラレーションと互換性のために調整された最適化設定を提供するプロファイルです。リソースURLのバージョンング、持続性のあるキャッシング、CSSスタイルシートのインライン化、CSSの最小化、バックグラウンド画像のインライン化、JavaScriptの最小化、圧縮、画像の圧縮およびメタデータ削除、高度なJavaScript互換性などの最適化設定があります。
SharePoint 2010	Microsoft SharePoint 2010 Internetおよびイントラネットのサイトのアクセラレーションと互換性のために調整された最適化設定を提供するプロファイルです。リソースURLのバージョンング、持続性のあるキャッシング、CSSスタイルシートのインライン化、CSSの最小化、バックグラウンド画像のインライン化、JavaScriptの最小化、圧縮、画像の圧縮およびメタデータ削除、高度なJavaScript互換性、SharePoint!リボンの最適化などの最適化設定があります。
SharePoint 2013	Microsoft SharePoint 2013 Internetおよびイントラネットのサイトのアクセラレーションと互換性のために調整された最適化設定を提供するプロファイルです。ダウンロード最小化戦略(MDS、Minimal Download Strategy)というSharePointの機能をサポートしています。ページ間を移動する際のホワイトスクリーンの影響が低減されたり、転送されるデータが低減されたりするように設計されています。 ブラウザでページ間を移動するとき、前のページとは異なるHTMLのフラグメントのみがHTTPページのレスポンスに含まれます。MDSが有効化されていない場合、各ページのレスポンスにはページ全体のコンテンツが含まれます。デフォルトでは、MDS機能は有効化されていますが、サイトの「ルック&フィール」がカスタマイズされると無効化されます。 SharePoint 2010のプロファイルとは以下の違いがあります。 Web Content AcceleratorがMDSのフラグメントをサポートし、最適化します。

Name	説明
	SharePoint 2013は多くの独自スクリプトをロードします。Web Content Acceleratorは、あらかじめ設定されたスクリプトセットを使用してスクリプトコンテンツを提供することにより、この動作を最適化します。例えば、SharePointが「/_layouts/15/init.js」を要求している場合、Web Content Acceleratorは、それを含むスクリプトセットと、通常ロードされるその他のスクリプトを同時に提供します。SharePointがそれらのスクリプトのうちの1つを必要とする場合、それは既にWeb Content Acceleratorのスクリプトセットの一部としてブラウザキャッシュに存在しているため、即座に使用することができます。
SharePoint 2013 Custom Website	SharePoint 2013のユーザインターフェースは大幅にカスタマイズすることが可能であり、デフォルトのSharePoint 2013サイトが使用するデフォルトのスクリプトやスタイル、画像の全てが使用されないこともあります。 このプロファイルは、標準の「SharePoint 2013」プロファイルを改造したもので、ブラウザがダウンロードするSharePointのデフォルトのスクリプトやリソースセットを必要としない、カスタマイズされたWebサイト用に設計されています。 デフォルトのSharePointサイトには、このプロファイルではなく「SharePoint 2013」プロファイルを使用してください。
Other Web application	最適化の基本的な設定を提供するプロファイルで、ほとんどのWebアプリケーションを安全に高速化することができます。CSSスタイルシートのインライン化、CSSの最小化、バックグラウンド画像のインライン化、JavaScriptの最小化、圧縮、画像の圧縮およびメタデータ削除などの最適化設定があります。このプロファイルは、Webアプリケーションの最適化要件にカスタマイズされた新しいプロファイルを作成する際に、テンプレートとして使用できます。
Basic (廃止予定)	幅広いWebサイトで適切に機能する、十分な数の基本的な最適化手法を有効化するプロファイルです。CSSの圧縮、CSSバックグラウンド画像のインライン化、JavaScriptの「最小化」、画像の圧縮、URLのバージョンング、高度なキャッシングなどの最適化手法があります。

Name	説明
Advanced (廃止予定)	Basic (基本)プロファイルの機能に加え、CSSスタイルシートやHTML画像を画像スプライトに組み合わせる、高度な最適化手法が有効になるプロファイルです。これらの手法でページのロード時間は大幅に短縮されますが、一部のサイトとの互換性を保つにはカスタマイズしたルールが必要になることがあります。この場合、カスタムのWeb Content Acceleratorプロファイルを作成するにはプロファイルを複製します(後述)。

これらのプロファイルで、Webサイトの全ての要素に対応する適切なレベルの最適化が得られない場合には、個別の要件を満たすカスタムのWeb Content Acceleratorプロファイルを新しく作成することができます。

カスタムプロファイルを作成するには、「Create a new Web Content Accelerator Profile (Web Content Acceleratorプロファイルの新規作成)」セクションを使用して、テンプレートとしていずれかの内蔵プロファイルを指定します。必要なテンプレートプロファイルをドロップダウンリストから選択し、**Create Web Content Accelerator Profile (Web Content Acceleratorプロファイルの作成)**をクリックしてください。**Edit Profile** (プロファイルの編集)ページで**Save As**を使用し、内蔵プロファイルを複製してカスタムプロファイルを作成することもできます。

カスタムプロファイルには、一連のアクセラレーションオプションが追加されています。詳細については、[カスタムアクセラレーションプロファイルについて](#)を参照してください。

最適化モードとページ情報

Web Content Acceleratorは3つのモードのいずれかで動作します。これは**aptimizer_mode**設定キーで定義されています。

- **Off:** アクセラレーションは無効ですが、Web Content Acceleratorリソースへのリクエストが予約されます。
- **Stealth:** アクセラレーションは無効ですが、以下のクエリ文字列コマンドを使用して有効にする場合は例外です。

```
?aptimizer=on
```

- **On:** 全てのリクエストに対してアクセラレーションが有効です。

Web Content Acceleratorプロファイルを介して渡されるWebページごとのリクエスト情報は、**Edit Profile** (プロファイルの編集)ページで**show_info_bar**設定を使用すると表示できます。これが「Yes (はい)」に設定されていてアクセラレーションが現在アクティブな場合、このセッションの全てのページにステータスバーが表示され、そのリクエストに固有の情報を提供するコントロールが示されます。

バックグラウンドでの最適化

大きな画像セットや、ネスト化された複数のインポートを伴うCSSファイルなど、複雑なリソースの最適化には時間がかかることがあります。Web Content Acceleratorは、この処理を必要なリソースに対して1回のみ行いますが、その結果、この最適化されたリソースを初めて必要とするクライアントにとっては、その処理が完了するまで待機しなければならないことがあります。

このセクションでは、最適化されたデータの準備が整うまで、コンテンツを要求した全てのクライアントに対して最適化されていない元のサーバコンテンツを提供しながら、最適化の処理をバックグラウンドで完了させるようにWeb Content Acceleratorを設定することができます。

 結合された画像セットなどのリソースは、最適化された後で自動的にキャッシングされます。Webページは、仮想サーバの**webcache!enabled**設定が「Yes (はい)」の場合にのみキャッシングされます。ただし、そのページの全てのリソースが既に最適化されている場合、完全に最適化されたページがすぐに提供されます。

Traffic Managerには、リソースをバックグラウンドで最適化するかどうか、およびどのように最適化するかを決めるための、2つのプロファイル設定が用意されています。

設定	説明
background_after	最適化がバックグラウンドに移され、元のサーバコンテンツがクライアントに提供されるまでの時間をミリ秒単位で指定します。 0に設定すると、Web Content Acceleratorは常に最適化の完了を待ってから、クライアントにレスポンスを送信します。 JavaScriptファイルなど、ファイルによっては最適化された状態で提供されなければなりません。そのため、そのようなファイルに関しては、この設定を使用してバックグラウンドで最適化することはできません。
background_on_additional_resource	あるページや、ネスト化されたインポートを伴うCSSドキュメントなどのリソースの最適化処理を完了するために、Web Content Acceleratorが追加リソースを取得する必要がある場合、バックグラウンドで最適化が行われます。

Web Content Acceleratorによる変化の測定

このセクションでは、各種のツールを使用してWeb Content Acceleratorをテストし、Webページのパフォーマンスを測定する方法について説明します。この測定を利用すると、Web Content AcceleratorによるWebサービスのアクセラレーションの効果を判定することができます。

Web Content Acceleratorがアクティブであることのチェック

Webコンテンツが高速化されているかどうかをチェックするには、Webページのロード後にブラウザの「HTMLソース表示」オプションを使用します。次のコメントを検索します。

```
<!-- Optimized for speed - Pulse Secure Web Accelerator -->
```

このコメントが見つからない場合には、[サービス用のWeb Content Acceleratorの設定](#)を参照してください。

Stealthモードを使用してWeb Content Acceleratorをテスト

開発環境やテスト環境を使用できない場合でも、Stealthモードでアクセラレータを実行してWeb Content Acceleratorをテストすることができます。



Web Content Accelerator ExpressではStealthモードを利用できません。

Stealthモードの場合、Web Content Acceleratorはクエリ文字列内でコマンドが渡されない限り、Webページに最適化を適用しません。Webページの動作や表示について、エンドユーザには何の変化も見えません。

StealthモードでWeb Content Acceleratorを実行しているとき、ページやオブジェクトを要求して受信されるリクエストは全て、元のフォームのまま保存されます。リクエストのクエリ文字列にaptimizer=onというパラメータが含まれていると、Web Content Acceleratorはこのブラウザセッションに対する全てのリクエストにアクセラレーションを明示的に指示するクッキーを送信します。

受信リクエストに対するクッキーの有効化がWeb Content Acceleratorで認識されると、そのスコープに適用されるWeb Content Acceleratorプロファイルに定義されているルールと設定に従って、リクエストが傍受され高速化されます。

有効なリクエストの例を次に示します。

```
www.example.com?aptimizer=on  
www.example.com/destinations?aptimizer=on
```


Web Content AcceleratorをStealthモードで実行するように設定するには、適切なプロファイルで **optimizer_mode** 設定キーを変更してください。

Webページの速度の測定

Web Content Acceleratorを有効化してから何が変わったのかを正確に把握するには、その前後でWebページのロード時間を測定する必要があります。

最初の測定をベースラインとして使用し、後続の測定をそれと比較します。ベースラインは、まだWeb Content Acceleratorを有効にせず、またはStealthモードでWeb Content Acceleratorを実行せずにWebサイトを運用しているときに記録するのが理想的です。

各段階で新しい測定のセットを取得し、ベースラインと比較することができます。各段階は、Web Content Acceleratorの配備における以下のような任意のマイルストーンにすることができます。

- デフォルト設定の初期配備
- ルールオプションに対する変更
- CSS設定に対する変更
- キャッシング動作に対する変更
- デフォルトの画像セットに対する変更

ツール

Webページのロード時間を検証し、Webページ構造にフィードバックを送る際には、様々なツールを利用できます。ごく細かいレベルで使用して個々のWebページをテストできるツールもあれば、Web Content Accelerator対応の実装に相関付けることのできるメトリクスを集めるツールもあります。

ページの構造と速度

Webページのテストやその情報を提示することは、多くのツールで可能です。オンラインサービスとして利用できるものや、FirefoxやChromeなどのWebブラウザのアドインとして利用できるものがあります。厳密な要件は異なるかもしれませんが、代表的な例を以下に挙げます。

- WEBPAGETEST (www.webpagetest.org)は、ブラウザの動作をエミュレートしてWebページを取得する無償のWebサイトサービスです。Webページを解析してその結果をカスケードスタイルのチャートで示します。

- これによりレンダリングされるページをブラウザがどう処理しているかがわかります。サイトを可能な限り高速にするには、チャート上のオブジェクトのリストを手順どおりに進んで各オブジェクトに適用する最適なアクセラレーションを決定すると簡単です。例えば、チャートの初めに2つのリダイレクトとページのロードが表示されているとします。この場合に理想的な対応は、リダイレクトが使われないようにサイトを再構築することです。あるいは、チャートの下部にあるページにロードされるスクリプトを確認します。
- YSlow (<http://developer.yahoo.com/yslow/help>)は、Yahoo! デベロッパーネットワークプロジェクトによって開発されたブラウザアドインです。内蔵ルールを使用し、様々なパフォーマンス指標に関してWebページをA (最高)からF (最低)までランク付けし、Webベーススコアの改善方法を提案します。FirefoxとChromeに対応しています。
- Google Page Speed (<http://code.google.com/speed/page-speed>)も、FirefoxとChromeで利用できるブラウザアドインです。0 (最低)から100 (最高)までのPage SpeedスコアでWebページにランクを付けます。Google Page Speedには、Webページスコアの改善方法を提案する機能もあります。

その他のメトリクス

メトリクスによっては、長期間にわたってデータを収集しないと実用的でないものもあります。つまり、Web Content Acceleratorを有効化するより前に収集を開始しなければならないということです。

- Google Analytics (www.google.com/analytics)は、ページのロード時間、ページビューの数、訪問ごとのページビュー、直帰率などのメトリクスを収集できる無償のツールです。Google Analyticsを使用するには、Webサイトの全てのページに小さいスクリプトを追加する必要があります。結果は、このスクリプトをインストールした数日後から閲覧できますが、一定期間データを収集しないと実効的な比較はできません。
- Google Webmasters (www.google.com/webmasters)も無償のツールで、GoogleでWebサイトがどのように表示されるかを管理することができます。特典として、Webサイト全体のグローバルなWebページ平均ロード時間を示すチャートが、6か月間のローリングウィンドウで表示されます。Google Webmastersにはサイト所有者の認証が必要です。これは、DNSを一部変更するか、Webサーバに設定ファイルを追加する、またはメタタグをデフォルトのWebページに追加して実行できます。

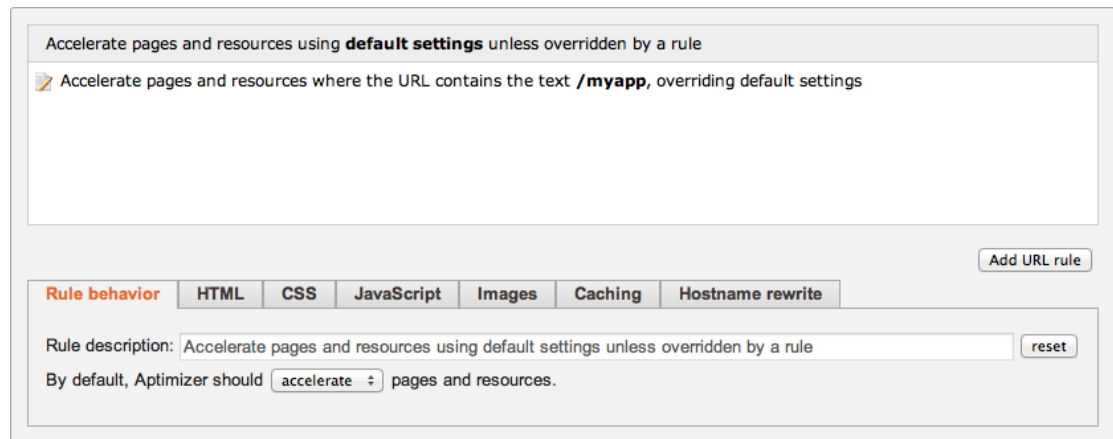
カスタムアクセラレーションプロファイルについて



Web Content Accelerator Expressではこのセクションを利用できません。

通常は、内蔵のWeb Content Acceleratorアクセラレーションプロファイルだけで、Webアプリケーションの汎用的な最適化は十分に可能ですが、Webサイトやアプリケーションの特定の要素に対して最適化をさらにきめ細かくコントロールするために、カスタムのアクセラレーションプロファイルが必要になる場合もあります。プロファイルの概要については、[Web Content Acceleratorプロファイル](#)を参照してください。

カスタムアクセラレーションプロファイルの**Edit** (編集) ページには、**Acceleration Settings** (アクセラレーション設定) ダイアログという追加のコントロールがあります。



ここには、Webアプリケーションで各種のコンテンツタイプの最適化を処理するコントロールとルールがあります。以下の2つの部分に分かれています。

- 上部には、Webコンテンツに適用されるルールのリストが、通常はURLロケーションまたはブラウザタイプに基づいて階層表示されます。
- その下に並んでいるタブには、現在のルールに適用されるプロパティが表示されています。特定のルールをハイライトすると、下のタブセットでそのプロパティが表示されます。

それぞれのアクセラレーションプロファイルには**default settings** (デフォルト設定) ルールがあり、これは削除できません。これを利用すると、プロファイルのデフォルト動作を定義し、プロパティの各タブのフォールバック値を設定することができます。この下に新しいルールを追加すると、最適化の動作をさらに定義したり、特定のURL (**Add URL rule** (URLルールの追加) をクリック) に基づいてデフォルトのプロパティをオーバーライドしたりできます。各ルールバーの右にある上下の矢印をクリックすると、新しく追加したルールをプライオリティまたは目的の効果に従って並べ替えることができます。右の削除ボタンをクリックすると、完全に削除できます。

ダイアログのルールプロパティのセクションに表示されている各タブには、CSS、JavaScript、画像など一般的なWebアプリケーションを構成するコンテンツに関係する設定があります。最初のRule behavior (ルールの動作) タブには、このルールの一般的な説明が記載され、スコープ内の全体的な動作について、このルールで定義されているコンテンツを高速化するのか除外するのかを管理します。

プロファイルのデフォルトルール

機能制限があるモバイルブラウザで互換性を保つために、拡張と基本のどちらのプロファイルにも、各ブラウザで一部のWeb Content Accelerator機能を選択的に無効化する設定があります。

具体的には、以下のブラウザで「画像の結合」、「スタイルシートの結合」、「CSSによる画像のインライン化」がそれぞれデフォルトで無効になります。

- IE mobile 7
- Opera mini
- Opera mobile
- Blackberry

また、ページのキャッシュ版が全てのブラウザに対応する汎用となるように、検索クローラエージェントに対しては特定の最適化が無効になっています。以下のクローラユーザエージェントでは、「画像の結合」と「CSSによる画像のインライン化」が無効です。

- bingbot
- googlebot



これらのプロファイルのデフォルトルールは、Web Content Accelerator設定のカスタムルールビューには表示されません。

デフォルト動作のオーバーライド

リストに追加した新しいURLルールにはそれぞれ独自のプロパティが設定されており、条件が一致すればデフォルトの動作をオーバーライドします。各プロパティタブには**Override** (オーバーライド)チェックボックスがあり、プロパティタブごとにこの機能を有効化または無効化することができます。特定のルールについて別の設定を指定したい場合には、Override (オーバーライド)チェックボックスをクリックしてそのタブの設定を有効化してください。デフォルトでは、Override (オーバーライド)チェックボックスを選択するまで全ての設定は無効で、淡色表示されています。

ルールのチェーン化

一定の状況におけるデフォルトの動作は、各リクエストがURLルールのリスト中を進んでいき、一致が見つかった時点で止まるようになっています。このような一致時に停止の動作は、URLルールで**Allow settings for this rule to chain to other rules** (このルールの設定に他のルールとのチェーン化を許可)チェックボックスを選択してオーバーライドすることができます。

このチェックボックスを有効にすると、一致したリクエストがそのルールで止まらず、2番目のルール一致が見つかるまで検索を続行します。ただし、この設定が有効なのはそれ以降の一致が見つかるまでです。その時点で、2番目のルールにチェーン化が設定されていない場合、リクエストはそれ以上のルールの一致を試行しません。全てのルールをプロファイル全体でテストしたい場合は、一致の数にかかわらず、それぞれでチェーン化を有効にする必要があります。

アクセラレーション設定

最適化の設定は、グループごとに別々のタブに表示されます。各タブは以下のとおりです。

Rule behaviourタブ

設定	説明
Rule Description	現在のルールの説明テキストです。このタブに含まれる基本ルールの動作設定から自動的に作成されます。例えば、URL test文字列または rule modeを変更すると、このフィールドの説明テキストに自動的に反映されます。このテキストを独自の説明に置き換えることもできますが、その場合は自動更新が行われなくなります。 自動生成されたデフォルトの文字列に説明をリセットするには、reset (リセット) ボタンをクリックしてください。
Rule mode	このルールで識別される全てのページとリソースについて accelerate (高速化) か exclude (除外) を設定できます。 accelerate (高速化) モードがデフォルトで、一致するリクエストのルールプロパティに定義されている目的の最適化を実行します。 もうひとつのルール設定では、指定されたURL testに一致するページまたはリソースについて最適化を除外します。 3つ目の strip (ストリッピング) モードに設定されている場合は、一致が見つかったところでリソース参照がストリッピングされます。

設定	説明
URL test	指定したテキストに一致するリクエストにのみルールが適用されます。テキストは、Match external resources (外部リソースに一致)ボックスをチェックしている場合にはホスト名を含む完全なURLと比較され、それ以外の場合にはURLパスのみと比較されます。URLに含まれる必要がある部分文字列や、URL全体に一致するワイルドカードまたは正規表現を指定できます。 正規表現を使用する前に、正規表現の使用で説明するセキュリティに関する注意事項を確認しておいてください。
Applies to externally hosted resources	Webアプリケーションの外部で保持されているリソースに対する照合が可能になります。この設定を有効にすると、ルールはアプリケーションのスコープに含まれないホストに対するリクエストにのみ一致し、URL testフィールドに含まれる文字列が、リクエストに含まれるホスト名とパスに対して照合されます。例えば、Webサイトをホストしている外部メディアから取得されるWebページ上のメディアファイルは除外したい場合があります。そのような場合には、このオプションを選択し、除外するメディアファイルのURLパスに加えてホスティングサイトの名前も含むワイルドカード文字列をURL testフィールドに指定することができます。
ルールのチェーン化	このルールを設定を他のルールにチェーン化することができます。詳細については、前述の「ルールのチェーン化」セクションを参照してください。

CSSタブ

設定	説明
CSS and Stylesheet optimization	None: スタイルシートは、どこで表示される場合でも元の形のまま維持されます。 Optimize CSS Content: ページと外部スタイルシートのスタイルシートブロックで、冗長な空白とコメントが全て削除されます。

設定	説明
	...and combine consecutive stylesheets: 連続参照として表示されるスタイルシート(間にインラインのスタイルブロックや、除外されたスタイルシートリンクない)が結合され、結合された参照のそれぞれがドキュメントで出現する最初のスタイルシートの場所に挿入されます。 ...and combine all stylesheets: ページでの出現場所にかかわらず、全てのスタイルシートが結合されます。結合された新しい参照は、ドキュメントのヘッドセクションの先頭に挿入されます。
CSS background image optimization	base64でエンコードしたデータをdata:uri表記またはMHTML参照を使ってインライン化する際の、許容最大サイズ(バイト単位)です。この上限を超える画像は、URLのままになります。

JavaScriptタブ

設定	説明
JavaScript optimization	Enabled (有効化): スクリプトファイルで、冗長な空白とコメントが全て削除されます。 Disabled (無効化): スクリプトファイルは元の形のまま維持されます。

Imagesタブ

設定	説明
Combine images	Enabled (有効化): HTMLでタグとして示される画像が、以下のいずれかの手法で結合されます。 CSS image spriting (CSS画像スプライト化): 画像は、画像フォーマットに従って1つの画像スプライトに結合されます。 Image streaming (画像ストリーミング): 画像は、1つ以上の画像ストリームに結合されます。 Disabled (無効化): HTMLでタグとして示される画像は結合されません。

設定	説明
Shrink images	メタデータや冗長な色情報などの不要な情報を削除し、JPEG画像の場合には画質を下げてファイルサイズを小さくすることによって、可能な場合に画像を圧縮します。 画像がBMPフォーマットでWebページに表示される場合、サイズが小さくなるようにTraffic Managerによって自動的にJPEGフォーマットに変換されます。JPEGフォーマットへの自動変換は、ルールを使用して画像を処理対象から除外することにより避けることができます。
Resample quality	データ損失を伴う画像を再サンプリングする際の画質レベル(0~100%)です。画像の画質維持を優先する場合には、この値を100に設定してください。 デフォルトは85です。
Convert images	画像が、元とは異なるフォーマットに最適化されるようになります。フォーマットは、明示的に選択するか「Optimal (最適)」に設定することができます。後者に設定した場合、各画像に最適なフォーマットが自動的に選択されます。
Resize images	画像サイズが、ピクセル数で指定した幅と高さに変更されます。

Cachingタブ

設定	説明
Honor resource Cache-Control header	Auto (自動): リソースのコンテンツタイプに基づき、Web Content Acceleratorがリソースをキャッシングするかどうかを決定します。スタイルシートや画像などの静的リソースは常にキャッシングされるのに対し、JSONフォーマットのAPIレスポンスなどの動的リソースは、レスポンスの「Cache-Control」の指示に従います。 Always (必ず従う): Web Content Acceleratorは、レスポンス内の全ての「Cache-Control」指示に従います。 Never (従わない): Web Content Acceleratorは、最適化した全てのコンテンツをキャッシングします。

設定	説明
Version resource URLs	Enabled (有効化): この場所にあるURLが、ファイルの内容のハッシュが含まれるように書き換えられることによってバージョンングされます。バージョンングされたURLのキャッシュ有効期限は1年間です。 Disabled (無効化): URLは変更されず、クライアントは再チェックされるまでCache on browserの全期間、この項目をキャッシングします。 詳細については、リソースの指定とURLのバージョンングを参照してください。
Mark resources as cacheable by intermediary proxies	Enabled (有効化): この場所にキャッシングされるリソースが、「Cache-Control」パブリックヘッダを使ってマークされ、中間プロキシサーバによってキャッシングできるようになります。 Disabled (無効化): キャッシング可能なリソースが、「Cache-Control」プライベートヘッダを使ってマークされ、中間プロキシサーバによってキャッシングできないようになります。
Optimize resources requested directly by their original URL and store them in the cache	Enabled (有効化): 画像やCSSスタイルシートなどのリソースが元のURLでリクエストされた場合に、このオプションを有効にするとリソースが最適化されキャッシングされてからクライアントに戻されます。これは、例えばページのロード後にサーバから画像その他のリソースを取得するJavaScriptがサイトに含まれている場合などに便利です。 Disabled (無効化): 直接リクエストされたリソースは最適化されずに戻され、Traffic Managerによってキャッシングされません。
Cache on browser	リソースがクライアントのブラウザにキャッシングされ、期限切れになるまでの期間(秒単位)です。これは、Expiresヘッダを設定するスライド式期限として使用されます。つまり、クライアントはファイルを受け取った時点から指定された秒数の間、ファイルをキャッシングします。 この設定は、最適化され、元のURLによってリクエストされたリソースにのみ適用されます。 結合リソースセット、またはURLの自動バージョンングを使用するリソースは例外です。この場合、リソース(またはリソースセット)のURLはコンテンツのハッシュから生成されるため、Expiresヘッダを変更すると、結果としてリソースURLが変更されます。したがって、リソースは変更後のURLの強度に合わせて再リクエストが必要になります。

設定	説明
Background recheck	サーバキャッシュにあるリソースが再チェックの対象としてマークされるまでの経過時間(秒単位)です。このようにマークされたリソースがリクエストされると、キャッシュにある既存のバージョンがただちに戻されますが、変更があった場合にはキャッシングされていたバージョンがバックグラウンドタスクによって更新されます。

Hostname Rewriteタブ

設定	説明
URL hostname rewriting	一致するURLを全て書き換えることによって、現在のドメインから、指定したホストまたはURLルートにリソースを移動します。単一のホスト名 (cdn.example.comなど)、またはオプションのスキームとパスも含めた完全なルートURL (「https://cdn.example.com/images」など)として指定できます。

最適化手法について

このセクションでは、Web Content Acceleratorで使われている各種の最適化の原理について説明します。最初に、Webページの速度の原則の概念と、Web Content Acceleratorがその原則を適用してWebコンテンツを最適化する方法を取り上げます。次に、Webアプリケーションに最適化を適用するために、Web Content Acceleratorで使用されるいくつかの手法について説明します。

Webページの速度の原則

原則は4つのカテゴリに分類されます。クライアントブラウザのレスポンスが向上するように、それぞれの原則をWebアプリケーションに適用できます。

- Webページによってロードされるオブジェクトの数を減らす
- サーバとの間で送受信されるあらゆるもののサイズを小さくする
- リピートビューの速度を上げるために可能な限りキャッシングする
- サーバがリクエストに応答する所要時間を減らす

それぞれの原則を、以下のセクションで詳しく説明します。

Webページによってロードされるオブジェクトの数を減らす

Webページ上の要素ごとに、ブラウザはそのリソースが保存されているホストに対してHTTPリクエストを送信します。各サーバに対して確立できる同時接続の数はブラウザによって制限されるため、Webページの要素が少なくなるほどWebページのロードは高速になります。Webページのロードを最も高速化できるのは、この最適化手順です。

Web Content Acceleratorは、適切な場合にファイルを結合して自動的にオブジェクトの数を減らし、リソースタイプ別に「結合リソースセット」を作成します。最も一般的なタイプの具体的な情報を以下に示します。

画像のインライン化と結合

画像は、どのWebページでも重要な要素です。画像はCSSバックグラウンド画像とHTML画像に分けることができ、それぞれでアプローチが異なります。

- CSSバックグラウンド画像

CSSバックグラウンド画像は、カスタムスタイルシートでバックグラウンドプロパティから参照されるか、Webページ内のスタイルプロパティから直接参照されます。このようなリソースに対するリクエストの数を減らすために、Web Content Acceleratorはそれらをbase64データに変換し、スタイルシートファイルまたはHTMLに埋め込みます。

実際のアクセラレーション動作は、ブラウザによって異なります。

- Internet Explorer (バージョン6および7)の場合、base64データは別のファイルに保存され、スタイルシートのバックグラウンドURLがMHTML表記を使ってこのファイルを参照します。
- それ以外のブラウザの場合、base64データは「data:uri」表記を使って直接CSSに埋め込まれます。
- HTML画像

HTML画像は、ページのタグから参照される画像です。HTML画像を高速化するために、Web Content Acceleratorは特定フォーマットの全ての画像を含む結合CSSスプライトを作成します。

Web Content Acceleratorは次に、このスプライトをバックグラウンド画像として参照するようにCSSスタイル情報をタグに追加し、透過GIFを参照するsource属性を設定します。

JavaScriptの最小化

HTMLの内部から実行されるJavaScriptファイルで余分なコンテンツの量を減らすと、ページのロード時間を全体的に短縮する上で効果があります。冗長な空白やコメントは、ソフトウェア開発の観点では有益ですが、ページの動作に対しては機能的に何の意味も持たないため、ファイルサイズを小さくするために削除されます。

スタイルシートの結合

スタイルシートのリクエスト数を減らすことは、ブラウザがWebページのレンダリングを開始する時間を短縮する上で重要なステップです。スタイルシートを結合すると、多くの場合、安全かつ効果的にページのロード時間を短縮することができます。

第三者ドメインのコンテンツの高速化

デフォルトでは、ページの表示元である同じWebサブドメインから提供される要素に対しても、Webページのアクセラレーションが適用されます。例として、「www.example.com」というWebサイトでタグの画像URLを考えてみます。

URL	結果
/images/sample1.jpg	Web Content Acceleratorで最適化
www.example.com/images/sample1.jpg	Web Content Acceleratorで最適化
images.example.com/sample1.jpg	最適化なし
images.otherexample.com/sample1.jpg	最適化なし

この設定を変更して他のWebサブドメイン、さらに第三者ドメインまでを含めると、Webページの表示に必要なパートナーのコンテンツを効果的に高速化することができます。

Web Content Acceleratorは以下の目的でリソース参照を書き換えることができます。

- オブジェクトをCSSスプライトセットに結合する
- ドメインホストから供給される元のURLに、バージョニングを適用する([リソースの指定とURLのバージョニング](#)を参照)

リソースの同時ダウンロード数を増やす手法を「ドメインシャーディング」と言い、この方法ではリソースの一部が別々のWebドメインまたはサブドメインからダウンロードされます。ドメインシャーディングではWebページ上で使用されるWebドメイン名ごとに、個別に余計なDNS解決の時間とTCPコネクション確立の時間がかかるため、このアプローチの方がドメインシャーディングよりも多くの点で優れています。また、コンテンツはWebサーバから提供されるため、一部のコンテンツが利用できないという可能性も大幅に少なくなります。

サーバとの間で送受信されるあらゆるもののサイズを小さくする

デフォルトでは、Web Content Acceleratorはサーバから送信されるコンテンツのサイズを自動的に小さくしようとします。

- テキストベースのオブジェクトは、gzip圧縮アルゴリズムを使用して圧縮されます。
- JavaScriptとスタイルシートのファイルは、冗長な空白文字とコメントを削除して圧縮されます。
- 冗長な画像情報(メタデータなど)が削除され、JPEG画像は元の画質の85% (デフォルト)で再サンプリングされます。

リピートビューの速度を上げるために可能な限りキャッシングする

2度目以降にWebページにアクセスしたとき高速で再利用できるように、ブラウザはリソースをローカルでキャッシングします。

Web Content Acceleratorは全てのリソースにキャッシュヘッダを自動的に適用し、それらのリソースの管理に必要な情報をブラウザに提供します。デフォルトでは、Web Content Acceleratorがリソースに適用するキャッシング期間はそれが参照される方法に応じて異なります。以下の表のとおりです。

条件	キャッシュ期間
オブジェクトがリソースセットに結合されている	1年間
オブジェクトURLがWeb Content Acceleratorによってバージョニングされている	90日間
元のオブジェクトURLがリクエストされた	1時間

パブリックプロキシのキャッシング

オプションで、Web Content Acceleratorは中間キャッシングを行うプロキシサーバを利用できます。これは企業環境やインターネットサービスプロバイダ(ISP)では一般的です。特定の属性を使用して、Web Content Acceleratorは特定のリソースの全てまたは一部に「cache-control: public」ヘッダを追加します。

リソースをプロキシキャッシュ可能とマークすると、他のユーザが既にアクセスしたことがあるWebサイトのWebページにアクセスする場合には、最初のビューでもリピートビューでもロード時間が大幅に短縮されます。

サーバがリクエストに応答する所要時間を減らす

この原則が適用されるのは、サーバがクライアントのブラウザにHTMLレスポンスの送信を開始するまでに要する時間です。前述したように、これでWebページのロード時間は最大20%短縮されます。

ロード時間の短縮は、サーバ、アプリケーション設計、保守に直接関係します。考えられる解決法としては、スケールアップ(既存のサーバにプロセッサやメモリなどのシステムリソースを追加する方法)、スケールアウト(サーバノードを増やし、さらに多くのノード間に負荷を分散する方法)、コードのプロファイリングとチューニング、データベースのチューニング、シャーディング(データベースを複数のサーバに分散する方法)、Webサーバのチューニングなどがあります。

この原則はWebサーバとアプリケーションに適用されるため、本章ではこれ以上扱いません。

リソースの指定とURLのバージョニング

Web Content Acceleratorは、Webページを高速化するプロセスで作成される「結合リソースセット」または「URLが自動バージョニングされたリソース」にそれぞれ新しいIDを作成します。この方法によってTraffic Managerでは、Traffic Managerを何度もチェックして変更の有無を確認することをブラウザに求める代わりに、リソースの変更をブラウザに通知することができます。これを可能にしているのが、リソースのURLをそのコンテンツによって決める仕組みです。

例えば、あるリソースセットを以下のように指定します。

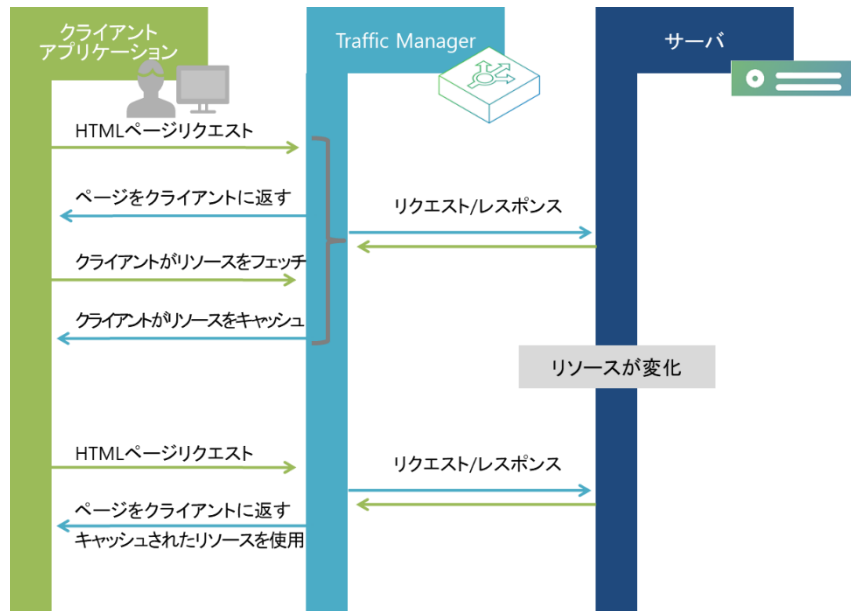
```
/optimized-gif-kjmMLxSEdXYqXt3coAasow4F=-ds3432d.gif
```

こうすると、自動バージョニングされたURLが以下ようになります。

```
/images/optimized-jpg-23dgdsds5563gfv=-P9vry3Uq22.jpg
```

これらのURLにおける数字と文字の文字列は、特定のリソースを構成する設定、コンテンツ、および下位リソースのユニークなハッシュ値を表しています。設定を変更するたびに、その設定が適用されるリソースは自動的に再計算されます。したがって、リソースのコンテンツが変更されれば、このユニークな値も変更されます。

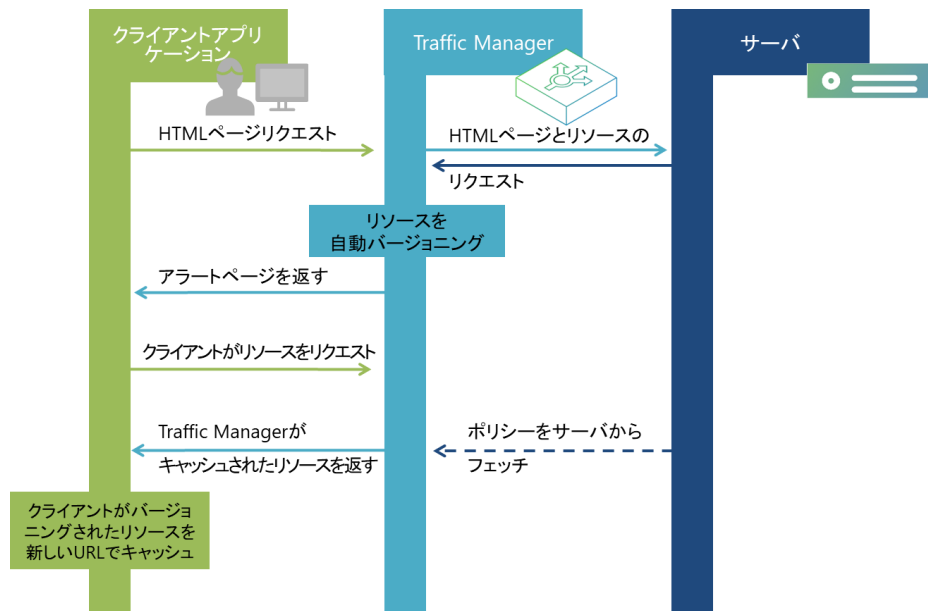
通常の場合、Traffic Managerは自動バージョンングを有効にせず単にページを表示するだけであるため、実行される典型的なデータフローは以下の図のようになります。



1. クライアントが、Traffic Managerで実行されているサービスを介してHTMLページをリクエストします。
2. クライアントが、通常のリソース(スタイルシート、画像など)へのリンクを含むHTMLページを受信します。
3. クライアントがそれらのリソースを取得します。
4. Traffic Managerがバックエンドサーバからそのリソースを供給します。Web Content Acceleratorの設定によっては最適化されています。
5. 使用可能なリソースをクライアントがローカルにキャッシングします。
6. 何らかの理由(バグ修正、レイアウト変更など)により、バックエンドサーバでリソースの一部が変更されます。
7. クライアントが、同じWebサイト上の別のHTMLページをリクエストします。
8. クライアントが、同じリソースへのリンクを含むHTMLページをTraffic Managerから受信します。
9. クライアントはサーバ上の変更を認識していないため、ローカルにキャッシングされた(期限が切れていない)バージョンのリソースを参照し、Webサービスからは再リクエストしません。

最後のステップでわかるように、クライアントは一定時間をおいてからキャッシュされたリソースを再リクエストするため、そのリソースは変更されている場合も変更されていない場合もありうるというのが、このシナリオで避けられない1つの欠陥です。そのためクライアントは、何らかのイベントが発生して再リクエストを要求するまで、古いバージョンのリソースを使い続けることになります。

自動バージョンングを有効にすると、クライアントが常にWebページに含まれる最新版のリソースを使っていることがTraffic Managerで保証されるようになります。これを示したのが以下の図です。

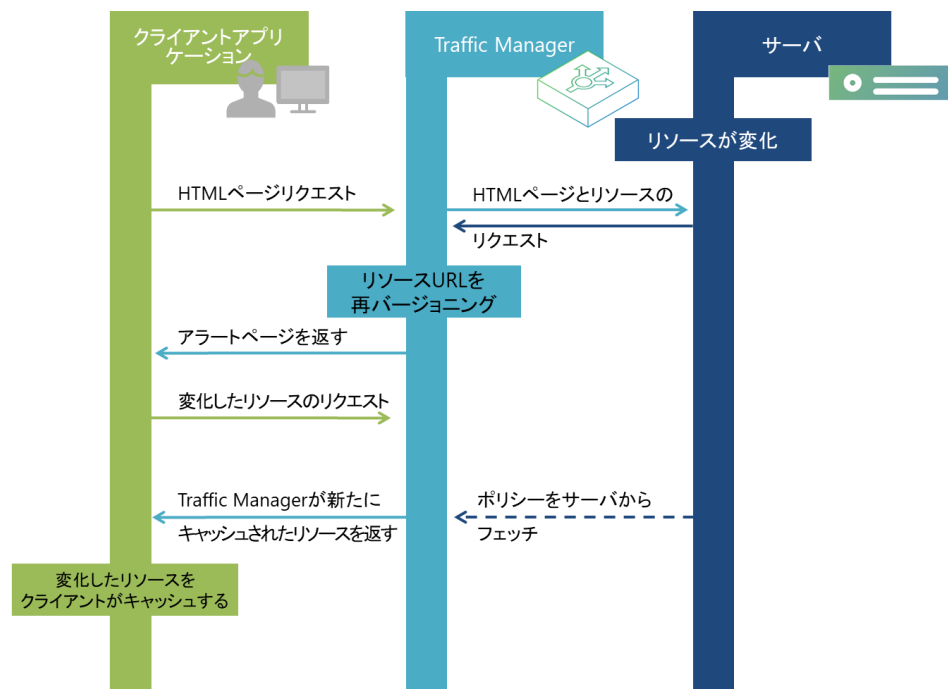


1. クライアントが、Traffic Managerを介して提供されるWebサービスからHTMLページをリクエストします。
2. HTMLページで1つ以上のリソースが参照されていることをTraffic Managerが認識します。
3. Traffic Managerがそれらのリソースを取得し(また、Web Content Acceleratorの設定によっては最適化し)、その内容確認に基づいてハッシュ値を作成します。
4. Traffic Managerが、計算済みのハッシュ値を含むリソースの新しいURLを作成し、HTMLページで参照箇所を全て置き換えます。
5. 変更されたページがクライアントに返されます。
6. クライアントが、新しいURLを介してページ内リソースをTraffic Managerからリクエストします。
7. Traffic Managerが、指定されたコンテンツをそのキャッシュから供給します(または、キャッシュになければバックエンドサーバから取得します)。
8. クライアントが、新しいURLを使用してリソースをローカルにキャッシングします。

これ以降のページリクエストに対しては、(バックエンドのリソースコンテンツに基づいて)各リソースの計算済みのハッシュ値が変更されていないことをテストした上で、Traffic Managerが応答します。

1. クライアントが、別のHTMLページをリクエストします。
2. Traffic Managerは、含まれるリソースがバックエンドサーバで変更されていないことを確認し、計算済みで前回と同じハッシュ値URLに書き込みます。
3. クライアントは、キャッシングされたコピーのリソースを使用します。

バックエンドサーバでリソースが変更された場合、これ以降のページリクエストは計算済みのハッシュ値が異なることによって影響されます。以下のようなシナリオになります。



1. リソースの変更により、コンテンツベースのURLが強制的に再計算されます。バックエンドサーバで1つ以上のリソースが変更されます。
2. クライアントが、別のHTMLページをリクエストします。
3. Traffic Managerは、目的のリソースがバックエンドサーバで変更されていることを確認し、新しいコンテンツに基づいて(以前のURLと異なる)新しいURLを作成します。
4. クライアントは新しいURLのキャッシュエントリを持たないため、Traffic Managerにそれをリクエストし、取得します。
5. クライアントはただちに、変更があったリソースの新しいバージョンを使い始めます。

コンテンツ分配ネットワークの使用

コンテンツ分配(または配信)ネットワーク(CDN)は、地理的に分散しているサーバ上のリソースの一部または全部をホストし、Webサーバとユーザの間に遅延を抑えます。CDNの詳細については、http://en.wikipedia.org/wiki/Content_distribution_networkを参照してください。

Web Content Acceleratorプロファイルの編集ページで**URL rewrite to CDN** (CDNにURL書き換え)設定を使用すると、Web Content AcceleratorでCDNを利用できます。

この設定でTraffic Managerは、ルールに一致するリクエストのHTMLを書き換えるよう指示します。これによって、リソースは独自のWebサーバではなくCDNで指定されたドメインからダウンロードされるようになります。

CDNホストにアクセスする際、Traffic Managerは受信したリクエストのスキーム(HTTP/HTTPS)を自動的に使用しますが、必要な場合には**URL rewrite to CDN** (CDNにURL書き換え)設定でスキームと仮想パスを任意に指定することもできます。このため、ターゲットのWebサイトがSSLコネクションを終端する別のデバイスを介してリバースプロキシされている場合には、全てのトラフィックをSSLに切り替えることが可能になります。

Web Content Acceleratorのトラブルシューティング



Web Content Accelerator Expressをご使用の場合、このセクションで説明されている一部のUIコントロールおよび設定オプションは該当しません。

予想外の動作のコントロール

Traffic Managerが、問題のあるコンテンツや標準外のユーザ定義されたコンテンツに直面すると、Web Content Acceleratorのプロセスが強制的に障害状態になることがあります。このようなことが起きると、Traffic Managerはエラーメッセージとともにイベントログに障害を記録してからプロセスを再開します。

状況によっては、エラーを起こしやすいコンテンツによって障害の発生とWeb Content Acceleratorのプロセスの再開が繰り返されることもあります。このような事態は、Traffic Managerの全体的なパフォーマンスの妨げになり、トラブルシューティングをいっそう複雑化します。これに対応するため、Traffic Managerには「ウォッチドッグ」機能が用意されています。この機能を使用すると、設定した制限に達するとWeb Content Acceleratorのプロセス再開が無効化され、頻度の高い障害による影響を抑えることができます。

ウォッチドッグを設定するには、**System** (システム) > **Global Settings** (グローバル設定) > **Web Content Accelerator**をクリックします。このセクションには、以下の設定があります。

設定	説明
optimizer!watchdog_limit	optimizer!watchdog_interval 設定で定義された期間内にWeb Content Acceleratorのプロセスが開始または再開される最大回数です。1台のTraffic Managerでこの回数だけプロセスが失敗すると、クラスタ内の全てのTraffic ManagerでWeb Content Acceleratorが無効化され、 System (システム) > Diagnose (診断)ページから手動でリスタートしなければならなくなります。 この制限を0に設定すると、Web Content Acceleratorは無効化されなくなります。
optimizer!watchdog_interval	ここで設定した期間(秒単位)を経過すると、前回の障害は、 optimizer!watchdog_limit で設定されている制限に加算されなくなります。

 ほとんどの場合、これらの設定を使用または変更する必要はありません。Web Content Acceleratorで頻繁に障害が発生する場合は、サポート窓口にお問い合わせください。

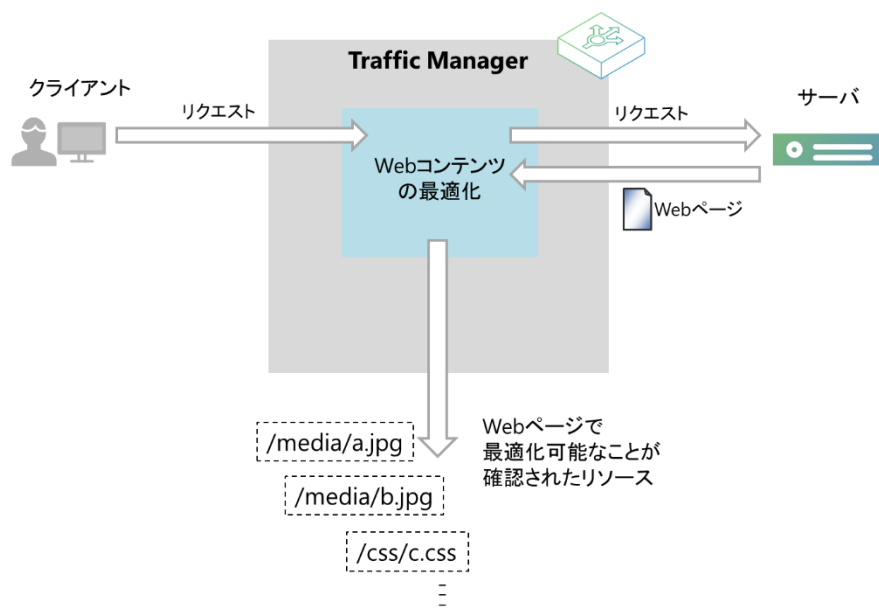
Web Content Acceleratorのその他のグローバル設定については、[その他の設定可能なグローバル設定](#)を参照してください。

その他のTraffic Manager機能との相互作用

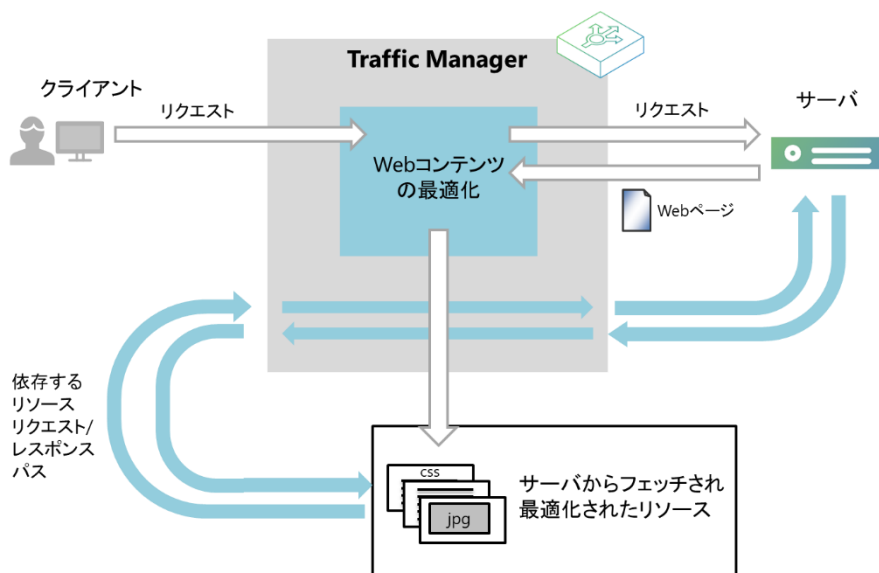
実際には、仮想サーバでWeb Content Acceleratorを有効にしても、他のTraffic Manager機能で相互運用性の問題が発生することはありません。しかし、内蔵されている一部の保護およびパフォーマンス機能の性質上、Web Content Acceleratorと他の機能との相互作用を理解しておくことは重要です。

依存リクエストの問題

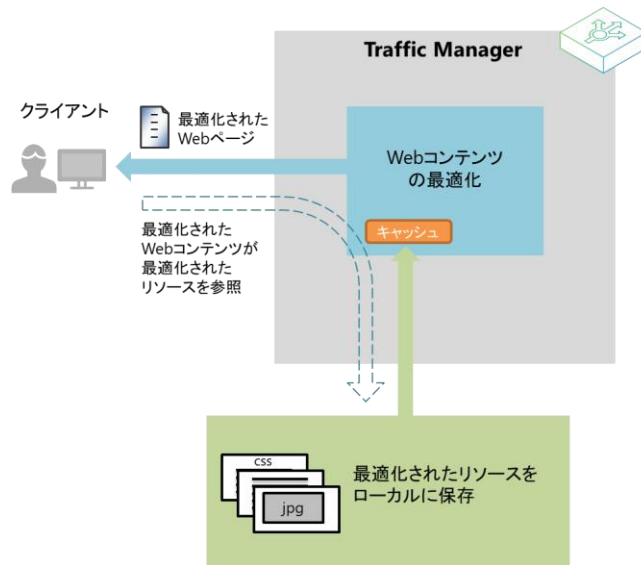
以下の図は、「依存リクエスト」と呼ばれるものによる影響を示しています。これは、Traffic Managerによって処理された1つのWebページリクエストに続いて、返されたページに含まれる依存リソースを要求するようなりクエストです。



最初にクライアントが、Traffic Managerによって処理されるサービスをWebページにリクエストします。Web Content AcceleratorがこのWebアプリケーションに対して有効で、最適化可能なリソースを識別しようとします。



次にTraffic Managerは、アプリケーションのページロード時間を短縮するためにストリッピングされ、または最適化された各種のリソースを探して取得できるように、追加の依存リクエストを多数実行する必要があります。



それらのリソースを取得すると、適切なリソースが最適化され、Web Content Acceleratorによってローカルでキャッシングされます。これ以降のページリクエストには、最適化されたコンテンツへの変更後のURLが含まれ、それがTraffic Managerから直接供給されます。

依存リクエストは、Traffic Managerを介して一般的なリクエストとして処理され、通常の処理が適用されます。ただし、これらのリクエストには、Web Content Acceleratorを介して直接行われたリクエスト/レスポンスと同じ識別ヘッダ、クッキー、その他の要素が含まれているとは限らないため、予期しない結果が生じる可能性があります。

以下のセクションでは、このような依存リクエストの取得に関する問題点を取り上げます。

サービスプロテクション

依存リクエストを取得すると、場合によってサービスプロテクションクラスが実行されることがあります(クラスターでTraffic Managerの1つからリクエストが大量に受信されるため)。これは、クラス設定のAccess Restrictions (アクセス制限)セクションで管理されている**Allowed IPs** (許可IP) (ホワイトリストとも言う)のリストに適切なTraffic ManagerのIPアドレスを追加することで防ぐことができます。サービスプロテクションの特徴と機能については、[サービスプロテクション](#)を参照してください。

TrafficScriptルール

Web Content Acceleratorが有効なとき、TrafficScriptルールの通常の処理順序は以下のようになります。

- リクエストの場合:

クライアントリクエストを受信 -> TSリクエストルール -> Web Content Accelerator -> バックエンドノードに送信

- レスポンスの場合:

バックエンドからレスポンスを受信 -> TSレスポンスルール -> Web Content Accelerator -> クライアントに送信



依存リクエストの場合、TrafficScriptルールは通常どおりに実行されます。

クライアントのIPアドレス、ブラウザヘッダ、クッキー、認証データなど、クライアントに固有の情報を利用するTrafficScriptルールは、依存リクエスト側にそのような情報がないため、予期しない動作を引き起こす場合があります。

http.stream.*TrafficScript関数の使用によっても、レスポンスの最適化が妨げられることに注意して下さい。

SSL復号化

SSL復号化を使用し、かつクライアント証明書を必要とする仮想サーバを最適化するためにTraffic Managerを使用することはできません。これは、Web Content Acceleratorが追加のリソースに対して取得を開始するとき、クライアント証明書が提供されないためです。

認証

Expressモードでは、認証を採用しているサービスを最適化するためにTraffic Managerを使用することはできません。Enterpriseモードでは、Traffic ManagerはNTLMを使用してサービスを最適化することができます。その他のクライアント認証方法はサポートしていません。

コンテンツキャッシング

Web Content Acceleratorを介して渡され、Web Content Acceleratorのスコープ内にあると見なされるWebコンテンツは、通常のTraffic Manager Webキャッシュで通常どおりにキャッシングされることはありません。代わりに、最適化されたWebコンテンツ/リソースのローカルコピーを保持するために設計されている専用のWeb Content Acceleratorキャッシュに保持され、そこから供給されます。この動作をコントロールするには、**Web Content Accelerator Profile** (Web Content Acceleratorプロファイル)編集ページで「Caching (キャッシング)」タブの設定を使用します。

ランタイムエラー

ランタイムエラーは通常、サーバに関連して発生します。ランタイムエラーが発生する原因は、サーバの誤設定、またはサードパーティとの干渉(ルータ、ファイアウォール、ロードバランサ)です。一般には、Webサイトに全くアクセスできなくなるか、Webサイトにアクセスできてもリソースが欠落した状態になります。

兆候: Web Content Acceleratorが有効な場合にのみエラーページが表示される。リソースが400または404 HTTPステータスで返される。アプリケーションプールが有効な場合でも、サーバからWebページを返すには非常に時間がかかる。

- **問題:** URLリライタまたは他のHttpModuleが、Web Content AcceleratorリソースセットのURLを処理しようとしています。
- **解決方法:** /optimized-gif-*.gifを含むURLを、他のHttpModuleによるURL書き換えまたは処理から除外します。

兆候: Webサービスが、ローカルでアクセスしたときには動作するが、外部クライアントからアクセスしたときには動作しない。

- **問題:** プロキシまたはアップストリームのデバイスが/optimized-gif-*.gif URLをブロックまたは変更しているため、無効になっています。
- **解決方法:** /optimized-*に対するリクエストを、変更しないままホストサーバに到達するように許可します。

兆候: HTMLにコメント「<!-- Optimized for speed – Pulse Secure Web Content Accelerator -->」が表示されない。

- **問題:** ブラウザで「HTMLソース表示」機能を使用すると、「<!-- Optimized for speed – Pulse Secure Web Content Accelerator -->」というコメントがなく、これはWebページが高速化されていないことを意味します。
- **解決方法:**
 - Web Content Acceleratorが特定の仮想サーバに対して有効であることを確認します。
 - URLがルールセットで除外されていないことを確認します。
 - Web Content AcceleratorがStealthモードで動作している場合は、URLリクエストで?aptimizer=onクエリ文字列を使用します。
 - URLが除外されておらず、Web Content Acceleratorが完全に有効な場合は、URLリクエストで?aptimizer=offが使用されていないことを確認します。

画像エラー

Webブラウザで、指定された位置に画像が表示されない、または表示される画像のサイズが正しくない場合は、画像エラーの可能性があります。Web Content Acceleratorはページにアクセスするたびに新しいリソースセットを作成するため、画像エラーは「目に見えない」場合もあります。

リソースセットの作成に関して問題を特定するには、ページの要素を調べます。1回作成されて何度もキャッシュから利用される一般的な要素には、特に注意してください。

兆候: ページにアクセスするたびに、新しいスプライトセットが作成されるように見える。

- **問題:** ページの画像のリストが毎回異なるため、ページアクセスのたびにWeb Content Acceleratorが新しいスプライトセットを作成しています。これが原因で、プロセッサが過剰に消費される、キャッシュメモリの使用率が高くなる、サーバでパフォーマンスが全体的に低下する、などの影響が現れます。
- **解決方法:** 動的な画像を結合しない形で含むURLルールを追加します(**Images**タブで**Combine HTML images into image sprites** (HTML画像を画像スプライトに結合)設定を無効にする)。

兆候: ページ上の画像でスペースが失われている、またはページ上で位置が揃っていない。

- **問題:** スペースの挿入は、画像を隣接するUI要素から分けるために使用されます。
- **解決方法:** 動的な画像を結合しない形で含めるURLルールを追加します(例えば、**Images**タブで**Combine HTML images into image sprites** (HTML画像を画像スプライトに結合)設定を無効にします)。

兆候: Web Content Acceleratorを有効にすると、いずれかの画像が大きく見える。

- **問題:** 画像を含むコンテナが、スタイルシートを適用して画像サイズを原寸から変更しています。
- **解決方法:**
 - 要素がwidth属性とheight属性を持つようにページテンプレートを変更します。これによって、Web Content Acceleratorは表示される画像の適切なサイズを判別できるようになり、ブラウザもページレイアウトを短時間で計算できるようになるため、レンダリングが高速になります。
 - 動的な画像を結合しない形で含めるURLルールを追加します(例えば、**Images**タブで**Combine HTML images into image sprites** (HTML画像を画像スプライトに結合)設定を無効にします)。

兆候: アクセラレーションは動作しているが、レイアウトに問題がある。

- **問題:** これは通例、スタイルシートまたはHTML画像に関係しています。
- **解決方法:** スタイルシートオブジェクトとHTML画像の数を減らします。この問題の原因になっている手法が何であっても、それを回避するにはアクセラレーションルールを使用する必要があります。

CSSエラー

Webブラウザで、指定された位置に画像が表示されない、または表示される画像のサイズが正しくない場合は、画像エラーの可能性があります。Web Content Acceleratorはページにアクセスするたびに新しいリソースセットを作成するため、画像エラーは「目に見えない」場合もあります。

リソースセットの作成に関して問題を特定するには、ページの要素を調べます。1回作成されて何度もキャッシュから利用される一般的な要素には、特に注意してください。

兆候: Internet Explorerで、スタイルシートに定義されているバックグラウンド画像が適用されない。

- **問題:** これは、Internet Explorerで設定されているセキュリティポリシーが原因です。あるドメインのHTMLページが別のドメイン上のリソースを参照しており、そのリソースタイプがrfc822である場合、ブラウザにはロードされますが、信頼できるかどうか不明なため処理はされません。ページと同じTLD上の外部リソースは、暗黙的に信頼されます。
- **解決方法:** メインWebサイトのサブドメインとなるCDNホスト名を変更し、そのリソースが暗黙的に信頼されるようにします。

JavaScriptエラー

Web Content Acceleratorを有効化する前に、最新のブラウザで検出されるようなJavaScriptエラーがないことを確認することが重要です。そうすれば、現在の問題が最適化の結果かどうかを判定できるからです。疑わしい場合には、Web Content Acceleratorを無効にした状態で再度ページをリクエストしてください。アクセラレータがバイパスされるので、最適化とは別にJavaScriptエラーが存在するのかどうかをチェックできます。

Internet Explorer 9では、開発者ツールとデバッグ情報を利用できます。FirefoxとChromeでは、Firebugアドオンをインストールして同様の情報を利用することができます。ほとんどのブラウザには、インストールされているツールやプラグインにかかわらず、JavaScriptエラーが発生したときにメッセージ/通知を識別する機能があります。このメッセージは、どのスクリプトが問題の原因になっているかを開示するときや、コードで問題のあるセクションの位置または行番号を特定するときによく役立ちます。

JavaScriptエラーのデバッグ

Web Content Acceleratorでは特定のURLにルールを適用できるため、Webサイトのデフォルト設定に手を加えることなく、個々の(またはグループの)リソースに対する特定の最適化をオフにすることができます。

Web Content Acceleratorを有効にした状態でJavaScriptエラーを解決するには、以下の1つ以上を実行します。

- 任意の照合ルールの**JavaScript**タブで、**JavaScript optimization** (JavaScript最適化)オプションを無効にします。
- どのスクリプトが機能していないかを特定し、それを最適化から完全に除外するURLルールを作成します。
- Web Content AcceleratorがJavaScriptエラーの原因になっているかどうかは、Webサイトで全てのJavaScript最適化を除外してみればすぐに判明します。「.js」を含むURLに対する最適化を全て除外する一時的なURLルールを作成してください。

一般的なJavaScriptエラー

兆候: スクリプトが実行され、予期しない結果が返される。

- **問題:** 外部スクリプト参照がページで削除されたことによるエラーです。
- **解決方法:** 外部スクリプトと他のスクリプトファイル、またはインラインスクリプトブロックとの間に、順序に依存するコードが存在する場合は、それがサイトの正常な機能を阻害している可能性があります。JavaScriptライブラリを作成する際のベストプラクティスは、関数のライブラリとして扱い、直接のコードは実行しないことです。こうすれば、スクリプトは可能な限り結合されます。設計に問題のあるスクリプトの回避策としては、適切なURLルールで完全にWeb Content Acceleratorから除外する方法があります。

兆候: スクリプトのロードと実行に失敗する。

- **問題:** スクリプトライブラリを動的にページにロードするJavaScriptフレームワークに問題があります。
- **解決方法:** これらのフレームワークは通常、使用するスクリプトに対する参照を検索し、一種の登録機能を使用してロードするスクリプトを挿入します。Web Content Acceleratorは結合されたスクリプトを代理URLで参照するため、このように動作するフレームワークが混乱する原因になる可能性があります。この場合の解決方法としては、これらのスクリプトを除外するか、ロード側のコードがWeb Content Acceleratorによる変更を理解できるように変更してください。

兆候: JavaScriptの非同期オプションを使用すると、JavaScriptの一部がWebページで実行されない。

- **問題:** 広告をブロックする一部のソフトウェアは、JavaScriptを非同期にロードして実行するための埋め込みWeb Content Acceleratorコードを誤って除外してしまうことがあります。
- **解決方法:** 必要なスクリプトを非同期コードから除外するURLルールを使用するか、JavaScriptを非同期モードでロードして実行するオプションを無効にします。

その他の設定可能なグローバル設定

System (システム) > **Global Settings** (グローバル設定) > **Web Content Accelerator**に設定可能な追加のグローバル設定があります。このセクションには、以下の設定があります。

設定	説明
optimizer!max_original_content_buffer_size	Web Content Acceleratorによる最適化が可能な、単一のバックエンドレスポンスの最大コンテンツサイズです。これよりも大きいレスポンスは、最適化されることなくクライアントに返されます。 バックエンドレスポンスが圧縮される場合、この設定はレスポンスボディの圧縮後のサイズに適用されます。
optimizer!max_dependent_fetch_size	Web Content Acceleratorに送られ、Web Content Acceleratorがサブプロセスで処理できる依存リソース(その時点で最適化が行われているコンテンツの一部が依存しているため、Web Content Acceleratorがバックエンドから要求したリソース)の最大サイズです。この値よりも大きなリソースは、Web Content Acceleratorによって常に見捨てられ、最適化されることなくクライアントに返されます。 制限を無効化するには、このフィールドを0に設定します。

通常は、これらの設定を変更する必要はありません。Traffic Managerは、通常、あらゆるシステムにおいて良好なパフォーマンスレベルを提供するデフォルト値を使用します。

イベント処理およびアラート

イベント処理機能を使用すると、管理者は、特定のイベントが発生した場合に、Traffic Managerにどのようなアクションを取らせるかを正確に設定することができます。

概要

「イベントハンドラ」は、特定タイプのイベントが発生したときにTraffic Managerが実行するアクションを指定します。イベントハンドラを設定するには、**System** (システム) > **Alerting** (アラート)をクリックします。

Alert Mappings

Event Type	Actions
All Events	➔ Log to event log Select action...
License Key Problem	➔ E-Mail Administrator Select action...

Select event type...

Manage Event Types

Manage Actions

Apply Changes

Update

Traffic Managerには、全てのイベントをグローバルイベントログに書き込むイベントハンドラが用意されています。この内蔵ハンドラとその「Log all Events (全てのイベントを記録する)」アクションは削除できませんが、必要に応じてアクションを追加することができます。

ライセンスキーに問題がある場合に電子メールを送信する、図に示すハンドラなど、イベントハンドラを追加することができます。イベントハンドラを追加するには、まずドロップダウンボックスからイベントタイプを選択し、次にそのタイプのイベントが発生した場合に実行するアクションを選択します。

イベントハンドラを作成するときに、「Bypass event log(イベントログのバイパス)」内部アクションをオプションで追加できます。結果として、そのイベントハンドラで処理される全てのイベントは、グローバルイベントログに記録されません。

「Connection Failures (接続失敗)」イベントタイプに一致するイベントはSyslogアクションとして扱われ、Traffic Managerイベントログに記録されません

Event Type	Actions
All Events	➔ Log to event log Select action... ▼
Connection Failures	✕ ➔ Bypass event log ✕ Syslog ✕ Select action... ▼
Select event type... ▼	
📄 Manage Event Types	📄 Manage Actions

イベントタイプ

Traffic Managerは、ノード障害、トラフィックIPアドレス転送、サービスレベルの不適合、ルールから報告されたカスタムイベントなど多様なイベントに対処することができます。便宜上、各イベントは、あらかじめ定義された多数のタイプに分類されています。また、必要に応じて新しいイベントタイプを作成することも可能です。

イベントタイプ	説明
All Events (全てのイベント)	Traffic Managerが検出可能な全てのイベントが含まれます。このタイプに一致する全てのイベントは、内蔵ハンドラによりグローバルイベントログに記録されます。
Critical Problem Occurred (重大な問題発生)	内部障害、ノード障害、IP転送(Traffic Managerの障害による)、その他のハードウェアの問題など重大な問題を示す全てのイベントが含まれます。
Critical Problem is Resolved (重大な問題が解決)	障害が発生したノードが再び動作開始したなど、「Critical Problem Occurred (重大な問題発生)」イベントタイプで報告された問題が解決されたことを示すイベントが含まれます。 問題発生時に電子メールメッセージを送信するようTraffic Managerを設定した場合、問題が解決して介入が不要になったときにも電子メールメッセージを送信するようTraffic Managerを設定することをお奨めします。
Default Events (デフォルトイベント)	Traffic Managerの旧バージョンとの互換性を確保するために含まれています。このイベントタイプには、Traffic Managerによる電子メールメッセージの送信やSNMPトラップの報告を伴う全てのイベントが含まれます。 Traffic Managerを旧バージョンからアップグレードすると、アップグレードプロセスにより、必要に応じて旧バージョンの動作を維持するためのイベントハンドラが追加されます。

イベントタイプ	説明
Infrastructure Problem (インフラストラクチャの問題)	これらのイベントは、サーバと通信できなくなるなど、Traffic Managerがインフラストラクチャの問題を検出した場合に報告されます。
Service Problem (サービスの問題)	これらのイベントは、サービスの正常な提供に影響を与える可能性のある問題が検出された場合に報告されます。

イベントタイプを表示するには、**Manage Event Types** (イベントタイプの管理)をクリックします。

新しいイベントタイプの作成

内蔵イベントタイプを変更することはできませんが、コピーを作成して編集したり、新しいイベントタイプを最初から作成したりできます。

例えば、Traffic Managerでいくつかのサービスを管理しており、特定のサービスで障害が発生した場合に一致するイベントタイプを作成するとします。これにより、そのイベントタイプを使用して、そのサービスの担当者に電子メールを送信するイベントハンドラを作成できるようになります。

これを行うには、「Service Failed (サービス失敗)」イベントタイプのコピーを作成し、特定の仮想サーバ、プール、およびノードからイベントが報告された場合にのみ一致するようこのコピーを編集します。以下の手順に従います。

1. **Manage Event Types** (イベントタイプの管理)をクリックし、次に「**Service Failed** (サービス失敗)」イベントタイプをクリックします。
2. このイベントタイプは変更できませんが、コピーを作成することはできます。「Make a Copy of this Built-in Event Type(この内蔵イベントタイプのコピーを作成)」セクションを使用して、「Custom Service Failed (カスタムサービス失敗)」という新しいイベントタイプを作成します。
3. Traffic Managerは、新しいイベントタイプの**Edit**(編集)ページを表示します。イベントはツリー構造に整理されます。ツリーのノードを展開すると、Traffic Managerが検出可能な全てのイベントが表示されます。イベントタイプごとに一致するイベントが表示されるため、イベントを簡単に確認できます。
4. 新しいイベントタイプを編集します。特定のサービスに関連するオブジェクトからイベントが報告された場合のみイベントタイプが一致するよう、イベントを報告するヘルスマニタ、ノード、プール、ルール、仮想サーバ、およびその他のオブジェクトのリストを変更できます。
5. 変更を適用するには、**Update** (更新)をクリックします。新しい「Custom Service Failed (カスタムサービス失敗)」イベントタイプに適したイベントハンドラを作成します。

アクション

イベントハンドラが実行されると、Traffic Managerは、そのハンドラに設定されているアクションを呼び出します。



Traffic Managerには、E-Mail (Eメール)、SNMP Trap (SNMPトラップ)、Syslogの3つのアクションが内蔵されています。Traffic Managerを新たにインストールした時点では、E-Mail (Eメール)およびSNMP Trap (SNMPトラップ)アクションは完全には設定されていません(最初に編集して必要な設定を指定する必要があります)。ただし、Traffic Managerを旧バージョンからアップグレードした場合は、旧バージョンでの設定が引き継がれます。

以下のタイプの追加アクションを作成できます。

タイプ	説明
E-Mail (Eメール)	このアクションは、ハンドラの契機となったイベントのレポートを、アクションに設定されている電子メールアドレスに送信します。 それぞれ異なる電子メールアドレスを設定した複数のE-Mail (Eメール)アクションを作成し、イベントタイプごとに異なる受信者に電子メールを送信するようTraffic Managerを設定することができます。
Log to external file (外部ファイルへのロギング)	イベントは常にグローバルイベントログに記録されますが、このアクションでは、さらに指定したログファイルにイベントを記録することができます。 使用フォーマットは、グローバルイベントログのフォーマットと同じです。このアクションは、特定イベントの専用ログを作成する場合に便利です。 Traffic Managerアプライアンス/クラウドのユーザ: この機能を仮想インスタンスまたはクラウドインスタンスで使用する場合は、ファイルをZEUSHOME内に保存してください。ユーザがZEUSHOME外に作成した全てのファイルおよびディレクトリは、アップグレードを適用すると失われる可能性があります。
Log to syslog (syslogへのロギング)	このアクションは、指定したサーバのsyslogデーモンにメッセージを送信します。 このアクションを使用すると、syslogメッセージを受け入れ可能な外部のモニタリングおよび管理ツールに特定のイベントを送信できます。

タイプ	説明
Program (プログラム)	このアクションは、アクションの Edit (編集)ページの Upload and Manage Programs (プログラムのアップロードと管理)ボタンを使用してTraffic Managerシステムにアップロードされたスクリプトまたはその他のプログラムを実行します。プログラム例については、 プログラムやスクリプトの呼び出し を参照してください。
SNMP Notify/Trap (SNMP通知/トラップ)	このアクションは、指定されたホストの指定のSNMPレシーバアプリケーションにSNMP通知またはトラップを送信します。トラップはTraffic Manager MIBで定義されており、アクションのEdit (編集)ページからダウンロードできます。 Traffic Managerは、SNMP v1、v2c、およびv3をサポートしています。このアクションに対して指定できる設定は、Edit Action (アクションの編集)ページで指定したバージョンによって異なり、System (システム) > SNMPページで利用できる設定(SNMPコマンドレスポндаオプション)と全く同じです。詳細については、SNMPによる性能のモニタリングを参照してください。
SOAP Callback (SOAPコールバック)	このアクションは、指定サーバにSOAPメッセージを送信します。 SOAPアクションとそれを処理するサーバの例については、SOAPメッセージの送信を参照してください。

アクションを編集したり、新しいアクションを作成したりするには、**Alerting** (アラート)ページの**Manage Actions** (アクション管理)リンクをクリックします。

「Bypass event log (イベントログのバイパス)」内部アクションを適用して、イベントがグローバルイベントログに記録されないようにすることもできます。

アクションのテスト

アクションを作成または編集したら、「テスト」イベントを使用してそのアクションをすぐに呼び出すことができます。これにより、アクションの設定や動作が正しいかどうかをテストすることができます。

イベントハンドラの設定

イベントハンドラを設定する前に、そのイベントハンドラの契機となるイベントに一致するイベントタイプを作成し、電子メールの送信など適切なタスクを実行するアクションを作成する必要があります。

イベントハンドラにより、イベントタイプと、イベントタイプに一致するイベントが報告された場合に呼び出す1つ以上のアクションが紐付けられます。

イベントハンドラを設定するには、**System** (システム) > **Alerting** (アラート) ページに移動します。ドロップダウンリストからイベントタイプを選択して新しいイベントハンドラを作成し、呼び出すアクションをそのイベントハンドラのエントリにあるドロップダウンリストから選択します。

変更を完了したら、**Update** (更新) をクリックします。

重複イベント

1つのイベントが複数のイベントタイプに一致し、これによって複数のイベントハンドラが実行される場合があります。これらの各ハンドラに同じアクションが設定されている場合、そのアクションが複数回呼び出されることになります。

送信される小さい電子メールの量を減らすために、E-Mail (Eメール) アクションはすぐには実行されず、電子メールメッセージをまとめて30秒間隔で送信します。E-Mail (Eメール) アクションのこの動作は、**System** (システム) > **Global Settings** (グローバル設定) ページの「Other Settings (その他の設定)」セクションで設定できます。

重要性の低いイベントが短時間に連続して報告された場合、Traffic Managerはイベントの反復を抑制します。深刻なイベントがこのように抑制されることはありません。

カスタムアクション

プログラムやスクリプトの呼び出し

Program (プログラム) アクションは、Traffic Managerにアップロードしたプログラムまたはスクリプトを実行するために使用します。プログラムやスクリプトは、**Catalogs** (カタログ) > **Extra Files** (追加ファイル) > **Action Programs** (アクションプログラム) ページを使用してアップロードします。

このアクションが呼び出されると、Traffic Managerは、そのアクションの契機となったイベントを識別するコマンドライン引数を渡してプログラムをバックグラウンドで実行します。

Traffic Managerがプログラムまたはスクリプトに渡す引数は以下のとおりです。

- アクションに設定したコマンドライン引数(オプション)
- プログラムの実行元アクションを呼び出したイベントハンドラの名前が指定された「eventtype」という引数
- イベントの記述: フォーマットはイベントログに書き込まれるイベントのログ行と全く同じ

例えば、TrafficScriptルールによりTestEventというイベントが報告され、このイベントが、Program (プログラム)アクションを呼び出す「My Event Handler」というイベントハンドラに捕捉された場合、プログラムは以下のように呼び出されます。

```
programname "--eventtype=My Event Handler" \  
  "INFO TrafficScript/TestEvent CustomEvent rules/<rulename> \  
  vservers/<vservname> <description>"
```

このスクリプトには2つの引数が与えられていることに注意してください(分かりやすくするため改行を挿入)。

SOAPメッセージの送信

SOAPアクションは、アクションに設定されたプロキシにSOAPリクエストを送信します。SOAPメッセージは、以下のWSDL仕様に適合しています。

```
ZEUSHOME/zxtm/etc/wsd1/AlertCallback.wsd1
```

このファイルは、SOAPアクションを編集するときに**Actions** (アクション) > **Edit** (編集)ページからダウンロードできます。この仕様については、Traffic Manager Control APIのドキュメントに記載されています。

SOAPリクエストは、インターフェース<http://soap.zeus.com/zxtm/1.0/AlertCallback>を使用し、以下の引数を渡してeventOccurredメソッドを呼び出します。

- イベントを報告したTraffic Managerの名前(タイプxsd:string)。
- xsd:dateTimeフォーマットのイベント発生時間。¹
- イベントの重大度(列挙タイプAlertCallback.Severity)。
- イベントを識別するタグ(列挙タイプAlertCallback.Tag)。
- タグの配列。現在は空ですが、将来使用するために予約されています。
- オブジェクトの配列(タイプAlertCallback.ObjectArray)。これらのオブジェクト(タイプAlertCallback.Object)により、イベントの報告元のオブジェクト(ノード、ルールなど)が記述されます。

一部のイベントは複数のオブジェクトを参照します。例えば、TrafficScriptルールから報告されるイベントには、それぞれイベント名、ルール、およびルールを実行した仮想サーバを識別する3つのオブジェクトが含まれます。

¹ <http://www.w3.org/TR/xmlschema-2/#dateTime>

- 人間が読み取ることのできるイベントの記述(タイプxsd:string)。これは、多くのイベントにあらかじめ定義されていますが、イベントがTrafficScript関数のevent.emit()によって報告されるときは、TrafficScriptが提供するメッセージが記述に含まれます。
- アクションによって提供される追加データ(タイプxsd:string)
- アクションを呼び出したイベントハンドラの名前

以下のPerl CGIスクリプトは、AlertCallbackメッセージの単純なSOAPコンシューマを実装します。このレスポンドは、メッセージの詳細をローカルファイルに記録します。このCGIスクリプトを適切なWebサーバ(PerlおよびSOAP::Lite Perlモジュール搭載)にインストールし、プロキシ値をCGIスクリプトのURLに設定してSOAPアクションを設定することができます。

```
#!/usr/bin/perl

package AlertCallback;

sub _addSerializerForEnum($;$)
{
    my ( $xsdtype, $ns_prefix ) = @_;
    $ns_prefix = "zeusns" unless $ns_prefix;

    $func = "SOAP::Deserializer::as_${xsdtype}";
    *$func = sub { return $_[1] };
}

BEGIN {
    _addSerializerForEnum( "AlertCallback.Tag" );
    _addSerializerForEnum( "AlertCallback.ObjectType" );
    _addSerializerForEnum( "AlertCallback.Severity" );
}

sub eventOccurred($$$$$$$)
{
    my $self = shift;
    my( $stm, $time, $severity, $tag, $tags, $objects, $desc, $additional, $handler ) =
    @_;

    open LOG, ">>events.log" or die( "Couldn't open log events.log: $!");

    print LOG "$stm - $time - $severity: $tag [";
```

```
print LOG join ', ', map { "$_->{type}:$_->{name}" } @$objects;
print LOG "]" $desc ($additional); Event handler: $handler\n";

close LOG;
}

package main;

use SOAP::Transport::HTTP;

SOAP::Transport::HTTP::CGI
-> dispatch_with( { "http://soap.zeus.com/zxtm/1.0/AlertCallback/" =>
'AlertCallback' } )
-> handle;
```

アクションが複数のイベントによって呼び出されるときは、SOAPメッセージ内のタグ値に基づいて正確なイベントを特定することが望ましい場合があります。これを行うには、イベントツリー(カスタムイベントタイプを設定したときのツリー)内のイベントの記述とWSDLファイル内の記述を照合して、AlertCallback.Tagの値を特定します。

TrafficScriptまたはJava Extensionsからのイベント報告

イベントは、TrafficScriptルールまたはJava Extensionから報告することができます。この場合、適切なイベントハンドラは非同期にバックグラウンドで実行されるため、TrafficScriptルールやJava Extensionがブロックされることはありません。

イベントを発生させるには、event.emit(event_name, event_message)関数(TrafficScript)またはemitEvent(event_name, event_message)関数(Java)を使用して報告します。

event.emit()またはemitEvent()によって報告されたイベントに基づいてアクションを呼び出すには、Custom Events(カスタムイベント)を含むイベントタイプを作成します。全てのカスタムイベントに一致させることも、特定のイベントにのみ一致させることもできます。

▼ Basic Settings

These settings allow you to specify which events make up this event type.

Name:

Events: **Events** Unfold All / Fold All

- ☐ Cloud Credentials
- ☐ Configuration Files
- ☐ Fault Tolerance
- ☐ General
- ☐ GLB Services
- ☐ Java
- ☐ License Keys
- ☐ Locations
- ☐ Monitors
- ☐ Pools
- ☐ Rules
- ☐ Service Protection Classes
- ☐ SLM Classes
- ☐ SSL Hardware
- ☐ Traffic Managers
- ☐ Virtual Servers
- ☒ Custom Events

Custom events can be triggered using the TrafficScript rule 'event.emit'

☐ No custom events
☐ All custom events
☒ Some custom events...

Add custom event

イベントタイプの特定イベントの名前は、そのイベントがTrafficScriptルールで報告されるときに使用される名前と一致している必要があります。

```
# raise an event named "TestEvent1", to be caught by an Event
# Type containing a custom event called "TestEvent1".

event.emit( "TestEvent1", "The request was ".$url );
```

例

単純な侵入検知システムが、特権URLへのアクセスを試行する全てのリモートクライアントを禁止するために使用されます。

```
# A TrafficScript rule that detects unauthorized
# accesses and raises an event.

$path = http.getPath();

if( string.StartsWith( $path, "/secure/admin" ) ) {
    $ip = request.getRemoteIP();
```

```
if( !string.ipMaskMatch( $ip, "10.0.0.0/8" ) ) {  
    event.emit( "UnauthorizedIP", "IP: ".$ip );  
}  
}
```

この例を使用するには、「UnauthorizedIP」というカスタムイベントを含むイベントタイプと、そのイベントタイプに基づくイベントハンドラを作成します。

このイベントハンドラから、侵入検知ルールを更新してそのIPからの接続を禁止するカスタムプログラムアクションを呼び出すことができます。

```
#!/usr/bin/perl -w  
  
# An Action script to process event messages raised by  
# the above TrafficScript rule.  
  
my $logLine = $ARGV[1] or exit;  
  
$logLine =~ /IP:([\d]+\.[\d]+\.[\d]+\.[\d]+)/;  
my $ip = $1;  
  
open OUT, ">>/tmp/events.log" or die $!;  
print OUT "The bad IP was $ip\n";  
close OUT;  
  
# reconfigure upstream firewall, or the  
# Traffic Manager's banned IP list to ban access  
# from $ip
```

システムレベルの設定

Traffic Managerのハードウェアアプライアンス、仮想アプライアンス、クラウドインスタンス、および一部のソフトウェア製品には、ネットワーク設定や時間および日付などの設定を管理するための追加の設定ページが多数あります。それらの製品では、通常はOSレベルで処理されるような、OSの様々なコンポーネントおよび機能の完全性チェックなど、数多くの追加的なシステムチェックおよびテストをTraffic Managerソフトウェアから実行できます。



本章の内容は、一部のTraffic Manager製品のみに該当します。詳細については、ご使用の製品仕様を参照してください。

Traffic Managerを工場出荷時の初期設定にリセットする詳細な方法など、初期のセットアップおよび設定方法については、ご使用の製品の『Installation and Getting Started Guide』をご覧ください。

ネットワーク設定



このセクションで説明する機能は、ご使用の製品によって異なる場合があります。詳細については、ご使用の製品仕様を参照してください。

Traffic Managerのホスト名、ネットワークインターフェース、DNS、ルーティング、IP転送、およびNATを設定するには、**System** (システム) > **Networking** (ネットワーキング) ページを使用します。



このページで行った設定は、管理UIにアクセスしているローカルのTraffic Managerのインスタンスにのみ適用されます。ただし、「Return Path Routing」(リターンパスのルーティング)は例外で、ご使用のTraffic Managerクラスタ全体に適用されます。クラスタに含まれる他のTraffic Managerのネットワーク設定をチェックまたは設定するには、そのインスタンスの管理UIを使用する必要があります。

ホスト名とIPアドレスの設定

ホスト名により、クラスタ内でTraffic Managerシステムがユニークに識別されます。また、IPアドレスにより、システム上のアクティブな各ネットワークインターフェースが識別されます。管理ポートを使用しない場合は、Traffic Managerシステムで指定したホスト名を固定IPアドレスに名前解決するために、ローカルDNSを設定します。

一般に、最初のネットワークインターフェース(eth0)は管理ポートとして指定されますが、指定するホスト名はそのインターフェースのIPアドレスに名前解決される必要はありません。Traffic Managerの管理IPアドレスでは管理情報のみの受信および送信が行われるため、対応するインターフェースは、内部または外部の公開トラフィックをルーティングするネットワークから切り離されたセキュアなネットワークに接続する必要があります。外部または内部トラフィックの管理には、その他のネットワークインターフェースを使用します。

管理ポートを使用しない場合、Traffic Managerは管理トラフィックを全てのインターフェースで受信するため、指定するホスト名は、Traffic ManagerシステムでアクティブないずれかのIPアドレスに名前解決されなければなりません。

ご使用のTraffic Manager製品のインターフェースとその番号については、『Pulse Secure Virtual Traffic Manager: Installation and Getting Started Guide』に記載されています。

DHCPの使用

 このセクションは、仮想製品およびハードウェアアプライアンス製品のみを対象としています。

Traffic Managerでは、Traffic Managerシステムの基本ネットワーク設定を管理するために、DHCP(Dynamic Host Control Protocol)の使用がサポートされています。DHCPまたは手動で指定する静的設定のいずれかを使用するように、各ネットワークインターフェースを設定できます。

DHCPを使用するには、制御するネットワークインターフェースに対して「DHCP」オプションをチェックします。**Update** (更新)をクリックして変更内容を適用します。

DHCPの制御下のインターフェースは淡色表示され、編集できません。

トラフィックIPアドレスをホストするように設定されたインターフェースで、DHCPを有効にすることはできません。また、管理UIにアクセスするためのインターフェースで、DHCPステータスを変更することもできません。

▼ Hostname and IP addresses

Name (hostname.domainname) of the appliance.
 Hostname:

Interface	IP Address	Netmask	MTU	DHCP	Remove
Management Port (eth0)	10.62.136.160	18	1500	☒	
eth1	10.62.135.155	18	1500	☒	
eth2	10.62.165.99	18	1500	☐	☐

Add IP Address:
 Interface:

IP Address:
 Netmask:
 DHCP: ☐

Traffic ManagerのデプロイでDHCPの使用を予定している場合は、システムの各インターフェースに対して、有効期間の長いIPを予約してネットワークインフラストラクチャを設定することをお奨めします。リースの期限切れ後のIPアドレスの更新によって、Traffic Managerクラスタにサービスの中断と通信の問題が生じることがあります。

DHCPを使用するために1つ以上のインターフェースを指定した場合、Traffic Managerは、ゲートウェイ、ネームサーバ、および検索ドメインに対して自動的な設定の取得も試みます。手動で入力した全ての設定は、利用可能な場合、DHCPで生成される値によってオーバーライドされます。

注記

RHIおよびRPRなどのルーティング機能は、DHCP対応インターフェースの使用によって影響を受けることがあります。詳細については、ネットワーク管理者またはサービスプロバイダに問い合わせることを強くお奨めします。

インターフェースの設定



このセクションは、特定の製品のみを対象としています。詳細については、ご使用の製品仕様を参照してください。

デフォルトでは、Traffic Managerは、各インターフェースの接続先スイッチとの間でネットワーク設定について自動的にネゴシエーションを行います。

場合によっては、インターフェース設定を手動で指定する必要があります。例えば、一部のスイッチが、サポートしている最高転送速度のネゴシエーションに失敗することがあります。

自動的なネゴシエーションを無効にするには、**System** (システム) > **Networking** (ネットワーキング) > **Ethernet Speeds** (イーサネット速度)をクリックします。該当する各インターフェースに対して、**Auto-negotiate** (自動的なネゴシエーション)を「No (いいえ)」に設定します。次に、**Duplex** (二重)および **Speed** (スピード)ドロップダウンリストを使用して、必要なインターフェース設定を手動で選択します。**Update** (更新)をクリックして変更を保存します。

トランキング/ボンディング

同じIPアドレスを複数の物理ネットワークインターフェースに設定すると、Traffic Managerは、これらのインターフェースをトランクのメンバにします(仮想アプライアンスのみ)。トランキングを使用するには、スイッチがIEEE 802.3adをサポートしていることを確認します。この機能は、スイッチのドキュメントによってはリンク集合、ボンディング、またはトランキングと呼ばれます。

VLANの設定



Microsoft Hyper-Vハイパーバイザで実行されている仮想アプライアンスに対して、Traffic Managerの管理UIからVLANを設定することはできません。そのような環境でVLANを有効化する必要がある場合は、ご使用のWindows ServerワークステーションのHyper-Vマネージャアプリケーションから直接VLANを設定してください。詳細については、Hyper-Vのドキュメントを参照するか、サポート窓口にお問い合わせください。

Traffic Manager管理UIを使用して、Traffic Managerの任意のネットワークインターフェースまたはボンディング/トランキングされたインターフェースにVLANを設定することができます。

VLANの設定

1. **System (システム) > Networking (ネットワーキング) > Virtual LANs (仮想LAN)**にアクセスします。
2. 仮想LANインターフェースを適切な物理インターフェースに追加します。

▼ **Virtual LANs**

Add Virtual LAN interfaces on top of your existing physical interfaces.

VLAN ID	Physical Interface	Remove
7	eth1	☐

Add New VLAN Interface:

Interface:

VLAN ID:

3. 新しいVLANにネットワーキング(IPアドレスとサブネット)を設定します。

Interface	IP Address	Netmask	MTU	DHCP	Remove
Management Port (eth0)	10.62.136.160	18	1500	☒	
eth1	10.62.135.155	18	1500	☒	
eth2	10.62.165.99	18	1500	☐	☐

Add IP Address:

Interface: eth0

IP Address: eth0 Netmask: DHCP: ☐

eth1

eth2

VLAN 7 (eth1)

► Virtual LANS

Add Virtual LAN interfaces on top of your existing physical interfaces.

VLANインターフェースを通してルーティングされるトラフィックには、VLAN IDのタグが付けられます。

DNSの設定

Traffic Managerは、ノード名をIPアドレスに名前解決するときや外部サービスと通信するとき、また受信したネットワークコネクションのIPアドレスを名前解決することがアクセス制限やその他の設定によって求められるときに、クラスター間通信用のローカルDNSサーバにクエリします。

Traffic Managerシステムに、1台以上のローカルDNSサーバのIPアドレスを設定する必要があります。通常、このIPアドレスは、Traffic Managerを最初にインストールするときに設定されますが、**System** (システム) > **Networking** (ネットワーキング) > **DNS**を使用して、この設定を編集したり、ホストとIPアドレス間の固定マッピングを追加したりできます。

ルーティングの設定

Traffic Managerは、ネットワークインターフェースに割り当てられたIPアドレスに従って、最適なネットワークインターフェースでトラフィックをルーティングします。Traffic Managerが外部ネットワークトラフィックを転送するためのデフォルトゲートウェイも設定することをお奨めします。

注記

デフォルトゲートウェイは、[フォールトトレランスの設定](#)で説明しているようにネットワークコネクションテストで使用し、外部(受信)トラフィックに使用されるネットワーク上に存在する必要があります。管理ネットワーク上に存在するゲートウェイを指定しないでください。または、より適切なIPアドレスを`flipper!frontend_check_addr`に設定してください。

デフォルトゲートウェイが適切でない場合は、必要に応じて、外部ネットワークへの追加ルートを手動で追加できます。この場合、外部ネットワーク(宛先およびネットマスク)、そのネットワークへのローカルゲートウェイ(オプション)、および外部ネットワークにアクセスするためのインターフェースを指定する必要があります。

これらの設定は、**System** (システム) > **Networking** (ネットワーキング) > **Routing** (ルーティング)で管理されます。

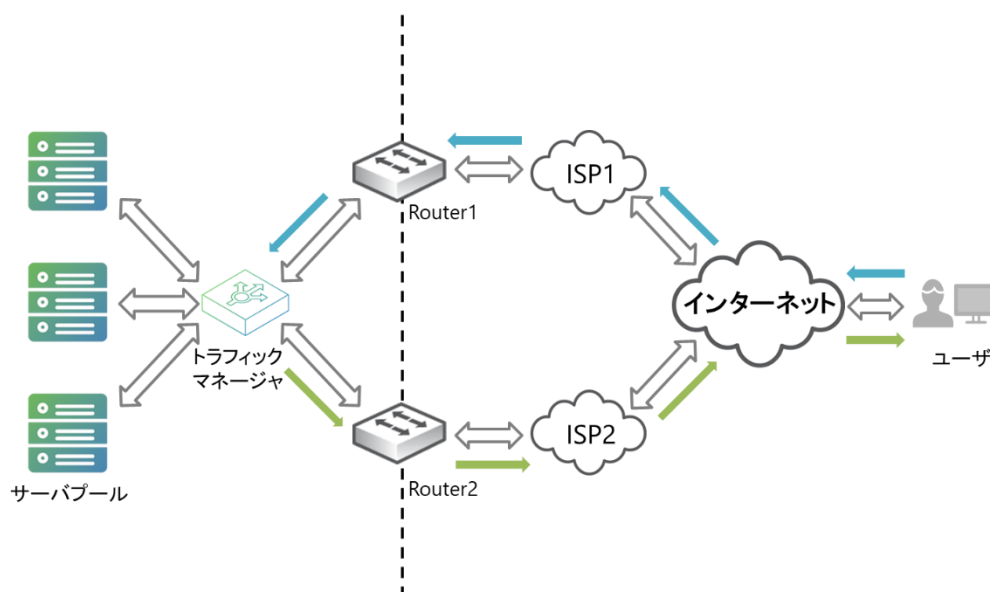
リターンパスルーティングの設定



このセクションの設定は、単体のTraffic Managerに特有ではなく、クラスタ内のその他全てのTraffic Managerのインスタンスに反映されます。

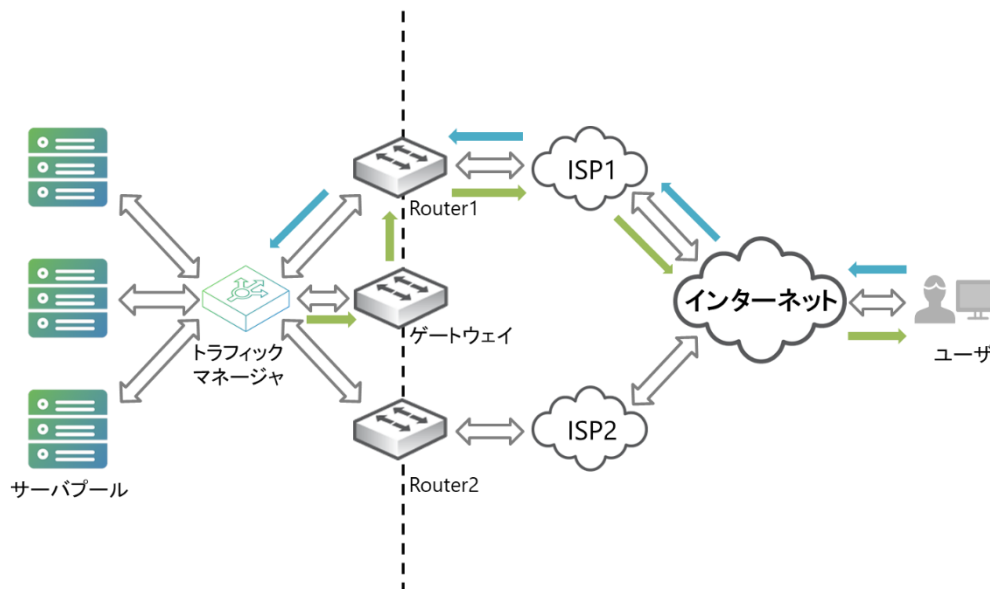
通常、どのルートデータパケットがネットワークを出てから宛先に到達するのかについて、それらが着信する限りは気にすることはありません。

従来のルーティングアルゴリズムは、データパケットの宛先アドレスを確認し、その宛先まで最もコストの低いパスであると考えられるルータにパケットを送信します。Traffic Managerとインターネットの間に複数のルートが存在する場合、クライアントからサーバへのルートがサーバからクライアントへのルートとは異なることがあります。通常、どちらのルートも期待される通りのパケットを受信している限り、この状態は問題になりません。

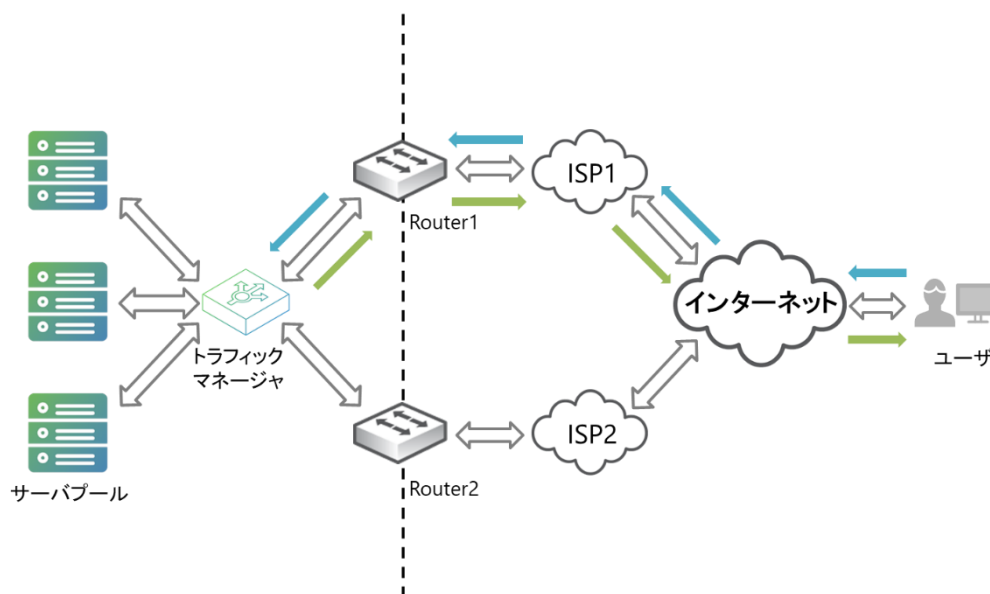


しかし、外部からの受信パスに対応するために、外部への送信パスが必要になることがあります。例えば、パス内にステートフルなファイアウォールまたはNATルータがある場合、そのデバイスは両方向でパケットを処理する必要があります。さもないと、デバイスが把握しているコネクションに属さないパケットは破棄されてしまいます。その代わりに、2つのISPを使用して負荷分散することができます(冗長性に加えて)。サーバからクライアントへの全てのトラフィックが同じISPを介して流れる場合、負荷の不均衡を引き起こす可能性があります。

この問題に対するソリューションの1つに、Traffic Manager用のデフォルトゲートウェイとしてゲートウェイルータを設置することが挙げられます。ゲートウェイルータは、それ自体のコネクション追跡を使用して、適切なISPにパケットを転送します。



ネットワークインフラストラクチャの都合でこのソリューションを利用できない場合は、Traffic Managerアプライアンスのリターンパスルーティング機能を使用すると、適切なリターンルートの提供がスムーズに行われます。



Traffic Managerは、最初のルータホップを介してのみリターンパスルーティングを戻すことができます。ネットワークが、必要なオンワードルーティングをクライアントに戻すことができるかどうかについては、お客様の責任においてご確認ください。

Traffic Managerは、所要のルータを介してリターンパケットをルーティングするために、「netfilter」のセッション追跡機能を使用します。Traffic Managerは新しいコネクションを受け取ると、イーサネットパケットヘッダのMACアドレスを使用して、どのルータを介してクライアントが接続されているかを特定します。Traffic Managerはこのアドレスを使用して、そのコネクションに属する全てのパケットにトラッキングマークでラベル付けします。

Traffic Managerがクライアントにレスポンスパケットを送信する際、トラッキングマークの存在によってIPフォーワーディングのルールが呼び出され、デフォルトゲートウェイや静的ルートではなく、適切なルータにパケットが送信されます。

Traffic Managerは、外部に送信される以下のデータパケットに対してリターンパスルーティングを適用します。

- クライアントによって開始されたTCPコネクションのパケット
- FTPなどの特定のプロトコルに対してクライアントが開始したTCPコネクションについては、Traffic Managerが開始した予備TCPコネクションのパケット
- クライアントからのUDPリクエストに関連するUDPレスポンス



全てのFTPコネクションのリターンパスルーティングを適切に確保するには、トラフィックを処理する仮想サーバを、デフォルトのFTPポート(21)を監視するように設定してください。

以下は適用対象外です。

- Traffic Managerが開始したその他のTCPコネクション
- その他のUDPパケット
- SIP (UDP)またはSIP (TCP)プロトコルで動作する仮想サーバが処理する全てのデータ

注記

コネクション追跡には、非常に大きな性能オーバーヘッドがかかります。そのため、リターンパスルーティングは、ネットワークインフラストラクチャにとって必要なときのみ有効化してください。

この機能を使用するには、リターンパスルーティング用に使用したいネットワークインフラストラクチャ内のルータを指定します。それぞれのルータに対しては、IPv4アドレス(および該当する場合はIPv6アドレス)と、そのIPが有効になるインターフェースのMACアドレスを特定します。ご使用のルータの設定インターフェースに、この情報が表示されることがあります。または、「ping」、「arp」、「tcpdump」などのネットワークツールを使用します。

Traffic Managerが複数のVLANを通じてルータに接続されている場合は、VLANに対応する特定のネットワークインターフェースを選択して、異なるVLANに対するリターンパスを区別することができます。ネットワークインターフェース「any」は、Traffic Manager上の全てのネットワークインターフェースを意味します。同じMACアドレスの複数のリターンパスがあり、その1つが「any」インターフェースであり、その他のリターンパスが特定のネットワークインターフェースである場合は、特定のネットワークインターフェースのルーティングの優先順が高くなります。

System (システム) > **Networking** (ネットワーキング) > **Return Path Routing** (リターンパスルーティング) の設定表にこの情報を追加します。このテーブルは、クラスタ内の全てのTraffic Managerインスタンスに反映されます。

リターンパスルーティングを有効化するには、**Yes** (はい)ラジオボタンをクリックしてから、**Update** (更新)をクリックします。

 バリデーションまたは通信の問題によるエラーや警告がイベントログに記録されることがあります。しかし、Traffic Managerは、データパケットが期待されるリターンルートを介して受信されたかどうかを初期のルータコネクションを超えてまで検出することはできません。要件に従ってパケットが受信されていることを確認するためには、補足的なネットワークテストを実施することをお奨めします。

IPフォワーディングおよびネットワークアドレス変換(NAT)の設定

ネットワークポロジによっては、Traffic Managerがネットワーク上のバックエンドノードまたはその他のデバイスのネットワークゲートウェイとして機能することがあります。例えば、IPトランスペアレンシを使用する場合、Traffic Managerを介して外部へと送信されるトラフィックをルーティングするようバックエンドノードを設定する必要があります([ルーティング設定](#)を参照)。

Traffic Manager(またはその他のネットワークデバイス)はゲートウェイとして機能する場合に、トラフィックを送信元ノードから宛先へと転送し、レスポンスを送信元ノードへと転送して返します。

Traffic Managerが転送する全てのトラフィックが、パブリックにルーティング可能なIPアドレスを持つソースから転送された場合、複雑な事態は発生しません。送信元から宛先までのリンクにある全てのホップが、レスポンスを送信元IPアドレスまでどのようにルーティングして返すかを認識しています。

ただし、バックエンドノードは、パブリックインターネットからルーティングできないプライベートネットワーク上に配置されるのが一般的です。このような場合、外部のインターネットクライアントはネットワークトラフィックをノードに直接送信することができず、パブリックIPアドレスを節約したり、セキュリティのレベルを高めたりすることができません。ゲートウェイデバイスはNATという手法を使用して、ルーティングできない送信元IPアドレスをパブリックにルーティング可能なゲートウェイのIPアドレスに透過的に置き換えます。

ネットワークポログラフィによっては、ダウンストリームのルータにNATを実行させることによって、全ての内部システムが、Traffic Managerを介した内部トラフィックがどのように適切にルーティングされているのかについて把握していることがあります。しかし、これに該当しない場合、転送パケットに対してNATを実行するようTraffic Managerを設定することにより、レスポンスがTraffic Managerゲートウェイを経由してルートバックされるようになります。

 Traffic ManagerのNAT機能は、IPv4でのみ利用できます。

NATを設定するには、管理UIの**System** (システム) > **Networking** (ネットワーキング) ページのNATセクションにある**Manage NAT Settings** (NAT設定の管理) リンクをクリックします。このページは、Traffic Manager 上でのNATの動作を管理するルールを定義するために使用します。

Traffic Managerを内部ネットワークトラフィック用の単純なIPv4またはIPv6ゲートウェイとして使用する場合など、NATを設定せずにIPフォワーディングを使用したいことがあります。そのような場合は、管理UIの**System** (システム) > **Networking** (ネットワーキング) ページのNATセクションにある**Enable IPv4 Forwarding** (IPv4 転送の有効化) または**Enable IPv6 Forwarding** (IPv6 転送の有効化) を選択します。



1つ以上のIPマッピングNATルールを有効化している場合や、Traffic Managerの任意のインターフェースを選択してNATを実行している場合、デフォルトでIPv4転送は有効化されます。したがって、**Enable IPv4 Forwarding** (IPv4 転送の有効化) コントロールは無効化されます。このような状況でも、IPv6転送は影響を受けません。

IPマッピングルール

Traffic Managerは、3つのタイプの外部送信IP NATルールをサポートしています。それぞれのルールは、そのルールが処理するトラフィックのソースIPアドレスを、指定したトラフィックIPグループのプライマリIPアドレスに置き換えます。

これらのルールは、以下の優先順位で適用されます。

ルールのタイプ	説明
Many to One: Port Locked	NATは、指定したプロトコルを使用するとともに、指定した宛先ポートをターゲットとすることにより、指定したプールのメンバからのトラフィックに適用されます。 このタイプのルールに有効なプロトコルは、icmp、sctp、tcp、udp、およびudpliteです。
One to One	NATは、単一の指定されたIPアドレスからのトラフィックに適用されます。 Add Inbound Rule? (外部からの受信ルールを追加) を選択すると、このルールが双方向に変わり、選択されたトラフィックIPグループのプライマリIPアドレスにアドレス指定されたトラフィックが、指定されたIPアドレスにも転送されます。この場合は、ソースIPアドレスは変更されません。

ルールのタイプ	説明
Many to One: All Ports	このルールは「Many to One: Port Locked」のルールと同様に動作します。しかし、ターゲットのポート番号にかかわらず、指定されたプールに対する全ての外部送信トラフィックを対象にする点で異なります。そのため、あるプールから開始され、「Many to One: Port Locked」のルールでは対象とされない任意のポート番号へのトラフィックに対するフォールバックルールとしてこれらのルールを使用することができます。

ポートマッピングルール

Traffic Managerは、以下の外部から受信するNATルールを仮想サーバでサポートしています。

ルールのタイプ	説明
Virtual Server Port Forwarding	指定されたポート範囲をターゲットとする受信トラフィックの宛先ポートが、選択された仮想サーバが監視するポートに置き換えられます。これにより、仮想サーバにより監視されるポートが1つではなく、一定範囲のポートが監視されるようになります。 ポート転送は、外部へと送信されるIP NATルールとは相互に影響し合いません。

インターフェースに対するデフォルトルール

このセクションは、内部ソースに代えて転送されるネットワークトラフィックに対してNATを実行する、Traffic Managerインターフェース全体を選択するために使用します。一般に、これらのインターフェースはパブリックにルーティング可能なIPアドレスを持ち、ネットワークの外側に位置します(外部に送信されるトラフィックの場合)。

 プライベートIPアドレスとパブリックIPアドレスが、NATが有効化された同じネットワークインターフェースに設定されている場合、NATは機能しません。有効なNAT設定にするためには、2つ以上のネットワークインターフェースを用意するか、このページでさらに具体的なルールを定義する必要があります。

このオプションを有効化すると、以下の効果があります。

- このインターフェースによって転送される全てのトラフィックに対してNATが有効化されます。Traffic Managerは、トラフィック内のソースIPアドレスをゲートウェイインターフェースのIPアドレスに置き換えます(Traffic Managerは、ソースポート番号を恣意的なポート番号に置き換えることもあります)。

- 宛先ノードからは、トラフィックがTraffic ManagerのゲートウェイIPアドレスから送信されているように見えます。宛先ノード(および全ての中間ホップ)がトラフィックをそのIPアドレスにルートバックする方法を認識します。
- 宛先ノードがレスポンスを返すと、Traffic Managerは、NATが有効化されたインターフェースでその応答を受信します。その後、元の送信元IPアドレスとポートがレスポンスに挿入され、接続元のバックエンドノードに転送されます。

時間と日付の設定



Xen仮想アプライアンスの場合、本セクションで説明する設定とは多少異なります。時間はホストハードウェアと同期されるため、この場合に変更できる設定はタイムゾーンのみです。

Traffic Managerとバックエンドノードのクロックを相互に同期しておく、各マシンのログファイルを比較すると共に、各システムで発生したイベントがどのように関係しているかを特定できるため便利です。

時間が相互に同期されていないと、Traffic Managerが記録したエラーとバックエンドサーバが記録したエラーが対応しているかどうかなど、イベントを相互に関連付けることが困難になります。

ローカルのTraffic Managerインスタンスの時間設定を管理するには、管理UIの**System** (システム) > **Time** (時間) ページを使用します。



管理UIのその他多くの設定とは異なり、このページで行う設定は、アクセスしている管理UIが属するローカルのTraffic Managerにのみ適用されます。クラスタに含まれる他のTraffic Managerマシンの時間設定をチェックまたは設定するには、そのマシンの管理UIを使用する必要があります。

時刻の手動設定

現在のシステム時間、日付、およびタイムゾーンを表示したり、必要に応じて設定を手動で行うには、「Basic Settings (基本設定)」セクションを使用します。

NTPサーバの使用

同じNTPサーバを使用するよう全てのTraffic Managerインスタンスおよびバックエンドサーバを設定すると、これらのインスタンスおよびサーバのクロックを正確に設定し、相互にほぼ同期させることができます。

パブリックNTPサーバを使用できますが、ネットワークの待ち時間や障害の可能性があることから、十分な信頼性が得られない場合があります。そのため、ローカルNTPサービスを実行し、全てのローカルシステムをこのサーバに同期することをお奨めします。ローカルNTPサーバは、ISPが管理するNTPサーバまたはパブリックNTPサーバと自動的に同期することができます。

何らかの理由により、Traffic Managerの仮想アプライアンスまたはクラウドインスタンスの時間が正しい時間から大幅に(30分など)ずれている場合、NTPはこのような大きなずれを補正しません。この場合に時間のずれを補正するには、**System** (システム) > **Time** (時間)をクリックし、**Sync Time Now** (時刻の同期化)をクリックします。

Traffic Managerからの時刻の同期

一般に、Traffic Managerインスタンスは複数のネットワークにまたがることから、ネットワーク接続された様々なデバイスの一貫したローカル時間管理ソースとして適しています。Traffic Managerは、全てのインターフェースでアクセス可能なローカルNTPサーバを実行するため、このサーバを基準に時間を同期するよう他のサーバを設定することができます。

Traffic Managerへのリモートログイン

Traffic Manager仮想アプライアンスまたはクラウドインスタンスのコマンドラインにアクセスするには、SSHを使用します。「管理」グループまたは「Appliance Console (アプライアンスコンソール)」パーミッションが有効化された他のグループのユーザのクレデンシャルを使用します。詳細については、[パーミッショングループ](#)を参照してください。

適切なクレデンシャルでログインしたユーザには、Traffic Managerインスタンスへのrootアクセスが与えられ、ホームディレクトリは/rootに設定されます。

パスワードを使用しない、SSH公開/秘密鍵による識別を使用したログインを設定するには、Traffic ManagerでSSH設定ディレクトリ/root/.sshを使用します。パスワードを使用しないログインへのSSHアクセスを制限するには、管理UIで**System** (システム) > **Security** (セキュリティ) > **SSH Server** (SSHサーバ)の設定を使用します。

カスタムカーネルパラメータの入力

sysctlは、システムの作動中にリアルタイムでカスタムアプライアンスカーネルパラメータを追加するために使用されるインターフェースです。これらのパラメータは、アプライアンスの管理UIの**System** (システム) > **Sysctl**ページから追加および変更することができます。

注記

このインターフェースは、高度な使用を行うための用意されています。これらの値を変更しようとすると、システムの性能が予期しない結果になったり、サポートされない結果になったりするおそれがあります。変更することによって生じる結果について十分に理解している場合にのみ、これらの設定を変更することをお奨めします。

一部のカーネルパラメータはTraffic Managerによって制御され、他の場所で適用した他の設定から生成されます。これらのパラメータのいずれかを変更しようとしても、Traffic Managerからエラーメッセージが返され、加えた変更は適用されません。

パラメータの追加または変更

カスタムカーネルパラメータを新たに追加するには、**Key** (キー)フィールドにパラメータ名を、**Value** (値)フィールドにそれに対応する値を入力します。複数の値を入力する場合は、各値の間にスペースを1つ入れて区切ります。後で参照するときや、問題が発生したときのために、オプションの**Description**フィールドにパラメータに関する説明を追加しておくことをお奨めします。**Submit Entry** (エントリを提出)をクリックして、アプライアンスにパラメータを追加します。

既存のパラメータの値を変更するには、新しい値をもう一度入力します。全ての変更は、**Existing Entries** (既存のエントリ)テーブルに反映されます。

注記

新しいパラメータおよび変更されたパラメータは、全てただちに有効になります。

既存のエントリ

既存のパラメータの説明を確認するには、マウスカーソルを**Info**タグの上に置いて説明を表示させます。説明が入力されていなかった場合、**Info**タグは存在しません。

カスタムパラメータを削除するには、削除したいパラメータの隣のボックスをチェックし、**Delete selected entries** (選択したエントリを削除)をクリックします。選択した全てのパラメータはただちに削除されますが、削除プロセスを完了するためには、サーバをリブートする必要があります。それまでにオーバーライドされたデフォルト値は、全て復元されます。

Traffic Managerは、入力されたパラメータの有効性をチェックします。エラーが見つかった場合は、このページ、イベントログ、および**Diagnose** (診断)ページに表示されます。

カーネルページテーブルの隔離

デフォルトでは、Traffic Managerアプライアンス製品は、カーネルページテーブル隔離(KPTI)機能により「メルトダウン」攻撃(CVE-2017-5754)から保護されています。KPTIは、権限のないソフトウェアがカーネルから任意のメモリを読み取ることができないように保護しますが、この保護によりシステムの全般的なパフォーマンスが低下します。

Traffic Manager インフラストラクチャにセキュリティリスクが発生しないよう、KPTI を有効にしたままにすることを強くお奨めします。ただし、信頼されるソフトウェアのみを実行していて、パフォーマンスと多層防御コストのバランスとして許容できる場合は、アプライアンスの KPTI を無効にしてください。

KPTI を無効化するには、**System** (システム) > **Traffic Managers** をクリックして、「Advanced Settings」(詳細設定) を展開し、**appliance!disable_kpti** を「Yes(はい)」に設定します。**Update** (更新) をクリックして変更内容を適用します。

システムセキュリティ

本章では、Traffic Managerソフトウェアをセキュアな環境で運用するための重要事項について説明します。この章でOSの設定を考慮する必要がある項目は、Traffic Managerソフトウェア製品のみには適用され、アプライアンスまたはクラウド製品には適用されません。

ファイアウォールとOSの設定

Traffic Managerはネットワークレベルのファイアウォールではないため、ファイアウォールの背後に配置することを強くお勧めします。ご使用のOSおよびUNIX全般に潜在するセキュリティ上の問題を減らすために、追加措置を取ることができます。ここでは、これらの措置について簡単に説明します。

注記

以降に示す注意事項は、特にTraffic Manager製品に重点を置いて要点を述べたものです。ここに記載されている内容以外にも、インターネットサービスの提供に関連するあらゆるセキュリティ上の考慮事項には十分留意し、必要に応じて専門的な助言を求めることをお勧めします。

ファイアウォールの設定手法

Traffic Managerは、リクエストに対してレスポンスを返すことができません。また、一部のプロトコル(FTPなど)では、クライアントとの間に追加のコネクションを確立することが必要な場合があります。ステートフルファイアウォールを設定すると、Traffic Managerから任意のポートで外部へのコネクションを確立し、接続先のクライアントから送信される以降のレスポンスをサーバまで通過させることができます。

Traffic Managerには、ステートフルファイアウォールを設定することをお勧めします。ステートフルファイアウォールは、デフォルトでは外部から受信する全てのトラフィックを拒否しますが、Traffic Managerの指定ポートとの通信は許可され、Traffic Managerへの全てのレスポンスはファイアウォールを通過することができます。

Traffic Managerが使用するデフォルトポート

Traffic Managerは、設定されている仮想サーバが使用するポートやプロトコルに加えて、以下のようなポートやプロトコルも必要とします。

- **管理サーバポート(HTTPS):** デフォルトでは、ポート9090(TCPおよびUDP)です。このポートをホスト名やIPアドレスと組み合わせて使用し、管理UIにアクセスします。このポートは、管理UIにアクセスするときに使用する、クラスタ内の他全てのTraffic Managerを含む、全てのデバイスに対して開かれている必要があります。このポートはまた、Traffic ManagerのSOAPベースの管理APIおよびコマンドラインインターフェース(CLI)でも使用されます。

Traffic Managerアプライアンス/仮想アプライアンス/クラウドインスタンスの各製品では、管理サーバポートの現在の値がコンソールに表示されます。ソフトウェア製品では、設定ファイルZEUSHOME/admin/websiteを読み取ります。このファイルにエントリがない場合、管理サーバポートはデフォルトの9090です。

- **コントロールポート:** デフォルトでは、TCPポート9080です。Traffic Managerはこのポートをコントロールプロトコルに使用して、複数のTraffic Manager間での設定データの変更を通信します。このポートは、クラスタ外部に公開されている必要はありません。

コントロールポートの現在の値を確認するには、設定ファイルZEUSHOME/zxtm/global.cfgを読み取ります。このファイルにエントリがない場合、コントロールポートはデフォルトの9080です。

- **REST APIポート:** デフォルトでは、TCPポート9070です。REST APIは、Traffic Managerを設定するための、プログラムによる代替方法です。詳細については、『*Pulse Secure Virtual Traffic Manager: REST API Guide*』を参照してください。このドキュメントは、IvantiのWebサイト(www.ivanti.com)から入手できます。

REST APIポートの現在の値を確認するには、管理UIで**System** (システム) > **Security** (セキュリティ) > **REST API** をクリックして、**rest!port**に設定された値を確認します。

初期設定時、デフォルトのポートのいずれかが使用できないとTraffic Managerが判断した場合、Traffic Managerはポート番号を1つずつ増やしながら空いているポートを検出します。そのため、この場合は管理サーバポートの設定が9091、9092、またはそれ以上に設定されます。このことは、コントロールポート(8081以上)またはREST APIポート(9071以上)も同じです。

注記

このガイドおよびその他全てのTraffic Managerのユーザドキュメントで参照しているポート番号は、「ここで示したデフォルト値、またはそれぞれの場所で設定/選択された値」を指すものとして読む必要があります。

管理サーバとREST APIのポートは、Traffic Managerの管理UIの**System** (システム) > **Security** (セキュリティ) ページから、必要に応じてユーザが定義し直すことができます。それにより、ファイアウォールルールの更新が必要になります。

Traffic Managerは、他のTraffic Managerサーバの状態を確認するためにブロードキャストも使用します。これらのブロードキャストは、マルチキャストアドレス239.100.1.1、ポート9090(または該当するポート)に送信されます。クラスタの全てのTraffic Managerがこのアドレスを監視します。ブロードキャストのIPアドレスおよびポートを変更するには、**System** (システム) > **Fault Tolerance** (フォールトトレランス)の設定を使用します。

[専用管理ネットワーク](#)で説明した、専用管理ネットワークを使用するようクラスタ内の各Traffic Managerを設定した場合、この管理トラフィックはそのネットワークにのみ制限され、Traffic Managerの管理ポートに他のネットワークから接続することはできません。ただし、設定エラーによってTraffic Managerの管理制限が緩和されたり、管理ネットワークが誤って公開された場合などに備えて、ファイアウォールによってこれらのポートを信頼されていない送信元から保護する必要があります。

Traffic Manager仮想アプライアンス/クラウドインスタンスは、全てのインターフェースでNTP (時間)リクエストを監視するローカルNTPサービスを実行します([Traffic Managerからの時刻の同期](#)を参照)。このNTPサービスはポート123で実行されます(UDPおよびTCP)。Traffic Managerをローカル時間管理ソースとして使用しない場合は、これらのポートをファイアウォールによって保護すると安全です。

注記

Traffic Managerがネットワークの可用性を検出するために使用するテストの1つにPingがあります。ファイアウォールやローカルTCP/IPの調整を無効化したり、デフォルトゲートウェイ、バックエンドノード、およびクラスタ内のその他のTraffic Manager宛てのICMPパケットにレート制限を適用したりすると、Traffic Managerが正常に機能しなくなる可能性があります。

ネットワーク設計

Traffic Managerは、高度なトラフィック管理機能を備えたプロキシです。すなわち、通常、Traffic Managerはパブリックネットワークとプライベートネットワークの間に配置され、これらのネットワークへとリクエストを渡します。各Traffic Managerは、トラフィックネットワークとバックエンドネットワークの両方のマシンと通信できなければなりません。

例えば、バックエンドプールがRFC1597プライベートネットワーク上にあり、そのIPアドレスが以下のいずれかのグループに該当するとします。

```
10.0.0.0/8
172.16.0.0/12
192.168.0.0/16
```

インターネット側からこれらのネットワークに直接ルーティングすることはできません。このような構成では、内部ネットワーク空間を都合よく割り当てることができる上、外部ルータが適切に設定されていればこれらのネットワーク宛てのパケットが受け入れられないため、セキュリティ上のメリットが得られます。

そのため、Traffic Managerでは、インターネットコネクション用にはトラフィックIPアドレス(トラフィックIPグループ)を使用し、全てのバックエンドプールとのコネクションには、別の物理ネットワーク(または正しく設定されたVLAN)上のRFC1597ネットワークアドレスを使用するのが最も安全です。これにより、インターネットからバックエンドサーバに直接アクセスできなくなり、外部からの全てのコネクションをTraffic Managerで安全かつ効率的に管理することができます。

さらに、Traffic Manager管理UIへのアクセスなどの内部管理トラフィック用に、別のRFC1597プライベートネットワークを使用することもできます。詳細については、[専用管理ネットワーク](#)を参照してください。

Unixユーザパーミッション

Traffic Managerソフトウェアは、rootとして起動する必要があります。これにより、「zeus.xtm」プロセスを特権ポート(1024番未満)にバインドすることができます。例えば、ポート80 (HTTP)は特権ポートです。Traffic Managerは、障害耐性を提供するためにスーパーユーザ権限も使用します。これらの権限は、イーサネットインターフェースを有効化/無効化するときに必要になります。

受信トラフィック(すなわち、信頼されていない入力)を管理するTraffic Managerプロセスは、プロセスの開始後にスーパーユーザ権限を放棄します。ソフトウェアのこの部分は、インストール中に指定したUNIXユーザおよびグループを使用して実行されます。Traffic Manager管理UIサービス(zeus.admin)は、rootユーザが所有する設定を管理するときにスーパーユーザ権限を使用します。

インストールプロセスの詳細については、ご使用の製品の『Pulse Secure Virtual Traffic Manager: **Installation and Getting Started Guide**』を参照してください。Traffic Managerプロセスのパーミッションが不明な場合は、「ps」コマンドを使用するか\$ZEUSHOME/ztm/global.cfgを調べることによって、zeus.xtmプロセスの所有ユーザを確認してください。通常、このユーザはnobody、グループはnobodyまたはnogroupです。

許可を変更する場合は、システムのスーパーユーザとしてTraffic Managerの「configure」スクリプト(ZEUSHOME/ztm/configure)を実行することをお奨めします。詳細については、ご使用の製品の『Pulse Secure Virtual Traffic Manager: **Installation and Getting Started Guide**』を参照してください。

ファイルシステムのセキュリティ

ZEUSHOME内の全てのファイルはインストール中に設定され、以降の変更はTraffic Manager管理UIによって行われます。すなわち、ZEUSHOME内の全てのファイルは、スーパーユーザが所有します。特定のファイルには、非特権許可で実行されるzeus.xtmプロセスがTraffic Managerの管理ファイルを読み取れるよう、全てのユーザに読み取り許可を与えるフラグが設定されています。

必要に応じて、Traffic Managerを「chroot」ジェイル内で実行することにより、ファイルシステムに対するZEUSHOME以外の全てのアクセスを効果的に阻止することができます。このとき、必要なシステムライブラリおよびその他のシステム固有の設定情報を用意し、/dev内にデバイスエントリを作成する必要があります。詳細については、ご使用のシステムのマニュアルを参照してください。



SyslogのロギングはUNIXソケットに基づくため、ファイルがZEUSHOME外に書き込まれる場合があります。

OSの設定

最高のセキュリティを確保するために、Traffic Manager以外のアプリケーション、特にネットワークサービスを提供するアプリケーションを実行するためにTraffic Managerを使用しないでください。新しいTraffic Managerの使用を開始すると、事前インストールチェックにより、使用されないUNIXサービスが削除または無効化されます。例えば、多くの場合、「lpd」(UNIXプリントサーバ)やRPC Portmapperなどのサービスはデフォルトで起動しますが、Traffic Managerの動作には必要ありません。これらのサービスを完全にアンインストールすることにより、Traffic Managerサーバへのアクセス手段となる公開インターフェースを大幅に減らすことができます。

サービスの中には必要なものもあります(Syslog、セキュアシェルなど)。これらのサービスが外部で必要ない場合は、可能であれば、これらのサービスをプライベートネットワークインターフェースまたはループバックインターフェースにのみバインドしてください。これにより、サーバに対する不要な関心を避けることができます。この対策は、ファイアウォールが配備されている場合も行ってください。

インストール前に、サーバのポート22を、外部からのsshコネクション用として管理ポート(設定されている場合)にのみ開いておくことをお勧めします。開いているポートは、無料で入手可能な「netstat」および「nmap」ツールで確認できます。

UNIXサーバの管理には、定期的なOSの保守作業が必要になります。また、セキュリティの確保は継続的なプロセスです。特に、ベンダーから提供されるパッチをトラッキングし、セキュリティ関連の問題に対する新しいパッチが提供されたら、インターネットに公開されているサービスをすぐにアップグレードするのが最善の方法です。これは、Traffic Managerが管理しているサービスだけでなく、管理用のSSHサーバなどTraffic Managerにインストールされているツールも対象になります。Linuxなど多くのOSでは、セキュリティ問題が特定されて修正されるたびに、カーネル自体のアップグレードが必要になる場合があります。

管理システムのセキュリティ

本章では、Traffic Managerシステムの管理コンポーネントへのアクセスをコントロールする方法について説明します。Webベースの管理UI、REST API、コントロールAPI (CLIも含む)、およびSSHによるリモートアクセス (仮想アプライアンス/クラウドインスタンスのみ)があります。



本章全体で、管理サーバ(またはAdmin Server)という用語は、全ての管理コンポーネントが制御またはルーティングされるサービスを表すために使用されます。

管理サーバの基本設定

基本セキュリティ設定を管理するには、**System** (システム) > **Security** (セキュリティ)をクリックします。これにより、「Administration Security (管理セキュリティ)」ページが表示されます。このページでは、SSL証明書、IPベースのアクセスコントロール、および管理サーバが使用するポートを設定することができます。

管理サーバのSSL証明書の変更

最初のオプションである**SSL Certificate** (SSL証明書)では、Traffic Manager管理サーバのWebベースのユーザインターフェース(管理UI)に使用するSSL証明書を変更できます。自己署名された新しい証明書を生成したり、新しい証明書と秘密鍵ファイルをアップロードしたりできます。**Update** (更新)をクリックして変更を保存します。



管理サーバはパブリックサービスではなく、Traffic Managerフレームワーク内の別の外部プロセスであるため、証明書をカタログから選択することはできません。

SSL証明書のSHA-1フィンガープリントはここに表示されます。これは、以下の理由で役立ちます。

- Webブラウザとの接続を初めて確立するときに、SSL証明書を検証する場合
- クラスタへの組み込み時にTraffic Managerの身元の真正性を検証する場合

新しいTraffic Managerを初めて設定するときに、フィンガープリントを書き留めておくことをお勧めします。ソフトウェア製品の場合、フィンガープリントは、インストールプロセスの完了後にコマンドラインに表示されます。アプライアンスとクラウドインスタンスの場合は、コンソールを参照してください。

フィンガープリントを再度確認する必要がある場合、コマンドラインで以下のコマンドを使用して表示することができます。

```
$ZEUSHOME/admin/bin/cert -f fingerprint -in $ZEUSHOME/admin/etc/admin.public
```

管理サーバに対するアクセスの制限

Traffic Managerは、オプションで管理ネットワークを使用できるだけでなく、以下の3つ方法で管理サーバに対するアクセスを制限することができます。

- クライアントIPアドレス(xx.xx.xx.xx)
- クライアントIPとネットマスク(xx.xx.xx.xx/nnまたはxx.xx.xx.xx/xx.xx.xx.xx)
- クライアントのDNS名またはDNSワイルドカード(admin.mysite.comまたは*.mysite.com)。例えば、

10.1.1.1, 10.0.0.0/24または*.mysite.comのように指定することができます。

DNSルールを使用する場合、Traffic ManagerはDNS検索を実行しなければなりません。これらの検索を行えるよう、必要なDNS設定が行われていることを確認してください。

Update (更新)をクリックすると、Traffic Managerにより、管理UIログインページへのアクセスを許可する前に管理サーバへの全てのコネクションがチェックされるようになります。

注記

これらの制限を機能させるためには、外部ファイアウォールおよびサービスが信頼されている必要があります。IPアドレスはスプーフィングされる可能性があります。また、DNSサービスが妨害またはハッキングされると、DNSテストを行う意味がなくなる可能性があります。

管理サーバポートの変更

各Traffic Managerがそれぞれの管理サーバをデフォルトのポート9090で実行します。初期設定時、デフォルトのポートを使用できないと判断した場合、Traffic Managerはポート番号を1つずつ増やしながら空いているポートを検出します。そのため、この場合は管理サーバポートの設定が9091、9092、またはそれ以上に設定されます。

このポートを手動で変更するには、「Management IP Address and Admin Server Port (管理IPアドレスと管理サーバポート)」で**adminport**設定を更新します。変更内容は、**Update** (更新)をクリックした直後に有効になります。このポートを変更すると、Traffic Managerによって新しいURLに自動的にリダイレクトされるようになります。

管理ポートの変更は、コントロールAPIを使用する全てのアプリケーションと、CLIによって確立されたコネクションに影響を与えます。

REST APIは、「REST API」セクションで設定可能な代替ポート(通常は9070)を使用します。

Traffic Managerのネットワークポートの要件の詳細については、[Traffic Managerが使用するデフォルトポート](#)を参照してください。

インストール時に専用の管理ポート(ネットワークインターフェース)を指定することができます。この値は、管理UIから変更することはできません。この値を変更したり、管理ポートを完全に無効化したりするには、インストール時に使用した初期設定手順を再実行します。詳細については、ご使用の製品の『Pulse Secure Virtual Traffic Manager: **Installation and Getting Started Guide**』を参照してください。

Traffic ManagerのSSHサーバのセキュリティ

i このセクションは、Traffic Managerソフトウェア製品には適用されません。

Traffic Manager仮想アプライアンスとクラウドインスタンスでは、標準のSSHサーバを通じてコマンドラインアクセスが提供されます。コマンドラインアクセスは、主にログの読み込みおよび特定のシステム保守操作に使用し、通常の管理機能には必要ありません。

Traffic ManagerのSSHサーバのセキュリティ設定を変更するには、「SSH Server (SSHサーバ)」セクションを使用します。

組織のセキュリティポリシーの一環として、SSHアクセスを制限することができます。この場合、**appliance!ssh!enabled**を「Yes (はい)」または「No (いいえ)」に設定して、全てのユーザに対してSSHアクセスを有効化または無効化します。アクセスを有効化した場合、**appliance!ssh!port**の優先ポート(通常は22)を設定します。**appliance!ssh!passwordallowed**を「No (いいえ)」に設定し、許可された鍵を/root/.sshにインストールすることによって、鍵ベースの認証を使用するようにSSHアクセスを制限することもできます。

Traffic Manager仮想アプライアンスとクラウドインスタンスには、ブルートフォースSSH攻撃の防止に役立つSSH Intrusion Preventionツールが含まれています。このツールによって、設定した時間内に、何度もコネクション試行に失敗したりリモートホストがブロックされます。SSH Intrusion Preventionツールの動作を設定するには、以下の設定を使用します。

設定	説明
Enabled (有効化):	SSH Intrusion Preventionが有効であるか無効であるか。
Whitelist hosts (ホワイトリストホスト)	SSH Intrusion Preventionツールによって禁止されないホストのスペース区切りリスト(そのホストがブラックリストにも含まれている場合を除きます)。IPアドレス、DNSホスト名、またはサブネットマスクによって指定されるIPアドレスの範囲を使用します。
Time (時間)	あるホストが禁止される時間(秒単位)。
Retry limit (再試行上限)	コネクション試行が禁止されるまでに、ホストがTraffic Managerに対して実行できるコネクション試行の失敗の回数。

設定	説明
	ホストが禁止されている間に実行されたコネクション試行は、次の禁止でカウントされません。ただし、ホストが禁止される時間(「Time (時間)」)が、モニタリング対象のタイムスパン(「Time span (タイムスパン)」)よりも短い場合を除きます。これに該当する場合は、ホストの禁止が解除されたときに、禁止の開始前に記録された失敗試行が、現在のモニタリング対象のタイムスパン内に残っている可能性があります。 禁止されている間のコネクション試行は、認証を受けずに、ただ破棄されます。
Time span (タイムスパン)	SSH Intrusion Preventionツールが失敗したコネクション試行をモニタリングする時間(秒単位)。
Blacklist hosts (ブラックリストホスト)	SSH Intrusion Preventionツールが、SSHを通じてTraffic Managerへのアクセスを決して許可してはならないIPアドレスまたはDNSホスト名のスペース区切りリスト。ホホワイトリストに登録したIPアドレスとホスト名を、ブラックリストに登録しないことをお奨めします(ただし、ブラックリストが優先されます)。 ホホワイトリストとは異なり、サブネットマスクによって指定されるIPアドレス範囲をブラックリストに登録することはできません。

Traffic Managerには、「Banned Hosts (禁止されたホスト)」の下に、ブラックリストに登録されたホストを含む現在禁止されているホストが表示されます。

禁止および禁止解除されたIPアドレス、SSH Intrusion Preventionツールが起動または停止された時間、および設定の変更の記録を含むログファイルをダウンロードするには、**Download Log** (ログのダウンロード)をクリックします。

クラスタ通信

クラスタメンバー間の通信は、中間者攻撃やその他のタイプの攻撃または介入の影響を受けないよう設計されたセキュアな通信によって処理されます。Traffic Managerは、SSLを使用したセキュアな公開/秘密鍵暗号化を提供することにより、場所に関係なくクラスタメンバーが安全かつ確実に通信できる環境を実現します。

「クラスタ通信」のセクションでは、クラスタ内通信に制限を適用して、ネットワーク環境に正確なセキュリティを実装する方法について説明します。

各Traffic Managerの内部管理ポートと通信できるホストIPアドレスは、**controlallow**設定で指定できます。IPアドレス、CIDR IPサブネット、およびlocalhostから成るリストを指定します。または、allに設定して全てのホストにコネクションを許可することもできます。

信頼された単一のネットワークの外にまで及ぶクラスタの場合、この信頼されたネットワーク外のクラスタメンバが、信頼性が損なわれた状態になっていないかどうかを常に判断できるわけではありません。他のTraffic Managerの信頼性が損なわれないようにするために、管理者が特定のクラスタメンバに対し、他のメンバより安全性が劣ることを示すマークを付けることができます。言い換えれば、信頼性を損なう高いリスクにさらされているクラスタメンバに対し、「信頼できない」とマークを付けることができます。信頼されないTraffic Managerは、引き続き設定の更新を受信でき、状態および統計データをブロードキャストしますが、実質的に設定の更新を他のクラスタメンバに複製できなくなります。

この機能を有効化するには、クラスタ内の各Traffic Managerに**control!canupdate**フラグを設定します。このフラグは、信頼されたTraffic Managerに対しては**有効化**し、信頼されていないTraffic Managerに対しては**無効化**することができます。この設定を使用するには、クラスタが複数のTraffic Managerで構成されている必要があります。



信頼されていないTraffic Managerの管理UIとコントロールAPIは、セキュリティ上の理由から無効化されます。マシン固有の設定(ネットワークング、時間/日付、SNMP設定など)を変更する必要がある場合、最初に**control!canupdate**を有効化する必要があります。ネットワークにおけるセキュリティの必要条件によっては、変更を行うために、Traffic Managerをクラスタから一時的に削除することが必要になる場合があります。

デフォルト設定**control!canupdate!default**は、新しいTraffic Managerがクラスタへの組み込み時に設定を継承するためのものです。信頼性の低い場所(クラウド環境など)にある新しいTraffic Managerを組み込む前に、この設定を「No (いいえ)」に変更することができます。

注記

信頼されていないTraffic Managerでは管理UIが無効化されるため、これらのクラスタメンバに対してログインしたり変更を行ったりすることはできません。そのため、何らかの理由で信頼されたメンバと通信できなくなった場合に、クラスタ全体を管理できなくなる可能性が生じます。そのクラスタのコントロールを取り戻すために、新しいTraffic Managerを組み込むこともできません。このリスクを回避するために、常にクラスタ内に信頼された冗長Traffic Managerを1台以上確保しておくことを強くお奨めします。詳細については、サポート窓口にお問い合わせください。

管理サーバおよび内部コネクションのSSL設定

これらの設定は、管理サーバとの間のコネクションおよびTraffic Manager内部のセキュアコネクションに関する高度なSSLオプションをコントロールします。デフォルト設定では、多くの環境に適した幅広いセキュリティレベルと互換性が提供されます。

特定のプロトコルまたはコネクション設定との互換性を確保する方法については、システム管理者またはサポート窓口にお問い合わせください。クラスタ内通信への潜在的な効果を期待してSSLまたはTLSバージョン(または暗号スイート)を切り換える場合は、十分に注意することをお奨めします。例えば、TLS 1.1からTLS 1.2に切り換えるには、**admin!support_tls1_1**を無効化する前に**admin!support_tls1_2**を有効化し、クラスタ全体にわたって変更した設定が再現されるようにします。この遷移の間、非互換性や通信不良を示すような永続的な警告またはエラーが発生していないか、Traffic Managerのイベントログを確認してください。

変更を有効にするには、管理サーバをリスタートする必要があります。管理サーバ自体から変更が行われた場合は自動的にリスタートしますが、RESTまたはSOAP APIなどの他のインターフェースによって変更が行われた場合は自動的にリスタートしません。その場合は、**System (システム) > Traffic Managers**ページで管理サーバを手動でリスタートしてください。

 Traffic ManagerのREST APIを使用しているときに、**admin!support_tls1_2**の設定を変更する場合、変更した設定でREST APIを使用し続ける前に、まずREST APIを無効化してから再度有効化する必要があります。REST APIのリスタートについては、[REST APIへのアクセス](#)を参照してください。

管理サーバに対するHTTP Strict Transport Security (HSTS)の設定

この設定は、管理サーバのHTTPページに追加されたHSTSヘッダをコントロールします。HSTSとは、プロトコルダウングレード攻撃やCookieハイジャックなどの中間者攻撃から管理サーバを保護するために役立つWebセキュリティポリシーメカニズム(RFC 6797で定義されています)です。これを有効にした場合、HSTSヘッダは、Webブラウザ(またはその他のユーザエージェント)がHTTPSコネクション、すなわち、SSLまたはTLSを使用して管理サーバと相互作用する必要があることを宣言します。

HSTSによる保護は、ユーザが最低1回サイトを訪れた後にのみ適用されます。ユーザがHTTPを指定しているサイトのURLを入力または選択した場合、HTTPリクエストは実行されず、HTTPSに自動的にアップグレードされます。これにより、HTTP中間者攻撃の発生が防止されます。

HSTSポリシーは、「Strict-Transport-Security」という名前のHTTPレスポンスヘッダフィールドを通じて管理サーバからユーザエージェントに伝えられます。また、管理サーバは、「max-age」キーを設定して、ユーザエージェントがHTTPSへの自動的なアップグレードを継続する期間を指定します。

Traffic Managerには、**System > Security (システムセキュリティ) > HTTP Strict Transport Security (HSTS) Settings for Admin Server(管理サーバに対するHTTP Strict Transport Security (HSTS)設定)**からHSTSポリシーの動作をコントロールするために使用する設定があります。

- **admin!hsts_enable:** 管理サーバに対してHSTSのサポートが有効であるかどうか (デフォルト: No (いいえ))。
- **admin!hsts_max_age:** HSTSヘッダが適用される秒単位の最大時間 (デフォルト: 「31536000」、すなわち365日)



これらの設定を有効にするには、管理サーバをリスタートする必要があります。管理サーバ自体から変更を加えた場合、リスタートは自動的にトリガされます。ただし、その他のインターフェース(RESTまたはSOAPなど)を通じて変更を加えた場合は、手動でリスタートを実行する必要があります。

HSTSを使用する場合は、次の点を考慮します。

1. HSTSは、管理サーバによる使用のために非自己署名SSL証明書が設定された場合にのみ有効にすることができます。
2. HSTS設定は、クラスタ間で複製されず、管理サーバの非自己署名SSL証明書をインストールした後に各クラスタメンバで有効にする必要があります。

管理認証局の管理

Traffic Managerは、設定された認証局によるユーザ認証の実行中に、接続先のSSL有効状態のLDAPサーバの認証を行います。そのLDAPサーバの証明書を発行した認証局(CA)からの公開証明書が、Traffic Managerに設定されている必要があります。

CA証明書を管理するには、**Catalogs (カタログ) > SSL > Administration Certificate Authorities and Certificate Revocation Lists Catalog (管理認証局および証明書失効リストカタログ)**をクリックします。

Traffic Managerに新しいCAをインポートするには、**Import certificate or CRL (証明書またはCRL のインポート)**をクリックし、以下のいずれかの方法を使用します。

- **Choose File (ファイルの選択)**をクリックして、ローカルワークステーションからTraffic Managerに証明書ファイルをアップロードします。
- Traffic Managerの**File URL (ファイルのURL)**ボックスにHTTP URLまたはHTTPS URLを入力して、直接ダウンロードします。
- **File Contents (ファイルコンテンツ)**ボックスにファイルのコンテンツを入力またはコピーします(PEM 符号化)。

Import File (ファイルのインポート)をクリックして、プロセスを完了します。Traffic ManagerがCAファイルをインポートし、新しい情報をクラスタ内の全てのTraffic Managerに伝搬します。

REST APIへのアクセス

「REST AP」のセクションには、Traffic ManagerのREST APIサービスで利用できる多数の設定があります。デフォルトでは、このサービスは有効化されています。

REST APIサービスを無効化するには、**rest!enabled**を「No (いいえ)」に設定します。REST APIサービスを有効化するには、**rest!enabled**を「Yes (はい)」に設定します。

サービスが監視するTCPポートを設定するには、**rest!port**を使用します。Traffic Managerシステムですでに実行されている他のサービスと競合しない限り、空いている任意のポートを使用することができますが、デフォルトのポートは9070です。

Update (更新)をクリックして変更を保存します。

このページの残りの設定については、『*Pulse Secure Virtual Traffic Manager: REST API Guide*』を参照してください。このドキュメントは、IvantiのWebサイト(www.ivanti.com)から入手できます。Traffic Managerのネットワークポートの要件の詳細については、[Traffic Managerが使用するデフォルトポート](#)を参照してください。

ユーザ管理

管理UI、コントロールAPI、REST API、およびCLIでは、各コネクションを認証するためにユーザ名とパスワードが必要になります。このログインによりユーザが認証され、そのパーミッショングループに基づいて、設定およびシステムに対してユーザに認可される読み取り、書き込み、または管理操作が定義されます。

ユーザとグループを設定するには、**System** (システム) > **Users** (ユーザ)をクリックします。

ユーザ認証

Traffic Managerは、ユーザのクレデンシャル(ユーザ名とパスワード)を以下の2つの認証ソースと照合して検証します。

- **ローカルユーザ:** クレデンシャルは、Traffic Managerのコンフィギュレーションにローカルに保存されます。

Traffic Managerには、完全な管理パーミッションを持つデフォルトのローカルユーザadmin (adminパーミッショングループのメンバ)が設定されています。基本的な管理環境では、このアカウントでログインして設定タスクを実行します。

ユーザアカウントの名前は変更できませんが、ユーザアカウントを削除することはできます。「管理」ユーザを削除する場合は、ログインしてクラスタを管理できるよう、削除する前に同等の管理パーミッションを持つ別のユーザアカウントが存在することを確認してください。

- **リモート認証されたユーザ:** クレデンシャルは、RADIUS、LDAP、またはTACACS+サービススペースの外部に存在するシステムに照らして認証されます。

認証コードでは、Traffic Managerがユーザ名とパスワードをデータベースに照合して検証する方法と、ユーザが属するパーミッショングループを特定する方法を定義します。認証を設定するには、**System** (システム) > **Users** (ユーザ)の「Authenticators (認証コード)」セクションを使用します。

ユーザがログインを試行すると、Traffic Managerは、まずクレデンシャルをローカルユーザのリストと照合し、ユーザが属するパーミッショングループを特定します。一致するエントリが見つからなかった場合、Traffic Managerは、ユーザが認証されてパーミッショングループが特定されるまで、各認証コードを順番に試します。

ローカルユーザ

新しいローカルユーザを追加するには、「Create New User (新規ユーザ作成)」ボックスにユーザ名とパスワードを指定し、ドロップダウンボックスからグループを選択します。これにより、このユーザが属するパーミッショングループが指定されます。

既存のユーザのパスワード、パーミッショングループ、またはUIプリファレンスを変更するには、「Local Users (ローカルユーザ)」ページで該当するユーザ名をクリックします。

ローカルユーザのUIプリファレンス

ローカルユーザは、管理UIの要素をコントロールする以下のプリファレンス設定にアクセスできます。

設定	説明
use_applet	全てのUIページに表示されるステータスアプレットの有効化/無効化を切り替えます。このアプレットを無効化すると、帯域を減らしたり、LiveHTTPHeadersやFireBugなどのツールを使用してTraffic Manager IPアドレスへのHTTPトラフィックをモニタリングするときのHTTPリクエストを減らすことができます。
appletwidth	アプレットの幅を変更します。これにより、仮想サーバごとのトラフィックをグラフ化したチャートに表示されるバーを増やす(または減らす)ことができます。
trafficscript_editor	高度なTrafficScriptエディタの有効化/無効化を切り替えます。無効化すると、単純なテキストボックスに置き換わります。

ユーザを削除するには、**Delete User** (ユーザ削除)をクリックします。

パスワードポリシー

Password Policy (パスワードポリシー) ページでは、ローカルユーザを作成したりローカルユーザがパスワードを変更するときに適用されるパスワードポリシーを設定することができます。ここでは、パスワードの長さ、含めなければならない文字タイプ、変更できる頻度、および更新しなければならない頻度の制限を設定できます。

以下のセキュリティ設定を指定できます。

- **No restrictions:** あらゆるパスワードを使用でき、制限は適用されません。
- **Default restrictions:** 新しいパスワードに標準のパスワードセキュリティが適用されます。具体的には以下のとおりです。
 - パスワードの長さは8文字以上でなければなりません。
 - パスワードに2文字以上のアルファベット文字を含める必要があります。
 - パスワードに1文字以上の大文字を含める必要があります。
 - パスワードに1文字以上の数字を含める必要があります。
 - パスワードにアルファベット以外の特殊文字を1文字以上含める必要があります。
 - パスワードで「aaaaa」のように同じ文字を繰り返すことはできません。
- **Custom restrictions:** パスワードの最小長さと、含めることのできる文字のタイプを手動で指定できます。

また、Traffic Managerでは、各ユーザが使用したパスワードの履歴が維持されます。**password_reuse_after** 設定を使用すると、ユーザがパスワードを何回変更した後に同じパスワードを再び使用できるかを指定できます。これは、ユーザがパスワードを変更するか変更を求められるたびに、単純に同じパスワードを再設定しないようにする上で役立ちます。値0は、ユーザが以前使用した任意のパスワードを再利用できることを示します。

password_changes_per_day 設定では、ユーザが24時間以内にパスワードを変更できる回数を指定します。0に設定すると、1日のパスワード変更回数は制限されません。

認証コード

ユーザがローカルユーザのリストで見つからない場合、管理サーバは、クレデンシャルを、完全に設定されている(また無効化されていない)全ての認証コードと照合します。認証コードにより外部データがクエリされ、有効なクレデンシャルを持つユーザのパーミッショングループの名前が返されます。一般に、パーミッショングループの名前はリモート認証データベースのフィールドに保存され、有効なパーミッショングループを持たないユーザには管理サーバへのアクセスが許可されません。

管理サーバは、LDAP、RADIUS、およびTACACS+認証データベースをサポートしています。

認証コードの作成

新しい認証コードを追加するには、**System** (システム) > **Users** (ユーザ) > **Authenticators** (認証コード)をクリックします。名前を指定し、適切な認証コードタイプを選択して、**Create Authenticator** (認証コード生成)をクリックします。これにより、未設定の認証コードが作成されます。選択したタイプに適した設定を行う必要があります。認証コードを設定してテストしたら、**auth!enabled**設定を「Yes (はい)」に設定して、その認証コードを有効化することができます。

Traffic Managerでは、この方法で認証コードが有効化されるまでその認証コードは使用されません。複数の認証コードを有効化した場合、Traffic Managerは、パーミッショングループの読み込みに成功するまで各認証コードを(アルファベット順に)試します。

LDAP認証コード

LDAP認証コードについて、以下の設定を行うことができます。

設定	説明
ldap!server	LDAPサーバのIPアドレスまたはホスト名です。
ldap!port	LDAPサーバに接続するために使用するポートです。
ldap!ssl	LDAPサーバへのコネクションをSSLで保護するかどうか、更にもし保護する場合はその保護にSTARTTLSまたはLDAPSメソッドを組み込むかどうかを指定します。

設定	説明
	コネクションに対してSSLを有効化する場合、 Administration Certificate Authorities and Certificate Revocation Lists Catalog (管理認証局および証明書失効リストカタログ)でLDAPサーバ証明書の認証局も設定する必要があります。詳細については、 管理認証局の管理 を参照してください。
ldap!timeout	LDAPサーバへのコネクションのタイムアウト時間(秒単位)です。
ldap!basedn	ディレクトリ検索用のベースDN (識別名)です。
ldap!filter	ベースDNにあるユーザをユニークに識別するフィルタです。文字列「%u」はユーザ名に置き換えられます。例: 「sAMAccountName=%u」 (Active Directory)または「uid=%u」 (Unix LDAP)
ldap!dnmethod	ユーザのバインドDNは、既知の文字列(「Construct」)から作成できます。または、ディレクトリ内を検索できます(「Search」 - ユーザが他のディレクトリパスにある場合に必要)。
ldap!binddn	ユーザ名からbinddnを作成するためのテンプレートです。文字列「%u」はユーザ名に置き換えられます。例: 「%u@mycompany.local」または「cn=%u, dn=mycompany, dn=local」。 この設定は、 ldap!dnmethod が「Construct」である場合にのみ適用されます。
ldap!searchdn / ldap!searchpass	ディレクトリでユーザのバインドDNを検索するときに使用するバインドDNとパスワードです。匿名バインドを使用したバインドDNの検索が可能な場合は、ブランクのままにします。 この設定は、 ldap!dnmethod が「Search」である場合にのみ適用されます。
ldap!groupfilter	上記の ldap!filter によって返されたユーザレコードに必要なグループ情報が含まれていない場合、この設定を使用して代替グループ検索フィルタを指定します。これは、一般にUNIX/POSIXタイプのユーザレコードがある場合に必要です。複数のレコードが返された場合、その全てからグループ名のリストが抽出されます。文字列%uはユーザ名に置き換えられます。 例: (&(memberUid=%u)(objectClass=posixGroup))

設定	説明
ldap!groupattr	ユーザのグループを指定するLDAP属性です。複数の値がLDAPサーバから返された場合、最初の有効な値が使用されます。
ldap!groupfield	ユーザのグループを指定するグループ属性(ldap!groupattr)のサブフィールドです。 例えば、 ldap!groupattr が「CN=mygroup, OU=groups, OU=users, DC=mycompany, DC=local」のような値を返す「memberOf」の場合、 ldap!groupfield を「CN」に設定します(一致する最初のフィールドが使用される)。
ldap!fallbackgroup	上記の ldap!groupattr が定義されていない、もしくはユーザに設定されていない場合、ここで指定するグループが使用されます。指定しない場合、 ldap!groupattr に一致する属性を持たないユーザはアクセスを拒否されます。

RADIUS認証コード

RADIUS認証コードについて、以下の設定を行うことができます。

設定	説明
radius!server	RADIUSサーバのIPアドレスまたはホスト名です。
radius!port	RADIUSサーバに接続するために使用するポートです。
radius!timeout	RADIUSサーバへのコネクションのタイムアウト時間(秒単位)です。
radius!secret	RADIUSサーバと共有する秘密鍵です。
radius!groupvendor	アカウントのグループを指定するRADIUS属性のベンダーを表すRADIUS識別子です。Filter-Idなどの標準属性*を使用する場合は、ブランクのままにします。
radius!groupattr	アカウントのグループを指定する属性を表すRADIUS識別子です。 radius!fallbackgroup を指定した場合は、ブランクのままにすることができます。
radius!fallbackgroup	ベンダーおよびグループ識別子によってグループが見つからないか、見つかったグループが有効でない場合、ここで指定するグループが使用されます。

設定	説明
radius!nas-ip-address	この値はRADIUSサーバに送信されます。ブランクのままにすると、サーバへのコネクションに使用されたインターフェースのアドレスが使用されます。
radius!nas-identifier	この値はRADIUSに送信されます。

TACACS+ 認証コード

TACACS+ 認証コードについて、以下の設定を行うことができます。

設定	説明
tacacsplus!server	TACACS+サーバのIPまたはホスト名です。
tacacsplus!port	TACACS+サーバに接続するためのポートです。
tacacsplus!timeout	TACACS+サーバへのコネクションのタイムアウト時間(秒単位)です。
tacacsplus!secret	TACACS+サーバと共有する秘密鍵です。
tacacsplus!authtype	使用する認証タイプです。PAPまたはACSIIに設定できます。
tacacsplus!groupsvc	各ユーザのグループフィールドを提供するTACACS+の「service」です。
tacacsplus!groupfield	各ユーザのグループを提供するTACACS+の「service」フィールドです。
tacacsplus!fallbackgroup	tacacsplus!groupsvc が定義されていないか、TACACS+サーバからユーザにグループ値が提供されていない場合、ここで指定するグループが使用されます。指定しない場合、TACACS+で定義されたグループを持たないユーザはアクセスを拒否されます。

認証コードのテスト

認証コードをテストするには、**System** (システム) > **Users** (ユーザ) > **Authenticators** (認証コード) > **Edit** (編集) ページの「Test Configuration (認証のテスト)」セクションを使用します。テストするユーザ名とパスワードを指定します。認証コードが実行され、成功した場合は、ユーザが属するパーミッショングループの名前が返されます。認証コードにより詳細なデバッグ情報も提供されるため、必要に応じて、この情報を基に設定を微調整したり修正したりできます。

使用例: Active Directoryに対する認証

LDAP認証コードを使用して、ユーザをActive Directoryデータベースに対して認証することができます。認証コードには以下の設定を使用します。

設定	値
ldap!server	dir.company.com
ldap!port	389
ldap!ssl	STARTTLS
ldap!basedn	OU=Company Users, DC=company, DC=local
ldap!filter	sAMAccountName=%u
ldap!dnmethod	Construct
ldap!binddn	%u@company.local
ldap!groupattr	memberOf
ldap!groupfield	CN

この認証コードを、ユーザ名とパスワードを指定してテストします。代表的な出力として、以下が作成されます。

```
Authenticating as username@company.local
Authenticated
Executing LDAP search, parameters:
    filter: sAMAccountName=username
    basedn: OU=Company Users,DC=company,DC=local
dir.company.com(fd6e:9138:1b6c:6401::a3e:9850:389): Connected
Bound as user username@company.local
Found group record, DN: CN=User Name,OU=Company Users,DC=company,DC=local
Found groups: Demo
No ldap!fallbackgroup defined
Groups returned by authenticator: Demo
SUCCEEDED, group: Demo
```

パーミッショングループ

パーミッショングループは、Traffic Manager管理サーバの各部分に対するアクセス権を様々なユーザに割り当てるために使用します。全てのローカルユーザとリモートユーザはいずれかのグループに属し、これらのグループによって、Traffic Managerの様々な部分に対するユーザのアクセス権が定義されます。デフォルトでは、ローカルのニーズに応じて、変更または削除できる定義済みグループがいくつか提供されます。新たにカスタマイズした任意の数のグループを追加することもできます。

新しいグループを追加するには、「Group name (グループ名)」ボックスに名前を指定し、**Update** (更新)をクリックします。変更内容が適用されたら、新しいグループ名をクリックします。

各グループには、基本的な「Description (記述)」および「Login Timeout (ログインタイムアウト)」設定があります。Description (記述)には、このグループの目的を示す簡単な説明を指定できます。また、Login Timeout (ログインタイムアウト)には、非アクティブな管理サーバセッションを終了するまでの時間を指定します。このグループに属する全てのユーザは、このタイムアウト値を継承します。

グループ内のユーザパーミッションの編集

Traffic Managerは、管理UI内で使用できるページの完全なリストを、それぞれのパーミッション設定とともに表示します。「admin (管理)」グループでは、全てのアクションが許可されます。新規作成されたグループは、作成時点では全ての許可が「None (なし)」に設定されます。

各グループでは、それぞれのページに対する許可を以下のいずれかに設定できます。

- **None:** グループメンバはこの機能を使用できません。
- **Read Only:** グループメンバは、このページに関連付けられた項目を表示できますが、変更することはできません。
- **Full:** メンバは、このページに関連付けられた項目を表示および変更できます。

Catalog (カタログ) > **SSL Certificates** (SSL証明書)ページを例として使用して、許可を「None (なし)」に設定した場合、このグループのメンバは、「SSL Certificates (SSL証明書)」カタログページを表示できません。Read Only許可を指定した場合、メンバはこのページを表示できますが、変更することはできません。許可を「Full (フル)」に設定すると、制限は適用されません。



一部のページ(カタログ概要ページなど)には、設定可能な項目がありません。また、特定ページへのアクセスをブロックしても、階層内でそのページの下位にあるページへのアクセスは制限されず、これらのページでの設定も制限されません。

注記

System (システム) > **Users** (ユーザ) ページに対するアクセス権を持つグループは、各自のアカウントの詳細を編集できます。これらの権限を与えるユーザを設定する場合は、注意することをお奨めします。

外部プログラムパーミッションに関するアドバイス

アクションプログラムおよびモニタプログラムは、Traffic Manager 自体と同じUNIXユーザ権限で実行されるため、これらの項目を制御するページへの権限を許可する際には注意することをお奨めします。

影響を受ける特定の権限は以下のとおりです。

- Extra Files (外部ファイル) > Action Programs (アクションプログラム)
- Extra Files (外部ファイル) > Monitor Programs (モニタプログラム)
- Monitors (モニタ) > Edit (編集)
- Alerting (アラート) > Actions (アクション) > Edit (編集)

ログインタイムアウト

ログインタイムアウトは分単位で測定されます。Traffic Managerが非アクティブな状態でこの時間が経過すると、管理UIとのセッションが終了します。セッションは、ブラウザウィンドウを閉じたり、**Logout** (ログアウト) をクリックして手動でログアウトした場合にも、自動的に終了します。

ログインタイムアウトは、パーミッショングループのプロパティです。**System** (システム) > **Users** (ユーザ) > **Permissions Groups** (パーミッショングループ) をクリックし、変更するパーミッショングループの名前をクリックします。「Inactivity Timeout (非アクティビティタイムアウト)」に新しいタイムアウト値を設定し、**Update** (更新) をクリックします。

サスペンドされているユーザ

このページでは、サスペンドされている1人以上のユーザを再度有効化することができます。ユーザのサスペンド理由として、最大許容ログイン試行回数の超過や、その他の管理上の理由があります。

再度有効化する各ユーザの横にあるボックスを選択し、**Enable Selected Users** (選択したユーザの有効化) をクリックします。

System (システム) > **Users** (ユーザ) > **Local Users** (ローカルユーザ) > **Edit** (編集) ページで **Status** (ステータス) を「Active (アクティブ)」に設定しても、個々のユーザを有効化できます。

ログインセキュリティとログイン行動

System (システム) > **Users** (ユーザ) > **Local Users** (ローカルユーザ) > **Password Policy** (パスワードポリシー) ページでパスワードを定義することにより、ユーザのログインパスワードをコントロールし、これらのパスワードに制限を適用することができます([ローカルユーザ](#)を参照)。これに加え、ログイン行動設定を定義することにより、システムのユーザに対するコントロールとセキュリティ対策を強化することができます。**System** (システム) > **Global Settings** (グローバル設定) ページの「Login and Security Settings (ログインおよびセキュリティ設定)」セクションには、設定が多数表示されます。これらの設定は大まかに以下の2つのテーマに分かれています。

UI画面バナー

システムの使用時にユーザに適切なメッセージまたはリマインダを提供する必要がある場合、ここで定義できます。

設定	説明
login_banner	管理UIまたはTraffic ManagerSSHコマンドラインにログインしようとするユーザに表示する、バナーのテキストメッセージです。
banner_accept	管理UIへのログイン前に、ユーザがlogin_bannerのテキストを明示的に受け入れて同意する必要があるかどうかを指定します。そのためのチェックボックスがログインページに追加されます。
uipage_banner	管理UIの各ページの上部および下部に表示するテキストメッセージです。

ログインコントロール

ユーザが試行できるログイン回数と、この回数を超えた場合の結果に関するコントロールを配置することができます。

設定	説明
max_login_attempts	ユーザアカウントがサスペンドされるまでに許容されるログインの失敗回数。値に0を指定すると、この機能が無効化されます。デフォルト: 0. ユーザアカウントは、max_login_suspension_timeで設定される遅延期間の後に、再アクティブ化されます。ただし、System (システム) > Users (ユーザ) > Local Users (ローカルユーザ) > Edit (編集) ページまたはSystem (システム) > Users (ユーザ) > Suspended Users (サスペンドされたユーザ) ページで、速やかに再アクティブ化することができます(サスペンドされたユーザを参照してください)。

設定	説明
	パスワードベースの認証でログイン試行制限を使用する場合は、注意することをお奨めします。DoS攻撃によって、管理ユーザのログインでTraffic Managerにアクセスできなくなる可能性を防止するために、ファイアウォールまたは独自のユーザ命名規則など、必ず追加のセキュリティ対策を利用してください。
max_login_external	外部で認証された(LDAP、RADIUS、またはTACACS+)ユーザが max_login_attempts 回ログインに失敗した場合に、このユーザをサスペンドするかどうかを指定します。
max_login_suspension_time	max_login_attempts 回ログインに失敗した後でユーザアカウントをサスペンドする時間。
login_delay	ログインに失敗してから次のログインを試行できるようになるまでの待ち時間。デフォルト: 4.

ログイン情報バナー

 Last successful login by admin: 2011-02-24 12:35:03 +0000 from 10.100.1.14 (UI) on coeus.
Failed login attempts since then: 1, last was: 2011-02-24 13:41:31 +0000 from 10.100.1.14 (UI) on coeus.

ホームページの上部に表示される情報バナーには、現在のユーザがこのシステムで成功した前回のログイン試行についての情報が表示されます。また、それ以降に失敗した試行も表示されるため、ログイン信用証明データが予測外または不正に使用された場合に、それを特定することができます。

イベントおよび監査ログ

イベントログには、クラスタ内の全てのTraffic Managerからのシステムメッセージおよび設定メッセージ、警告、およびエラーが含まれます。Traffic Manager管理UIのHome (ホーム)ページには、イベントログの短縮バージョンがあります。完全なログを表示するには、**Examine Logs** (イベントログ)または**Diagnose** (診断) > **Event Log** (イベントログ)をクリックします。

監査ログは、最新のログイン試行、設定の変更(ユーザおよび送信元IP別)、およびユーザログアウトを監視します。また、コントロールAPIおよびREST APIを使用して行われた変更の記録も行います。監査ログを表示するには、Traffic Manager管理UIの**Diagnose** (診断) > **Audit Log** (監査ログ)をクリックします。

仮想アプライアンスおよびクラウドインスタンスの場合は、Traffic Managerがイベントログと監査ログのローテーションを自動的にコントロールします。ログは、現在のログで50MBの最大ファイルサイズに達するとローテーションします。ログがローテーションするときに、その時点までのログエントリの新しいアーカイブファイルが作成され、現在のログがクリアされます。その後、アーカイブファイルは「gzip」で圧縮され、以下のフォーマットのファイル名が付けられます。

- **監査ログの場合:** audit- <date> - <sequence> .gz
- **イベントログの場合:** audit- <date> - <sequence> .gz

上記の <sequence> は、同じ日に複数回ローテーションが行われた場合に付けられる連番です。Traffic Managerソフトウェア製品の場合は、独自のローテーションポリシーを制定する必要があります。

アーカイブされた監査ログファイルにアクセスするには、**Diagnose** (診断) > **Audit Log** (監査ログ) ページの **Manage archived audit logs** (アーカイブされた監査ログの管理) をクリックします。アーカイブされたイベントログファイルにアクセスするには、**Diagnose** (診断) > **Event Log** (イベントログ) ページの **Manage archived event logs** (アーカイブされたイベントログの管理) をクリックします。

アーカイブされた各ログは、表示されるコントロールを使用してダウンロードしたり削除したりできます。



監査ログおよびイベントログファイルのローテーションのためにTraffic Managerが使用するデフォルトの最大ファイルサイズは、システムレベルの環境変数AUDITLOG_ROTATE_THRESHOLDまたはEVENTLOG_ROTATE_THRESHOLDをそれに応じて設定することによりオーバーライドできます。特定のアプライアンス/クラウド製品には、この環境変数の代わりに実装される自動しきい値がある場合があります。詳細については、ご使用の製品の『*Pulse Secure Virtual Traffic Manager: Installation and Getting Started Guide*』を参照してください。

一般的なセキュリティに関する配慮事項

正規表現の使用

正規表現は、データの検索、一致、および操作の各機能を提供します。これらが提供する範囲で達成可能なものは定量化または分析が困難です。このため、これらが露出させる脅威の実効表面積は完全に列举するには広すぎます。このセクションでは、正規表現を使用する際のセキュリティ上の意味に関する情報およびコンテキストを説明します。

正規表現はTraffic Managerの特定の機能のために設定できます。TrafficScriptルールで使用できる機能を通じて、任意の方法で 사용할 こともできます。

非常にスキルの高い攻撃者には次の2つのベクトルを利用できる可能性があります。

1. **サービス妨害:**「ReDoS」とも呼ばれます。正規表現の「悪意のある正規表現」を作成することは可能で、それが実行されると、メモリまたはCPUのかなりの量が消費されます。
2. **データ流出:** CPUキャッシュベースのサイドチャンネル攻撃(「Spectre」など)の発見から、正規表現はウィジェットを生成するために使用でき、ジャストインタイムコードジェネレータを通じて、Traffic Managerプロセスからデータを漏洩させるために使用できます。

こうした攻撃を準備したり実行したりするのは、容易なことでないことを強調しておきます。しかしながら、このセクションはプロアクティブな緩和およびポリシー決定のためのコンテキストを与えることを目的としています。以下の提案はこうした脅威の対処する方法を示すものです。

1. 正規表現の作成で使用されたデータをサニタイズします。最も単純なケースでは\Qおよび\Eを使用して、提供されたデータが正規表現構文の一部と見なされないようにします。
2. Traffic Managerのグループパーミッション機能を使用して、最小権限の原則に従い、管理者に必要な権限のみを割り当てます。

Traffic ManagerのコントロールAPI

本章では、Traffic ManagerのSOAPベースのコントロールAPIについて説明します。詳細については、製品に付属の『*Pulse Secure Virtual Traffic Manager: Control API Guide*』を参照してください。



Traffic ManagerのREST APIの詳細については、インストラクションを参照してください。

Traffic ManagerのコントロールAPIの概要



Traffic Manager製品によっては、Traffic ManagerコントロールAPIを利用できない場合があります。詳細については、ご使用の製品仕様を参照してください。

Traffic ManagerコントロールAPIは、Traffic Managerクラスタをリモートから管理および設定することのできる、規格に準拠したSOAP APIです。Webベースの管理UIの代わりとして使用できるため、Traffic Managerを他のアプリケーションから設定する場合に適しています。

例えば、侵入検知システムによってリモート攻撃の試行を検出する場合、Traffic ManagerコントロールAPIを使用して、疑わしいIPアドレスからの全ての接続を破棄するようTraffic Managerを設定することができます。

プロビジョニングシステムでは、Traffic Managerのサービスレベルモニタリング機能とSNMPインターフェースを使用してサーバノードのレスポンス時間をモニタリングすることにより、サーバの過負荷状態を検出することができます。サーバを追加してプロビジョニングした後に、Traffic ManagerコントロールAPIを使用してサーバプールを再設定することにより、Traffic Managerに通知できます。

Traffic ManagerコントロールAPIは、SOAPサービスをサポートしている任意のプログラミング言語およびアプリケーション環境で使用できます。一般には、Perl、Python、Java、およびC#が使用されます。



コントロールAPIの通信では、Traffic Managerの管理サーバポートが使用されます(デフォルトは9090)。場合によっては、管理サーバが代替ポート番号で実行されることがあります。以降の例では、ポート9090の参照を実際のポートに置き換えてください。Traffic Managerのネットワークポートの要件の詳細については、[Traffic Managerが使用するデフォルトポート](#)を参照してください。

例: 実行中の仮想サーバのリスト表示

PerlとSOAP::Lite

```
#!/usr/bin/perl -w

use SOAP::Lite 0.60;

# This is the url of the Traffic Manager Admin UI
my $admin_server = 'https://username:password@host:9090';

my $conn = SOAP::Lite
    -> uri('http://soap.zeus.com/zxtm/1.0/VirtualServer/')
    -> proxy("$admin_server/soap");

# Get a list of virtual servers
my $res = $conn->getVirtualServerNames();
my @names = @{$res->result};

# Establish which are enabled
$res = $conn->getEnabled( \@names );
my @enabled = @{$res->result};

# Print those which are enabled
for( my $i = 0; $i <= $#names; $i++ ) {
    if( $enabled[$i] ) {
        print "$names[$i]\n";
    }
}
```

この例を以下のように実行します。

```
$ ./listVS.pl
Main website
Mail servers
Test site
```

C SharpまたはMono

```
using System;
using System.Net;
```

```
using System.IO;
using System.Security.Cryptography.X509Certificates;

public class AllowSelfSignedCerts : ICertificatePolicy {
    public bool CheckValidationResult(
        ServicePoint sp, X509Certificate cert,
        WebRequest request, int problem )
    {
        return true;
    }
}

public class listVS {

    public static void Main( string [] args )
    {
        System.Net.ServicePointManager.CertificatePolicy =
            new AllowSelfSignedCerts();

        string url= "https://host:9090/soap";
        string username = "username";
        string password = "password";

        try {
            ZXTM.VirtualServer p = new ZXTM.VirtualServer();
            p.Url = url;
            p.Credentials = new NetworkCredential( username, password );

            string[] names = p.getVirtualServerNames();
            bool[] enabled = p.getEnabled( names );

            for ( int i = 0; i < names.Length; i++ ) {
                if( enabled[i] ) {
                    Console.WriteLine( "{0}", names[i] );
                }
            }
        } catch ( Exception e ) {
            Console.WriteLine( "{0}", e );
        }
    }
}
```

このコードは、.NET 1.1 SDKおよびMonoで動作します。.Net 1.1を使用する場合、この例を以下のようにコンパイルして実行します。

```
C:\> wsdl -o:VirtualServer.cs -n:ZXTM VirtualServer.wsdl
C:\> csc /out:listVS.exe VirtualServer.cs listVS.cs
C:\> listVS.exe
Main website
Mail servers
Test site
```

Monoを使用する場合、以下のようにコンパイルして実行します。

```
$ wsdl -o:VirtualServer.cs -n:ZXTM VirtualServer.wsdl
$ msc /out:listVS.exe /r:System.Web.Services \
    VirtualServer.cs listVS.cs
$ listVS.exe
Main website
Mail servers
Test site
```

その他の例

その他のプログラム言語を使用した追加のサンプルについては、『*Pulse Secure Virtual Traffic Manager: Control API Guide*』を参照してください。

コマンドラインインターフェース

本章では、Traffic Managerのコマンドラインインターフェース(CLI)について説明します。

はじめに

通常、Traffic Managerの管理は、Webベースの管理UIを使用して行います。設定をスクリプト化したり他のシステムと統合する場合のために、Traffic Managerには業界標準のSOAP APIおよびREST APIも用意されています。

SOAPベースのコントロールAPIやREST APIを使用するアプリの作成には時間がかかり、小規模で単純なタスクまたはその場限りの設定には適さない場合があります。Traffic Managerの全機能にすばやくアクセスするには、CLIを使用する方法もあります。

CLIはインタラクティブなシェルとして機能し、スクリプティングも可能です。CLIは、コントロールAPIを使用してTraffic Managerクラスタと通信します。また、CLIで使用できるコマンドは、コントロールAPIで提供されるメソッドに対応しています。



コントロールAPIの通信では、Traffic Managerの管理サーバポートが使用されます(デフォルトは9090)。場合によっては、管理サーバが代替ポート番号で実行されることがあります。以降の例では、ポート9090の参照を実際のポートに置き換えてください。Traffic Managerのネットワークポートの要件の詳細については、[Traffic Managerが使用するデフォルトポート](#)を参照してください。

CLIへのアクセス

CLIを起動するには、以下のプログラムを使用します。

```
ZEUSHOME/zxtm/bin/zcli
```

CLIにアクセスするには、SSHを使用して仮想プライアンス/クラウドインスタンスに直接ログインし([Traffic Managerへのリモートログイン](#)を参照)、zcliコマンドを入力します。または、zcliコマンドの引数としてユーザとホストを指定することにより、リモートから接続することもできます。

```
zcli [user@]host:port
```

以下に例を挙げます。

```
zcli admin@ztml:9090
```

zcliの使用方法和オプションを表示するには、以下のように引数として「--help」を指定します。

```
zcli --help

Usage: ./zcli [OPTIONS] [user@][host:port] [script file]
Configure the Traffic Manager from a script file or standard input.

--user USER      Set the username (default: admin)
--passfile FILE  Read the password from this file
--nossll         Your Traffic Manager does not have SSL enabled admin server
--verbose        Verbose output (for testing)
--continue       When running a script, continue even if an error occurs
--help           Show this help
--version        Show the version of this program
--formatoutput   Use 'human readable' output, even in script mode
--json           Use strict JSON output, even in interactive mode
--timeout SECONDS Set timeout for commands
```

If no host:port is specified, the local Traffic Manager will be used.

許可

CLIは、コントロールAPIに対する許可を持つユーザ名とパスワードを使用して、Traffic Managerの認証を受ける必要があります([ネットワーク設計](#)を参照)。「admin (管理)」グループの全てのユーザは適切な許可を持っており、必要に応じてこれらの許可を他のユーザグループにも与えることができます。

デフォルトでは、CLIは、ユーザ名admin(管理)を使用してTraffic Managerと通信します(--userオプションを使用して代替的なユーザ名を選択することもできます)。

```
$ zcli
Please enter your password for user 'admin' on the local Traffic Manager
Connected to 127.0.0.1:9090
admin@127.0.0.1 >
```

コネクションが確立されると、CLIに(Traffic Managerのユーザ名とホスト名が示された)プロンプトが表示され、コマンド入力待ち状態になります。

```
admin@127.0.0.1 >
```

コマンドは、一度に1つずつ入力できます。コマンド履歴が維持されるため、上下の矢印キーを押して以前入力したコマンドを確認できます。多くのコマンドではオートコンプリート機能も利用できます。Tabキーを1回押すと、途中まで入力したコマンドやオブジェクト名が補完されます。「Tab」キーを2回押すと、有効なオプションのリストが表示されます。

helpコマンドを使用すると、CLIの概要と、使用可能なコマンドに関する詳細を参照できるヘルプトピックが表示されます。

```
Developer mode
admin@10.62.165.97 > help
The Pulse Secure Virtual Traffic Manager command-line interface provides direct
access to all of the Control API calls that control and configure your traffic
manager, together with additional commands to monitor it and its configuration.

Getting started

help overview - brief overview of the command line interface
help syntax   - explain the syntax for entering commands and data (important!)
help output   - explain the output format for commands
help commands - describe all of the commands
help classes  - describe all of the API classes
help <class>  - show the methods for a particular class, e.g. "help Pool"
help <cmd>    - describe a command, e.g. "help Pool.addNode" or "help show"

info          - show a brief summary of this traffic manager
catalogs      - list all configuration objects in this traffic manager
show          - show specific configuration, e.g. "show Pool"
stats         - show traffic manager statistics/counters, e.g. "stats webcache"

admin@10.62.165.97 >
```

コマンド

使用可能なコマンドを調べる場合に、詳細なリストを表示するにはcommandsと入力し、コマンドとオブジェクトの簡易リストを表示するにはTabコンプリート機能を使用します(キーボードのTabキーを2回押す)。

コマンドは、以下の3つの主要グループに分類されます。

1. コントロールAPIメソッド

これらのコマンドは、以下のようにClass.method [arguments...]の形式を取ります。

```
admin@127.0.0.1 > Pool.getNodes "Web Servers"
[ "92.52.65.222:80", "92.52.65.213:80" ]
```

2. ヘルパコマンド

これらのコマンドは、Traffic Managerの設定の概要を表示します。主なコマンドは以下のとおりです。

- info: このTraffic Managerの概要を表示します。
- catalogs: このTraffic Managerの全設定オブジェクトのリストを表示します。
- show: 特定の設定を表示します。例: show Pool
- stats: Traffic Managerの統計/カウンタを表示します。例: stats webcache

3. CLIヘルプ

- `help syntax`: コマンドおよびデータを入力するときの構文について説明します。
- `help output`: コマンドの出力フォーマットについて説明します。
- `help commands`: 全てのコマンドについて説明します。
- `help classes`: 全てのAPIクラスについて説明します。
- `help <class>`: 特定クラスのメソッドを表示します(例: `help Pool`)。例: `help Pool`
- `help <cmd>`: コマンドについて説明します。例: `help Pool.addNodes`または`help show`

コントロールAPIメソッド

CLIでは、全てのコントロールAPIメソッドにアクセスできます。コントロールAPIメソッドに対応するコマンドは、『*Pulse Secure Virtual Traffic Manager: Control API Guide*』に記載されているクラス名およびメソッド名で構成されます。各コマンドの詳細に関してはこのドキュメントを参照してください。

例えば、全ての仮想サーバのリストを表示するには、以下のコマンドを使用します。

```
admin@127.0.0.1 > VirtualServer.getVirtualServerNames
[ Intranet, "Secure Site", webmail ]
```

CLIコマンドの引数

コマンドの引数は、スペースで区切った名前として指定します。引数間のカンマはオプションです。スペースまたはその他の特殊文字を含む引数値は、構文上異なる意味に解釈される可能性があるため、「引用符」で括ってください。リストは角括弧([])の中に入れます。

例えば、『*Pulse Secure Virtual Traffic Manager: Control API Guide*』には、Poolクラスの「`getNodes`」メソッドが以下のように記載されています。

`getNodes( names ) throws ObjectDoesNotExist`

指定した各プールのノードのリストを取得するには、以下のように記述します。

```
String[][]
getNodes( String[] names
);
```

このメソッドに対応するCLIコマンドは`Pool.getNodes`です。マニュアルには、このコマンドがプール名の配列(`String[] names`)を使用することが示されています。2つのプールに対してこのメソッドを呼び出すコマンド行は以下ようになります。

```
admin@127.0.0.1 > Pool.getNodes [ pool1 pool2 ]
{ pool1: [ "serverA:80", "serverB:80" ],
  pool2: [ "serverC:80", "serverD:80" ] }
```

Pool.getNodesの出力は、String[][]、すなわち配列の配列として宣言されます。指定した各プールについて、ノードの配列が返されます。

便宜上、インタラクティブモードでは、CLIはコマンド出力にインデックスを付け、参照先のプール名のプレフィックスを付けてノードのリストを表示します。

単一引数の使用

コントロールAPIの多くのコマンドでは引数のリストを使用できるため、一度に複数のオブジェクトに対して操作することができます。単一の項目に対してコマンドを使用する場合、以下のように配列を省略して簡略化することができます。

```
admin@127.0.0.1 > Pool.getNodes pool1
[ "serverA:80", "serverB:80" ]
```

ワイルドカードの使用

*記号は、複数のオブジェクトに一致させることのできるワイルドカードです。

```
admin@127.0.0.1 > VirtualServer.getDefaultPool *
{ Intranet: pool1,
  "Secure Site": "Secure pool",
  webmail: "Mail servers" }
```

名前の一部にワイルドカードを使用することもできます。例えば、名前が「Pool」で始まっている全てのプールでkeepaliveを有効化する場合、以下のようにします。

```
admin@127.0.0.1 > Pool.setKeepalive Pool* 1
```

複雑な引数

一部のコマンドは、さらに複雑な引数構造体を必要とします。例えば、VirtualServer.addVirtualServerは、『*Pulse Secure Virtual Traffic Manager: Control API Guide*』で、以下のように定義されています。

```
void addVirtualServer(
    String[] names,
    VirtualServer.BasicInfo[] info )
```

BasicInfo構造体は以下のように定義されています。

```
struct VirtualServer.BasicInfo {  
    # The port to listen for incoming connections on.  
    Integer port;  
  
    # The protocol that this virtual server handles.  
    VirtualServer.Protocol protocol;  
  
    # The default pool that traffic to this virtual server  
    # will go to.  
    String default_pool;  
}
```

構造体をCLIで使用するときは「{ }」および「[]」中括弧で括り、各メンバをキー:値ペアの形式で指定します。以下に例を挙げます。

```
admin@127.0.0.1 > VirtualServer.addVirtualServer Intranet { port: 80, protocol: http,  
default_pool: pool1 }
```

コマンドで想定される引数については、CLIから利用可能なヘルプで参照できます。

```
admin@127.0.0.1 > help VirtualServer.addVirtualServer
```

カスタム設定のセット

Traffic Manager設定システムの任意のname:value設定ペアは、CLIを使用して保存したり、読み込んだりできます。この設定はクラスタ全体で反復され、REST API、SOAP API、およびCLIを介してのみアクセスすることができます。

カスタム設定の保存および読み込みに関連付けられているコマンドの一覧は、プロンプトでhelp Customと入力することで確認できます。

カスタム設定を保存するには、Custom.setStringsListsコマンドを使用します。以下は、「counters」という名前のリストの配下にname:valueのペアを設定する場合の例です。

```
Custom.setStringLists [counters], [ { "name": "Test_Counter", "value" : ["12345",  
"67890"]} ]
```

必要に応じて、{"name":<name>, "value":<value>}の項目を追加することによって、複数のname:value設定ペアをリストに追加できます。

「counters」設定を読み込むには、以下のコマンドを使用します。

```
Custom.getStringLists counters
```

このシステムを使用すると、最初にname:valueペアごとに新規リストを作成し、さらにその中で、カスタム設定を論理グループに区分することができます。value自体は、単一の項目でも、項目のリストでも構いません。

内蔵コマンド

CLIには、簡単に設定を表示または変更できる追加のコマンドが多数用意されています。詳細なヘルプおよび使用方法は、`help <commandname>`を使用して参照することができます。

- `info`: 管理UIのホームページと同様に、このTraffic Managerの概要情報を表示します。クラスタ内のTraffic Manager、設定されている仮想サーバとプール、トラフィックIPアドレス、設定または操作エラー、およびイベントログに記録されている最新エントリが表示されます。
- `connect`: 別のTraffic Managerに接続します。CLIでは、ローカルマシンと同じクラスタ内にはTraffic Managerなど、リモートのTraffic Managerと通信することができます。スクリプトファイルにはTraffic Managerのホスト名、ユーザ名、およびパスワードを全て記述できるため、このコマンドはスクリプトを作成するときにも役立ちます。
- `errorlog`: Traffic Managerのイベントログに記録されている最新のメッセージを表示します。
- `catalogs`: モニタ、TrafficScriptルール、Java Extensionsなど、設定カタログのコンテンツのリストを表示します。
- `show`: Traffic Managerの設定オブジェクトの設定を表示します。以下に例を挙げます。
 - `show VirtualServer Intranet`: 仮想サーバ「Intranet」の概要を表示します。
 - `show Pool`: 全てのプールの概要を表示します。
 - `show VirtualServer customer*`: 仮想サーバcustomer*の概要を表示します。

以下のように、オブジェクト名のみを指定してshowを使用することもできます。この場合、指定した条件に一致する全てのオブジェクトの概要が表示されます。

- `show Staging`: 「Staging」という名前の全てのオブジェクトの概要を表示します。
- `show *`: 全てを表示します(時間がかかる場合があります)。

showコマンドを使用して、現在Traffic Managerによって処理される全ての接続の詳細を指定することもできます(`show connections`コマンドを使用します)。出力は、管理UIの**Activity** (アクティビティ) > **Connections** (コネクション)ページで表示される接続情報に基づいており、コネクションの進行中には以下のデータ項目がリストされます。

データ項目	説明
From	接続するクライアントのIPアドレスとポート
Via	クライアントが接続したローカルIPアドレスとポート
To	コネクションに使用されるバックエンドノード
State	以下のいずれかのような、現在のコネクション状態を表示します。 C: クライアントへのコネクションが切断された f: 接続完了 R: クライアントから読み込み中 W: クライアントへ書き込み中 X: クライアントリクエストに対してルールを実行中 c: バックエンドノードに接続中 w: バックエンドノードへ書き込み中 r: バックエンドノードから読み込み中 C: クライアントとのコネクションを切断中 K: クライアントコネクションをkeepalive状態で保持中
仮想サーバ	このコネクションを処理する仮想サーバ
ルール	コネクションで最後に実行されたルール
Pool	使用されたプール
BytesIn	クライアントから受信されたバイト数
BytesOut	クライアントへ送信されたバイト数
Timings	<num1> / <num2> / <num3>のフォーマット。意味は以下のとおりです。 <num1>: クライアントがこのコネクションを確立してからの時間 <num2>: クライアントコネクション上で最後にデータが送信されてからの時間

データ項目	説明
	<num3>: 別のバックエンドコネクション上で最後にデータが送信されてからの時間
Retries	サーバがバックエンドノードへのコネクションを試行する回数
SLM	使用しているサービスレベルモニタリングクラス
VSBW	仮想サーバが使用した帯域クラス
PoolBW	プールが使用した帯域クラス
Code	バックエンドノードにより返されたレスポンスコード(プロトコルによって異なる)
リクエスト	実行されたリクエストの概要。HTTPの場合はリクエストされたURL

 追加の情報がコネクションに適用可能と識別された場合には、追加のフィールドが表示されることもあります。

show (表示)の引数として指定できるオブジェクトタイプは、以下のとおりです。

action, authenticator, bandwidth, ca, certificate, clientcertificate, connections, event, global、info, java, licensekey, monitor, persistence, pool, protection, rate, rule, slm, trafficip, virtualserver

- stats: Traffic Managerの様々なオブジェクトのカウンタと統計情報を表示します。以下に例を挙げます。
 - stats VirtualServer Web - 仮想サーバ「Web」の統計情報を表示します。
 - stats Pool - 全てのプールの統計情報を表示します。
 - stats * - 全ての統計情報を表示します。

以下のように、オブジェクト名のみを指定してstatsを使用することもできます。この場合、指定した条件に一致する全てのオブジェクトの統計情報が表示されます。

- stats Staging - 「Staging」という名前の全てのオブジェクトの統計情報を表示します。
- stats * - 全ての統計情報を表示します(時間がかかる場合があります)。

指定できる統計オブジェクトは以下のとおりです。

authenticator, bandwidth, event, interface, node, perpoolnode, pool, protection, rate, rule, session, slm, slmpernode, ssl, system, trafficip, virtualserver, webcache, zxtm

- help: ヘルプトピックを表示します。以下に例を挙げます。
 - help overview - コマンドラインインターフェースの概要を表示します。
 - help syntax - コマンドおよびデータを入力するときの構文について説明します。
 - help output - コマンドの出力フォーマットについて説明します。
 - help commands - 全てのコマンドについて説明します。
 - help classes - 全てのAPIクラスについて説明します。
 - help <class>: 特定クラスのメソッドを表示します。例: help Pool
 - help <cmd> - コマンドについて説明します。例: help Pool.addNodeまたはhelp show
- rule: Traffic ManagerのTrafficScriptルールの上ロード、ダウンロード、および構文チェックを行います。
- getbackup: Traffic Managerからバックアップをダウンロードします。
- putbackup: バックアップファイルをTraffic Managerにアップロードします。
- download: オフラインで編集またはアーカイブするために、Traffic Managerから特定の設定ファイルをダウンロードします。
- upload: 特定の設定ファイルをTraffic Managerにアップロードします。
- edit: Traffic Managerから特定の設定ファイルをダウンロードして編集し、変更後のファイルをアップロードします。
- watch: コマンドを繰り返し実行し、出力を監視します。ログの監視や統計情報の記録など、変化をモニタリングする場合に役立ちます。以下に例を挙げます。

```
admin@127.0.0.1 > watch 10 stats webcache
```

これにより、全てのWebキャッシュの統計情報が表示されます。この情報は10秒ごとに更新されます。

CLIのスクリプティング

通常、CLIは、コマンドをキーボードから入力して即座に実行できるインタラクティブモードで動作します。

ユーザ入力を行わずに、CLIコマンドをスクリプトとして実行することも可能です。例えば、以下のスクリプトは、Webキャッシュの統計情報を出力し、キャッシュから一部の項目をクリアし、その後再度Webキャッシュの統計情報を出力します。

```
System.Stats.getWebCacheEntries  
  
System.Cache.clearCacheContentItems "Intranet" "http" "www.example.com" "**/product/*"  
  
System.Stats.getWebCacheEntries
```

このスクリプトをローカルマシン上のファイルに保存し、以下のように実行します。

```
$ ZEUSHOME/zxtm/bin/zcli /home/user/webcache-script
```

CLIにより、このファイルが読み取られ、スクリプト内の各コマンドが順番に実行されます。スクリプトが適切に完了した場合、終了コード0 (ゼロ)が返されます。コマンドの実行中にエラーが発生した場合、スクリプトの実行が中止され、CLIはゼロ以外の戻り値で終了します。ただし、--continueがコマンドラインで指定されている場合、処理中に発生したエラーに関係なく、スクリプトは続けられます。この場合の戻り値は、CLIが何らかの内部エラー(不適切なパラメータの使用など)に直面した場合を除き、0のままです。

スクリプトを完全に自動化するには、ユーザ名とパスワードを指定する必要があります。これは以下の2通りの方法で行えます。

1. パスワードをファイルに保存し、このファイルをzcliスクリプトに渡します。

```
ZEUSHOME/zxtm/bin/zcli --user username --passfile /home/user/password  
/home/user/webcache-script
```

パスワードが含まれるファイルは、他のユーザに読み取られないようアクセス権を適切に制限する必要があります。

2. 以下のように、connectコマンドをスクリプトの先頭に追加します。

```
connect localhost 9090 username password  
  
System.Stats.getWebCacheEntries  
  
System.Cache.clearCacheContentItems "Intranet" "http" "www.example.com" "**/product/*"  
  
System.Stats.getWebCacheEntries
```

スクリプトの出力

スクリプトとして実行されるCLIコマンドの出力は、インタラクティブモードのCLIコマンドの場合と多少異なります。インタラクティブモードでは、出力は読み取りやすいようフォーマットされて、スペースが挿入されたり、必要に応じて複数の行にわたったりすることもあります。スクリプトモードでは、各コマンドの出力は1行に表示され、スペースの挿入やフォーマットは行われません。

出力フォーマットはJSONフォーマットのため、他のプログラムでのインポートや解析を簡単に行えます。複数の入力(例えば、複数のプール名など)を取るコマンドの出力はテーブル形式で表示されないため、各項目がどの結果に対応しているのかが分かりにくくなります。

例えば、同じコマンドをインタラクティブセッションとスクリプトセッションで実行したときの出力を比較します。イ

ンタラクティブモード:

```
admin@127.0.0.1 > Pool.getNodes [ pool1 pool2 ]
{ pool1: [ "serverA:80", "serverB:80" ],
  pool2: [ "serverC:80", "serverD:80" ] }
```

スクリプトモード:

```
$ ZEUSHOME/zxtm/bin/zcli --user username --passfile /home/user/password
/home/user/getnodes
[["serverA:80","serverB:80"],["serverC:80","serverD:80"]]
```

読み取りやすい出力にするには、--formatoutput引数を指定してzcliを実行します。

zconfを使用したきめ細かい設定のインポート/エクスポート

この章では、「zconf」コマンドラインユーティリティについて説明します。このユーティリティを使用すると、個々のTraffic Manager設定オブジェクトのインポートとエクスポートを実行することができます。

はじめに

非常に多い数のTraffic Managerを定期的にプロビジョニングする場合、このプロセス中に行う作業の重複を最小限に抑えることにより、配備に要する時間的なコストを削減することができます。このプロセスでは、ポリシーの標準セットまたは設定の共通要素を多くのサービスに展開することになる可能性が高く、また多くのTraffic Managerに対して同じ設定の変更を行わなければならない場合もあります。

ポリシーや設定ファイルは、何らかの変更管理データベースを使用して保存することができます。ソースコードだけでなく、このような設定ファイルに対する変更を保存し、管理することのできるツールの一例として、Subversion (SVN)があります。Traffic Managerには、詳細レベルの設定のエクスポートおよびインポートを行ったり、これらの(部分的な)設定をtarアーカイブに保存したりできる、「zconf」というツールが用意されています。tarアーカイブは、必要に応じて抽出し、いずれかの変更管理データベースにチェックインすることができます。

zconfの使用

zconfを使用するには、以下のコマンドを実行します。

```
$ZEUSHOME/zxtm/bin/zconf
```

\$ZEUSHOMEが正しく設定されていないと、zconfユーティリティが正常に機能しないことに注意してください。このコマンドラインユーティリティには、ヘルプシステムが内蔵されています。引数を指定せずにこのプログラムを実行すると、以下のような出力が得られます(以下の例はTraffic Manager仮想アプライアンスのものです)。

```
root@ubuntu1:~# $ZEUSHOME/zxtm/bin/zconf
```

```
ERROR: No command specified.
```

```
Usage: /opt/zeus/zxtm/bin/zconf [COMMAND] [OPTIONS] [ARGUMENTS]
```

```
Commands:
```

```
copy
```

```
diff
export
getdefaultfilter
help
import
info
list
replicate
version
```

For more information on a given command execute:

```
/opt/zeus/zxtm/bin/zconf help <COMMAND>
```

この出力には、使用可能なzconfコマンドのリストが示されます。詳細については、各コマンドのヘルプを参照してください。

完全バックアップのエクスポート

Traffic Managerのクラスタ設定の完全バックアップを行う場合、zconfのexportコマンドを使用します。デフォルトでは、exportコマンドの出力はSTDOUTに出力されます。出力をファイルに保存するには、--file引数を使用するか、出力をSTDOUTからファイルにリダイレクトします(以下の例を参照)。多くのLinuxシステムでは、その後tarコマンドを使用してtarアーカイブのコンテンツを抽出することができます(Subversionリポジトリにチェックインするためなど)。

```
root@ubuntu1:~# $ZEUSHOME/zxtm/bin/zconf export > backup.tar
root@ubuntu1:~# ls
backup.tar
root@ubuntu1:~# tar xf backup.tar
root@ubuntu1:~# ls
backup.tar conf
root@ubuntu1:~# ls conf
actions      groups      monitors security  users
activitymonitor licensekeys PARTIAL services  VERSION_8.0
commkey      locations   rules      settings.cfg zxtms
events       locations.cfg scripts  TIMESTAMP
root@ubuntu1:~#
```

設定のリスト表示

設定のバックアップはプレーンテキストファイルに保存され、ディレクトリツリーとして整理されます。listコマンドを使用すると、現在稼働中のTraffic Managerの設定ファイルと、設定のバックアップまたはバックアップアーカイブのリストを表示することができます。例えば、現在のTraffic Managerに設定されている仮想サーバのリストを取得する場合、以下のように入力します。

```
root@ubuntu1:~# $ZEUSHOME/zxtm/bin/zconf list vservers
root@ubuntu1:~#
```

この出力には、vserversディレクトリが空であること、すなわち設定されている仮想サーバがないことが示されています。設定アーカイブ内の仮想サーバのリストを表示する場合は、以下のように入力します。

```
root@ubuntu1:~# $ZEUSHOME/zxtm/bin/zconf list vservers --conf-path backup/monitors/Full HTTP
pools/Bar
vservers/Foo
```

listコマンドのglob引数としてvserversを指定しましたが、プールおよびモニタのリストも出力に含まれています。これは、その仮想サーバを想定どおりに動作させるために必要な全ての依存オブジェクトが出力されるためです。

部分インポート

現在のディレクトリに設定アーカイブtarアーカイブがあるとします。(例えば「Foo」という名前の)仮想サーバとその全ての依存オブジェクトをインポートするには、以下のコマンドを使用します。

```
zconf import --include vservers/Foo backup.tar
```

このインポートにより、設定がクラスタの他のメンバに複製されます。



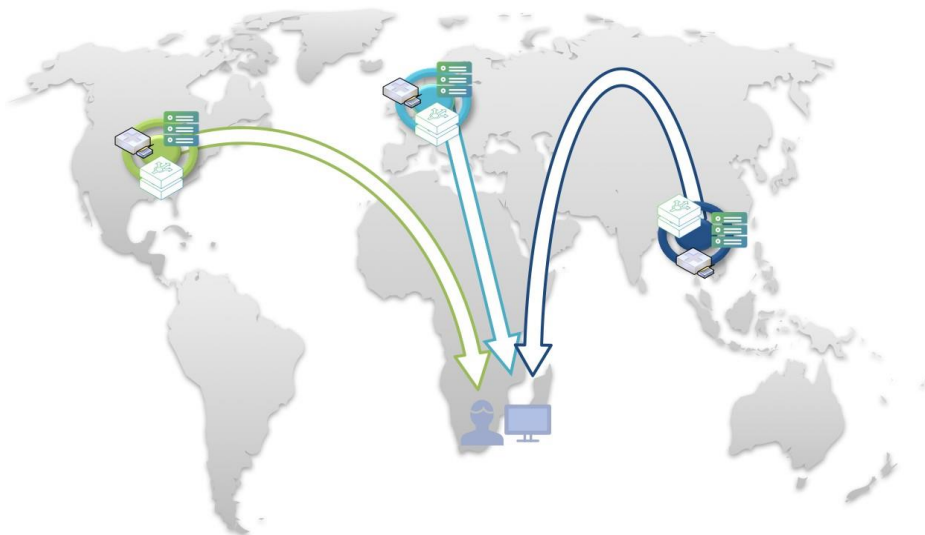
同様の機能は、管理UIの**System** (システム) > **Backups** (バックアップ) > **Partial Backups** (部分バックアップ)ページからも利用できます。詳細については、[System \(システム\)](#) > [Backups \(バックアップ\)](#) > [Partial Backups \(部分バックアップ\)](#)を参照してください。

マルチサイトクラスタの管理

本章では、複数のクラスタを複数のサイトにまたがる1つの集中管理されるTraffic Managerクラスタにまとめるアイデアを紹介します。

はじめに

本章で説明するマルチクラスタ管理機能は、Traffic Managerが備える高度な機能をベースに、分散した複数の物理、仮想、またはクラウドベースのTraffic Managerを管理するためのサポート機能を追加して作成されています。これらのクラスタの地理的な場所、稼働しているプラットフォーム、Traffic Managerインスタンスの数は様々です。マルチクラスタ管理機能では、これらの各Traffic Managerでホストされている特定のサービスを1箇所から集中管理することができます。



本章では、Traffic Managerシステムのロケーション固有の設定を維持しながら、世界各地に分散したTraffic Managerインフラストラクチャを単一クラスタとして集中管理する方法について説明します。

世界の複数の拠点でホストされるサービスにわたって地理を意識した負荷分散を行うグローバルサーバ負荷分散(GSLB)とこの機能を混同しないでください。Traffic ManagerへのGSLBの実装については、[グローバル負荷分散](#)で説明します。



本章では、本マニュアルの他のセクションで説明しているTraffic Managerのユーザインターフェースおよび一般的な設定コンセプトに関する知識があることを前提としています。先に進む前に、[Traffic Manager管理インターフェースの主要機能](#)を参照することを強くお奨めします。

アクティブ化と非アクティブ化

マルチサイト管理機能を有効化するには、管理UIの**System** (システム) > **Traffic Managers** ページにアクセスします。「Multi-Site Management (マルチサイト管理)」セクションで**Confirm** (確認) チェックボックスを選択して**Enable** (有効化) をクリックします。

同様に、同じ方法でマルチサイト機能を非アクティブにすることにより、Traffic Managerを元の状態に戻すことができます。この場合、まず定義されているロケーション設定のいずれかを全てのTraffic Managerのマスタ設定として定義する必要があります。その他全てのロケーション設定は、それぞれに保存されているロケーション固有の設定値とともに削除されます。マスタにするロケーションをドロップダウンリストから選択し、**Confirm** (確認) チェックボックスを選択して**Disable** (無効化) をクリックします。

 ロケーション設定が削除されても、GLBロケーションは非アクティブな状態で残ります。

主なコンセプト

ロケーション設定

基本的に、ロケーション設定はTraffic Managerのグループです。これらのTraffic Managerは、場合によっては特定の物理的な場所に存在し、クラスタ全体において同じサービスをホストすることが求められます。ロケーション設定は、**Catalogs** (カタログ) > **Locations** (ロケーション) ページで定義し、特定ロケーションに存在するものとして各Traffic Managerを指定します。

管理UI全体のサービスおよびその他の設定要素をロケーション固有にすることができます。ロケーション固有にしたサービスおよび設定は、そのロケーションに存在するものとして指定されたTraffic Managerでのみ使用されます。

このコンセプトにより、世界各地に存在する全てのTraffic Managerアプリケーション配信リクエストを単一の集中管理システムに集めることができます。この環境では、構成の複雑さに関わらず、単一の管理UIから全てのサービスを設定し、モニタリングすることができます。

 Traffic Managerは必ずしも物理的に特定の場所に存在する必要はありません。GLBロケーションとは対照的に、ロケーション設定のコンセプトは、主にTraffic Managerクラスタ内での仮想的なグループ化を可能にするためのメカニズムです。

クラスタ

マルチサイト構成におけるクラスタは、世界中で1箇所以上の場所に配備され、集中管理されるTraffic Managerのグループです。これによってフォールトトレランスを実現しながらアプリケーションサービスが提供されます。

Traffic Managerサービスの設定は、クラスタ内の全てのマシンに複製されます。マルチサイト管理では、非マルチサイト環境と同様に設定の多くが共有されますが、さらにこの機能が拡張され、特定のロケーションに存在するものとして指定されたTraffic Managerでのみアクティブになるロケーション固有の設定を行うことができます。これにより、一種のサブクラスタ化、すなわちローカルサービスの提供が可能になります。クラスタでは、Traffic Managerの管理UIを使用して設定全体を編集でき、編集した内容はその他全てのTraffic Managerに複製されます。

各ロケーションに固有の設定(ローカルDNSサーバなど)も特定のマシンに固有の設定(SNMP設定など)も、システム設定のうちのほんの一部です。ロケーション固有のシステム情報を設定するには、そのロケーションに含まれるいずれかのTraffic ManagerのUIを使用します。この方法で設定した情報は、同じロケーションに定義されている他のTraffic Managerに複製されます。マシン固有のシステム情報を設定するには、そのマシンのUIを使用する必要があります。

 リモートのTraffic Managerをクラスタに組み込む場合、最初にそのTraffic Managerに外部IPアドレスを指定する必要があります。また、新しいマシンを追加する前に、このアドレスと、特にポート番号9080と9090(またはTraffic Managerがコントロール通信と管理UIに使用するいずれかのポート)をクラスタ全体で利用できるか確認することをお奨めします。これを怠ると、クラスタ内での設定の複製時に問題が生じる可能性があります。外部IPの設定方法の詳細については、[Traffic Managerのロケーションの設定](#)を参照してください。Traffic Managerのネットワークポートの要件の詳細については、[Traffic Managerが使用するデフォルトポート](#)を参照してください。

デプロイシナリオ

 このセクションで使用されている「ロケーション」という言葉は**ロケーション設定**を意味します。さらに、全てのシナリオは、外部ロケーションへの適切なIP/ポート接続性を引き受けます(上記を参照)。

第2のTraffic Managerロケーションの作成と管理

サービスの利用者増加や高度化に伴い、世界の他の場所に並行して稼働する第2のTraffic Managerを配備する必要性が生じる場合があります。

- まず、新しいロケーションの新たな各Traffic Managerに必要なインストールと初期設定を行います。ただし、既存のTraffic Managerクラスタに組み込むと、追加する各マシンにクラスタ設定が複製され、システム固有でない設定が上書きされることに注意してください。
- 各Traffic Managerの管理UIから、**Join a Cluster** (クラスタに組み込む)ウィザードを実行して、既存のTraffic Managerクラスタに接続します。
- あらかじめ既存のクラスタに新しい第2のロケーション設定を定義しておくか、ウィザードの手順4で「new location」オプションを指定します。ウィザードで新しいロケーションを指定しても、ロケーション定義のposition設定に値は適用されません。この設定は手動で行う必要があります。
- 以降追加する各Traffic Managerについては、このウィザードの手順4で、新たに作成したロケーションを選択します。

新しい各Traffic Managerがクラスタに追加され、全ての設定がコピーされますが、第2のロケーション設定に存在するものとして指定されます。ロケーション固有としてこれまでに行われた、または今後行われる設定の変更は、クラスタ全体に適用されます。

マルチサイトクラスタへの新しいTraffic Managerの追加

Traffic Managerが管理する任意のロケーションにTraffic Managerを追加するには、追加するマシンの管理UIから**Join a Cluster** (クラスタに組み込む)ウィザードを実行します。これにより、クラスタの設定が新しいTraffic Managerにインポートされ、管理UIには、追加したTraffic Managerが管理対象マシンとして表示されます。

複数の既存Traffic Managerクラスタのマージ

通常のクラスタ組み込み操作では、組み込まれるマシンにクラスタの設定が継承され、既存の設定は全て削除されます。このシナリオでは、複数の設定をマージするため、全てのマシンの設定が維持されるよう十分に注意する必要があります。

Traffic Managerには、きめ細かい設定のインポート/エクスポート操作を実行する「zconf」コマンドラインツールが用意されています。このツールを使用すると、既存のサービスに悪影響を与えることなく、セカンダリクラスタの個々のサービスをマスタクラスタの設定にインポートすることができます。また、zconf内蔵の「dry-run」モードでインポート手順を実行すると、あらかじめ予測される結果を全て確認できるというメリットもあります。完全な使用構文および詳細については、zconfから直接利用可能なオンラインヘルプを参照してください。

このモードの目的は、サービスのダウンタイムを最小限に抑えることです。ただし、実際にはトラフィックIPアドレスが転送されるときにサービスが中断される可能性があるため、インポート操作はWebトラフィックが最小になる時間帯に行うのが最善です。

注記

いかなる場合も、この手順を実行する前に、全てのクラスタの設定をバックアップすることを強くお奨めします。また、期待どおりの結果を観察するために、以下の手順を最初に重要性の低いシステムで試すこともお奨めします。

複数の既存クラスタのマージ

1. 全てのクラスタ内の全Traffic Managerで最新のTraffic Managerソフトウェアが実行されていることを確認します。いずれかのクラスタをマスタクラスタとして指定します。
2. マスタクラスタで新しいセカンダリロケーションを設定します。
3. ファイル名の競合を最小限に抑えるために、全てのクラスタの仮想サーバ、ルール、プールなどの名前を、先頭にロケーション名を付けるなどしてユニークに識別できる名前に変更します。例えば、「Intranet」という名前を「london-Intranet」に変更します。これは必須ではありませんが、設定が競合する可能性を減らすことができます。zconfではデフォルトで、インポートの実行時に同じ名前の既存の設定ファイルが上書きされますが、これは切り替え可能です。
4. **System** (システム) > **Backups** (バックアップ) ページを使用してリモートクラスタの設定をエクスポートし、結果のtarアーカイブをマスタクラスタにコピーします。
5. zconfを使用して、設定のバックアップから仮想サーバ(/vservers)ディレクトリをインポートします。zconfは、自動的に各仮想サーバファイル内の依存関係に従い、関連するサポートファイルをインポートします。このときの唯一の注意点はJava Extensionsの使用であり、手動で/jarsおよび/servletsディレクトリにインポートする必要があります。
6. ここで、インポートしたこれらのサービスについて、新たに定義したリモートロケーションに対してのみ有効化のマークを付ける必要があります。これを行うには、**Services** (サービス) > **Virtual Server** (仮想サーバ) > **Edit** (編集) ページで、**Enabled** (有効化) 設定の横にある **Edit setting by location** (ロケーション別の設定の編集) アイコンをクリックします。この仮想サーバがリモートロケーションで有効化され、その他全てのロケーションでは無効化されていることを確認します。
7. トラフィックIPグループが使用されている場合、2つのクラスタに同じIPアドレスが設定されていると、サービスが競合する可能性があります。マスタクラスタで同じIPアドレスを設定するために、既存のリモートクラスタのIPアドレスをオフラインにすることが必要になる場合があります。リモートTraffic Managerをマスタクラスタに追加するときは、まずそのTraffic Managerをリモートクラスタから削除する必要があります。これにより、リモートIPグループからも削除されます。

8. 各クラスタが複数のTraffic Managerマシンで構成されている場合、各クラスタから1つのTraffic Managerを切り離してマスタクラスタに組み込みます。このとき、該当する定義済みのリモートロケーションに存在するものとしてそのTraffic Managerを指定します。システム固有の設定は、その設定の対象となるマシンの管理UIから直接適用する必要があることに注意してください。
9. このTraffic Managerを新たに作成したトラフィックIPグループに追加できます。その後、サービスを再開します。
10. 次に、残りの各Traffic Managerをリモートクラスタから切り離してマスタクラスタに組み込むことができます。このとき、該当するロケーションに存在するものとしてそのTraffic Managerを指定し、適切なトラフィックIPグループに追加します。各Traffic Managerは、完全にマージされた設定を継承しますが、指定されたロケーションのサービスにのみレスポンスを返します。

設定

ロケーションの設定

ロケーションは**Catalog** (カタログ) > **Locations** (ロケーション)ページに表示され、このページで変更を行います。また、**System** (システム) > **Traffic Managers**ページの**Manage Locations** (ロケーションの管理)リンクをクリックしても表示されます。このページには、定義されているロケーションのリストが表示され、ロケーションをリストに追加またはリストから削除することができます。



グローバル負荷分散機能を持つ製品では、「GLB Locations」という追加のセクションが表示されます。詳細については、[Traffic ManagerのDNSサーバ](#)を参照してください。

各ロケーションは、単純に名前で定義されます。ロケーション名または関連する**Edit** (編集)リンクをクリックすると、ロケーションの設定を変更したり、ロケーションを完全に削除したりできます。

新しいロケーションを追加するには、メインリストの下に「Add new Location」(新規ロケーションの追加)セクションを使用します。新しいロケーションを作成するときは、既存のロケーションをテンプレートとして使用します(既存のロケーションがない場合を除く)。

注記

この既存ロケーションに「基づく」メカニズムにより、テンプレートとなるロケーションの全てのロケーション固有の設定が新しいロケーションにコピーされます。

ロケーションを削除するには、関連する**Edit** (編集)ページの下部にあるセクションを使用します。ただし、1台以上のTraffic Managerで使用中のロケーションを名前変更したり削除したりすることはできません。このような操作を行う前に、まずTraffic Managerとこのロケーションとの関連付けを解除する必要があります。

Traffic Managerのロケーションの設定

System (システム) > **Traffic Managers**ページでは、クラスタ内の各Traffic Managerに対して、新たに以下の2つの設定を行えます。

- **Location** (ロケーション): このTraffic Managerのロケーション設定をドロップダウンリストから選択します。
- **External IP** (外部IP): Traffic Managerのホスト名がグローバルに名前解決出来ない場合、クラスタメンバ間の通信に使用する、外部から利用可能なIPアドレスを指定する必要があります。

これらの設定を変更する場合は、まず**Update** (更新)ボタンの横にあるチェックボックスをクリックして確認する必要があります。

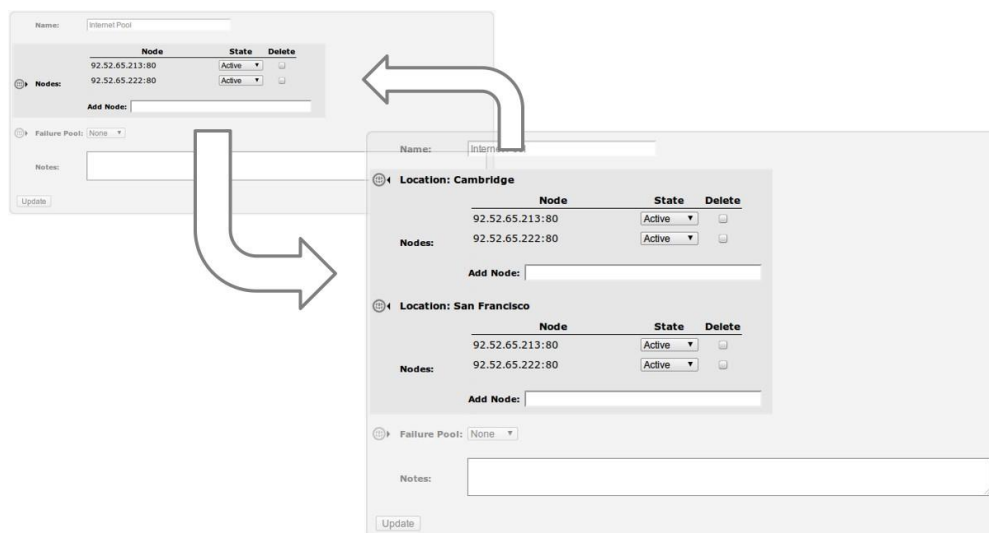
ロケーション固有の設定

マルチサイトモードでのTraffic Managerによる設定の扱いが大きく異なることは、各設定の横に新しいアイコンが追加されることから分かります。

Locationアイコン



このアイコンをクリックすると、設定が、ロケーション設定ごとの複数のインスタンスに分割されます。このアイコンを再度クリックすると、全てのロケーションで使用する単一のインスタンスに戻ります。



このメカニズムにより、全てのクラスタメンバに適用されるグローバル設定を維持しながら、個々の設定をロケーションごとに指定することができます。

ホームページの変更点

マルチサイトクラスタ機能が有効化されている場合、通常とは異なるUIホームページが表示されます。新しい方式は、Traffic Managerの想定運用環境に適したものとなっており、以下に示すいくつかのセクションが表示されます。

- **Locations** (ロケーション): ここには、Traffic Managerのロケーション設定が表示されます。定義されていても使用されていないロケーションは表示されません。ロケーション名をクリックすると、そのロケーションの**Catalogs** (カタログ) > **Locations** (ロケーション) > **Edit** (編集)ページが表示されます。
- **Services** (サービス): ここには、enabled (有効化)のマークが付けられ、このTraffic Managerによって管理されているサービスのリストが表示されます。サービスは、仮想サーバ、ルール、プール、およびその他の関連オブジェクトを組み合わせたものとして考えることができます。サービス名をクリックすると、そのサービスの**Services** (サービス) > **Virtual Servers** (仮想サーバ) > **Edit** (編集)ページが表示されます。
- **イベントログ**: これは、イベントログの簡単な概要です。イベントログは全てのTraffic Managerから書き込まれ、各エントリの横には対象のマシンの名前が示されます。ログ全体を表示するには、**Examine Logs** (ログ確認)リンクをクリックします。

それぞれのロケーションおよびサービスは、以下の適切なステータスインジケータとともに表示されます。



正常動作を表します



警告を表します



重大なエラーを表します

警告は重大な問題ではありません。すなわち、ロケーションまたはサービスは、性能が低下した状態でまだ動作しています。エラーには、ロケーションまたはサービスの動作を妨げている問題を解決するためのアクションが必要です。これらの状態は、ステータスアプレットに反映されます。警告またはエラーの場合、ロケーション/サービス名をクリックすると、**Diagnose** (診断)ページに移動し、ここで問題の詳細な説明を確認することができます。

ワールドマップ



Activity (アクティビティ) > **Map** (マップ)をクリックすると、ロケーションとサービスが分かりやすく視覚化されたワールドマップが表示されます。このマップは、**Services** (サービス) > **Virtual Servers** (仮想サーバ) > **Edit** (編集)ページからのワールドマップにトラフィックを表示リンクでも表示できます。ここには、システムに定義されているロケーションの世界規模の概要が示されます。

マップはインタラクティブに設計されており、マップ上を移動したり、インフラストラクチャに必要な詳細レベルにズームインしたりするためのツールが用意されています。マウスの左ボタンを押しながらマップを所要の目的地にドラッグすることにより、任意の地理的位置を中央に表示することができます。下部にあるドロップダウンコントロールを使用すると、特定の仮想サーバのトラフィックを表示できます。デフォルトでは、全ての仮想サーバのトラフィックが表示されます。

色分けされた標的マークはロケーションを表します。また、マップ上に表示される小さいドットは特定のサービスに対するリクエストを送信したユーザを表し、リクエストしたサービスがホストされているロケーション別に色分けされて表示されます。役立つ情報を表示するには、Connection Archivingを有効化する必要があります(高度なコネクション解析を表示する場合にも必要です)。Connection Archivingが無効化されている場合、下部にあるconnection archiving is disabled (Connection Archivingが無効)リンクをクリックすると、有効化できます。このリンクをクリックすると、**System** (システム) > **Global Settings** (グローバル設定)ページに移動し、このページでrecent_conns設定に適切な値(500など)を指定する必要があります(デフォルト値0の場合、コネクションアーカイブが無効化される)。



ユーザの位置に対応するアクティビティをマップに表示できるようにするために、Traffic ManagerではGeoIPデータベースへのアクセスが要求されます。詳細については、[GeoIPデータベースの使用](#)を参照してください。

トラフィックの可視化

グローバル負荷分散と同様に、システムでマルチクラスタ管理機能をアクティブにすると、Traffic Managerの既存の診断および可視化ツールで追加のオプションが使用可能になります。詳細については、[トラフィックの可視化](#)を参照してください。

Traffic ManagerのDNSサーバ

Traffic Managerには、権威ドメインネームシステム(DNS)サーバ機能が内蔵されています。この機能により、Traffic Managerはゾーンファイルをホストし、DNSクエリに応答することができます。

グローバル負荷分散(GLB)のデプロイの場合、内部のTraffic ManagerのDNSサーバを実装するか、トラフィックを外部DNSサーバノードのプールに分散するかを選択できます。

DNSの概要

はじめに

DNSは、インターネットの電話帳と考えることができます。ドメイン名に関する様々な情報の断片を検索するために使用します。例えば、Webブラウザは、WebサーバにWebサイトをリクエストするために、どのIPアドレスに接続すればよいかを知る必要があります。同様に、電子メールを送信するときは、電子メールサーバは、メッセージを受信者に配信する役割を担っているメールサーバのドメイン名とIPアドレスの両方を知らなければなりません。この情報を見つける手段を提供するのがDNSです。

DNSのレイアウト

DNSのレイアウトはトップダウンツリー構造に似ています。この構造のルートは、ルートネームサーバのグループから成ります。ルートの下にはトップレベルドメインサーバがあり、「com」(「www.example.com」など)や「net」はどちらもトップレベルドメインの例です。トップレベルより下位では、DNSに「ネームサーバ」以外の明確な分類はありません。

権威の委任

DNSにおける「権威」は、特定ドメインに対するリクエストを処理するネームサーバの責任を表します。ネームサーバが、example.comの下位にある全てに対するDNSリクエストの処理に権威を持っている場合、このネームサーバには、上位のドメインサーバからこのような処理を行う権威が委任されています。Traffic Managerの内蔵DNS機能は、権威DNSサーバの一例です。

名前解決

DNSを使用して「情報の断片を検索する」プロセスを「名前解決」と言います。例えば、Webブラウザが特定WebサイトのIPアドレスを見つける必要がある場合、ドメイン名を名前解決してIPアドレスを取得しようとしします。電子メールサーバが特定ドメインへのメッセージ配信を担っているメールサーバを知る必要がある場合、「メール交換(MX)名」とそのIPアドレスを名前解決しようとしします。

ネットワーク設定の過程で、通常、コンピュータは少なくとも1つの「ネームサーバ」を認識し、DNSリクエストに対するレスポンスはこのネームサーバによって与えられます。コンピュータがDNSリクエストを別のネームサーバに送信するのは、特殊な状況¹のみです。

リソースレコード

ドメイン名に関する各情報は、リソースレコードに保持されます。通常、リソースレコードはタイプによって識別されます。例えば、Webブラウザがwww.example.comのIPアドレスを知ろうとしている場合、www.example.comに一致する「Aレコード」を問い合わせるクエリを送信します。メールクライアントがjohn.smith@example.com宛でのメッセージをどこに送信すればよいかを知る必要がある場合は、example.comに一致する「MXレコード」を問い合わせるクエリを送信します。一般的に使用およびサポートされているリソースレコードタイプには、以下のものがあります。

- **SOA (権威の開始):** 特定のドメインに関する役立つ情報を与えます。
- **NS (ネームサーバ):** 特定ドメインに対して権威を持つメールサーバのドメイン名を示します。このリソースは、権威がどのように上位ドメインから下位ドメインへと委任されるかを決定します。
- **MX (メール交換):** 特定ドメイン宛でのメールを受け入れるメールサーバのドメイン名を示します。
- **A (アドレス):** ドメイン名のIPv4アドレスが保持されます。
- **AAAA (アドレス):** ドメイン名のIPv6アドレスが保持されます。
- **CNAME (カノニカル名):** ドメインネームのエイリアスが保持されます。
- **PTR (ドメインポインタ):** IPからDNS名への逆引きに使用されます。
- **SRV (サービスレコード):** 指定されたサービスに使用されるサーバのホスト名とポート番号を定義します。
- **TXT (テキストメッセージ):** 人間が読み取れる任意のテキストメッセージが保持され、様々な目的で使用されます。

¹ このような状況として、テストを実施するDNS管理者が意図的に行う場合などがあります。

ゾーンファイル

通常、リソースレコードは、権威DNSサーバ上のテキストファイルに保存されます。ゾーンファイルには、ドメインに固有の一連のレコードが保持され、少なくとも1つのSOAレコードとNSレコードが含まれている必要があります。

Traffic Managerは、以下のフォーマット例に則ったゾーンファイルをサポートしています。

```
$TTL 604800 ; This TTL declaration applies to all
                ; records defined below that do not define
                ; their own TTL

; In this file, @ means the "current origin domain"

@      IN      SOA      ns.example.com. root.example.com. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800     ; Negative Cache TTL
                                );

@      NS      ns
ns     A       192.168.0.1
@      NS      ns.example.com.
ns     A       192.168.0.254

@      IN      MX       1 host
@      IN      MX       2 mail2.example.net. ; note this is
                                ; in another
                                ; zone/domain

host   IN      A       192.168.0.2
        AAAA    2001:db8:85a3:0:0:8a2e:370:7334

$TTL 1w2d3h4m5s ; The TTL defined here overwrites the
                ; setting above and applies to all the
                ; records below it that do not define their
```

```
        ; own TTLS.

foo.bar      A    192.168.0.3 ; An example of a
                ; relative address:
                ; foo.bar.example.com.
foo.bar.example.com. A    192.168.0.4 ; An example of an
                ; absolute address

alias        CNAME host

_sip._udp    SRV 0 5 5060 sipserver
sipserver 3600 A    192.168.0.5 ; This record has
                ; its own TTL (3600)

text1        TXT "This is a TXT record"
*.wildcards  TXT "This is a wildcard TXT record"
```

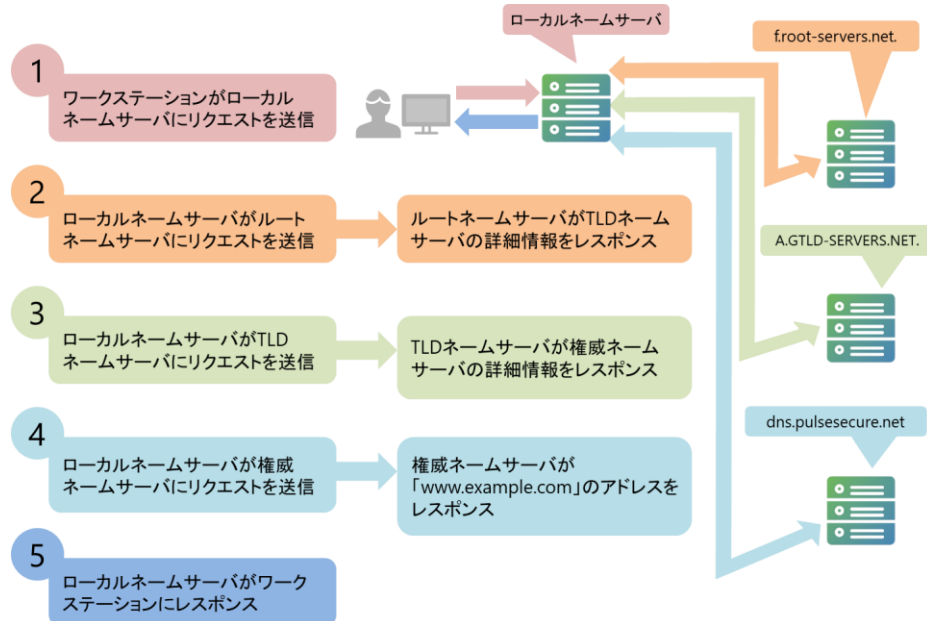
Traffic Managerは、トップダウンのアプローチによってゾーンファイルを解析し、インデントが生じた先行ラインから継承します。

当初のデプロイからゾーンファイルを変更または更新した場合は、DNSサービスが変化を認識するように、「シリアル」カウンタを増加させる必要があります。

通常、ゾーンファイルには少なくとも1つのTTL(存続時間)定義が、具体的にはファイルの最上部に保持されています。Traffic Managerは、この値を使用して、DNS関連情報が更新されるまでクライアントを待機させる時間の長さを決定します。ゾーンファイルのTTLの値には、以下のルールが適用され、監視されます。

- ゾーンファイルにTTLの値を定義しない場合、Traffic Managerは代わりにSOA定義からネガティブキャッシュTTLの値を使用します。ネガティブキャッシュTTLは、通常、「否定的なレスポンス」のTTLとして使用されます。この否定的なレスポンスとは、NXDOMAIN(ドメイン名が不明)およびNODATA(ドメイン名は判明したが、特定タイプのレコードが存在しない)です。
- 同じドメイン名を持つレコードには、全て同じ値のTTLが定義されています。
- TTLは、秒数を表す一つの数字で指定することも、1w2d3h4m5s (この場合、1週間2日3時間4分5秒を表す)の代替フォーマットで指定することもできます。このフォーマットのアルファベット要素には大文字と小文字の区別はなく、大文字と小文字を混合させて指定することもできます (例: 3W1h5M4s)。

名前解決プロセス



ここからは、ユーザのクライアントワークステーションがWebサイトのURLを正しいIPアドレスに名前解決する方法について、手順を追って説明します。



Traffic Managerの内蔵DNSサーバ機能を使用している場合、このシナリオの「dns.example.com」が権威DNSサーバとしてのTraffic Managerに相当することがあります。

1. ユーザがWebブラウザを起動し、アドレスバーにwww.example.comと入力した場合、ワークステーションはHTTPリクエストをWebサーバに送信する必要があります。ただし、その前に、Webサーバのドメイン名を名前解決しなければなりません。そこで、ワークステーションはIPアドレスを特定するために、DNSリクエストをローカルネームサーバに送信します。
2. ローカルネームサーバはルートヒントゾーンファイルを参照します。ルートヒントゾーンファイルには、ルートネームサーバのドメイン名とIPアドレスのリストが保持されています。次に、ローカルネームサーバはwww.example.comに対するDNSリクエストを、ルートヒントゾーンファイルに登録されているいずれかのルートネームサーバに送信します。「f.root-servers.net.」は、DNSリクエストと、このルートネームサーバが権威を持っているゾーンのリストを比較します。この場合のルートネームサーバは「.」ドメインに権威を持っています。「.」のゾーンファイル内には、全てのトップレベルドメインの委任レコードがあります。www.example.comのトップレベルドメインは「.com.」であるため、「f.root-servers.net.」は、「.com.」に権威を持っているネームサーバのドメイン名とIPアドレスを記述したDNSレスポンスをローカルネームサーバに返します。

3. A.GTLD-SERVERS.NET¹は、(いくつかあるトップレベルドメインサーバのうち)「.com.」ドメインに権威を持つトップレベルドメインサーバとして登録されているため、ローカルネームサーバは同じリクエストをこのトップレベルドメインサーバに送信します。これに対し、A.GTLD-SERVERS.NETは、「example.com.」に権威を持つネームサーバのリストをレスポンスとして返します。
4. 「dns.example.com.」は「example.com.」の権威ネームサーバであるため、ローカルネームサーバはこの権威ネームサーバにリクエストを送信します。「dns.example.com.」のゾーンファイルには、「example.com.」の全てのリソースレコードが保持されているため、この権威ネームサーバから(1つ以上の「A」レコードの形で) IPアドレスが与えられます。
5. これで、ローカルネームサーバはwww.example.comの(1つまたは複数の)IPアドレスを取得できたため、DNSリクエストの最初の送信元であるワークステーションにこの情報を渡します。再帰的検索を行ったマシンはローカルネームサーバです。

クライアントが行うこのタイプのクエリは、ローカルネームサーバがDNSツリーを最上位から下位へと検索することから「再帰的」クエリと呼ばれます。「非再帰的」クエリを送信することも可能です。この場合、問い合わせられたドメイン名に権威を持っているネームサーバのみがそのドメイン名のIPアドレスを返し、クエリを受信したそれ以外のネームサーバは、そのクエリに対してレスポンスを返すことを拒否します。

通常、ローカルネームサーバは、エンドユーザに最終的なDNSの結果を返すだけでなく、中間の検索結果のキャッシングも行います。結果的に、全てのリクエストが前述のフルセットのDNSリクエストを必要とするわけではありません。このため、ローカルネームサーバは、「再帰的ネームサーバ」または「キャッシングネームサーバ」と呼ばれることがあります。

サポートされるDNS機能

Traffic ManagerのDNSサーバは、RFC 1034およびRFC 1035で定義されているDNSの技術仕様の大部分に適合するように設計されています。このセクションでは、実装されている機能と除外事項を説明します。

RFC 1034準拠の実装機能

Traffic ManagerのDNSサーバは以下の機能を実行します。

- 権威サーバの動作
- ドメイン名空間のデータベース
- 肯定的および否定的応答 (NODATA、NXDOMAIN、SERVFAIL、FORMAT ERROR、REFUSED、NOT_IMPLEMENTED)

¹ 「GTLD」は「Global Top Level Domain (グローバルトップレベルドメイン)」の頭字語です。

- 委任(referralとも言う)
- DNSデータベースのクラスIN
- レコードタイプANY
- ワイルドカードDNSレコード(RFC 4592も参照)

RFC 1035準拠の実装機能

Traffic ManagerのDNSサーバは以下の機能を実行します。

- サポートされているレコードタイプ
 - A
 - CNAME
 - MX
 - NS
 - PTR
 - SOA
 - TXT
- その他のサポートされているレコードタイプ
 - AAAA (RFC 3596も参照)
 - SRV (RFC 2782も参照)
- サポートされているDNSSECレコードタイプ(RFC 4034も参照)
 - DNSKEY
 - RRSIG
 - DS
 - NSEC
 - NSEC3
 - NSEC3PARAM

- DNSの応答にはメッセージ圧縮が使用されます(RFC 1035のセクション4.1.4も参照)。
- UDPおよびTCPは、IPv4およびIPv6による転送プロトコルとして完全サポートされています。
- RFC 1035のセクション5で定義されているとおり、ゾーンテキストファイル(「マスタファイル」とも言う)がサポートされています。Traffic Managerは、ゾーンファイルの以下の要素を処理することができます。
 - このセクションで説明されているサポート対象タイプの、リソースレコードの定義
 - \$TTL指示文
 - \$ORIGIN指示文
 - 丸括弧の使用および複数行レコードの定義
 - 「@」記号
 - セミコロン(;)に続くコメント
- RFC 1035のセクション3.5のとおり、in-addr.arpaドメインを使用した逆引きがサポートされています。

RFC 1034に関する除外事項

以下の機能は実装またはサポートされていません。

- CS、CH、およびHSのリソースレコードクラス
- 逆クエリ(逆クエリはRFC 3425で廃止)
- ゾーン転送
- ステータスクエリ(実験)
- 完了クエリ(廃止)

RFC 1035に関する除外事項

以下の機能は実装またはサポートされていません。

- \$INCLUDE指示文
- リテラル文字または制御文字を定義するためのバックスラッシュ(\)演算子の使用。例えば、「\。」によるラベル内へのドット文字配置
- それぞれのDが、DDDと記述される10進数に相当するオクテット数字である\DDD。Traffic Managerは、結果として生じるオクテットをテキストと見なし、特別な意味ではチェックしません

その他の実装機能

Traffic ManagerのDNSサーバは以下の機能を実行します。

- 大文字と小文字の違いは完全サポートされています。
- RFC 2308の「Negative Caching of DNS Queries (DNSクエリのネガティブキャッシング)」は完全実装されています。
- EDNS(0) はRFC 6891の「Extension Mechanisms for DNS (DNS用拡張機構)」に準じて実装されています。Traffic Managerは、EDNSを使用して、512バイトを超えるUDPレスポンスサイズを処理します。さらにTraffic Managerは、EDNSの「DO」フラグを使用して、受信するDNSクエスチョンがDNSSECサポートを示しているかを、RFC 3225に準じて判断します。その他のEDNS機能はサポートされていません。
- DNSSECプロトコルは、以下のRFCでの定義に準じてサポートされています。
 - RFC 4033、「DNS Security Introduction and Requirements (DNSセキュリティ拡張の紹介とその要件)」
 - RFC 4034、「Resource Records for the DNS Security Extensions (DNSセキュリティ拡張で使用するリソースレコード)」
 - RFC 4035、「Protocol Modifications for the DNS Security Extensions (DNSセキュリティ拡張のためのプロトコル修正)」
 - RFC 5155、「DNS Security (DNSSEC) Hashed Authenticated Denial of Existence (ハッシュ化された認証による存在否定 (NSEC3))」
- DNSSECのサポートについては、以下の点に注意してください。
 - Traffic Managerは、署名済みのDNSSECゾーンファイルを解析することができます。
 - Traffic Managerは、規格に準拠した方法により、DNSSECの署名済みレコードを使用して応答することができます。
 - Traffic Managerは、ゾーンが仮想サーバに関連付けられている場合に、ゾーンファイルでDNSSEC署名の有効期限をチェックし、その後も1時間ごとに定期的にチェックします。署名の有効期限が切れているか、7日以内に有効期限が切れる場合は、Traffic Managerはアラートシステムを通じてイベントが発生します(第21章、「イベント処理およびアラート」を参照してください)。
- RFC8659、「DNS Certification Authority Authorization (CAA) Resource Record」がサポートされています。

その他の除外機能

以下の機能は実装またはサポートされていません。

- RFC 2317に準じたクラスレスな委任はサポートされていません。
- RFC 3597の「Handling of Unknown DNS Resource Record (RR) Types (未知のDNSリソースレコード (RR)タイプの取り扱い)」はサポートされていません。
- DNSSECのサポートについては、以下の点に注意してください。
 - Traffic Managerには、ゾーンファイルに署名するツールは用意されていません。
 - Traffic Managerには、暗号化キーのライフサイクルを管理するツールは用意されていません。

DNSサーバの設定

このセクションでは、Traffic Managerを権威DNSサーバとして設定する方法について説明します。

設定概要

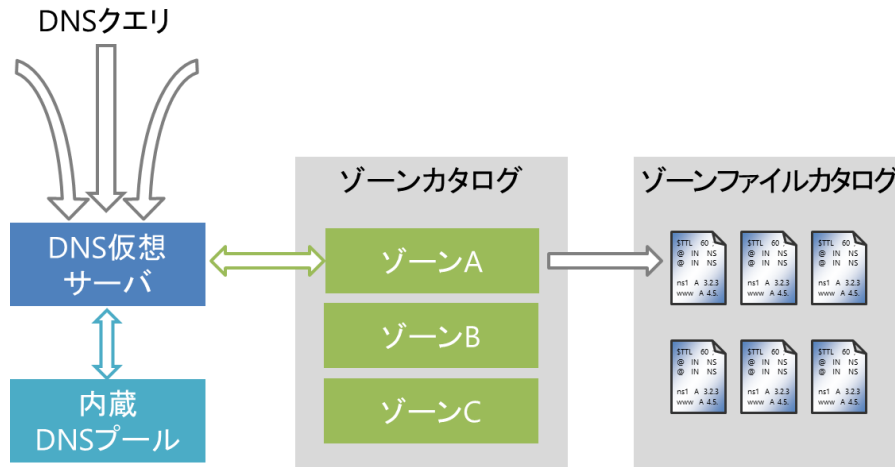
以下のタスクを実行するには、管理UI、REST API、SOAP API、またはコマンドラインインターフェースを使用します。それぞれについては、後で詳しく説明します。

- DNSゾーンファイルを「ゾーンファイル」カタログにアップロードする
- 「ゾーン」設定を作成し、「オリジンドメイン」を定義するとともに、アップロードしたDNSゾーンファイルをカプセル化する
- DNSを設定した仮想サーバを作成し、そのサーバに1つ以上のゾーンを割り当て、受信するDNSリクエストに対処する

ゾーンファイルは再利用したり、複数のゾーン設定に関連付けることができます。そうすることによって、複数のゾーンが異なるオリジンドメイン下の同じホストにアダプタイズできるようになります。さらにDNS仮想サーバは、あらゆる数のゾーンを利用して、多くのドメインをカバーする権威DNSサービスを提供することもできます。



1つのゾーンファイルを複数のゾーン設定に関連付ける場合は、ゾーンファイル内部のドメインオリジンを(\$ORIGIN宣言を使用して)再定義しないことをお奨めします。



この仕組みを利用して、Traffic Managerは、外部DNSサーバノードのプールにリクエストを渡すのではなく、内蔵のDNS機能を使用してDNSリクエストに応答します。

Traffic ManagerへのDNSゾーンファイルのアップロード

ゾーンファイルは、Traffic Managerがドメインに対するDNSクエリを受信したときに、Traffic Managerにどのような応答を返すかを指示するリソースコードを保持したテキストファイルです。各ゾーンファイルは、RFC 1034およびRFC 1035で規定されたフォーマットに準じている必要があります。

Traffic Managerは、DNSゾーンファイルカタログにゾーンファイルを保存します。新しいゾーンファイルをアップロードするには、**Catalogs** (カタログ) > **DNS Server** (DNSサーバ) > **Zone Files Catalog** (ゾーンファイルカタログ)を選択します。



ゾーンファイルがゾーン設定と関連付けられていない限り、Traffic Managerはゾーンファイルの内容を検証しません。検出された異常は、全てイベントログに追加されます。

Traffic Managerのゾーンの設定

Traffic Managerが、アップロードされたゾーンファイルに直接アクセスしたり、解析したりすることはありません。ゾーンファイルを使用するには、まずカタログ内のゾーンファイルを参照するゾーン設定を作成します。

各ゾーンには、関連付けられたゾーンファイル内のリソースレコードと併せてTraffic Managerが使用する、ドメイン「オリジン」も保持されています。つまり、ゾーンファイルにリストされた全てのドメインはこのオリジンに分岐し、<domain>.<origin>という形式を取ります。

例えば、オリジンが「example.com」の場合、ゾーンファイルで「www」および「www.support」のドメインを指定し、「www.example.com」および「www.support.example.com」に対するドメイン検索を作成することができます。

ゾーン設定を作成するには、**Catalogs** (カタログ) > **DNS Server** (DNSサーバ) > **Zones Catalog** (ゾーンカタログ)を選択します。ゾーンに対する名前とドメインオリジンを入力し、アップロードしておいた使用するゾーンファイルを選択します。**Create Zone** (ゾーンの作成)をクリックし、ゾーン設定を保存します。

ゾーンファイルを関連付けてゾーン設定を作成するときに、ゾーンファイルの内容に構文上の問題があると、Traffic Managerはイベントログにレポートを残します。

DNS仮想サーバの設定

一般に、仮想サーバは、受信するDNSリクエストを処理し、そのリクエストをバックエンドDNSサーバノードのプールに渡すために作成します。その代わりに、内部DNSサーバ機能を使用すると、DNS仮想サーバを設定して内蔵DNSプール「builtin_dns」を使用したり、DNS検索に使用する所要のゾーン設定をリンクさせたりすることができます。



十分なサービス範囲を確保するためには、DNS (UDP)およびDNS (TCP)の両方の仮想サーバをデプロイすることをお奨めします。

DNS仮想サーバの作成

1. **Services** (サービス) > **Virtual Servers** (仮想サーバ)を選択します。
2. 「Create a new Virtual Server (新しい仮想サーバの作成)」セクションで、以下を入力します。
 - **Virtual Server Name** (仮想サーバ名): 作成する仮想サーバの識別名です。
 - **Protocol** (プロトコル): ドロップダウンリストから「DNS (UDP)」または「DNS (TCP)」を選択します。
 - **Port** (ポート): この仮想サーバが監視する受信ポート番号を選択します (通常は「53」)。
 - **Default Traffic Pool** (デフォルトトラフィックプール): Traffic Managerに内蔵されたDNS機能を使用するよう指示するには、「builtin_dns」を選択します。
3. **Create Virtual Server** (仮想サーバの作成)をクリックし、新しい仮想サーバを作成します。
4. Virtual Server edit (仮想サーバ編集)ページで**DNS Server** (DNSサーバ)をクリックし、この仮想サーバによるDNS検索に使用するゾーン設定を1つ以上選択します。
5. 仮想サーバを有効化するには、Virtual Server edit (仮想サーバ編集)ページで**Enabled** (有効化)を「Yes (はい)」に設定するか、Traffic Managerのホームページの仮想サーバ名の隣にあるplay(作動)アイコンをクリックします。



デフォルトのトラフィックプール「builtin_dns」は、DNS (TCP)またはDNS (UDP)の内部プロトコルを使用する仮想サーバでのみ利用でき、通常のプールリストには記載されないか、またはリストから選択することはできません。

Virtual Server (仮想サーバ) > **Edit** (編集) > **DNS Server** (DNSサーバ)ページでは、以下の設定も変更できます。

- **dns!edns_udpsize**: EDNSのUDプレスポンスでアドバタイズされるメッセージのサイズです。
- **dns!max_udpsize**: UDプレスポンスに許容される最大サイズです。
- **dns!verbose**: デバッグの効果を高めるには、この設定を有効化し、Traffic Managerに対してイベントログにより詳細なDNS情報を含めるように指示します。製品を通常どおり使用する場合、この設定は無効化しておきます。
- **dns!rrset_order**: DNSレコードのレスポンス順序です。ゾーンファイル内で定義された順序で、同じ名前とタイプのDNSレコードを返すようにTraffic Managerに指示するには「Fixed (固定)」に設定します。ローテーションするラウンドロビンシステムを使用するようにTraffic Managerに指示するには、「Cyclic (循環)」に設定します。一連のレスポンスに対して、同じ名前とタイプのDNSレコードは、循環方式で返されます。(レスポンスのたびに、各レコードが1つ進みます)。この方法によって、効率的な負荷分散が可能になります。



場合によっては、キャッシングリゾルバによって、権威サーバから受信するレスポンスのDNSレコードの順序が不変となることがあります。

グローバル負荷分散

この章では、Traffic Managerのグローバル負荷分散(GLB)について説明します。



Traffic Manager製品によっては、この機能を利用できない場合があります。サポート窓口にお問い合わせください。

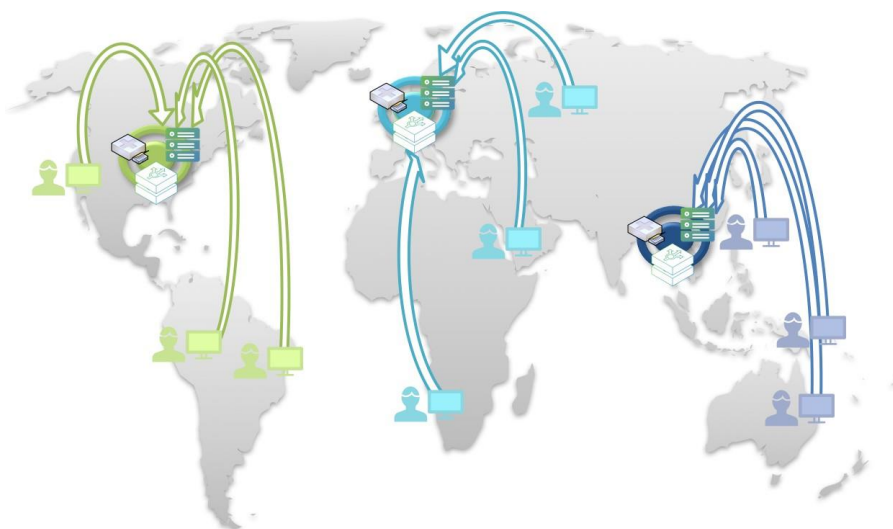
概要と前提条件

GLBは、ビジネス継続性と世界各地のロケーションにわたる性能向上を実現するための機能です。この機能は、グローバルサーバ負荷分散(GSLB)として知られている技術をベースにしています。Traffic Managerは地理的負荷分散およびフェールオーバー機能を使用し、可用性、性能、およびユーザ定義のルールに基づいてこれらのロケーションへのトラフィックの分配を可能にします。

本章では、Traffic ManagerのデプロイでGLBを計画および設定する方法について説明します。ここで説明する設定を行うためには、ネットワーキングやDNS管理を含め、標準的なシステム管理タスクを十分に理解している必要があります。DNSの基礎知識に関しては、[DNSの概要](#)を参照してください。

GSLBの詳細な原理については、[グローバルサーバ負荷分散について](#)を参照してください。www.ivanti.com およびPulse CommunityのWebサイト(<https://community.pulsesecure.net>)で公開されている基礎知識のホワイトペーパーおよびドキュメントも参照することをお奨めします。

グローバルサーバ負荷分散について



グローバルサーバ負荷分散(GSLB)は、サービスが複数のロケーションでホストされている場合に、クライアントの接続先となる地理的ロケーションをどのように決定するかを管理する手法です。

- 「アクティブ-パッシブ」(またはプライマリ-バックアップ)構成では、サービスごとに1つのロケーションがアクティブなロケーションとして指定されます。他のロケーションは、そのサービスについて待機状態になります。アクティブなロケーションが使用不可になった場合、いずれかのパッシブロケーションがアクティブになり、全てのクライアントのトラフィックがこのロケーションに送信されます。
- 「アクティブ-アクティブ」(またはマルチプライマリ)構成では、全てのロケーションが使用され、ロケーションの性能と近接性に基づいてクライアントのトラフィックがロケーションに負荷分散されます。

GSLBシステムの主な目的は、ビジネス継続性、すなわち1つ以上のサービスロケーションが使用不可になっても、常にサービスを利用できる状態に保つことです。

GSLBの第2の目的は、より良いカスタマ経験を提供すること、すなわち各ユーザのトラフィックを、いくつかあるロケーションのうち最適なロケーションに負荷分散することです。ロケーションの選択は、ロケーションの性能と近接性に基づいて行うことができるため、クライアントのトラフィックを最も近くて最も性能の高いロケーションに送信することができます。この機能により、クライアントには可能な限り最高のサービスレベルが提供されます。

Traffic Manager内でのGSLB

Traffic Managerは、グローバル負荷分散(GLB)機能のような、GSLB技術を実装しています。GLB機能により、Traffic Managerは、エンドユーザが行うDNS検索にロケーション認識を加えることができ、DNS検索の結果がソースに依存するようになります。

GLBサービスは、DNSプロトコルを使用している任意の仮想サーバに対して追加することができます。この仮想サーバにDNSリクエストが届くと、Traffic Managerは、関連付けられたプールにそのリクエストを転送します。このプールは、内蔵された内部DNSサーバプール([Traffic ManagerのDNSサーバ](#)を参照)であることもあれば、外部DNSサーバのプールであることもあります。いずれのオプションを使用する場合も、複数の「A」レコードを返すためのプールバックエンドを、名前解決されるドメイン名に対する全てのIPアドレスを含め、全てのロケーションにわたって設定する必要があります。

名前解決されたドメイン名が、仮想サーバに設定されているGLBサービスのドメイン名と一致する場合、Traffic Managerは、GLBサービスの設定に基づいてリストをフィルタしてから、エンドユーザのロケーションに適した(1つまたは複数の)IPアドレスのみをエンドユーザに返します。

必須ではありませんが、ローカルの性能と可用性をモニタリングするためにTraffic Managerを各GLBロケーションにデプロイすることができます。また、他の場所にTraffic Managerをデプロイしなければならない場合でも、全てのモニタリングをリモートから行うことができます。Traffic Managerは、実際にデプロイされている場所とは関係なくピアとの間でモニタリング情報を交換するため、全てのトラフィックが連携して処理を調整することができます。



GLBサービスは「A」レコードのみをフィルタし、「AAAA」レコードはフィルタしません。DNSレスポンスに「AAAA」レコードが保持されている場合、Traffic Managerはそのレスポンスをそのまま通過させます。

デプロイ計画

Traffic Managerの位置決め

Traffic Managerは、パブリックIPv4アドレス空間の地理的ロケーションに関するデータベースを保持しており、権威DNSサーバ、または別のバックエンドDNSサービス用のプロキシのいずれかとして機能します。Traffic Managerは、現在のDNSインフラストラクチャと並行して、またはその前に、またはそれに代えてデプロイすることができます。Traffic Managerは、クライアントからDNSリクエストを受け取ると、Traffic Managerが管理している内部または外部DNSサービスのいずれかからレスポンスを特定します。その後、Traffic Managerは多数の設定メトリクスに基づいて(地理的ロケーションも含め)レスポンスを変更してから、クライアントにそのレスポンスを返します。

Traffic Managerの内蔵DNS機能の詳細については、[Traffic ManagerのDNSサーバ](#)を参照してください。

デプロイ手法

DNSドメインは、親ドメインがネームサーバによって権威を持っていると指定された、一連のネームサーバを必要とします。グローバルトップレベルドメインなど、親ドメインがパブリックドメインの場合は、DNSレジストラを通じてご使用のドメイン用に権威ネームサーバを変更します。

GLBをホスト名で使用するには、以下のいずれかの方法でTraffic Managerをデプロイすることができます。

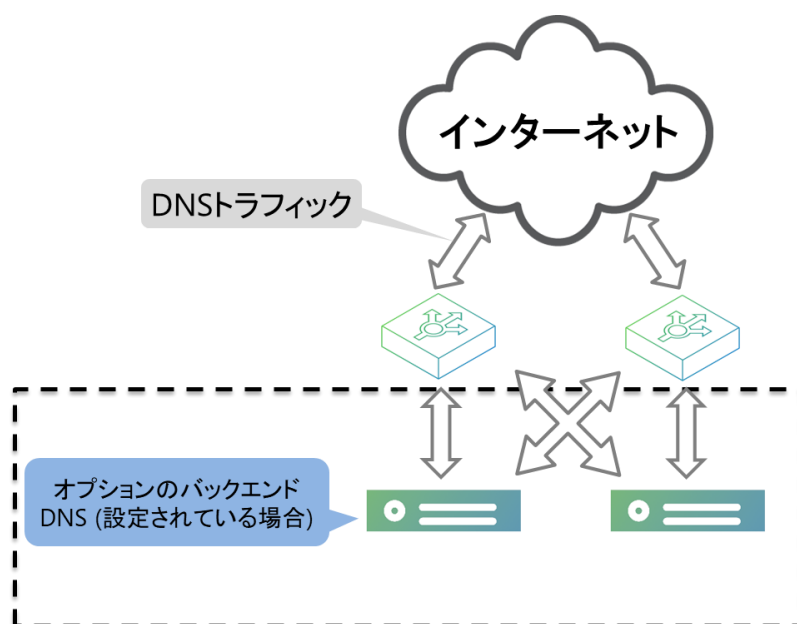
- 「インライン」デプロイの場合、Traffic Managerは、そのホスト名を保持するドメインに対する権威DNSサーバになります。親ドメインの委任レコードを変更することによって、そのドメインに対する全てのDNSトラフィックが、ご使用のTraffic Managerに送信されるようになります。Traffic Managerは、内蔵のDNSサーバ機能を使用してリクエストを処理するか、ご使用の既存DNSサーバにリクエストを転送してから、GLBが設定されたホスト名でレスポンスを操作します。
- 「並列」デプロイでは、Traffic Managerにホスト名を保持するドメインに対する権威はなく、その代わりに、Traffic Managerが権威を持つ別のDNSドメインで並列のホスト名を作成します。既存ドメインでは、他のドメインの並列のホスト名に対するCNAMEを作成し、その並列のホスト名でGLBサービスを設定します。

ご使用の(権威のある)ネームサーバがホスト名に対するDNSリクエストを受信すると、ネームサーバはCNAMEを使ってレスポンスを返し、それによりクライアントが、並列のホスト名に対するDNSリクエストをTraffic Managerから行います。インラインデプロイと同様に、Traffic Managerは内蔵のDNSサーバを使用してリクエストを処理するか、別の権威DNSサーバにリクエストを転送します。その後、レスポンスはGLBサービスの設定に従って操作されます。

並列デプロイは、設定が複雑になりますが、多数のドメインを管理する場合や頻繁に変更を行うことが予測される場合は、このデプロイの方が適しています。

インラインデプロイ

インラインデプロイの場合、Traffic Managerは、ホスト名を保持するドメイン全体に対する権威DNSサーバとして機能します。DNSリクエストはDNS仮想サーバに受信され、内蔵のDNSサーバまたはDNSサーバの外部プール of のいずれかである、関連付けられたプールによって処理されます。



Traffic Managerをインラインの手法でデプロイするには、グローバルに分散させたい各ホスト名に対して複数の「A」レコードを返すように、内蔵のDNSサーバまたは外部DNSサーバを設定します。内蔵のDNSサーバの設定方法については、[Traffic ManagerのDNSサーバ](#)を参照してください。

次に、Traffic Manager上のDNS仮想サーバを作成します。範囲を最大にするには、DNS (TCP)およびDNS (UDP)の両方の仮想サーバを作成することをお奨めします。必要に応じて、内蔵のDNSサーバプールまたはご使用の外部DNSサーバのいずれかにデフォルトプールを設定します。外部DNSプールを使用する場合、Traffic Managerは全てのDNSリクエストをこれらのネームサーバにプロキシします。

グローバルに分散させたいホスト名に対するDNS仮想サーバにGLBサービスを追加し、親ドメインの委任レコードを更新して、Traffic Managerにそのドメインに対する権威を持たせます。

親ドメインには、多数のレコードタイプが保持されています。

- **NS (委任)レコード:** ドメインに権威を持つDNSサーバのホスト名です。通常、複数のNSレコードを用意します。
- **AおよびAAAA (グルー)レコード:** 全てのNSレコードに対して、AレコードまたはAAAAレコードのいずれかを用意する必要があります。特に、NSレコードのホスト名がご使用のドメインに保持されている場合、親ゾーンには、そのホスト名のIPアドレスを提供するグルーAまたはAAAAレコードが保持されている必要があります。

親ドメインを更新するには、その親ドメインを管理する組織を最初に特定します。組織とは、外部のDNSレジストラ、ご自身の組織の別の部署、ご自身を管理するドメインなどです。

DNSレジストラによる委任レコードの更新

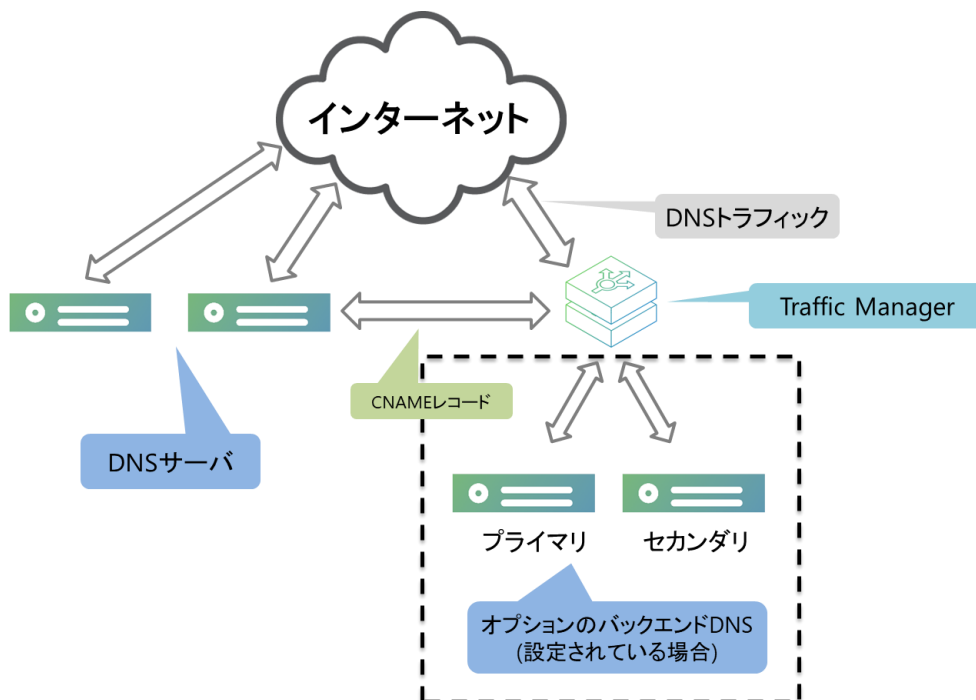
1. 通常、ご利用のDNSレジストラがコントロールするDNS設定は、Webベースのインターフェースから設定することができます。設定インターフェースの詳細については、ご利用のDNSレジストラにお問い合わせください。
2. ドメイン設定ページにログインし、ネームサーバのリストを見つけます。
3. 各レコードは、完全修飾ドメイン名(FQDN)と、それに対応するIPアドレス(オプション)で構成されています。これらのレコードを更新するには、現在のDNSサーバのFQDNとIPアドレスをTraffic Managerのものに置き換えます。

この手順の詳細については、ご利用のDNSプロバイダまたはDNSレジストラにお問い合わせください。

親ドメインの委任レコードが更新されると、Traffic Managerは、負荷分散を行うホスト名の全てのDNS検索を受信ようになります。リクエストを受信すると、Traffic Managerは適切なDNSバックエンドにそのリクエストを転送し、IPアドレスのリストをレスポンスとして受信します。Traffic Managerはこのリストを処理して、エンドユーザに返すIPアドレスを正確に選択します。

並列デプロイ

並列デプロイでは、DNSエントリのCNAMEを、別のドメイン(ほとんどの場合、ご使用のドメインのサブドメイン)のホスト名を指すよう設定します。次に、並列のホスト名に対するGLBを、インラインデプロイの手法に従って設定します。



例えば、「example.com」のドメインを管理し、代替ホスト名の「www.gslb.example.com」を備える並列デプロイを使用して、「www.example.com」でGLBを有効化するとします。

この場合、「example.com」のネームサーバを、以下の2つのルールを満たすように設定します。

- 「www.example.com」から「www.gslb.example.com」へのリダイレクトを行う
- Traffic Managerに対して「gslb.example.com」の権威を委任する

次に、内部のTraffic Manager DNSサーバ機能を使用するか、またはバックエンドDNSサーバの別のプールを介して、Traffic Managerで「gslb.example.com」ドメインをホストします。「example.com」のゾーンファイルでは、以下のエントリを追加します。

```
; Delegation for gslb.example.com using the Traffic Manager.
gslb.example.com.      IN NS stm1.gslb.example.com.
                       IN NS stm2.gslb.example.com.

; Glue records for gslb.example.com
stm1.gslb.example.com. IN A 192.0.2.11
stm2.gslb.example.com. IN A 192.0.2.12
```



```
; CNAME for www.example.com to parallel GLB-enabled name.  
www.example.com.      CNAME www.gslb.example.com.
```

DNSクライアントは、ドメイン名www.example.comを参照するときに、「example.com」ドメインのネームサーバと通信します。このネームサーバは、「代わりにwww.gslb.example.comを試す」に相当するCNAMEレスポンスを返します。ネームサーバは「example.com」ドメインに対して権威があるため、「gslb.example.com」に対する一部のネームサーバ(この場合はご使用のTraffic Manager)の名前とIPアドレスに関する補足情報もネームサーバから得られます。

これにより、DNSクライアントは「www.gslb.example.com」に対するリクエストをTraffic Managerに送信し、Traffic Managerはそれを受けて全てのGLBロケーションのIPアドレスが含まれるリストを取得します(内部DNSサーバを使用するか、バックエンドDNSサーバにクエリします)。Traffic Managerはこのリストを処理し、いずれかのGLBロケーションのIPアドレス(1つまたは複数)を保持する1つ以上のAレコードを返します。

並列デプロイにおいて、DNSサーバがTraffic Managerではない場合、並列のホスト名(www.gslb.example.com)に対するDNSサーバは、元のホスト名(www.example.com)に対する変更されていないAレコードを返してはなりません。そのため、トップレベルのDNSサーバはバックエンドDNSサーバとは異なるネームサーバにすることを強くお奨めします。



さらに、複雑になりますが、スプリットホライズンDNSを使用することもできます。ただし、この手法を用いる場合は、変更されていないレスポンスが外部に漏れないように注意が必要です。

バックエンドDNSサーバが全てTraffic Managerの場合、このような問題は起こりません。代わりに、グローバルに負荷分散されたホスト名をパブリックに提供する全てのDNS仮想サーバで、GLBサービスを確実に設定する必要があります。

並列デプロイでは、DNSサブドメインを新たに作成および設定する必要があるため、デプロイがより複雑になる可能性があります。ただし、このデプロイはより柔軟性に優れており、DNS設定を頻繁に変更することが予測される場合や、多数のドメインに同じサブドメインを使用して設定を一元化する場合に最適です。

例えば、多数のドメイン名(www.example.com、www.mysite.com、およびwww.example.org)のエイリアスを同一のCNAME(www.gslb.example.com)に設定することにより、ご使用のTraffic Managerを使用してホストできるようになります。そうすることで、Traffic ManagerのバックエンドDNSサービスのデプロイを適切に設定することにより、多数のドメインを単一のCNAMEから一元的に設定することができます。

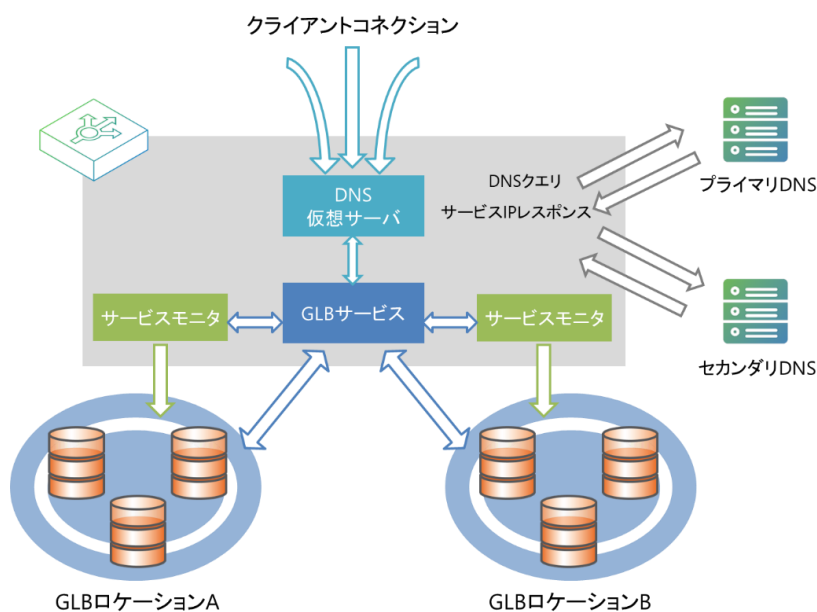
存続時間(TTL)フィールド

DNSレコードを変更する場合、その変更が即座にインターネットトラフィックで有効にならない可能性があります。これは、各DNSレコードに存続時間(TTL)フィールドが含まれているからです。一般に、TTLフィールドは、ダウンストリームのDNSサーバに対してこのレコードを指定秒数だけキャッシングするよう指示するものです。その秒数の経過後、DNSサーバは新しいリクエストを送信しなければなりません。レコードがタイムリにインターネット全体に伝搬されるようにするためには、TTL値を30～60秒に設定することをお奨めします。

Traffic Manager GLBデプロイのコンポーネント

このセクションでは、Traffic Manager GLBのアーキテクチャの概要とともに、有効なグローバル負荷分散の基本コンセプトについて取り上げます。

ご使用のGLBデプロイを設定する前に、このセクションをお読みいただくことをお奨めします。GLBの設定については、[GLBの設定](#)を参照してください。



GLBロケーション

Traffic ManagerはGLBロケーションを、サービスの提供に使用する独立したサイトと見なします。GLBロケーションはデータセンタを表すことがよくあります。世界の複数の国に2つ以上のGLBロケーションをデプロイすることにより、各地の必要条件に適したアプリケーション配信インフラストラクチャを提供したり、重要システムの完全な災害復旧を実現したりできます。Traffic ManagerがGLBロケーションに物理的に存在する必要はありませんが、物理的に同じロケーションにある場合、性能および可用性上のメリットが得られます。

Traffic Managerはサービスモニタを使用して、GLBロケーションにより提供されているサービスの健全性を判断します。クライアントがDNS検索を行うと、Traffic ManagerのGLBサービスは、そのクライアントにとって最適なGLBロケーションを選択します。Traffic Managerはその後、選択したGLBロケーションに送信しているクライアントにDNSレスポンスを返します。



GLBロケーションは、[ロケーション設定](#)で説明しているロケーション設定とは別のコンセプトです。GLBを実装している場合、Traffic Managerは、それ自体がロケーション設定に存在するものとして指定されますが、Traffic Managerによって管理されるGLBサービスとは関係ありません。

複数のロケーションにあるTraffic Managerが同じDNSサーバを使用することもできます。ただし、性能上および信頼性上の理由から、ローカルにあるDNSサーバを使用するようTraffic Managerを設定することをお奨めします。これを行うには、GLBロケーションと、[マルチサイトクラスタの管理](#)で説明したロケーション設定機能を組み合わせます。

GLBサービス

GLBサービスは、一連のDNSドメインに使用するグローバル負荷分散設定を表します。GLBサービスでは、主に以下の情報を設定します。

- GLBサービスの適用先ドメイン
- これらのドメインのサービスをホストしているGLBロケーション
- そのGLBロケーションで使用する負荷分散アルゴリズム

さらに、以下の設定を行うことができます。

- DNSクライアントがそのドメインのDNS情報を確認する頻度を定義するTTL (存続時間)値
- ドレイン: GLBロケーションをドレインすると、クライアントからそのロケーションにトラフィックが送信されなくなります。
- モニタ: 1つ以上のサービスモニタを設定することにより、各GLBロケーションのサービスおよびホストが正常に動作しているかどうかを確認できます。
- ロギング: チェックおよびデバッグのために、全てのDNSリクエストおよびレスポンスを記録することができます。

GLBを設定した仮想サーバとプール

各Traffic Managerは、GLBサービス用に特別に設定された仮想サーバを通して、受信するDNSリクエストを監視します。この仮想サーバは、DNS (UDPまたはTCP)を使用して、バックエンドDNSサーバノードで構成されるプールへのトラフィックを負荷分散します。

仮想サーバに割り当てられた複数のGLBサービスがある場合、Traffic Managerは、リクエストされたドメインに基づいて適切なGLBサービスを選択します。

DNSサーバ

DNSサーバをノードとして含むプールを作成すると、Traffic Managerは、そのプール設定に定義されている負荷分散アルゴリズムに従ってプールを負荷分散します。特定のDNSサーバノードがタイムアウトになると、アラートが報告され、リクエストを処理しているTraffic Managerは次のDNSサーバを試します。

性能上の理由から、Traffic ManagerをDNSサーバと同じ物理的ロケーションにインストールしようとするかもしれませんが、こうすることで必ずしも性能が向上するとは限りません。

サービスIPアドレス

Traffic Managerがクライアントに送信するレスポンスを操作するためには、どのIPアドレスがどのGLBロケーションに属しているかを認識していなければなりません。サービスIPアドレスは、管理対象のドメインについてのクエリに対してDNSサーバが一般にラウンドロビンのメカニズムで返すIPアドレスに一致します。

サービスモニタ

各GLBサービスに、各GLBロケーションで実行するサービスモニタのリストを設定することができます。GLBロケーションに定義された全てのサービスモニタが成功しないと、Traffic ManagerによるGLBサービスの負荷分散の決定でそのGLBロケーションが考慮されません。

いずれかのモニタが(そのモニタに定義されている失敗しきい値制限内に)失敗した場合、そのGLBロケーション全体に障害発生マークが付けられ、その失敗を検出したTraffic Managerは、クライアントの負荷分散時にそのロケーションを除外するようになります。



環境によっては、地理的に分散されたGLBデプロイ内の1つ以上のTraffic Managerが、隔離されたネットワークの障害などによって、ピアと同じ失敗を検出しないことがあります。そのようなTraffic Managerに対してサービスモニタから成功したレスポンスが返されると、Traffic Managerは、ピアが利用不可と検出したGLBロケーションを使用し続けることがあります。

一般に、サービスモニタは、指定された特定のサービスIPアドレスまたはホスト名が正常に動作しているかどうかをテストします。例えば、適切にコンテンツを提供しているかどうかを検証するために、サービスモニタを、そのGLBロケーションにあるプライマリサーバの負荷分散をテストするように設定することがあります。プライマリサーバの負荷分散には、DNSによってそのGLBロケーションにあるそのサービスに対して発行されたIPアドレスが割り当てられます。

複数のサービスまたはホストが正常に動作しているかどうかをテストする場合は、1つのGLBロケーションに複数のサービスモニタを割り当てることができます。

複数のGLBサービスに複数のGLBロケーションを設定した場合、それぞれのサービスモニタは独立して動作します。あるGLBロケーションで1つのGLBサービスのモニタが失敗した場合でも、他のGLBサービスのモニタが成功すれば、これらのGLBサービスでは引き続きこのGLBロケーションが使用されます。

「Load」または「Adaptive」負荷分散手法を使用している場合、サービスモニタはオプションで性能情報を負荷分散アルゴリズムに提供することができます。これにより、負荷分散アルゴリズムでは、最高の性能が報告されたGLBロケーションに多くのトラフィックが送信されます。

性能データは、サービスモニタが完了までに要した時間、すなわちモニタされているGLBロケーションのレスポンス時間に基づきます。

GLBの設定

Traffic Managerは、[Traffic Manager GLBデプロイのコンポーネント](#)で説明されているコンポーネントを使用して、グローバル負荷分散を実行します。このセクションでは、一般的なTraffic Managerのデプロイでそれらの各コンポーネントを設定する方法について説明します。

概要

Traffic Managerが、地理的に離れたロケーション間で負荷分散を行うには、以下の条件を満たす必要があります。

- GLBロケーションが提供されるように定義する
- サービスモニタを作成し、各GLBロケーションの可用性をモニタリングする
- GLBサービスを作成し、リクエストされるドメインに対するDNSクエリを管理する
- バックエンドDNSサーバのプールを作成する
- DNS仮想サーバを作成し、DNSリクエストを監視する

GLBロケーションの定義

GLBロケーションを表示、変更および新規追加するには、**Catalogs** (カタログ) > **Locations** (ロケーション) ページを使用します。既存のGLBロケーションの名前または地理的位置を変更するには、ロケーション名をクリックするか、関連付けられた**Edit** (編集) ボタンをクリックします。

 このページには、ロケーション設定のリストとGLBロケーションのリストが別々に表示され、各リストで別々にロケーションの追加および削除を行えます。ロケーション設定の詳細については、[マルチサイトクラスタの管理](#)」を参照してください。

新しいGLBロケーションの作成

1. **Catalogs** (カタログ) > **Locations** (ロケーション) に移動します。
2. 「Create new GLB Location (新しいGLBロケーションの作成)」セクションで、**Name** (名前) テキストボックスに識別名を入力します。
3. **Type** (タイプ) では「GLB」を選択します。

 マルチサイトクラスタの管理が無効化されている場合、このページで作成した全てのロケーションのタイプはGLBとなるため、このフィールドは表示されません。

4. **Add Location** (ロケーションの追加) をクリックすると、このGLBロケーションが追加され、ロケーション編集ページにアクセスします。
5. ロケーション編集ページでは、GLBロケーションの**Position** (位置) を選択します。国名のドロップダウンリストから選択するか、特定の緯度と経度の座標を選択するか、表示されるマップ上のポイントを選択します。
6. **Update Location** (ロケーションの更新) をクリックして変更を保存します。

GLBロケーションを削除するには、まずそのロケーションを使用しているGLBサービスとの関連付けを解除します。次に、**Confirm** (確認) チェックボックスをクリックし、**Delete Location** (ロケーションの削除) をクリックしてシステムからロケーションを削除します。

サービスモニタの作成

GLBロケーションにサービスモニタを設定するには、新しいプール/GLBモニタを作成し、各GLBロケーションでホストされているサービスのIPアドレスまたはホスト名をモニタリングするように指示します。選択するモニタタイプによっては、追加のパラメータが必要なことがあります。詳しくは、[ヘルスモニタリング](#)を参照してください。

サービスモニタの作成

1. **Catalogs** (カタログ) > **Monitors** (モニタ)に移動します。
2. 「Create new monitor (新しいモニタの作成)」セクションで、以下を入力します。
 - **Name (名前)**: 作成するサービスモニタの識別名です。
 - **Type (タイプ)**: 必要条件に最も適したモニタタイプを選択してください。
 - **Scope (スコープ)**: **Pool/GLB** (プール/GLB)をクリックし、モニタリングしたいGLBロケーションに存在するサービスのIPアドレスまたはホスト名を入力します。モニタタイプによっては、ポート番号の入力も必要です。
3. **Create Monitor** (モニタの作成)をクリックすると、このサービスモニタが作成され、編集ページにアクセスします。
4. オプションで必要なパラメータを調整し、**Update** (更新)をクリックして変更を保存します。
5. この手順を、モニタリングしたい各GLBロケーションに対して行います。

GLBサービスの作成

GLBサービスは、FQDNとIPアドレスのセットに基づきます。新しいGLBサービスを作成するときは、各ドメイン名についてそれぞれのGLBロケーションでホストされるIPアドレスをTraffic Managerに指定します。これにより、接続されている全てのTraffic Managerが各GLBサービスのDNS検索を管理するようになります。

GLBサービスの作成

1. **Catalogs** (カタログ) > **GLB Services** (GLBサービス)に移動します。
2. 「Create a new GLB service (新しいGLBサービスの作成)」セクションで、以下を入力します。
 - **Service Name (サービス名)**: このGLBサービスの識別名です。
 - **Domains (ドメイン)**: このGLBサービスにグローバルに負荷分散させたいドメイン名を少なくとも1つ入力します。複数のドメインを入力するには、リストをカンマまたはスペースで区切ります。
 - **Add locations (ロケーションの追加)**: GLBサービスをホストさせたい、あらかじめ定義されたGLBロケーションを1つ以上クリックします。GLBロケーションの作成については、[GLBロケーションの定義](#)を参照してください。
3. **Create GLB Service** (GLBサービスの作成)をクリックすると、このGLBサービスが作成され、編集ページにアクセスします。

GLBサービスの設定を変更するには、それぞれのGLBサービスの編集ページを使用します。このページには、サービスを作成した際に入力されたGLBサービスの基本設定と、サブページに分類されたその他の利用可能な設定が表示されます。

新たに作成されたGLBサービスは、それぞれ無効化された状態で追加されます。GLBサービスを有効化するには、**Enabled** (有効化)を「Yes (はい)」に設定します。

GLBサービスが負荷分散するドメインを追加または削除するには、**Domains** (ドメイン)テーブルを使用します。グローバル負荷分散を行う全てのFQDNのエントリはここで作成します。

ワイルドカード文字の指定には「*」を使用します。例えば、全ての「example.com」ドメインの全てのDNS検索を管理する場合は、「*.example.com」と入力します。

Locations and Monitoring (ロケーションとモニタリング)

このGLBサービスに設定されているGLBロケーションおよびサービスモニタを設定するには、**Catalogs** (カタログ) > **GLB Services** (GLBサービス) > **Locations and Monitoring** (ロケーションとモニタリング)をクリックします。

Traffic Managerには、設定されている各GLBロケーションに対する専用の設定サブセクションが用意されています。**Update** (更新)をクリックして変更を保存します。

通常、最初にGLBサービスを作成するときに、使用したい各GLBロケーションを選択します。追加のGLBロケーションを追加するには、「Add Location (ロケーションの追加)」セクションを使用します。GLBロケーションを削除するには、目的のGLBロケーション設定サブセクションで、専用の**Remove this location** (このロケーションを削除)リンクを見つけてクリックします。



新たに作成したGLBサービスの場合、選択したGLBロケーションに関連する警告がTraffic Managerによって表示されることがあります。この警告は、各GLBロケーションに対して適切なサービスIPアドレスを提示するまで発し続けられます。

設定した各GLBロケーションについて、以下の設定を行うことができます。

- **Draining:** このGLBロケーションへのトラフィックの送信を停止するには、これを「Yes (はい)」に設定します。例えば、装置のあるロケーションから別のロケーションへと物理的に移動する場合や、バックアップ電源システムをアップグレードするときに、UPS、ジェネレータ、インバータ、バッテリーを切断する必要がある場合、このオプションを使用して、Traffic ManagerにこのGLBロケーションへのトラフィックの送信停止を指示することができます。
- **Service IPs:** Traffic Managerがクライアントに送信するレスポンスを操作するためには、どのIPアドレスがどのGLBロケーションに属しているかを認識していなければなりません。この設定オプションを使用すると、このGLBロケーションに対するIPv4アドレスまたはIPv4 CIDRサブネットのリストまたは範囲を入力することができます。これらのIPアドレスは、管理対象のドメインについてのクエリに対してDNSサーバがラウンドロビンのメカニズムで返すIPアドレスに一致します。



一般に、これらのIPアドレスは、このGLBロケーションに割り当てるサービスモニタで設定するIPアドレスと同じです。

- **Monitors:** このGLBロケーションにTraffic Managerが使用するべきサービスモニタを選択します。各モニタでは1つのテストが実行され、各GLBロケーションに複数のモニタを設定することができます。選択できるのは、ホスト全体またはプール全体のスコープを持ったモニタのみです。モニタテストは、サービスモニタで設定されている特定のターゲットに対して、設定された間隔で実行されます。モニタが連続失敗回数の限度を超えた場合、特定のデータセンタのサービスに利用不可のフラグが設定されます。その結果、このGLBロケーションは、以降のDNSレスポンスから除外されます。

負荷分散

このGLBサービスに対して負荷分散および障害復旧を設定するには、**Catalogs** (カタログ) > **GLB Services** (GLBサービス) > **Load Balancing** (負荷分散)をクリックします。

利用できる負荷分散手法には以下のものがあります。

- **Load:** GLBロケーションで検出された負荷に基づいてトラフィックを分配します。Traffic Managerはヘルスマニタを使用して、GLBロケーションに送信されたリクエストのレスポンスを受信するまでの時間を測定し、負荷を判断します。つまり、負荷値はGLBロケーションに送信されたデータパケットの、測定された往復時間(ミリ秒単位)となります。50ms未満の往復時間が測定された場合は常に、Traffic Managerは、類似のローカルロケーションが同等に扱われるように、代わりに50の負荷値を使用します。

各GLBロケーションに特定の負荷値をプログラムで設定することにより、このメカニズムをオーバーライドし、負荷分散の決定に影響を及ぼすことができます。これを行うには、Traffic ManagerのSOAP APIを使用するか、外部プログラムモニタを作成します。これらの方式では、Traffic Managerは、トラフィックの分配を決定する場合に、代わりにこの値を使用します。

SOAP API方式の場合、各ロケーションに対して負荷値を宣言するには、GLB.Service.setLoad()関数を使用します。SOAP APIを無効化し、GLBロケーションのヘルスマニタに負荷レポートを戻すには、負荷値を0に設定します。

外部プログラムモニタ方式の場合は、ロケーションのGLB負荷を計算し、次のフォーマットを使用してSTDOUTにこの値を出力するスクリプトまたはプログラムを作成します。

```
vTM-Set-node-load: <load-value>
```

ここで、<load-value>は整数です。

次に、ロケーションに対する外部プログラムモニタとしてそのスクリプトまたはプログラムを追加します。



SOAP APIまたは外部プログラムモニタを使用して負荷を50未満に設定すると、Traffic Managerは、代わりに50の負荷値を使用します。すなわち、この動作は、50ms未満の往復時間が測定された場合と同じです。



詳細モードを使用すると、プログラムのSTDOUT出力がTraffic Managerのイベントログに出力されます。

- **Geographic:** 各クライアントの地理的位置にのみ基づいてトラフィックを分配します。
- **Adaptive:** 現在のGLBロケーションの負荷とクライアントの地理的位置の両方に基づいてトラフィックを分配します。特定のロケーションがトラフィックのほとんどを受信している(ほぼ過負荷状態になっている)ことが判明した場合、この負荷分散アルゴリズムを使用することで問題が解消されることがあります。この場合、このアルゴリズムによって、距離による待ち時間と負荷による待ち時間のバランスを最適な状態で取ることができます。
- **Geo Effect:** このオプションを使用すると、Traffic Managerがクライアントのトラフィックの送信先GLBロケーションを決定するときの、地理的位置による影響の程度を微調整することができます。
- **Round Robin:** 利用可能な各GLBロケーションの現在のシーケンシャルな選択に基づいてトラフィックを分配します。時間が経つにつれ、全てのGLBロケーションが同数のリクエストを受信します。
- **Weighted Random:** ランダムセレクションジェネレータを使用して、**Location Weights**の設定による加重に従ってトラフィックを分配します。Traffic Managerは、より大きな加重が設定されたGLBロケーションにより多くのトラフィックを送信します。
- **Location Weights:** 各GLBロケーションに対して必要な比例加重を設定します。時間が経つにつれ、GLBロケーションは、この設定を使用して宣言された比率に等しい数のリクエストを受信します。
- **Primary/Backup:** Traffic Managerは、プライマリGLBロケーションで障害が発生した場合にはバックアップGLBロケーションのみを頼りにして、常に1つのプライマリGLBロケーションに対するDNSレコードを送信します。状態情報に依存するサービス(買い物かごWebサイトなど)を提供し、データベースをリアルタイムにグローバルに複製していない場合、このアルゴリズムが適している場合があります。

GLBロケーションのリスト内の優先順を決定するには、各GLBロケーションのインジケータバーの左にあるドラッグアンドドロップタブを使用します。(トップダウンの優先順で)GLBロケーションを昇格させるには、リスト内でそのロケーションを目的の位置までドラッグします。

- **Automatic Failback:** 優先順の高いGLBロケーションが障害から復旧したときに、どのような処理を行うかを定義します。「Yes (はい)」をクリックすると、優先順に高いGLBロケーションが自動的にアクティブとしてマークされ、Traffic Managerは全てのユーザをそのロケーションに送信するようになります。「No (いいえ)」をクリックすると、復旧したGLBロケーションは非アクティブのままになります。この場合、Traffic Managerはユーザをその時点でアクティブなGLBロケーションに送信し続けます。
- **Activate best available location:** Automatic Failbackが「No (いいえ)」に設定されている場合にこのボタンをクリックすると、Traffic Managerは、その時点で有効化されているか利用可能なGLBロケーションのリストから新しいアクティブなGLBロケーションを選択します。



GeographicとAdaptiveのいずれの負荷分散アルゴリズムでもGeoIPデータベースのインストールが必要です。詳細については、[GeoIPデータベースの使用](#)を参照してください。

全ての負荷分散アルゴリズムに障害復旧オプションを設定するには、以下の設定を使用します。

- **disable_on_failure:** GLBロケーションの復旧が自動的に有効化されないようにするには、「Yes (はい)」をクリックします。その後、特定の無効化されたGLBロケーションを手動で有効化するには、目的のGLBロケーションインジケータバーのEnable (有効化)をクリックします。この設定は、復旧したGLBロケーションを使用する前に状態またはコンテンツを同期する必要があるデプロイに適しています。
- **autorecovery:** 全てのGLBロケーションで障害が発生した場合に「Yes (はい)」をクリックすると、Traffic Managerは、最後に障害が発生したGLBロケーションを、そのロケーションが復旧したときに自動的に有効化します。「Primary/Backup」アルゴリズムを使用している場合、復旧したGLBロケーションもアクティブになります。「No (いいえ)」をクリックすると、復旧した全てのロケーションは無効化されたままになります。



「autorecovery」の設定は、有効化されている場合、「Automatic Failback」および「disable_on_failure」に優先します。

- **last_resort_response:** このGLBサービスに接続されている全てのGLBロケーションが利用不可になった場合、Traffic Managerは、この設定の内容を最終手段のDNSレスポンスとして全てのクライアントリクエストに返します。Traffic ManagerがDNSサーバから受信した全てのIPアドレスを返すようにするには、スペースで区切られたIPアドレスのリスト、単一のドメイン名、または「ALL」のいずれかを使用します。

DNS Authentication (DNS認証) (DNSSEC)

このGLBサービスに設定されているドメインネームシステムセキュリティ拡張(DNSSEC)を設定するには、**Catalogs (カタログ) > GLB Services (GLBサービス) > DNS Authentication (DNSSEC) (DNS認証 (DNSSEC))**をクリックします。

DNSSEC仕様は、DNSにセキュリティ強化および認証のための一連の拡張機能を付加したものです。Traffic ManagerがDNSSEC認証されたレスポンスを変更できるようにするには、このページで、署名ドメインとDNSSEC秘密鍵の関連付けを設定します。

関連付けの追加

1. 表示されるボックスに署名ドメインを入力し、プラス記号をクリックします。
2. ドロップダウンボックスからドメインに関連付けられた秘密鍵を選択します。
3. **Update** (更新)をクリックして関連付けを確定します。

複数の鍵を1つのドメインに関連付けたり、個々の鍵を使用して多数のドメインを認証(多対多の関係)したりすることができます。

DNSSEC秘密鍵を追加するには、**Manage DNSSEC private keys** (DNSSEC秘密鍵の管理)をクリックして **Catalogs** (カタログ) > **SSL** > **DNSSEC Keys** (DNSSEC鍵)ページにアクセスします。SSL証明書および鍵のインポートおよびアップロードの詳細については、[SSL暗号化](#)を参照してください。

DNSSECの詳細については、www.dnssec.netを参照してください。

Rules (ルール)

このGLBサービスに対してTrafficScriptルールを設定するには、**Catalogs** (カタログ) > **GLB Services** (GLBサービス) > **Rules** (ルール)をクリックします。

このGLBサービスにTrafficScriptルールを適用することで、追加的なトラフィック管理機能を利用できるようになります。ただし、DNSプロトコルと互換性のない機能(http.*など)は使用できません。TrafficScriptとその機能の詳細については、[TrafficScriptルール](#)を参照してください。

Request Logging (リクエストロギング)

このGLBサービスに対してリクエストロギングを設定するには、**Catalogs** (カタログ) > **GLB Services** (GLBサービス) > **Request Logging** (リクエストロギング)をクリックします。

環境によっては、Traffic Managerクラスタに送信されてくるDNSリクエストをさらに正確に把握することが必要になる場合があります。リクエストロギングでは、各リクエストを指定のファイルに記録し、後で解析に利用することができます。

以下を設定することができます。

- **log!enabled:** リクエストロギングを有効化するには、「Yes (はい)」に設定します。
- **log!filename:** Traffic ManagerがこのGLBサービスに対するリクエストログを保存するローカルファイルシステムのフルパスおよびファイル名です。**Macros...** (マクロ)をクリックすると、利用可能なマクロのリストを確認できます。

- **log!format:** リクエストの文字列フォーマットです。各リクエストに関する情報には、「%」でエスケープしたマクロによってアクセスできます。**Macros...** (マクロ)をクリックすると、利用可能なマクロのリストを確認できます。

ご使用のGLBサービスに対するリクエストログを表示するには、**View Request Logs in file** (リクエストログをファイルで表示)をクリックします。

DNS設定

このGLBサービスに対するDNS設定を行うには、**Catalogs** (カタログ) > **GLB Services** (GLBサービス) > **DNS Settings** (DNS設定)をクリックします。以下を設定することができます。

- **Time To Live (TTL):** DNSサーバにより、DNSレスポンスの各リソースレコードのTTL値が設定されます。TTL値は、キャッシングリゾルバがDNSレコードを再度名前解決するまでキャッシングしておく時間をコントロールします。GLBロケーションに障害が発生し、DNSサーバによって設定されたTTL値が長すぎる場合(数時間または数日など)、一般に、GLBサービスでは迅速なフェールオーバーが必要です。Traffic Managerによって処理されるリソースレコードに対してTTLをオーバーライドするには、**Custom** (カスタム)をクリックし、新しい値(秒単位)を入力します。

DNSサーバプールの作成

GLB仮想サーバがグローバル負荷分散を行うためには、DNSサーバノードが含まれるプールを作成する必要があります。

DNSサーバプールの設定

1. **Services** (サービス) > **Pools** (プール)に移動します。
2. 「Create a new Pool (新しいプールの作成)」セクションで、以下を追加します。
 - **Pool Name** (プール名): 新しいプールの識別名です。
 - **Nodes** (ノード): 1つ以上のDNSサーバのIPアドレス、または<host>:<port>のフォーマットで記されたホスト名と対応するポート番号のリストです。複数のノードを指定するには、スペースまたはカンマ区切りを使用します。
 - **Use autoscaling** (オートスケーリングの使用): このボックスのチェックを外します。
 - **Monitor** (モニタ): ご使用のDNSサーバのヘルスをチェックするモニタをオプションで指定します。
3. **Create Pool** (プールの作成)をクリックすると、このプールが作成され、編集ページにアクセスします。

DNS仮想サーバの作成

仮想サーバは、受信するDNSリクエストを処理し、そのリクエストをバックエンドDNSサーバプールに渡します。作成したGLBサービスは、この仮想サーバに関連付けます。

DNS仮想サーバの作成

1. **Services** (サービス) > **Virtual Servers** (仮想サーバ)に移動します。
2. 「Create a new Virtual Server (新しい仮想サーバの作成)」セクションで、以下を入力します。
 - **Virtual Server Name** (仮想サーバ名): 作成する仮想サーバの識別名です。
 - **Protocol** (プロトコル): ドロップダウンリストから「DNS (UDP)」または「DNS (TCP)」を選択します。
 - **Port** (ポート): この仮想サーバが監視する受信ポート番号を選択します。
 - **Default Traffic Pool** (デフォルトトラフィックプール): ご使用のDNSサーバノードが含まれるプールを選択します。詳細については、[DNSサーバプールの作成](#)を参照してください。
3. **Create Virtual Server** (仮想サーバの作成)をクリックしてこの仮想サーバを作成し、編集ページにアクセスします。
4. **GLB Services** (GLBサービス)をクリックすると、「Edit GLB Services (GLBサービスの編集)」ページにアクセスします。
5. 「Add new GLB Service (新しいGLBサービスの追加)」セクションで、ドロップダウンボックスから目的のGLBサービスを選択し、**Add Service** (サービスを追加)をクリックします。GLBサービスの作成の詳細については、[GLBサービスの作成](#)を参照してください。
6. 仮想サーバを有効化するには、Virtual Server edit (仮想サーバ編集)ページで**Enabled** (有効化)を「Yes (はい)」に設定するか、Traffic Managerのホームページの仮想サーバ名の隣にあるplay(作動)アイコンをクリックします。

EDNS0 Client Subnetのサポート

一般に、Traffic Managerでは、ロケーションに基づいてGLBの決定を下す場合に、受信DNSリクエストのIPアドレスを使用して、認識されたクライアントの地理的ロケーションが決定されます。しかし、このIPアドレスは、送信元のクライアントではなく、アップストリームサーバまたは再帰的リゾルバに属することがあります。多くの場合、DNSリクエストのIPアドレスは、トポロジ的にリクエスト元に近いと見なすことが安全です。しかし、状況によっては、クライアントとアップストリームリゾルバ間の距離が大きく、Traffic Managerであまり効率的でないGLBの決定が下されることがあります。

この状況における問題を軽減するために、Traffic Managerは、EDNS0 Client Subnetプロトコル拡張機能の使用をサポートして、DNSリクエストで送信元クライアントのサブネットIPアドレスの識別を容易にします。この機能を有効化すると、GLBサービスでは、DNSリクエストの送信元アドレスではなく、この拡張機能で保存されたネットワークアドレスが使用されます。

EDNS0 Client Subnetのサポートを有効化するには、DNS仮想サーバで`dns!edns_client_subnet`をYes (はい)に設定します。

EDNS0 Client Subnetオプションの詳細については、RFC 7871(<https://tools.ietf.org/html/rfc7871>)を参照してください。

トラフィックの可視化

GLBが有効化されている場合、Traffic Managerでは、管理UIの既存の可視化および診断ツールに追加情報が表示されます。例えば、ワールドマップには、受信したDNSリクエストと、その後クライアントが接続されたGLBロケーションがリアルタイムに表示されます(詳細については、[ワールドマップ](#)を参照)。

現在のアクティビティグラフ

システム上のアクティビティのリアルタイムのグラフを確認するには、**Activity** (アクティビティ) > **Current Activity** (現在のアクティビティ)をクリックします。

「Settings (設定)」セクションで、データをTraffic Manager別またはロケーション設定別にプロットできます。どちらの場合も、システムに設定されている対応するTraffic Managerまたはロケーションの全部または一部を選択したり、結果を統合してプロットするか別々にプロットするかを選択したりできます。

GLBが有効化されていると、Traffic Managerは、GLBサービスおよびロケーションに関するメトリクスの新しいセット2つを、**Current Activity** (現在のアクティビティ) > **Edit** (編集)ページに表示されるモニタ値ツリーに追加します。

現在のアクティビティグラフの詳細については、[Traffic Manager管理インターフェースの主要機能](#)を参照してください。

ヒストリカルアクティビティグラフ

Traffic Managerは、過去90日間のシステムのアクティビティを記録します。このデータをグラフ形式で表示するには、**Activity** (アクティビティ) > **Historical Activity** (ヒストリカルアクティビティ)をクリックします。データは、線形軸または対数軸にプロットできます。

「Settings (設定)」セクションで、データをTraffic Manager別またはロケーション設定別にプロットできます。どちらの場合も、システムに設定されている対応するTraffic Managerまたはロケーションの全部または一部を選択したり、結果を統合してプロットするか別々にプロットするかを選択したりできます。タイムスケールは1時間から90日間までの範囲でプロットできます。

ヒストリカルアクティビティグラフの詳細については、[Traffic Manager管理インターフェースの主要機能](#)を参照してください。

Connections (コネクション) ページ

GLBクラスタで行われた最近の負荷分散の決定に関する情報を表示するには、**Activity** (アクティビティ) > **Connections** (コネクション) をクリックします。Connections (コネクション) ページの詳細については、[Traffic Manager管理インターフェースの主要機能](#)を参照してください。

GLBリクエストログ

各GLBサービスが処理する個々のリクエストを表示するには、**Activity** (アクティビティ) > **View Logs** (ログの表示) をクリックし、確認したいGLBサービスの名前をクリックします。

Traffic Managerがログ情報の収集を開始するためには、まずGLBサービスに対してリクエストロギングを有効化する必要があります。詳細については、[リクエストロギング](#)を参照してください。

DigによるDNSのテスト

「Dig」は、DNSシステムのテストおよびトラブルシューティングに推奨されるツールで、Berkley Internet Name Domain (BIND) ソフトウェアの管理も行っているInternet Systems Consortium, Inc.によって管理されています。Digはオープンソースソフトウェアのため、多くのアーキテクチャおよびOSで利用できます。

Linuxの場合、digは一般に、実行するディストリビューションに応じて、BINDとともにインストールされているか追加パッケージとして提供されます。

詳細については、Internet Systems Consortium公式Webサイト(<http://www.isc.org/downloads/bind>)をご覧ください。

Digの使用例は以下のとおりです。

```
$ dig -t any +norec www.example.com @192.0.2.1
```

上記の例では、

- 「-t any」は、全てのタイプのレコード(A、NS、CNAME、PTRなど)をリクエストします。

- 「+norec」は、再帰的フラグを「off」に設定してリクエストを送信するよう指定します。
- 「www.example.com」は、テスト対象のレコードです。
- 「@192.0.2.1」は、リクエストの送信先ネームサーバです。

DNSサーバまたはサービスをテストするときに、必ず非再帰的クエリを実行することが非常に重要です。これにより、信頼できる回答を受け取ることができるか、または名前解決しようとしているドメイン名に権威を持つネームサーバを正確に知ることができます。クエリ先のネームサーバが検索した回答を受け取ることはありません(必ずしも情報源を検証できるわけではないため)。

GeoIPデータベースの使用

Traffic Managerの一部の機能には、IPアドレスをロケーション(経緯度、都市、郡および国)にマッピングするGeoIPデータベースが必要です。GLBは、このGeoIPデータベースを使用して、リモートユーザとローカルのデータセンタ間の距離を推測します。

データベースには、含まれているgeo.*TrafficScriptおよびJava関数を使用して直接アクセスできます。例えば、サイト訪問者のアクセス元の2文字の国コードを検出するには、以下のTrafficScriptを使用します。

```
$ip = request.getRemoteIP();  
$countryCode = geo.getCountryCode( $ip );
```

Traffic Managerの旧バージョンでは、各リリースにMaxMind社のGeoLite2 Cityデータベースから得られるデータベースが含まれていました。Traffic Managerバージョン20.3以降では、このデータベースを個別に入手し、ダウンロードする必要があります。Pulse Secure TechnicalサポートのWebサイトで入手できる更新パッケージを使用して、GeoIPデータベースのダウンロード、インストール、更新を実行できます。

あるいは、独自のMaxMind GeoLite2データベースまたはGeoIP2データベースを使用します。このプロセスの詳細については、<https://community.pulsesecure.net/t5/Pulse-Secure-vADC/Updating-the-Pulse-Secure-vTM-GeoIP-database/ta-p/35215>を参照してください。

Traffic ManagerのGeoIPデータベースのインストール

このセクションに示す手順に従って、GeoIPパッケージを入手し、Traffic Managerクラスタにインストールします。



このプロセスでは、Traffic Managerインスタンスのリスタートが必要です。サービスへの影響を最小限に抑えることのできるタイミングでこの手順を実行することを検討してください。

GeoIPデータベースのインストール:

1. <https://my.pulsesecure.net>にログインし、**Software Licensing and Download** (ソフトウェアのライセンスとダウンロード) > **Pulse License and Download Center** (Pulseライセンス/ダウンロードセンター) > **Software Download** (ソフトウェアのダウンロード) > **Pulse Secure vADC** > **Pulse Secure Virtual Traffic Manager** > **Related Software** (関連ソフトウェア) > **Accept** (同意) > **Accept** (同意)に移動します。
2. GeoIPデータベースのダウンロードパッケージ(「geoip_update_20210119.tgz」など)を選択し、**View Details** (詳細を表示)をクリックし、**Download** (ダウンロード)をクリックします。
3. ローカルワークステーションにGeoIPパッケージをダウンロードします。
4. いずれかのターゲットTraffic Managerインスタンスの管理UIにログインし、**System** (システム) > **Traffic Managers** > **Upgrade** (アップグレード)ページを使用して、上の手順で入手したGeoIPパッケージをアップロードします。
5. 「Upgrade specified traffic managers (指定したTraffic Managerをアップグレード)」を選択します。
6. ログインしているTraffic Managerインスタンスを選択します。



この方法では、Traffic Managerが個別に更新され、ダウンタイムが最小限に抑えられます。この手順で、サービスの中断がユーザに影響を与えないことを確信した場合は、クラスタ内の全てのTraffic Managerインスタンスの選択を検討してください。

7. **Upgrade** (アップグレード)をクリックします。

Traffic Managerは、GeoIPパッケージの適用後にリスタートされます。
8. アップグレードプロセス中にレポートされた問題をエラーログで確認してください。
9. 全ての他のクラスタメンバに対して、[手順4](#)～[手順8](#)を順番に繰り返します。

Traffic ManagerのGeoIPデータベースの拡張

データベースにはプライベートIPアドレスの範囲に対するロケーションは含まれていません(RFC1918を参照)。また、その他のIPアドレスの範囲も、最近割り当てられたか移動された場合は含まれていないか正確でないことがあります。このセクションでは、内部のGeoIPデータベースを拡張して、IPアドレスの範囲を追加またはオーバーライドする方法について説明します。

データベースへの拡張は、\$ZEUSHOME/zxtm/conf/locations.cfgファイルに保存されます。

1行で指定される各エントリに対して以下のフォーマットを使用します。

```
firstIP lastIP lat lon CC RR city
```

各マッピングには、以下のルールと定義が適用されます。

- 行内の要素は空白で区切ります。
- IPアドレスの範囲は包括的で、緯度(「lat」)および経度(「lon」)は「-」または10進法の度数のいずれかとします。
- 各ロケーションマッピングでは、firstIPとlastIPのみが必須です。全ての他のフィールドは、オプションです。しかし、GeographicまたはAdaptive GLB負荷分散アルゴリズムを使用するには、緯度と経度も指定する必要があります。
- 国コード(「CC」)と地域コード(「RR」)は、「-」または2文字のコードのいずれかです例えば、テキサス州はUS TXになります。専用ファイルの\$ZEUSHOME/zxtm/etc/geo/country_codes.txt および \$ZEUSHOME/zxtm/etc/geo/region_codes.txtに該当コードの一覧を含めます。
- 「San Francisco」のように、都市名にスペースを含めることができますが、ある都市を具体的に言及する必要はありません(説明文であれば許容されます)。

以下にマッピングの例を挙げます。

```
192.168.0.1 192.168.0.128 52.1234 -0.5678 US TX New datacntr
172.16.0.1 172.16.255.255 - - - - Test VPN
99.98.97.96 99.98.97.99
```

IPアドレスマッピングのテスト

以下のTrafficScriptリクエストルールの例を使用して、全ての変更をテストすることができます。

```
$text = "";
# Test your own IP addresses with the whereis function here.
$text .= whereis( "192.168.35.40" ); $text .= "\n";
$text .= whereis( "192.168.199.199" ); $text .= "\n";
$text .= whereis( "17.18.19.20" ); $text .= "\n";

http.sendResponse( "200", "text/plain", $text, "" );

sub whereis( $ip ) {
    return $ip . " is in:\n" .
        " Country:    " . geo.getCountry( $ip ) . "\n" .
```

```
" CountryCode: " . geo.getCountryCode( $ip ) ."\n" .  
" Region:      " . geo.getRegion( $ip ) ."\n" .  
" City:        " . geo.getCity( $ip ) ."\n" .  
" Long/Lat:    " . geo.getLongitude( $ip ) .'/'. geo.getLatitude( $ip ) ."\n";  
}
```

Traffic Managerのクラスタ設定の更新

locations.cfgファイルは直接編集することができ、それを受けてローカルのTraffic Manager設定システムは、このファイルが変更されたことを通知し、ファイル内で定義されたロケーションマッピングを自動的に受け入れます。

しかし、ファイルシステム内の設定ファイルを直接編集することによって、ご使用のTraffic Managerの設定は他のクラスタメンバに自動的に複製されなくなります。更新された設定を手動で複製するには、更新を行ったTraffic Managerで以下のいずれかの手法を使用します。

- 管理UIの**Diagnose** (診断) > **Cluster Diagnosis** (クラスタ診断) ページで複製オプションを使用する
- \$ZEUSHOME/zxtm/bin/replicate-config scriptスクリプトをコマンドラインで実行する

Traffic ManagerのFIPS認証

本章では、連邦情報処理標準(FIPS)に固有の情報を説明するとともに、Traffic ManagerにFIPSを実装する方法について詳しく説明します。

FIPSについて

FIPSは、コンピュータシステムでの使用を目的として、アメリカ合衆国連邦政府によって開発されています。利用可能かつ必要なライセンスを所有している場合、Traffic Managerでは、FIPSに準拠した特別動作モードやサードパーティの暗号化動作モジュールを使用することができます。

注記

Traffic Managerは、「FIPSによる認証」を受けていません。使用可能な場合に限り、Traffic Managerは全てのFIPS動作において認証済みの暗号化モジュールを使用します。

FIPSおよびFIPSとTraffic Managerの関係についての詳細は、サポート窓口にお問い合わせください。

FIPSモード

FIPSモードをアクティブにすることによって得られる最も重要な機能として、トラフィックを処理するときに**FIPS 140-2準拠暗号化モジュール**を使用できることが挙げられます。一般に、Traffic ManagerのFIPSモードは、組織がFIPS準拠に向けて取り組むうえで役立つ機能です。

さらに、以下の2つの理由のため、FIPSモードを設定すると、Traffic Managerの様々な設定/動作が制限されます。

1. 暗号化モジュールへの入力がFIPS 140-2の認証範囲内ではなりません。
2. FIPS 140-2の要件と互いに排他的な動作を有効化できはなりません。制限される動作として、SSL 3.0が挙げられます(FIPS 140-2に適合しているのはTLS 1.0およびそれ以降のみです)。

FIPS 140-2

FIPS 140-2は、アメリカ国立標準技術研究所(NIST)によって発行された、FIPS 140発行シリーズの第2版です。詳細については、以下のサイトで確認できます。<http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>

本ドキュメントでは、暗号化モジュールに対するセキュリティ要件について説明します。FIPS 140-2の要件には、1～4の範囲で4つのセキュリティレベルが定義されています。一般に、ソフトウェアのみのセキュリティモジュールは、最低レベル(1)を達成することしかできないと考えられています。それよりも高いセキュリティレベルには、改ざん保護のような、物理的製品でのみ達成できる要件が設けられています。通常、これらの要件は、暗号化動作に関与するプロセスに関係しています。

さらに、その他にも以下のようなNISTの「特別刊行物」も考慮する必要があります。

- 「Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations (トランスポート層セキュリティ(TLS)実装の選択、設定および使用に関するガイドライン)」、SP 800-52r1 (<http://dx.doi.org/10.6028/NIST.SP.800-52r1>)
- 「Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths (遷移: 暗号化アルゴリズムの使用遷移および鍵長に関する推奨事項)」、SP 800-131A (<http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf>)
- 「鍵強度」の構成要素に関する検討事項、SP 800-51パート1 (http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf)



Traffic Managerは、本ドキュメントに記載された全ての推奨事項を実装しているわけではありません。

FIPS 140-2とTraffic Manager

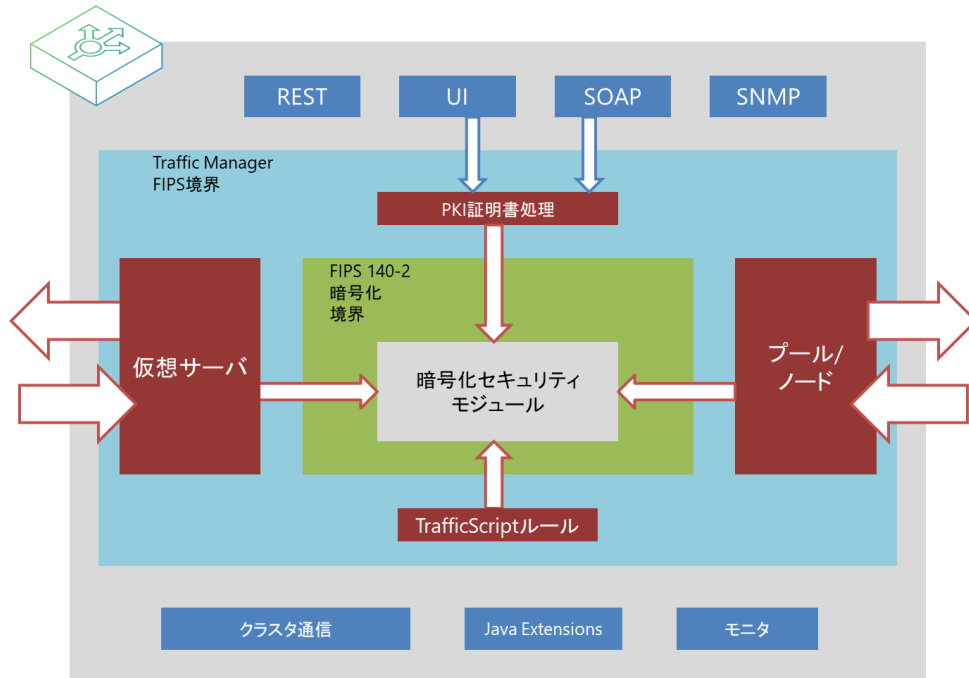
Traffic ManagerのLinux系製品で使用されているFIPS 140-2準拠暗号化モジュールは、Riverbed®暗号化セキュリティモジュール(RCSM)(証明書番号2099)です。このモジュールのFIPS 140-2有効性について、詳しくは以下のNIST CMVPページに記載されています。

<http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2014.htm#2099>

FIPSモードが有効化されているときにTraffic Managerを起動すると、RCSMが読み込まれて初期化されます。初期化(パワーオンセルフテストなど)の最中に発生した障害は、全て致命的なエラーとして報告され、「Traffic Manager FIPS境界」での全ての暗号化動作が無効化されます(SSL復号化仮想サーバ、SSL暗号化プールなど)。

Traffic Manager FIPS境界

Traffic Managerは、SSL/TLSをはじめ、数多くの機能で暗号化を使用しています。ただし、以下に示す専用Traffic Manager FIPS境界内でこれらの暗号化が全て行われているわけではありません。



この境界は、以下のように極めて簡単に定義づけられています。

1. 全ての暗号化動作がトラフィックの処理のために行われる。
2. トラフィックを処理するときに、暗号化オブジェクトとの全ての暗号化相互作用が使用される。

FIPSモードが有効であると、以下のとおり動作します。

項目(1)は主としてTLSを対象とします。仮想サーバおよびプールに関してTraffic Managerが処理する全てのデータは、全ての暗号化においてFIPS 140-2準拠の暗号化モジュールを使用します。これには、OCSPリクエストなど、TLSによる下位動作も含まれます。

項目(2)は、保存されている暗号化オブジェクトの操作をTraffic Manager設定の一部と見なします。PKI証明書や秘密鍵などが主な例に挙げられます。これらのオブジェクトとの暗号化相互作用は、全てFIPS 140-2準拠暗号化モジュールにより行われます。

Traffic Manager FIPS境界には、「コントロールプレーン」の暗号化は一切含まれません。具体的には、以下に挙げる機能はTraffic Manager FIPS境界に含まれません(ただし、完全に網羅しているわけではありません)。

- Traffic ManagerSNMP相互作用
- Traffic ManagerSOAP API
- Traffic Manager REST API
- Traffic Manager管理UI

- Traffic ManagerアプライアンスへのSSHアクセス

境界において(2)が重要である理由は、例えばSOAPまたは管理UIの場合、転送に使用される暗号化にFIPS 140-2準拠暗号化モジュールが使用されていなくても、Traffic Manager FIPS境界内での使用のために暗号化オブジェクトを操作する可能性のある全ての要求された動作が、FIPS 140-2準拠暗号化モジュールによって行われることにあります。

FIPSモードがご使用のデプロイ要件に適しているかどうかを検討するときには、Traffic Manager FIPS境界の機能的な対象範囲が十分であるか検討する必要があります。

例外としてのTLS

前述したように、境界内で行われる全ての暗号化動作がFIPS 140-2暗号化モジュールを使用して行われるというのがTraffic Manager FIPS境界の一般的定義です。

ただし、MD5がTLS 1.0ハンドシェイクプロトコルの要件の一部である場合、これには例外があります。MD5を使用しない場合、最新バージョンのTLSのみを要求するTLSクライアントと相互運用できなくなります。このケースは、前述のドキュメント『Guidelines for the Selection and Use of Transport Layer Security (TLS) Implementations (トランスポート層セキュリティ(TLS)実装の選択および使用に関するガイドライン)』に記された考察に該当します。それによると、TLSハンドシェイクプロトコルのこのような極めて特定の部分におけるMD5の特定用途は、安全であると見なされます。技術的な詳細については、このドキュメントを参照してください。

このため、通常MD5はFIPS 140-2準拠暗号化モジュールから利用可能ではありませんが、このような特定のケースにおいてのみ、Traffic ManagerのSSL実装では、別の暗号化実装からのMD5アルゴリズムが使用されます。

同様に、別の用途においてもTraffic Manager FIPS境界内でMD5アルゴリズムが使用されますが、それらの用途は暗号化またはセキュリティに関する目標には無関係であると考えられています(例えば、暗号化ダイジェストではなくハッシュ関数としてMD5アルゴリズムが使用されます)。

FIPSモードのデプロイ

準備

FIPSモードの設定は、その他の特定の設定に関する前提条件が満たされている場合に限り行うことができます。その前提条件は、トラフィックを処理するためにFIPS 140-2準拠暗号化モジュールを使用するための適切な環境を整えるために満たす必要があります。

そのため、FIPSモードを有効化するための準備において、以下の設定要素をチェックし、必要に応じて更新してください。

サポートされているSSL/TLSバージョン

SP 800-52r1のガイドラインでは、FIPS 140-2準拠暗号化モジュールを使用する環境での動作に適しているのは、TLS 1.0およびそれ以降のSSLバージョンのみとされています。

したがって、FIPSモードを有効化するためには、Traffic Manager設定でSSLv3が無効化されていることを確認してください。これは、**System** (システム) > **Global Settings** (グローバル設定) > **SSL Configuration** (SSL設定) ページで「**ssl!support_ssl3**」を「No (いいえ)」に設定することにより無効化されます。さらに、仮想サーバおよびプールがグローバルSSLをオーバーライドしないことを個々のレベルで確認してください(**Virtual Server** (仮想サーバ) > **Edit** (編集) > **SSL Decryption** (SSL復号化) および **Pool** (プール) > **Edit** (編集) > **SSL Encryption** (SSL暗号化) ページの **ssl_support_ssl3** 設定を使用します)。

注記

ご使用のSSL仮想サーバクライアントおよびプールノードがTLS 1.0またはそれ以降をサポートしていない場合、それらはSSLを使用して接続できなくなります。

SSL3暗号スイートの選択

FIPS 140-2の要件には、FIPS 140-2準拠暗号化モジュールに含めることができる承認済みの暗号化手法のリストが提示されています。逆に、承認されていない手法は暗号化モジュールでは許可されていません。

SSL/TLS暗号スイートの中には、FIPS 140-2による承認を得ていない暗号化手法が含まれています。例えば、RC4ストリーム暗号やMD5ダイジェストアルゴリズムがこれに該当します。FIPSモードにおいて、全てのTLS暗号化がFIPS 140-2準拠暗号化モジュールを使用して行われるようにするために、Traffic Managerとの使用に向けて設定できる暗号スイートのリストが以下のセットに減らされています。

- SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- SSL_RSA_WITH_AES_128_GCM_SHA256
- SSL_RSA_WITH_AES_128_CBC_SHA256
- SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- SSL_DHE_RSA_WITH_AES_128_GCM_SHA256

- SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- SSL_DHE_RSA_WITH_AES_128_CBC_SHA256
- SSL_RSA_WITH_AES_128_CBC_SHA
- SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA
- SSL_DHE_RSA_WITH_AES_128_CBC_SHA
- SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- SSL_RSA_WITH_AES_256_GCM_SHA384
- SSL_RSA_WITH_AES_256_CBC_SHA256
- SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- SSL_DHE_RSA_WITH_AES_256_GCM_SHA384
- SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- SSL_DHE_RSA_WITH_AES_256_CBC_SHA256
- SSL_RSA_WITH_AES_256_CBC_SHA
- SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA
- SSL_DHE_RSA_WITH_AES_256_CBC_SHA

また、TLSv1.3では以下も含まれます。

- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384

注記

これは、SSL仮想サーバクライアントおよびプールノードとの相互運用性に影響します。SSL仮想サーバクライアントおよびプールノードがFIPS 140-2準拠のいずれの暗号スイートもサポートしていない場合、それらはSSLを使用して接続できなくなります。

TrafficScript関数

TrafficScriptを使用することで、Traffic Managerで実行されるルールの多くの暗号化動作にアクセスできます。FIPSモードで動作しているときは、これらの動作はFIPS 140-2準拠暗号化モジュールにより行われます。

FIPS 140-2準拠暗号化モジュール用に承認されていない暗号化手法を使用するTrafficScript関数の場合、FIPSモードでの動作では、それらの関数は適切に実行されません。以下に挙げるTrafficScript関数を使用している場合は、FIPS 140-2準拠暗号化モジュール向けに承認されている別の暗号化ダイジェスト実装に移行することをお奨めします。

FIPSモードで使用するとエラーを発生させるTrafficScript関数は、MD5アルゴリズムを使用するTrafficScript関数です。

FIPSモードで無効な関数	代用に推奨される関数
<code>string.hashMD5()</code>	<code>string.hashSHA1()</code>
<code>ssl.serverCertHash()</code>	<code>ssl.serverCertSHA1()</code>
<code>ssl.clientCertHash()</code>	<code>ssl.clientCertSHA1()</code>

PKI証明書

公開鍵基盤(PKI)証明書および鍵は、FIPS 140-2準拠のデプロイの一部と見なす必要がある、暗号化アセットです。



FIPSモードを有効にすると、Traffic Managerでは、RSAまたはECDSA公開鍵/秘密鍵のみの使用がサポートされます。DSA鍵はサポートされません。

鍵強度

FIPS 140-2のドキュメントでは直接言及されていませんが、SP 800-131Aでは、暗号化動作で使用される鍵の許容強度に関する一連の要件が取り扱われています。一般に、これらの要件はFIPS 140-2準拠暗号化モジュールによって強化されるため、FIPSモードを有効化する前に検討する必要があります。この文脈における「鍵強度」の実際の定義についての公式の説明は、上記の「FIPS 140-2」のセクションで説明したSP 800-51パート1の刊行物に詳述されています。

署名アルゴリズム

証明書によって示される鍵の強度のほかに、証明書自体にも暗号化シグネチャが含まれています。これにより、発行者によって認証された証明書と同じ証明書であることが立証されます。

暗号化シグネチャを作成および認証するために使用されるアルゴリズムに対しては、FIPSモードにおいてFIPS 140-2準拠暗号化モジュールで 사용할 ことが承認されていなければなりません。FIPSモードでは、署名ダイジェストがFIPS承認済みのSHA-1、SHA256、SHA384、およびSHA512アルゴリズムを使用する証明書のみが使用可能です。

Traffic ManagerのSSL設定

FIPSモードを有効化する前に、以下に関して、Traffic Manager SSLカタログ(**Catalogs** (カタログ) > **SSL**から確認)を監査する必要があります。

- 1024ビット未満のモジュラス値を備えた証明書および鍵
- 未承認のダイジェストアルゴリズムを使用する署名を備えた証明書(通常、MD2およびMD5)

この方法でFIPSモードを有効化しようとする、現在の設定の中で動作の進行の妨げになる全ての証明書が検証エラーによってハイライトされます。

代わりに、インストールされたTraffic Managerに含まれているコマンドライン「cert」ツールを使用して、個々の証明書を評価することができます。例えば、SSLサーバ証明書の「example」証明書ファイルをチェックするには、以下のコマンドを使用します。

```
$ZEUSHOME/admin/bin/cert --format text \  
--in $ZEUSHOME/zxtm/conf/ssl/server_keys/example.public
```

このような「脆弱なアセット」が特定された場合は、Traffic Managerデプロイにおけるそのアセットの目的を評価し、本当にそのアセットが必要なのか、またそのアセットは、FIPSモードの有効化を目的にTraffic Managerデプロイを適切に動作させるうえで変更/更新/廃止されるその他のシステム/サービスにとって必要なのかを判断する必要があります。FIPSモードを有効化するためには、まずこのような「脆弱なアセット」を取り除く必要があります。

注記

証明書および秘密鍵を削除すると、Traffic Managerデプロイとの接続性に影響することがあります。例えば、SSLによってプール内のノードに接続する際に、SSLクライアントが「脆弱な」CAによって署名された証明書の提示を求められたり、Traffic Manager自体が特定の証明書の提示を求められたりすることがあります。このような場合は、「脆弱なアセット」を取り除くことによってクライアントとTraffic ManagerがSSLコネクションを行わなくなります。

デプロイの検討事項

これまでに説明した事項に加えて、Traffic Managerを動作させるデプロイについても検討する必要があります。

FIPSモードで動作中、Traffic Managerは暗号化「脆弱アセット」を処理するリクエストを、それがローカルの設定によるものか、リモートシステムから受信したものを問わず拒否します。

注記

SSLによってTraffic Managerに接続する際、またはTraffic ManagerがSSLによってプールノードに接続しているときに、仮想サーバクライアントに証明書を提示させようとする場合、仮想サーバクライアントおよびプールノードが提示する証明書はFIPSモードでの動作に適したものでなければなりません。

証明書チェーン

証明書チェーンを使用して動作しているとき、Traffic Managerがその証明書チェーンをFIPSモードで適切に使用するためには、チェーン内の全ての証明書が「脆弱」であってはなりません。

鍵交換強度

暗号スイートのなかには、ディフィーヘルマン鍵交換アルゴリズムを使用するものもあります。SP 800-131Aには鍵の許容強度に関する一連の要件も含まれており、その要件はディフィーヘルマンアルゴリズムにも適用されます。

Traffic Managerが使用する有限フィールドディフィーヘルマン鍵合意の中の法数のサイズは、**ssl!diffie_hellman_modulus_size**設定キー(**System** (システム) > **Global Settings** (グローバル設定) ページの「SSL Configuration (SSL設定)」セクション)を使用して設定することができます。

有効性の確保/維持

FIPS 140-2準拠暗号化モジュールの有効性は、認定を受けたテスト研究所が実施するテストによって定義されます。ソフトウェア暗号化モジュールに対するテストは、特定のハードウェア/OSプラットフォーム(運用環境)で実施されます。FIPS 140-2準拠暗号化モジュールが認証済みの運用環境で使用されている場合、そのモジュールはFIPS 140-2認証済みであると見なすことができます。

RCSMが認証されている運用環境の詳細については、NISTのWebサイトに掲載されている一覧で確認できます。

テストが、限定された数の運用環境において実施された場合(さらに、その環境で使用されるRCSMがFIPS 140-2に対する明示的な有効性を備えている場合)、認証済みのFIPS 140-2準拠暗号化モジュール(レベル1)は、(FIPS 140-2に対して)明示的に認証されていない環境で運用する場合でも、その有効性は失われないと理解されています。

FIPS 140-2に関する実施要項(<http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf>)では、G.5のセクションでこの件が取り扱われています。そこでは、認証済みとして明示的にリストされていないプラットフォームにおいてFIPS 140-2に対する有効性を維持するために、ベンダー(RCSMの場合はRiverbed)またはユーザ(Traffic Manager製品の管理者など)が従わなければならない「移植」ルールが取り扱われています。この場合も、FIPS 140-2準拠暗号化モジュールが有効であるかどうかを判断する総合的な裁定者はNIST (具体的にはCVMP)であることに注意してください。

注記

ご使用のTraffic Manager製品のRCSMで、FIPS 140-2に対する有効性が維持されているかどうかについてご不明な点があれば、サポート窓口にお問い合わせください。

Linux以外のOS

RCSMの有効性は、Linux OSのバージョンに対してのみテストされています。Linux以外のシステムに対するサポートおよび有効性は提供されていません。

FIPSモードの有効化

前述の方法に従ってTraffic Managerデプロイを準備すると、管理UIの**System** (システム) > **Global Settings** (グローバル設定) ページにアクセスし、**fips!enabled** 設定キーを**Yes** (はい) に設定することでFIPSモードを有効化できます。



fips!enabled は、ロケーションごとに変更できないクラスタ全体の設定です。

Traffic Managerのリスタート

Traffic Managerを完全にFIPSモードで動作させるためには、**fips!enabled** キーを設定するだけでは十分とは言えません。クラスタ内の各Traffic Managerインスタンスをリスタートさせる必要があります。

注記

Traffic Managerをリスタートすると、アクティブなコネクションが全て切断されます。デプロイの可用性を高め、サービスの中断を最低限に抑えるために、この操作をスケジュールして実行することをお奨めします。詳細については、ご使用の製品の『*Pulse Secure Virtual Traffic Manager: Installation and Getting Started Guide*』を参照してください。

バリデーション

FIPSモードを有効化してから管理UIまたはSOAPベースのコントロールAPIを使用する場合、Traffic Managerはご使用の設定に対するバリデーションを実行し、FIPSモードに対する様々な基準が満たされているか確認します。

FIPSモードと競合する設定要素があれば、全てエラーとして報告されます。そのようなエラーが発生した場合は、上記の「準備」のセクションで確認したエラーに関連する記載を参照してください。

FIPSモードでの動作

FIPSモードで動作しているとき、クラスタ内の各Traffic Managerは、FIPSモードでの動作に異常がないかについてイベントログで報告します。以下のようなメッセージが記録されます。

```
FIPS 140-2 Cryptographic module loaded: FIPS Mode in operation
```

FIPSモードのエラー

FIPSモードを有効化しようとしているときにエラーが発生した場合、Traffic Managerがイベントログでそのエラーを報告します。Traffic ManagerコンポーネントでFIPSモードの障害が発生した場合、そのコンポーネント内の暗号化動作(およびその動作を使用するサービス)は無効化されます。

例えば、1台はSSL復号化あり、もう1台はSSL復号化なしという2台のHTTP仮想サーバを備えたTraffic Managerがあるとします。FIPSモードの初期化が正しく行われなかった場合、SSL復号化仮想サーバは無効化され、一切のトラフィックは処理されなくなりますが、もう一方の仮想サーバは正常に動作し続けます。

注記

FIPSモードが動作した状態でTraffic Managerが適切に起動したとしても、まれではありますが、動作中にFIPS 140-2の要件に対する適合性が失われることがあります。このような場合、初期化の失敗時と同じ動作が想定されます。つまり、初期化に失敗したTraffic Managerからエラーが報告され、FIPSモードで障害が発生した場合は、暗号化動作が実行不能になります。

Kerberos制約付き委任のサポート

本章では、Traffic ManagerによるKerberos制約付き委任のサポートについて説明します。

Kerberosプロトコル

Kerberosプロトコルを使用することで、エンティティ(「プリンシパル」とも呼ばれる)は、広く知られた信頼できる鍵配布センタ(KDC)サービスによって互いに認証し合うことができます。あるレルム用のKDCは、そのアドレス系に関連付けられたプリンシパルを、そのプリンシパルによって提示されたクレデンシャルを検証することによって認証します。その結果、レルムで機能するプリンシパルどうしが交換できるチケットが生成され、認証されたピアツーピアの情報交換を確立することができます。KDCがユーザに対して発行するプリンシパルへのチケットには、プリンシパルとKDCのみによって共有される秘密を使用して暗号化された、そのユーザに対する識別データが保持されています。



Microsoft®によるKerberosプロトコルの実装では、レルムは「ドメイン」と呼ばれ、KDCサービスは「ドメインコントローラ」と呼ばれています。

Kerberosプロトコルの遷移と制約付き委任

クライアントのプリンシパルが一部を成すレルムのKDCに、クライアントが直接アクセスできる場合、「Kerberos対応」のアプリケーション間のKerberos認証は正常に機能します。しかし、KDCへ直接アクセスできない場合、ユーザは、認証のためにKerberosチケットの交換を要求するサービスに必ずしもアクセスできるとは限りません。例えば、パブリックインターネットに単独で接続されているラップトップコンピュータは、ファイアウォールのルールによって、ラップトップユーザのKDCに直接アクセスできないことがあります。このような事態に対処するために、プロトコル遷移と制約付き委任という、Kerberosプロトコルに対する2つの拡張機能が用意されています。

プロトコル遷移

これは、適切に特権が与えられたKerberosサービスが、特定のレルムの任意のユーザプリンシパル用にそのサービス自体へのチケットを取得できるようにするメカニズムです。KDCは、Kerberosプロトコルでのユーザ用のチケットを生成する前に、その他の手段によって認証を実施し、ユーザの身元を確認することをサービスに対して要求します。すなわち、サービスは1つの認証プロトコルからKerberosへの遷移を提供します。

制限付き委任

プロトコル遷移を提供しているサービスでは、ユーザの身元確認はできているため、そのサービスに対する任意のユーザプリンシパル用のチケットを持つことの効果は限定的です。制限付き委任を使用すると、サービスは、そのサービスに対するチケットを別のサービスに委任することができます。つまり、そのチケットをそのサービスから(元のユーザに関連付けられたリクエストとともに)さらに先のKerberizedサービスに転送できるようになります。この新しいサービスは、転送されたリクエストを元のユーザに照らして処理します。

レルム内の任意のユーザになりすませることは、サービスにとって強力すぎる権限に思われるかもしれませんが、この機能には「制限付き」という側面があるため、KDCは、チケットが委任されるサービスの選択において制限を課すことができます。

Kerberosを使用する論理的根拠

Traffic Managerとの関連において、Kerberosプロトコル遷移は、Traffic Managerが内部ネットワークの境界上に位置し、内部サービスへのアクセスを提供している場合に活用できます。Kerberosを使用して認証を行っている社内LAN上の内部サービスは、KDCへのアクセスが許可されていない、よりパブリックなネットワークで利用することができます。Traffic Managerは、代替手段を使用してユーザのプリンシパル名を特定し、プロトコル遷移を行います。

Traffic Managerは、TLSクライアント証明書ベースの認証からのプロトコル遷移をサポートしています。

Traffic ManagerでのKerberos設定

Traffic ManagerがKerberosをサポートしていることによって、TLSクライアント証明書を使用したユーザの認証が可能になります。Traffic Managerは、それ自体のサービスプリンシパルを使用してレルムのKDCと通信を行い、仮想サーバでリクエストを行うクライアントに代わって、バックエンドノードで実行されているサービスへのチケットを取得します。

Traffic Managerのサービスプリンシパル

Kerberosのレルムに参加するには、Traffic Managerに、自らを特定するために使用する、適切に権限が与えられたプリンシパルが備わっていなければなりません。ご使用の身元管理システムおよび関連付けられたKDC設定には、Traffic Manager用のKerberosプリンシパルが備わっていることを必ず確認してください。また、Traffic Managerが管理しているサービスへのプロトコル遷移および制限付き委任が許可されているように設定する必要もあります。

クラスタ内の各Traffic Managerに対して、KDC管理者は、Traffic Managerのサービスプリンシパルが他のサービスに委任できるようにKDCを設定する必要があります。また、サービスプリンシパルの名前は、Traffic Managerが使用する名前と一致しなければなりません。これは、以下の形式の文字列で記します。

<service>/<Traffic Manager hostname>@<REALM>

この文字列の<service> および<REALM>の値には、Kerberosプリンシパルカタログから使用したいKerberosプリンシパルレコードの値を使用します。

例えば、Traffic Managerのホスト名が「stm1.example.com」で、サービス名が「HTTP」、レルムが「EXAMPLE.COM」のKerberosプリンシパルレコードを使用して接続する場合、KDCは「HTTP/stm1.example.com@EXAMPLE.COM」と指定されたプリンシパルを信頼し、エンドサービスへの委任を行います。

Microsoft実装では、このことは、Traffic Managerに対するコンピュータアカウントがドメイン上に作成され、Traffic Manager上の関連するサービスにサーバプリンシパル名が設定され、それらのサービスに対してTraffic Managerへの委任が許可されたことを意味します。

Traffic Managerが、作成されたプリンシパルとしてTraffic Manager自体を認証するためには、適切なクレデンシャルを指定する必要があります。ユーザプリンシパルに対しては、パスワード入力の形式でクレデンシャルを指定します。インタラクティブなパスワード入力はサービスプリンシパルとして不便なため、代わりに「Keytab」ファイルをTraffic Managerサービスプリンシパルのクレデンシャルに使用しなければなりません。Keytabファイルには、1つ以上のプリンシパルに対するクレデンシャルを保持させることができます。

Kerberosのレルムでプロトコル遷移を実施するようにTraffic Managerを設定するには、クラスタ内の全てのTraffic Managerに対するクレデンシャルを保持するKeytabファイルをアップロードし、Keytabファイルを使用するようにKerberosプリンシパルを設定します。

KeytabファイルはTraffic ManagerのKeytabカタログに保存されます。Keytabファイルをカタログにアップロードするには、管理UIの**Catalog** (カタログ) > **Kerberos** > **Kerberos Keytabs**ページを使用します。

Catalogs: Locations DNS Server GLB Services Rules Java Web Accelerator Monitors SSL Authenticators

Kerberos > Kerberos Keytabs SAML Protection Persistence Bandwidth SLM Rate Cloud Credentials Extra Files

Kerberos Keytab Files

Kerberos keytab files for authenticating configured Kerberos principals.

Name	Size	Select (all / none)
krbtabfile Rename	9 bytes	☐
krbtabfile-2 Rename	9 bytes	☐

☐ Confirm operation

Upload Kerberos Keytab File

Upload a file to the keytabs directory.

File: No file chosen

Kerberosプリンシパルのカタログ項目を作成するには、管理UIの**Catalog** (カタログ) > **Kerberos** > **Kerberos Principals** (Kerberosプリンシパル)ページを使用します。各プリンシパルに対して、以下のフィールドを指定します。

設定項目	説明
name	プリンシパル設定の識別名です。
service	Traffic ManagerがTraffic Manager自体を認証するために使用するKerberosプリンシパル名のサービス名要素です。
keytab	このプリンシパル用のクレデンシャルが保持されるKerberos KeytabsカタログのKeytabファイルの名前です。
realm	Kerberosプリンシパルが一部を成すレルムです。
kdc	レルムに使用するKDCサービスを明確に指定するための、<host>:<port>のペアのオプションリストです。リストを指定しない場合、Traffic ManagerはDNSを使用して代わりに使用すべきKDCを特定します。
krb5conf	このプリンシパルと使用する、あらかじめ定義されたKerberos設定ファイルの名前です(オプション)。Kerberos設定ファイルを使用するには、まず管理UIの Catalog (カタログ) > Kerberos > Kerberos Configuration Files (Kerberos設定ファイル)ページを使用して、設定ファイルをアップロードする必要があります。

仮想サーバプロトコル遷移の設定

仮想サーバでKerberosプロトコル遷移を使用するには、**Virtual Servers** (仮想サーバ) > **Edit** (編集) > **Kerberos Protocol Transition** (Kerberosプロトコル遷移)ページを使用します。

Configuring: **Traffic IP Groups** **Virtual Servers > Web Traffic > Kerberos Protocol Transition** **Pools** **Config Summary**

Kerberos Protocol Transition

Virtual Server: Web Traffic (HTTP, port 80)

Kerberos Protocol Transition allows clients to use services that use Kerberos authentication without having Kerberos tickets themselves.

Kerberos Protocol Transition settings

These settings control Kerberos Protocol Transition for this virtual server.

Whether or not the virtual server should use Kerberos Protocol Transition.
kerberos_protocol_transition!enabled: ☐ Yes ☒ No

The Kerberos principal this virtual server should use to perform Kerberos Protocol Transition.

	Name	
kerberos_protocol_transition!principal:	☒ Not specified	
	☐ KCDP1	Edit
	☐ KCDP2	Edit

The Kerberos principal name of the service this virtual server targets.
kerberos_protocol_transition!target:

Apply Changes

[Update](#)

Kerberosを有効化するには、**kerberos_protocol_transition!enabled**をYes (はい)に設定します。このページを使用して、オプションでKerberosプリンシパルと、仮想サーバが処理するターゲットのサービスに対するプリンシパル名を設定できます。

i プロトコル遷移が適切に実行されるためには、Traffic Managerとターゲットのプリンシパルの両方を設定する必要があります。これらの設定は、仮想サーバまたはプールのいずれかで行います。

Traffic Managerは、クライアントSSL証明書からクライアントプリンシパルを取得する必要があります。これを可能にするには、仮想サーバで**require_cert**と併せて**ssl_decrypt**も有効化し、クライアントにクライアント証明書を求めるように設定します。クライアントが適切に認証されるためには、適切な認証局とOCSPを仮想サーバに設定してください。

プールプロトコル遷移の設定

Pool (プール) > Edit (編集) > Kerberos Protocol Transition (Kerberosプロトコル遷移)ページを使用して、Kerberosプロトコル遷移を設定することもできます。

Configuring: **Traffic IP Groups** **Virtual Servers** **Pools > Web Traffic > Kerberos Protocol Transition** **Config Summary**

Kerberos Protocol Transition

Pool: Web Traffic

Kerberos Protocol Transition allows clients to use services that use Kerberos authentication without having Kerberos tickets themselves.

Kerberos Protocol Transition settings

These settings control Kerberos Protocol Transition for this pool.

The Kerberos principal the traffic manager should use when performing Kerberos Protocol Transition.

	Name	
kerberos_protocol_transition!principal:	☒ Not specified	
	☐ KCDP1	Edit
	☐ KCDP2	Edit

The Kerberos principal name of the service this pool targets.

kerberos_protocol_transition!target:

Apply Changes

[Update](#)

このページでプリンシパルまたはターゲットを指定すると、この設定が、このプールを使用する仮想サーバに存在する全てのKerberos設定をオーバーライドします。

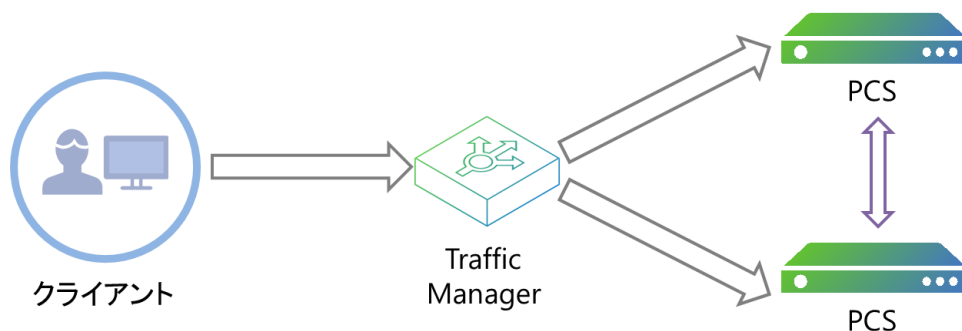
Pulse Connect Secureサービスの負荷分散

本章では、VPN接続をPulse Connect Secure (PCS)サービスで負荷分散するための、Traffic Managerの設定方法を説明します。また、地理的に離れた複数のPCSクラスタのデプロイ時に、Traffic Managerを設定して最適なゲートウェイの選択と動的フェールオーバーの接続を行う方法についても説明します。

PCS負荷分散の設定の概要

着信ユーザVPNセッションを一連のPCSインスタンスに分散するようTraffic Managerを設定できます。Traffic Managerはさらに、PCSインスタンスの健全性を監視して、PCSデプロイに負荷分散およびフェールオーバー機能を与えます。

操作を単純にするため、Traffic Managerには、「Load-balance Pulse Connect Secure (Pulse Connect Secureの負荷分散)」ウィザードがあり、提供された詳細に基づいて、自動的にTraffic Managerサービス設定を作成します。



注記

ウィザードはTraffic Manager外の設定は何も実行しません。ウィザードを実行する前に、PCSインスタンスのセットが完全に設定されていることを確認します。これらのインスタンスは、アクティブ/アクティブ設定にクラスター化したり、何らかの他の方法で設定を共有したりすることができます。詳細については、IvantiのWebサイト (www.ivanti.com)にあるPulse Connect Secureのドキュメントを参照してください。

ウィザードを使用するには、以下の情報が必要です:

- 新しいサービスの識別名。

- 新しいサービスが監視すべき全てのIPアドレスを含むすでに作成済みのトラフィックIPグループ。
- Traffic Managerで使用するUDPポート番号(PCSインスタンスでEPSモードが設定されます)。

i 本ガイドの画像およびリファレンスでは、デフォルト値の4500を前提としています。

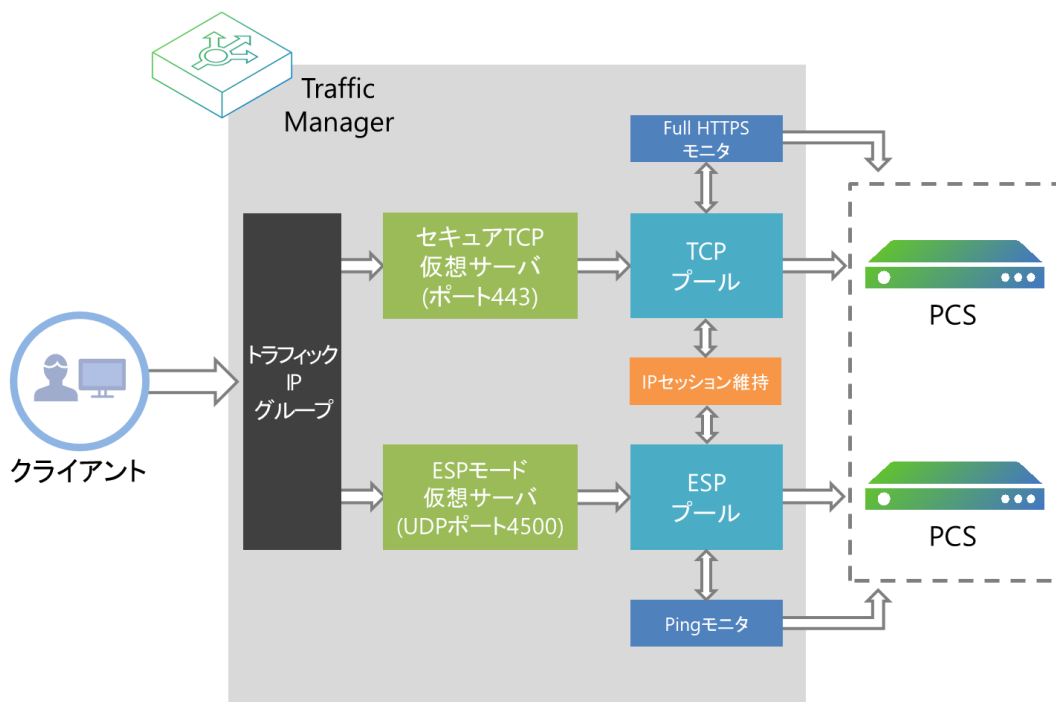
- PCSインスタンスのホスト名またはIPアドレス。

さらに、Traffic Managerで以下の機能を実行するかどうかを決定します。

- HTTPリクエストをHTTPSにリダイレクトする。
- IPトランスペアレンシを使用する。

i ウィザードを実行する前に、使用するトラフィックIPアドレスでポート443または選択されたUDPポートを使用している既存のサービスがないことを確認してください。

PCS負荷分散の設定のトポロジ



Traffic Managerは、一般にトラフィックIPグループ内の1つ以上のトラフィックIPアドレスを使用して、公開されたVPMサービスをホストします。クライアントがVPNセッションの確立を試みる際は、最初にセキュアTCP制御接続を作成します。この接続で認証後、クライアントはセキュアUDPチャネル上でESPTraフィックの送信を試みます。このようにして、1つはセキュアなTCPトラフィック用、もう1つはUDPストリーミングトラフィック用という、サービスのペアを使用して両方のタイプのトラフィックを受信するよう、ウィザードでTraffic Managerが設定されます。Traffic Managerは、同じPCSインスタンスをノードとして含む各プールである、各プロトコルタイプ用に別個のプールを維持します。この方法で、Traffic Managerは、同じクライアントから発して同じPCSインスタンスに向かうTCPおよびUDP接続の両方を負荷分散できます。

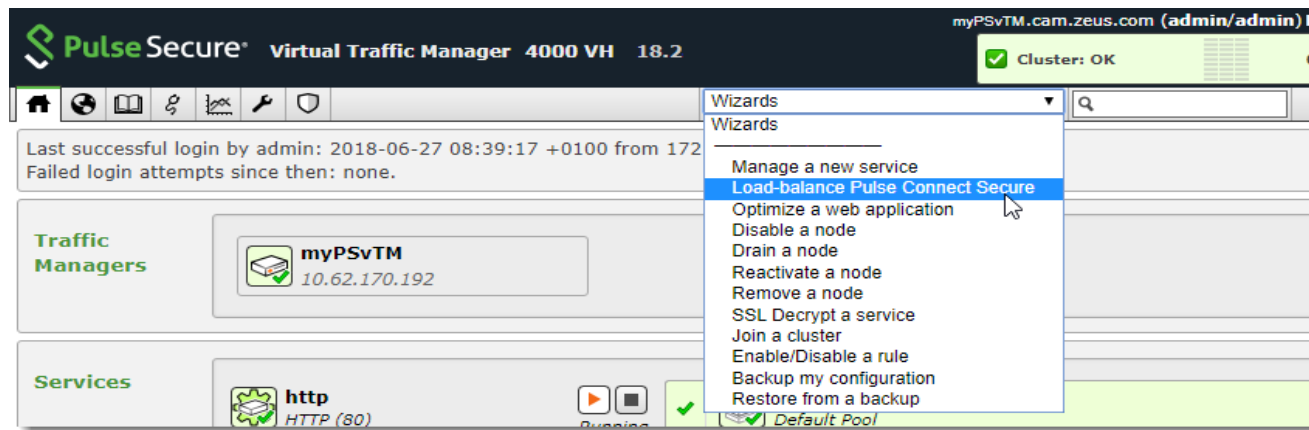
クライアントとPCSサーバの間でセキュアUDPチャネルを確立できない場合、クライアントはVPNTraフィックに対してTCP接続を使用するようフォールバックします。

VPNTraフィックが対応する制御接続を処理する同じPCSインスタンスに必ず送信されるようにするために、ウィザードはIPベースのセッション永続性を両方のプールに設定して適用します。

最後に、Traffic Managerが効果的な負荷分散決定を必ず実行できるように、ウィザードで両方のプールに適切なヘルスマonitoringを設定します。

Load-balance a Pulse Connect Secure (Pulse Connect Secureの負荷分散)ウィザードの実行

ウィザードを実行するには、ツールバーで「Wizards (ウィザード)」ドロップダウンメニューをクリックしてから、「Load-balance Pulse Connect Secure (Pulse Connect Secureの負荷分散)」を選択します。



Traffic Managerにウィザードの最初のページが表示されます。

Load-balance Pulse Connect Secure, step 1 of 8

1. Load-balance Pulse Connect Secure

This wizard will guide you through the process of configuring the traffic manager to load-balance VPN connections to a Pulse Connect Secure service.

The wizard requires you to input a hostname or IP address for each Pulse Connect Secure instance.

If the service is to listen on a Traffic IP address, then an appropriate Traffic IP group should be created before commencing this wizard.



Cancel

◀ Back

Next ▶

Next (次へ)をクリックして操作を続けます。



それぞれのウィザードのページで、**Cancel** (キャンセル)を使用すると、Traffic Managerを変更せずにウィザードがキャンセルされます。**Back** (戻る)を使用すると前のページに戻り、**Next** (次へ)を使用すると次のページに進むことができます。

Load-balance Pulse Connect Secure, step 2 of 8

2. Name service

Enter a name to identify the service that is to be load-balanced. All configuration created by this wizard will be prefixed with the name entered here.

Name:

Traffic Managerはここで指定する識別子を、このプロセスで作成される全ての設定オブジェクトのプレフィックスとして使用します。

識別名を入力し、**Next** (次へ)をクリックして操作を続けます。

Load-balance Pulse Connect Secure, step 3 of 8

3. Listen on address

Select on which IP addresses or Traffic IP groups the service should listen:

- ☒ All IP addresses
- ☐ Traffic IP Groups ...

このページを使用して、Traffic Managerがホストする全てのIPアドレスをVPNサービスに監視させるか、既に定義済みのトラフィックIPグループを使用するかを決定します。

Traffic Managerのフロントエンドネットワークインターフェースに割り当てられたIPアドレスが、VPMサービスの公開での使用に適していないことがあります。Traffic Managerクラスターでハードウェアやシステムに障害が発生した場合に、サービスの一部または全部が利用できなくなるためです。

Traffic Managerのフォールトトレランス機能を利用すると、トラフィックIPアドレスを設定することができます。これらのトラフィックIPアドレスは特定のTraffic Managerインスタンスに紐付けられないため、一部のTraffic Managerインスタンスで障害が発生しても、クラスターでは各IPアドレスは完全に利用可能な状態に保たれます。

トラフィックIPアドレスはトラフィックIPグループに整理できます。グループは、Traffic Managerインスタンスの一部または全てにまたがるように定義します。グループのメンバーはお互いの間でネゴシエーションを行い、トラフィックIPアドレスを共有し、各Traffic Managerで自分に割り当てられた1つまたは複数のIPアドレスが有効になります。トラフィックIPアドレスおよびグループについては、[トラフィックIPグループとフォールトトレランス](#)を参照してください。

リストからオプションを選択し、**Next** (次へ)をクリックして操作を続けます。

Load-balance Pulse Connect Secure, step 4 of 8

4. ESP Mode

If your PCS instances are configured to support ESP mode, PCS clients will attempt to send VPN traffic over a separate UDP channel. Configure this port to match the UDP port setting on the PCS instances.

UDP Port:

Traffic Managerは、ここで指定した値を使用して、ESPモードのUDPストリーミング仮想サーバを設定します。指定するポート番号は、PCSインスタンスのUDPポート設定と一致している必要があります。

Load-balance Pulse Connect Secure, step 5 of 8

5. Redirect HTTP to HTTPS

Specify whether the traffic manager should redirect HTTP requests to HTTPS. If enabled, users attempting to connect to PCS over HTTP will be redirected to the secure endpoint.

HTTP Redirect: ☒ Yes
☐ No

Cancel ◀ Back Next ▶

この設定を使用すると、HTTP上で送信されたリクエストをTraffic ManagerがセキュアなHTTPSエンドポイントに必ずリダイレクトするように設定できます。このオプションを有効にする前に、組織のネットワークセキュリティポリシーを確認することをお奨めします。

Next (次へ)をクリックして操作を続けます。

Load-balance Pulse Connect Secure, step 6 of 8

6. Pulse Connect Secure instance addresses

Enter the hostnames or IP addresses of the PCS instances:

Hostname:

PCS instances:

To remove an address from the list, select it and press 'Remove PCS instance':

Cancel ◀ Back Next ▶

このページを使用すると、Traffic ManagerにPCSクラスタを追加できます。クラスタメンバーごとに、ホスト名およびIPアドレスを**Hostname** (ホスト名)フィールドに入力し、**Add PCS instance** (PCSインスタンスの追加)をクリックしてそれをリストに追加します。PCSインスタンスを削除するには、そのリストエントリを選択して、**Remove PCS instance** (PCSインスタンスの削除)をクリックします。

Next (次へ)をクリックして操作を続けます。

Load-balance Pulse Connect Secure, step 7 of 8

7. Pool IP Transparency

Specify whether connections from the traffic manager to the PCS instances should appear to originate from the original client's source IP address.

Note that if IP transparency is enabled, your PCS instances must be configured to route return traffic to the traffic manager cluster. See the Network Layouts section of the **User's Guide** for details on how to configure the traffic manager and PCS instances to enable traffic to be routed correctly when IP transparency is enabled.

IP Transparency: ☒ Yes
☐ No

Cancel ◀ Back Next ▶

従来の負荷分散デプロイによって、Traffic Managerは着信接続を監視し、PCSインスタンス間にそれを分散します。PCSインスタンスでは、着信トラフィックはTraffic ManagerのバックエンドIPアドレスが発信元と見なし、同じアドレスに応答を送信します。その後、Traffic Managerはこの応答をクライアントに渡します。

状況によっては、クライアントIPアドレスをPCSインスタンスを通じて伝搬し、接続がTraffic ManagerのIPアドレスではなく、クライアント自体のIPアドレスから開始されていると、PCSが見なすようにする場合があります。これを**IPトランスペアレンシ**と言います。

IPトランスペアレンシをVPNサービスで有効にするには、**IP transparency** (IPトランスペアレンシ)を「Yes (はい)」に設定します。

注記

IPトランスペアレンシを有効にすると、PCSはリクエストを送信したクライアントのIPアドレス宛に応答を返します。ただし、トランスペアレンシが正しく動作するには、リクエストを送信したTraffic Managerを経由して応答を返すよう、各PCSインスタンスがルーティングする必要があります。そのため、Traffic Managerをデフォルトゲートウェイとして使用するよう、PCSインスタンスを設定する必要があります。IPトランスペアレンシによるトラフィックルーティングの詳細については、[ネットワーク構成の基本コンポーネント](#)を参照してください。

PCSの設定方法の詳細については、IvantiのWebサイト(www.ivanti.com)にあるドキュメントを参照してください。

Next (次へ)をクリックして操作を続けます。

Load-balance Pulse Connect Secure, step 8 of 8

8. Summary

The following configuration will be created to load-balance Pulse Connect Secure:

Virtual servers:

HTTPS Virtual Server:	Cambridge_pcs_https
HTTPS Port:	443
ESP Virtual Server:	Cambridge_pcs_esp
ESP Port:	4500
HTTP Virtual Server:	Cambridge_pcs_http
HTTP Port:	80
Listening on:	All IP addresses

Pools:

HTTPS Pool:	Cambridge_pcs_https
HTTPS Pool Nodes:	192.0.2.0:443, 192.0.2.1:443
ESP Pool:	Cambridge_pcs_esp
ESP Pool Nodes:	192.0.2.0:4500, 192.0.2.1:4500

Using IP Transparency: Yes

Node health monitors:

HTTPS Monitor:	Cambridge_pcs_https
ESP Monitor:	Cambridge_pcs_ping

HTTP Redirect:

Enabled:	Yes
Rule:	Cambridge_pcs_redirect

Session persistence:

Persistence class:	Cambridge_pcs_ip

To create this service, press 'Finish'. To change your settings, press 'Back'.

Cancel

◀ Back

Finish

このページには、提案されたTraffic Manager設定の要約が表示されます。**Cancel** (キャンセル)をクリックすると、変更を行わずにウィザードを終了できます。**Back** (戻る)をクリックすると前のページに戻り、**Finish** (完了)をクリックするとウィザードが完了して、Traffic Managerが設定されます。

ウィザードで全ての設定が完了すると、Traffic Managerのホームページが更新され、実行中の全てのサービスが表示されます。

サービスディスカバリの加重負荷分散

要件に従ってIvanti Connect Secureのリリース9.1R3以降では、Traffic Managerがオプションでサービスディスカバリを使用して、デプロイされた各ICSインスタンスの未使用ライセンス数をクエリすることができます。Traffic Managerはこの情報を加重負荷分散で使用することで、1つのICSインスタンスに過剰にプロビジョニングされることを防止します。

Traffic Managerは、ICSのヘルスチェックAPIを使用して未使用ライセンス数を検出し、よりライセンス容量の多い記録をもつデバイスに対して新しい接続を優先的に振り向けます。

この機能を使用するには、Traffic ManagerのバックエンドIPアドレスからヘルスチェック要求を受け付けるように、あらかじめICSインスタンスを設定しておく必要があります。詳細については、Ivanti Technical Publicationsのウェブサイト (<https://www.ivanti.com/support/product-documentation>) でIvanti Connect Secureのドキュメントを参照してください。

ヘルスチェックAPIの使用を開始するには、内蔵のサービスディスカバリプラグインを使用するようにTraffic Manager Admin UIのICSプールを再設定します。サービスディスカバリのサブセクションを選択して(**Services** (サービス) > **Pools** (プール) > **Edit** (編集) > **Service Discovery** (サービスディスカバリ)をクリック)、以下の必要な設定項目を設定します。

設定項目	設定
service_discovery!enabled	「Yes (はい)」に設定します。
service_discovery!plugin	「builtin-PCS_PPS」に設定します。
service_discovery!plugin_args	HTTPSプールの場合は次の引数を使用します。 --nodes="192.0.2.0:443 192.0.2.1:443" --info ESPプールの場合は次の引数を使用します。 --nodes="192.0.2.0:4500 192.0.2.1:4500" --info --nodes引数を使用すると、「Load-balance Pulse Connect Secure (Pulse Connect Secure の負荷分散)」ウィザードで指定したような、スペースまたはカンマで区切られたICSノードのIPアドレスのリストで置き換えられます。

設定項目	設定
	--info引数を使用すると、相対ノード加重(負荷分散アルゴリズムで使用されるもの)の変化が検出された場合に、Traffic ManagerのイベントログにINFOメッセージが記録されます。このログメッセージが不要な場合はこの引数を省略できます。 ESPプールの場合は、使用するポート番号が「Load-balance Pulse Connect Secure (Pulse Connect Secureの負荷分散)」ウィザードで指定したものと一致している必要があります。

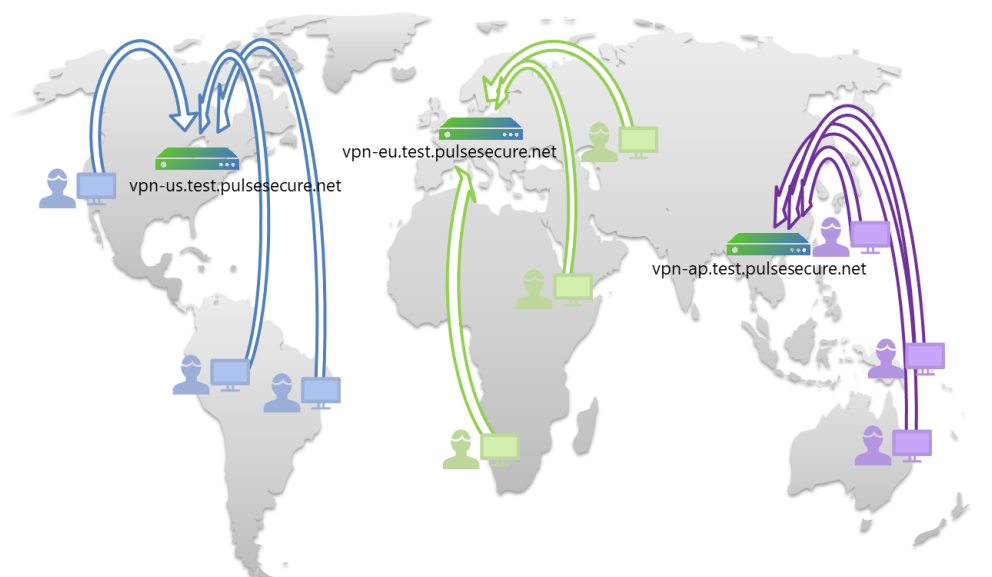
変更を保存するには、**Update** (更新)をクリックします。プール編集ページでノードリストを変更できなくなることにご注意してください。

次に、「加重」負荷分散アルゴリズムを選択します(**Services** (サービス) > **Pools** (プール) > **Edit** (編集) > **Load Balancing** (負荷分散) をクリック)。Traffic Managerの設定で両方のICSプールに対してこの処理を行う必要があります。

サービスディスカバリの詳細については、[サービスディスカバリ](#)を参照してください。

Traffic Managerを使用した最適なゲートウェイの選択

組織は、PCSの複数のインスタンスを、一連のグローバルデータセンタのロケーションにまたがってデプロイし、その地域にいるユーザに対してセキュアなリモートアクセス機能を提供できます。各ロケーションには単体のPCSデバイス、もしくは、単体のデバイスに障害が起こった時でもVPNサービスが利用できるように、冗長構成のPCSデバイスが存在します。しかし、地域全体の中断が発生すると、サービス全体が失われる可能性があります。



さらに、特定の地理的ロケーションについて手動で最適なPCSゲートウェイを決定する必要があります。全てのロケーションの設定とURL設定を、全てのユーザと全てのデバイスについて維持する必要があります。

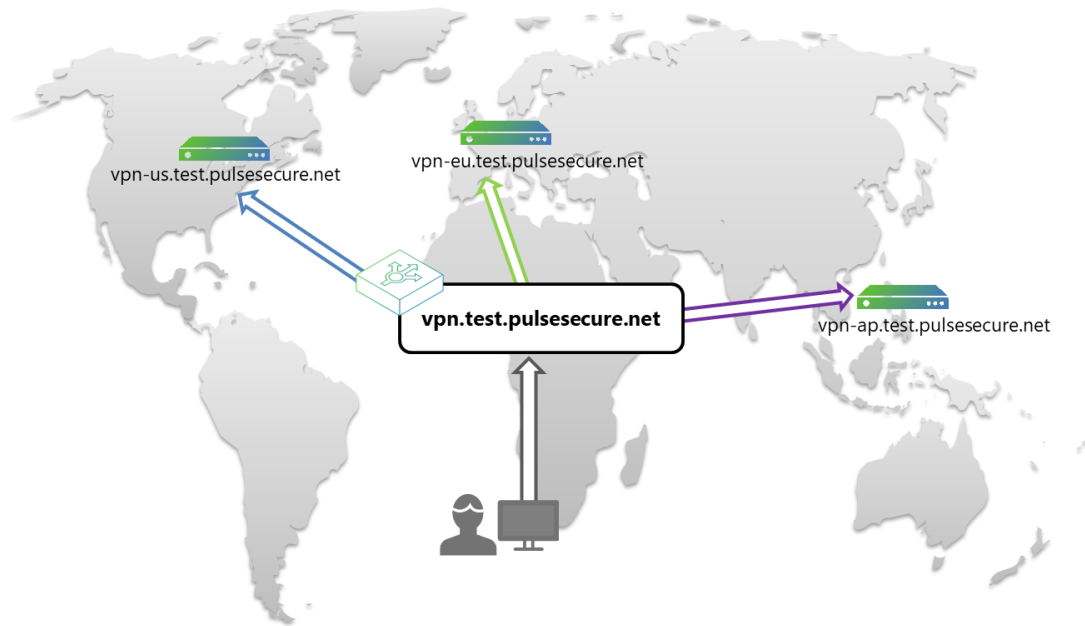
いずれかのロケーションで大規模な障害が発生した場合は、ユーザや管理者が介入し、他のロケーションにある代替PCSゲートウェイへの強制フェールオーバーが必要になります。

この問題を軽減するため、Traffic Managerには「Optimal Gateway Selection」(最適なゲートウェイの選択)ウィザードが用意され、世界中に分散されたPCSゲートウェイサービス用の、単一で、統合された、企業全体のエンドポイントURLを設定できます。Traffic ManagerはDNSベースのグローバルサーバ負荷分散(GSLB)を使用し、観察された一連の条件に基づいて、最適なゲートウェイの選択と動的なフェールオーバーを提供します。

- 最も近い利用可能なPCSゲートウェイに対するユーザの地理的ロケーション
- 選択したPCSゲートウェイの現在の可用性
- 選択したPCSゲートウェイで測定された負荷



GSLB機能を提供するためには、Traffic Managerを承認DNSサーバとして使用する必要があります。Optimal Gateway Selection (最適なゲートウェイの選択)ウィザードでは、個別のバックエンドのDNSサービスの使用はサポートされません。



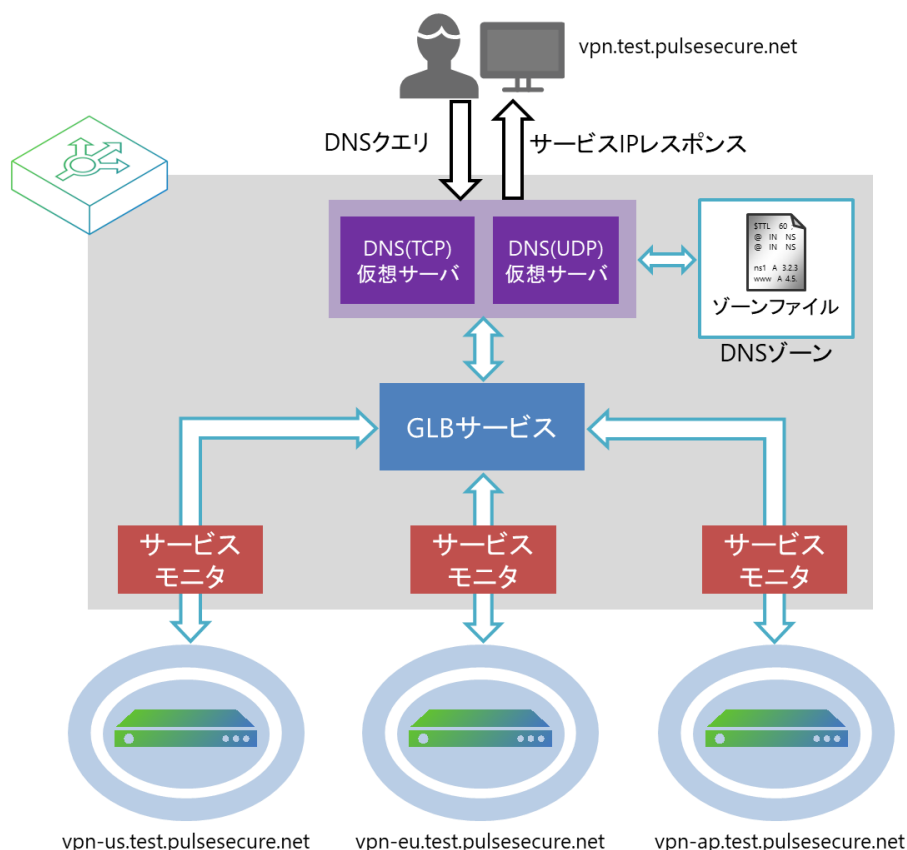
Traffic Managerが、統合エンドポイントURLについてクライアントからDNSリクエストを受け取ると、Traffic Managerはロケーション認知をDNS検索に追加します。そのため、検索の結果はそのソースによって異なります。レスポンスをクライアントに返す前に、設定可能なメトリック(地理的近さや負荷など)に基づいてレスポンスを決定および修正します。

Traffic Managerは継続的に全てのPCSサービスのロケーションの健全性をモニタリングし、サービス障害時にユーザを動作可能なロケーションへと自動でリダイレクトします。

あるロケーションを他のロケーションへ置き換えるなど、後でPCSインフラストラクチャに変更を加えるには、Traffic Managerの設定を更新するだけ済みます。全てのユーザが統合エンドポイントURLを使用し続けることができ、クライアントの設定の更新は不要です。

Optimal Gateway Selection (最適なゲートウェイの選択)ウィザードの使用

Traffic Managerには、世界中に分散されたPCSのインフラストラクチャと、統合VPNエンドポイントの要件についての情報を集めたウィザードがあります。ウィザードではTraffic Managerで必要な設定を行い、以下のトポロジに基づいて完全に機能するGSLBを作成できます。

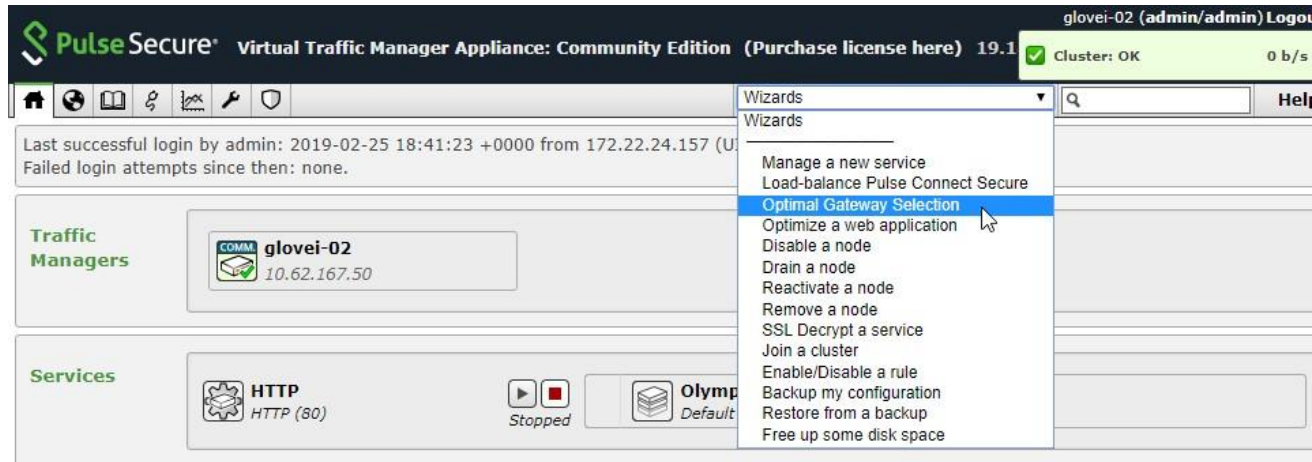


Traffic Managerは、グローバル負荷分散(GLB)機能のような、GSLB技術を実装しています。GLB機能では、以下にリストした多くのコンポーネントが使用されます。

- **DNS仮想サーバ:** デュアル仮想サーバはTCPとUDPの両方を介してDNSリクエストを監視し、最も近い利用可能なPCSサービスロケーションのIPアドレスを返します。
- **DNSゾーン:** DNSゾーンファイルをカプセル化し、ドメイン元を指定します。DNSゾーンファイルには、PCSサービス用のIPアドレスとホスト名のマッピングが含まれます。
- **GLBサービス:** 一連のDNSドメインに使用するグローバル負荷分散設定を表します。GLBサービスはGLBロケーションにリンクし、特定の条件下でのDNSリクエストへのレスポンスを決定するために使用される、負荷分散とモニタリングのロジックが含まれます。
- **サービスモニタ:** GLBロケーションのステータスをモニタリングし、各受信リクエストについて負荷分散の決定を通知します。

- **GLBロケーション:** 世界中の地理的ロケーションの定義です。GLBロケーションは、近さと利用可能性に基づいて、DNSレスポンスでクライアントのトラフィックをどこに送信するかを判断するために、GLBサービスによって使用されます。各PCSサービスのロケーションはGLBロケーションによって表されます。

ウィザードを実行するには、ツールバーで「Wizards (ウィザード)」ドロップダウンメニューをクリックしてから、「Optimal Gateway Selection (最適なゲートウェイの選択)」を選択します。



Traffic Managerにウィザードの最初のページが表示されます。

Optimal Gateway Selection, step 1 of 6

1. Overview

This wizard will guide you through the process of configuring the Virtual Traffic Manager for Optimal Gateway Selection of your globally-distributed Pulse Connect Secure gateway services.

This is achieved by configuring a single, unified, enterprise-wide endpoint URL. The Virtual Traffic Manager uses DNS-based Global Server Load Balancing (GSLB) to provide optimum gateway selection and dynamic failover based on a series of observed criteria:

- The geographic location of the user relative to the PCS gateways
- The current availability of the PCS gateways
- The measured load on the PCS gateways

The wizard requires you to input a domain, hostnames and service location IPv4 addresses for the service to be balanced.

If the service is to listen on a Traffic IP address, then an appropriate Traffic IP group should be created and enabled before commencing this wizard.

Cancel ◀ Back Next ▶

Next (次へ)をクリックして操作を続けます。



それぞれのウィザードのページで、**Cancel** (キャンセル)を使用すると、Traffic Managerを変更せずにウィザードがキャンセルされます。**Back** (戻る)を使用すると前のページに戻り、**Next** (次へ)を使用すると次のページに進むことができます。

Optimal Gateway Selection, step 2 of 6

2. Service name

Enter a name to identify the service that is to be load-balanced.
All configuration created by this wizard will be prefixed with the name entered here.

Service Name:

Cancel

◀ Back

Next ▶

Traffic Managerはここで指定する識別子を、このプロセスで作成される全ての設定オブジェクトのプレフィックスとして使用します。

識別名を入力し、**Next** (次へ)をクリックして操作を続けます。

Optimal Gateway Selection, step 3 of 6

3. Virtual Server configuration

Select which IP addresses or Traffic IP groups the service should listen on (port 53):

- ☐ All IP addresses
☒ Traffic IP Groups ...

Traffic IP Group	Select
TIP	☒
TIP2	☐

Cancel ◀ Back Next ▶

このページを使用して、新しいDNSサービスがTraffic Managerクラスタによってホストされた全てのIPアドレスを監視するか、または特定のトラフィックIPグループ内のトラフィックIPのみを監視するかを決定します。トラフィックIPアドレスおよびグループの詳細については、[トラフィックIPグループとフォールトトレランス](#)を参照してください。

Next (次へ)をクリックして操作を続けます。

Optimal Gateway Selection, step 4 of 6

4. Domain configuration

Enter a fully qualified domain name to be load-balanced.

Domain:

Enter the nameserver details below.

Primary nameserver domain: ▼
Primary nameserver IP address: ▼
Secondary nameserver domain: ▼
Secondary nameserver IP address: ▼

Cancel ◀ Back Next ▶

統合エンドポイントURLとして使用する、完全修飾ドメイン名を入力します。次に、プライマリとセカンダリのネームサーバのドメインとIPアドレスを選択します。

Next (次へ)をクリックして操作を続けます。

Optimal Gateway Selection, step 5 of 6

5. Service locations

Add details of your PCS endpoint locations. For each endpoint, enter a name, IP address(es), and location information.

	Hostname	IP address(es)	Country	Latitude	Longitude	Remove
1.	eu-vpn	192.0.2.0	United Kingdom ▼	54.75844	-2.69531	Remove
2.	us-vpn	192.0.2.1	United States ▼	39.76	-98.5	Remove
3.	ap-vpn	192.0.2.2	Japan ▼	35.68536	139.75309	Remove

Add location

Enter a TCP port the backend service locations are listening on.

Service port: 443

Cancel ◀ Back Next ▶

このページを使用して、PCSサービスのロケーションを定義します。ロケーションごとに、名前、IPアドレス、および物理的なロケーションを入力します。Traffic Managerはこの情報を使用して、サービスで使用するGLBロケーションを構築します。ロケーションを更に追加するには、**Add location** (ロケーションを追加)をクリックします。ロケーションを削除するには、横にある**Remove** (削除)ボタンをクリックします。



GLBロケーションでは、Traffic ManagerはIPv4アドレスのみをサポートしています。

ロケーションが監視するポート番号を指定するには、「Service Port (サービスポート)」を使用します。

Next (次へ)をクリックして操作を続けます。

Optimal Gateway Selection, step 6 of 6

6. Summary

The following configuration will be created:

Service:	PCS-OGS					
Domain:	vpn.example.com					
Service locations:		Name	IP address(es)	Country	Latitude	Longitude
	1.	eu-vpn	192.0.2.0	United Kingdom	54.75844	-2.69531
	2.	us-vpn	192.0.2.1	United States	39.76	-98.5
	3.	ap-vpn	192.0.2.2	Japan	35.68536	139.75309
Service port:	443					
Listening on:	TIP					

To create this service, press 'Finish'. To change your settings, press 'Back'.

このページには、提案されたTraffic Manager設定の要約が表示されます。**Cancel** (キャンセル)をクリックすると、変更を行わずにウィザードを終了できます。**Back** (戻る)をクリックすると前のページに戻り、**Finish** (完了)をクリックするとウィザードが完了して、Traffic Managerが設定されます。

ウィザードで全ての設定が完了すると、Traffic Managerのホームページが更新され、実行中の全てのサービスが表示されます。

トラブルシューティング

本章では、Traffic Managerが正常に動作しているかどうかをテストする方法と、発生した問題の詳しい対処方法について説明します。

ツールおよび手法

Traffic Managerの設定をテストする場合、様々なツールおよび手法を利用して、設定およびネットワークトラフィックの挙動を把握することができます。

- **診断:** Cluster Diagnosis (クラスタ診断)ページでは、Traffic Managerシステムにわたってネットワークコネクション、ヘルスマニタ、設定エラー、および設定の競合を確認する様々なテストを実行できます。
- **イベントログ:** イベントログ(ホームページおよびEvent Log (イベントログ)ページに表示)には、発生した重要イベントが記録されます。
- **コネクション:** Connections (コネクション)レポートには、継続中および最近のコネクションについて、接続元、処理方法、転送されたデータ量などが即座に表示されます。
- **リクエストロギング:** Traffic Managerでは完全なログを書き込むことができるため、各リクエストおよびレスポンスの詳細を全て記録し、後で解析に利用することができます。
- **設定概要:** 高度な設定を行った場合も、Configuration Summary(設定概要)で、使用されている設定オブジェクトおよび相互の関係を確認することができます。
- **監査ログ:** 監査ログには、全ての設定の変更について、変更がいつ行われたか、どのユーザが行ったか、そのユーザはどこに属しているかが記録されます。

本章では、これらの各ツールと手法について詳しく説明します。

診断とイベントログ

Traffic Managerシステムまたは設定で問題が発生した場合、まず、Traffic Manager管理UIの**Cluster Diagnosis** (クラスタ診断)ページを確認してください。

注記

クラスタ内のどのTraffic Managerからも管理UIにアクセスできない場合は、Traffic Managerソフトウェアを起動またはリスタートすることをお奨めします。このプロセスについては、ご使用の製品の『*Pulse Secure Virtual Traffic Manager: Installation and Getting Started Guide*』を参照してください。エラーメッセージ、特に \$ZEUSHOME/zxtm/log/errorsのエラーログファイルに追加された全てのメッセージに十分に注意を払ってください。

管理UIの任意のページにある**Diagnose** (診断) ボタンをクリックすると、現在のシステムステータスを確認できます。**Diagnose** (診断) ページは複数のセクションに分かれており、発生した問題を理解し、解決する上で役立ちます。

Cluster Diagnosis (クラスタ診断) セクションには、使用不可のマシンや不正な設定など、Traffic Managerが検出した問題が報告されます。

イベントログには、各Traffic Managerマシンのエラーログの内容が表示されます。最近作成されたログを、期間を指定して表示することができます。Event Log (イベントログ) ページは、新しいイベントメッセージがログに追加されると動的に更新されます。

監査ログには、最近行われた設定の変更について、それがいつ行われたか、どのユーザが行ったかが表示されます。

Technical Support (技術サポート) セクションには、様々な資料への便利なリンクのほか、必要に応じて Pulse Secure Technical サポートチームに送信する技術サポートデータをダウンロードするためのオプションが表示されます。

 Pulse Secure Services Director デプロイ内で Traffic Manager インスタンスを使用している場合、お問い合わせいただく際には、このセクションに表示される「SSC サポート資格」の識別子を使用してください。Ivanti は、お客様のサポート資格レベルをユニークに特定するためにこの情報を使用します。

通常の運用時に問題が検出された場合、**ステータスアプレット**に警告またはエラー状態が表示されます。

Cluster Diagnosis (クラスタ診断) ページに移動して、問題の詳細を調べてください。



リクエストとレスポンスのモニタリング

観察された問題に関するエラーが **Cluster Diagnosis** (クラスタ診断) ページに報告されない場合、Traffic Manager がトラフィックを正しく管理しているか、すなわちバックエンドサーバに代わってリクエストを読み取り、バックエンドサーバからのレスポンスをリモートクライアントに返しているかどうかを検証します。

問題を再現するときに、**テストリクエストの生成**が役立つ場合があります。

コネクション状況レポート

Traffic Managerで接続アクティビティを確認するには、Traffic Managerの管理UIで**Activity** (アクティビティ) > **Connections** (コネクション)をクリックします。**Connections** (コネクション)ページには、Traffic Managerが管理した全ての継続中のコネクションおよび最近完了したコネクションのレポートが表示されます。**Key** (キー) セクションを展開すると、その情報の詳細が表示されます。

現在コネクションを処理しているTraffic Managerとそれに関与しているバックエンドノードや現在のコネクションの状態に加え、コネクションに関与している仮想サーバ、ルール、およびプールを確認できます。また、コネクションの再試行回数、アイドル時間、およびバイト数も表示されます。

データを最新の状態にするには、インターネットブラウザの更新ボタンをクリックします。データを.tsvファイルとしてダウンロードすることもできます。

Connections (コネクション)レポートの使用



テストしているリクエストがTraffic Managerによって処理されたかどうかを検証します。このリストにテストリクエストが表示されない場合、Traffic Managerがリクエストを受信しなかった可能性があります。クライアント設定とDNS設定を確認し、テストクライアントがTraffic ManagerのIPアドレスにリクエストを送信しているかどうか、またリクエストがいずれかのバックエンドサーバで処理されているかどうかを検証してください。

一般的な問題として、バックエンドサーバが自己参照型のレスポンスを発行していることが挙げられます。例えば、Traffic Managerがwww.example.com宛でのトラフィックを処理していて、バックエンドサーバの名前が「server1.example.com」および「server2.example.com」であるとします。一方のサーバが「server1.example.com」へのリンクまたはリダイレクトを含むレスポンスを発行すると、以降、クライアントはTraffic Managerをバイパスしてバックエンドサーバと直接通信しようとするため、問題の原因となります。このタイプの問題は、レスポンスにDNS名またはIPアドレスを埋め込むその他のプロトコルでも発生する可能性があります。

コネクションレポートのサイズは、**System** (システム) > **Global Settings** (グローバル設定)ページで拡張することができます。**recent_conns**キーの「Logging (ロギング)」セクションを確認してください。

リクエストログ

仮想サーバのリクエストロギング機能を使用して、トランザクションの様々な側面を記録することができます。この機能は、断続的なエラーや再現が難しいエラーについて、長期的なテストを行う場合に非常に役立ちます。設定手順については[リクエストロギング](#)を参照してください。

TrafficScriptロジックが複雑な場合、コネクションのデバッグ情報を保存し、それをリクエストログに記録することができます。TrafficScriptルールで`connection.data.set()`関数を使用してコネクションに関する特定データを保存し、次に「`%{Key-name}d`」マクロを使用してそのデータをリクエストログに記録します。

高度なロギング

TrafficScriptルールを使用して、仮想サーバが処理したトランザクションに関する情報を記録することができます。

TrafficScript関数は、リクエストのあらゆる部分の情報を読み込むことができます。`http.getHeader()`や`http.cookie()`などの関数を使用すると、HTTPリクエストに関する情報を取得できます。また、`ssl.getSessionID()`などの関数を使用すると、SSLトラフィックに関する情報を取得できます。その他のプロトコルの場合は、`connection.getData()`関数を使用して、リクエストの指定バイト数の情報を取得することができます。

`log.info()`関数を使用すると、受信したリクエストの詳細をTraffic Managerのイベントログファイルに書き込み、これを基に受信トラフィックで発生している問題を突き止めることができます。また、`event.emit()`関数を使用し、適切なイベントハンドラを設定すると、デバッグメッセージの記録場所をより厳密にコントロールすることができます。

利用可能なTrafficScript関数の詳細については、『TrafficScript Reference』を参照してください。

注記

全ての受信リクエストについてこの処理を行うと、ログファイルが急速に増大します。このような情報を記録する場合はご注意ください。

イベントのモニタリング

TrafficScript関数の`counter.increment()`を使用すると、特定イベントの発生回数をカウントすることができます。詳細については、[Activity \(アクティビティ\) > Current Activity \(現在のアクティビティ\)](#)を参照してください。

コネクションの詳細なデバッグ

コネクションに失敗すると、コネクション状況レポートおよびリクエストログに記録される情報が不完全になります。これは、コネクションが正常に完了しなかったためです。コネクションの失敗原因は、プロトコルエラー、タイムアウト、予測外のTCP終了など様々です。

このような予測外の問題をデバッグするには、[エラーの処理](#)で説明しているように、影響を受けたサービスの **Virtual Server** (仮想サーバ) > **Edit** (編集) > **Error Handling** (エラー処理)設定ページで **log!client_connection_failures**および**log!server_connection_failures**設定を有効化します。これらの設定により、コネクションに失敗した場合に必ず詳細なデバッグメッセージをイベントログに書き込むようTraffic Managerが設定されます。

Event Log

Event Log

The Event Log shows the contents of the error logs on each traffic manager machine.

Timescale:

Last N events

15

Event Filter:

Connection Failures

Show 'Configuration modified' messages:

Yes

No

Yes

Stop updating

Display

Download

✓	15/Jul/2009:11:56:46 +0100	INFO	Virtual Server zws-8765: "Invalid HTTP response" S 10.100.1.186:38653 10.100.1.186:8765 "zws-18765" "yinkin:18765" "-" r 71 106 0 0 0 0 0 0 500 GET "http://yinkin:8765/nph-junk.cgi?foo=bar&quote=%22&tab=%09"	yinkin
✓	15/Jul/2009:11:56:45 +0100	INFO	Virtual Server zws-8765: "Pool has no back-end nodes responding" 10.100.1.186:38651 10.100.1.186:8765 "testtcp" "-" c 38 0 0 0 0 0 1 - GET "http://10.100.1.186:8765/"	yinkin
✓	15/Jul/2009:11:56:45 +0100	INFO	Virtual Server zws-8765: "Connect failure - Connection refused" S 10.100.1.186:38651 10.100.1.186:8765 "testtcp" "yinkin:13765" "-" c 38 0 0 0 0 0 0 - GET "http://10.100.1.186:8765/"	yinkin

上図のような「Connection Failures (接続失敗)」イベントフィルタを使用して、接続失敗に関するメッセージのみが表示されるようイベントログをフィルタすることができます。

ログ行には、エラーの記述(「Invalid HTTP Response」など)と、コネクションの性質についての詳細(使用ノード、接続回数、再試行回数など)が表示されます。ログフォーマットの詳細は、**Services** (サービス) > **Virtual Server** (仮想サーバ) > **Edit** (編集) > **Error Logging** (エラーロギング)設定ページのオンラインヘルプで参照することができます。

エラーの記述部分をクリックすると、接続エラーの内容が分かりやすい形で表示されます。

Copyright © 2022 Ivanti, Inc. 著作権所有。 [プライバシーおよび法律](#)

ページ522/634

✓	15/Jul/2009:11:56:46 +0100	INFO	Virtual Server zws-8765: "Invalid HTTP response"	
			Peer	Server
			Source Address	10.100.1.186:38653
			Destination Address	10.100.1.186:8765
			Pool	zws-18765
			Node	yinkin:18765
			Rule	-
			Connection State	Reading from server
			Bytes from client	71
			Bytes to server	106
			Bytes from server	0
			Bytes to client	0
			Connection Established	0s
			Client Idle Time	0s
			Server Idle Time	0s
Connection Retries	0			
Response Code	500			
Request Line	GET "http://yinkin:8765/nph-junk.cgi?foo=bar&quote=%22&tab=%09"			
			 yinkin	

デフォルトでは、コネクションエラーはグローバルイベントログに記録されます。全ての接続失敗イベントをキャプチャし、グローバルイベントログをバイパスして別のログファイルに書き込むイベントハンドラを作成できます。詳細については、[イベント処理およびアラート](#)を参照してください。

個々のノードのテスト

プール内の各バックエンドノードがTraffic Managerからリクエストを受信できるかどうかを確認するには、プールの負荷分散アルゴリズムを「ラウンドロビン」に切り替えます([負荷分散](#)を参照)。ラウンドロビン負荷分散アルゴリズムでは、リクエストが各ノードに順番にルーティングされるため、バックエンドサーバのログを使用して、全てのノードでこのアルゴリズムが正常に機能しているかどうかを確認することができます。

コネクションをドレインするようノードを設定することもできます([ノードのドレイン](#)、[無効化](#)、[削除](#)を参照)。プールが複数のノードで構成されている場合、一部のノードをドレインするよう設定して、Traffic Managerから一時的にこれらのノードにトラフィックが送信されないようにできます。テストするノードを除く全てのノードをドレインすることにより、各ノードを順番にテストすることができます。

各ノードでサーバロギングを有効化し、ノードによって書き込まれたログとシステムのアクティビティが一致していることを検証してください。

設定の確認

Configuration Summary (設定概要)には、Traffic Managerでリクエストがどのように処理されたかを示す概要が表示されます。リクエストは、Traffic Managerが受信し、仮想サーバおよびプールでの処理後にノードに転送され、その後、戻り経路上にあるプールおよび仮想サーバによってレスポンスが処理されます。

多数のTrafficScriptルールおよびTraffic Manager機能が有効になっている状態では、処理経路が非常に複雑になる可能性があります。

Configuration Summary (設定概要)については、[Services \(サービス\) > Configuration Summary \(設定概要\)](#)で説明しています。

調査中の問題を特定の時間にトレースできる場合は、**Diagnose (診断) > Audit Log (監査ログ)**ページを使用し、必要であれば**System (システム) > Backups (バックアップ)**ページを使用して、問題を引き起こした可能性のある設定変更を特定します。

監査ログには、Traffic Manager管理UI、REST API、およびSOAP APIの各ユーザが行った設定の変更が全て記録されます。認可された各管理者に個別のアカウントを用意すれば、これらの管理者の権限を管理し、そのアクティビティをモニタリングすることができます。ユーザおよびグループの設定については、[ユーザ管理](#)に説明があります。

Backup Management (バックアップ管理)では、手動で設定のスナップショットを作成することができます。現在の設定と正常に動作していたときのスナップショットを比較し、設定に重大な問題がある場合にバックアップを復元することができます。バックアップ管理は[System \(システム\) > Backups \(バックアップ\)](#)に説明があります。

トラブルシューティングのヒント

トラフィック管理される複雑なクラスタで発生した障害の診断プロセスは単純ではないため、体系的なアプローチでトラブルシューティングを行う必要があります。問題が発生した場合、以下の事柄を順番に確認し、障害の原因を突き止めて診断してください。

テストリクエストの生成

システムのテストは、Webブラウザや電子メールクライアントなどユーザ用の一般的なクライアントソフトウェアを使用して行うことができます。MozillaブラウザのLive HTTP Headers拡張などのツールは、クライアントとTraffic Manager間でやり取りされるリクエストおよびレスポンスの流れをチェックする場合に非常に役立ちます。

Wireshark (<http://www.wireshark.org/>)などのネットワークスプーフィングツールでも、ネットワークトラフィックを記録したり、リクエストおよびレスポンスセッションを作成したりすることができます。

httpclient

Traffic Managerに付属の「httpclient」プログラムを使用すると、特定のマシンに対してHTTPリクエストを送信することができます。このプログラムは\$ZEUSHOME/admin/bin/httpclientにあります。

以下の構文を使用します。

```
httpclient --hostheader=<website_name> http://<trafficIP>/
```

telnetクライアントを使用して、サービスに対して基本的なテストを行う初期テストを実行することもできます。

```
$ telnet www.mysite.com 80
Trying 62.254.209.66...
Connected to 62.254.209.66.
GET / HTTP/1.1<RETURN>
Host: www.mysite.com<RETURN>
<RETURN>
```

OpenSSLツールキット(<http://www.openssl.org/>)には、SSL関連の問題をテストするための、SSLプロトコルを使用するtelnetに似たクライアントが含まれています。

zeusbench

「ZeusBench」プログラムは、多数のHTTPリクエストを送信して負荷および性能テストを実行できる便利なベンチマークツールです。このプログラムは\$ZEUSHOME/zxtm/bin/zconfiにあります。

多数のコマンドラインオプションが用意されています。単純な負荷テストは、以下のコマンドを使用して実行できます。

```
zeusbench -t 30 -c 100 -k http://host/url
```

zeusbench -hを実行すると、全てのコマンドラインオプションのリストが表示されます。

自動バックエンドフェールオーバーの確認

Traffic Managerによるバックエンドの自動フェールオーバーを確認するには、フェールオーバー先のマシンが必要になるため2台以上のバックエンドサーバが必要です。フェールオーバーをテストするには、1台のバックエンドマシンのネットワークケーブルを引き抜くか、そのバックエンドで実行中のサービスを手動で停止します。次に、バックエンドのそのポートで何も監視されていないことを確認します。

バックエンドサーバが1台のみの場合は、サービスの2つのインスタンスを同じサーバマシンの異なるポートで実行し、いずれかのインスタンスを手動で停止させることで代用できます。

Traffic Managerを通してサービスを利用してみます。SMTPの場合は、サーバファームを通して電子メールを送信します。HTTPの場合は、Webページをリクエストします。少なくとも1台のバックエンドサーバでリクエストを満たすことができれば、このテストは成功です。

Diagnose (診断) > Event Log (イベントログ) ページのリンクからイベントログにアクセスし、バックエンドマシンでの障害発生を示す通知があることを確認してください。

自動フロントエンドフェールオーバーの確認

複数のTraffic Managerを使用している場合、自動フロントエンドフェールオーバーが正常に機能しているかどうかを確認することができます。全てのTraffic ManagerにまたがるトラフィックIPグループと、このトラフィックIPグループを使用するサービス(ポート80でWebコンテンツを管理する仮想サーバなど)を設定します。このサービスのWebページを、グループに含まれる各トラフィックIPアドレスでリクエストできることを確認します。これを行うには、ブラウザにDNS名ではなくIPアドレスを入力します。

Services (サービス) > Traffic IP Groups (トラフィックIPグループ) をクリックし、**Unfold All (全て開く)** をクリックしてトラフィックIPグループの詳細を表示します。ここには、どのマシンでどのIPアドレスが有効になっているかが示されます。マシンの台数がグループ内のトラフィックIPアドレスの数より多い場合、一部のマシンはスタンバイ状態になり、トラフィックをアクティブに処理しないことに注意してください。

次に、いずれかのトラフィックIPアドレスが有効になっている1台のマシンを選択し、そのマシンのネットワークケーブルを全て引き抜きます。これにより、そのIPアドレスがグループ内の別のマシンで有効になります。ブラウザで再度そのトラフィックIPにアクセスし、コンテンツを受信できることを確認します。キャッシングされたコンテンツが使用されないよう、ブラウザでページを更新することが必要な場合があります。

[フォールトトレランスの設定](#) では、Traffic Managerのフォールトトレランスの動作、実行されるテスト、および決定について説明しています。

起こりやすい問題

rootとして起動されない

Traffic Managerは、メールやHTTPなどのネットワークサービスの負荷分散を行うために特権ポートにバインドする必要があります(特権ポートはポート番号が1024より小さいポート)。これを行うには、インストール後、configureスクリプトを実行し、Traffic Managerをrootとして起動する必要があります。

フロントエンドフォールトトレランスモードでは、Traffic Managerは、ネットワークインターフェースを有効/無効にする必要があります。このソフトウェアをrootとして設定および起動しなかった場合、フェールオーバーアクティビティは機能しません。

Traffic Managerを非特権ユーザとして実行する場合でも、rootユーザとして設定して起動する必要があることに注意してください。

評価のために、root以外のユーザでTraffic Managerを設定および起動することもできます。ただし、この場合は、特権ポートにバインドしたり(上記を参照)フォールトトレランスを使用したりすることはできません。

コネクションが拒否される

フロントエンドで最も起こりやすい設定エラーは、外部DNS名をフロントエンドマシンの外部IPアドレスにマッピングした不正なDNS設定です。フロントエンドのDNS名でサーバに接続したときに「Connection Refused (コネクション拒否)」メッセージを受信した場合、DNSが正しく設定されていない可能性があります。

DNS設定を確認するには、hostまたはnslookupコマンドを使用します。

```
$ host www.w3.org
www.w3.org has address 18.29.1.35
www.w3.org has address 18.7.14.127
www.w3.org has address 18.29.1.34
```

Traffic ManagerマシンがこれらのIPアドレスで監視していることを確認します。

これらのIPアドレスでトラフィックを監視するよう明示的に仮想サーバを設定することができます。これにより、Traffic ManagerクラスタからこれらのIPアドレスを使用できない場合に、Diagnose (診断)ページにエラーが報告されるようになります。

トラフィックIPアドレスが正しく設定されていない

フロントエンドフォールトトレランスを実現するためにトラフィックIPグループを設定すると、選択したトラフィックIPアドレスが適切かどうかを確認するために、Traffic Managerによっていくつかのチェックが行われますが、念のため、これらのIPが現在のネットワーク構成に適しているかどうかを再確認してください。一般的な間違いとして、選択したトラフィックIPアドレスが他の場所で使用されている(Traffic Managerマシンの固定IPアドレスなど)、Traffic Managerが使用しているサブネット上にないIPアドレスを選択した、選択したIPアドレスに他の場所からルーティングできない、などがあります。

Traffic Managerがプロトコル開始前にコネクションを破棄する

Server Firstプロトコルでは、Traffic ManagerがServer Firstプロトコル用の特定ポートでバックエンドサーバと通信できない場合、クライアントとのコネクションを破棄せざるを得ません。(telnetなどを使用して)Traffic Managerマシンの適切なポートに接続し、コネクションに成功しても(すなわち「Connection Refused (コネクション拒否)」メッセージが返されない)、その直後にコネクションが破棄される場合、Traffic Managerはバックエンドと通信しようとしていますが、バックエンドサーバが見つからない状態にあります。

Diagnose (診断) > Event Log (イベントログ) ページで、バックエンドサーバのステータスに関するメッセージを確認してください。

WebサーバがError 400を返す

トラフィック管理されているWebサイトのテストでError 400 Bad Requestタイプのエラーが返される場合、アクセスしようとしているWebサイトが、他のホストヘッダ(またはエイリアス)を受け入れるよう設定されていない可能性があります。この問題を解決するには、適切なホストヘッダを設定するルールを追加します。

```
http.setHeader( "Host", "www.mysite.com" );
```

間違ったポート番号が設定されている

この他、起こりやすい問題として、負荷分散のため使用しているポート番号が間違っていることが挙げられます。例えば、Webサイトを80以外のポートで運用している場合、HTTPプロトコルでデフォルトのポートではなくそのポート番号を使用して負荷分散を行う必要があります。SSL暗号化サイトを運用している場合は、HTTPではなくHTTPSプロトコルを選択するか、SSLトラフィックを復号化してください。HTTPSのデフォルトポートは443です。

適切なプロトコルが選択されていれば、1つのTraffic Managerにより多数のポートで負荷分散を行うことができます。

ファイル記述子を使い果たした

Traffic Managerは、「ファイル記述子」と呼ばれるOSのリソースを使用して、クライアントまたはサーバへの各ネットワーク接続を管理します。Traffic Managerが割り当てることのできるファイル記述子の数にはハードリミットがあり(CPUコア単位などプロセス単位で適用される)、この制限はOSで設定されます。

ベンチマーク時など負荷が非常に大きいときは、「ファイル記述子を使い果たした」ことを示すエラーメッセージが表示されることがあります。この場合、Traffic Managerが使用するファイル記述子の数を増やすことで対処できます。

ファイル記述子の最大数を増やすには、管理UIにログインして**System (システム) > Global Settings (グローバル設定)**をクリックします。「System Settings (システム設定)」セクションで、**maxfds**の値を大きくします。既にOSによる許容最大数に達している場合、次のPulse Secure Knowledge Centerの記事を参照して、このハードリミットを増加させることができるかどうかを確認してください。

https://kb.pulsesecure.net/articles/Pulse_Secure_Article/KB43935

ディスク空き容量を使い果たした



このセクションは、アプライアンスおよびクラウド製品のみを対象としています。

時間とともに、アプライアンスのディスク空き容量が不足する可能性があります。例えば、詳細なリクエストログ情報を生成するようにTraffic Managerを設定している場合、システムログが大きくなることがあります。さらに、アーカイブされたソフトウェアの改訂版(Traffic Managerがロールバックで使用)が不要になっている可能性もあります。

ディスク空き容量が不足し始めると、Traffic Managerは **Diagnose** (診断) > **Event Log** (イベントログ) および **Diagnose** (診断) > **Cluster Diagnosis** (クラスタ診断) ページで警告を発します。また、**System** (システム) > **Traffic Managers** ページからは、いつでもディスク使用容量を確認できます。

ディスク空き容量を確保するには、トップナビゲーションバーにある **Wizards**: (ウィザード:) ドロップダウンメニューから「Free up some disk space (ディスク空き容量の確保)」をクリックします。また、このウィザードは、**System** (システム) > **Traffic Managers** ページの「Free Disk Space (ディスク空き容量の確保)」リンクから、またディスク空き容量の警告が表示されているときは **Diagnose** (診断) > **Cluster Diagnosis** (クラスタ診断) ページからいつでも実行できます。

注記

この操作は取り消すことができません。ウィザードを実行する前に、残しておかなければならない全てのファイルのバックアップを必ず作成してください。また、その後作成する「テクニカルサポートレポート」には、ウィザードが実行された後に生成されたログしか含まれないことにも注意してください。

サポートを利用するには

技術サポートエンジニアに問い合わせるときは、テクニカルサポートレポート(TSR)をダウンロードし、プロBLEMレポートに添付してください。テクニカルサポートレポートには、最新のログファイル、設定、システムのアクティビティ、および様々な内部情報が含まれています。

TSRを作成、ダウンロード、および管理するには、管理UIの **Diagnose** (診断) > **Technical Support** (テクニカルサポート) ページにある **Manage Technical Support Reports** (テクニカルサポートレポートの管理) をクリックします。

このページを使用すると、生成されるTSRに含めるオプションを選択できたり、以前に生成されたレポートにアクセスしたりできます。Traffic Managerには、最後に生成された10のTSRが保存されており、それぞれをダウンロードまたは削除するオプションが用意されています。TSRを外部ロケーションにダウンロードし、表示されるチェックボックスを使用してTraffic ManagerからそのTSRを削除することによって、ディスク空き容量を確保することができます。

また、コマンドラインを使用するか、SOAP APIからTSRを生成することもできます。このオプションは、管理UIへのアクセスが中断されているときに使用してください。

TSRをコマンドラインから生成するには、以下のスクリプトを実行します。

```
$ZEUSHOME/zxtm/bin/support-report [--all-logs] <outfile>
```

オプションの--all-logs引数を使用すると、デフォルトで設定された10 MBのログファイルサイズのダウンロード制限を回避することができます。これにより、サイズにかかわらず、ログセット全体がサポートレポートに含まれるようになります。

用語集

ここでは、本書で使用されている用語について解説します。用語によっては、コンピュータ関連の文献において様々な意味で使用されているものもありますが、この用語集では、本書の範囲に限定して意味を定義しています。

用語	意味
アクション	アクションは、特定イベントが発生した結果としてイベントハンドラが実行されると呼び出されます。アクションには、ログファイルへの書き込み、電子メールメッセージの送信、SYSLOGメッセージまたはSNMPトラップの送信、カスタムアクションプログラムの実行などがあります。
管理サーバ	Webベースの管理UIとして機能し、コアとなるTraffic Managerソフトウェア、管理UI、および全ての製品API間の設定とステータス通信を処理するプロセスです。
監査ログ	Traffic Managerに対するユーザのアクティビティが記録されるTraffic Managerの監査ログ。Diagnose (診断)ページから表示することができます。
バックエンドIPアドレス	Traffic Managerマシンがバックエンドサーバとの通信に使用するIPアドレス、またはバックエンドサーバのIPアドレス。
バックエンドサーバ	Webやメールサーバなどのサービスを実行するマシン。通常、バックエンドサーバはローカルネットワーク内に存在し、Traffic Managerが処理したリクエストを受け取ります。Traffic Managerでは、バックエンドサーバと指定ポートの組み合わせによってノードが形成されます。
帯域制御	特定サービスまたは特定タイプのリクエストに利用できるネットワーク帯域をモニタリングおよびコントロールする機能。
カタログ	サービスに適用可能なオブジェクトのセントラルリポジトリ。ルール、モニタ、サービスプロテクションクラス、および様々なSSL項目のカタログがあります。
認証局	SSL証明書への署名を独自に行う広く認められた機関。
証明書失効リスト(CRL)	無効として扱うべきクライアントSSL証明書のリスト。認証局が発行します。

用語	意味
証明書署名リクエスト(CSR)	自己署名証明書を基に作成されるリクエスト。CSRは、認証局の署名を得るために認証局に送信することができます。
CIDR IPサブネット	Classless Inter-Domain Routing (CIDR) IPサブネットは、標準の32ビットIPアドレスと、ネットワークプレフィックスの使用ビット数に関する追加情報で構成されます。例えば、「10.13.1.48/22」というCIDR IPサブネットの場合、「/22」は最初の22ビットが固有のネットワークプレフィックスを識別するために使用され、残り10ビットによって特定のホストが識別されることを示します。または、Traffic Managerでは、サブネットマスク「10.13.1.48/255.255.255.0」または明示的に「10.13.1.」を使用してCIDR IPサブネットを指定することができます。
クライアント証明書	クライアントが保有するSSL証明書で、制限されたサイトへのアクセスをクライアントに許可するために使用されます。
クラスタ	同じ設定を共有するTraffic Managerのグループ。
configureスクリプト	Traffic Managerをインストールしてから起動するまでの間に実行する必要があるスクリプト。パスワードなどの基本的な設定、Traffic Managerをスタンドアロンで運用するか既存のクラスタに組み込むかの指定を処理します。
クッキー	サーバからクライアントに与えられ、クライアントのファイルシステムまたはブラウザクライアントセッションに保存される小さいデータ項目。クライアントはこのデータを保存し、以降のリクエスト時にサーバに提供します。クッキーは、セッション維持マップなどのクライアントデータをトラッキングするために使用されます。
DoS攻撃	サービスの正当な利用を妨げる悪意のある試みで、一般にはネットワークへの過剰なデータの送信やマシン間の接続の妨害といった形を取ります。
Diagnose (診断)ページ	Traffic Manager環境の障害の診断と解決に役立つ管理UIのセクション。
無効化されたノード	プール内の使用されていないノード。このノードに対して、モニタリングやトラフィックの送信は行われません。ノードを無効化すると、ノードをプールから削除した場合と同じ状態になりますが、復旧を容易に行えます。
DDoS攻撃	マシン群によるDoS攻撃。多くの場合、これらのマシンはワームやウイルスに感染しています。

用語	意味
DMZ	ファイアウォールの非武装地帯。この地帯は内部ネットワークとインターネットとの間に位置し、多くの場合、ここに外部からアクセス可能なサーバが配置されます。
ドレイン中のノード	プール内の、新しいコネクションが送信されないノード。このノードに対する既存のコネクションおよびセッションが全て期限切れになった時点で、このノードをプールから安全に削除することができます。
エラーファイル	リクエストに対してレスポンスを返すことのできるノードがない場合にクライアントに送信されるファイル。エラーファイルはプール単位で設定されます。
イベント	IPアドレス転送やノードの障害または復旧など、特定の変更や事象が起これば、イベントが報告されます。全てのイベントはグローバルイベントログに記録されます。
イベントハンドラ	イベントハンドラは、関連付けられたイベントタイプのイベントが発生したときにTraffic Managerが実行するアクションを設定します。
イベントログ	Traffic Manager内で発生したイベントが記録されるTraffic Managerのログ。重大度に応じてINFO (情報)、WARN (警告)、SERIOUS (重大)、FATAL (致命的)などに分類され、Diagnose (診断)ページから表示することができます。
イベントタイプ	管理を容易にするためにグループ化されたイベントのセット。特定イベントタイプのイベントが発生した場合、Traffic Managerはそのイベントタイプに関連付けられたアクションを実行できます。
フェールオーバ	障害が発生したマシンの負荷をクラスタ内の他のマシンが引き継ぐ動作。
障害プール	プール内の全てのノードで障害が発生した場合、リクエストは、そのプールの障害プールに送信されます(設定されている場合)。
フォールトトレランス	サービスを中断することなく、1台以上のサーバの障害に対処するためのクラスタまたはプールの機能。
フロントエンドIPアドレス	Traffic Managerの、外部からアクセス可能な固定IPアドレス。
フロントエンドサーバ	Traffic Managerなど、インターネット側から通信可能なサーバ。

用語	意味
ヘルスモニタリング	障害が発生したマシンにリクエストが送信されないようにするために、Traffic ManagerがTraffic Managerおよびバックエンドのヘルスをトラッキングするプロセス。
IPv6	現在のプロトコルであるIPv4の後継として考えられているネットワークレイヤプロトコル。従来プロトコルよりアドレス範囲が広く、セキュリティが強化されているなどの特長があります。
Java Extension	Java Extensionsにより、TrafficScriptの機能を強化(拡張)することができます。リクエストまたはレスポンスを処理するために、TrafficScriptルール(またはRuleBuilderルール)からJava Extensionを呼び出すことができます。Java Run-time環境では、TrafficScriptのみを使用した場合よりはるかに強力なプログラミング言語とライブラリを利用できます。
負荷分散	複数のバックエンドサーバにリクエストを分配すること。これにより、全てのリクエストをできる限り迅速に処理することができます。
MD5	暗号化において、MD5 (Message-Digest algorithm 5)は広く利用されている暗号化ハッシュ関数で、128ビットのハッシュ値を出力します。インターネット規格(RFC 1321)として様々なセキュリティアプリケーションに採用されており、ファイルの完全性チェックにも広く利用されています。
MIMEタイプ	Multipart Internet Mail Extension (多目的インターネットメール拡張)。転送されるデータのタイプを識別する名前。例えば、PDFファイルの場合は「application/pdf」となります。この規格は当初電子メールアプリケーションに使用されていましたが、その後Webサーバなど他のインターネット技術に広く採用されました。
モニタ	Traffic Managerがプールのヘルスをトラッキングするために使用する定期的なテスト。
ネームサーバ	ネームサービスプロトコルを実装するサーバ。例えば、ドメインネームシステム(DNS)サーバは、ドメイン名www.example.comをIPアドレス208.70.196.59に変換します。

用語	意味
NAT	ネットワークアドレス変換(NAT、ネットワークマスカレードとも呼ばれる)では、IPパケットがルータまたはファイアウォールを通過するときはその送信元または宛先アドレスの書き換えが行われます。NATを使用する多くのシステムでは、プライベートネットワーク上の複数のホストが単一のパブリックIPアドレスでインターネットにアクセスできるようにするためにこの機能が使用されます。
ノード	ポートが関連付けられたバックエンドサーバ。Traffic Managerから送信されたリクエストを受信し、レスポンスとしてリクエストされたコンテンツを返します。
PEM	Private Enhanced Mail。 .pemファイル拡張子は、Base 64で符号化されたX.509鍵および証明書に使用されます。これらは出力可能なテキストで、----- BEGIN CERTIFICATE -----形式の行から始まります。
PKCS#11	セキュアなハードウェアとの通信に使用される規格。
プール	ノードの論理グループ。仮想サーバがリクエストをプールに割り当てると、プールはリクエストをノードに負荷分散します。
プロトコル設定	リモートクライアントと仮想サーバ間またはプールとそのノード間のコネクションを管理する設定。
Real Time Streaming Protocol (RTSP)	TCPおよびUDPを使用してオーディオおよびビデオフォーマットなどのリアルタイムデータのオンデマンド配信を可能にするアプリケーションレベルプロトコル。
Real-time Transport Protocol (RTP)	オーディオおよびビデオファイルを配信するためのUDPパケットフォーマット。ストリーミングデータを配信するために、RTSPとともに広く使用されています。
リクエストルール	Traffic Managerでどのようにリクエストを処理するかをコントロールするTrafficScriptルールまたはRuleBuilderルール。ルールは仮想サーバから呼び出され、リクエストをチェック、操作、書き換え、およびルーティングすることができます。また、リクエストを処理するためにTraffic Managerの他の機能をどのように使用するかもコントロールできます。
レスポンスルール	レスポンスルールはリクエストルールと対称をなすもので、Traffic Managerでどのようにレスポンスを処理するかをコントロールします。

用語	意味
RuleBuilder	ルールを作成するためのビジュアルエディタ。リクエストのチェック条件と、その条件の一部または全てが満たされた場合のアクションを選択することができます。
自己署名SSL証明書	認証局ではなく所有者によって署名されたSSL証明書。セキュリティテストには役立ちますが、商用サイトでは使用しないでください。認証局の署名のある証明書にするために、証明書署名リクエストを送信することができます。
サーバ証明書	SSLサービス(Webサイトなど)を識別するために使用されるSSL証明書で、暗号化されたトランザクションを確立するために必要な公開鍵が含まれます。
サービス	一般的なインターネットサービスとして、Web、メール、DNS、.NETやSOAPなどのアプリケーションがあります。IPまたはDNSアドレスとポートの組み合わせとして、パブリックまたはローカルに提供されます。
サービスレベルモニタリング	バックエンドサーバのレスポンス時間をモニタリングし、それを適合値と比較して、利用可能なリソースを動的に調整するなど変化に対処する機能。
サービスプロテクション	インターネットサービスをDoS(サービス妨害)やDDoS(分散型サービス妨害)などの攻撃から保護すること。
サービスプロテクションクラス	ユーザが定義するサービスプロテクション設定のグループ。このクラスはサービスプロテクションカタログに保持され、任意の仮想サーバに適用できます。
Session Initiation Protocol (SIP)	カンファレンス、電話、プレゼンス、イベント、およびインスタントメッセージングのためにインターネットで使用するプロトコル。
セッション維持	同じユーザセッションに属するリクエストを常に同じノードに送信するためのプロセス。セッションは、IPアドレス、クッキー、SSLセッションIDなど複数の方法で識別できます。
SNMP	Simple Network Management Protocol (簡易ネットワーク管理プロトコル)。ネットワークのモニタリングおよび管理に関するオープン規格。SNMPにより、ネットワーク上のデバイスをモニタリングし、性能データを収集することができます。

用語	意味
SOAP	Simple Object Access Protocol。WebサービスメッセージのためのXMLベースの規格。
SSL	Secure Sockets Layer。暗号化されたネットワーク通信を実現するプロトコル。リクエストはTraffic Manager内で復号化し、再度暗号化することができます。
Tarアーカイブ	tarファイルフォーマットのファイル。アーカイブファイルフォーマットの標準タイプ。ユーザやグループの許可、日付、ディレクトリ構造などのファイルシステム情報を維持しながら、ファイルをアーカイブおよびアーカイブ解除するために広く使用されています。
トラフィックIPアドレス	インターネットサービスを公開するために使用するIPアドレスで、通常はパブリックDNSエントリに紐付けられます。
トラフィックIPグループ	複数のトラフィックIPアドレスから成る論理グループで、一部または全てのTraffic Managerにまたがります。これらのTraffic Managerは、グループ内のIPアドレスを常時使用可能にするためのネゴシエーションを行います。
Traffic Manager	Traffic Managerを実行し、サービスを管理するフロントエンドサーバ。
トラフィッククラスタ	無制限数のアクティブおよびパッシブTraffic Managerを配備する機能。これにより、複数の障害が発生しても可用性を確保し、必要に応じてTraffic Managerクラスタを拡張することができます。N+M冗長性とも言います。
TrafficScript	Traffic Managerの一連のソリューションで使用するルールを作成するためのスクリプティング言語。
誘導手段	DoS攻撃やDDoS攻撃を行うための手段。
仮想サーバ	Traffic Managerとインターネット間のインターフェース。仮想サーバはいずれかのサービスを管理します。インターネットからリクエストを受信し、場合によってはSSL復号化を行い、その後ルールを適用してプールに割り当てます。

用語	意味
XML	Extensible Markup Language。構造化データの保存および交換に使用されます。
XPath	XMLドキュメントの各部分のアドレス指定を行うための言語。 TrafficScriptルールと組み合わせて使用することにより、ドキュメントのコンテンツに基づいて決定を行うことができます。
\$ZEUSHOME	Traffic Managerのインストール場所。デフォルトでは、Traffic Manager ソフトウェア製品の場合は/usr/local/zeus、アプライアンスとクラウドイ ンスタンスの場合は/opt/zeusが使用されます。

ソフトウェア使用許諾契約書

本章では、コアTraffic Managerソフトウェアで使用されているサードパーティソフトウェアのライセンス許諾契約書の一覧を示します。

Traffic Managerアプライアンスおよびクラウド製品で使用されているオペレーティングシステムレイヤに関係するサードパーティソフトウェアのライセンス許諾契約書については、代わりに、IvantiのWebサイト (www.ivanti.com)にある『Pulse Secure Virtual Traffic Manager: Appliance License Acknowledgements』を参照してください。

Berkeley DBコード(バージョン1.85)の使用許諾条件

以下の使用許諾条件は、Traffic Managerが紐付けられているlibdbコードに関するものです。このコードのソースは以下のサイトで参照できます。 <http://www.sleepycat.com/update/snapshot/db.1.85.tar.gz>

Copyright (c) 1991, 1993 The Regents of the University of California. 著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。

バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。

本ソフトウェアの機能または使用に言及する全ての広告物には、以下の情報を表記する必要があります。

本製品には、カリフォルニア大学バークレー校およびその貢献者により開発されたソフトウェアが含まれています。

事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、当大学名および貢献者の氏名を使用することはできません。

本ソフトウェアは、REGENTSおよび貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、REGENTSまたは貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯およびに基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

RSA PKCS11

本ソフトウェアのコピーまたは使用は、本ソフトウェアまたは本機能に言及する全てのものに「RSA Security Inc. PKCS #11 Cryptographic Token Interface (Cryptoki)」と明記することを条件に許諾されます。

また、派生物の作成および使用についても、派生物に言及する全てのものに「derived from the RSA Security Inc. PKCS #11 Cryptographic Token Interface (Cryptoki)」と明記することを条件に許諾されます。

OpenLDAPの使用許諾条件

以下の使用許諾条件は、Traffic Managerが紐付けられているlibldapコードに関するものです。このコードのソースは以下のサイトで参照できます。<http://ftp.openldap.org/pub/OpenLDAP/openldap-release/openldap-2.4.47.tgz>

バージョン: 2.4.47

ライセンス概要: OpenLDAP Public License

著作権:

OpenLDAP Public License

バージョン2.8、2003年8月17日

本ソフトウェアおよび付随するドキュメント(以下「ソフトウェア」)の再配布および使用は、変更の有無を問わず、以下の条件が満たされていることを条件に許諾されます。

1. ソース形式での再配布物には、著作権の宣言文および情報を含める必要があります。

2. バイナリ形式での再配布物には、該当する著作権の宣言文および情報、この条件リスト、および以下の免責事項を再配布物とともに提供されるドキュメントおよび/またはその他資料に複製する必要があります。

3. 再配布物には、一切の変更を加えずに本ドキュメントのコピーを含める必要があります。

OpenLDAP Foundationは、本使用許諾条件を改訂することがあります。各改訂は、バージョン番号によって管理されます。本ソフトウェアは、使用許諾条件の本改訂または以降の改訂の条件に従って使用することができます。

本ソフトウェアは、OPENLDAP FOUNDATIONおよび貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、OPENLDAP FOUNDATION、その貢献者、または本ソフトウェアの作成者もしくは所有者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

事前に書面による明確な許諾を得ない限り、広告内、あるいは本ソフトウェアの販売、使用、またはその他取引を促進する目的で、作成者および著作権所有者の名称を使用することはできません。本ソフトウェアの著作権は、常に著作権所有者が留保します。

OpenLDAPは、OpenLDAP Foundationの登録商標です。

Copyright 1999-2003 The OpenLDAP Foundation, Redwood City, California, USA. 著作権所有。本ドキュメントをコピーし、一切の変更を加えずに本ドキュメントのコピーを配布することを許可します。

PCRE2の使用許諾条件

以下の使用許諾条件は、Traffic Managerが紐付けられているpcre2コードに関するものです。

バージョン: 10.39

ライセンス概要: BSD + 免責

著作権:

PCRE2とは、正規表現をサポートするための関数のライブラリです。その構文とセマンティックは、Perl 5言語の正規表現に可能な限り近づけられています。

PCRE2のリリース10.00以降は、以下に規定する「BSD」ライセンスの規約に従い配布され、特定のバイナリ再配布に関して免責事項が1つあります。「doc」ディレクトリで提供されるPCRE2のドキュメントは、ソフトウェア自体と同じ条件に基づいて配布されます。testdata directoryディレクトリ内のデータは著作権保護されず、パブリックドメインに属します。

基本ライブラリ関数はCで記述され、独立したものとなっています。配布ファイルには、パターンマッチングを最適化するために使用できるJIT(Just-In-Time)コンパイラも含まれています。これは、ライブラリの作成時に省略できるオプションの機能です。

基本ライブラリ関数

作成者: Philip Hazel

電子メールのローカル部: Philip.Hazel

電子メールのドメイン: gmail.com

英国ケンブリッジ、ケンブリッジ大学コンピューティングサービス
学科の元教授。

Copyright (c) 1997-2021 University of Cambridge
著作権所有。

PCRE2 JITコンパイルのサポート

作成者: Zoltan Herczeg

電子メールのローカル部: hzmester: hzmester

電子メールのドメイン: freemail.hu

Copyright(c) 2010-2021 Zoltan Herczeg
著作権所有。

スタックレスJITコンパイラ

作成者: Zoltan Herczeg

電子メールのローカル部: hzmester: hzmester

電子メールのドメイン: freemail.hu

Copyright(c) 2009-2021 Zoltan Herczeg
著作権所有。

「BSD」ライセンス

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

- ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
- バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
- 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、ケンブリッジ大学の名称および貢献者の氏名を使用することはできません。

本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権所有者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

バイナリライブラリ等のパッケージに関する免責事項

BSDライセンスの第2条(バイナリの再配布について)は、一連のソフトウェア全てには適用されません。バイナリパッケージAにPCRE2が含まれている場合は条件が適用されますが、パッケージBがパッケージAを含むソフトウェアである場合、PCRE2を別途使用していない限り、パッケージBに第2条は適用されません。

Yahoo!の使用許諾条件UIライブラリ

Copyright (c) 2009, Yahoo! Inc.著作権所有。

ソースおよびバイナリ形式での本ソフトウェアの再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。

バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。

事前に書面によるYahoo! Inc.の明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、Yahoo! Inc.の名称およびその貢献者の氏名を使用することはできません。

本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

ssleay暗号化ライブラリの使用許諾条件

以下の使用許諾条件は、Pulse Secure Virtual Traffic Managerで使用されているSHA-1およびMD5アルゴリズムの実装に関するものです。

Copyright (C) 1997 Eric Young (eay@cryptsoft.com)。著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。

3. 本ソフトウェアの機能または使用に言及する全ての広告物には、以下の情報を表記する必要があります。「本製品には、Eric Young (eay@cryptsoft.com)により作成された暗号化ソフトウェアが含まれています。」使用するライブラリのルーチンが暗号化に関連するものでない場合は、「暗号化」という語を省略することができます。
4. appsディレクトリ(アプリケーションコード)にあるWindows固有のコード(またはその派生物)を含める場合、次の情報を含める必要があります:「本製品には、Tim Hudson (tjh@cryptsoft.com)により作成されたソフトウェアが含まれています。」

本ソフトウェアは、ERIC YOUNGにより「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、作成者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

libxml2およびlibxsltの使用許諾条件

以下の使用許諾条件は、MIT Licenseによる許諾のもとPulse Secure Virtual Traffic Managerで使用されているXMLパーサおよびツールキットライブラリに関するものです。

libxml2 2.9.12

libxslt 1.1.34

libxml2 の使用権許諾

ソースコード(hash.c、list.c、類似したライセンスの対象となるが著作権表示が異なるtrioファイルなど)に特記のある場合を除き、全てのファイルには以下が適用されます。

Copyright (C) 1998-2012 Daniel Veillard.著作権所有。

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

libexsltを除くlibxsltの使用権許諾

Copyright (C) 2001-2002 Daniel Veillard.著作権所有。

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

本通知に記載がある場合を除き、Daniel Veillardから事前に書面の許可を得ない限り、Daniel Veillardの氏名は、本ソフトウェアの販売、使用またはその他の取引を推進するために、広告またはその他で使用してはなりません。

libxsltの使用権許諾

Copyright (C) 2001-2002 Thomas Broyer, Charlie Bozeman and Daniel Veillard.

著作権所有。

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であると問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

本通知に記載がある場合を除き、作成者から事前に書面の許可を得ない限り、作成者の氏名は、本ソフトウェアの販売、使用またはその他の取引を推進するために、広告またはその他で使用してはなりません。

Java Servlet APIの使用許諾条件

Pulse Secure Virtual Traffic Managerで使用されているJava Servlet APIには、Apache Jakarta Projectのコードが含まれており、ASFライセンスバージョン2.0により使用が許諾されています。

Copyright 1999-2007, The Apache Software Foundation

Apache Licenseバージョン2.0 (以下「License」)のもと使用が許諾されています。お客様は、本ファイルを使用するときは、必ず以下のLicenseに従う必要があります。Licenseのコピーは<http://www.apache.org/licenses/LICENSE-2.0>から入手できます。

適用法で義務付けられているか書面により合意がない限り、Licenseのもと配布されるソフトウェアは、明示であると暗黙であるとを問わず、何ら保証または条件のない状態で「現状のまま」配布されます。Licenseにおいて許諾または制限を規制する特定言語については、Licenseを参照してください。

expat XMLパーサの使用許諾条件

以下の使用許諾条件は、Pulse Secure Virtual Traffic Managerで使用されているexpat XMLパーサ(expat 2.4.6)に関するものです。

Copyright (c) 1998 - 2000 Thai Open Source Software Center Ltd and Clark Cooper

Copyright (c) 2001 - 2019 Expat maintainers.

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

MooToolsの使用許諾条件

Pulse Secure Virtual Traffic Managerには、オープンソースMIT Licenseによる許諾のもとMooTools JavaScriptフレームワークのコードが含まれています。

MooTools 1.2

Copyright (c) 2006-2009 Valerio Proietti

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であると問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

OpenLayersの使用許諾条件

Pulse Secure Virtual Traffic Managerには、MetaCarta、Rico、およびXMLHttpRequest.jsのコードが含まれています。以下の著作権情報が適用されます。

Copyright (c) 2005-2008 MetaCarta, Inc.

著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に(以下の免責事項に記載された制限に従って)許諾されます。

- ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
- バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
- 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、MetaCarta, Inc.の名称および貢献者の氏名を使用することはできません。

本使用許諾では、明示であると暗黙であるとを問わず、いかなる第三者の著作権に対する使用許諾も与えられません。本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権所有者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

Ricoの一部を含む <<http://openrico.org/>>

Copyright 2005 Sabre Airline Solutions

Apache Licenseバージョン2.0 (以下「License」)のもと使用が許諾されています。お客様は、本ファイルを使用するときは、必ず以下のLicenseに従う必要があります。Licenseのコピーは以下のサイトから入手できます。

<http://www.apache.org/licenses/LICENSE-2.0>

適用法で義務付けられているか書面により合意がない限り、Licenseのもと配布されるソフトウェアは、明示であると暗黙であるとを問わず、何ら保証または条件のない状態で「現状のまま」配布されます。Licenseにおいて許諾または制限を規制する特定言語については、Licenseを参照してください。

XMLHttpRequest.jsを含む <<http://code.google.com/p/xmlhttprequest/>>

Copyright 2007 Sergey Ilinsky (<http://www.ilinsky.com>)

Apache Licenseバージョン2.0 (以下「License」)のもと使用が許諾されています。お客様は、本ファイルを使用するときは、必ず以下のLicenseに従う必要があります。Licenseのコピーは以下のサイトから入手できます。

<http://www.apache.org/licenses/LICENSE-2.0>

rsyncの使用許諾条件

Pulse Secure Virtual Traffic Managerには、高速リモートファイルコピープログラムであるrsyncのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 2.6.9

ライセンス概要: GPL v2

著作権:

rsyncは、当初Andrew Tridgellにより作成され、世界中の多くの開発者により改善が行われてきました。rsyncは、<http://www.fsf.org/licenses/old-licenses/gpl-2.0.html>で公開されているGNU General Public Licenseの条項に従って使用、変更、および再配布することができます。

mod_imap.cの使用許諾条件

Pulse Secure Virtual Traffic Managerには、mod_imap.cのコードが含まれています。mod_imap.cは、Apache HTTP Server (<http://httpd.apache.org>)によって作成され、Apache Software Licenseバージョン1.1のもと使用が許諾されています。以下の著作権情報が適用されます。

Apache Software Licenseバージョン1.1

Copyright (c) 2000 The Apache Software Foundation。著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 再配布物にエンドユーザ向けドキュメントを含める場合は、以下の情報を含める必要があります。

「本製品には、Apache Software Foundation (<http://www.apache.org/>)により開発されたソフトウェアが含まれています。」

または、ソフトウェア自体の中でこのようなサードパーティ情報が通常表示される箇所にこの情報を表示することもできます。

4. 事前に書面による許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、「Apache」および「Apache Software Foundation」の名称を使用することはできません。書面による許諾については、apache@apache.orgにお問い合わせください。
5. 事前に書面によるApache Software Foundationの許諾を得ない限り、本ソフトウェアから派生した製品の名称を「Apache」とすることはできません。また、その製品名に「Apache」を含めることもできません。

本ソフトウェアは、「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、APACHE SOFTWARE FOUNDATIONまたはその貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

Antlrおよびlibantlrの使用許諾条件

Pulse Secure Virtual Traffic Managerには、JavaScriptのレキサジェネレータおよびランタイムライブラリであるAntlrおよびlibantlrのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 3.4

ライセンス概要: BSD

著作権:

Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC

<http://www.temporal-wave.com>

<http://www.linkedin.com/in/jimidle>

著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、作成者の名称を使用することはできません。

本ソフトウェアは、作成者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、作成者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

es3-grammarの使用許諾条件

Pulse Secure Virtual Traffic Managerには、es3-grammarを使用して生成されたコードが含まれています。es3-grammarは、ANTLR 3用の完全なECMAScript 3 grammarです。以下の著作権情報が適用されます。

バージョン: N/A

ライセンス概要: BSD

著作権:

Copyright (c) 2008-2010, Xebic Research B.V. 著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、作成者の名称を使用することはできません。

本ソフトウェアは、作成者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、作成者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

jsoncppの使用許諾条件

Pulse Secure Virtual Traffic Managerには、設定ファイルの解析に使用されるJSONライブラリであるjsoncppのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 1.9.4

ライセンス概要: MIT

著作権:

Copyright (c) 2007-2010 Baptiste Lepilleur and The JsonCpp Authors

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

libjpegの使用許諾条件

Pulse Secure Virtual Traffic Managerには、JPEG操作ライブラリであるlibjpegのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 9c

ライセンス概要: IJGライセンス

著作権:

本ソフトウェア、その品質、正確性、商品性、または特定目的への適合性に関して、明示であると暗黙であることを問わず作成者は何ら保証も明言もしません。本ソフトウェアは、「現状のまま」提供されるものであり、その品質と正確性に関する全てのリスクは購入者とその利用者が負うものとします。

本ソフトウェアの著作権: Copyright (C) 1991-2018, Thomas G. Lane, Guido Vollbeding. All Rights Reserved (以下の場合を除く)

本ソフトウェア(またはその一部)を任意の目的に無償で使用、コピー、変更、配布する権利が、以下の条件に従って許諾されます。

1. 本ソフトウェアのソースコードのいかなる部分を配布する場合でも、著作権情報および無保証の但し書きとともに本READMEファイルを含める必要があります。また、元のファイルに対する追加、削除、変更がある場合には、付属文書にてそれを明記しなければなりません。
2. 実行可能コードのみを配布する場合には、付属文書に「this software is based in part on the work of the Independent JPEG Group (本ソフトウェアには、Independent JPEG Groupで作成されたコードの一部が利用されています)」と記載する必要があります。
3. 本ソフトウェアの使用許諾は、望ましくない結果に対する責任を全てユーザが負う場合に限って付与されます。作成者はかかる損害に対して一切の責任を負いません。

以上の条件は、変更されないライブラリのみならず、IJGコードから派生した、またはIJGコードに基づく全てのソフトウェアに適用されます。弊社の作成物を使用する場合は、ご連絡いただく必要があります。

IJGの作成者名または会社名を、本ソフトウェアまたはそれに派生する製品の広告宣伝に使用することは一切許可されません。本ソフトウェアは、「the Independent JPEG Group's software (Independent JPEG Groupのソフトウェア)」と称する必要があります。

本ソフトウェアは、全ての保証または賠償請求を製品ベンダーが負う限りにおいて、商用製品の基盤として使用することを許諾または推奨されるものです。

UNIX設定スクリプト「configure」は、GNU Autoconfにより生成されました。その著作権はFree Software Foundationが保有していますが、自由に配布できます。サポートスクリプト(config.guess、config.sub、ltmain.sh)についても同様です。もうひとつのスクリプト、install-shの著作権はX Consortiumが保有していますが、これも自由に配布が可能です。

IJGのディストリビューションには、GIFファイルを読み書きするコードが正式に含まれていました。Unisys LZW 特許(現在は期限切れ)の侵害を回避するために、GIFの読み取りサポートは完全に除外され、GIFライタは非圧縮GIFを生成するように簡略化されています。この手法で、LZWアルゴリズムは使用しません。生成されるGIFファイルは通常より大きくなりますが、全ての標準GIFデコーダで読み取ることができます。

libunwindの使用許諾条件

Pulse Secure Virtual Traffic Managerには、スタックのアンwindライブラリであるlibunwindのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 1.1

ライセンス概要: MIT

著作権:

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

Perl JSONライブラリの使用許諾条件

Pulse Secure Virtual Traffic Managerには、JSON (JavaScript Object Notation) 暗号化および復号化用のPerlモジュールであるPerl JSONライブラリのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 2.53

ライセンス概要: Perl Artistic License v.1

著作権:

前文

本書は、パッケージの使用者に事実上の慣習的方法においてパッケージを使用および配布する権利、および合理的変更を行う権利を付与しながらも、著作権所有者がパッケージ開発に関する一定の芸術的管理権を維持するように、パッケージの複製が許容される条件を示すことを目的としています。

定義:

「パッケージ」とは、著作権所有者が配布するファイルの集まり、および当該のファイルの集まりに対してテキストの変更を行うことで作成された派生物を指します。

「標準バージョン」とは、変更されていないパッケージ、または以下に規定する著作権所有者の意思に従って変更されたパッケージを指します。

「著作権所有者」とは、パッケージの著作権で指名された全ての人物を指します。「お客様」とは、あなたが本パッケージを複製または配布しようとしている場合、あなた自身を指します。

「合理的複製料」とは、メディア費用、複製費用、関係者の拘束時間などに基づいてお客様が正当化できる料金を指します(お客様は、著作権所有者に対してではなく、当該料金を負担する市場としての一般的コンピュータ関係社会に対してのみ当該複製料を正当化するよう求められます)。

「自由に利用可能」とは、商品自体に対して料金が課されないことを指しますが、商品の取り扱いに関する料金が課されることがあります。また、商品の受領者は、当該商品を受領したときと同じ条件に基づいて当該商品を再配布できます。

1. お客様は、元の著作権情報および関連する免責事項の全てを複製する限りにおいて、本パッケージの標準バージョンのソース形式の逐語的複製を無制限に作成および配布することができます。
2. お客様は、バグの修正、移植性の修正、およびパブリックドメインまたは著作権所有者に由来するその他の変更を加えることができます。そのような方法において変更されたパッケージは、標準バージョンと見なされるものとします。
3. いつどのような変更をファイルに対して行ったかを記述した明示的な情報を変更された各ファイルに挿入し、かつ以下の少なくとも1つに該当する限り、本パッケージの複写を、形を問わず別の方法で変更することができます。

- a) お客様が行った変更事項をパブリックドメインに掲載する、またはそれらの変更事項を自由に利用可能な状態にする。これには、Usenetまたは同等の媒体へ前述の変更事項を掲示する、uunet.uu.netなどの大手アーカイブサイトへ変更事項を掲載する、お客様が行った変更事項を著作権所有者が本パッケージの標準バージョン取り込むことを許可するなどの方法が該当します。
 - b) 変更したパッケージをお客様の会社または組織内においてのみ使用する。
 - c) 標準の実行ファイルと名前が競合しないように、非標準の全ての実行ファイルの名前を変更し、標準の実行ファイルと併せて提供する。また、非標準の実行ファイルごとに、標準バージョンとの差異を明確に記した個別のマニュアルページを提供する。
 - d) 著作権所有者と配布に関するその他の取り決めを行う。
4. 以下の少なくとも1つに該当する限り、お客様は本パッケージのプログラムをオブジェクトコードまたは実行ファイルの形式で配布することができます。
- a) 実行ファイルおよびライブラリファイルの標準バージョンを、(マニュアルページまたはそれと同等のものにおいて)使用法とともに標準バージョンを入手できる場所において配布する。
 - b) コンピュータが解読できるパッケージのソースにお客様が行った変更事項を加えて配布する。
 - c) 非標準の実行ファイルに非標準の名前を付け、標準バージョンを入手できる場所において、その差異を使用法とともにマニュアルページ(または同等のもの)に明確に記す。
 - d) 著作権所有者と配布に関するその他の取り決めを行う。
5. お客様は、本パッケージの任意の配布に対して合理的複製料を課することができます。お客様は、本パッケージのサポートに対して任意の料金を課することができます。お客様は、本パッケージ自体に対する料金を課することはできません。ただし、お客様が本パッケージをお客様自身の製品として広告しない限り、より大規模な(できる限り商用の)ソフトウェアの配布の一環として、その他の(できる限り商用の)プログラムと併せて本パッケージを配布することができます。お客様は、本パッケージのインタープリタを実行ファイルの内部に(リンク付けによって)埋め込むことができます。インタープリタの完全に標準のバージョンが同様に埋め込まれている場合、アグリゲーションに似せた形式で解釈される必要があります。

6. 本パッケージへのインプットとして供給される、または本パッケージからのアウトプットとして生成されるスクリプトおよびライブラリファイルは、自動的に本パッケージの著作権の影響を受けることはなく、それらを生成した人物に属するものとし、それらを商業的に販売したり、本パッケージと集約したりすることができます。そのようなスクリプトまたはライブラリファイルが、バイナリ実行可能イメージを生成するいわゆる「undump」または「unexec」の方法で本パッケージに集約される場合、お客様がそのような実行可能イメージを本パッケージの標準バージョンであると明言しない限り、そのようなイメージの配布が本パッケージの配布と解釈されたり、第3および第4段落の制限が適用されたりすることはないものとします。
7. 本パッケージで定義されている言語のサブルーチンおよびバリアブルをエミュレートするために、お客様によって供給され、本パッケージにリンクされるCサブルーチン(またはその他の言語で同等にコンパイルされたサブルーチン)は、当該ルーチンが、当該言語に対する回帰テストが失敗に終わる原因となるような方法で当該言語を変更しない限り、本パッケージの一部と見なされることはなく、第6段落のインプットに相当します。
8. 本パッケージの使用が埋め込まれている場合、すなわち、本パッケージのインターフェースが商業的配布のエンドユーザに見えるようにする公然たる試みが一切行われらない限り、商業的配布を伴う本パッケージのアグリゲーションは常に認められます。そのような使用は、本パッケージの配布とは解釈されません。
9. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、著作権所有者の名称を使用することはできません。
10. 本パッケージは、「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。

OpenSSLの使用許諾条件

Pulse Secure Virtual Traffic Managerには、暗号化およびSSL/TLSツールキットであるOpenSSLライブラリのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 1.1.1n

ライセンス概要: BSDスタイル

著作権情報:

ライセンス交付

OpenSSLツールキットはダブルライセンスを受けています。つまり、OpenSSLライセンスの条件と元のSSLLeayライセンスがツールキットに適用されます。実際のライセンス文については、以下を参照してください。実際に、両ライセンスともBSDスタイルのオープンソースライセンスです。OpenSSLに関するライセンス交付については、openssl-core@openssl.orgにお問い合わせください。

OpenSSLライセンス

Copyright (c) 1998-2019 The OpenSSL Project. 著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 本ソフトウェアの機能または使用に言及する全ての広告物には、以下の情報を表記する必要があります。
「本製品には、OpenSSL Projectにより開発されたOpenSSLツールキットを使用するためのソフトウェアが含まれています。(<http://www.openssl.org/>)”
4. 事前に書面による許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、「OpenSSLツールキット」および「OpenSSL Project」の名称を使用することはできません。書面による許諾については、openssl-core@openssl.orgにお問い合わせください。
5. 事前に書面によるOpenSSL Projectの許諾を得ない限り、本ソフトウェアから派生した製品の名称を「OpenSSL」とすることはできません。また、その製品名に「OpenSSL」を含めることもできません。
6. いかなる形式の再配布も、以下の情報を表記する必要があります。
「本製品には、OpenSSL Projectにより開発されたOpenSSLツールキットを使用するためのソフトウェアが含まれています。(<http://www.openssl.org/>)」

本ソフトウェアは、OpenSSL Projectにより「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、OpenSSLプロジェクトまたは貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

=====
==

「本製品には、Eric Young (eay@cryptsoft.com)により作成された暗号化ソフトウェアが含まれています。本製品には、Tim Hudson (tjh@cryptsoft.com)により作成されたソフトウェアが含まれています。

オリジナルSSLeyライセンス

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)

著作権所有。

本パッケージは、Eric Young (eay@cryptsoft.com)により作成されたSSL実装です。この実装は、Netscape SSLに適合するように作成されています。

以下の条件が順守される限り、本ライブラリは営利使用および非営利使用において無料です。以下の条件は、本配布に使用される全てのコード(SSLコードだけでなく、RC4、RSA、lhash、DESなどのコード)に適用されます。本配布に含まれるSSLドキュメントは、著作権所有者がTim Hudson (tjh@cryptsoft.com)である場合を除き、同様の著作権条件が適用されます。

著作権はEric Youngが保有するため、コード内のいかなる著作権情報も削除してはなりません。本パッケージが製品内で使用される場合、使用されるライブラリの一部の作成者としての属性がEric Youngに付与されます。これは、プログラム開始処理のテキストメッセージの形式、またはパッケージに付属するドキュメント内(オンラインまたはテキスト)に記します。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、著作権情報、この条件リスト、および以下の免責事項を含める必要があります。

2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 本ソフトウェアの機能または使用に言及する全ての広告物には、以下の情報を表記する必要があります。
「本製品には、Eric Young (eay@cryptsoft.com)により作成された暗号化ソフトウェアが含まれています。」使用するライブラリのルーチンが暗号化に関連するものでない場合は、「暗号化」という語を省略することができます。
4. appsディレクトリ(アプリケーションコード)にあるWindows固有のコード(またはその派生物)を含める場合、次の情報を含める必要があります:
「本製品には、Tim Hudson (tjh@cryptsoft.com)により作成されたソフトウェアが含まれています。」

本ソフトウェアは、ERIC YOUNGにより「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、作成者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

パブリックに利用可能なバージョンまたは本コードの派生物に対するライセンスおよび配布条件を変更することはできません。すなわち、本コードを単純に複製し、別の配布ライセンス(GNU公衆ライセンス)の適用を受けることはできません。

WebPの使用許諾条件

Pulse Secure Virtual Traffic Managerには、イメージフォーマットであるWebPのコードが対応するコンバージョンライブラリとともに含まれています。以下の著作権情報が適用されます。

バージョン: 0.3.1

ライセンス概要: BSDスタイル

著作権情報:

Copyright (c) 2010, Google Inc. 著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

- ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
- バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
- 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、Googleの名称および貢献者の氏名を使用することはできません。

本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

Flexの使用許諾条件

Pulse Secure Virtual Traffic Managerには、高速な字句解析ツールであるFlexにより生成されるコードが含まれています。以下の著作権情報が適用されます。

バージョン: 2.5.37

ライセンス概要: BSDスタイル

著作権情報:

Flexは、エネルギー省との契約に基づき運営されるLawrence Berkeley (Livermoreではありません) Laboratoryにより考案されたため若干変更されている、BSDソフトウェアに使用される著作権を保有します。

Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007 The Flex Project.

Copyright (c) 1990, 1997 The Regents of the University of California.

著作権所有。

本コードは、Vern Paxsonによりバークレー校に寄与されたソフトウェアの派生物です。

アメリカ合衆国政府は、アメリカ合衆国エネルギー省およびカリフォルニア大学との間で締結された契約(No. DE-AC03-76SF00098)に準じて本著作物の権利を保有します。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。

事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、当大学名および貢献者の氏名を使用することはできません。

本ソフトウェアは、「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。

これは、基本的には「この情報を削除したり、大学(またはflexの作成者)の名前を悪用したりすることを除けば、本ソフトウェアを使用して何を行ってもよい」ことを意味します。

「flex.skl」スキヤナスケルトンには著作権情報がないことに注意してください。flexを使用して生成されるスキヤナを使用して行うことに制限はありません。それらの行為に対しては、上記の著作権に制約されることもありません。

zlibの使用許諾条件

Pulse Secure Virtual Traffic Managerには、圧縮ライブラリであるzlibのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 1.2.8

著作権情報:

zlib.h -- 「zlib」汎用圧縮ライブラリのインターフェース、バージョン1.2.8、2013年4月28日

Copyright (C) 1995-2013 Jean-loup Gailly and Mark Adler

本ソフトウェアは、「現状のまま」提供されるものであり、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合にも、作成者は、本ソフトウェアの使用に起因する損傷に責任を負いません。

以下の制限を条件として、商用アプリケーションを含め任意の目的に本ソフトウェアを使用し、自由に修正および再配布する許可があらゆる者に与えられます。

1. 本ソフトウェアの元の作成者について偽りの表記をしてはならず、元のソフトウェアを自ら開発したと主張してはなりません。製品内で本ソフトウェアを使用する場合、製品ドキュメントでその旨を記載することが望まれますが、必須ではありません。
2. 修正したソースバージョンは、その旨を明記しなければならず、元のソフトウェアとして偽って表記してはなりません。
3. 本通知は、配布するソースから削除し、または修正することはできません。

Jean-loup Gailly Mark Adler

jloup@gzip.org madler@alumni.caltech.edu

XML::Twigの使用許諾条件

Pulse Secure Virtual Traffic Managerには、XMLデータの処理に使用されるPerlライブラリであるXML::Twigのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 3.48

ライセンス概要: Perl 5 License (Artistic 1 & GPL 1)

著作権情報:

著作権

Copyright (c) 1999-2012, Michel Rodriguez. 著作権所有。

このライブラリはフリーソフトウェアであり、Perl自体と同じ条件に基づいて

このライブラリを再配布および/または変更することができます。

MIT Kerberosの使用許諾条件

Pulse Secure Virtual Traffic Managerには、ネットワーク認証プロトコルであるMIT Kerberosのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 1.18.3

ライセンス概要: MIT

著作権情報:

Copyright (C) 1985-2020 by the Massachusetts Institute of Technology.

著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

- ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
- バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。

本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権所有者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

本ソフトウェアのダウンロードは、合衆国輸出規則(EAR) 15 CFR 730-774が課されるアメリカ合衆国からの暗号化ソフトウェアの輸出になることがあります。その他の法令が適用されることもあります。必要とされるあらゆるライセンスをアメリカ合衆国政府から取得することも含め、適用される全ての輸出法令を順守する責任は、輸出しようとする個人または団体が負います。

アメリカ合衆国政府は、キューバ、イラン、北朝鮮、スーダン、シリア、およびそれらの諸国の住民および国民を含め(ただしこれらに限定されない)、一部の国および個人に対する暗号化ソースコードの輸出を禁止しています。

本ソフトウェア配布のドキュメント要素は、クリエイティブコモンズ表示-継承3.0非移植ライセンスを受けています。[\(http://creativecommons.org/licenses/by-sa/3.0/\)](http://creativecommons.org/licenses/by-sa/3.0/)

個々のソースコードファイルは、MIT、Cygnus Support、Novell、OpenVision Technologies、Oracle、Red Hat、Sun Microsystems、FundsXpressおよびその他の著作権で保護されています。

Project Athena、Athena、Athena MUSE、Discuss、Hesiod、Kerberos、Moirā、およびZephyrは、マサチューセッツ工科大学(MIT)の商標です。事前に書面によるMITの許諾を得ない限り、これらの商標を営利使用することはできません。

「営利使用」とは、名前を製品やその他の営利目的の方法で使用することを指します。これは、営利目的の会社が情報の伝達を目的としてMITの商標に言及することを妨げるものではありません(ただし、そのような場合でもそれが商標であるという認識を得なければなりません)。

以下の著作権および許可情報は、kadmin/create、kadmin/dbutil、kadmin/passwd、kadmin/server、lib/kadm5、およびlib/rpcの一部にあるOpenVision Kerberos管理システムに適用されます。

Copyright, OpenVision Technologies, Inc., 1993-1996, 著作権所有。

警告: 後述のとおり、OpenVision Kerberos管理システムのソースコードを取り込むことは、以下の条件が受諾されたことを意味します。以下の条件に同意されない場合は、OpenVision Kerberos管理システムを取り込まないでください。

ソースコードおよびそこからコンパイルされたオブジェクトコードを、変更の有無を問わず自由に使用および配布できますが、このソースコードは「現状のまま」提供されるものであり、商品性および特定目的への適合性のいかなる保証(ただしこれらに限定されない)またはその他の保証を含め、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、OpenVisionは本ソースコードの使用、ソースコードの実行失敗、またはその他のいかなる理由を含め(ただしこれらに限定されない)、本合意により生じた利益の損失、データの喪失、代替品またはサービスの調達費用に対し、またはいかなる例外的、間接的、または必然的損傷に対し、一切責任を負いません。

OpenVisionは、提供されたソースコードに対するあらゆる著作権を保有します。またOpenVisionは、OpenVisionまたはサードパーティのいずれによって作成されたかを問わず、ソースコードの派生的著作物に対する著作権も保有します。派生的著作物が提供されたソースコードをベースに作成されている場合、OpenVisionの著作権情報が引き継がれる必要があります。

OpenVision Technologies, Inc.は、標準Kerberos 5配布の一部として本Kerberos管理システムをMITに提供しています。この提供は、Kerberos技術の継続的開発に対する当社の深い関与を示すと同時に、MITおよびKerberosコミュニティにより行われている価値ある取り組みへの感謝の意も表しています。

Matt Crawford (crawdad@fnal.gov)により提供される部分は、Universities Research Association, Inc.により運営されるフェルミ国立加速器研究所において、アメリカ合衆国エネルギー省との契約(DE- AC02-76CHO3000)に基づいて行われた取り組みです。

src/lib/cryptoの部分は、以下の著作権を有します。

Copyright © 1998 by the FundsXpress, INC.

著作権所有。

本ソフトウェアをアメリカ合衆国から輸出する場合、アメリカ合衆国政府から特定ライセンスを取得しなければならないことがあります。輸出前に当該ライセンスを取得する責任は、輸出しようとする個人または組織が負います。

その制限内において、本ソフトウェアおよびそのドキュメントを無償で使用、複製、変更、および配布することは、その目的を問わず、前述の著作権情報があらゆる複製に記され、当該著作権情報および本許可情報の両方が解説文書に記され、さらにFundsXpress.の名称が事前に書面による明確な許諾を得ずに本ソフトウェアの配布に関する広告または広報で使用されない限り、本合意によって許諾されます。FundsXpressは、いかなる目的においても、本ソフトウェアの適合性に関して明言しません。本ソフトウェアは、「現状のまま」提供されるものであり、明示であると暗黙であるとを問わず、何ら保証されません。

本ソフトウェアは、「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。

src/lib/crypto/builtin/aesへのAES暗号化アルゴリズムの実装は、以下の著作権を有します。

Copyright © 2001, Dr Brian Gladman brg@gladman.uk.net, Worcester, UK.

著作権所有。

ライセンス条項

以下の場合、ソースおよびバイナリの両形式による本ソフトウェアの無料配布および使用は(変更の有無を問わず)許可されます。

1. 本ソースコードの配布に、前述の著作権情報、この条件リスト、および以下の免責事項が含まれている。

2. バイナリ形式での配布に、上記の著作権情報、この条件リスト、および以下の免責事項がドキュメントおよび/またはその他の関連資料に含まれている。
3. 書面による明確な許諾を得ないまま、本ソフトウェアを使用してビルドされた製品を宣伝するために著作権所有者の名前が使用されていない。

保証責任の排除

本ソフトウェアは、「現状のまま」提供されるものであり、正確性および目的に応じた適合性を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、あらゆる属性に関して何ら保証されません。

Red Hatにより提供される部分は、事前認証プラグインフレームワークおよびNSS暗号実装を含め、以下の著作権を有します。

Copyright © 2006 Red Hat, Inc.

Portions copyright © 2006 Massachusetts Institute of Technology

著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

- ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
- バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
- 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、Red Hat, Inc.の名称および貢献者の氏名を使用することはできません。

本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権所有者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

バンドリングされるVertoソースコードには、以下のライセンスが適用されます。

Copyright 2011 Red Hat, Inc.

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

クライアントに実装されるMS-KKDCPIは、以下の著作権を有します。

Copyright 2013.2014 Red Hat, Inc.

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。

2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。

本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権所有者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

以下のファイルを含む、src/lib/gssapiのGSSAPI-SPNEGOへのGSSAPI Mechglueの実装

lib/gssapi/generic/gssapi_err_generic.et

lib/gssapi/mechglue/g_accept_sec_context.c

lib/gssapi/mechglue/g_acquire_cred.c

lib/gssapi/mechglue/g_canon_name.c

lib/gssapi/mechglue/g_compare_name.c

lib/gssapi/mechglue/g_context_time.c

lib/gssapi/mechglue/g_delete_sec_context.c

lib/gssapi/mechglue/g_dsp_name.c

lib/gssapi/mechglue/g_dsp_status.c

lib/gssapi/mechglue/g_dup_name.c

lib/gssapi/mechglue/g_exp_sec_context.c

lib/gssapi/mechglue/g_export_name.c

lib/gssapi/mechglue/g_glue.c

lib/gssapi/mechglue/g_imp_name.c

lib/gssapi/mechglue/g_imp_sec_context.c
lib/gssapi/mechglue/g_init_sec_context.c
lib/gssapi/mechglue/g_initialize.c
lib/gssapi/mechglue/g_inquire_context.c
lib/gssapi/mechglue/g_inquire_cred.c
lib/gssapi/mechglue/g_inquire_names.c
lib/gssapi/mechglue/g_process_context.c
lib/gssapi/mechglue/g_rel_buffer.c
lib/gssapi/mechglue/g_rel_cred.c
lib/gssapi/mechglue/g_rel_name.c
lib/gssapi/mechglue/g_rel_oid_set.c
lib/gssapi/mechglue/g_seal.c
lib/gssapi/mechglue/g_sign.c
lib/gssapi/mechglue/g_store_cred.c
lib/gssapi/mechglue/g_unseal.c
lib/gssapi/mechglue/g_userok.c
lib/gssapi/mechglue/g_utils.c
lib/gssapi/mechglue/g_verify.c
lib/gssapi/mechglue/gssd_pname_to_uid.c
lib/gssapi/mechglue/mglueP.h
lib/gssapi/mechglue/oid_ops.c
lib/gssapi/spnego/gssapiP_spnego.h
lib/gssapi/spnego/spnego_mech.c

および以下の新規または変更済みファイルを含む、増分の初期実装

include/iprop_hdr.h

kadmin/server/ipropd_svc.c

lib/kdb/iprop.x

lib/kdb/kdb_convert.c

lib/kdb/kdb_log.c

lib/kdb/kdb_log.h

lib/krb5/error_tables/kdb5_err.et

slave/kpropd_rpc.c

slave/kproplog.c

には、以下のライセンスが適用されます。

Copyright © 2004 Sun Microsystems, Inc.

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

Kerberos V5には、カリフォルニア大学バークレー校で開発された、本著作権情報を含むドキュメントおよびソフトウェアが含まれています。

Copyright © 1983 Regents of the University of California.

著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、当大学名および貢献者の氏名を使用することはできません。

本ソフトウェアは、REGENTSおよび貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、REGENTSまたは貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯およびに基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

LDAPデータベースバックエンドを含め、Novell, Inc.により提供される部分には、以下のライセンスが適用されます。

Copyright © 2004-2005, Novell, Inc.

著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

- ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
- バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
- 事前に書面による明確な許諾を得ないまま、本ソフトウェアから派生した製品を宣伝または販売促進するために著作権所有者の名前が使用されていない。

本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙の保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権所有者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

サンディア国立研究所により提供され、ミシガン大学情報技術統合センターにより開発される部分には、PKINIT実装も含め、以下のライセンスが適用されます。

COPYRIGHT © 2006-2007

THE REGENTS OF THE UNIVERSITY OF MICHIGAN

ALL RIGHTS RESERVED

派生的著作物の使用、複製および作成、および本ソフトウェアおよび派生的著作物の再配布は、その目的を問わず、ミシガン大学の名称が事前に書面による明確な許諾を得ずに本ソフトウェアの配布での使用に関する広告または広報で使用されない限り許諾されます。本ソフトウェアの任意の部分の複製に、前述のミシガン大学の著作権情報およびその他の識別情報が含まれている場合、以下の免責事項も含まれていなければなりません。

本ソフトウェアは、「現状のまま」提供されるものであり、目的を問わずミシガン大学はその適合性に関して明言しません。また、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、ミシガン大学は何ら保証しません。ミシガン大学のREGENTSは、本ソフトウェアの使用によりまたはそれに関連して生じた請求に関して、例外的、間接的、偶発的、または必然的損傷に対し、そのような損傷の可能性をすでに知らされていたまたは今後知らされる場合でも、一切責任を負いません。

PKINITコードに含まれるpkcs11.hファイルには、以下のライセンスが適用されます。

Copyright 2006 g10 Code GmbH

Copyright 2006 Andreas Jellinghaus

本ファイルは無料ソフトウェアです。この情報が引き継がれている限り、作成者は特例として、変更の有無を問わず、本ファイルの複製および/または配布を無制限に許可します。

本ファイルは、有益であることを願って配布されますが、法律が許す範囲内で、商品性または特定目的への適合性の暗黙的保証を含め、何ら保証されません。

Apple Inc.により提供される部分には、以下のライセンスが適用されます。

Copyright 2004-2008 Apple Inc. 著作権所有。

本ソフトウェアをアメリカ合衆国から輸出する場合、アメリカ合衆国政府から特定ライセンスを取得しなければならないことがあります。輸出前に当該ライセンスを取得する責任は、輸出しようとする個人または組織が負います。

その制限内において、本ソフトウェアおよびそのドキュメントを無償で使用、複製、変更、および配布することは、その目的を問わず、前述の著作権情報があらゆる複製に記され、当該著作権情報および本許可情報の両方が解説文書に記され、さらにApple Inc.の名称が事前に書面による明確な許諾を得ずに本ソフトウェアの配布に関する広告または広報で使用されない限り、本合意によって許諾されます。Apple Inc.は、いかなる目的においても、本ソフトウェアの適合性に関して明言しません。本ソフトウェアは、「現状のまま」提供されるものであり、明示であると暗黙であるとを問わず、何ら保証されません。

本ソフトウェアは、「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。

src/util/supportおよびsrc/lib/krb5/unicodeへのUTF-8文字列操作の実装には、以下の著作権および許可情報が適用されます。

OpenLDAP Public License

バージョン2.8、2003年8月17日

本ソフトウェアおよび付随するドキュメント(以下「ソフトウェア」)の再配布および使用は、変更の有無を問わず、以下の条件が満たされていることを条件に許諾されます。

1. ソース形式での再配布物には、著作権の宣言文および情報を含める必要があります。
2. バイナリ形式での再配布物には、該当する著作権の宣言文および情報、この条件リスト、および以下の免責事項を再配布物とともに提供されるドキュメントおよび/またはその他資料に複製する必要があります。
3. 再配布物には、一切の変更を加えずに本ドキュメントのコピーを含める必要があります。

OpenLDAP Foundationは、本使用許諾条件を改訂することがあります。各改訂は、バージョン番号によって管理されます。本ソフトウェアは、使用許諾条件の本改訂または以降の改訂の条件に従って使用することができます。

本ソフトウェアは、OPENLDAP FOUNDATIONおよび貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、OPENLDAP FOUNDATION、その貢献者、または本ソフトウェアの作成者もしくは所有者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

事前に書面による明確な許諾を得ない限り、広告内、あるいは本ソフトウェアの販売、使用、またはその他取引を促進する目的で、作成者および著作権所有者の名称を使用することはできません。本ソフトウェアの著作権は、常に著作権所有者が留保します。

OpenLDAPは、OpenLDAP Foundationの登録商標です。

Copyright 1999-2003 The OpenLDAP Foundation, Redwood City, California, USA。著作権所有。本ドキュメントをコピーし、一切の変更を加えずに本ドキュメントのコピーを配布することを許可します。

src/lib/krb5/krbのマークテストプログラムは、以下の著作権を有します。

Copyright © 2006 Kungliga Tekniska Högskola

(Royal Institute of Technology, Stockholm, Sweden).

著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、KTHの名称および貢献者の氏名を使用することはできません。

本ソフトウェアは、KTHおよび貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、KTHまたは貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

OS Xで使用するKCM Mach RPC定義ファイルは、以下の著作権を有します。

Copyright © 2009 Kungliga Tekniska Högskola

(Royal Institute of Technology, Stockholm, Sweden).

著作権所有。

Portions Copyright © 2009 Apple Inc. 著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、当大学名および貢献者の氏名を使用することはできません。

本ソフトウェアは、大学および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、大学または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

src/lib/rpcおよびsrc/include/gssrpcへのRPC実装の一部は、以下の著作権および許可情報を有しています。

Copyright © 2010, Oracle America, Inc.

著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。

3. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、「Oracle America, Inc.」の名称および貢献者の氏名を使用することはできません。

本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権所有者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

Copyright © 2006,2007,2009 NTT (日本電信電話株式会社)。著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布には、上記の著作権情報、この条件リスト、および以下の免責事項を、変更されていないこのファイルの1行目として含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。

本ソフトウェアは、NTTにより「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、NTTは、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

Copyright 2000 by Carnegie Mellon University

All Rights Reserved

本ソフトウェアおよびそのドキュメントを無償で使用、複製、変更、および配布することは、その目的を問わず、前述の著作権情報があらゆる複製に記され、当該著作権情報および本許可情報の両方が解説文書に記され、さらにカーネギーメロン大学の名称が事前に書面による明確な許諾を得ずに本ソフトウェアの配布に関する広告または広報で使用されない限り、本合意によって許諾されます。

カーネギーメロン大学は、商品性および適合性のあらゆる暗黙的保証を含め、本ソフトウェアに関していかなる保証もしません。また、いかなる場合も、カーネギーメロン大学は、本ソフトウェアの使用または性能によりまたはそれに関連して生じた例外的、間接的または必然的損傷、または使用できないこと、データの喪失または利益の損失から生じるいかなる損傷に対して、契約の行為、過失またはその他の不法行為であるかどうかに関わらず、一切責任を負いません。

Copyright © 2002 Naval Research Laboratory (NRL/CCS)

本ソフトウェアおよびそのドキュメントを使用、複製、変更、および配布することは、著作権情報および本許可情報の両方がソフトウェアの複製、派生的著作物または変更されたバージョン、およびそれに関連する部分の全てにおいて記される限り、本合意によって許諾されます。

NRLは、本ソフトウェア「現状のまま」の状態での無料使用を許可するとともに、本ソフトウェアの使用により生じるあらゆる損傷に対して一切責任を負いません。

インターネットRFCからの引用部分には、以下の著作権情報が適用されます。

Copyright (C) The Internet Society (2006).

本ドキュメントは、BCP 78に含まれる権利、ライセンスおよび制限の対象であり、そこで説明されている場合を除き、作成者がそれらの全権利を有します。

本ドキュメントおよびここに記されている情報は、「現状のまま」を基本として提供されるものであり、貢献者、その貢献者が代表するまたは支援を受ける組織(該当する場合)、インターネット社会、およびインターネットエンジニアリング作業部会は、個々に記されている情報の使用がいかなる権利も侵害しないという保証、または商品性または特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、いかなる保証もしません。

Copyright © 1991, 1992, 1994 by Cygnus Support.

本ソフトウェアおよびそのドキュメントを無償で使用、複製、変更、および配布することは、その目的を問わず、前述の著作権情報があらゆる複製に記され、当該著作権情報および本許可情報の両方が解説文書に記される限り、本合意によって許諾されます。Cygnus Supportは、いかなる目的においても、本ソフトウェアの適合性に関して明言しません。本ソフトウェアは、「現状のまま」提供されるものであり、明示であると暗黙であることを問わず、何ら保証されません。

Copyright © 2006 Secure Endpoints Inc.

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

Fortunaを模した疑似乱数生成器の実装の一部には、以下の情報が適用されます。

Copyright © 2005 Marko Kreen

著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。

本ソフトウェアは、作成者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、作成者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

Copyright © 1994 by the University of Southern California

本ソフトウェアをアメリカ合衆国から輸出する場合、アメリカ合衆国政府から特定ライセンスを取得しなければならないことがあります。輸出前に当該ライセンスを取得する責任は、輸出しようとする個人または組織が負います。

その制限内において、本ソフトウェアおよびそのドキュメントをソースまたはバイナリ形式で複製、変更、および配布することは、そのような配布または使用に関連するあらゆるドキュメントまたはその他の資料において、本ソフトウェアは南カリフォルニア大学が開発したものであることが認知されている限り、本合意によって許諾されます。

保証責任の排除。本ソフトウェアは、「現状のまま」で提供されます。南カリフォルニア大学は、明示であると暗黙であるとを問わず、一切の明言または保証を行いません。例として(ただしこれに限定されない)、南カリフォルニア大学は、商品性および特定目的への適合性を一切明言または保証しません。南カリフォルニア大学は、KSUソフトウェアの使用者または配布者による請求に関するいかなる直接的、間接的、または必然的損傷に対しても、一切責任を負いません。

Copyright © 1995

The President and Fellows of Harvard University

本コードは、Jeremy Rassenによりハーバード大学に寄与されたソフトウェアの派生物です。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。

2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 本ソフトウェアの機能または使用に言及する全ての広告物には、以下の情報を表記する必要があります。

本製品には、カリフォルニア大学バークレー校およびその貢献者により開発されたソフトウェアが含まれています。
4. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、当大学名および貢献者の氏名を使用することはできません。

本ソフトウェアは、REGENTSおよび貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、REGENTSまたは貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

Copyright © 2008 by the Massachusetts Institute of Technology.

Copyright 1995 by Richard P. Basch. 著作権所有。

Copyright 1995 by Lehman Brothers, Inc. 著作権所有。

本ソフトウェアをアメリカ合衆国から輸出する場合、アメリカ合衆国政府から特定ライセンスを取得しなければならないことがあります。輸出前に当該ライセンスを取得する責任は、輸出しようとする個人または組織が負います。

その制限内において、本ソフトウェアおよびそのドキュメントを無償で使用、複製、変更、および配布することは、その目的を問わず、前述の著作権情報があらゆる複製に記され、当該著作権情報および本許可情報の両方が解説文書に記され、さらにRichard P. Basch、Lehman BrothersおよびM.I.T.の名称が事前に書面による明確な許諾を得ずに本ソフトウェアの配布に関する広告または広報で使用されない限り、本合意によって許諾されます。Richard P. Basch、Lehman BrothersおよびM.I.T.は、いかなる目的においても、本ソフトウェアの適合性に関して明言しません。本ソフトウェアは、「現状のまま」提供されるものであり、明示であると暗黙であるとを問わず、何ら保証されません。

以下の情報はsrc/lib/krb5/krb/strptime.cおよびsrc/include/k5-queue.hに適用されます。

Copyright © 1997, 1998 The NetBSD Foundation, Inc.

著作権所有。

本コードは、Klaus KleinによりThe NetBSD Foundationに寄与されました。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 本ソフトウェアの機能または使用に言及する全ての広告物には、以下の情報を表記する必要があります。

本製品には、The NetBSD Foundation, Inc.およびその貢献者により開発されたソフトウェアが含まれています。

4. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、The NetBSD Foundationの名称および貢献者の氏名を使用することはできません。

本ソフトウェアは、NetBSD Foundation, Inc.により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、The NetBSD Foundationまたは貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯およびに基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

以下の情報は、src/lib/krb5/unicode内のUnicodeライブラリファイルに適用されます。

Copyright 1997, 1998, 1999 Computing Research Labs,

ニューメキシコ州大学

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、Computing Research Labsまたはニューメキシコ州大学は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他の責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

以下の情報は、src/util/support/strncpy.cに適用されます。

Copyright © 1998 Todd C. Miller Todd.Miller@courtesan.com

本ソフトウェアを、有償または無償を問わず、使用、複製、変更、および配布することは、その目的を問わず、前述の著作権情報およびこの許可情報があらゆる複製に記される限り、本合意によって許諾されます。

本ソフトウェアは、「現状のまま」提供されるものであり、商品性および適合性のあらゆる暗黙的保証を含め、作成者は本ソフトウェアに関して何ら保証しません。いかなる場合も、作成者は、本ソフトウェアの使用または性能によりまたはそれに関連して生じた例外的、直接的、間接的または必然的損傷、または使用できないこと、データの喪失または利益の損失から生じるいかなる損傷に対して、契約の行為、過失またはその他の不法行為であるかどうかに関わらず、一切責任を負いません。

以下の情報はsrc/util/profile/argv_parse.cおよびsrc/util/profile/argv_parse.hに適用されます。

Copyright 1999 by Theodore Ts'o.

本ソフトウェアを、有償または無償を問わず、使用、複製、変更、および配布することは、その目的を問わず、前述の著作権情報およびこの許可情報があらゆる複製に記される限り、本合意によって許諾されます。本ソフトウェアは、「現状のまま」提供されるものであり、商品性および適合性のあらゆる暗黙的保証を含め、Theodore Ts'o (作成者)は本ソフトウェアに関して何ら保証しません。いかなる場合も、作成者は、本ソフトウェアの使用または性能によりまたはそれに関連して生じた例外的、直接的、間接的または必然的損傷、または使用できないこと、データの喪失または利益の損失から生じるいかなる損傷に対して、契約の行為、過失またはその他の不法行為であるかどうかに関わらず、一切責任を負いません。(この種の免責事項を必要とする、弁護士がすぐに訴訟を起こすアメリカの文化は病的ではないか?)

以下の情報はsrc/util/profile/profile_tcl.cのSWICにより生成されたコードに適用されます。

Copyright © 1999-2000, The University of Chicago

本ファイルは、この著作権に関する声明がそのまま残されている限り、ライセンスまたは課金なしで自由に再配布できます。

以下の情報はsrc/lib/rpcおよびsrc/include/gssrpcの一部に適用されます。

Copyright © 2000 The Regents of the University of Michigan. 著作権所有。

Copyright © 2000 Dug Song dugsong@UMICH.EDU. 著作権所有、全ての誤りは無効

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、当大学名および貢献者の氏名を使用することはできません。

本ソフトウェアは、「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、REGENTSまたは貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

MD4アルゴリズムの実装には、以下の情報が適用されます。

Copyright © 1990, RSA Data Security, Inc. 著作権所有。

本ソフトウェアの複製および使用は、本ソフトウェアまたは本機能に言及または参照する全てのものに「RSA Data Security, Inc. MD4 Message Digest Algorithm」と明記する限り許諾されます。

また、派生的著作物の作成および使用についても、派生的著作物に言及または参照する全てのものに「derived from the RSA Data Security, Inc. MD4 Message Digest Algorithm」と明記する限り許諾されます。

RSA Data Security, Inc.は、本ソフトウェアの商品性、および特定目的への本ソフトウェアの適合性のいずれに関しても明言しません。本ソフトウェアは、「現状のまま」提供されるものであり、明示であると暗黙であるとを問わず、何ら保証されません。

これらの情報は、本ドキュメントおよび/またはソフトウェアのいかなる部分の複製においても保持されなければなりません。

MD5アルゴリズムの実装には、以下の情報が適用されます。

Copyright © 1990, RSA Data Security, Inc. 著作権所有。

本ソフトウェアの複製および使用は、本ソフトウェアまたは本機能に言及または参照する全てのものに「RSA Data Security, Inc. MD5 Message Digest Algorithm」と明記する限り許諾されます。

また、派生的著作物の作成および使用についても、派生的著作物に言及または参照する全てのものに「derived from the RSA Data Security, Inc. MD5 Message Digest Algorithm」と明記する限り許諾されます。

RSA Data Security, Inc.は、本ソフトウェアの商品性、および特定目的への本ソフトウェアの適合性のいずれに関しても明言しません。本ソフトウェアは、「現状のまま」提供されるものであり、明示であると暗黙であることを問わず、何ら保証されません。

これらの情報は、本ドキュメントおよび/またはソフトウェアのいかなる部分の複製においても保持されなければなりません。

以下の情報は、src/lib/crypto/crypto_tests/t_mddriver.cに適用されます。

Copyright © 1990-2, RSA Data Security, Inc. Created 1990. 著作権所有。

RSA Data Security, Inc.は、本ソフトウェアの商品性、および特定目的への本ソフトウェアの適合性のいずれに関しても明言しません。本ソフトウェアは、「現状のまま」提供されるものであり、明示であると暗黙であることを問わず、何ら保証されません。

これらの情報は、本ドキュメントおよび/またはソフトウェアのいかなる部分の複製においても保持されなければなりません。

src/lib/krb5の一部には、以下の情報が適用されます。

Copyright © 1994 CyberSAFE Corporation.

Copyright 1990,1991,2007,2008 by the Massachusetts Institute of Technology.

著作権所有。

本ソフトウェアをアメリカ合衆国から輸出する場合、アメリカ合衆国政府から特定ライセンスを取得しなければならないことがあります。輸出前に当該ライセンスを取得する責任は、輸出しようとする個人または組織が負います。

その制限内において、本ソフトウェアおよびそのドキュメントを無償で使用、複製、変更、および配布することは、その目的を問わず、前述の著作権情報があらゆる複製に記され、当該著作権情報および本許可情報の両方が解説文書に記され、さらにM.I.T.の名称が事前に書面による明確な許諾を得ずに本ソフトウェアの配布に関する広告または広報で使用されない限り、本合意によって許諾されます。さらに、本ソフトウェアを変更する場合、そのソフトウェアに「変更されたソフトウェア」であることを示す名前を付ける必要があります、オリジナルのM.I.T.ソフトウェアと混同されるおそれのある方法で配布してはなりません。M.I.T.、Open Computing Security Group、およびCyberSAFE Corporationは、いかなる目的においても、本ソフトウェアの適合性に関して明言しません。本ソフトウェアは、「現状のまま」提供されるものであり、明示であると暗黙であることを問わず、何ら保証されません。

PADL Softwareにより提供される部分には、以下のライセンスが適用されます。

Copyright (c) 2011, PADL Software Pty Ltd. 著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、PADL Softwareの名称および貢献者の氏名を使用することはできません。

本ソフトウェアは、PADL Softwareおよび貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、PADL Softwareまたは貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

バンドリングされるlibevソースコードには、以下のライセンスが適用されます。

All files in libev are Copyright (C)2007,2008,2009 Marc Alexander Lehmann.

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

- ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
- バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。

本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権所有者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

その代わりに、本パッケージのコンテンツは、GNU 一般公衆利用許諾書(「GPL」)バージョン2またはそれ以降のバージョンの条項の下で 사용할 ことができます。その場合、GPLの規定が前述の規定に代わり適用されます。本パッケージの独自のバージョンの使用をGPLの条項の下でのみ許可し、本ファイルの独自のバージョンをBSDライセンスの下で他人に使用させないようにするには、前述の条項を削除することによってその決定を示し、それらの条項に代えて、GPLが必要とする情報およびその他の条項を本パッケージの本ファイルおよびその他のファイルにおいて使用してください。前述の条項が削除されていない場合、受領者は本ファイルの独自のバージョンをBSDまたはGPLのいずれでも使用できるようになります。

Intel AESNI Sample Libraryから複製されたファイルには、以下のライセンスが適用されます。

Copyright © 2010, Intel Corporation.

著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

- ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
- バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
- 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、Intel Corporationの名称および貢献者の氏名を使用することはできません。

本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権所有者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

以下の情報は、src/ccapi/common/win/OldCC/autolock.hxxに適用されます。

Copyright (C) 1998 by Danilo Almeida. 著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

- ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
- バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。

本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権所有者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

以下の情報は

「src/plugins/preauth/spake/edwards25519.c」および

「src/plugins/preauth/spake/edwards25519_tables.h」の一部に適用されます。

MIT License(MIT)

Copyright (c) 2015-2016 the fiat-crypto authors (AUTHORSファイルを参照してください)。

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

以下の情報は

「src/plugins/preauth/spake/edwards25519.c」の一部に適用されます。

Copyright (c) 2015-2016, Google Inc.

本ソフトウェアを、有償または無償を問わず、使用、複製、変更、および/または配布することは、その目的を問わず、前述の著作権情報およびこの許可情報があらゆる複製に記される限り、本合意によって許諾されます。

本ソフトウェアは、「現状のまま」提供されるものであり、商品性および適合性のあらゆる暗黙的保証を含め、作成者は本ソフトウェアに関して何ら保証しません。いかなる場合も、作成者は、本ソフトウェアの使用または性能によりまたはそれに関連して生じた例外的、直接的、間接的または必然的損傷、または使用できないこと、データの喪失または利益の損失から生じるいかなる損傷に対して、契約の行為、過失またはその他の不法行為であるかどうかに関わらず、一切責任を負いません。

Libeditの使用許諾条件

Pulse Secure Virtual Traffic Managerには、コマンドラインエディタライブラリであるLibeditのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 0.3

ライセンス概要: BSD第3項「新規」または「改訂」ライセンス

著作権情報:

Copyright (c) 1997 The NetBSD Foundation, Inc. 著作権所有。

本コードは、Jaromir DolecekによりThe NetBSD Foundationに寄与されたソフトウェアの派生物です。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 本ソフトウェアの機能または使用に言及する全ての広告物には、以下の情報を表記する必要があります。

本製品には、The NetBSD Foundation, Inc.およびその貢献者により開発されたソフトウェアが含まれています。

4. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、The NetBSD Foundationの名称および貢献者の氏名を使用することはできません。

本ソフトウェアは、NetBSD Foundation, Inc.により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、The NetBSD Foundationまたは貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

Copyright (c) 1992, 1993 The Regents of the University of California. 著作権所有。

本コードは、コーネル大学のChristos Zoulasによりバークレー校に寄与されたソフトウェアの派生物です。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 本ソフトウェアの機能または使用に言及する全ての広告物には、以下の情報を表記する必要があります。

本製品には、カリフォルニア大学バークレー校およびその貢献者により開発されたソフトウェアが含まれています。

4. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、当大学名および貢献者の氏名を使用することはできません。

本ソフトウェアは、REGENTSおよび貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、REGENTSまたは貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

ZebOSの使用許諾条件

Pulse Secure Virtual Traffic Managerには、IP Infusion, Inc.のライセンスの下で配布されるZebOSルーティングスイートのコードが含まれています。The ZebOS routing suite is Copyright (C) 2001-2014 IP Infusion, Inc.著作権所有。

Curlの使用許諾条件

Pulse Secure Virtual Traffic Managerには、データ転送用のコマンドラインツールおよびライブラリであるCurlのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 7.58.0

ライセンス概要: MIT

著作権情報:

著作権情報および許可情報

Copyright (c) 1996 - 2018, Daniel Stenberg, <daniel@haxx.se>, and many contributors. THANKSファイルを参照してください。

著作権所有。

本ソフトウェアを、有償または無償を問わず、使用、複製、変更、および配布することは、その目的を問わず、前述の著作権情報およびこの許可情報があらゆる複製に記される限り、本合意によって許諾されます。

ソフトウェアは、商品性、特定目的への適合性、第三者への権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

本通知に記載がある場合を除き、著作権者から事前に書面の許可を得ない限り、著作権者の氏名は、本ソフトウェアの販売、使用またはその他の取引を推進するために、広告またはその他で使用してはなりません。

Janssonの使用許諾条件

Pulse Secure Virtual Traffic Managerには、JSONサポートライブラリであるJanssonのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 2.7

ライセンス概要: MIT

著作権情報:

Copyright (c) 2009-2014 Petri Lehtinen <petri@digip.org>

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

Digest::SHAの使用許諾条件

Pulse Secure Virtual Traffic Managerには、SHAメッセージダイジェストの計算に使用されるPerlライブラリであるDigest::SHAのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 5.95

ライセンス概要: Perl Artistic License

著作権情報:

Copyright (C) 2003-2015 Mark Shelor

このライブラリはフリーソフトウェアであり、Perl自体と同じ条件に基づいてこのライブラリを再配布および/または変更することができます。詳細については、Perl Artistic License(<http://search.cpan.org/perl/doc?perlartistic>)を参照してください。

Sys::SysLogの使用許諾条件

Pulse Secure Virtual Traffic Managerには、UNIX syslogプログラムとのインターフェースとして使用されるPerlライブラリであるSys::SysLogのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 0.27

ライセンス概要: Perl Artistic License

著作権情報:

Copyright (C) 1990-2012 by Larry Wall and others.

このライブラリはフリーソフトウェアであり、Perl自体と同じ条件に基づいてこのライブラリを再配布および/または変更することができます。詳細については、Perl Artistic License(<http://search.cpan.org/perl/doc?perlartistic>)を参照してください。

libpcapの使用許諾条件

Pulse Secure Virtual Traffic Managerには、パケットキャプチャをサポートするために使用されるライブラリであるlibpcapのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 1.8.1

ライセンス概要: BSD

著作権情報:

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

1. ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
2. バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
3. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、作成者の名称を使用することはできません。

本ソフトウェアは、「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。

elfutilsの使用許諾条件

Pulse Secure Virtual Traffic Managerには、ELF(Executable and Linkable Format)ファイルを処理するためのユーティリティとライブラリのセットであるelfutilsのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 0.175

ライセンス概要: GPLv2

著作権情報:

GNU一般公衆利用許諾書

バージョン2、1991年6月

Copyright (C) 1989, 1991 Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

いかなる者も、本ライセンス文書を複製し、一切の変更を加えずに本文書の複製を配布することを許可されますが、変更を加えることは許可されません。

前文

ほとんどのソフトウェアのライセンスは、その共有および変更の自由を制限することを目的としています。これとは対照的に、GNU一般公衆利用許諾書は、フリーソフトウェアを共有および変更する自由を保証すること、すなわち全てのユーザにとってソフトウェアがフリーであることを保証することを目的としています。本GNU一般公衆利用許諾書は、Free Software Foundationのほとんどのソフトウェアおよび作成者がその使用を決定したその他のプログラムに適用されます(GNU劣等一般公衆利用許諾書の対象となるFree Software Foundationのソフトウェアもあります)。被許諾者(「ライセンシー」)は、自らのプログラムにGNU一般公衆利用許諾書を適用することもできます。

フリーソフトウェアとは、無償であることではなく、自由であることを意味しています。GNU一般公衆利用許諾書は、ソースコードを受け取った、または必要な場合に入手できるフリーソフトウェアを配布する(または、必要な場合はこのサービスに対して課金する)自由が与えられること、ソフトウェアを変更し、または新しいフリープログラムでソフトウェアの一部を使用できることを保証し、かかる行為が可能であることを確実に周知することを目的としています。

ライセンシーの権利を保護するために、何らかの者がかかる権利を否定し、またはかかる権利の放棄をライセンシーに求めることを制限する必要があります。かかる制限により、ソフトウェアを頒布する場合、またはソフトウェアを変更する場合には、特定の責任を負うことになります。

例えば、無償または有償のいずれであるかにかかわらず、かかるプログラムを配布する場合、ライセンシーは、自ら有する全ての権利を受領者に付与しなければなりません。また、受領者もソースコードを受け取り、または入手できるようにしなければなりません。さらに、受領者が自らの権利を認識するように、これらの条件を示さなければなりません。

ライセンシーの権利は、以下の2つの手順で保護されます。(1)ソフトウェアを著作権で保護し、(2)ソフトウェアの複製、配布および/または変更の法的な許可を付与する本ライセンスを提供します。

また、各作成者および当財団の保護のために、当財団は、全ての者がこのフリーソフトウェアに対していかなる保証も提供されないことを確実に理解することを求めます。ソフトウェアが何らかの者によって変更され、配布された場合、その受領者は、受け取ったソフトウェアがオリジナルではないため、かかる者に起因する問題が、オリジナルの作成者の評判に影響しないことを認識する必要があります。

最後に、あらゆるフリープログラムは、常にソフトウェア特許による脅威に直面しています。当財団は、フリープログラムの再配布者が個別に特許ライセンスを取得し、実質的にプログラムを独占的にする危険を回避することを望んでいます。これを防止するために、あらゆる特許は、全ての者が自由に使用できるようにライセンスするか、またはまったくライセンスしないかのいずれかにする必要があることを当財団は明確にしました。

複製、配布および変更に対する正確な条件は、以下のとおりです。

GNU一般公衆利用許諾書

複製、配布および変更に対する条件

0. 本ライセンスは、著作権者が本一般公衆利用許諾書の条件に基づいて配布されることを記載し、著作権者によって提示される著作権情報を含むプログラムまたはその他の著作物に適用されます。以下の「プログラム」は、かかるプログラムまたは著作物を指し、「プログラムに基づく著作物」とは、そのプログラムまたは著作権法に基づく派生著作物(つまり、変更の有無、別の言語に翻訳されるか否かにかかわらず、プログラムまたはその一部を含む著作物)のいずれかを意味します(以下、「変更」という用語には、翻訳が含まれますが、これに限定されません)。ライセンスの許諾を受けた者は、それぞれ「ライセンシー」と呼びます。

複製、配布および変更以外の活動は、本ライセンスに含まれず、その範囲外となります。プログラムを実行する行為は制限されず、プログラムからの出力は、(プログラムの実行によって生成されたこととは関係なく)その内容がプログラムに基づく著作物に当たる場合にのみ本ライセンスの対象となります。これに該当するか否かは、プログラムが何を実行するかによって異なります。

1. ライセンシーは、受け取ったプログラムのソースコードを複製し、一切変更を加えずに、任意のメディアで配布することができます。ただし、適切な著作権情報および保証の免責事項を各プログラムに目立つように適切に表示し、本ライセンスおよび保証のないことに言及しているいかなる情報も変更せず、プログラムと共に本ライセンスをプログラムの他の受領者に提供することを条件とします。

ライセンシーは、プログラムを転送する物理的行為に対して料金を請求することができ、自らの裁量により、料金の対価として保証による保護を提供することができます。

2. ライセンシーは、1つ以上のプログラムまたはその一部を変更することにより、プログラムに基づく著作物を作成し、上記の第1条の条件に基づいて、かかる変更または著作物を複製し、配布することができます。ただし、以下の全ての条件も満たさなければなりません。
 - a) ライセンシーは、変更したファイルについて、ファイルを変更したこと、および変更の日付を記載した情報を目立つように伝えなければなりません。

b) ライセンシーは、プログラムまたはその一部につき、その全部または一部を含みまたはこれらから派生し、ライセンシーが配布または公開する全ての著作物を、本ライセンスの条件に基づいて全ての第三者に対して全体として無償でライセンスしなければなりません。

c) 通常、変更したプログラムが実行時に対話式でコマンドを読み取る場合、ライセンシーは、最も一般的な方法でかかる対話式の利用のために実行を開始するときに、適切な著作権情報およびいかなる保証も提供されないという情報(またはライセンシーが保証を提供することを示す情報)、ユーザがかかる条件に基づいてプログラムを再配布できるという情報、および本ライセンスを表示する方法をユーザに伝える情報を含む告知を出力または表示しなければなりません(例外:プログラム自体が対話式であるが、通常かかる告知を出力しない場合、プログラムに基づく著作物で、告知を出力する必要はありません)。

上記の要件は、変更された著作物全体に適用されます。著作物の識別可能な部分が、プログラムから派生したものではなく、それ自体独立した別の著作物であると合理的に見なすことができる場合、本ライセンスおよびその条件は、別の著作物として配布するときに、当該部分に適用されません。ただし、プログラムに基づく著作物全体の一部として当該部分を配布する場合、その全体の配布は、本ライセンスの条件に基づいて行わなければなりません。他のライセンシーに対する本ライセンスの許可は、作成者にかかわらず、全体、つまり全ての部分に適用されます。

したがって、本条は、ライセンシーによって全て作成された著作物に対するライセンシーの権利に対して権利を主張し、または異議を申し立てることを目的としていません。本条の目的は、プログラムに基づく派生著作物または集合著作物の配布を制御する権利を行使することです。

また、あるストレージのボリュームまたは配布メディアで、プログラムに基づかない別の著作物と、プログラム(またはプログラムに基づく著作物)を同梱するだけで、他の著作物が本ライセンスの適用範囲に含まれることはありません。

3. ライセンシーは、上記の第1条および第2条の条件に基づいて、オブジェクトコードまたは実行可能形式で、プログラム(または第2条の下でのプログラムに基づく著作物)を複製し、配布することができます。ただし、以下のいずれかを行うことを条件とします。

a) コンピュータで解読可能な対応する全ソースコードを添付します。これは、ソフトウェアの交換のために慣習的に使用されるメディアで、上記の第1条および第2条の条件に基づいて配布しなければなりません。

b) ソースの配布を物理的に実行するコストを超えない料金で、コンピュータで解読可能な対応する全ソースコードを第三者に提供するための最低3年間有効な書面のオファーを添付します。かかるソースコードは、ソフトウェアの交換のために慣習的に使用されるメディアで、上記の第1条および第2条の条件に基づいて配布されます。

c) 対応するソースコードを配布するオファーに関してライセンシーが受け取った情報を添付します(この代替手段は、非営利目的の配布のみで、ライセンシーが、上記のb項に従って、当該オファーでオブジェクトコードまたは実行可能形式のプログラムを受け取った場合にのみ許可されます)。

著作物のソースコードとは、著作物に変更を加えるために望ましい著作物の形式を意味します。実行可能な著作物の場合、全ソースコードとは、含まれる全てのモジュールの全てのソースコードと、関連するインターフェース定義ファイル、実行可能ファイルのコンパイルおよびインストールを制御するために使用されるスクリプトを意味します。しかし、特別な例外として、配布するソースコードには、通常、実行可能ファイルが動作するオペレーティングシステムの主要なコンポーネント(コンパイラ、カーネルなど)と共に(ソース形式またはバイナリ形式のいずれかで)配布されるコードは含める必要がありません。ただし、当該コンポーネント自体が、実行可能ファイルに付属する場合はこの限りではありません。

実行可能ファイルまたはオブジェクトコードの配布が指定した場所からのアクセスの提供によって行われる場合、第三者がオブジェクトコードと共にソースコードを複製することを強制されなくても、同じ場所からソースコードを複製するために同等のアクセスを提供することは、ソースコードの配布と見なされます。

4. 本ライセンスで明示的に規定されていなる場合を除き、プログラムの複製、変更、サブライセンス、または配布を行うことはできません。その他の方法によるプログラムの複製、変更、サブライセンスまたは配布の試みは無効であり、これにより本ライセンスに基づくライセンシーの権利は自動的に解除されます。ただし、本ライセンスに基づいてライセンシーからプログラムまたは権利を受け取った当事者のライセンスは、当該当事者が本ライセンスを全て遵守している限り解除されません。
5. ライセンシーは本ライセンスに署名しないため、本ライセンスを受け入れることを要求されません。ただし、他のいかなるライセンスも、プログラムまたはその派生著作物を変更または配布する許可をライセンシーに付与しません。本ライセンスを受け入れない場合、かかる行為は法律によって禁止されます。したがって、プログラム(またはプログラムに基づく著作物)を変更または配布することによって、ライセンシーは、かかる行為のために、本ライセンス、およびプログラムまたはプログラムに基づく著作物の複製、配布または変更に対する全ての条件を受け入れることを示します。
6. ライセンシーがプログラム(またはプログラムに基づく著作物)を再配布するたびに、受領者は、かかる条件に従ってプログラムを複製、配布または変更するためのライセンスをオリジナルのライセンサーから自動的に受け取ります。ライセンシーは、本ライセンスで付与される権利の受領者による行使に対して、追加の制限を課すことはできません。ライセンシーは、本ライセンスに対する第三者の遵守を強制することに責任を負いません。

7. 裁判所の判決もしくは特許侵害の主張の結果として、またはその他の理由のために(特許の問題に限られませんが)、本ライセンスの条件と矛盾する条件が、(裁判所の命令、契約またはその他のいづれによるかにかかわらず)ライセンシーに課される場合、これによりライセンシーは本ライセンスの条件から免責されません。本ライセンスに基づく自らの義務およびその他の関連する義務を同時に満たすように配布することができない場合、結果的に、ライセンシーは、プログラムをまったく配布することができなくなります。例えば、特許ライセンスで、ライセンシーを通じて直接または間接にプログラムを受け取った全ての者によるプログラムのロイヤルティ無料の再配布が許可されていない場合、かかる特許ライセンスと本ライセンスの両方を満たす唯一の方法は、プログラムの配布を全て控えることです。

本条のいずれかの部分が、特定の状況の下で無効または強制不可能と判断された場合は、本条の残りの部分を適用し、その他の状況では本条全体を適用することが意図されています。

本条は、ライセンシーによる特許もしくはその他の財産権の主張に対する侵害、またはかかる主張の有効性に対する異議申し立てを誘発することを目的としていません。本条は、一般公衆利用許諾書によって実施されるフリーソフトウェアの配布制度の整合性を保護することのみを目的としています。多くの人々が、当該制度の一貫した適用を信頼して、当該制度を通じて配布される幅広いソフトウェアに対して惜しみない貢献を行いました。作成者/貢献者は、他の制度を通じてソフトウェアを配布することを決定することができ、ライセンシーはその選択を強制することはできません。

本条は、本ライセンスの残りの部分に起因すると見なされている制度を、徹底的に明確にすることを目的としています。

8. プログラムの配布および/または使用が、特許または著作権を有するインターフェースのいずれかによって特定の国で制限される場合、プログラムに本ライセンスを適用するオリジナルの著作権者は、かかる諸国を除外する明示的な地理的配布制限を追加して、除外されない諸国内または諸国間のみで配布が許可されるようにすることができます。このような場合、本ライセンスには、本ライセンスの本文に記載されているかのように、その制限が組み込まれます。
9. Free Software Foundationは、一般公衆利用許諾書の改訂バージョンおよび/または新バージョンを随時公開することがあります。かかる新バージョンは、本質的に現行バージョンと類似していますが、新しい問題または懸念事項に対処するために詳細が異なる場合があります。

各バージョンには、識別のためのバージョン番号が付けられます。プログラムで適用される本ライセンスのバージョン番号および「それ以降のバージョン」が指定される場合、ライセンシーは、当該バージョンまたはFree Software Foundationによって公開される後続のバージョンのいずれかの条件に従う選択肢を有します。プログラムで本ライセンスのバージョン番号が指定されない場合、ライセンシーは、Free Software Foundationによって公開された任意のバージョンを選択することができます。

10. ライセンシーが、プログラムの一部を配布条件の異なる他のフリープログラムに組み込む場合は、作成者に連絡を取って許可を求めてください。Free Software Foundationが著作権を有するソフトウェアの場合は、当財団に連絡してください。当財団は、これに関して例外を設けることがあります。当財団の決定は、そのフリーソフトウェアの全ての派生著作物のフリーな状態を維持し、ソフトウェア一般の共有および再利用を促進するという2つの目的に基づいて決定されます。

保証の否認

11. プログラムは無償でライセンスされるため、適用される法律によって許可される限り、プログラムに対する保証はありません。書面に別途記載される場合を除き、著作権者および/またはその他の当事者は、商品性、特定目的への適合性の暗黙的な保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」プログラムを提供します。プログラムの品質およびパフォーマンスに関する全てのリスクは、ライセンシーが負います。プログラムに欠陥があった場合、ライセンシーは、全ての必要な修正、修復または訂正のコストを負担します。
12. 適用される法律によって要求されるか、または書面によって合意されない限り、いかなる場合にも、上記の許可に従ってプログラムを修正および/または再配布することができる著作権者またはその他の当事者は、プログラムの使用または使用不可に起因する一般損害賠償、特別損害賠償、付随的損害賠償または結果的損害賠償を含む損害賠償に責任を負います(これには、データの損失もしくは不正確なデータの表示、またはライセンシーもしくは第三者が被る損失、またはプログラムが他のプログラムと共に動作しないことが含まれますが、これらに限定されません)。これは、かかる著作権者またはその他の当事者が、かかる損害賠償の可能性について通知を受けていた場合でも同様です。

条件の末尾

新しいプログラムにかかる条件を適用する方法

新しいプログラムを開発し、できる限り多くの人々に使用してもらいたい場合、これを実現するための最適な方法は、全ての人々がかかる条件に基づいてプログラムを再配布および変更することができるフリーソフトウェアにすることです。

これを行うには、プログラムに以下の情報を追加します。保証の排除について最も効果的に伝えるために、各ソースファイルの冒頭に、当該情報を追加することが最も安全です。各ファイルには、少なくとも「著作権」を示す行および情報の全文がある場所を示す参照が必要です。

<プログラムの名前およびその実行内容の概要>

Copyright (C) <年> <作成者の氏名>

本プログラムは、フリーソフトウェアです。Free Software Foundationによって公開されたGNU一般公衆利用許諾書のバージョン2または(作成者の裁量により)それ以降のバージョンの条件に基づいて、このプログラムを再配布および/または変更することができます。

本プログラムは、有益であることを願って配布されますが、商品性または特定目的への適合性の暗黙的保証を含め、何ら保証されません。詳細については、GNU一般公衆利用許諾書を参照してください。

本プログラムには、GNU一般公衆利用許諾書が付属しています。付属していない場合は、Free Software Foundation, Inc. (米国02110-1301マサチューセッツ州ボストン、フランクリン通り51、5階)までご連絡ください。

また、電子メールおよび郵便によって作成者に連絡を取る方法に関する情報も追加します。

プログラムが対話式である場合は、対話式モードが開始したときに、以下のような短い情報を出力します。

Gnomovision version 69, Copyright (C) <年> <作成者の氏名>

Gnomovisionでは、いかなる保証も提供されません。詳細については、「show w」と入力してください。これはフリーソフトウェアであり、特定の条件に基づいて再配布することができます。詳細については、「show c」と入力してください。

コマンドの例「show w」および「show c」によって、一般公衆利用許諾書の適切な部分が表示されるようになります。当然ながら、「show w」および「show c」以外の名称のコマンドを使用することができます。さらに、マウスのクリックやメニュー項目など、プログラムに適した任意の方法を利用できます。

必要な場合は、雇用主(プログラマーとして働いている場合)または学校(該当する場合)に、プログラムの「著作権の放棄」に署名してもらう必要もあります。以下にサンプルを示します。名前を変更してください。

Yoyodyne, Inc.は、James Hackerによって作成された(コンパイラでコンパイルされた)プログラム「Gnomovision」に関する全ての著作権を本書により放棄します。

<Ty Coonの署名>、1989年4月1日

Ty Coon副社長

本一般公衆利用許諾書は、プログラムを独占的なプログラムに組み込むことを許可していません。プログラムがサブルーチンライブラリである場合は、独占的なアプリケーションにそのライブラリとのリンクを許可する方が有益であると見なすことができます。これを行う場合は、本ライセンスの代わりにGNU劣等一般公衆利用許諾書を使用してください。

libpngの使用許諾条件

Pulse Secure Virtual Traffic Managerには、PNG参照ライブラリであるlibpngのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 1.2.57

ライセンス概要: MIT

著作権:

著作権情報、免責事項、およびライセンス:

libpngを変更する場合は、この文の直後に追加の情報を挿入することができます。

このコードは、libpngのライセンスに基づいてリリースされます。

libpngバージョン1.0.7、2000年7月1日～1.2.56、2015年12月17日(Copyright (c) 2000-2002, 2004, 2006-2015 Glenn Randers-Pehrson)は、libpng-1.0.6から派生したものであり、協力作成者のリストに以下の個人を追加して、libpng-1.0.6と同じ免責事項およびライセンスに従って配布されます。

Simon-Pierre Cadieux

Eric S. Raymond

Cosmin Truta

Gilles Vollant

また、免責事項に、以下の情報を追加します。

ライセンスの利用の妨げまたは侵害に対して、いかなる保証も提供されません。当社の取り組みまたはライブラリによって、ユーザの特定の目的またはニーズが満たされるという保証はありません。本ライブラリは、全ての欠陥と共に提供され、納得のいく品質、パフォーマンス、精度、および取り組みに関連する全てのリスクはユーザが負います。

libpngバージョン0.97、1998年1月～1.0.6、2000年3月20日(Copyright (c) 1998-2000 Glenn Randers-Pehrson)は、libpng-0.96から派生したものであり、協力作成者のリストに以下の個人を追加して、libpng-0.96と同じ免責事項およびライセンスに従って配布されます。

Tom Lane

Glenn Randers-Pehrson

Willem van Schaik

libpngバージョン0.89、1996年6月～0.96、1997年5月(Copyright (c) 1996-1997 Andreas Dilger)は、libpng-0.88から派生したものであり、協力作成者のリストに以下の個人を追加して、libpng-0.88と同じ免責事項およびライセンスに従って配布されます。

John Bowler

Kevin Bracey

Sam Bushell

Magnus Holmgren

Greg Roelofs

Tom Tanner

libpng versions 0.5、1995年5月から～0.88、1996年1月(Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.)

本著作権およびライセンスの目的のために、「協力作成者」とは、以下の一連の個人として定義されます。

Andreas Dilger

Dave Martindale

Guy Eric Schalnat

Paul Schmidt

Tim Wegner

PNG参照ライブラリは「現状のまま」提供されます。協力作成者およびGroup 42, Inc.は、明示であると暗黙であるとを問わず、全ての保証を否認します。これには、商品性および何らかの目的への適合性の保証が含まれますが、これらに限定されません。協力作成者およびGroup 42, Inc.は、損害賠償の可能性について通知を受けていた場合でも、PNG参照ライブラリの使用に起因することがある直接損害賠償、間接損害賠償、付随的損害賠償、特別損害賠償、懲罰的損害賠償、または結果的損害賠償に対していかなる責任も負いません。

本ソースコードまたはその一部を任意の目的に無償で使用、コピー、変更、配布する権利が、以下の制限を条件として許諾されます。

1. 本ソースコードの元の作成者について偽りの表記をしてはなりません。
2. 修正したバージョンは、その旨を明記しなければならず、元のソースとして偽って表記してはなりません。
3. 著作権に関するこの情報は、配布するソースまたは修正したソースから削除し、または修正することはできません。

協力作成者およびGroup 42, Inc.は、商用製品でPNGファイルフォーマットをサポートするためのコンポーネントとして、本ソースコードの使用を無償で明示的に許可し、奨励します。製品内で本ソースコードを使用する場合、その旨を記載することが望まれますが、必須ではありません。

著作権情報、免責事項、およびライセンスの末尾

Perlの使用許諾条件

Pulse Secure Virtual Traffic Managerには、修正版のPerlが含まれています。このライセンス属性に適合する修正版パッケージの詳細を確認するには、ご使用の製品の『Pulse Secure Virtual Traffic Manager: Installation and Getting Started Guide』の「オープンソースソフトウェアライセンス」の章の説明を参照してください。

以下の著作権情報が適用されます。

バージョン: 5.10.1

ライセンス概要: Perl Artistic License v1.0

著作権:

「Artistic License」

前文

本書は、パッケージの使用者に事実上の慣習的方法においてパッケージを使用および配布する権利、および合理的変更を行う権利を付与しながらも、著作権所有者がパッケージ開発に関する一定の芸術的管理権を維持するように、パッケージの複製が許容される条件を示すことを目的としています。

定義:

- 「パッケージ」とは、著作権所有者が配布するファイルの集まり、および当該のファイルの集まりに対してテキストの変更を行うことで作成された派生物を指します。
- 「標準バージョン」とは、変更されていないパッケージ、または以下に規定する著作権所有者の意思に従って変更されたパッケージを指します。
- 「著作権所有者」とは、パッケージの著作権で指名された全ての人物を指します。
- 「お客様」とは、あなたが本パッケージを複製または配布しようとしている場合、あなた自身を指します。

- 「合理的複製料」とは、メディア費用、複製費用、関係者の拘束時間などに基づいてお客様が正当化できる料金を指します(お客様は、著作権所有者に対してではなく、当該料金を負担する市場としての一般的なコンピュータ関係社会に対してのみ当該複製料を正当化するように求められます)。
 - 「自由に利用可能」とは、商品自体に対して料金が課されないことを指しますが、商品の取り扱いに関する料金が課されることがあります。また、商品の受領者は、当該商品を受領したときと同じ条件に基づいて当該商品を再配布できます。
1. お客様は、元の著作権情報および関連する免責事項の全てを複製する限りにおいて、本パッケージの標準バージョンのソース形式の逐語的複製を無制限に作成および配布することができます。
 2. お客様は、バグの修正、移植性の修正、およびパブリックドメインまたは著作権所有者に由来するその他の変更を加えることができます。そのような方法において変更されたパッケージは、標準バージョンと見なされるものとします。
 3. いつどのような変更をファイルに対して行ったかを記述した明示的な情報を変更された各ファイルに挿入し、かつ以下の少なくとも1つに該当する限り、本パッケージの複写を、形を問わず別の方法で変更することができます。
 - a) お客様が行った変更事項をパブリックドメインに掲載する、またはそれらの変更事項を自由に利用可能な状態にする。これには、Usenetまたは同等の媒体へ前述の変更事項を掲示する、uunet.uu.netなどの大手アーカイブサイトへ変更事項を掲載する、お客様が行った変更事項を著作権所有者が本パッケージの標準バージョンに取り込むことを許可するなどの方法が該当します。
 - b) 変更したパッケージをお客様の会社または組織内においてのみ使用する。
 - c) 標準の実行ファイルと名前が競合しないように、非標準の全ての実行ファイルの名前を変更し、標準の実行ファイルと併せて提供する。また、非標準の実行ファイルごとに、標準バージョンとの差異を明確に記した個別のマニュアルページを提供する。
 - d) 著作権所有者と配布に関するその他の取り決めを行う。
 4. 以下の少なくとも1つに該当する限り、お客様は本パッケージのプログラムをオブジェクトコードまたは実行ファイルの形式で配布することができます。
 - a) 実行ファイルおよびライブラリファイルの標準バージョンを、(マニュアルページまたはそれと同等のものにおいて)使用法とともに標準バージョンを入手できる場所において配布する。
 - b) コンピュータが解読できるパッケージのソースにお客様が行った変更事項を加えて配布する。

- c) 非標準の実行ファイルに非標準の名前を付け、標準バージョンを入手できる場所において、その差異を使用方法とともにマニュアルページ(または同等のもの)に明確に記す。
- d) 著作権所有者と配布に関するその他の取り決めを行う。
5. お客様は、本パッケージの任意の配布に対して合理的複製料を課すことができます。お客様は、本パッケージのサポートに対して任意の料金を課すことができます。お客様は、本パッケージ自体に対する料金を課すことはできません。ただし、お客様が本パッケージをお客様自身の製品として広告しない限り、より大規模な(できる限り商用の)ソフトウェアの配布の一環として、その他の(できる限り商用の)プログラムと併せて本パッケージを配布することができます。お客様は、本パッケージのインタープリタを実行ファイルの内部に(リンク付けによって)埋め込むことができます。インタープリタの完全に標準のバージョンが同様に埋め込まれている場合、アグリゲーションに似せた形式で解釈される必要があります。
6. 本パッケージへのインプットとして供給される、または本パッケージからのアウトプットとして生成されるスクリプトおよびライブラリファイルは、自動的に本パッケージの著作権の影響を受けることはなく、それらを生成した人物に属するものとし、それらを商業的に販売したり、本パッケージと集約したりすることができます。そのようなスクリプトまたはライブラリファイルが、バイナリ実行可能イメージを生成するいわゆる「undump」または「unexec」の方法で本パッケージに集約される場合、お客様がそのような実行可能イメージを本パッケージの標準バージョンであると明言しない限り、そのようなイメージの配布が本パッケージの配布と解釈されたり、第3および第4段落の制限が適用されたりすることはないものとします。
7. 本パッケージで定義されている言語のサブルーチンおよびバリアブルをエミュレートするために、お客様によって供給され、本パッケージにリンクされるCサブルーチン(またはその他の言語で同等にコンパイルされたサブルーチン)は、当該ルーチンが、当該言語に対する回帰テストが失敗に終わる原因となるような方法で当該言語を変更しない限り、本パッケージの一部と見なされることはなく、第6段落のインプットに相当します。
8. 本パッケージの使用が埋め込まれている場合、すなわち、本パッケージのインターフェースが商業的配布のエンドユーザに見えるようにする公然たる試みが一切行われないう限り、商業的配布を伴う本パッケージのアグリゲーションは常に認められます。そのような使用は、本パッケージの配布とは解釈されません。
9. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、著作権所有者の名称を使用することはできません。
10. 本パッケージは、「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。

終わり

xmlsecの使用許諾条件

Pulse Secure Virtual Traffic Managerには、LibXML2に基づくXMLセキュリティライブラリであるxmlsecのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 1.2.24

ライセンス概要: MIT

著作権:

xmlsec、xmlsec-openssl、xmlsec-gnutls、xmlsec-gcryptライブラリ

Copyright (C) 2002-2016 Aleksey Sanin <aleksey@aleksey.com>. 著作権所有。

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、ALEKSEY SANINは、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

本通知に記載がある場合を除き、本人から事前に書面の許可を得ない限り、Aleksey Saninの氏名は、本ソフトウェアの販売、使用またはその他の取引を推進するために、広告またはその他で使用してはなりません。

xmlsec-nssライブラリ

Copyright (C) 2002-2016 Aleksey Sanin <aleksey@aleksey.com>. 著作権所有。

Copyright (c) 2003 America Online, Inc. 著作権所有。

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアの一部は、Mozilla Public License (MPL)により規制されるソースコードおよび/またはAPIを使用して作成されています。MPLは、<http://www.mozilla.org/MPL/MPL-1.1.html>で参照できます。MPLでは、そうした部分に関してMPLの要件を満たしている限り、MPLで規制されないコードとともに、そうした部分を配布することを許可しています。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、ALEKSEY SANINは、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

本通知に記載がある場合を除き、本人から事前に書面の許可を得ない限り、Aleksey Saninの氏名は、本ソフトウェアの販売、使用またはその他の取引を推進するために、広告またはその他で使用してはなりません。

xmlsec-mscryptoライブラリ

Copyright (C) 2002-2016 Aleksey Sanin <aleksey@aleksey.com>. 著作権所有。

Copyright (C) 2003 Cordys R&D BV, 著作権所有。

Copyright (C) 2007 Roumen Petrov.

Copyright (c) 2005-2006 Cryptocom LTD (<http://www.cryptocom.ru>).

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であると問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、ALEKSEY SANINは、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

本通知に記載がある場合を除き、本人から事前に書面の許可を得ない限り、Aleksey Saninの氏名は、本ソフトウェアの販売、使用またはその他の取引を推進するために、広告またはその他で使用してはなりません。

参照

* AOL

<http://www.aleksey.com/pipermail/xmlsec/2003/005488.html>

<http://www.aleksey.com/pipermail/xmlsec/attachments/20030729/0e25648e/attachment.htm>

* Cordys R&D BV

<http://www.aleksey.com/pipermail/xmlsec/2003/005581.html>

* Cryptocom LTD

<http://www.aleksey.com/pipermail/xmlsec/2006/007410.html>

ltraceの使用許諾条件

Pulse Secure Virtual Traffic Managerには、ダイナミックリンクライブラリのランタイムライブラリ呼び出しを追跡するために使用されるプログラムであるltraceのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 0.7.3

ライセンス概要: GPLv2

著作権:

GNU一般公衆利用許諾書

バージョン2、1991年6月

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

いかなる者も、本ライセンス文書を複製し、一切の変更を加えずに本文書の複製を配布することを許可されますが、変更を加えることは許可されません。

前文

ほとんどのソフトウェアのライセンスは、その共有および変更の自由を制限することを目的としています。これとは対照的に、GNU一般公衆利用許諾書は、フリーソフトウェアを共有および変更する自由を保証すること、すなわち全てのユーザにとってソフトウェアがフリーであることを保証することを目的としています。本GNU一般公衆利用許諾書は、Free Software Foundationのほとんどのソフトウェアおよび作成者がその使用を決定したその他のプログラムに適用されます(GNUライブラリー一般公衆利用許諾書の対象となるFree Software Foundationのソフトウェアもあります)。被許諾者(「ライセンシー」)は、自らのプログラムにGNU一般公衆利用許諾書を適用することもできます。

フリーソフトウェアとは、無償であることではなく、自由であることを意味しています。GNU一般公衆利用許諾書は、ソースコードを受け取った、または必要な場合に入手できるフリーソフトウェアを配布する(または、必要な場合はこのサービスに対して課金する)自由が与えられること、ソフトウェアを変更し、または新しいフリープログラムでソフトウェアの一部を使用できることを保証し、かかる行為が可能であることを確実に周知することを目的としています。

ライセンシーの権利を保護するために、何らかの者がかかる権利を否定し、またはかかる権利の放棄をライセンシーに求めることを制限する必要があります。かかる制限により、ソフトウェアを頒布する場合、またはソフトウェアを変更する場合には、特定の責任を負うことになります。

例えば、無償または有償のいずれであるかにかかわらず、かかるプログラムを配布する場合、ライセンシーは、自ら有する全ての権利を受領者に付与しなければなりません。また、受領者もソースコードを受け取り、または入手できるようにしなければなりません。さらに、受領者が自らの権利を認識するように、これらの条件を示さなければなりません。

ライセンシーの権利は、以下の2つの手順で保護されます。(1)ソフトウェアを著作権で保護し、(2)ソフトウェアの複製、配布および/または変更の法的な許可を付与する本ライセンスを提供します。

また、各作成者および当財団の保護のために、当財団は、全ての者がこのフリーソフトウェアに対していかなる保証も提供されないことを確実に理解することを求めます。ソフトウェアが何らかの者によって変更され、配布された場合、その受領者は、受け取ったソフトウェアがオリジナルではないため、かかる者に起因する問題が、オリジナルの作成者の評判に影響しないことを認識する必要があります。

最後に、あらゆるフリープログラムは、常にソフトウェア特許による脅威に直面しています。当財団は、フリープログラムの再配布者が個別に特許ライセンスを取得し、実質的にプログラムを独占的にする危険を回避することを望んでいます。これを防止するために、あらゆる特許は、全ての者が自由に使用できるようにライセンスするか、またはまったくライセンスしないかのいずれかにする必要があることを当財団は明確にしました。

複製、配布および変更に対する正確な条件は、以下のとおりです。

GNU一般公衆利用許諾書

複製、配布および変更に対する条件

0. 本ライセンスは、著作権者が本一般公衆利用許諾書の条件に基づいて配布されることを記載し、著作権者によって提示される著作権情報を含むプログラムまたはその他の著作物に適用されます。以下の「プログラム」は、かかるプログラムまたは著作物を指し、「プログラムに基づく著作物」とは、そのプログラムまたは著作権法に基づく派生著作物(つまり、変更の有無、別の言語に翻訳されるか否かにかかわらず、プログラムまたはその一部を含む著作物)のいずれかを意味します(以下、「変更」という用語には、翻訳が含まれますが、これに限定されません)。ライセンスの許諾を受けた者は、それぞれ「ライセンシー」と呼びます。

複製、配布および変更以外の活動は、本ライセンスに含まれず、その範囲外となります。プログラムを実行する行為は制限されず、プログラムからの出力は、(プログラムの実行によって生成されたこととは関係なく)その内容がプログラムに基づく著作物に当たる場合にのみ本ライセンスの対象となります。これに該当するか否かは、プログラムが何を実行するかによって異なります。

1. ライセンシーは、受け取ったプログラムのソースコードを複製し、一切変更を加えずに、任意のメディアで配布することができます。ただし、適切な著作権情報および保証の免責事項を各プログラムに目立つように適切に表示し、本ライセンスおよび保証のないことに言及しているいかなる情報も変更せず、プログラムと共に本ライセンスをプログラムの他の受領者に提供することを条件とします。

ライセンシーは、プログラムを転送する物理的行為に対して料金を請求することができ、自らの裁量により、料金の対価として保証による保護を提供することができます。

2. ライセンシーは、1つ以上のプログラムまたはその一部を変更することにより、プログラムに基づく著作物を作成し、上記の第1条の条件に基づいて、かかる変更または著作物を複製し、配布することができます。ただし、以下の全ての条件も満たさなければなりません。

a) ライセンシーは、変更したファイルについて、ファイルを変更したこと、および変更の日付を記載した情報を目立つように伝えなければなりません。

b) ライセンシーは、プログラムまたはその一部につき、その全部または一部を含みまたはこれらから派生し、ライセンシーが配布または公開する全ての著作物を、本ライセンスの条件に基づいて全ての第三者に対して全体として無償でライセンスしなければなりません。

c) 通常、変更したプログラムが実行時に対話式でコマンドを読み取る場合、ライセンシーは、最も一般的な方法でかかる対話式の利用のために実行を開始するときに、適切な著作権情報およびいかなる保証も提供されないという情報(またはライセンシーが保証を提供することを示す情報)、ユーザがかかる条件に基づいてプログラムを再配布できるという情報、および本ライセンスを表示する方法をユーザに伝える情報を含む告知を出力または表示しなければなりません(例外:プログラム自体が対話式であるが、通常かかる告知を出力しない場合、プログラムに基づく著作物で、告知を出力する必要はありません)。

上記の要件は、変更された著作物全体に適用されます。著作物の識別可能な部分が、プログラムから派生したものではなく、それ自体独立した別の著作物であると合理的に見なすことができる場合、本ライセンスおよびその条件は、別の著作物として配布するときに、当該部分に適用されません。ただし、プログラムに基づく著作物全体の一部として当該部分を配布する場合、その全体の配布は、本ライセンスの条件に基づいて行わなければなりません。他のライセンシーに対する本ライセンスの許可は、作成者にかかわらず、全体、つまり全ての部分に適用されます。

したがって、本条は、ライセンシーによって全て作成された著作物に対するライセンシーの権利に対して権利を主張し、または異議を申し立てることを目的としていません。本条の目的は、プログラムに基づく派生著作物または集合著作物の配布を制御する権利を行使することです。

また、あるストレージのボリュームまたは配布メディアで、プログラムに基づかない別の著作物と、プログラム(またはプログラムに基づく著作物)を同梱するだけで、他の著作物が本ライセンスの適用範囲に含まれることはありません。

3. ライセンシーは、上記の第1条および第2条の条件に基づいて、オブジェクトコードまたは実行可能形式で、プログラム(または第2条の下でのプログラムに基づく著作物)を複製し、配布することができます。ただし、以下のいずれかも行うことを条件とします。

a) コンピュータで解読可能な対応する全ソースコードを添付します。これは、ソフトウェアの交換のために慣習的に使用されるメディアで、上記の第1条および第2条の条件に基づいて配布しなければなりません。

b) ソースの配布を物理的に実行するコストを超えない料金で、コンピュータで解読可能な対応する全ソースコードを第三者に提供するための最低3年間有効な書面のオファーを添付します。かかるソースコードは、ソフトウェアの交換のために慣習的に使用されるメディアで、上記の第1条および第2条の条件に基づいて配布されます。

c) 対応するソースコードを配布するオファーに関してライセンシーが受け取った情報を添付します(この代替手段は、非営利目的の配布のみで、ライセンシーが、上記のb項に従って、当該オファーでオブジェクトコードまたは実行可能形式のプログラムを受け取った場合にのみ許可されます)。

著作物のソースコードとは、著作物に変更を加えるために望ましい著作物の形式を意味します。実行可能な著作物の場合、全ソースコードとは、含まれる全てのモジュールの全てのソースコードと、関連するインターフェース定義ファイル、実行可能ファイルのコンパイルおよびインストールを制御するために使用されるスクリプトを意味します。しかし、特別な例外として、配布するソースコードには、通常、実行可能ファイルが動作するオペレーティングシステムの主要なコンポーネント(コンパイラ、カーネルなど)と共に(ソース形式またはバイナリ形式のいずれかで)配布されるコードは含める必要がありません。ただし、当該コンポーネント自体が、実行可能ファイルに付属する場合はこの限りではありません。

実行可能ファイルまたはオブジェクトコードの配布が指定した場所からのアクセスの提供によって行われる場合、第三者がオブジェクトコードと共にソースコードを複製することを強制されなくても、同じ場所からソースコードを複製するために同等のアクセスを提供することは、ソースコードの配布と見なされます。

4. 本ライセンスで明示的に規定されていなる場合を除き、プログラムの複製、変更、サブライセンス、または配布を行うことはできません。その他の方法によるプログラムの複製、変更、サブライセンスまたは配布の試みは無効であり、これにより本ライセンスに基づくライセンシーの権利は自動的に解除されます。ただし、本ライセンスに基づいてライセンシーからプログラムまたは権利を受け取った当事者のライセンスは、当該当事者が本ライセンスを全て遵守している限り解除されません。

5. ライセンシーは本ライセンスに署名しないため、本ライセンスを受け入れることを要求されません。ただし、他のいかなるライセンスも、プログラムまたはその派生著作物を変更または配布する許可をライセンシーに付与しません。本ライセンスを受け入れない場合、かかる行為は法律によって禁止されます。したがって、プログラム(またはプログラムに基づく著作物)を変更または配布することによって、ライセンシーは、かかる行為のために、本ライセンス、およびプログラムまたはプログラムに基づく著作物の複製、配布または変更に対する全ての条件を受け入れることを示します。

6. ライセンシーがプログラム(またはプログラムに基づく著作物)を再配布するたびに、受領者は、かかる条件に従ってプログラムを複製、配布または変更するためのライセンスをオリジナルのライセンサーから自動的に受け取ります。ライセンシーは、本ライセンスで付与される権利の受領者による行使に対して、追加の制限を課すことはできません。ライセンシーは、本ライセンスに対する第三者の遵守を強制することに責任を負いません。

7. 裁判所の判決もしくは特許侵害の主張の結果として、またはその他の理由のために(特許の問題に限られませんが)、本ライセンスの条件と矛盾する条件が、(裁判所の命令、契約またはその他のいずれによるかにかかわらず)ライセンシーに課される場合、これによりライセンシーは本ライセンスの条件から免責されません。本ライセンスに基づく自らの義務およびその他の関連する義務を同時に満たすように配布することができない場合、結果的に、ライセンシーは、プログラムをまったく配布することができなくなります。例えば、特許ライセンスで、ライセンシーを通じて直接または間接にプログラムを受け取った全ての者によるプログラムのロイヤルティ無料の再配布が許可されていない場合、かかる特許ライセンスと本ライセンスの両方を満たす唯一の方法は、プログラムの配布を全て控えることです。

本条のいずれかの部分が、特定の状況の下で無効または強制不可能と判断された場合は、本条の残りの部分を適用し、その他の状況では本条全体を適用することが意図されています。

本条は、ライセンシーによる特許もしくはその他の財産権の主張に対する侵害、またはかかる主張の有効性に対する異議申し立てを誘発することを目的としていません。本条は、一般公衆利用許諾書によって実施されるフリーソフトウェアの配布制度の整合性を保護することのみを目的としています。多くの人々が、当該制度の一貫した適用を信頼して、当該制度を通じて配布される幅広いソフトウェアに対して惜しみない貢献を行いました。作成者/貢献者は、他の制度を通じてソフトウェアを配布することを決定することができ、ライセンシーはその選択を強制することはできません。

本条は、本ライセンスの残りの部分に起因すると見なされている制度を、徹底的に明確にすることを目的としています。

8. プログラムの配布および/または使用が、特許または著作権を有するインターフェースのいずれかによって特定の国で制限される場合、プログラムに本ライセンスを適用するオリジナルの著作権者は、かかる諸国を除外する明示的な地理的配布制限を追加して、除外されない諸国内または諸国間のみで配布が許可されるようにすることができます。このような場合、本ライセンスには、本ライセンスの本文に記載されているかのように、その制限が組み込まれます。

9. Free Software Foundationは、一般公衆利用許諾書の改訂バージョンおよび/または新バージョンを随時公開することがあります。かかる新バージョンは、本質的に現行バージョンと類似していますが、新しい問題または懸念事項に対処するために詳細が異なる場合があります。

各バージョンには、識別のためのバージョン番号が付けられます。プログラムで適用される本ライセンスのバージョン番号および「それ以降のバージョン」が指定される場合、ライセンシーは、当該バージョンまたはFree Software Foundationによって公開される後続のバージョンのいずれかの条件に従う選択肢を有します。プログラムで本ライセンスのバージョン番号が指定されない場合、ライセンシーは、Free Software Foundationによって公開された任意のバージョンを選択することができます。

10. ライセンシーが、プログラムの一部を配布条件の異なる他のフリープログラムに組み込む場合は、作成者に連絡を取って許可を求めてください。Free Software Foundationが著作権を有するソフトウェアの場合は、当財団に連絡してください。当財団は、これに関して例外を設けることがあります。当財団の決定は、そのフリーソフトウェアの全ての派生著作物のフリーな状態を維持し、ソフトウェア一般の共有および再利用を促進するという2つの目的に基づいて決定されます。

保証の否認

11. プログラムは無償でライセンスされるため、適用される法律によって許可される限り、プログラムに対する保証はありません。書面に別途記載される場合を除き、著作権者および/またはその他の当事者は、商品性、特定目的への適合性の暗黙的な保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」プログラムを提供します。プログラムの品質およびパフォーマンスに関する全てのリスクは、ライセンシーが負います。プログラムに欠陥があった場合、ライセンシーは、全ての必要な修正、修復または訂正のコストを負担します。

12. 適用される法律によって要求されるか、または書面によって合意されない限り、いかなる場合にも、上記の許可に従ってプログラムを修正および/または再配布することができる著作権者またはその他の当事者は、プログラムの使用または使用不可に起因する一般損害賠償、特別損害賠償、付随的損害賠償または結果的損害賠償を含む損害賠償に責任を負います(これには、データの損失もしくは不正確なデータの表示、またはライセンシーもしくは第三者が被る損失、またはプログラムが他のプログラムと共に動作しないことが含まれますが、これらに限定されません)。これは、かかる著作権者またはその他の当事者が、かかる損害賠償の可能性について通知を受けていた場合でも同様です。

条件の末尾

新しいプログラムにかかる条件を適用する方法

新しいプログラムを開発し、できる限り多くの人々に使用してもらいたい場合、これを実現するための最適な方法は、全ての人々がかかる条件に基づいてプログラムを再配布および変更することができるフリーソフトウェアにすることです。

これを行うには、プログラムに以下の情報を追加します。保証の排除について最も効果的に伝えるために、各ソースファイルの冒頭に、当該情報を追加することが最も安全です。各ファイルには、少なくとも「著作権」を示す行および情報の全文がある場所を示す参照が必要です。

<プログラムの名前およびその実行内容の概要>

Copyright (C) <年> <作成者の氏名>

本プログラムは、フリーソフトウェアです。Free Software Foundationによって公開されたGNU一般公衆利用許諾書のバージョン2または(作成者の裁量により)それ以降のバージョンの条件に基づいて、このプログラムを再配布および/または変更することができます。

本プログラムは、有益であることを願って配布されますが、商品性または特定目的への適合性の暗黙的保証を含め、何ら保証されません。詳細については、GNU一般公衆利用許諾書を参照してください。

本プログラムには、GNU一般公衆利用許諾書が付属しています。付属していない場合は、Free Software Foundation, Inc. (米国02110-1301マサチューセッツ州ボストン、フランクリン通り51、5階)までご連絡ください。

また、電子メールおよび郵便によって作成者に連絡を取る方法に関する情報も追加します。

プログラムが対話式である場合は、対話式モードが開始したときに、以下のような短い情報を出力します。

Gnomovision version 69, Copyright (C) <年> <作成者の氏名>

Gnomovisionでは、いかなる保証も提供されません。詳細については、「show w」と入力してください。

これはフリーソフトウェアであり、特定の条件に基づいて再配布することができます。詳細については、「show c」と入力してください。

コマンドの例「show w」および「show c」によって、一般公衆利用許諾書の適切な部分が表示されるようになります。当然ながら、「show w」および「show c」以外の名称のコマンドを使用することができます。さらに、マウスのクリックやメニュー項目など、プログラムに適した任意の方法を利用できます。

必要な場合は、雇用主(プログラマーとして働いている場合)または学校(該当する場合)に、プログラムの「著作権の放棄」に署名してもらう必要もあります。以下にサンプルを示します。名前を変更してください。

Yoyodyne, Inc.は、James Hackerによって作成された(コンパイラでコンパイルされた)プログラム「Gnomovision」に関する全ての著作権を本書により放棄します。

<Ty Coonの署名>、1989年4月1日

Ty Coon副社長

本一般公衆利用許諾書は、プログラムを独占的なプログラムに組み込むことを許可していません。プログラムがサブルーチンライブラリである場合は、独占的なアプリケーションにそのライブラリとのリンクを許可する方が有益であると見なすことができます。これを行う場合は、本ライセンスの代わりにGNUライブラリ一般公衆利用許諾書を使用してください。

YAML::XSの使用許諾条件

Pulse Secure Virtual Traffic Managerには、XSおよびlibyamlを使用するPerl YAML Serializationである、YAML::XSのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 0.74

ライセンス概要: Perl Artistic License v1.0

著作権:

Copyright 2007-2018.Ingy döt Net.

「Artistic License」

前文

本書は、パッケージの使用者に事実上の慣習的方法においてパッケージを使用および配布する権利、および合理的変更を行う権利を付与しながらも、著作権所有者がパッケージ開発に関する一定の芸術的管理権を維持するように、パッケージの複製が許容される条件を示すことを目的としています。

定義:

- 「パッケージ」とは、著作権所有者が配布するファイルの集まり、および当該のファイルの集まりに対してテキストの変更を行うことで作成された派生物を指します。
- 「標準バージョン」とは、変更されていないパッケージ、または以下に規定する著作権所有者の意思に従って変更されたパッケージを指します。
- 「著作権所有者」とは、パッケージの著作権で指名された全ての人物を指します。
- 「お客様」とは、あなたが本パッケージを複製または配布しようとしている場合、あなた自身を指します。
- 「合理的複製料」とは、メディア費用、複製費用、関係者の拘束時間などに基づいてお客様が正当化できる料金を指します(お客様は、著作権所有者に対してではなく、当該料金を負担する市場としての一般的コンピュータ関係社会に対してのみ当該複製料を正当化するように求められます)。
- 「自由に利用可能」とは、商品自体に対して料金が課されないことを指しますが、商品の取り扱いに関する料金が課されることがあります。また、商品の受領者は、当該商品を受領したときと同じ条件に基づいて当該商品を再配布できます。

1. お客様は、元の著作権情報および関連する免責事項の全てを複製する限りにおいて、本パッケージの標準バージョンのソース形式の逐語的複製を無制限に作成および配布することができます。
2. お客様は、バグの修正、移植性の修正、およびパブリックドメインまたは著作権所有者に由来するその他の変更を加えることができます。そのような方法において変更されたパッケージは、標準バージョンと見なされるものとします。
3. いつどのような変更をファイルに対して行ったかを記述した明示的な情報を変更された各ファイルに挿入し、かつ以下の少なくとも1つに該当する限り、本パッケージの複写を、形を問わず別の方法で変更することができます。
 - a) お客様が行った変更事項をパブリックドメインに掲載する、またはそれらの変更事項を自由に利用可能な状態にする。これには、Usenetまたは同等の媒体へ前述の変更事項を掲示する、uunet.uu.netなどの大手アーカイブサイトへ変更事項を掲載する、お客様が行った変更事項を著作権所有者が本パッケージの標準バージョンに取り込むことを許可するなどの方法が該当します。
 - b) 変更したパッケージをお客様の会社または組織内においてのみ使用する。
 - c) 標準の実行ファイルと名前が競合しないように、非標準の全ての実行ファイルの名前を変更し、標準の実行ファイルと併せて提供する。また、非標準の実行ファイルごとに、標準バージョンとの差異を明確に記した個別のマニュアルページを提供する。
 - d) 著作権所有者と配布に関するその他の取り決めを行う。
4. 以下の少なくとも1つに該当する限り、お客様は本パッケージのプログラムをオブジェクトコードまたは実行ファイルの形式で配布することができます。
 - a) 実行ファイルおよびライブラリファイルの標準バージョンを、(マニュアルページまたはそれと同等のものにおいて)使用法とともに標準バージョンを入手できる場所において配布する。
 - b) コンピュータが解読できるパッケージのソースにお客様が行った変更事項を加えて配布する。
 - c) 非標準の実行ファイルに非標準の名前を付け、標準バージョンを入手できる場所において、その差異を使用法とともにマニュアルページ(または同等のもの)に明確に記す。
 - d) 著作権所有者と配布に関するその他の取り決めを行う。

5. お客様は、本パッケージの任意の配布に対して合理的複製料を課すことができます。お客様は、本パッケージのサポートに対して任意の料金を課すことができます。お客様は、本パッケージ自体に対する料金を課すことはできません。ただし、お客様が本パッケージをお客様自身の製品として広告しない限り、より大規模な(できる限り商用の)ソフトウェアの配布の一環として、その他の(できる限り商用の)プログラムと併せて本パッケージを配布することができます。お客様は、本パッケージのインタープリタを実行ファイルの内部に(リンク付けによって)埋め込むことができます。インタープリタの完全に標準のバージョンが同様に埋め込まれている場合、アグリゲーションに似せた形式で解釈される必要があります。
6. 本パッケージへのインプットとして供給される、または本パッケージからのアウトプットとして生成されるスクリプトおよびライブラリファイルは、自動的に本パッケージの著作権の影響を受けることはなく、それらを生成した人物に属するものとし、それらを商業的に販売したり、本パッケージと集約したりすることができます。そのようなスクリプトまたはライブラリファイルが、バイナリ実行可能イメージを生成するいわゆる「undump」または「unexec」の方法で本パッケージに集約される場合、お客様がそのような実行可能イメージを本パッケージの標準バージョンであると明言しない限り、そのようなイメージの配布が本パッケージの配布と解釈されたり、第3および第4段落の制限が適用されたりすることはないものとします。
7. 本パッケージで定義されている言語のサブルーチンおよびバリアブルをエミュレートするために、お客様によって供給され、本パッケージにリンクされるCサブルーチン(またはその他の言語で同等にコンパイルされたサブルーチン)は、当該ルーチンが、当該言語に対する回帰テストが失敗に終わる原因となるような方法で当該言語を変更しない限り、本パッケージの一部と見なされることはなく、第6段落のインプットに相当します。
8. 本パッケージの使用が埋め込まれている場合、すなわち、本パッケージのインターフェースが商業的配布のエンドユーザに見えるようにする公然たる試みが一切行われないう限り、商業的配布を伴う本パッケージのアグリゲーションは常に認められます。そのような使用は、本パッケージの配布とは解釈されません。
9. 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、著作権所有者の名称を使用することはできません。
10. 本パッケージは、「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。

終わり

GeoNamesの使用許諾条件

Pulse Secure Virtual Traffic Managerには、地理データベースであるGeoNamesのコードが含まれています(www.geonames.org)。以下の著作権情報が適用されます。

ライセンス概要: クリエイティブコモンズ表示4.0

著作権:

ライセンス権利(以下で定義)を施行することで、このクリエイティブコモンズ表示4.0国際公開ライセンス(「公開ライセンス」)の規約と条件に従うことを承諾および同意したことになります。この公開ライセンスが契約と解釈される範囲において、これらの規約と条件の承諾に対してライセンス権利が付与され、これらの規約と条件の下ライセンス済みのマテリアルを利用可能にすることでライセンサーが得た利点を考慮し、ライセンサーがそのような権利を付与します。

セクション 1 – 定義。

- **適用されるマテリアル:** ライセンスされたマテリアルに由来する、またはそれに基づく著作権および同等の権利の対象となり、ライセンスされたマテリアルが翻訳、変更、アレンジ、変形、またはライセンサーが保持する著作権および同等の権利に従い、許可が必要な方法で変更されたマテリアルを指します。このパブリックライセンスの目的として、ライセンスされたマテリアルが音楽作品、パフォーマンス、または録音であり、動く画像と時間の関連で同期された場合、適用されたマテリアルは常に生成されます。
- **適用者のライセンス:** このパブリックライセンスの規約と条件に従って適用されたマテリアルに対する使用において、適用した著作権および同等の権利に適用されるライセンスを指します。
- **著作権および同等の権利:** 権利がどのようにラベル付けまたは分類されたかを問わず、パフォーマンス、放送、録音、独自のデータベースの権利を含むがこれに限定されない、著作権に深く関連する、著作権および/または同等の権利を指します。このパブリックライセンスの目的として、セクション2(b)(1)-(2)で指定された権利は著作権および同等の権利ではありません。
- **有効技術方策:** 1996年12月20日に適用されたWIPO Copyright Treatyの記事11および/または同等の国際協定の義務を履行する法律の下、適切な権限がない場合、回避してはいけない方策を指します。
- **例外と制限:** ライセンスされたマテリアルの使用に適用される、著作権または同等の権利に対する、公平な使用、公平な取引、および/またはその他の例外や制限を指します。
- **ライセンスされたマテリアル:** ライセンサーがこのパブリックライセンスに適用した、芸術または文芸作品、データベース、またはその他のマテリアルを指します。

- **ライセンスされた権利:** このパブリックライセンスの規約と条件の下、付与された権利を指します。これは、ライセンスされたマテリアルに適用され、ライセンサーがライセンス付与の権限を持つ、全ての著作権および同等の権利に限定されます。
- **ライセンサー:** このパブリックライセンスに従い、権利を付与する個人または組織を指します。
- **共有:** マテリアルを何らかの方法で一般に提供すること、または、複製、公開展示、公開パフォーマンス、配布、流布、通信、輸入などの、ライセンスされた権利に従い許可が必要となるプロセス、および公衆が選択したある場所およびある時間にマテリアルにアクセスできるような方法で、マテリアルを公開することを指します。
- **独自のデータベース権利:** 改正および/または継続されたとおりの、データベースの法的保護についての、欧州議会および1996年3月11日の議会のDirective 96/9/ECの結果生じた著作権以外の権利、および世界各地の本質的に同等の権利を指します。
- **あなた:** このパブリックライセンスに従い、ライセンスされた権利を行使する個人または組織を指します。**あなたのはそれに対応する意味を持ちます。**

セクション 2 – 範囲。

- **ライセンス付与。**
 - このパブリックライセンスの規約および条件に従い、ここでいうライセンサーは、あなたに、全世界での、ロイヤリティのない、サブライセンス不可能な、非独占的で、取消不能なライセンスを、ライセンスされたマテリアルのライセンスされた権利を行使するために、次の目的で付与します。
 - A. ライセンスされたマテリアルの全体または一部を再現および共有すること、および
 - B. 適用されたマテリアルの製造、複製、および共有。
 - **除外および制限。** 除外および制限があなたの使用に適用される場合、このパブリックライセンスは適用されず、あなたはその規約および条件に準拠する必要があることを明確にしておきます。
 - **規約。** このパブリックライセンスの規約はセクション6(a)で規定されています。

- **メディアと形式: 技術的な修正は許可されます。**ライセンサーは、既知のものであれ今後作成するものであれ、ライセンスされた権利をあらゆるメディアおよび形式で行使し、必要な場合は技術的な修正を加えることを許可します。ライセンサーは、ライセンス権利を行使するために必要な技術的修正をあなたが行うことを禁じる権利の行使を放棄し、および/または同意しません。これには、有効な技術方策を避けるために必要な技術的修正が含まれます。このパブリックライセンスの目的において、このセクション2(a)(4)によって許可された修正を行うことが、適用されたマテリアルの生成につながることはありません。

- **下流の受領者。**

A. **ライセンサーからのオファー – ライセンスされたマテリアル。**ライセンスされたマテリアルの全ての受領者は、このパブリックライセンスの規約と条件に従い、ライセンスされた権利を行使するためのオファーを自動でライセンサーから受け取ります。

B. **下流の制限なし。**ライセンスされた受領者によるライセンス権利の行使が制限される場合、追加または別の規約や条件を課したり、ライセンスされたマテリアルに有効技術方策を適用してはいけません。

- **承認なし。**このパブリックライセンスには、あなたが、またはあなたによるライセンスされたマテリアルの使用が、ライセンサーやセクション3(a)(1)(A)(i)に記載された、属性を受領するよう指定された他者と関連があるか、スポンサーを受けるか、承認されるか、または公式ステータスを付与された場合に、その使用を行使または示唆するための権利として構成または解釈される個所はありません。

- **その他の権利。**

- 同一性保持権などの人格権は、このパブリックライセンスに従いライセンスされません。公表、プライバシー、および/またはその他の同等の人権も同様です。しかし、可能な範囲で、あなたがライセンス権利を行使できるようにするために限定された範囲で、ライセンサーはライセンサーにより保持されるそのような権利を行使することを放棄し、および/または同意しませんが、それ以外はその限りではありません。
- 特許および商標権はこのパブリックライセンスに従いライセンスされません。
- 可能な範囲で、ライセンサーは、ライセンスされた権利の行使によるロイヤリティをあなたから収集する権利を放棄します。これは、直接であるか、またはボランティアのまたは放棄可能な法定のまたは義務的なライセンススキームによる管理団体を通してであるかは問いません。その他全ての場合、ライセンサーはそのようなロイヤリティを収集する権利を明示的に留保します。

セクション3 – ライセンス条件。

あなたによるライセンス権利の行使には、以下の条件が明示的に適用されます。

- **属性。**
 - ライセンスされたマテリアルを共有する場合(修正された形式を含む)、次のことを行う必要があります。
 - A. ライセンスされたマテリアルとともにライセンサーから供給されたものである場合、以下を保持すること:
 - i. ライセンサーから要求された合理的な何らかの方法で、ライセンスされたマテリアルの作成者、および帰属を受けるよう指定されたその他全ての人物の身分証明(指定された場合はペンネームによる場合も含む)
 - ii. 著作権情報
 - iii. このパブリックライセンスに言及する注意書き
 - iv. 保証責任の排除に言及する注意書き
 - v. 合理的に実行可能な範囲での、ライセンスされたマテリアルへのURIまたはハイパーリンク
 - B. ライセンスされたマテリアルを変更したかどうかを示し、これまでの変更を記すこと、および
 - C. ライセンスされたマテリアルがこのパブリックライセンスに従いライセンスされたことを示し、このパブリックライセンスのテキスト、またはURIまたはハイパーリンクを含めること。
 - 媒体、方法、ライセンスされたマテリアルを共有する状況に基づいて、セクション3(a)(1)の条件を何らかの合理的方法で満たしてください。例えば、必要な情報を含むリソースへのURIやハイパーリンクを提示することで条件を満たすことは合理的です。
 - ライセンサーから求められた場合、セクション3(a)(1)(A)によって必要とされた情報を、合理的に実行可能な範囲で削除する必要があります。
 - 自分で作成した適用されたマテリアルを共有する場合、適用した適用者のライセンスは、適用されたマテリアルの受領者によるこのパブリックライセンスへの準拠の妨げにはなりません。

セクション4 – 独自のデータベース権利。

ライセンスされた権利に、ライセンスされたマテリアルの使用に適用される独自のデータベース権利が含まれる場合:

- 誤解を避けるため、セクション2(a)(1)では、データベースのコンテンツの全てまたは一部を抽出、再使用、および共有する権利を付与しています。
- 独自のデータベース権利を持つデータベース内に、データベースのコンテンツの全てまたは一部を含める場合、独自のデータベース権利(個別のコンテンツではない)を持つデータベースは適用されたマテリアルであり、
- データベースのコンテンツの全てまたは一部を共有した場合、セクション3(a)の条件に準拠する必要があります。

このセクション4は、ライセンスされた権利が他の著作権および同等の権利を含む、このパブリックライセンスの下での義務を補完するものであり、置き換えるものではないことを明確にしておきます。

セクション 5 – 保証責任の排除および責任の制限。

- ライセンサーから個別に保証されない限り、ライセンサーはライセンスされたマテリアルをそのまま、および利用可能な状態で提供し、明示、暗黙、法定、およびその他のいずれかを問わず、ライセンスされたマテリアルに関して一切の表明または保証を行いません。これには、タイトル、商品性、特定の目的への適用性、非侵害性、潜在的またはその他の欠陥がないこと、正確性、または問題の有無、既知であるか発見可能であるかが含まれますがこれらに限定されません。保証責任の排除が完全または一部許可されない場合、この責任の排除は適用されません。
- 可能な範囲で、何らかの法理論(過失を含むがこれに限定されない)において、または直接的、特別、間接的、偶発的、結果的、懲罰的、模範的、またはこのパブリックライセンスやライセンスされたマテリアルの使用により生じた、その他の損失、費用、支出、損傷に対して、ライセンサーがあなたに対して責任を負うことはありません。ライセンサーがそのような損失、費用、支出、損傷の可能性について助言を受けていた場合も含まれます。責任の制限が完全または一部許可されない場合、この制限は適用されません。
- 上述の保証責任の排除と責任の制限は、可能な範囲で、絶対的な責任の排除および全ての責任の権利放棄に最も近い方法で解釈する必要があります。

セクション6 – 規約と終了。

- 著作権および同等の権利の規約に適用するこのパブリックライセンスは、ここでライセンスされます。ただし、このパブリックライセンスに準拠できない場合、このパブリックライセンスにおける権利は自動で終了します。
- ライセンスされたマテリアルの使用の権利がセクション6(a)で終了した場合、
 - 違反の発見から30日以内に修復されることを条件に、違反が修復された日付をもって自動的に、または
 - ライセンサーによる明示的な回復をもって回復されます。

このセクション6(b)は、このパブリックライセンスの違反に対して、ライセンサーが修復案を探すために持つことができる一切の権利に影響を与えないことを明確にしておきます。

- ライセンサーはまたライセンスされたマテリアルを個別の規約や条件に従い提供し、またはライセンスされたマテリアルの配布を停止する場合があります。しかし、これらの行為によってこのパブリックライセンスが終了することはないことを明確にしておきます。
- セクション1、5、6、7、および8は、このパブリックライセンスが終了しても存続します。

セクション7 – その他の規約と条件。

- ライセンサーは、明示的な合意がない限り、あなたが伝えた追加のまたは別の規約または条件に縛られることはありません。
- ここに記載されていない、ライセンスされたマテリアルに関するあらゆる合意、理解、同意は、このパブリックライセンスの規約と条件とは無関係であり、独立しています。

セクション8 – 解釈。

- このパブリックライセンスは、このパブリックライセンスに従い、許可なく合法的に行われる可能性のある、ライセンスされたマテリアルのあらゆる使用の条件を、解釈、削減、制限、または強制するものではなく、そうであってはいけないことを明確にしておきます。
- 可能な範囲で、このパブリックライセンスの配布が法的強制力のないものとして見なされる場合、法的強制力のあるものにするために必要な最低限の範囲で、自動的に改善されなければなりません。配布を改善できない場合は、他の規約と条件の法的強制力に影響を与えることなく、このパブリックライセンスから分離されなければなりません。

- このパブリックライセンスの規約または条件は一切破棄されることなく、ライセンサーから明示的に同意されない限り、同意に準拠しなければなりません。
- このパブリックライセンスには、何らかの司法権または機関の法的処置を含め、ライセンサーまたはあなたに適用される、あらゆる権利および免除を制限し、または放棄するものとして構成され、または解釈される箇所はありません。

LibYAMLの使用許諾条件

Pulse Secure Virtual Traffic Managerには、YAMLを解析および出力するためのCライブラリであるLibYAMLのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 0.2.1

ライセンス概要: MIT

著作権:

Copyright (c) 2017-2019 Ingy döt Net

Copyright (c) 2006-2016 Kirill Simonov

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

giflibの使用許諾条件

Pulse Secure Virtual Traffic Managerには、gif画像の読み書き用のライブラリであるgiflibが含まれています。以下の著作権情報が適用されます。

バージョン: 5.2.1

ライセンス概要: MIT

著作権:

The GIFLIB distribution is Copyright (c) 1997 Eric S. Raymond

本ライセンスにより、本ソフトウェアおよび付随するドキュメントファイル(以下「ソフトウェア」)のコピーを入手した全ての人は、以下の条件に従い、ソフトウェアのコピーを使用、コピー、変更、マージ、公開、配布、サブライセンス、および/または販売する無制限の権利を含め、ソフトウェアを自由に扱うことが無償で許諾されます。また、ソフトウェアを提供された人についても同様です。

上記の著作権情報および本使用許諾情報をソフトウェアの全てのコピーまたは重要な部分に含めるものとします。

ソフトウェアは、商品性、特定目的への適合性、権利侵害がないことの保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証のない状態で「現状のまま」提供されるものです。いかなる場合も、作成者または著作権所有者は、ソフトウェアから、またはソフトウェアに関連して、あるいはソフトウェアの使用またはその他の扱いによって生じる申し立て、損傷、またはその他責任に対し、契約の行為、不法行為、またはその他の有無を問わず、一切責任を負いません。

Gorilla WebSocketの使用許諾条件

Pulse Secure Virtual Traffic Managerには、GoLang用のWebSocketの実装であるGorilla WebSocketのコードが含まれています。以下の著作権情報が適用されます。

バージョン: 1.4.0

ライセンス概要: BSD 2-Clause

著作権:

Copyright (c) 2013 The Gorilla WebSocket Authors. 著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

- ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
- バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。

本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、特定目的の暗黙の商品性及び適合性を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

Goの使用許諾条件

Pulse Secure Virtual Traffic Managerには、Go(オープンソースプログラミング言語)を使用したコードが含まれています。以下の著作権情報が適用されます。

バージョン: 1.15.7

ライセンス概要: BSD 3-Clause

著作権:

Copyright (c) 2009 The Go Authors. 著作権所有。

ソースおよびバイナリ形式での再配布および使用は、変更の有無を問わず、以下の条件を満たしていることを条件に許諾されます。

- ソースコードの再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項を含める必要があります。
- バイナリ形式での再配布物には、上記の著作権情報、この条件リスト、および以下の免責事項をドキュメントおよび/または再配布物とともに提供されるその他資料に複製する必要があります。
- 事前に書面による明確な許諾を得ない限り、本ソフトウェアから派生した製品を支持または促進する目的で、Google Inc.の名称および貢献者の氏名を使用することはできません。

本ソフトウェアは、著作権所有者および貢献者により「現状のまま」提供されるものであり、商品性および特定目的への適合性の暗黙的保証を含め(ただしこれらに限定されない)、明示であると暗黙であるとを問わず、何ら保証されません。いかなる場合も、著作権所有者または貢献者は、本ソフトウェアの使用により生じた直接的、間接的、偶発的、例外的、典型的、または必然的損傷(代替品またはサービスの調達、使用、データ、または利益の喪失、あるいは事業中断を含むが、これらに限定されない)に対し、その経緯および基づく責任の法理に関わらず、また契約、厳格責任、または不法行為(過失その他を含む)の有無を問わず、一切責任を負いません。このような損傷の可能性を知らされていた場合も同様です。

オープンソースライセンスの注意点

この商品にはサードパーティで作成されたソフトウェアが含まれ、以下の1つ以上が適用されます。

- The GNU Library/Lesser General Public License (LGPL)
- The GNU General Public License (GPL)
- The Berkeley Software Distribution (BSD) License
- The OSI Artistic License
- 各種のGPL/BSD形式分散ライセンス

適用されるサードパーティの全てのソフトウェアパッケージおよびそれに付随するライセンスは本書、Pulse Secure Virtual Traffic Manager: Appliance License Acknowledgements、およびIvantiのWebサイトのTraffic Manager製品ページにあるにリストがあります。

Ivanti, Inc.は、前述のライセンスに従いソフトウェアの全ソースコードをCD-ROMにより、実費でご提供しています。この費用には、メディア代、送料、管理費用など配布に必要な費用が含まれます。Ivanti, Inc.まで書面でご請求ください。住所は以下のとおりです。

Source Code Requests VTM-APPLIANCE (GPL)

Ivanti, Inc.

The Jeffreys Building

Cowley Road

Cambridge

CB4 0DS

United Kingdom

提供期間は、本製品をIvanti, Inc.が販売後3年間です。お客様の権利について、該当するライセンスの各条項を参照してください。