



Workload Optimization Manager 3.2.1 インストールガイド

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとしします。このマニュアルに記載されている製品の使用は、すべてユーザー側の責任になります。

対象製品のソフトウェアライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

シスコが導入する TCP ヘッダー圧縮は、カリフォルニア大学バークレー校 (UCB) により、UNIX オペレーティングシステムの UCB パブリック ドメイン バージョンの一部として開発されたプログラムを適応したものです。All rights reserved. Copyright © 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。Cisco およびこれら各社は、商品性の保証、特定目的への準拠の保証、および権利を侵害しないことに関する保証、あるいは取引過程、使用、取引慣行によって発生する保証をはじめとする、明示されたまたは黙示された一切の保証の責任を負わないものとしします。

いかなる場合においても、Cisco およびその供給者は、このマニュアルに適用できるまたは適用できないことによって、発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性が Cisco またはその供給者に知らされていても、それらに対する責任を一切負わないものとしします。

このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号を示すものではありません。マニュアル内の例、コマンド出力、ネットワークポロジ図、およびその他の図は、説明のみを目的として使用されています。説明の中に実際のアドレスおよび電話番号が使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

ハード コピーおよびソフト コピーの複製は公式版とみなされません。最新版はオンライン版を参照してください。

Cisco は世界各国 200 箇所にオフィスを開設しています。各オフィスの住所、電話番号、FAX 番号については、Cisco のウェブサイト www.cisco.com/go/offices をご覧ください。

Cisco および Cisco ロゴは、シスコ またはその関連会社の米国およびその他の国における商標または登録商標です。シスコの商標の一覧は、www.cisco.com/go/trademarks でご確認いただけます。掲載されている第三者の商標はそれぞれの権利者の財産です。「パートナー」または「partner」という用語は、シスコと他社との間のパートナーシップ関係を意味するものではありません。(1110R)

Copyright © 2021 Cisco, all rights reserved

目次

概要.....	5
最小要件.....	6
仮想マシンイメージへのインストール.....	8
OVA : vCenter イメージのインストール.....	9
VHD : Microsoft Hyper-V イメージのインストール.....	10
Workload Optimization Manager コンポーネントの展開.....	12
全般設定タスク.....	17
(必須) 時刻の同期.....	17
(重要) MariaDB バージョンの確認.....	19
使用可能なディスク容量の増加.....	22
(任意) LDAP を介したセキュアなアクセスの適用.....	25
自己署名証明書のインストール.....	25
(任意) 信頼できる証明書を介したセキュアなアクセスの適用.....	27
(任意) プローブのセキュアなアクセスの有効化.....	31
(任意) Cluster Manager の証明書の変更.....	34
(任意) 組み込みレポートの有効化.....	35
(任意) レポートの編集、PDF、および電子メールサブスクリプション.....	40
組み込みレポートのストレージ要件の見積もり.....	42
(任意) データエクスポートの有効化.....	44
(任意) プラットフォームノードの IP アドレスの変更.....	49
(任意) プローブコンポーネントの有効化と無効化.....	51
ライセンスのインストールおよび初回ログイン.....	54
シングル サインオン認証.....	55
SAML 認証の設定.....	56
IdP メタデータの例.....	59
OpenID 認証の設定.....	60

シングルサインオンの無効化.....	64
Workload Optimization Manager の新規バージョンへの更新.....	66
更新前の確認.....	67
外部 DB と Workload Optimization Manager の更新.....	68
オフライン更新.....	71
付録：IdP の一般的な設定.....	74
付録：段階的なプラットフォーム展開.....	76
付録：段階的なオフライン更新.....	80
付録：Workload Optimization Manager の Classic から XL への移行.....	83
Workload Optimization Manager インスタンスの準備.....	84
移行ツールのインストール.....	86
移行の実行.....	86
新しいバージョンへの切り替え.....	91



概要

仮想環境の Intelligent Workload Automation Management ソリューションである Workload Optimization Manager をお選びいただきありがとうございます。このガイドは、仮想化環境に Workload Optimization Manager をインストールし、ライセンスをインストールし、リソースの管理を開始するのに必要な情報を提供します。

ご不明な点がございましたら、シスコ サポートまでお問い合わせください。

最後まで読んでいただき、ありがとうございました。

Workload Optimization Manager チーム



最小要件

ライセンス要件

使用環境で Workload Optimization Manager を実行するには、適切なライセンスをインストールする必要があります。ライセンスにより、Workload Optimization Manager の異なる機能のセットが有効になり、使用環境内で指定された数のワークロードがサポートされます。

ユーザーインターフェイスの要件

Workload Optimization Manager のユーザーインターフェイスを表示するには、HTML5 のページを表示できるブラウザでプラットフォームにログインする必要があります。現在、Workload Optimization Manager は次のブラウザをサポートしています。

- Apple Safari
- Google Chrome
- Microsoft Edge
- Mozilla Firefox

ネットワークアドレッシングの要件

Workload Optimization Manager には、静的 IP アドレッシングが必要です。静的 IP のセットアップは、Workload Optimization Manager VM イメージのインストール手順で説明されています。

コンピューティングとストレージの要件

Workload Optimization Manager インスタンスを実行するための要件は、管理している環境のサイズによって異なります。Workload Optimization Manager では、環境のリアルタイム表現がメモリに保持されます。管理するエンティティの数が多く、エンティティ間の関係が広ければ広いほど、Workload Optimization Manager を実行する VM に必要なリソースも多くなります。また、VM の要件が増えると、VM をホストする物理マシンの要件も増えます。

ここにリストされている要件は、Workload Optimization Manager の展開を計画する際に留意する必要がある推奨事項です。展開後、VM のメモリ容量、CPU 容量、または両方を変更する必要がある場合は、VM をシャットダウンして変更を加えた後、再度電源を投入することで新しい容量を使用できます。

注：

Workload Optimization Manager プラットフォームをホストするマシンは、SSE4.2 命令セットをサポートしている必要があります。この命令セットのサポートの導入時期は、チップメーカーごとに異なっています。

- Intel : 2008 年 11 月
- AMD : 2011 年 10 月

Workload Optimization Manager をホストするために使用するマシンは、これらの日付以降に製造されている必要があります。Linux システムでは、次のコマンドを実行してこの命令セットのサポートを確認できます。

```
cat /proc/cpuinfo | grep sse4
```

詳細については、<http://www.cpu-world.com/Glossary/S/SSE4.html> にある用語集エントリを参照してください。

ほとんどの場合、次の最小要件を満たすホストで Workload Optimization Manager を実行できます。

サポート対象の VM イメージテクノロジー		ストレージ要件	メモリ	CPU
VMware	vCenter バージョン 5.5、6.0、6.5、6.7、および 7.0	1.25 TB 以上。 注： ストレージの要件によってシンプロビジョニング可能です。	<ul style="list-style-type: none"> ■ デフォルト : 128 GB ■ 10,000 VM 以下の場合は、64 GB 	8 vCPU
Microsoft	Hyper-V Server 2012 R2 以降			

シスコでは、事前設定された VM イメージ (OVA または VHD ファイル) を 2 つのハードドライブで提供しています。適切な容量のストレージをドライブに確保するには、最低 1.25 TB が必要です。



仮想マシンイメージへのインストール

Workload Optimization Manager プラットフォームは、次のイメージとしてダウンロードできます。

- VMware OVA 1.0 イメージ
- Microsoft Hyper-V イメージ

注：

最小要件として、Workload Optimization Manager をホストする VM には 128 GB のメモリをお勧めします。ただし、小規模な環境（10,000 VM 以下）を管理する予定の場合は、64 GB のメモリを備えた VM にインストールできます（[「最小要件」 \(6 ページ\)](#) を参照）。

64 GB のメモリを備えた VM をインストールする場合は、VM メモリのデフォルトを変更する必要があります（[「Workload Optimization Manager VM の展開」 \(9 ページ\)](#) を参照）。

プラットフォームは、次の 2 つの主な手順でインストールします。

1. Workload Optimization Manager VM イメージをネットワークにインストールします。
Workload Optimization Manager プラットフォームのインスタンスをホストする VM がインストールされて起動します。
2. Workload Optimization Manager コンポーネントを VM に展開します。

Workload Optimization Manager VM イメージについて

Workload Optimization Manager は、CentOS Linux OS を実行する VM としてインストールされます。新しいバージョンごとに、製品を実行するためにインストールする VM イメージ（OVA または VHD）が提供されます。通常は、VM イメージを 1 回インストールします。後続の Workload Optimization Manager の更新では、インストールされた VM で製品の更新を実行します。これは次の 2 つのことを意味します。

- 製品の更新では、Workload Optimization Manager アプリケーションスタックの新しいコンポーネントのパッチが、最初に VM イメージをインストールしたときに取得したのと同じ CentOS プラットフォームに適用されます。製品の更新は、基盤となる OS に影響を及ぼしません。
- 時間の経過とともに、CentOS ディストリビューションの重要なセキュリティパッチが提供されることがあります。OS を最新の状態に保つのはユーザーの責任です。それらのパッチは、必要なときにいつでも Workload Optimization Manager VM にインストールできます。

注：

シスコは、現在、CentOS Linux OS で VM イメージをリリースしています。CentOS Linux OS は、全体的なセキュリティ要件を満たしていることを確認しています。CentOS Linux OS のプラットフォームが実行可能で安全である限り、CentOS を使用し続ける予定です。

OVA : vCenter イメージのインストール

Workload Optimization Manager をインストールする場合、最初にプラットフォームをホストする VM を展開します。

vCenter Server 環境の場合、各四半期リリースで OVA イメージが提供されます。vCenter Server で Workload Optimization Manager を実行する場合は、四半期リリースをインストールしてから、必要に応じて新しいポイントリリースに更新できます。

注：

最小要件として、Workload Optimization Manager をホストする VM には 128 GB のメモリをお勧めします。ただし、小規模な環境（10,000 VM 以下）を管理する予定の場合は、64 GB のメモリを備えた VM にインストールできます（[「最小要件」](#)（6 ページ）を参照）。

64 GB のメモリを備えた VM をインストールする場合は、VM メモリのデフォルトを変更する必要があります（[「Workload Optimization Manager VM の展開」](#)（9 ページ）を参照）。

Workload Optimization Manager OVA をインストールするには、次の手順を実行します。

1. Workload Optimization Manager インストール パッケージをダウンロードします。

Workload Optimization Manager Software Download ページ

（<https://software.cisco.com/download/home/286328879/type/286317011/release>）に移動して、最新の OVA イメージへのリンクを確認します。

インストールパッケージには、`cisco_cwom-<version>-<XXXXXXXXXXXXXXXXXX>.ova` ファイルが含まれています。<version> は、Workload Optimization Manager のバージョン番号、<XXXXXXXXXXXXXXXXXX> はタイムスタンプです。

例：`cisco_cwom-3.0.0-20190916164429000.ova`

OVA ファイルは、インストールの準備ができている Workload Optimization Manager コンポーネントを備えた VM として展開されます。

2. OVA ファイルをデータセンターにインポートします。

vCenter Server クライアントを使用して、OVA を使用環境にインポートします。

3. Workload Optimization Manager VM を展開します。OVA ファイルから展開された VM を設定します。

最小要件として、Workload Optimization Manager をホストする VM には 128 GB のメモリをお勧めします。ただし、小規模な環境（10,000 VM 以下）を管理する予定の場合は、64 GB のメモリを備えた VM にインストールできます（[「最小要件」](#)（6 ページ）を参照）。

64 GB のメモリを備えた VM を展開する場合は、メモリのデフォルト値を手動で変更します。

- a. VM を右クリックして、[Edit Settings] を選択します。
- b. [Memory] に **64** と入力します。
- c. [OK] をクリックして、設定を保存します。
- d. VM の電源を投入します。

4. リモートコンソールを開きます。

展開直後の Workload Optimization Manager VM の場合：

- a. [Summary] タブを選択します。
- b. [Launch Remote Console] をクリックします。

5. Workload Optimization Manager システム管理者アカウントをセットアップします。

- a. リモートコンソールで、次のデフォルトログイン情報でログインします。

- ユーザー名：turbo
アカウント名 root は使用しないでください。
- パスワード：vmturbo

次に、新しいパスワードの入力を求められます。

- b. 新しいパスワードを入力します。

新しいパスワードは、強力なパスワードポリシー（大文字と小文字、数字、記号の組み合わせ）に準拠している必要があります。新しいパスワードは誰にも教えないでください。

注：

変更したアカウントのログイン情報は安全な場所に保存してください。セキュリティ上の理由から、このアカウントが、Workload Optimization Manager VM にアクセスして設定できる唯一のアカウントです。

- c. 新しいパスワードを再度入力して確認します。

6. ルートパスワードを更新します。

プラットフォームでは、`/var/log/messages` にあるログメッセージのロールアップなど、特定のプロセスに `root` アカウントが使用されます。アカウントのログイン情報が最新であることを確認するには、パスワードを変更する必要があります。

- a. スーパーユーザーセッションを開きます。

- リモートコンソールで、`su -` と入力します。
- パスワードプロンプトで、デフォルトのパスワード (`vmturbo`) を入力します。

- b. 新しいパスワードをリセットします。

デフォルトのパスワードを使用して `root` としてログインすると、新しいパスワードの入力を求めるプロンプトが表示されます。この新しいパスワードは、強力なパスワードポリシー（大文字と小文字、数字、記号の組み合わせ）に準拠している必要があります。

新しいパスワードは誰にも教えないでください。

注：

`root` アカウントのログイン情報は安全な場所に保存してください。

- c. スーパーユーザーセッションを終了します。

`exit` と入力します。

7. その他の必要な設定手順を実行してから、Workload Optimization Manager コンポーネントをインストールします。

Workload Optimization Manager インスタンスに必要な重要な設定手順を実行するには、[「全般設定タスク」 \(17 ページ\)](#) を参照してください。

Workload Optimization Manager コンポーネントをインストールするには、[「Workload Optimization Manager コンポーネントの展開」 \(76 ページ\)](#) を参照してください。

VHD : Microsoft Hyper-V イメージのインストール

Workload Optimization Manager をインストールする場合、最初にプラットフォームをホストする VM を展開します。

Hyper-V 環境の場合、各四半期リリースで Hyper-V イメージが提供されます。Hyper-V VM で Workload Optimization Manager を実行する場合は、四半期リリースをインストールしてから、必要に応じて新しいポイントリリースに更新できます。

注：

最小要件として、Workload Optimization Manager をホストする VM には 128 GB のメモリをお勧めします。ただし、小規模な環境（10,000 VM 以下）を管理する予定の場合は、64 GB のメモリを備えた VM にインストールできます（[「最小要件」 \(6 ページ\)](#) を参照）。

64 GB のメモリを備えた VM をインストールする場合は、VM メモリのデフォルトを変更する必要があります（[「Workload Optimization Manager VM の展開」 \(9 ページ\)](#) を参照）。

Workload Optimization Manager をインストールするには？

1. Workload Optimization Manager インストール パッケージをダウンロードします。

Workload Optimization Manager Software Download ページ

(<https://software.cisco.com/download/home/286328879/type/286317011/release>) に移動して、最新の Hyper-V イメージへのリンクを確認します。

2. .Zip ファイルを展開し内容をコピーして、Hyper-V サーバに（またクラスタ共有ボリュームかローカル ハード ドライブのどちらかに）仮想マシン イメージを含みます。
3. Hyper-V マネージャの仮想マシン インポート ウィザードを使用して、環境内に仮想マシンをインポートします。
4. 仮想ネットワーク アダプタが正しい仮想ネットワークに接続されていることを確認します。
5. Workload Optimization Manager インスタンスに十分なメモリが必要です。

シスコでは、Workload Optimization Manager インスタンスに静的メモリを使用することを推奨します。ただし、インスタンスに静的または動的なメモリを指定することができます。デフォルトのインストールでは、静的メモリは 128 GB に設定されます。

6. Workload Optimization Manager アプライアンスを開始し、その IP アドレスを記録します。
7. Workload Optimization Manager システム管理者アカウントをセットアップします。

- a. 次のデフォルトのログイン情報で VM の Hyper-V コンソールにログインします。

- ユーザー名：turbo
アカウント名 root は使用しないでください。
- パスワード：vmturbo
次に、新しいパスワードの入力を求められます。

- b. 新しいパスワードを入力します。

新しいパスワードは、強力なパスワードポリシー（大文字と小文字、数字、記号の組み合わせ）に準拠している必要があります。新しいパスワードは誰にも教えないでください。

注：

変更したアカウントのログイン情報は安全な場所に保存してください。セキュリティ上の理由から、このアカウントが、Workload Optimization Manager VM にアクセスして設定できる唯一のアカウントです。

- c. 新しいパスワードを再度入力して確認します。

8. ルートパスワードを更新します。

プラットフォームでは、/var/log/messages にあるログメッセージのロールアップなど、特定のプロセスに root アカウントが使用されます。アカウントのログイン情報が最新であることを確認するには、パスワードを変更する必要があります。

- a. スーパーユーザーセッションを開きます。

- リモートコンソールで、su - と入力します。
- パスワードプロンプトで、デフォルトのパスワード (vmturbo) を入力します。

- b. 新しいパスワードをリセットします。

デフォルトのパスワードを使用して root としてログインすると、新しいパスワードの入力を求めるプロンプトが表示されます。この新しいパスワードは、強力なパスワードポリシー（大文字と小文字、数字、記号の組み合わせ）に準拠している必要があります。

新しいパスワードは誰にも教えないでください。

注：

root アカウントのログイン情報は安全な場所に保存してください。

- c. スーパーユーザーセッションを終了します。

exit と入力します。

9. インストールされた VM の NIC を設定します。

Workload Optimization Manager インスタンスの設定には、2 つの NIC が含まれていますが、各 NIC は有効になっていないか、ネットワークに接続されていません。Hyper-V マネージャで NIC を表示して設定します。

- NIC #1 :
クラスタのネットワーク要件に合わせてネットワーク VLAN の設定を設定します。
- NIC #2 :
NIC を有効にします。

10. その他の必要な設定手順を実行してから、Workload Optimization Manager コンポーネントをインストールします。

Workload Optimization Manager インスタンスに必要な重要な設定手順を実行するには、[「全般設定タスク」 \(17 ページ\)](#) を参照してください。

Workload Optimization Manager コンポーネントをインストールするには、[「Workload Optimization Manager コンポーネントの展開」 \(12 ページ\)](#) を参照してください。

Workload Optimization Manager コンポーネントの展開

注：

このセクションでは、デフォルトのインストールプロセスについて説明します。インストールをカスタマイズする場合は、[「付録：段階的なプラットフォーム展開」 \(76 ページ\)](#) の手順を実行することを検討してください。

たとえば、次のことを実行するには、段階的なインストールを実行する必要があります。

- 展開の Kubernetes ホスト名を変更する
- Workload Optimization Manager VM のプロキシサーバーを設定する

プラットフォームをホストする Workload Optimization Manager VM をインストールしたら、次の手順でプラットフォーム コンポーネントをインストールできます。

まず、インストールの実行に必要な情報を収集します。

■ 時刻同期用のネットワークタイムソース (任意)

この手順は、インストール中に、または後で実行できます。VM のクロックを今すぐ同期する場合は、ネットワークタイムソースの入力を求められます。VM のクロックの同期の詳細については、[「時刻の同期」 \(17 ページ\)](#) を参照してください。

■ 更新されたルートパスワード

インストールスクリプトを実行するには、VM のルートパスワードを更新しておく必要があります。[「OVA : vCenter イメージのインストール」 \(9 ページ\)](#) または [「VHD : vCenter イメージのインストール」 \(10 ページ\)](#) の手順に従った場合、この手順はすでに実行しているはずですが。

必要な情報を準備できたら、インストールスクリプトを実行できます。

1. インストールスクリプトを起動します。

Workload Optimization Manager VM で、turbo ユーザーとしてセキュアセッション (SSH) を開始し、インストールスクリプトを実行します。

```
sudo t8cInstall.sh
```

2. Workload Optimization Manager VM に静的 IP アドレスを設定していることを確認します。

スクリプトの最初の手順で、次のプロンプトが表示されます。

```
Have you run the ipsetup script to setup networking yet? [y/n] n
```

プラットフォーム VM の静的 IP を設定していない場合は、`n` を入力してインストールスクリプトを終了し、静的 IP を設定します。プラットフォーム VM の静的 IP をすでに設定している場合は、`y` を入力してインストールを続行します。スクリプトの出力には、VM に対して認識される IP アドレスが表示されます。次に例を示します。

```
-----
Old IP Address: 10.0.2.15
New IP Address: 10.10.123.123
-----
```

注：

プラットフォーム コンポーネントと VM アドレス間の依存関係により、コンポーネントのインストール後は IP アドレスを簡単に変更できません。Workload Optimization Manager の実稼働インストールの場合、VM は静的 IP で実行する必要があります。テストまたは評価インストールの場合、DHCP を使用できますが、後に実稼働環境で使用する予定がある場合は、必ず静的 IP を設定する必要があります。

[「OVA : vCenter イメージのインストール」 \(9 ページ\)](#) または [「VHD : vCenter イメージのインストール」 \(10 ページ\)](#) の手順に従った場合、`ipsetup.sh` スクリプトを実行して、すでに設定しているはずで

3. スクリプトによりインストールが実行されている間は待機します。インストールプロセスの進行中、スクリプトにより以下の作業が実行されます。

- 必要な証明書を使用したプラットフォーム環境を設定する
- VM で Kubernetes クラスタを設定する

この設定は、成功するまで数回試行されることがあります。試行が失敗するたびに、次のようなメッセージが表示されます。

```
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

接続に成功すると、スクリプトは次の手順に進みます。

- プラットフォームのローカルストレージを確立する
- プラットフォームの kubernetes 名前空間を `turbonomic` として作成する
- 必要な Kubernetes シークレットにアクセスするための承認を設定する
- プラットフォームの履歴データを管理するために MariaDB データベースサーバーを初期化する

このスクリプトでは、MariaDB に完全な権限を持つ 2 つのアカウントが作成されます。

```
- root@localhost
```

このアカウントにはパスワードは不要です。このアカウントを介して接続するには、ユーザーは `system root` である必要があります。

```
- mysql@localhost
```

このアカウントにはパスワードは不要です。このアカウントを介して接続するには、ユーザーは `system mysql` である必要があります。

これらのアカウントには、`sudo` 経由で接続できます。例：`sudo mysql`。接続後、これらのアカウントにパスワードを設定できます。詳細については、<https://mariadb.com/kb> にある MariaDB ナレッジベースを参照してください。

- 組み込みレポート用のタイムスケールデータベースとデータエクスポートをインストールする
- プラットフォーム コンポーネントを展開して起動する

展開が開始されると、次の内容が出力されます。

```
#####
Start the deployment rollout
#####
```

コンポーネントが展開されたら、コンポーネントが起動するのを待ちます。

```
The installation process is complete, waiting for all the components to start up.
** The script will wait for as long as 30 minutes. **
```

すべてのコンポーネントが 30 分以内に起動した場合、インストールは完了し、成功しています。

すべてのコンポーネントが 30 分以内に起動しない場合、次の内容が表示され、スクリプトが終了します。

```
=====
One or more of your deployments has not started up yet.
** Please give your environment another 30 minutes to stabilize. **
To check the status of your components, execute the following command:
kubectl get pods
If some components are still not ready, contact your support representative
Deployments not ready:
```

次に、`kubectl get pods` コマンドのフォーマットされた結果が表示され、Workload Optimization Manager プラットフォームのポッドの現在のステータスがわかります。

注：

コンポーネントがすべて起動する前にスクリプトが終了した場合は、さらに 30 分待つことをお勧めします。コンポーネントのステータスを定期的に確認するには、`kubectl get pods` を実行します。さらに 30 分待ってもすべてのコンポーネントが起動しない場合は、サポート担当者に連絡してください。

インストールが成功し、コンポーネントがすべて起動している場合、次のようなメッセージが表示され、VM の静的 IP アドレスが示されます。

```
#####
Deployment Completed, please login through the UI
https://10.10.123.123
#####
```

これで、次の手順に進めます。

4. プラットフォームのマスターキーシークレットのコピーを保存します。

インストール手順により、Kubernetes クラスタにマスターキーシークレットが作成されます。Workload Optimization Manager は、このシークレットを使用して、プラットフォーム コンポーネントへのアクセスを提供します。キーデータは安全な場所に保存する必要があります。何らかの理由でキーデータが破損したり、使用できなくなったりすると、Workload Optimization Manager は動作しなくなります。動作しなくなった場合は、サポート担当者に連絡し、保存したキーデータを使用してプラットフォームを回復できます。

データを保存するには、次の手順を実行します。

a. プラットフォーム シークレットを一覧表示します。

コマンドを実行します：

```
kubectl get secrets
```

結果には、次のようなマスターキーシークレットが含まれます。

```
...
master-key-secret          Opaque          1          57d
...
```

b. マスターキーデータを表示します。

マスターキーの名前が見つかったら、キーデータを表示できます。

```
get secret master-key-secret -o yaml
```

コマンドの結果は次のようになります。

```
apiVersion: v1
data:
  primary_key_256.out: AfnJWutxNHAduaIOdAii3DRA2fMa6lzX4rWetZxxZvc=
kind: Secret
metadata:
  creationTimestamp: "2021-06-30T02:59:19Z"
  managedFields:
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:data:
        .: {}
        f:primary_key_256.out: {}
      f:type: {}
    manager: kubectl-create
    operation: Update
    time: "2021-06-30T02:59:19Z"
  name: master-key-secret
  namespace: turbonomic
  resourceVersion: "1072"
  uid: a314b2ba-2061-4b41-b844-56caf2c3728d
type: Opaque
```

保存する重要なキーデータは、`primary_key...` データです。前述の例では、次の行を保存する必要があります。

```
primary_key_256.out: AfnJWutxNHAduaIOdAii3DRA2fMa6lzX4rWetZxxZvc=
```

c. データを安全な場所に保存します。

このデータをファイルに書き込み、安全なバックアップ場所に保存します。マスターキーを回復する必要がある場合は、サポート担当者がこのデータを使用して回復を実行します。

5. Workload Optimization Manager のユーザーインターフェイスにログインし、管理者ユーザーアカウントのパスワードを設定します。

コンポーネントが起動したら、Web ブラウザに移動し、Workload Optimization Manager VM の静的 IP アドレスを入力します。Workload Optimization Manager ユーザーのログインページにリダイレクトされます。

Workload Optimization Manager には、ADMINISTRATOR ロールを持つ `administrator` という名前のデフォルトのユーザーアカウントが含まれています。初めてログインするときは、そのアカウントに独自のパスワードを設定する必要があります。ADMINISTRATOR ロールを持つ他のアカウントの作成や削除は可能ですが、Workload Optimization Manager のインストールには、ADMINISTRATOR ロールを持つアカウントが常に少なくとも 1 つ必要です。

ログインページで、必要に応じて情報を入力し、メモしておきます。

- USERNAME : `administrator` のデフォルトのログイン情報を使用します。
- PASSWORD のパスワードを入力します。

新しいパスワードは、強力なパスワードポリシー（大文字と小文字、数字、記号の組み合わせ）に準拠している必要があります。新しいパスワードは誰にも教えないでください。

- パスワードをもう一度入力して、[REPEAT PASSWORD] を確認します。
- [CONFIGURE] をクリックします。

これは、Workload Optimization Manager のユーザーインターフェイスに管理者権限でアクセスするために使用するアカウントです。ユーザーインターフェイス管理者アカウントのログイン情報は安全な場所に保存してください。

注：

最初のログインは、常に管理者アカウントで行います。これは管理ユーザーアカウントです。VM 自体のシェルセッションにログインするために以前に設定した Workload Optimization Manager システム管理者アカウントと、このアカウントを混同しないでください。

6. 管理者としてログインしたら、他のユーザーアカウントを作成し、さまざまなロールを付与できます。ユーザーアカウントとロールの詳細については、Workload Optimization Manager ユーザーガイド [英語] を参照してください。



全般設定タスク

Workload Optimization Manager インスタンスをインストールした後、次の設定タスクを実行します。

- (必須) システムクロックを同期し、タイムサーバーを設定します。
- (任意) LDAP を介してセキュアなアクセスを適用します。
- (任意) 信頼できる証明書を通じてセキュアなアクセスを適用します。
- (任意) プローブのセキュアなアクセスを有効にします。
- (任意) Cluster Manager の証明書を変更します。
- (任意) 組み込みレポートを有効にします。
- (任意) データエクスポートを有効にします。
- (任意) プラットフォームノードの IP アドレスを変更します。
- (任意) プローブコンポーネントを有効または無効にします。

(必須) 時刻の同期

同じネットワークにある他のデバイスと、Workload Optimization Manager インスタンスのクロックを同期させることが重要です。デフォルトでは、Workload Optimization Manager サーバーは、次のいずれかのタイムサーバーと同期するように設定されています。

- 0.centos.pool.ntp.org
- 1.centos.pool.ntp.org
- 2.centos.pool.ntp.org
- 3.centos.pool.ntp.org

これらのサーバーと同期させるには、インストールした Workload Optimization Manager がインターネットにアクセスできる必要があります。インターネットアクセスが制限されている環境の場合は、ネットワーク上のタイムサーバーとの同期を設定する必要があります。

いずれの場合も、Workload Optimization Manager のクロックが正しく同期されていることを確認する必要があります。システムクロックを確認するには、次の手順を実行します。

1. Workload Optimization Manager インスタンスへの SSH ターミナルセッションを開きます。
Workload Optimization Manager のインストール時に設定したシステム管理者でログインします。
 - ユーザー名 : turbo
 - ユーザー名 : [your_private_password]

2. 時刻設定を確認します。

`date` コマンドを実行します。次のような結果が表示されます。

```
Thu Feb 2 14:25:45 UTC 2019
```

時刻を確認するには、`timedatectl` コマンドを実行します。出力は次のようになります。

```
Local time: Fri 2019-12-06 21:09:26 UTC
Universal time: Fri 2019-12-06 21:09:26 UTC
RTC time: Fri 2019-12-06 21:09:27
Time zone: UTC (UTC, +0000)
NTP enabled: yes
NTP synchronized: yes
RTC in local TZ: no
DST active: n/a
```

出力から、NTP が有効になっているかどうか、現在同期されているかどうか、およびその他の時刻同期情報を確認できます。出力が正しく、使用環境がインターネットにアクセスできる場合、システムクロックが同期されていると見なすことができます。出力が正しくない場合、またはネットワーク上のタイムサーバーとの同期を設定する必要がある場合、サーバーインスタンスで `chrony` を設定する必要があります。

Workload Optimization Manager インスタンスで `chrony` を設定するには、次の手順を実行する必要があります。

1. Workload Optimization Manager インスタンスへの SSH ターミナルセッションを開きます。
2. `chrony` 構成ファイルを開きます。

たとえば、次のコマンドを実行します。`sudo vi /etc/chrony.conf`

3. 使用環境で使用するタイムサーバーを指定します。

`chrony` ファイルには、タイムサーバーを設定するための次のステートメントが含まれています。

```
server 0.centos.pool.ntp.org iburst
server 1.centos.pool.ntp.org iburst
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst
```

使用するサーバーのステートメントを入力します。次に、使用しないステートメントを削除するか、コメントアウトします。次のコマンドシンタックスを使用してタイムサーバーを指定します。

```
server My_Time_Server_Name iburst
```

4. ファイルを保存します。
5. `chrony` サービスを再起動します。
コマンドを実行します:`sudo systemctl restart chronyd`
6. 時間が正しいことを確認してください。

`date` コマンドを実行します。次のような結果が表示されます。

```
Fri Dec 6 21:09:26 UTC 2019
```

時刻が同期されていることを確認するには、`timedatectl` コマンドを実行します。出力は次のようになります。

```

Local time: Fri 2019-12-06 21:09:26 UTC
Universal time: Fri 2019-12-06 21:09:26 UTC
RTC time: Fri 2019-12-06 21:09:27
Time zone: UTC (UTC, +0000)
NTP enabled: yes
NTP synchronized: yes
RTC in local TZ: no
DST active: n/a
  
```

時刻を確認するには、`date` の出力を既知の UTC タイムサーバーからの出力と比較します。

出力が正しい場合、システムクロックが同期されていると見なすことができます。

出力が正しくない場合は、サポート担当者に連絡してください。

(重要) MariaDB バージョンの確認

デフォルトの履歴データベースの場合、Workload Optimization Manager は現在、MariaDB バージョン 10.5.12 をサポートしています。このサポートには、Workload Optimization Manager による履歴データベースの使用に関する包括的なテストと品質管理が含まれます。

VM イメージ (OVA または VHD) としてインストールされた Workload Optimization Manager を実行しており、そのイメージのインストールに含まれているデータベースを使用している場合は、バージョン 10.5.12 を使用する必要があります。

このセクションでは、Workload Optimization Manager の VM イメージのインストールで MariaDB のバージョンを確認する方法について説明します。また、更新スクリプトを使用して Workload Optimization Manager をバージョン 3.1.5 以降に更新した場合は、このセクションの手順を使用して MariaDB を更新できます。

重要事項：

MariaDB バージョン 10.5.12 以降を実行する必要があります。

Workload Optimization Manager を初めてインストールした場合、そのインストールには特定のバージョンを実行する MariaDB が含まれており、Workload Optimization Manager のバージョンを更新しても、MariaDB のバージョンは同じままです。MariaDB を明示的に 10.5.12 に更新していない場合は、ここで更新する必要があります。

Workload Optimization Manager は、他のバージョンの MariaDB でも動作できますが、MariaDB バージョン 10.5.12 での動作が完全にテストされています。

Workload Optimization Manager は、カスタムインストールとして展開された MySQL 5.7.x もサポートしています。

VM イメージのインストールの場合、リモートデータベース (VM の外部) を使用するようにインストールを設定できます。リモートの MariaDB インスタンスを使用している場合は、バージョン 10.5.12 を使用することをお勧めします。リモート MySQL の場合、バージョン 5.7.x を使用する必要があります。そのような展開では、データベースのバージョンングを自分で管理する必要があります。

(Workload Optimization Manager VM イメージとして展開されていない) Kubernetes クラスタにインストールし、MariaDB を使用している場合は、バージョン 10.5.12 を使用することをお勧めします。MySQL の場合、バージョン 5.7.x を使用する必要があります。そのような展開では、データベースのバージョンングを自分で管理する必要があります。

MariaDB バージョンの確認

Workload Optimization Manager OVA で実行されている MariaDB のバージョンを確認するには、次の手順を実行します。

1. Workload Optimization Manager インスタンスへの SSH ターミナルセッションを開きます。
Workload Optimization Manager のインストール時に設定したシステム管理者でログインします。

- ユーザー名 : turbo
- ユーザー名 : [your_private_password]

2. MariaDB のバージョンを確認します。

```
mysql -u root --password=vmturbo -e "SHOW VARIABLES LIKE 'version';"
```

出力は次のようになります。

```
+-----+-----+
| Variable_name | Value           |
+-----+-----+
| version       | 10.5.12-MariaDB |
+-----+-----+
```

バージョンが 10.5.12-MariaDB よりも古い場合は、データベースを更新する必要があります。

バージョンが 10.5.12-MariaDB 以降の場合は、以下の更新手順を実行しないでください。

MariaDB の更新

VM イメージとしてインストールされた Workload Optimization Manager を使用していて、そのイメージとともにインストールされたデフォルトの MariaDB を使用している場合は、MariaDB バージョン 10.5.12 を実行する必要があります。

Workload Optimization Manager VM の MariaDB を更新するには、次の手順を実行します。

1. Workload Optimization Manager インスタンスへの SSH ターミナルセッションを開きます。
Workload Optimization Manager のインストール時に設定したシステム管理者でログインします。
 - ユーザー名 : turbo
 - パスワード : [your_private_password]
2. VM が Workload Optimization Manager 更新 ISO イメージにマウントされていることを確認します。

注 :

Workload Optimization Manager の更新が完了すると、ISO イメージが自動的にアンマウントされます。MariaDB の更新を実行するには、バージョン 3.1.5 以降に更新するために使用したのと同じ ISO イメージに Workload Optimization Manager インスタンスをマウントする必要があります。

オフライン更新と ISO イメージのマウントについては、[「オフライン更新」 \(71 ページ\)](#) を参照してください。

3. MariaDB 更新スクリプトを実行します。

スクリプトを実行する前に、Mysql パスワードを確認しておく必要があります。デフォルトでは、このパスワードは vmturbo です。

a. スクリプトディレクトリに移動します。

```
cd /opt/local/bin
```

b. スクリプトを実行可能にします。

```
chmod +x mariadbUpgrade.sh
```

c. データベース更新スクリプトを実行します。

```
sudo ./mariadbUpgrade.sh
```

このスクリプトを実行すると、MariaDB のバージョンが更新され、許可されたパケットのサイズ制限、および innodb のバッファとログのサイズが増えます。スクリプトの出力には、次の内容が含まれている必要があります (Total Memory と buffer pool size は、VM 構成によって異なります)。

```
=====
Update the mariadb configuration
=====
Total Memory: 128773 MB
Changing Innodb buffer pool size to: 9216 MB Changing max allowed packets to: 1G
Changing innodb log file size to: 10G
=====
```

4. 更新された MariaDB バージョンを確認します。

スクリプトが完了すると、バージョン 10.5.12 が実行されている必要があります。バージョンを確認するには、次のコマンドを実行します。

```
mysql -u root --password=vmturbo -e "SHOW VARIABLES LIKE 'version';"
```

出力は次のようになります。

```
+-----+-----+
| Variable_name | Value          |
+-----+-----+
| version       | 10.5.12-MariaDB |
+-----+-----+
```

5. Workload Optimization Manager プラットフォームのポッドをスケールアップします。

データベースを更新するために、スクリプトによってプラットフォームポッドがスケールダウンされます。完了すると、次のプロンプトが表示されます。

```
#####
When confirmed the mariadb has been upgraded and is properly working, run:
kubectl scale deployment --replicas=1 t8c-operator -n turbonomic
#####
```

正しいバージョンの MariaDB が実行されていることを確認したら、プラットフォームをスケールアップします。

```
kubectl scale deployment --replicas=1 t8c-operator -n turbonomic
```

使用可能なディスク容量の増加

VM イメージへの Workload Optimization Manager の標準インストールには、履歴データ用の MariaDB データベースサーバーが含まれています。組み込みレポートを有効にすると、プラットフォームでは TimescaleDB Postgres データベースも使用してレポートデータが管理されます。さまざまな理由から、データベースサービス用のデフォルトのストレージ容量が十分でない場合があります。十分でない場合は、利用可能なストレージ容量を増やす必要があります。

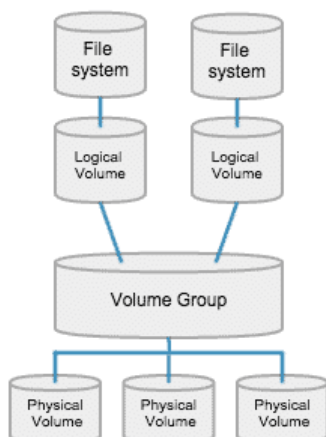
ストレージ容量を増やす一般的な理由は、組み込みレポートの推定ニーズに対応するためです。組み込みレポートのストレージ要件は、使用環境の変化に応じて、または Workload Optimization Manager インストールを設定するターゲット数の増加に応じて、時間の経過とともに変化する可能性があります。組み込みレポートの要件の見積もりについては、[「組み込みレポートのストレージ要件の見積もり」 \(42 ページ\)](#) を参照してください。

実行する手順の概要は次のとおりです。

- VM に新しいディスクを追加する
- SCSI デバイスを再スキャンする
- 新しい LVM パーティションを作成する
- 物理ボリューム (pv) を作成する
- pv を既存のボリュームグループ (vg) に追加する
- 論理ボリューム (lv) を拡張する
- 新しい lv を使用するようにファイルシステムを拡張する
- 組み込みレポートのストレージを増やすために、XFS クォータを増やす
MariaDB のスペースを増やすために、この手順を実行する必要はありません。

Workload Optimization Manager ストレージの論理ボリューム管理

プラットフォームでは、論理ボリューム管理 (LVM) を使用して VM ディスクが管理されます。データベースストレージを増やすには、新しいディスクを VM に追加し、そのディスクを使用して LVM 論理ボリューム (`/dev/turbo/var_lib_mysql`) を拡張する必要があります。この論理ボリュームは、履歴データベースと組み込みレポートデータベースの両方を提供します。



ストレージの増加：手順

データベースで使用可能なストレージ容量を増やすには、次の手順を実行します。

1. VM に新しいディスクを追加します。

VM データセンターの手順を使用して、新しいディスクを VM に追加します。Workload Optimization Manager は、VMware または Hyper-V VM としてインストールされます。新しいディスクを追加する手順については、ハイパーバイザのドキュメントを参照してください。

2. Workload Optimization Manager インスタンスへの SSH ターミナルセッションを開きます。

Workload Optimization Manager のインストール時に設定したシステム管理者でログインします。

- ユーザー名 : turbo
- ユーザー名 : [your_private_password]

3. SCSI デバイスを再スキャンします。

新しいディスクが使用可能なことを確認するには、SCSI デバイスを再スキャンしてから、ブロックデバイスを一覧表示します。デバイスをスキャンするには、次を実行します。

```
echo "- - -" > /sys/class/scsi_disk//0\:0\:0\:0/device/rescan
```

新しいディスクを確認するには、次を実行します。

```
lsblk
```

新しいディスクは、/dev/sdc のような名前が表示されます。新しいディスクが表示されない場合は、次の代替手順を試して強制的に再スキャンします。

- VM 上にある SCSI ホストデバイスの数を確認する

```
ls /sys/class/scsi_host
```

host0、host1、host2... hostn などのデバイスのリストが表示されます。

- 各デバイスをスキャンする

デバイスごとにコマンドを実行します (<hostn> は、host0 または host1 などの番号付きのホストデバイスです)。

```
echo "- - -" > /sys/class/scsi_host/host0/scan
```

- ブロックデバイスを一覧表示する

lsblk を再度実行して、ブロックデバイスを一覧表示します。

4. 新しい LVM パーティションを作成します。

新しいディスクの名前が /dev/sdc1 である場合、次のコマンドを実行します。

```
cfdisk /dev/sdc1
```

以下の操作を実行します。

- new
- primary
- confirm size
- change type to 8E
- write
- quit

5. 物理ボリューム (pv) を作成します。

新しいディスクの名前が `/dev/sdc1` である場合、次のコマンドを実行します。

```
pvcreeate /dev/sdc1
```

6. 新しい pv を既存のボリュームグループに追加します。

新しいディスクの名前が `/dev/sdc1` である場合、次のコマンドを実行します。

```
vgextend /dev/turbo /dev/sdc1
```

7. 論理ボリューム (lv) を拡張して、新しい pv の空き領域を使用します。

まず、使用可能な物理エクステント (PE) をリストします。コマンドを実行します：

```
vgdisplay
```

次のような結果が表示されます。

```
Free PE / Size          128000 / 500.00 GiB
```

この例では、128000 が lv を拡張する量です。この例では、次のコマンドを実行します。

```
lvextend -l +128000 /dev/turbo/var_lib_mysql
```

8. XFS ファイルシステムを拡張して、現在の lv 領域をすべて使用します。

XFS を拡張する前に、空きディスク容量を表示し、その容量を記録します。使用可能な領域が増えたことを確認するには、XFS 拡張後の空き領域とこの値を比較します。コマンドを実行します：

```
df -h
```

次に、XFS 容量を拡張します。

```
xfs_growfs /dev/turbo/var_lib_mysql
```

次に、更新された空きディスク容量を一覧表示して、元の容量と比較します。

```
df -h
```

9. 組み込みレポートの容量を増やす場合は、TimescaleDB の XFS クォータを拡張します。

MariaDB のスペースを増やすために、この手順を実行する必要はありません。

Timescale DB の容量を増やすには、そのプロセスのクォータを必要な量だけ増やす必要があります。クォータ名は `Postgresql` です。

たとえば、400 GB のボリュームを追加し、現在の `Postgresql` クォータが 400 GB である場合、クォータを 800 GB に増やすことができます。この例では、次のコマンドを実行します。

```
xfs_quota -x -c 'limit -p bhard=800g Postgresql' /var/lib/dbs
```

`/var/lib/dbs` に設定されている現在のクォータを表示するには、次のコマンドを実行します。

```
xfs_quota -xc 'report -pbih' /var/lib/dbs
```


(任意) LDAP を介したセキュアなアクセスの適用

会社のポリシーでセキュアなアクセスが必要な場合は、LDAP サービスで証明書を使用して、ユーザーのセキュアなアクセスを設定できます。たとえば、Active Directory (AD) アカウントを設定して、ユーザーまたはユーザーグループの外部認証を管理できます。AD を有効にするためのユーザーインターフェイスには、証明書ベースのセキュリティを適用する [Secure] オプションが含まれています。詳細については、Workload Optimization Manager ユーザーガイド [英語] の「Managing User Accounts」を参照してください。

LDAP サービスが認証局 (CA) を使用している場合、その CA によって署名された証明書で、この機能がそのままサポートされる必要がありますが、AD 接続を設定するときに、[Secure] オプションをオンにするだけです。

LDAP サービスが自己署名証明書を使用している場合は、その証明書を Workload Optimization Manager 認証ポッドにインストールする必要があります。実行する手順は次のとおりです。

- LDAP サーバーから証明書を取得する
- 証明書をプラットフォームのトラストストアにインポートする
- Workload Optimization Manager プラットフォームの認証ポッドに証明書を追加する
- Workload Optimization Manager プラットフォームの Operator チャートでトラストストアを有効にする

自己署名証明書のインストール

セキュアなアクセスを設定するには、次の手順を実行します。

1. Workload Optimization Manager インスタンスへの SSH ターミナルセッションを開きます。

Workload Optimization Manager のインストール時に設定したシステム管理者でログインします。

- ユーザー名 : turbo
- パスワード : [your_private_password]

2. LDAP サーバー証明書を Workload Optimization Manager インスタンスにダウンロードします。

LDAP 管理者から証明書を取得し、Workload Optimization Manager プラットフォームにダウンロードします。たとえば、/tmp/ldapserver.crt ファイルにダウンロードできます。

3. .crt ファイルを Workload Optimization Manager トラストストアにインポートします。

これで、Workload Optimization Manager プラットフォームの cacerts ファイルが変更されます。

注 :

証明書を Workload Optimization Manager トラストストアにインポートするには、keytool ユーティリティを使用する必要があります。このユーティリティをインストールするには、次のコマンドを実行します。

```
sudo yum install java-1.8.0-openjdk
/usr/bin/keytool にユーティリティがインストールされます。
```

LDAP 証明書のエイリアスがすでに存在する場合は、その証明書を削除します。たとえば、エイリアスが ldapcert1 である場合、次のコマンドを実行します。

```
keytool -delete -alias ldapcert1 -keystore cacerts -storepass changeit
次のコマンドを使用して、新しい証明書をトラストストアにインポートします。
```

```
keytool -import -alias ldapcert1 -file /tmp/ldapserver.crt -keystore cacerts \
  -deststoretype jks -storepass changeit -noprompt
```

4. cacerts ファイルから auth シークレットを作成します。

```
base64 cacerts > auth-secrets.yaml
```

5. 編集のためにシークレットファイルを開きます。

```
vi auth-secrets.yaml
```

6. ファイルを編集して、有効な yaml ファイルにします。

a. 証明書のすべての行を 4 つのスペースでインデントします。

ファイルの作成時に、証明書の内容を連結します。最初の手順では、証明書を 4 つのスペースでインデントします。たとえば、vi エディタで次のコマンドを実行します。

```
:%s/^/    /g
```

b. シークレットファイルにデータフィールドを追加します。

ファイルの先頭に次のテキストを追加します。

```
apiVersion: v1
kind: Secret
metadata:
  name: auth-secret
data:
  cacerts: |
```

c. 変更を保存します。

完成したファイルは次のようになります。

```
apiVersion: v1
kind: Secret
metadata:
  name: auth-secret
data:
  cacerts: |
    /u3+7QAAAAIAAAAABAAAAAgAFY2VydDEAAAF5H2lEigAFWC4lMDkAAAYQMIIGDDCCBPsgAwIBAgIT
    HAAAAARHIFJdLbG90sAAAAAABETANBkgkqhkiG9w0BAQUFADBcMRMwEQYKZImiZPyLQGBGRYDY29t
    MRcwFQYKZImiZPyLQGBGRYHdm10dXJibzEUMBIGCgMjJomT8ixkARKWBGnVcnAxFjAUBGNVbAMT
    DWNvcnAtREVMTDEtQ0EwHhcNMjEwNDA4MDM0OTEyWjcNMjEwNDA4MDM0OTEyWjAhMR8wHQYDVQQD
    ExZkZWxsMS5jb3JwLnZtdHVyYm8uY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
    sCXuh2MtrFERyU/aKgdbgyjLezNuWf6nmZveZUhDaJDpfLHJlzhwfyYRTGfSSusVo4polJS4WqPZ
    T3Zk8f2IaX04RpfpQErq5N3uY/BxFkATWLMdiQuSd0Di798k2diYXAxVzMMfmIkBBYJta9oztum
    uXyh/42dXOGznQ5fFuxosgAksZ6CnXGDKrTB1b0bHpST1z1Pdg+fJ+f9Tq7IffOYdVbuedFTwsik
    Z0JgDCIRmms0JphiHdBqJ6ZLdbSeEzBIbboiQs81pAELw7V0ZUZUfKV6y8+zMTACGwpVPJSFv7LX
    RlW1TWcqhXVAOmroe2WcU8KJE6XZTbXp7z7dzWIDAQABo4IDADCCAvwwLwYJKwYBBAGCNxQCBCIE
    IABEAG8AbQBhAGkAbgBDAG8AbgB0AHIAbwBsAGwAZQByMB0GA1UdJQQWMBQGCCsGAQUFBwMCBggr
    BgEFBQcDATAOBgNVHQ8BAf8EBAMCBaAweAYJKoZIhvcNAQkPBGswaTAOBggqhkiG9w0DAGICAIAw
    DgYIKoZIhvcNAwQCAgCAMAsGCWCGSsAF1AwQBKjALBglghkgBZQMEAS0wCwYJYIzIAWUDBAECMAAG
    CWCGSsAF1AwQBTAHBgUrDgMCBzAKBggqhkiG9w0DBzBCBgNVHREEOzA5oB8GCSsGAQQBggjczAAAS
    BBDswj1Hut/nQZ0uK2aUg1GbghZkZWxsMS5jb3JwLnZtdHVyYm8uY29tMB0GA1UdDgQWBRR6M7Hb
    BiiRrpjIXQ3PXXScB8LkmRDAfBgNVHSMEGDAWgBRjs9l3el7SuKUDMlrHHRhBkEngaDCB0QYDVR0f
    BIHJMIHGMIHDoIAAoIG9hoG6bGRhcDovLy9DTj1jb3JwLURFTExwLUNBLENOPWRlbGwxLENOPUNE
    UCxDtj1QdWjsaWm1mJBLZkXlMjBTZJ2aWn1cyxDtj1TZXJ2aWn1cyxDtj1Db25maWd1cmF0aW9u
    LERDPWNvcnAsREM9dm10dXJibzEUMBIGCgMjJomT8ixkARKWBGnVcnAxFjAUBGNVbAMT
    P29iamVjdENsYXNzPWNSTERpc3RyaWJldGlvb1BvaW50MIHBBggrBgEFBQcBAQSBujCBtZCBtAYI
    KwYBBQUHMAKGGadsZGFwOi8vL0NOPWNvcnAtREVMTDEtQ0EsQ049QUBLENOPVB1YmXpYyUyMETl
    eSUyMFN1cnZpY2VzLENOPVN1cnZpY2VzLENOPUNvbmZpZ3VyYXRpb24sREM9Y29ycCxEQz12bXRl
    cmJvLERDPWNvbT9jQUJlcncRZm1jYXRlP2Jhc2U/b2JqZWNoQ2xhc3M9Y2VydGlmawNhdGlvbkF1
    dGhvcml0eTANBgkqhkiG9w0BAQUFAAOCAQEADP6OYLONkZ2j6gaBdfdoIJtvn1g1qXTsRrtFuUcF
    C9mUxL0G5Tudr0VlyEnLH2wtj10CGsIi54+aPGYiElXijThEelWTHa02hklRLdNrM8KxUp3tUNb/
```

```
cP4d+EYt297wVWgxp19MStiND8+7M2+65daoEu5IOltq4lC7YlCSXay19N5HdiGBHV5L07PTZ261
qDzShSb0ZwtG7++5VvkqveVEIfs3hUYdaItz0Zu6sym90aUcvn5wohV1GPPqGDvVCg5Kf50hsZfmy
ltNlaqiiqLMnYVma93CkpFFjoP9gmGFJky0yTfh6G8HuqbI7guddDsUqMQTT3uv3EBwSYeImOya7
Zye5C4NnsAfnx8kOwXdsVERC
```

7. このシークレットファイルをプラットフォーム環境に適用します。

```
kubectl apply -f auth-secrets.yaml
```

8. プラットフォームの Operator チャートを更新して、シークレットファイルで作成した cacerts 証明書を使用します。

- a. 編集のためにチャートファイルを開きます。

/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml ファイルを開きます。

- b. コンポーネントオプションの認証仕様として認証シークレットを追加します。

チャートファイルで、spec: セクションを見つけます。そのセクション内で、auth: サブセクションを見つけます。

これは、spec: の global: の後にある 2 番目のサブセクションである必要があります。auth: サブセクションがない場合は、spec: に追加できます。

- c. 証明書シークレットをファイルに追加します。

auth: サブセクション内の javaComponentOptions: ステートメントにシークレットのパスを追加します。パスを -D オプションとして追加します。auth: サブセクションは次のようになります。auth は 2 つのスペースでインデントされ、javaComponentOptions は 4 つのスペースでインデントされます。

```
# Pass in the JAVA_OPTS to the auth POD to set up additional options such as
# a trustStore for AD Certificate(s) for LDAPS (Secure LDAP)
auth:
  javaComponentOptions: "-Djavax.net.ssl.trustStore=/home/turbonomic/data/helper_dir/cacerts"
```

- d. Operator チャートの変更を Workload Optimization Manager プラットフォームに適用します。

次のコマンドを実行します。

```
kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

承認コンポーネントが再起動され、新しい設定を使用できるようになります。

(任意) 信頼できる証明書を介したセキュアなアクセスの適用

会社のポリシーにより、信頼できる証明書を介した SSL 接続が必要な場合、Workload Optimization Manager により既知の認証局からの信頼できる証明書をインストールできます。

証明書の要求

最初の手順では、証明書を取得します。次の手順では、証明書要求を生成する方法について説明します。

1. シェルターミナルセッションを開きます。

Workload Optimization Manager インスタンスの SSH ターミナルセッションを開きます。turbo としてログインし、前述のインストール手順で管理アカウント用に作成したパスワードを使用します。詳細については、インストール手順、「[Workload Optimization Manager システム管理者アカウントのセットアップ](#)」(9 ページ) を参照してください。

2. 秘密キーファイルを保存するディレクトリに変更します。

シェルセッションが Workload Optimization Manager インスタンス上にある場合は、/opt/turbonomic ディレクトリを使用する必要があります。

```
cd /opt/turbonomic
```

3. 秘密キーファイルを作成して保存します。

秘密キーファイルを作成するコマンドを実行します。

この例では、秘密キーファイルの名前は myPrivate.key です。

```
openssl genrsa -out myPrivate.key 2048
```

このファイルは後で必要になります。Workload Optimization Manager インスタンスでセッションを行っている場合は、ファイルをローカルマシンにコピーできます。

4. 証明書署名要求 (CSR) を生成する情報を含むファイルを作成します。

```
vi certsignreq.cfg
```

5. 要求データを certsignreq.cfg ファイルに追加します。

このファイルに、次のコードを挿入します。山カッコでマークされたフィールド (<city> など) は、示された値を入力します。たとえば、国、都市、会社など。

```
[req]
ts = 2048
prompt = no
default_md = sha256
req_extensions = req_ext
distinguished_name = dn

[dn]
C=<country, 2 letter code>
L=<city>
O=<company>
OU=<organizational unit name>
CN=<FQDN>
emailAddress=<email address>

[req_ext]
subjectAltName = @alt_names

[alt_names]
DNS.1 = <FQDN>
DNS.2 = <server's short name>
DNS.3 = <server's IP address>
```

注:

[CN] フィールドに、Workload Optimization Manager インスタンスの完全修飾ドメイン名を指定します。

代理ユーザー名は、Workload Optimization Manager インスタンスにアクセスする別の方法です。[alt_names] セクションでは、[DNS.1] フィールドの値は必須です。[DNS.1] に、Workload Optimization Manager インスタンスの完全修飾ドメイン名を指定します。[DNS.2] および [DNS.3] の値は任意です。必要に応じて、複数の DNS.<n> フィールドを追加できます。

次に例を示します。

```

root@turbonomic:/etc/pki/tls/private:
ts = 2048
prompt = no
default_md = sha256
req_extensions = req_ext
distinguished_name = dn

[dn]
C=US
ST=New York
L=White Plains
O=Turbonomic
OU=Educational Services
CN=demo.turbonomic.com
emailAddress= <first.lastname>@turbonomic.com

[req_ext]
subjectAltName = @alt_names

[alt_names]
DNS.1 = demo.turbonomic.com
DNS.2 = demo
DNS.3 = my.ip.add.ress
  
```

- 書き込みし、ファイルを終了します。

Esc を押して :wq! と入力し、Enter を押します。

- 証明書要求ファイルを生成します。

この例では、ファイルに myRequest.csr という名前を付けます。

コマンドを実行します:

```

openssl req -new -sha256 -nodes -out myRequest.csr -key \
  myPrivate.key -config certsignreq.cfg
  
```

- 生成された要求ファイルを認証局に送信します。

Workload Optimization Manager インスタンスでファイルを生成した場合は、そのファイルをローカルマシンに転送する必要があります。リモートマシンの証明書要求ファイルへのパスは、/opt/turbonomic/myRequest.csr です。

証明機関は証明書を作成するためにこのファイルを使用します。

認証局から DER と Base64 の間のエンコーディングを選択できる場合は、**Base64** を選択してください。

- 証明書を受信したら、ディスクに保存します。

Base 64 でエンコードされた証明書でない場合は、DER から Base 64 に変換する必要があります。証明書の名前が MyCertificate.crt である場合、次のコマンドを実行します。

```

openssl x509 -inform der -in MyCertificate.der -out MyCertificate.crt
  
```

Workload Optimization Manager への署名証明書のインストール

署名証明書を取得したら、Workload Optimization Manager インスタンスにインストールできます。署名証明書を要求するときに取得した秘密キーと証明書ファイルを使用します。

- myPrivate.key
- MyCertificate.crt

署名証明書をインストールするには、次の手順を実行します。

1. Workload Optimization Manager インスタンスの SSH ターミナルセッションを開きます。
2. キーと証明書データを Workload Optimization Manager の chart.yaml ファイルに追加します。

/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml ファイルを開きます。

global パラメータのセクションを見つけます。global パラメータの下で、ingress:secrets セクションを作成し、certificate、key、および name のエントリを作成します。

global パラメータは次のようになります。

```
global:
  ingress:
    secrets:
      - certificate: |
          -----BEGIN CERTIFICATE-----
          SAMPLE PUBLIC KEY
          -----END CERTIFICATE-----
        key: |
          -----BEGIN RSA PRIVATE KEY-----
          SAMPLE PRIVATE KEY
          -----END RSA PRIVATE KEY-----
        name: nginx-ingressgateway-certs
```

追加したフィールドの場合：

- [certificate:] このフィールドには、MyCertificate.crt ファイルの内容が保持されます。ファイルを開いて内容をコピーし、ここに貼り付けます。
 - [key:] このフィールドには、myPrivate.key ファイルの内容が保持されます。ファイルを開いて内容をコピーし、ここに貼り付けます。
 - [name:] このフィールドは必須で、名前は nginx-ingressgateway-certs である必要があります。
3. CR ファイルに加えた変更を適用します。

コマンドを実行します：

```
kubectl apply -f \
  kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

4. nginx ポッドを再起動します。

HTTPS アクセスに証明書を要求するには、nginx ポッドを再起動する必要があります。

- a. ポッドのフルネームを取得します。

kubectl get pods -n turbonomic コマンドを実行します。出力で、nginx のエントリを探します。次のようなエントリが見つかります。

```
nginx-5b775f498-sm2mm          1/1      Running    0
```

- b. ポッドを再起動します。

次のコマンドを実行します。<UID> は、ポッドインスタンスに対して生成された ID です。

```
kubectl delete pod nginx-<UID>
```

nginx ポッドが再起動します。再起動後、Workload Optimization Manager は HTTPS アクセス用の証明書を要求します。

(任意) プローブのセキュアなアクセスの有効化

ターゲットが信頼できる証明書を介した SSL 接続を必要とする場合、Workload Optimization Manager を使用すると、関連するプローブコンポーネントに信頼できる証明書をインストールできます。

Workload Optimization Manager プラットフォームには、ターゲットに接続してターゲットのデータを検出するために使用する多くのプローブコンポーネントが含まれています。この手順では、1 つのコンポーネントである Dynatrace プローブのセットアップを想定しています。他のプローブにも同じ手順を使用し、各プローブに異なる Kubernetes シークレット名を指定できます。

注：

複数のプローブの SSL 通信を設定するには、プローブごとに固有の証明書を取得する必要があります。

プローブコンポーネントに証明書をインストールするには、特定のプローブの Kubernetes シークレット名を知っている必要があります。次の表に、設定できるプローブとプローブのシークレット名を示します。

プローブ：	K8s シークレット名：
mediation-awsbilling mediation-aws mediation-awscost mediation-awslambda	aws
mediation-appinsights	appinsights
mediation-newrelic	newrelic
mediation-azuresp mediation-azure mediation-azurevolumes mediation-azurecost	azure
mediation-azureea	azureea
mediation-dynatrace	dynatrace

プローブコンポーネントへの署名証明書のインストール

この手順は、有効な .crt ファイルがすでにあることを前提としています。署名証明書を取得する手順については、[「証明書の要求」\(27 ページ\)](#) を参照してください。ただし、Workload Optimization Manager インスタンスで証明書を生成する代わりに、ローカルマシンで生成する必要があります。

署名証明書を取得したら、プローブインスタンスにインストールできます。署名証明書を要求するときに取得した証明書ファイルを使用します。

```
MyCertificate.crt
```

プローブで署名証明書をインストールするには、次の手順を実行します。

1. 証明書をローカルマシンから Workload Optimization Manager インスタンスにコピーします。
SCP を使用して、MyCertificate.crt をローカルマシンからインスタンスの /tmp ディレクトリにコピーします。完了したら、kubectl を使用して証明書をプローブコンポーネントに転送できます。
2. turbo ユーザーアカウントを使用して、Workload Optimization Manager インスタンスの SSH ターミナルセッションを開きます。
3. 証明書ファイルをプローブコンポーネントにコピーします。

最初に、プローブを実行するポッドの ID を取得します。ID を取得するには、次のコマンドを実行します。

```
kubectl get pod
```

Workload Optimization Manager プラットフォームで実行されているポッドとポッドの ID が一覧表示されます。設定するポッドの ID を記録します。

証明書をコピーするには、次のコマンドを実行します。<Probe-Pod-Id> は記録した ID です。

```
kubectl cp /tmp/MyCertificate.crt <Probe-Pod-Id>:/tmp
```

4. キーストアの cacerts ファイルをコンポーネントの /tmp ディレクトリにコピーします。
次のコマンドを使用して、ポッドで bash セッションを開き、ファイルをコピーします。

- `kubectl exec -ti <Probe-Pod-Id> bash`
- `cp /etc/pki/ca-trust/extracted/java/cacerts /tmp`

5. 証明書をポッドのキーストアにインポートします。

この手順の一環として、証明書が Base64 形式であることを確認します。プローブコンポーネントの bash セッションにいる間に、次のコマンドを実行します。

- `chmod 775 /tmp/cacerts`
- `keytool -import -alias MyCertificate.crt -file \
/tmp/ca.crt -keystore /tmp/cacerts -deststoretype jks \
-storepass changeit -no-prompt`

MyCertificate.crt は、取得した証明書の名前です。

- `base64 /tmp/cacerts > base64.txt`

6. 証明書データを含む yaml ファイルを作成します。

プローブの K8s シークレット名を使用して、次の命名規則で yaml ファイルを作成します。

```
/tmp/<Secret_Name>-secrets.yaml
```

たとえば、Dynatrace プローブの SSL を有効にする場合、シークレット名が `dynatrace` である次のような yaml ファイルを作成します。

```
/tmp/dynatrace-secrets.yaml
```

- a. プローブコンポーネントの bash セッションにいる間に、vi エディタセッションで yaml ファイルを開きます。

```
vi /tmp/<Secret_Name>-secrets.yaml
```

- b. 次の内容をファイルに追加します。

```
apiVersion: v1
kind: Secret
metadata:
  name: dynatrace
data:
  cacerts: |
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```


cacerts セクションで、生成した base64.txt データで xxx 文字を置き換えます。

- c. base64 データを yaml 形式に揃えます。

Esc を押して、内容をファイルに追加するために使用していた編集モードを終了します。次に : と入力して、コマンドモードを開始します。コマンドの場合、次のように入力します。空白トークンは 4 つの余白文字です。

```
7,$s/^/ /
```

Return を押してコマンドを実行します。次に、vi エディタを保存して終了します。

7. プローブコンポーネントでセッションを終了します。

これで、プローブコンポーネントに、そのプローブからの通信用の証明書を設定する yaml ファイルが作成されました。プローブのセッションを終了し、yaml を Workload Optimization Manager プラットフォームに適用する必要があります。

コンポーネントセッションを終了して、Workload Optimization Manager VM のセッションに戻るには、シェルで exit と入力します。

8. yaml ファイルを Workload Optimization Manager プラットフォームに適用します。

yaml ファイルをプラットフォームに適用するには、ファイルをプローブコンポーネントからプラットフォームにコピーして適用します。

プローブコンポーネントからファイルをコピーするには、ポッド ID を知っている必要があります。ID を取得するには、kubectl get pod コマンドを実行し、設定したポッドの ID を記録します。次のコマンドを実行します。<Probe-Pod-Id> は記録した ID です。

- `kubectl cp <Probe-Pod-Id>:/tmp/<Secret_Name>-secrets.yaml \`
`/tmp/<Secret_Name>-secrets.yaml`
- `kubectl apply -f /tmp/<Secret_Name>-secrets.yaml`

9. SSL 証明書を使用して設定するプローブごとに、chart_v1alpha1_cl_cr.yaml ファイルにエントリを追加します。

- a. Workload Optimization Manager プラットフォームで実行されているシェルセッションにおいて、テキストエディタで次のファイルを開きます。

```
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

- b. 設定しているプローブのエントリをファイルで検索します。前述の表にリストされているプローブ名を使用します。たとえば、Dynatrace プローブを設定している場合は、mediation-dynatrace のエントリを見つけます。
- c. プローブエントリの下に、javaComponentOptions に関する次のエントリを追加します。

```
javaComponentOptions: -Djavax.net.ssl.trustStore=/etc/targets/cacerts -Dorg.eclipse.jetty.websocket.jsr356.ssl-trust-all=true -DLog4jContextSelector=com.vmturbo.mediation.common.context.ThreadGroupPrefixContextSelector
```

たとえば、Dynatrace プローブを設定している場合、エントリは次のようになります。

```
mediation-dynatrace:
  javaComponentOptions: -Djavax.net.ssl.trustStore=/etc/targets/cacerts -Dorg.eclipse.jetty.websocket.jsr356.ssl-trust-all=true -DLog4jContextSelector=com.vmturbo.mediation.common.context.ThreadGroupPrefixContextSelector
  resources:
    limits:
      memory: 2Gi
```

- d. chart_v1alpha1_cl_cr.yaml ファイルを保存して終了します。
- e. 変更したファイルを Workload Optimization Manager プラットフォームに適用します。

コマンドを実行します:

```
kubectl apply -f /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

(任意) Cluster Manager の証明書の変更

ファイアウォールの内側にインストールする場合、clustermgr コンポーネントから診断をアップロードするには、その証明書を変更する必要があります。

clustermgr の証明書を変更する手順の前提として、Cluster Manager に追加する証明書がすでに生成されている必要があります。

1. Workload Optimization Manager インスタンスの SSH ターミナルセッションを開きます。

Workload Optimization Manager のインストール時に設定したシステム管理者でログインします。

- ユーザー名:

```
turbo
```

- Password:

```
[your_private_password]
```

2. clustermgr ポッドのフルネームを取得します。

コマンドを実行します:

```
kubectl get pods -n turbonomic | grep clustermgr
```

次のような結果が表示されます。

```
clustermgr-5f487f58f-tf84b    0/1    Running    52    2d4h
```

この例では、clustermgr-5f487f58f-tf84b がポッドのフルネームであり、5f487f58f-tf84b が POD_ID です。

3. ポッドの現在の ca-bundle.crt ファイルのコピーを /tmp に保存します。

次のコマンドを実行します。<POD_ID> は、ポッドのフルネームから取得した ID です。

```
kubectl cp \
clustermgr-<POD_ID>:etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem \
/tmp/ca-bundle.crt
```

4. バンドルに証明書を追加します。

証明書ごとにこのコマンドを繰り返します。<MY_CERT> はユーザーの証明書ファイルです。

```
cat <MY_CERT> >> /tmp/ca-bundle.crt
```

5. 変更された証明書の Kubernetes シークレットを作成します。

```
kubectl create secret generic clustermgr-secret --from-file=/tmp/ca-bundle.crt
```

6. 編集のために cr.yaml ファイルを開きます。

次に例を示します。

```
vi /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

7. このシークレットを使用するように cr.yaml ファイルを変更します。

ファイルに以下の内容を追加します。

```

clustermgr:
  env:
    - name: component_type
      value: clustermgr
    - name: instance_id
      valueFrom:
        fieldRef:
          fieldPath: metadata.name
    - name: instance_ip
      valueFrom:
        fieldRef:
          fieldPath: status.podIP
    - name: serverHttpPort
      value: "8080"
    - name: kafkaServers
      value: kafka:9092
    - name: kafkaNamespace
      valueFrom:
        fieldRef:
          apiVersion: v1
          fieldPath: metadata.namespace
    - name: CURL_CA_BUNDLE
      value: /home/turbonomic/data/ca-bundle.crt
  
```

8. 変更を保存し、cr.yaml ファイルを適用します。

```

kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
  
```

grep ^clustermgr でログを確認すると、診断を送信するたびに適切な curl コマンドが実行されていることがわかります。

次のコマンドを使用して、Cluster Manager ポッドの .crt ファイルを確認することもできます。<POD_ID> は、ポッドのフルネームから取得した ID です。

```

kubectl exec -it clustermgr-<POD_ID> bash
vi /home/turbonomic/data/ca-bundle.crt
  
```

(任意) 組み込みレポートの有効化

組み込みレポートは、Workload Optimization Manager プラットフォームの一部として独自のコンポーネントで実行されます。このアーキテクチャにより、パフォーマンスが向上し、ストレージ要件が軽減されます。組み込みレポートを有効にすると、複数の環境の詳細をチャート化する一連の構築済みダッシュボードに移動できます。ダッシュボードとチャートは、Grafana の可観測性プラットフォームによって強化されています。Grafana を使用すると、既存のダッシュボードを簡単にナビゲートし、コーディングせずに独自のチャートやダッシュボードを作成できます。

組み込みレポートを有効にするために使用する方法は、次のように、Workload Optimization Manager インスタンスのバージョンステータスによって異なります。

- [「スクリプトインターフェイス」 \(36 ページ\)](#)

Workload Optimization Manager バージョン 3.0.0 以降が新しい VM イメージ (OVA または VHD) としてインストールされている場合 ([「仮想マシンイメージへのインストール」 \(8 ページ\)](#) を参照)、`enable_reporting.py` スクリプトを実行して組み込みレポートを設定できます。

- [「Workload Optimization Manager の cr.yaml ファイルの編集」 \(37 ページ\)](#)

Workload Optimization Manager のインストールに VM イメージを使用せずに、Workload Optimization Manager を Kubernetes クラスタとしてインストールした場合、

Workload Optimization Manager のインストール用に `charts_v1alpha1_x1_cr.yaml` ファイルを手動で編集します。

スクリプトインターフェイス

Workload Optimization Manager をバージョン 3.0.0 以降の VM イメージとしてインストールした場合、組み込みレポートを有効にするスクリプトは、次の場所にインストール済みです。

```
/opt/local/bin/enable_reporting.py
```

このスクリプトを実行するには、次の手順を実行します。

1. Workload Optimization Manager インスタンスへの SSH ターミナルセッションを開きます。
Workload Optimization Manager のインストール時に設定したシステム管理者でログインします。
 - ユーザー名 : `turbo`
 - ユーザー名 : `[your_private_password]`
2. スクリプトディレクトリに移動します。

```
cd /opt/local/bin
```

3. 次のスクリプトを実行します。

```
./enable_reporting.py
```

次の 2 つのパスワードの入力を求められます。

- Grafana 管理者パスワード。

このパスワードにより、外部 URL から、およびデータを Grafana にフィードするエクストラクタ コンポーネントから Grafana にアクセスできます。

特殊文字は使用しないでください。

重要事項 :

Grafana 管理者パスワードを変更する必要があるのは、このときだけです。

この手順が完了した後で Grafana 管理者パスワードを変更すると、組み込みレポートコンポーネントはプラットフォーム内の他のコンポーネントと正しく通信できなくなります。このパスワードを後で変更した場合は、サポート担当者に連絡してください。

- Grafana データベースパスワード。

このパスワードにより、Grafana とレポートデータを格納する Postgres データベース間の通信が可能になります。

パスワードを指定すると、スクリプトによって次のような確認メッセージが表示されます。

```
Successfully applied new changes to /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml.
```

```
Backup written to /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml.bak
```

これは、Workload Optimization Manager の設定が正常に更新されたことを示しています。次に、変更された設定が適用されて、組み込みレポート機能が有効になります。次のようなメッセージが表示されます。

```
Applying CR file /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
Warning: kubectl apply should be used on resource created by either kubectl create --save-config or kubectl apply
xl.charts.helm.k8s.io/xl-release configured
Waiting for changes to take effect...
Restarting api pod to apply configuration changes. pod "api-65cf47986f-jxszd" deleted
Changes have been successfully applied. Embedded reporting is now enabled.
```

4. インストールを確認します。

コマンドを実行します:

```
./enable_reporting.py --validate
```

組み込みレポートが正常に有効になっている場合、スクリプトの出力は次のようになります。

```
No obvious embedded reporting installation errors detected.
```

Workload Optimization Manager の cr.yaml ファイルの編集

ここでは、VM イメージのインストール用の charts_v1alpha1_xl_cr.yaml を見つけて編集する方法について説明します。Kubernetes ノードクラスタにインストールした場合、このファイルは別の場所にある可能性があります。

組み込みレポートを有効にするには、次の手順を実行します。

- 組み込みレポートを実装するプロセスを有効にします。
- API ポッドを更新して、新しい検索機能とデータインジェスト機能を有効にします。
- インストールを再確認します。
- PDF レポートと電子メールサブスクリプションを有効にします (任意)。

Grafana Exporter、TimescaleDB、およびデータ抽出プロセスを有効にする必要があります。有効にするには、charts_v1alpha1_xl_cr.yaml ファイルを編集します。

1. Workload Optimization Manager インスタンスへの SSH ターミナルセッションを開きます。

Workload Optimization Manager のインストール時に設定したシステム管理者でログインします。

- ユーザー名: turbo
- ユーザー名: [your_private_password]

2. 次のファイルをテキストエディタで開きます。

```
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

3. TimescaleDB データベースへの外部アクセス用に、Workload Optimization Manager インスタンスの IP アドレスを指定します。ファイルの global: セクションで、次の行を追加します。<Platform_IP> はインスタンスの IP アドレスです。

```
global:
  externalTimescaleDBIP: <Platform_IP>
```

4. Grafana プロセスを有効にします。

crds/charts_v1alpha1_xl_cr で grafana: セクションを見つけます。yaml ファイルを開き、enabled: true の行のコメントを外します。

5. データベースタイプとして Postgres を有効にします。

Postgres を有効にすると、組み込みレポートの履歴データの永続ストレージが設定されます。grafana: セクションで、grafana.ini: database: のサブセクションを見つけ、type: postgres の行のコメントを外します。

これまでに行った変更は、次のようになります。

```
global:
  externalTimescaleDBIP: <Platform_IP>
  ...
grafana:
  enabled: true
  adminPassword: admin
  grafana.ini:
    database:
      type: postgres
  ...
```

6. 管理者パスワードとデータベースパスワードを変更します。

パスワードはデフォルト値のままではなく、変更することをお勧めします。

重要事項：

パスワードには英数字のみ使用できます。

パスワードにより、さまざまな組み込みレポートコンポーネント間の通信が可能になります。一部のコンポーネントは英数字のみを受け入れます。特殊文字を使用すると、コンポーネントは通信できません。さらに、パスワードを修正する手順を実行するには、サポートエンジニアの支援が必要です。

パスワードを設定するには、次の手順を実行します。

- Grafana 管理者パスワードを設定します。

このパスワードにより、外部 URL から、およびデータを Grafana にフィードするエクストラクタ コンポーネントから Grafana にアクセスできます。grafana: セクションで、adminPassword の値を変更します。

特殊文字は使用しないでください。

パスワードが MyNewGrafanaPassword である場合、adminPassword:

MyNewGrafanaPassword を設定します。

重要事項：

Grafana 管理者パスワードを変更する必要があるのは、このときだけです。

この手順が完了した後で Grafana 管理者パスワードを変更すると、組み込みレポートコンポーネントはプラットフォーム内の他のコンポーネントと正しく通信できなくなります。このパスワードを後で変更した場合は、サポート担当者に連絡してください。

- Grafana データベースパスワードを設定します。

このパスワードにより、Grafana とレポートデータを格納する Postgres データベース間の通信が可能になります。grafana: セクションで、grafana.ini: database: password: のサブセクションを見つけ、パスワード値を変更します。

7. 3 つの組み込みレポートプロセスを有効にします。

追加した properties: セクションのすぐ後で、同じレベルで次のエントリを追加してレポートプロセスを有効にします。

```
reporting:
  enabled: true
timescaledb:
  enabled: true
extractor:
  enabled: true
```

これらのエントリを grafana: セクションおよび properties: セクションのインデントに合わせる必要があります。行った変更は次のようになります。

```
global:
  externalTimescaleDBIP: <Platform_IP>
  ...

grafana:
  enabled: true
  adminPassword: MyNewGrafanaPassword
  grafana.ini:
    database:
      type: postgres
      password: MyNewDatabasePassword

properties:
  extractor:
    grafanaAdminPassword: MyNewGrafanaPassword

reporting:
  enabled: true
timescaledb:
  enabled: true
extractor:
  enabled: true
```

8. chart_v1alpha1_xl_cr.yaml ファイルの編集が完了したら、変更を保存して適用します。

- 変更を保存して、テキストエディタを終了します。
- 変更を適用します。

コマンドを実行します:

```
kubectl apply -f \
  /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

- API およびエクストラクタポッドを削除します。

エクストラクタポッドを削除すると、再起動がトリガーされ、行った変更が読み込まれます。

完全なポッド名を取得するには、`kubectl get pods -n turbonomic` コマンドを実行します。次に、`api` と `extractor` で始まるポッドの 2 つのエントリを見つけます。たとえば、次のエントリがあるとします。

```
...
api-7887c66f4b-shndq          1/1      Running   0
...
extractor-5b86976bc8-vxwz4    1/1      Running   0
...
```

次のコマンドを実行します。

```
- kubectl delete pod -n turbonomic api-7887c66f4b-shndq
- kubectl delete pod -n turbonomic extractor-5b86976bc8-vxwz4
```

9. インストールを確認します。

インストールを再確認するには、次の手順を実行します。

- 組み込みレポートポッドが実行されていることを確認します。

ポッドが実行されていることを確認するには、`kubectl get pods -n turbonomic` を実行します。出力には、次のようなエントリが含まれている必要があります。

NAME	READY	STATUS	RESTARTS
extractor-7759dbcb47-vs6hr	1/1	Running	0
grafana-84ccb4bfb-17sp7	1/1	Running	0

- **Postgres が実行されていることを確認します。**

Postgres データベースは、Workload Optimization Manager サーバマシン上でデーモンとして実行されている必要があります。ステータスを確認するには、次のコマンドを実行します。

```
sudo systemctl status postgresql-12.service.
```

以下のような出力が表示されます。

```
postgresql-12.service - PostgreSQL 12 database server
Loaded: loaded (/usr/lib/systemd/system/postgresql-12.service; enabled; vendor preset: disabled)
Active: active (running) since Wed 2020-07-29 06:39:43 UTC; 14h ago
      Docs: https://www.postgresql.org/docs/12/static/
Process: 1536 ExecStartPre=/usr/pgsqli-12/bin/postgresql-12-check-db-dir ${PGDATA} (code=exited, status=0/SUCCESS)
Main PID: 1562 (postmaster)
      Tasks: 15
      Memory: 145.5M
      CGroup: /system.slice/postgresql-12.service
              ## 419 postgres: TimescaleDB Background Worker Scheduler
              ## 1562 /usr/pgsqli-12/bin/postmaster -D /var/lib/pgsqli/12/data/
              ## 1928 postgres: logger
              ## 1986 postgres: checkpointer
              ## 1988 postgres: background writer
              ## 1989 postgres: walwriter
              ## 1990 postgres: autovacuum launcher
              ## 1991 postgres: stats collector
              ## 1992 postgres: TimescaleDB Background Worker Launcher
              ## 1994 postgres: logical replication launcher
              ## 4054 postgres: grafana_backend grafana 10.233.90.172(33038) idle
              ## 4884 postgres: grafana_backend grafana 10.233.90.172(35814) idle
              ## 4912 postgres: grafana_reader extractor 10.233.90.172(33898) idle
              ##11365 postgres: grafana_reader extractor 10.233.90.172(40728) idle
              ##32367 postgres: TimescaleDB Background Worker Scheduler
```

(任意) レポートの編集、PDF、および電子メールサブスクリプション

組み込みレポートを有効にする手順を完了したら、Workload Optimization Manager のナビゲーションバーで [Reports] をクリックして、新しいブラウザタブで Grafana ダッシュボードを開くことができます。Grafana ダッシュボードから、既存のダッシュボードのリストを確認できます。

このビューからレポートを作成および編集し、それらのレポートをサブスクライバに送信するには、次の手順を実行する必要があります。

- 単一のレポートエディタのユーザーアカウントを設定します。

このユーザーには、レポートの作成と編集、PDF の作成、および電子メールサブスクリプションの設定権限があります。

- Workload Optimization Manager に Grafana ライセンスをインストールします。
このライセンスは無料です。
- Workload Optimization Manager の電子メールプロキシを設定して、Workload Optimization Manager から送信される電子メールメッセージを有効にします。
有効にすると、Workload Optimization Manager からレポートを電子メールで送信できます。

注：

Workload Optimization Manager の組み込みレポートは、Grafana ダッシュボードを使用してレポートを表示します。レポートから PDF を生成し、Grafana からサブスクライバに PDF を電子メールで送信する手順については、Grafana レポートのドキュメントを参照してください。

<https://grafana.com/docs/grafana/latest/enterprise/reporting/>

レポートエディタ権限を持つユーザーアカウントの設定

レポートを編集するには、ライセンスユーザーが必要です。ここで設定するユーザーアカウントは、レポートを編集してサブスクリプションを設定するユーザー権限が付与されるライセンスにアクセスできます。

ユーザーアカウントを作成するには、次の手順を実行します。

1. Workload Optimization Manager で、[User Management] ページを表示します。
[Settings] > [User Management] に移動します。
2. レポートエディタとして設定するユーザーアカウントを選択します。
既存のアカウントを編集するか、新しいアカウントを作成できます。
3. ユーザーのロールを選択します。
ユーザーは、共有ロールまたはスコープロール以外の任意のロールを持つことができます。
4. レポートエディタの権限を設定します。
[Options] で、[DESIGNATE AS REPORT EDITOR] を選択します。
5. 必要なユーザーアカウントにその他のプロパティを設定し、ユーザーアカウントを保存します。
ユーザーアカウントの作成方法の詳細については、Workload Optimization Manager ユーザーガイド [英語] の「Managing User Accounts」を参照してください。

Grafana ライセンスのインストール

レポートエディタのユーザーアカウントでライセンス権限を使用するには、Grafana ライセンスをインストールする必要があります。Grafana ライセンスをインストールするには、次の手順を実行します。

1. 無料ライセンスを要求します。
サポート担当者に連絡して、ライセンスファイルを要求します。このリクエストには、レポートページへの Workload Optimization Manager の URL が含まれている必要があります。たとえば、ドメイン **MyCompany.com** から Workload Optimization Manager のユーザーインターフェイスにログインする場合、次の URL を指定します。

`https://MyCompany.com/reports/`

この URL は、指定するドメインにライセンスが適用されることを保証するために、ライセンスファイルに組み込まれています。

2. ファイルをローカル マシンに保存します。
このファイルには、.jwt ファイル名の拡張子が付いている必要があります。
3. .jwt ファイルを Workload Optimization Manager のライセンスとしてインストールします。

Workload Optimization Manager にライセンスをインストール場合と同じように、ライセンスファイルをインストールします。

- a. [License] ページに移動します。
[Settings] > [License] に移動します。
- b. Workload Optimization Manager インスタンスにライセンスを適用します。
[Import License] をクリックします。ライセンスファイルを [Enter License] スライドアウトにドラッグします。あるいは、ローカルマシンでライセンスファイルを参照して開くこともできます。

ライセンスをインストールしたら、[Reports] ビューに移動します (Workload Optimization Manager のナビゲーションバーで [Reports] をクリックします)。ダッシュボードに移動して [Share Dashboard] をクリックすると、PDF の作成を選択できるようになりました。ダッシュボードの PDF を電子メールで送信するには、Workload Optimization Manager で SMTP リレーを設定する必要があります。

SMTP リレーの設定

SMTP リレーにより、Workload Optimization Manager はレポートサブスクライバに電子メールを送信できます。Workload Optimization Manager で SMTP リレーを設定するには、次の手順を実行します。

1. [Email Settings] ページに移動します。
[Settings] > [Email and Trap Notifications] に移動します。
2. SMTP 設定を設定します。
[SMTP Settings] フィールドでは、Workload Optimization Manager からの電子メール通信を有効にするためにネットワークで使用するメールリレーサーバーを指定します。ここで設定したリレーによって、レポートをサブスクライバに送信する電子メールが有効になります。
サーバーに認証が必要な場合は、ユーザー名とパスワードをここで入力します。次の通知の暗号化オプションも選択できます。
 - なし
 - Ssl
 - Tls
3. Workload Optimization Manager によって送信される電子メールの返信先アドレスを設定します。
[General Email Settings] セクションで FROM アドレスを指定します。

レポートエディタユーザーを作成し、Grafana ライセンスをインストールし、SMTP リレーを設定したら、ダッシュボードの PDF を作成して、サブスクライバに送信できます。

組み込みレポートのストレージ要件の見積もり

組み込みレポート機能では、TimescaleDB サーバーを使用してチャートデータが管理されます。これは、TimescaleDB 拡張機能を使用して実行される PostgreSQL サーバーです。Workload Optimization Manager インスタンスのデータストアを設定して、TimescaleDB 要件をサポートするのに十分なスペースを確保する必要があります。

最初に組み込みレポートを有効にするときは、必要なストレージ容量を見積もり、その容量に応じてプラットフォームストレージを設定する必要があります。組み込みレポートをすでに有効にしている場合は、現在のストレージ構成を確認し、現在および将来のニーズを満たしているか判断する必要があります。

TimescaleDB に必要なストレージは、以下の内容に依存します。

- データ保持期間
TimescaleDB データを保存する期間。
- 使用環境のサイズ
Workload Optimization Manager が管理する使用環境内のエンティティの数。この数は時間の経過とともに変化するため、所定のデータ保持期間における使用環境内のエンティティの平均数と考える必要があります。

また、エンティティ数が増えると、他のアクティビティと同様にデータ要件も増えることに注意してください。ストレージ要件は、次のような理由で時間の経過とともに増加する可能性があります。

- ワークロード、アプリケーション コンポーネント、ストレージ、またはホストなどのエンティティを使用環境に追加する。
- 新しいターゲットを設定する。

ストレージ見積もりルックアップテーブル

シスコでは、さまざまなトポロジと保持期間の TimescaleDB ストレージ要件を調査しています。次の表に、計算した見積もりを示します。使用環境によって要件が異なる場合があることに注意してください。

保持期間	エンティティの数						
	10k	25k	50k	100k	250k	500k	1000k
6 ヶ月	36GB	91GB	182GB	364GB	910GB	1.8TB	3.6TB
1 年	72GB	181GB	361GB	723GB	1.8TB	3.5TB	7TB
2 年	144GB	361GB	721GB	1.4TB	3.5TB	7.2 TB	14TB

デフォルトのインストールでは、TimescaleDB に 200GB のディスククォータが付与されます。デフォルトのインストールでは、データベースが次のエンティティ数をサポートできると見積もっています。

保持期間	エンティティ数
6 ヶ月	55k
1 年	27k
2 年	14k

データ保持期間の設定

デフォルトでは、Workload Optimization Manager の組み込みレポートの保持期間は 365 日に設定されます。現在設定されている保持期間を確認し、Workload Optimization Manager のユーザーインターフェイスで変更できます。

これらのアクションを実行するには、[Maintenance Options] ページに移動し、次の手順を実行します。

1. [Settings] ページに移動します。



クリックして [Settings] ページに移動します。

2. [Maintenance Options] を選択します。



Maintenance Options

3. 組み込みレポートのデータ保持期間を設定します。

コントロールの [Data Retention] グループで、[Saved Reporting Data] のフィールドを見つけます。組み込みレポートの現在のデータ保持期間が日単位で表示されます。デフォルトは 365 日です。

保持期間を変更するには、別の日数を入力し、[Apply Settings] をクリックします。

TimescaleDB のストレージ容量の増加

Workload Optimization Manager のインストール後に組み込みレポートのストレージ要件を見積もると、TimescaleDB で使用できるストレージ容量を増やす必要があることがわかる場合があります。

プラットフォームでは、論理ボリューム管理 (LVM) を使用して VM ディスクが管理されます。データベースストレージを増やすには、新しいディスクを VM に追加し、そのディスクを使用して LVM 論理ボリューム (/dev/turbo/var_lib_mysql) を拡張する必要があります。この論理ボリュームは、履歴データベースと組み込みレポートデータベースの両方を提供します。完了したら、TimescaleDB のクォータを増やします。

詳細については、[「使用可能なディスク容量の増加」 \(22 ページ\)](#) を参照してください。

エンティティ数の見積もり

使用環境内のエンティティ数を把握するには、各検出サイクルのエンティティ数をリストする INFO メッセージをログファイルで検索します。文字列 INFO [Stages\$BroadcastStage] を検索できます。INFO 文字列は次のようになります。

```
topology-processor-6f6486df64-zf 2021-09-27 20:51:33,724 INFO [Stages$BroadcastStage] : Successfully sent 1505 entities within topology...
```

この例は、トポロジに 1505 のエンティティがあることを示しています。時間の経過とともにインベントリがどのように変化するかを考慮する必要があります。たとえば、エンティティ数を時間の経過とともにチェックして、定期的に増加するかどうかを確認できます。

(任意) データエクスポートの有効化

データエクスポートをサポートするために、Workload Optimization Manager には、データを標準形式にストリーミングできるエクストラクタ コンポーネントがあります。ストリーミングされたデータは、Elasticsearch などの検索および分析サービスにロードできます。

データエクスポートを有効にするには、次の手順を実行する必要があります。

- エクストラクタ コンポーネントを有効にします。

エクストラクタは、Workload Optimization Manager のインストールの一部として実行されるコンポーネントです。エクストラクタはデフォルトでは有効になっていません。

- エクストラクタのストリームをデータサービスに配信するコネクタを展開します。

エクストラクタは、Workload Optimization Manager データを Kafka トピックとして公開します。コネクタにより、データサービスがデータトピックを使用できるようになります。このドキュメントには、サンプルの Elasticsearch コネクタの展開ファイルが含まれています。

エクストラクタ コンポーネントの有効化

データエクスポートを有効にする最初の手順では、エクストラクタ コンポーネントを有効にします。エクストラクタを有効にするには、次の手順を実行します。

1. Workload Optimization Manager インスタンスへの SSH ターミナルセッションを開きます。
Workload Optimization Manager のインストール時に設定したシステム管理者でログインします。
 - ユーザー名 : turbo
 - ユーザー名 : [your_private_password]
2. cr.yaml ファイルを編集して、エクストラクタ コンポーネントを有効にします。
同じ SSH セッションで、編集のために cr.yaml ファイルを開きます。次に例を示します。

```
vi /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

3. エクストラクタ コンポーネントのエントリを編集します。

注：

組み込みレポートを有効にしている場合、エクストラクタ コンポーネントはすでに有効になっています (`true` に設定されています)。

データエクスポートを有効にしなくても組み込みレポートを有効にできるのと同様に、組み込みレポートを有効にしなくてもデータエクスポートを有効にできることを理解する必要があります。

`cr.yaml` ファイルでエクストラクタエントリを検索します。次のように表示されます。

```
extractor:
  enabled: false
```

エントリを `true` に変更します。

4. エクストラクタプロパティのエントリを編集します。

`cr.yaml` ファイルでエクストラクタエントリを検索します。次のように表示されます。

```
properties:
  extractor:
    enableDataExtraction: false
```

エントリを `true` に変更します。

5. 変更を保存してプラットフォームに適用します。

変更を保存したら、`kubectl` を使用して変更を適用します。

```
kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

6. エクストラクタ コンポーネントが実行されていることを確認します。

プラットフォームでコンポーネントが再起動されるのを待ちます。次のコマンドを実行します。

```
kubectl get pods -n turbonomic
```

以下のような出力が表示されます。

NAME	READY	STATUS	RESTARTS
...			
extractor-5f41dd61c4-4d61q	1/1	Running	0
...			

エクストラクタ コンポーネントのエントリを探します。エントリが存在する場合、エクストラクタ コンポーネントはインストールされて、実行されています。

コネクタの導入

エクストラクタは、Workload Optimization Manager データを Kafka トピックとして公開します。このデータを検索および分析サービスにロードするには、該当サービスにコネクタを展開する必要があります。たとえば、データを Elasticsearch にロードする場合は、Elasticsearch コネクタを展開する必要があります。

Workload Optimization Manager プラットフォームを実行するのと同じ Kubernetes ノードにコネクタを展開します。展開するには、コネクタに必要なポッドを宣言する Kubernetes Deployment を作成します。以下に、Elasticsearch へのコネクタのサンプル展開を示します。

コネクタを展開するには、エクストラクタ コンポーネントを実行しているのと同じホストに展開 yml ファイルを作成し、次のコマンドを実行します。

```
kubectl create -f <MyConnectorDeployment.yaml>
```

<MyConnectorDeployment.yaml> は展開ファイルの名前です。

展開されたポッドの名前が es-kafka-connect である場合、コネクタが実行されていることを確認するには、`kubectl get pods -n turbonomic` を実行します。以下のような出力が表示されます。

NAME	READY	STATUS	RESTARTS
...			
es-kafka-connect-5f41dd61c4-4d61q	1/1	Running	0
...			

コネクタを展開したら、Workload Optimization Manager の分析サイクル (約 10 分) が終了するのを待ちます。その後、データサービスに JSON としてロードされた、Workload Optimization Manager 環境からのエンティティとアクションを確認できます。

コネクタの展開例

サービスがエクスポートされたデータを処理できるように、コネクタを Elasticsearch に展開する場合、たとえば、Elasticsearch で Kibana を使用して、データダッシュボードを表示できます。次の手順を実行しているとします。

- Workload Optimization Manager を実行しているネットワーク上の VM に Elasticsearch を展開している。Elasticsearch ホストは、Workload Optimization Manager Kubernetes ノードから確認できます。コネクタの展開でこのホストアドレスを指定します。
- Workload Optimization Manager データをロードする Elasticsearch インデックスを設定している。コネクタの展開でこのインデックスを指定します。

次のリストは、Logstash イメージを使用してエクストラクタデータを収集し、Elasticsearch ホストにパイプする展開です。展開では、ストレージボリュームもセットアップし、エクストラクタからの入力を設定し、Elasticsearch インスタンスへの出力を設定します。

リストを確認するときは、次の点に注意してください。

■ Elasticsearch ホストの場所とログイン情報 :

```
...
  env:
    - name: ES_HOSTS
      value: "<UrlToMyElasticsearchHost>"
    - name: ES_USER
      value: "<MyElasticsearchUser>"
    - name: ES_PASSWORD
      valueFrom:
        secretKeyRef:
          name: <MyES_KeyName>
          key: <MyES_Key>
...

```

Logstash は次の環境変数を使用します。

- ES_HOSTS : エクスポートされたデータをパイプする場所を識別します。
- ES_USER : Elasticsearch のユーザーアカウントを識別します。
- ES_PASSWORD : アカウントのログイン用。このコネクタの例は、Elasticsearch パスワードを Kubernetes シークレットとして保存していることを前提としています。

Logstash は、ES_HOSTS 環境変数を使用して、エクスポートされたデータをパイプする場所を識別します。

■ Kafka トピックの名前 :

```
...
  logstash.conf: |
    input {
      kafka {
        topics => ["turbonomic.exporter"]
      }
    }
...

```

Logstash の入力設定では、turbonomic.exporter という名前の単一のトピックが必要です。

■ Logstash の出力設定は、ES_HOSTS 環境変数によって識別される Elasticsearch サーバーに対するものです。
 <MyElasticsearchIndex> の代わりに独自の Elasticsearch インデックスを指定します。

```
...
  output {
    elasticsearch {
      index => "<MyElasticsearchIndex>"
      hosts => [ "${ES_HOSTS}" ]
    }
  }
...

```

サンプルリスト : Elasticsearch コネクタ

このリストは、データエクスポートの Elasticsearch コネクタを作成するために使用できる展開ファイルのサンプルです。ユーザー名やパスワードなど、一部の設定を変更する必要があることに注意してください。また、コネクタを特定の環境に準拠させるために、ポートやその他の設定を指定する必要がある場合もあります。

```
apiVersion: apps/v1
kind: Deployment
metadata:
```

```

name: elasticsearch-kafka-connect
labels:
  app.kubernetes.io/name: elasticsearch-kafka-connect

spec:
  replicas: 1
  selector:
    matchLabels:
      app.kubernetes.io/name: elasticsearch-kafka-connect
  template:
    metadata:
      labels:
        app.kubernetes.io/name: elasticsearch-kafka-connect
    spec:
      containers:
        - name: logstash
          image: docker.elastic.co/logstash/logstash:7.10.1
          ports:
            - containerPort: 25826
          env:
            - name: ES_HOSTS
              value: "<UrlToMyElasticsearchHost>"
            - name: ES_USER
              value: "<MyElasticsearchUser>"
            - name: ES_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: <MyES_KeyName>
                  key: <MyES_Key>
      resources:
        limits:
          memory: 4Gi
        volumeMounts:
          - name: config-volume
            mountPath: /usr/share/logstash/config
          - name: logstash-pipeline-volume
            mountPath: /usr/share/logstash/pipeline
      volumes:
        - name: config-volume
          configMap:
            name: logstash-configmap
            items:
              - key: logstash.yml
                path: logstash.yml
        - name: logstash-pipeline-volume
          configMap:
            name: logstash-configmap
            items:
              - key: logstash.conf
                path: logstash.conf
    ---
  apiVersion: v1
  kind: ConfigMap
  metadata:
    name: logstash-configmap

```



```

data:
  logstash.yml: |
    http.host: "0.0.0.0"
    path.config: /usr/share/logstash/pipeline
  logstash.conf: |
    input {
      kafka {
        topics => ["turbonomic.exporter"]
        bootstrap_servers => "kafka:9092"
        client_id => "logstash"
        group_id => "logstash"
        codec => "json"
        type => "json"
        session_timeout_ms => "60000" # Rebalancing if consumer is found dead
        request_timeout_ms => "70000" # Resend request after 70 seconds
      }
    }
    filter {
    }
    output {
      elasticsearch {
        index => "<MyElasticsearchIndex>"
        hosts => [ "${ES_HOSTS}" ]
        user => "${ES_USER}"
        password => "${ES_PASSWORD}"
      }
    }
  ---
  apiVersion: v1
  kind: Service
  metadata:
    labels:
      app: elasticsearch-kafka-connect
    name: elasticsearch-kafka-connect
  spec:
    ports:
      - name: "25826"
        port: 25826
        targetPort: 25826
    selector:
      app: elasticsearch-kafka-connect
  
```

(任意) プラットフォームノードの IP アドレスの変更

Workload Optimization Manager の標準インストール (VM イメージとしてインストール) の場合、プラットフォームの IP アドレスの変更が必要になる場合があります。たとえば、VM を移動する場合、別のアドレスを割り当てる必要がある場合があります。プラットフォームの IP アドレスを変更する必要がある場合、提供されているスクリプトを使用できます。

注：

Workload Optimization Manager インストールの IP アドレスは、できるだけ変更しないでください。これは、予期しない依存関係に影響を与える可能性のあるセンシティブなアクションです。

Workload Optimization Manager バージョン 3.0.0 以降の IP アドレスのみ、以下の手順を使用して変更する必要があります。IP アドレスを変更する必要があり、バージョン 3.0.0 以降に更新できない場合は、サポート担当者に連絡してください。

Workload Optimization Manager VM の IP アドレスを変更するには、次の手順を実行します。

1. 必要な情報を用意します。

プラットフォームの現在の IP アドレスと、使用する新しい IP アドレスの両方を特定します。

VM でシェルセッションを開いてコマンドを実行するためのログイン情報も知っている必要があります。

2. VM の完全なスナップショットを作成します。

IP アドレスを変更する前に、インストールの完全なスナップショットを作成することが重要です。

3. VM の IP アドレスを変更します。

Workload Optimization Manager VM には、このタスクを実行するための `ipsetup` スクリプトが含まれています。

- a. Workload Optimization Manager VM への SSH ターミナルセッションを開きます。

次のログイン情報を使用します。

- ユーザー名 : turbo
- パスワード : プラットフォームを最初にインストールしたときに turbo アカウントに割り当てたパスワードを指定します。

- b. セッションが開いたら、`ipsetup` スクリプトを実行します。

```
sudo /opt/local/bin/ipsetup
```

スクリプトを実行すると、次の入力を求められます。

注：

これらの必須フィールドには値を指定する必要があります。指定しない場合、インストールが失敗するか、VM にアクセスできなくなる可能性があります。

- **必須** : Do you want to use DHCP or set a static IP...
Choose static
- **必須** : Please enter the IP Address for this machine
- **必須** : Please enter the network mask for this machine
- **必須** : Please enter the Gateway address for this machine
- **必須** : Enter DNS Server(s) IP Address for this machine

指定した IP アドレスをメモしておく必要があります。

- c. IP の変更を VM 上の Kubernetes クラスタに伝達します。

```
sudo /opt/local/bin/kubeNodeIPChange.sh
```

- d. 変更が成功したことを確認します。

新しく配置されたインストールの Workload Optimization Manager ユーザーインターフェイスにログインし、正しく表示されることを確認します。サブライチェーン、グループ、およびポリシーを確認し、チャートにデータが正しく表示されることも確認する必要があります。

変更が成功したことを確認したら、古い場所にある VM のスナップショットを削除できます。

(任意) プローブコンポーネントの有効化と無効化

Workload Optimization Manager では、プローブはターゲットに接続するプラットフォーム コンポーネントです。ターゲットのエンティティを検出して Workload Optimization Manager のサプライチェーンにロードし、ターゲット環境のデバイスでアクションを実行できます。Workload Optimization Manager には、Workload Optimization Manager を使用環境に接続するために使用できる多数のプローブコンポーネントが付属しています。

Workload Optimization Manager を初めてインストールすると、デフォルトで特定のプローブセットが有効になり、他のプローブは無効のままになります。各プローブは、Workload Optimization Manager インストールのリソースを消費します。不要なプローブがある場合は、無効化することを検討する必要があります。一方、必要なプローブが無効化されている場合は、有効にしてサービスを開始する必要があります。

注：

Workload Optimization Manager の進化に伴い、提供されるプローブのセットは変わる可能性があります。また、バージョンごとに、デフォルトで有効になっているプローブのセットは変わる可能性があります。ただし、新しいバージョンに更新しても、プローブ設定が変更されることはありません。新しいバージョンに更新しても、展開内の新しいプローブは自動的に有効になりません。更新時に新しいプローブを利用する場合は、手動で有効にする必要があります。

使用可能なプローブの最新リストの表示

Workload Optimization Manager のバージョンを更新すると、より多くのプローブが更新で利用できるようになります。ただし、この更新では、有効または無効になっているプローブの現在の設定は変更されないため、更新に付属する新しいプローブは、デフォルトでは使用できません。

新しいプローブを有効にするには、まずプローブの内部名を知っている必要があります。現在のバージョンで使用できるプローブのリストを取得するには、`values.yaml` ファイルの内容を表示します。

1. Workload Optimization Manager インスタンスの SSH ターミナルセッションを開きます。

Workload Optimization Manager のインストール時に設定したシステム管理者でログインします。

- ユーザー名：

```
turbo
```

- Password:

```
[your_private_password]
```

2. 使用可能なプローブのリストを表示します。

```
cat /opt/turbonomic/kubernetes/operator/helm-charts/xl/values.yaml
```

次のような結果が表示されます。

```
customdata:
  enabled: false
dynatrace:
  enabled: false
gcp:
  enabled: false
hpe3par:
  enabled: false
...
```

このリストは、プローブの内部名を示します。設定済みプローブのリストに新しいプローブを追加する場合は、内部名を使用し、`enabled: true` に設定する必要があります。

設定済みプローブの最新リストの表示

Workload Optimization Manager の現在のインストールには、使用可能なプローブの特定のセットがあります。一部のプローブが有効になり、一部のプローブが無効になる可能性があります。現在使用可能なプローブの現在の設定を表示するには、Workload Optimization Manager インストールの `cr.yaml` ファイルを開き、プローブエントリを確認します。

1. 同じ SSH セッションで、編集のために `cr.yaml` ファイルを開きます。次に例を示します。

```
vi /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

2. プローブのリストを検索します。

リストには、現在のインストール用に設定されているすべてのプローブが含まれます。リストは次のようになります。

```
actionscript:
  enabled: true
appdynamics:
  enabled: true
appinsights:
  enabled: true
aws:
  enabled: true
azure:
  enabled: true
dynatrace:
  enabled: true
hpe3par:
  enabled: true
horizon:
  enabled: false
hyperflex:
  enabled: false
...
```

このリストは、インストール用に現在設定されているすべてのプローブを識別し、プローブが有効 (`true`) か無効 (`false`) かを示します。

注：

このプローブのリストは、インストールで実行されているプローブポッドのリストとは異なります。一部のプローブは複数のポッドを使用します。プローブポッド名には次の規則を使用します。{**ProbeName**} はプローブの内部名（前述のリスト）であり、{**NameExtension**} は名前の任意の拡張子です（このプローブに複数のポッドがある場合）。

```
mediation-{ProbeName}{NameExtension}
```

たとえば、`kubectl get pods -n turbonomic` を実行すると、vcenter プローブの結果が次のように表示されます。

NAME	READY	STATUS	RESTARTS
mediation-vcenter-5bc4f5fbd4-nzm4j	1/1	Running	0
mediation-vcenterbrowsing-5c5987f66c-bfjq4	1/1	Running	0

プローブの有効化/無効化

Workload Optimization Manager でプローブを有効または無効にするには、`cr.yaml` ファイルを編集して新しいプローブを追加し、`enabled`: プロパティの値を変更します。次に、変更を適用して、プラットフォーム コンポーネントをリロードします。

1. 同じ SSH セッションで、編集のために `cr.yaml` ファイルを開きます。次に例を示します。

```
vi /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

2. プローブエントリを編集します。

現在設定されているプローブを有効または無効にするには、編集するプローブを見つけ、設定を変更してプローブを有効または無効にします。

新しいプローブをリストに追加するには、`cat` を使用して使用可能なプローブを表示し、出力から必要なプローブエントリをコピーします。次に、そのエントリを `cr.yaml` ファイルに貼り付け、`enabled: true` に設定します。

3. 変更を保存してプラットフォームに適用します。

変更を保存したら、`kubectl` を使用して変更を適用します。

```
kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

4. プローブが正しくインストールされ、Workload Optimization Manager のすべてのポッドが開始していることを確認します。

`kubectl get pods -n turbonomic` を実行し、プローブを実装するメディエーションポッドのリストを確認します。すべてのポッドに、次のような `READY` および `STATUS` 状態が表示されることに注意してください。

```
NAME                READY   STATUS    RESTARTS
[...]              1/1    Running   0
```

5. ユーザーインターフェイスで新しいプローブ設定を表示します。

ブラウザを更新して、[Target Management] ページに移動します。設定の変更に一致するターゲットカテゴリとタイプが表示されます。



ライセンスのインストールおよび 初回ログイン

開始する前に、別の電子メールで送信されたフル ライセンス キー ファイルまたはトライアル ライセンス キー ファイルがあることを確認してください。Workload Optimization Manager のインストールにアップロードできるように、ライセンスファイルをローカルマシンに保存します。

Workload Optimization Manager を初めて使用するには、以下のステップを実行します。

1. インストールされている WorkloadOptimizationManager インスタンスの IP アドレスを Web ブラウザに入力して接続します。
2. Workload Optimization Manager へログインします。
 - USERNAME : administrator のデフォルトのログイン情報を使用します。
 - PASSWORD のパスワードを入力します。
 - パスワードをもう一度入力して、[REPEAT PASSWORD] を確認します。
 - [CONFIGURE] をクリックします。
3. Workload Optimization Manager のインストールのセットアップを続行します。

[開始 (LET'S GO)] をクリックします。
4. [Enter License] スライドアウトを開きます。

[ライセンスのインポート (IMPORT LICENSE)] をクリックします。
5. ライセンス キー ファイルをアップロードします。
 - a. [ライセンスの入力] スライド アウトで、次の方法のいずれかを使用してライセンスをアップロードすることができます。
 - [ライセンスの入力] スライド アウトにライセンス キーのファイルをドラッグします。
 - ライセンス キー ファイルを参照します。

.xml or .lic ファイルのみをアップロードしてください。
 - b. [SAVE] をクリックします。



シングルサインオン認証

会社のポリシーがシングルサインオン (SSO) 認証をサポートしている場合、セキュリティ アサーション マークアップ言語 (SAML) 2.0 または OpenID Connect 1.0 を使用して、SSO 認証をサポートするように Workload Optimization Manager を設定できます。

大まかに次の手順を実行します。

- SSO 用に複数の外部グループ、または 1 人以上の外部ユーザーを作成します。『*Workload Optimization Manager ユーザー ガイド*』の「ユーザー アカウントの管理」を参照してください。
- SSO 認証を使用するように Workload Optimization Manager を設定します。次のいずれかを設定します。
 - SAML ID プロバイダー (IdP) を介した SSO。 [「SAML 認証の設定」 \(56 ページ\)](#) を参照してください。
 - OpenID ID プロバイダーを介した SSO。 [「OpenID 認証の設定」 \(60 ページ\)](#) を参照してください。

このセクションでは、SAML または OpenID を使用して SSO をサポートするように Workload Optimization Manager を設定する方法について説明します。

SSO が有効になっている場合、ユーザーは自身の SSO ログイン情報を使用して Workload Optimization Manager インスタンスにログインします。SSO が有効になると、ユーザーはログイン用のローカルまたは Active Directory (AD) のログイン情報を入力できなくなります。ID プロバイダ (IdP) により認証が実行されます。

前提条件

開始する前に、IdP が SSO 用に設定されていることを確認してください。独自仕様または公開 IdP を使用できます。パブリック Okta IdP の設定例については、 [「IdP の一般的な設定」 \(74 ページ\)](#) を照してください。

SAML 認証の設定

セキュリティ アサーション マークアップ言語 (SAML) は、当事者間で認証および承認データを交換するための XML ベースのオープン標準です。SAML を使用して認証するように Workload Optimization Manager を設定するには、次の手順を実行します。

1. (必須) 複数の外部グループまたは SSO に最低でも 1 個の外部グループを作成します。

重要事項：

SSO が有効になっている場合、Workload Optimization Manager は SSO IdP を介したログインのみを許可します。Workload Optimization Manager のインストールに移動するたびに、ユーザーは認証のために SSO ID プロバイダ (IdP) にリダイレクトされてから、Workload Optimization Manager のユーザーインターフェイスが表示されます。

Workload Optimization Manager のインストールで SSO を有効にする前に、Workload Optimization Manager の管理者権限を持つ SSO ユーザーを少なくとも 1 人設定する必要があります。そうしないと、SSO を有効にした後、Workload Optimization Manager で SSO ユーザーを設定できなくなります。SSO ユーザーを管理者として承認するには、[EXTERNAL AUTHENTICATION] を使用して次のいずれかを実行します。

- 管理者承認を持つ 1 人の SSO ユーザーを設定します。
外部ユーザーを追加します。ユーザー名は、IdP で管理されているアカウントと一致する必要があります。
- 管理者承認を持つ SSO ユーザー グループを設定します。
外部グループを追加します。グループ名は IdP でユーザー グループと一致する必要があり、そのグループは少なくとも 1 人のメンバーが必須です。

SSO への複数の外部グループまたは外部ユーザーの作成については、『*Workload Optimization Manager ユーザー ガイド*』の「ユーザー アカウントの管理」を参照してください。

2. (必須) chrony が設定されており、Workload Optimization Manager インスタンスのシステム時刻が正しいことを確認してください。

手順については、[「時刻の同期」 \(17 ページ\)](#) を参照してください。

3. IdP からメタデータを取得します。

そのメタデータを使用して、次の場所にある Workload Optimization Manager CR ファイルで SSO を設定します。

```
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

メタデータを取得するには、次の手順を実行します。

- a. セキュリティ管理者に連絡して、IdP からメタデータを取得します。
- b. メタデータファイルをローカルマシンのディレクトリに保存します。たとえば、ファイルを次の場所に保存します。

```
/tmp/MySamlMetadata.txt
```

- c. メタデータを [「IdP メタデータの例」 \(59 ページ\)](#) にあるサンプルと比較します。
保存したファイルを取り出します。ファイルはサンプルに類似している必要があります。

4. IdP から証明書を取得します。

セキュリティ管理者に連絡して、IdP から証明書を取得します。

5. SAML 設定で CR ファイルを更新します。

これで、SAML 経由で SSO を設定するために必要なデータが揃いました。Workload Optimization Manager ノードを設定する cr.yaml ファイルを編集してから、ノードを展開または再起動します。

- ダウンロードした SAML メタデータの内容を表示します。
たとえば、ローカルマシンの次の場所にファイルを保存したと仮定して、次のコマンドを実行します。


```
cat /tmp/MySamlMetadata.txt
```

- 編集のために CR ファイルを開きます。

シェルで、Workload Optimization Manager VM の `deploy/crds` ディレクトリにディレクトリを変更します。

```
cd /opt/turbonomic/kubernetes/operator/deploy/crds
```

次に、編集のために CR ファイルを開きます。たとえば、VI でファイルを開くには、次のように入力します。

```
vi charts_v1alpha1_xl_cr.yaml
```

このファイルを編集するときに、IdP から取得したメタデータを参照します。

- CR ファイルで、API コンポーネントのエントリに移動します。
CR ファイルで、次のエントリを検索またはスクロールします。

```
apiVersion: charts.helm.k8s.io/v1alpha1
```

`spec:properties:api:` の下で、このコンポーネントの仕様を変更します。

- SAML 機能をオンにします。

最初の API プロパティについては、次のように設定します。

```
samlEnabled: true
```

- SSO エンドポイントを設定します。

SAML メタデータで、`md:SingleSignOnService` のエントリを見つけます。その要素内で、`Location` 属性を見つけます。`Location` の値は SSO エンドポイントです。提供されているサンプルメタデータを使用して、CR ファイルで次の設定を行います。

```
samlWebSsoEndpoint: https://dev-771202.oktapreview.com/app/ibmdev771202_turbo2_1/exkex16xc9MhzqiC30h7/sso/saml
```

- SAML エンティティ ID を設定します。

SAML メタデータで、`md:EntityDescriptor` のエントリを見つけます。その要素内で、`entityID` 属性を見つけます。提供されているサンプルメタデータを使用して、CR ファイルで次の設定を行います。

```
samlEntityId: http://www.okta.com/exkex16xc9MhzqiC30h7
```

- SAML 登録を設定します。

次のプロパティを設定します。

```
samlRegistrationId: simplesamlphp
```

- SAML SP エンティティ ID を設定します。

次のプロパティを設定します。

```
samlSpEntityId: turbo
```

- SAML 証明書をを入力します。

IdP から取得したメタデータで、`<ds:X509Certificate>` のエントリを見つけます。このタグの内容をコピーします。つまり、`<ds:X509Certificate>` と `</ds:X509Certificate>` の間にある文字をコピーします。

CR ファイルの API プロパティセクションに証明書のエントリを作成します。新しい行で、次のように入力します。

```
samlIdpCertificate: |
```

次に、作成したエントリの後に新しい行を開き、メタデータファイルからコピーした証明書の内容を貼り付けます。

CR ファイルの完成した API セクションは、次のようになります。

```
apiVersion: charts.helm.k8s.io/v1alpha1
kind: Xl
metadata:
  name: xl-release
spec:
  properties:
    api:
      samlEnabled: true
      samlWebSsoEndpoint: https://dev-771202.oktapreview.com/app/ibmdev771202_turbo2_1/exkexl6xc9Mhz
qiC30h7/sso/saml
      samlEntityId: http://www.okta.com/exkfdsn6oy5xywqCO0h7
      samlRegistrationId: simplesamlphp
      samlSpEntityId: turbo
      samlIdpCertificate: |
        -----BEGIN CERTIFICATE-----
        MIIDpDCCAoygAwIBAgIGAWMnhv7cMA0GCSqGSIb3DQEBCwUAMIGSMQswCQYDVQQGEwJVUzETMBEG
        A1UECAwKQ2FsaWZvcn5pYTEwZWVMBQGA1UEBwwNU2FuIEZyYW5jaXNjbzENMAsGA1UECgwET2t0YTEU
        MBIGA1UECwwLU1NPUHJvdmlkZXIxEzARBgNVBAMMCMRldi03NzEyMDIxHDAaBgkqhkiG9w0BCQEW
        DWluZm9Ab2t0YS5jb20wHhcNMTgwNTAzMTk0MTI4WmcNMjgwNTAzMTk0MjI4WjCBkjELMAkGA1UE
        BhMCVVMxEzARBgNVBAGMCKNhbg1mb3JuaWEwFjAUBGNVBAcMDVhbiBGcmFuY2lyZ28xDTALBgNV
        BAOMBE9rdGExFDASBgNVBASMC1NTT1Byb3ZpZGVyMRMwEQYDVQQDDApkZXxytNzcxMjAyMRwwGyYJ
        KoZlIhvcNAQkBFglpbmZvZG9rdGEuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
        ugxQGqHAXpjVQZws09n818bFCoEevH3AZbz7568XuQm6MK6h7/O9wB4C5oUYddemt5t2Kc8GRhf3
        BDXX5MVZ8G9AUpG1MSqe1CLV2J96rMnwMIJsKeRr01LYxv/J4kjktpOC389wmcy2fE4RbPoJne
        P4u2b32c2/V7xsJ7UEjPPSD4i812QG6qsUkx3AyNsjo89PekMfm+Iu/dFKXkdjwXZXPxaL0HrNW
        PTpzek8NS5M5rvf8yaD+eElzS0I/HichbPOVvLal0JZyN/f4bp0XJkxZJz6jF5DvBkwIs8/Lz5GK
        nn4XW9Cqjk3equSCJPo5o1Msj8v1LrJYVarqhwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQC26kYe
        LgqjIkF5rvxB2QzTgcd0LvzXOuiVVTZr8Sh5714jJqbDoIgwAQQRxRSQzD/X+hcmhuwdp9s8zPHS
        JagtUJXiypwNtrzbF6M71trWB9sdNrqc99d1gOVRr0Kt5pLTaLe5kkq7dRaQoOIVIJhX9wgynaAK
        HF/SL3mHUytjXggs88AAQa8JH9hEpwG2srN8EsizX6xwQ/p92hM2oLvK5CSMwTx4VBuGod70Eowp
        6TaluRLQh6jCCOCWRuZbbz2T3/sOX+sibC4rLilwfyTkcUopF/bTSdWwknORskK4dBekFcvN9N+C
        p/qaHYcQd6i2vyor888DLHDPXhSKWhpG
        -----END CERTIFICATE-----
```

6. 変更を CR ファイルに保存します。
7. 変更した cr.yaml ファイルを適用します。
コマンドを実行します:

```
kubectl apply -f /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

8. API コンポーネントを再起動して、新しい仕様をロードします。
 - a. Workload Optimization Manager インスタンスへの SSH ターミナルセッションを開きます。
 - b. API コンポーネントを再起動します。

```
kubectl delete api
```
9. 設定が正しいことを確認します。
 - a. Workload Optimization Manager のユーザー インターフェイスに移動します。
認証のため、IdP に自動的にリダイレクトされます。
 - b. 以前に設定した外部グループのメンバーまたは外部ユーザーであるユーザー名でログインします。
 - c. Workload OptimizationManager インスタンスのシステム時刻が正しいことを確認してください。

OpenID 認証の設定

OpenID Foundation によると、「OpenID Connect 1.0 は、OAuth 2.0 プロトコル上の単純な ID レイヤです」。OpenID Connect を使用すると、クライアントは特定の認証サーバーを介してユーザー ID を確認できます。Workload Optimization Manager は、以下のプロバイダーを介して OpenID 認証をサポートします。

- Google
- IBM-MCM
- Okta

OpenID を使用した Workload Optimization Manager へのログイン

Workload Optimization Manager で OpenID を設定すると、指定した OpenID クライアントがプラットフォームによって登録されます。登録された OpenID クライアントを介してログインするには、使用するクライアントを Workload Optimization Manager に指示する URL に手動で移動します。そのクライアントの OpenID ログイン画面にリダイレクトされます。

指定する URL の形式は次のとおりです。

```
https://${hostname}/vmturbo/oauth2/authorization/${openIdClients}
```

それぞれの説明は次のとおりです。

- `${hostname}` は、Workload Optimization Manager のインストールのホストアドレスです。
- `${openIdClients}` は、使用する OpenID プロバイダーのクライアント名です。
これは、OpenID を設定するときに `openIdClients` プロパティとして指定します。

たとえば、Workload Optimization Manager のホストアドレスが 10.10.12.34 で、Okta OpenID クライアントを設定した場合、Workload Optimization Manager のログイン画面が表示されたら、次の場所に移動します。

```
https://10.10.12.34/vmturbo/oauth2/authorization/okta
```

前述の URL に移動すると、OpenID ログイン画面にリダイレクトされるため、単一のユーザーまたはユーザーグループのログイン情報を入力できます。

注：

ユーザーグループを認証するには、そのグループを OpenID プロバイダーで設定し、Workload Optimization Manager でも設定する必要があります。グループ名は、両方の設定で同一である必要があります。

OpenID プロバイダーでは、使用しているクライアントに、ユーザーグループに特定の名前を付与するグループスコープ値を含める必要があります。OpenID 管理者に連絡して、グループ名を取得します。次に、Workload Optimization Manager で、同じ名前を使用するユーザーグループを作成する必要があります。

たとえば、OpenID ID トークンに次のグループクレームが含まれているとします。

```
{
  "sub": "1234567890",
  "name": "My_User_Name",
  "iat": "12121212",
  "groups": "My_Special_User_Group"
}
```

認証に `My_Special_User_Group` グループを使用するには、Workload Optimization Manager に `My_Special_User_Group` という名前のユーザーグループを作成する必要があります。そのグループのメンバーはすべて、そのユーザーグループに割り当てられたロールを取得します。

Workload Optimization Manager での OpenID の設定

OpenID を使用して認証するように Workload Optimization Manager を設定するには、次の手順を実行します。

1. (必須) chrony が設定されており、Workload Optimization Manager インスタンスのシステム時刻が正しいことを確認してください。

手順については、[「時刻の同期」 \(17 ページ\)](#) を参照してください。

2. OpenID プロバイダーから必要なデータを取得します。

セキュリティ管理者に連絡して、プロバイダーからデータを取得します。そのデータを使用して、次の場所にある Workload Optimization Manager CR ファイルで SSO を設定します。

```
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

必要なデータと CR ファイルで宣言するプロパティは、使用する OpenID プロバイダーによって異なります。

■ Google :

CR フィールド :	説明 :
openIdClients	google 認証の実行に使用している OpenID クライアントの名前。
openIdClientId	使用している OpenID クライアントの OAuth2 クライアント識別子。
openIdClientSecret	使用している OpenID クライアントの OAuth2 クライアントシークレット。

■ IBM-MCM :

CR フィールド :	説明 :
openIdClients	ibm 認証の実行に使用している OpenID クライアントの名前。
openIdClientAuthentication	post クライアント認証方式。
openIdUserAuthentication	form ユーザー認証方式。
openIdClientId	使用している OpenID クライアントの OAuth2 クライアント識別子。
openIdClientSecret	使用している OpenID クライアントの OAuth2 クライアントシークレット。
openIdAccessTokenUri	アクセストークンを取得するためにログインプロセスで使用される URI。
openIdUserAuthorizationUri	OpenID Connect の認証エンドポイントへの URI。
openIdUserInfoUri	OpenID Connect UserInfo エンドポイントへの URI。
openIdJwkSetUri	アクセストークンを検証できる JSON Web キーセットを取得するための URI。

■ Okta :

CR フィールド :	説明 :
openIdClients	okta 認証の実行に使用している OpenID クライアントの名前。
openIdClientId	使用している OpenID クライアントの OAuth2 クライアント識別子。

CR フィールド :	説明 :
openIdClientSecret	使用している OpenID クライアントの OAuth2 クライアントシークレット。
openIdAccessTokenUri	アクセストークンを取得するためにログインプロセスで使用される URI。
openIdUserAuthorizationUri	OpenID Connect の認証エンドポイントへの URI。
openIdUserInfoUri	OpenID Connect UserInfo エンドポイントへの URI。
openIdJwkSetUri	アクセストークンを検証できる JSON Web キーセットを取得するための URI。

3. Workload Optimization Manager CR ファイルを設定データで更新します。

これで、OpenID 経由で SSO を設定するために必要なデータが揃いました。Workload Optimization Manager ノードを設定する `cr.yaml` ファイルを編集してから、ノードを展開または再起動します。

- 編集のために CR ファイルを開きます。

シェルで、Workload Optimization Manager VM の `deploy/crds` ディレクトリにディレクトリを変更します。

```
cd /opt/turbonomic/kubernetes/operator/deploy/crds
```

次に、編集のために CR ファイルを開きます。たとえば、VI でファイルを開くには、次のように入力します。

```
vi charts_v1alpha1_x1_cr.yaml
```

ファイルを編集するときに、認証プロバイダーから取得したデータを参照します。

- CR ファイルで、API コンポーネントのエントリに移動します。

CR ファイルで、次のエントリを検索またはスクロールします。

```
apiVersion: charts.helm.k8s.io/v1alpha1
```

`spec:properties:api:` の下で、このコンポーネントの仕様を変更します。

- OpenID 機能をオンにします。

最初の API プロパティについては、次のように設定します。

```
openIdEnabled: true
```

ファイルは次のようになります。

```
apiVersion: charts.helm.k8s.io/v1alpha1 kind: X1
metadata:
  name: x1-release
spec:
  properties:
    api:
      openIdEnabled: true
```

- 認証プロバイダーに関連する OpenId データを入力します。CR ファイルは、使用するプロバイダーに応じて、次の例のようになります。

- Google :

ファイルは次のようになります。

```
apiVersion: charts.helm.k8s.io/v1alpha1
kind: X1
metadata:
```

```

name: xl-release
spec:
  properties:
    api:
      openIdEnabled: true
      openIdClients: google
      openIdClientId: xxxx-4vinrdgllag5p84jjebc6xxxxxx5u.apps.googleusercontent.com
      openIdClientSecret: xxxxxhGcdFEjQa-xxxxxxx
  
```

– IBM-MCM :

ファイルは次のようになります。

```

apiVersion: charts.helm.k8s.io/v1alpha1
kind: Xl
metadata:
  name: xl-release
spec:
  properties:
    api:
      openIdEnabled: true
      openIdClients: ibm
      openIdClientAuthentication: post
      openIdUserAuthentication: form
      openIdClientId: turbonomic-mcm-demo
      openIdClientSecret: "xxxxxxvZ2ZscDhtOFVxxxxxxU3d6cXR4cTZh2xxxxxxRT0K"
      openIdAccessTokenUri: https://icp-console.apps.blue-13.dev.multicloudops.io/idprovider/v1/auth/token
      openIdUserAuthorizationUri: https://icp-console.apps.blue-13.dev.multicloudops.io/idprovider/v1/auth/authorize
      openIdUserInfoUri: https://icp-console.apps.blue-13.dev.multicloudops.io/v1/auth/userInfo
      openIdJwkSetUri: https://icp-console.apps.blue-13.dev.multicloudops.io/oidc/endpoint/OP/jwk
  
```

– Okta

ファイルは次のようになります。

```

apiVersion: charts.helm.k8s.io/v1alpha1
kind: Xl
metadata:
  name: xl-release
spec:
  properties:
    api:
      openIdEnabled: true
      openIdClients: okta
      openIdClientId: xxxxxxxxxxxh1xhQnSKxxxx
      openIdClientSecret: xxxxxxxxxxxtIhVCIRUnhq4xxxxxxxxDdhLdqx0
      openIdAccessTokenUri: https://vmturbo.okta.com/oauth2/v1/token
      openIdUserAuthorizationUri: https://vmturbo.okta.com/oauth2/v1/authorize
      openIdUserInfoUri: https://vmturbo.okta.com/oauth2/v1/userinfo
      openIdJwkSetUri: https://vmturbo.okta.com/oauth2/v1/keys
  
```

4. 変更を CR ファイルに保存します。

5. 変更した cr.yaml ファイルを適用します。

コマンドを実行します:

```
kubectl apply -f /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

6. API コンポーネントを再起動して、新しい仕様をロードします。
 - a. Workload Optimization Manager インスタンスへの SSH ターミナルセッションを開きます。
 - b. API コンポーネントを再起動します。

```
kubectl delete api
```
7. 設定が正しいことを確認します。
 - a. Workload Optimization Manager のユーザー インターフェイスに移動します。
認証のために、認証プロバイダーに自動的にリダイレクトされます。
 - b. 以前に設定した外部グループのメンバーまたは外部ユーザーであるユーザー名でログインします。
 - c. Workload Optimization Manager インスタンスのシステム時刻が正しいことを確認してください。
時刻が同期されていない場合、ブラウザで HTTP Status 401 -authentication failed 例外が発生する可能性があります。
 - d. 設定に失敗した場合は、製品ログで HTTP Status 500 例外を探します。この例外が存在する場合は、CR ファイルで無効なエントリを確認してください。

シングルサインオンの無効化

いずれかの理由で SSO を使用今後使用しない場合、Workload Optimization Manager インストールで無効にできます。シングルサインオンを無効にするには、これらの手順を実行します。

1. SSO の設定を更新して無効にします。
 - a. Workload Optimization Manager インスタンスへの SSH ターミナルセッションを開きます。
 - b. 編集のために CR ファイルを開きます。
シェルで、Workload Optimization Manager VM の deploy/crds ディレクトリにディレクトリを変更します。

```
cd /opt/turbonomic/kubernetes/operator/deploy/crds
```


次に、編集のために CR ファイルを開きます。たとえば、VI でファイルを開くには、次のように入力します。

```
vi charts_v1alpha1_xl_cr.yaml
```
 - c. CR ファイルで、API コンポーネントのエントリに移動します。
CR ファイルで、次のエントリを検索またはスクロールします。

```
apiVersion: charts.helm.k8s.io/v1alpha1
```


spec:properties:api: の下で、このコンポーネントの仕様を変更します。
 - d. SSO 機能をオフにします。
false に設定するエントリは、SAML または OpenID 認証を使用するかどうかによって異なります。
 - SAML 認証:

samlEnabled: プロパティを探して false に設定します。次のように表示されます。

```
samlEnabled: false
```

- OpenID 認証 :

openIdEnabled: プロパティを探して false に設定します。次のように表示されます。

```
openIdEnabled: false
```

- e. 変更を CR ファイルに保存します。
2. API コンポーネントを再起動します。
CR ファイルを編集するために開いたのと同じ SSH ターミナルセッションで、次の手順を実行します。
 - a. root として sudo を使用します。
sudo bash
 - b. API コンポーネントを再起動します。
kubectl delete api
3. 設定が正しいことを確認します。
 - a. Workload Optimization Manager のユーザー インターフェイスに移動します。
認証のため、IdP にはリダイレクトされません。デフォルトの Workload Optimization Manager ログイン画面にリダイレクトされます。
 - b. ローカルのアカウントまたはアクティブ ディレクトリ (AD) アカウントでログインします。



Workload Optimization Manager の新規バージョンへの更新

シスコは、この製品のあらゆる側面を継続的かつ迅速に革新し、改善しています。これは、製品の新しいバージョンを定期的にリリースすることを意味します。新しいバージョンが使用可能かどうかを定期的に確認する必要があります。

新しいバージョンが利用可能になったら、最新バージョンの新しい VM イメージをインストールするだけでなく、既存のインストール済みインスタンスを適切に更新することが重要です。最初に Workload Optimization Manager をインストールしたとき、高度なデータ収集と分析プロセスを行うことができます。インストールの内部には、仮想環境全体のパフォーマンス データを保持する統合データベースが存在します。Workload Optimization Manager は、右サイジング、傾向予測、その他の分析を行うこの履歴データを使用します。つまり、Workload Optimization Manager にとってデータベースは重要であり、時間の経過とともにより重要になっていきます。Workload Optimization Manager のインストールを適切に更新すると、データベースを継続して使用できるようになります。

更新手順を開始する前に、次の手順を実行します。

- Workload Optimization Manager OVA ファイルと ISO イメージへのリンクが記載された、シスコから届いた電子メールがあることを確認してください。
- オンプレミスインストールの場合、VM をホストする物理マシンが最小要件を満たしていることを確認してください ([「最小要件」 \(6 ページ\)](#) を参照)。
- 正しいバージョンの履歴データベースを実行していることを確認してください。
デフォルトの履歴データベースの場合、Workload Optimization Manager は現在、MariaDB バージョン 10.5.12 をサポートしています。このサポートには、Workload Optimization Manager による履歴データベースの使用に関する包括的なテストと品質管理が含まれます。
詳細については、[「MariaDB バージョンの確認」 \(19 ページ\)](#) を参照してください。
- `upgrade-precheck.sh` スクリプトを実行します。
このスクリプトを使用して、Workload Optimization Manager の現在のインストールが更新可能な状態であることを確認できます。インストールの更新に進む前に、このスクリプトを実行することを強くお勧めします ([「更新前の確認」 \(67 ページ\)](#) を参照)。
- ダウンロードした ISO イメージを使用して、オフライン更新を実行します ([「オフライン更新」 \(71 ページ\)](#) を参照)。

更新前の確認

Workload Optimization Manager インスタンスの更新を実行する前に、`upgrade-precheck.sh` スクリプトを実行する必要があります。このスクリプトは、インストールを検査して以下のことをチェックします。

- 十分な空きディスク容量
- オンライン更新の場合、必要なエンドポイント (`index.docker.to`、`github.com` など) へのアクセス権
- MariaDB サービスが実行されている

このチェックは、MariaDB サービスのデフォルトのインストールのみを対象としていることに注意してください。たとえば、MySQL や MariaDB が設定済みの履歴データベースである場合、外部インストールはチェックされないため、MariaDB サービスが実行されていないことが示されます。外部データベースの展開の場合、これは正常な結果です。
- Kubernetes サービスが実行されている
- 必要な Kubernetes 証明書が有効である

証明書が有効でない場合、`kubeNodeCertUpdate.sh` スクリプトを実行して問題を修正できます。このスクリプトは、インストール (`/opt/local/bin`) に配置する必要があります。詳細については、サポート担当者に問い合わせてください。
- ルートパスワードの有効期限が設定されていない
- 時刻同期が有効で、実行中の場合は最新である
- すべての Workload Optimization Manager ポッドが実行されている

このスクリプトを実行するには、次の手順を実行します。

1. スクリプトの最新バージョンをダウンロードします。
 - a. Workload Optimization Manager VM にログインします。

SSH を使用し、Turbonomic アカウントとパスワードを使用して Workload Optimization Manager VM にログインします。
 - b. スクリプトディレクトリに変更します。

```
cd /opt/local/bin
```

- c. スクリプトの最新バージョンを取得します。

サポート担当者に連絡して、このスクリプトの最新バージョンを取得します。
- d. スクリプトを次の場所に保存します。

```
/opt/local/bin/upgrade-precheck.sh
```

- e. スクリプトを実行可能にします。

```
chmod +x upgrade-precheck.sh
```

2. 次のスクリプトを実行します。

```
./upgrade-precheck.sh
```

スクリプトを実行すると、更新を実行する前に対処する必要がある問題が特定されます。

外部 DB と Workload Optimization Manager の更新

外部データベースサーバーを使用して Workload Optimization Manager を展開した場合、更新によっては、その展開用の新しいデータベースとユーザーを手動で作成する必要がある場合があります。この点は、外部データベースサーバーがマルチテナントの場合、または展開で Workload Optimization Manager に管理権限が付与されていない場合に重要です。

注：

外部データベースサーバーがマルチテナントである場合、またはデータベースサーバーで Workload Optimization Manager に管理権限が付与されていない場合は、この設定要件を続行する必要があります。

Azure データベースサービスはマルチテナントです。Azure に外部データベースを展開した場合、この設定要件が適用されます。

新しいデータベースと新しいユーザーを作成するための Workload Optimization Manager 権限を付与する方法でデータベースサーバーを展開した場合、製品の更新によって必要なデータベースが自動的に作成されます。この設定要件は適用されないため、アクションを実行する必要はありません。

一部の Workload Optimization Manager の更新では、更新されたバージョンには、履歴データベースサーバー上の新しいデータベースが含まれています。それらのバージョンのいずれかに更新する場合は、最初に新しいデータベースを作成し、そのデータベースにアクセスする権限を持つユーザーアカウントを作成する必要があります。

次の表は、新しいデータベースが必要な Workload Optimization Manager のバージョンを示しています。前のバージョンからこれらのバージョンに更新する場合は、示されている新しいデータベースを作成する必要があります。たとえば、バージョン 3.0.1 から 3.0.5 に更新する場合は、api データベースを作成する必要があります。

Workload Optimization Manager のバージョン：	新しいデータベース：	注：
3.0.5	api	3.0.5 より前のバージョンから更新する場合は、api という名前の新しいデータベースと、api という名前のユーザーアカウントを作成する必要があります。

注：

これらいずれかのバージョンの Workload Optimization Manager にすでに更新していて、外部 DB を更新する手順を実行していない場合は、サポート担当者に連絡してください。

データベースとユーザーを作成するには、次の手順を実行します。

- 必要な各データベースを手動で作成する

これには、DB インスタンスでのデータベースの作成、データベースにアクセスするためのユーザーの作成、ユーザーへの権限の付与が含まれます。
- 必要な各データベースを cr.yaml ファイルに手動で追加する

cr.yaml ファイルにより、各コンポーネントデータベースのエントリが宣言されます。各エントリはコンポーネントに名前を付け、データベースにアクセスするためにコンポーネントが使用できるユーザーとパスワードを提供します。新しいデータベースごとに新しいエントリを追加する必要があります。

新しいデータベースを作成するには、次の手順を実行します。

1. グローバルアカウントを使用して外部 DB に接続します。

アカウントには、データベースとユーザーを作成する権限が必要です。cr.yaml ファイルで dbRootUsername を指定した場合は、そのアカウントを使用できます。

2. データベースを作成します。<New_Database> は前述の表のデータベース名と一致します。

```
create database <New_Database>;
```

たとえば、新しい api データベースを作成するには、次を実行します。

```
create database api;
```

3. Workload Optimization Manager がデータベースにアクセスするために使用するアカウントを作成します。<New_Database> は前述の表のデータベース名と一致します。

```
create user '<New_Database>'@'%' identified by 'vmturbo';
```

たとえば、api データベースのユーザーを作成するには、次を実行します。

```
create user 'api'@'%' identified by 'vmturbo';
```

注：

値 vmturbo は、Workload Optimization Manager がすべてのコンポーネント データベース アカウントに使用するデフォルトのパスワードです。別のログイン情報を使用して手動でアカウントを作成した場合は、このデータベースに対しても同様に作成できます。

4. 新しいユーザーアカウントのユーザーアカウント権限を設定します。<New_Database> は前述の表のデータベース名と一致します。

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EVENT, TRIGGER ON <New_Database>.* TO '<New_Database>'@'%';
```

たとえば、api ユーザーのアカウント権限を設定するには、次を実行します。

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EVENT, TRIGGER ON api.* TO 'api'@'%';
```

5. 権限を有効にするために権限をフラッシュします。

```
flush privileges;
```

外部 DB サービスに新しいデータベースが作成されたので、そのデータベースへのアクセスを Workload Optimization Manager cr.yaml リソースに宣言する必要があります。

1. 編集のために .cr ファイルを開きます。ファイルの場所は、設定している Workload Optimization Manager インストールのタイプによって異なります。

Workload Optimization Manager の VM イメージのインストール：	Kubernetes ノードまたはノードクラスタ上の Workload Optimization Manager：
Workload Optimization Manager インスタンスの SSH ターミナルセッションを開きます。 Workload Optimization Manager のインストール時に設定したシステム管理者でログインします。	編集のために次のファイルを開きます。 deploy/crds/charts_v1alpha1_xl_cr.yaml

Workload Optimization Manager の VM イメージのインストール :	Kubernetes ノードまたはノードクラスタ上の Workload Optimization Manager :
<ul style="list-style-type: none"> ■ ユーザー名 : turbo ■ パスワード : [your_private_password] <p>次のファイルを編集します。</p> <pre>/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml</pre>	

2. 一致するポッドのログイン情報を追加して、新しいデータベースにアクセスします。

cr.yaml ファイルの `properties:` セクションにエントリを追加します。`vmturbo` はそのユーザーアカウントに割り当てたパスワード、`yourDB` は外部 DB またはマルチテナント DB パーティションの修飾名、`<New_Database>` は新しいデータベースの名前です。追加するエントリで次のように宣言します。

```
<New_Database>:
  <New_Database>DbUsername: <New_Database>@yourDB
  <New_Database>DbPassword: vmturbo
```

たとえば、api データベースを追加した場合、結果の cr.yaml ファイルは次のようになります。

```
properties:
  global:
    enableSecureDBConnection: true
    sqlDialect: MYSQL
    dbRootPassword: yourAdminPassword
    dbRootUsername: xladmin@yourDB
    #dbUserPassword:
    #dbUsername:
  action-orchestrator:
    actionDbUsername: action@yourDB
    actionDbPassword: yourPassword
  auth:
    authDbUsername: auth@yourDB
    authDbPassword: yourPassword
  clustermgr:
    clustermgrDbUsername: clustermgr@yourDB
    clustermgrDbPassword: yourPassword
  cost:
    costDbUsername: cost@yourDB
    costDbPassword: yourPassword
  group:
    groupComponentDbUsername: group_component@yourDB
    groupComponentDbPassword: yourPassword
  history:
    historyDbUsername: history@yourDB
    historyDbPassword: yourPassword
  plan-orchestrator:
    planDbUsername: plan@yourDB
    planDbPassword: yourPassword
  topology-processor:
    topologyProcessorDbUsername: topology_processor@yourDB
    topologyProcessorDbPassword: yourPassword
  repository:
```

```

repositoryDbUsername: repository@yourDB
repositoryDbPassword: yourPassword
market:
  marketDbUsername: market@yourDB
  marketDbPassword: yourPassword
api:
  apiDbUsername: api@yourDB
  apiDbPassword: yourPassword

```

完了したら、最新バージョンの Workload Optimization Manager に更新できます（アップグレードにより、このファイルのバージョン情報に変更が適用されることに注意してください）。

オフライン更新

Workload Optimization Manager インストールのオフライン更新を実行するには、次の手順を実行します。

1. 現在の Workload Optimization Manager VM のスナップショットを保存します。

更新する前に、Workload Optimization Manager VM を適切にシャットダウンする必要があります（電源オフではありません）。シャットダウンするには、次のように入力します。

```
sudo init 0
```

次に、スナップショットを実行（または VM のクローンを作成）します。これにより、更新中に問題が発生した場合に、信頼性の高い復元ポイントが確保できます。スナップショットを作成したら、VM をオンラインに戻します。

2. ISO イメージをダウンロードしてマウントします。

Workload Optimization Manager Software Download ページ

(<https://software.cisco.com/download/home/286328879/type/286317011/release>) に移動して、最新の ISO 更新イメージへのリンクを確認します。

Workload Optimization Manager を実行する VM で使用できる場所に ISO イメージをダウンロードし、CD ドライブとしてマウントします。

たとえば、vCenter Server で Workload Optimization Manager VM を実行する場合は、次の手順を実行します。

- a. vCenter で、Workload Optimization Manager VM に移動します。
 - b. VM を右クリックして、[Edit Settings] を選択します。
 - c. [CD/DVD Drive] ドロップダウンメニューで、次の手順を実行します。
 - i. [Datastore ISO] を選択します。
 - ii. Workload Optimization Manager の更新 ISO イメージを参照して、選択します。
 - d. [Connect at power on] チェックボックスがオンになっていることを確認します。
3. Workload Optimization Manager インスタンスへの SSH ターミナルセッションを開きます。
- 現在の Workload Optimization Manager VM のスナップショットまたはクローンを作成したら、SSH セッションを開きます。Workload Optimization Manager のインストール時に設定したシステム管理者でログインします。

- ユーザー名 : turbo
- ユーザー名 : [your_private_password]

4. 更新バージョンの isoUpdate.sh スクリプトを取得します。

このスクリプトは、Workload Optimization Manager を実行する VM の /opt/local/bin/ に保存する必要があります。

[Cisco Software Central](#) から isoUpdate.sh スクリプトをダウンロードします。

5. スクリプトを実行可能にします。

```
chmod +x /opt/local/bin/isoUpdate.sh
```

6. オフラインインストールを実行します。

```
/opt/local/bin/isoUpdate.sh
```

スクリプトの実行時、次の手順を実行します。

- インストール内の古いスクリプトをバックアップします。
- インストールの設定とコード資産を更新します。
- プラットフォームを新しいバージョンに更新します。
- カスタムリソースを更新します。
- MariaDB 構成を更新します。
- 組み込みレポートまたはデータエクスポートを有効にしている場合は、組み込みレポートおよびデータ エクスポート データベース (Postgres および TimescaleDB) をインストールします。
- t8c-operator および Workload Optimization Manager コンポーネントをスケールダウンします。
- このバージョンの最終更新を実行します。
- t8c-operator をスケールアップしてから、Workload Optimization Manager コンポーネントを再起動します。

7. Workload Optimization Manager アプリケーションが正しくインストールされていることを確認します。

スクリプトによるプラットフォームの更新が完了したら、すべてのコンポーネントが再起動するまで待つ必要があります。

アプリケーションのインストールを確認するには、次のコマンドを実行します。

```
kubectl get pods -n turbonomic
```

すべてのポッドが起動すると、各ポッドの READY 列に 1/1 と表示され、STATUS 列に Running と表示されます。

以下のような出力が表示されます。

NAME	READY	STATUS	RESTARTS
action-orchestrator-b6454c9c8-mfl185	1/1	Running	0
api-7887c66f4b-shndq	1/1	Running	0
arangodb-7f646fc5fc-zhcwf	1/1	Running	0
auth-5b86976bc8-vxwz4	1/1	Running	0
clustermgr-85548678d9-r5wb8	1/1	Running	0
consul-7f684d8cb8-6r677	1/1	Running	0
cost-5f46dd66c4-6d6cb	1/1	Running	0
extractor-5f41dd61c4-4d6lq	1/1	Running	0
group-5bfdfbc6f8-96bsp	1/1	Running	0
history-5fc7fbc855-6zslq	1/1	Running	0
kafka-74cc77db94-dfrbl	1/1	Running	0
market-5f54699447-z4wkm	1/1	Running	0
mediation-actionscript-57b4fc6df-4lzfz	1/1	Running	0
mediation-appdynamics-6d65f8766f-kb441	1/1	Running	0
mediation-hpe3par-d7c475c4c-v8ftc	1/1	Running	0
mediation-hyperv-6bd8c94df5-4dbzx	1/1	Running	0
mediation-netapp-7f8fc955d9-4kkdl	1/1	Running	0
mediation-oneview-7dbd7b54cf-7rfqp	1/1	Running	0
mediation-pure-58c4bd8cd9-8n256	1/1	Running	0
mediation-ucs-6f4bb9889-9rnqk	1/1	Running	0
mediation-vcenter-5bc4f5fbd4-nzm4j	1/1	Running	0
mediation-vcenterbrowsing-5c5987f66c-bfjq4	1/1	Running	0
mediation-vmax-6c59969b89-28t9j	1/1	Running	0

mediation-vmm-9c4878cf9-rfxnl	1/1	Running	0
nginx-5b775f498-sm2mm	1/1	Running	0
plan-orchestrator-6dfc4c9b6-p5t5n	1/1	Running	0
reporting-b44fbdfb4-8fjv5	1/1	Running	0
repository-6d555bb4bf-fxldh	1/1	Running	0
rsyslog-fd694878c-5tb2c	1/1	Running	0
t8c-operator-558bcc758d-5h8mp	1/1	Running	0
topology-processor-b646b786b-9skp7	1/1	Running	0
zookeeper-5f65b5bf69-nnmbt	1/1	Running	0

8. ブラウザのデータを消去し、ブラウザを更新してください。

ブラウザのデータを消去してブラウザを更新すると、Workload OptimizationManager の機能に完全にアクセスできるようになります。ただし、現在の分析データに依存する機能は、完全なマーケット サイクル（通常は10分）が経過するまで利用できません。たとえば、保留中のアクションのチャートには、完全なマーケットサイクルが終了するまでアクションは表示されません。

9. 必要に応じて、使用環境で新しいプローブを有効にします。

注：

Workload Optimization Manager の進化に伴い、提供されるプローブのセットは変わる可能性があります。また、バージョンごとに、デフォルトで有効になっているプローブのセットは変わる可能性があります。ただし、新しいバージョンに更新しても、プローブ設定が変更されることはありません。新しいバージョンに更新しても、展開内の新しいプローブは自動的に有効になりません。更新時に新しいプローブを利用する場合は、手動で有効にする必要があります。

更新されたバージョンで新しいプローブを有効にする手順については、[「プローブコンポーネントの有効化と無効化」\(51 ページ\)](#)を参照してください。次の手順を使用してプラットフォームの cr.yaml ファイルを編集しますが、kubectl を介した変更の適用はまだしないでください。

10. 他のユーザーに、ブラウザ データを消去し、Workload OptimizationManager ブラウザ セッションを更新するように通知します。



付録：IdP の一般的な設定

シングルサインオン (SSO) の構成を開始する前に、IdP が SSO 用に設定されていることを確認する必要があります。IdP を設定するときに役立つ可能性のあるパブリック OktaIdP の一般的な設定を次に示します。

SAML 設定：全般	
設定	例
シングルサインオン URL (<hostname> は、Workload Optimization Manager が実行されるホスト、<samlRegistrationID> は、SSO プロバイダーから取得した登録 ID)	https://<hostname>/vmturbo/saml2/sso/<samlRegistrationID>
受信者の URL (<hostname> は、Workload Optimization Manager が実行されるホスト、<samlRegistrationID> は、SSO プロバイダーから取得した登録 ID)	https://<hostname>/vmturbo/saml2/sso/<samlRegistrationID>
宛先 URL (<hostname> は、Workload Optimization Manager が実行されるホスト、<samlRegistrationID> は、SSO プロバイダーから取得した登録 ID)	https://<hostname>/vmturbo/saml2/sso/<samlRegistrationID>
聴衆の制限	urn:test:turbo:markharm
デフォルトのリレー状態	
名前 ID の形式	未指定
アプリケーション ユーザー名	Okta で管理されているアカウントのユーザー名
応答	署名済み
アサーション署名	署名済み
署名アルゴリズム	RSA_SHA256
デジタル アルゴリズム	SHA256
アサーションの暗号化	非暗号化
SAML シングル ログアウト	イネーブル

SAML 設定 : 全般	
設定	例
シングルログアウト URL (<hostname> は、Workload Optimization Manager が実行されるホスト)	https://<hostname>/vmturbo/rest/logout
SP 発行者	turbo
署名証明書	Example.cer (CN=apollo)
authnContextClassRef	PasswordProtectedTransport
オナー強制認証	対応
SAML 発行者 ID	http://www.okta.com/\$(org.externalKey)



付録：段階的なプラットフォーム展開

プラットフォームをホストする Workload Optimization Manager VM をインストールしたら、次の手順でプラットフォーム コンポーネントをインストールできます。

1. 必要に応じて、このインストール用にシングルサインオン認証 (SSO) を設定します。

Workload Optimization Manager ユーザーの認証に SSO を使用する予定の場合は、ここで設定できます。SSO を設定するには、`charts_v1alpha1_xl_cr.yaml` ファイルを編集します。インストール完了前に今すぐ編集するか、後で編集して影響を受けるコンポーネントを再起動することができます。詳細については、[「シングルサインオン認証」 \(55 ページ\)](#) を参照してください。

2. Workload Optimization Manager Kubernetes ノードを展開します。

Workload Optimization Manager を Kubernetes に展開するときは、Workload Optimization Manager コンポーネントを実行するポッドをホストする VM として 1 つの Kubernetes ノードを展開します。Kubernetes ノードを展開して初期化するスクリプトでは、Workload Optimization Manager アプリケーションを構成する Kubernetes ポッドも展開されます。

turbo ユーザーとして Workload Optimization Manager VM でセキュアセッション (SSH) を開始し、次の手順を実行します。

- a. Kubernetes ノードを初期化し、ポッドを展開します。

```
sudo /opt/local/bin/t8cInstall.sh スクリプトを実行します。
```

スクリプトの完了までに最大 20 分かかります。

- b. 展開が成功したことを確認します。

スクリプトの出力の最後にある要約セクションで、エラーが報告されていないことを確認します。エラーが報告されている場合は、Workload Optimization Manager サポートに連絡してください。

- c. Workload Optimization Manager アプリケーションが正しくインストールされていることを確認します。

アプリケーションのインストールを確認するには、次のコマンドを実行します。

```
kubectl get pods -n turbonomic
```

すべてのポッドが起動すると、各ポッドの READY 列に 1/1 と表示され、STATUS 列に Running と表示されます。

以下のような出力が表示されます。

NAME	READY	STATUS	RESTARTS
action-orchestrator-b6454c9c8-mf185	1/1	Running	0
api-7887c66f4b-shndq	1/1	Running	0
arangodb-7f646fc5fc-zhcwf	1/1	Running	0
auth-5b86976bc8-vxwz4	1/1	Running	0
clustermgr-85548678d9-r5wb8	1/1	Running	0
consul-7f684d8cb8-6r677	1/1	Running	0

cost-5f46dd66c4-6d6cb	1/1	Running	0
extractor-5f41dd61c4-4d61q	1/1	Running	0
group-5bfdfbc6f8-96bsp	1/1	Running	0
history-5fc7fbc855-6zslq	1/1	Running	0
kafka-74cc77db94-dfrbl	1/1	Running	0
market-5f54699447-z4wkm	1/1	Running	0
mediation-actionscript-57b4fc6df-4lzfz	1/1	Running	0
mediation-appdynamics-6d65f8766f-kb441	1/1	Running	0
mediation-hpe3par-d7c475c4c-v8ftc	1/1	Running	0
mediation-hyperv-6bd8c94df5-4dbzx	1/1	Running	0
mediation-netapp-7f8fc955d9-4kkdl	1/1	Running	0
mediation-oneview-7dbd7b54cf-7rfqp	1/1	Running	0
mediation-pure-58c4bd8cd9-8n256	1/1	Running	0
mediation-ucs-6f4bb9889-9rnqk	1/1	Running	0
mediation-vccenter-5bc4f5fbd4-nzm4j	1/1	Running	0
mediation-vccenterbrowsing-5c5987f66c-bfjq4	1/1	Running	0
mediation-vmax-6c59969b89-28t9j	1/1	Running	0
mediation-vmv-9c4878cf9-rfxnl	1/1	Running	0
nginx-5b775f498-sm2mm	1/1	Running	0
plan-orchestrator-6dfffc4c9b6-p5t5n	1/1	Running	0
reporting-b44fbdfb4-8fjv5	1/1	Running	0
repository-6d555bb4bf-fxldh	1/1	Running	0
rsyslog-fd694878c-5tb2c	1/1	Running	0
t8c-operator-558bcc758d-5h8mp	1/1	Running	0
topology-processor-b646b786b-9skp7	1/1	Running	0
zookeeper-5f65b5bf69-nnmbt	1/1	Running	0

d. システムクロックを同期します。

データを正しく表示し、シングルサインオン (SSO) 認証をサポートするには、システムクロックを同期する必要があります。詳細については、[「時刻の同期」 \(17 ページ\)](#) および [「シングルサインオン認証」 \(55 ページ\)](#) を参照してください。

e. ロードバランサが正しくインストールされていることを確認します。

ロードバランサの存在を確認するには、次のコマンドを実行します。

```
kubectl get services -n turbonomic | grep LoadBalancer
```

以下のような出力が表示されます。

```
Nginx LoadBalancer 10.10.10.10 10.10.10.11 443:32669/TCP,80:32716/TCP 17h
```

f. メディエーションを設定します。

インストールスクリプトにより、メディエーションプロンプのデフォルトセットが自動的に有効になります。インストールが完了したら、有効なメディエーションプロンプのセットを変更できます ([「プロンプコンポーネントの有効化と無効化」 \(51 ページ\)](#) を参照)。

Workload Optimization Manager で IT 環境を管理するには、検出を実行してアクションを実行できるように、環境内のターゲットに Workload Optimization Manager を接続する必要があります。検出とアクションの実行プロセスの組み合わせがメディエーションです。このリリースの Workload Optimization Manager は、以下のターゲットを介したメディエーションをサポートします。リストにない追加のターゲットを使用する必要がある場合は、Workload Optimization Manager サポートに連絡してください。

■ アプリケーションとデータベース

- Apache Tomcat 7.x、8.x および 8.5.x
- AppDynamics 4.1+

- ApplInsights
- Dynatrace 1.1+
- IBM WebSphere Application Server 8.5+
- Instana
- JBoss Application Server 6.3+
- JVM 6.0 +
- Microsoft SQL Server 2012、2014、2016、2017、および 2019
- MySQL 5.6.x および 5.7.x
- NewRelic
- Oracle 11g R2 および 12c
- Oracle WebLogic 12c
- クラウドネイティブ
 - Kubernetes
 - OpenShift 3.3 +
- ファブリックとネットワーク
 - Cisco UCS Manager 3.1+
 - HPE OneView 3.00.04 +
- ゲスト OS プロセス
 - SNMP
 - WMI : Windows バージョン 2019、2016、2012/2012 R2、2008 R2、10、8/8.1、および 7
- ハイパーコンバージド
 - Hyperflex 2.5、2.6
 - Nutanix Community Edition
 - VMware vSAN
- ハイパーバイザ
 - Citrix XenServer 5.6. x および 6.x
 - Microsoft Hyper-V 2008 R2、Hyper-V 2012/2012 R2、Hyper-V 2016、Hyper-V 2019
 - VMware vCenter 6.0、6.5、6.7、および 7.0+
- オーケストレータ
 - Action Script
 - ServiceNow
- プライベート クラウド
 - Microsoft System Center 2012/2012 R2 Virtual Machine Manager および System Center 2016 Virtual Machine Manager
- パブリック クラウド
 - Amazon AWS
 - Amazon AWS Billing
 - Microsoft Azure Service Principal
 - Microsoft Enterprise Agreement
- ストレージ
 - EMC ScaleIO 2.x および 3.x
 - SMI-S 8.1 + を使用した EMC VMAX

- 仮想ボリュームと LUN の 1:1 マッピングを搭載した EMC VPLEX ローカルアーキテクチャ
- EMC XtremIO XMS 4.0 +
- ONTAP 8.0 + を使用した NetApp Cmode/7mode (AFF および SolidFire を除く)
- Pure Storage F-シリーズおよび M-シリーズアレイ
- Virtual Desktop Infrastructure
 - VMware Horizon

これらのターゲットの詳細については、Workload Optimization Manager ターゲット設定ガイド [英語] を参照してください。

3. Workload Optimization Manager のユーザーインターフェイスにログインし、管理者ユーザーアカウントのパスワードを設定します。

重要事項：

すべてのプラットフォーム コンポーネントが起動して稼働し、完全に準備が整ってから、最初のログインをしてください。すべてのコンポーネントの準備が整う前に、プラットフォームにライセンスやターゲットを追加しようとすると、プラットフォームが正しく初期化されないことがあります。詳細については、[「Workload Optimization Manager アプリケーションが正しくインストールされていることを確認する」 \(76 ページ\)](#) を参照してください。

コンポーネントが起動したら、Web ブラウザで Workload Optimization Manager VM の静的 IP アドレスを入力します。Workload Optimization Manager ユーザーのログインページにリダイレクトされます。

Workload Optimization Manager には、ADMINISTRATOR ロールを持つ administrator という名前のデフォルトのユーザーアカウントが含まれています。初めてログインするときは、そのアカウントに独自のパスワードを設定する必要があります。ADMINISTRATOR ロールを持つ他のアカウントの作成や削除は可能ですが、Workload Optimization Manager のインストールには、ADMINISTRATOR ロールを持つアカウントが常に少なくとも 1 つ必要です。

ログインページで、必要に応じて情報を入力し、メモしておきます。

- USERNAME : administrator のデフォルトのログイン情報を使用します。
- PASSWORD のパスワードを入力します。
新しいパスワードは、強力なパスワードポリシー（大文字と小文字、数字、記号の組み合わせ）に準拠している必要があります。新しいパスワードは誰にも教えないでください。
- パスワードをもう一度入力して、[REPEAT PASSWORD] を確認します。
- [CONFIGURE] をクリックします。

これは、Workload Optimization Manager のユーザーインターフェイスに管理者権限でアクセスするために使用するアカウントです。ユーザーインターフェイス管理者アカウントのログイン情報は安全な場所に保存してください。

注：

最初のログインは、常に管理者アカウントで行います。これは管理ユーザーアカウントです。VM 自体のシェルセッションにログインするために以前に設定した Workload Optimization Manager システム管理者アカウントと、このアカウントを混同しないでください。

4. 管理者としてログインしたら、他のユーザーアカウントを作成し、さまざまなロールを付与できます。ユーザーアカウントとロールの詳細については、Workload Optimization Manager ユーザーガイド [英語] を参照してください。



付録：段階的なオフライン更新

Workload Optimization Manager インストールの段階的なオフライン更新を実行するには、次の手順を実行します。

1. 現在の Workload Optimization Manager VM のスナップショットを保存します。
更新する前に、Workload Optimization Manager VM を適切にシャットダウンする必要があります（電源オフではありません）。シャットダウンするには、次のように入力します。

```
sudo init 0
```


次に、スナップショットを実行（または VM のクローンを作成）します。これにより、更新中に問題が発生した場合に、信頼性の高い復元ポイントが確保できます。スナップショットを作成したら、VM をオンラインに戻します。
2. ISO イメージをダウンロードし、Workload Optimization Manager を実行する VM にアタッチします。
Workload Optimization Manager OVA ファイルおよび ISO イメージへのリンクについては、シスコから受信した電子メールを参照してください。
3. vCenter にログインして ISO イメージをマウントします。
 - a. vCenter で、Workload Optimization Manager VM に移動します。
 - b. VM を右クリックして、[Edit Settings] を選択します。
 - c. [CD/DVD Drive] ドロップダウンメニューで、次の手順を実行します。
 - i. [Datastore ISO] を選択します。
 - ii. Workload Optimization Manager の更新 ISO イメージを参照して、選択します。
 - d. [Connect at power on] チェックボックスがオンになっていることを確認します。
4. Workload Optimization Manager VM にログインします。
SSH を使用し、Turbonomic アカウントとパスワードを使用して Workload Optimization Manager VM にログインします。
5. ISO イメージをマウントします。
Type:

```
sudo mount /dev/cdrom /mnt/iso
```
6. 正しいバージョンの ISO イメージがマウントされていることを確認します。
Type:

```
ls /mnt/iso
```


ISO イメージに更新用の正しいバージョンが含まれていることを確認します。
7. 最新の Docker イメージをロードします。
Type:

```
sudo /mnt/iso/turboinstall.sh
```


このスクリプトは、すべてのイメージを Workload Optimization Manager インスタンスにロードします。ロードが成功すると、次のようなメッセージが表示されます。

```
The t8c upgrade iso has been mounted
Image check:

=====
*****
All images have been loaded
*****
```

ロードが成功しなかった場合、ロードされなかったすべてのイメージと、イメージを手動でロードするための手順が表示されます。

- 以下のコマンドを実行して、Workload Optimization Manager を更新します。

```
■ /mnt/iso/turboupgrade.sh | tee \
  /opt/turbonomic/t8c_upgrade_$(date +%Y-%m-%d_%H_%M_%S).log
```

スクリプトが終了するまで待ちます。

- 正しいバージョンの MariaDB を実行していることを確認します。

このバージョンの Workload Optimization Manager は、MariaDB バージョン 10.5.12 をサポートします。この Workload Optimization Manager バージョンに更新した後でも、インストールで以前のバージョンの MariaDB が実行されている可能性があります。

まだ SSH セッションにいる間に、MariaDB のバージョンを確認します。

```
mysql -u root --password=vmturbo -e "SHOW VARIABLES LIKE 'version';"
```

出力は次のようになります。

```
+-----+-----+
| Variable_name | Value           |
+-----+-----+
| version       | 10.5.12-MariaDB |
+-----+-----+
```

MariaDB のバージョンが 10.5.12 より前の場合は、MariaDB を更新する必要があります。完全な手順と情報については、[「MariaDB バージョンの確認」 \(19 ページ\)](#) を参照してください。

- ISO イメージをアンマウントします。

次のコマンドを入力します。

```
sudo umount /dev/cdrom
```

- ブラウザのデータを消去し、ブラウザを更新してください。

ブラウザのデータを消去してブラウザを更新すると、Workload OptimizationManager の機能に完全にアクセスできるようになります。ただし、現在の分析データに依存する機能は、完全なマーケット サイクル (通常は10分) が経過するまで利用できません。たとえば、保留中のアクションのチャートには、完全なマーケットサイクルが終了するまでアクションは表示されません。

- 必要に応じて、使用環境で新しいプローブを有効にします。

注：

Workload Optimization Manager の進化に伴い、提供されるプローブのセットは変わる可能性があります。また、バージョンごとに、デフォルトで有効になっているプローブのセットは変わる可能性があります。ただし、新しいバージョンに更新しても、プローブ設定が変更されることはありません。新しいバージョンに更新しても、展開内の新しいプローブは自動的に有効になりません。更新時に新しいプローブを利用する場合は、手動で有効にする必要があります。

更新されたバージョンで新しいプローブを有効にする手順については、[「プローブコンポーネントの有効化と無効化」\(51 ページ\)](#)を参照してください。次の手順を使用してプラットフォームの `cr.yaml` ファイルを編集しますが、`kubectl` を介した変更の適用はまだしないでください。

13. Workload Optimization Manager アプリケーションが正しくインストールされていることを確認します。

アプリケーションのインストールを確認するには、次のコマンドを実行します。

```
kubectl get pods -n turbonomic
```

すべてのポッドが起動すると、各ポッドの READY 列に 1/1 と表示され、STATUS 列に Running と表示されます。

14. 他のユーザーに、ブラウザ データを消去し、Workload OptimizationManager ブラウザ セッションを更新するように通知します。



付録：Workload Optimization Manager の Classic から XL への移行

シスコは、Workload Optimization Manager プラットフォームの XL ファミリを紹介できることを誇りに思います。

Workload Optimization Manager の XL ファミリは、Classic バージョンファミリと同じ基準でアプリケーションのパフォーマンスを管理するための新しいプラットフォームです。より大規模な環境を管理できるコンポーネントベースのアーキテクチャと、アプリケーションの管理とパフォーマンスを強化するサプライチェーンとユーザーインターフェイスの拡張機能が導入されています。

これらの変更が存在するため、XL バージョンファミリに移行する必要がありますが、同じファミリの新しいバージョンに更新する場合のように、単純な更新は実行できません。移行することで、ターゲット、グループ、ポリシー、およびその他のデータが Classic インストールから XL インストールに転送されます。

シスコは、移行を自動的に実行するツールを提供しています。Classic インストールと XL インストールの IP アドレスをツールで設定します。次に、移行できるデータを検出し、検出したデータを XL インストールにロードします。現在、次の情報が移行されます。

- ターゲットの設定
- ポリシーとスケジュール
- テンプレート
- カスタムグループ
- ポリシーとターゲットのスコープ設定に使用される検出されたグループ
- ユーザーおよびユーザーグループ（ローカルおよび Active Directory ユーザーアカウントを含む）

注：

このマニュアルの作成時点では、すべてのターゲットタイプの移行はテストされていません。これは、テストラボでは一部のターゲットタイプへのアクセスが制限されているためです。

制限事項

移行ツールには次の制限があることに注意してください。

- 常に XL の四半期リリースに移行する

シスコでは、Workload Optimization Manager の四半期リリースと隔週リリースを提供していますが、各四半期リリースに対して移行スクリプトをテストしています。

Classic から XL に移行するときは、常に XL の最新の四半期リリースに移行する必要があります。後続の隔週リリースを使用する場合は、最初に四半期リリースに移行し、移行完了後に隔週リリースに更新します。

- Classic インストールの複数のインスタンスを単一の XL インストールに移行しないでください。単一の Classic インスタンスのみを単一の XL インスタンスに移行する必要があります。Classic の複数のインスタンスを移行する場合は、サポート担当者に連絡してください。
- AWS ターゲットの場合、XL インスタンスは別のターゲットを使用して課金情報を管理します。このツールでは、Classic AWS ターゲットが移行されますが、XL インスタンスに個別の課金ターゲットは作成されません。AWS 環境の場合、XL に移行後、適切な AWS Billing ターゲットを手動で設定する必要があります。
- このツールでは、次のものは移行されません。
 - 現在のライセンス
 - 履歴データ
 - ダッシュボードとチャート
 - 計画
 - 配置と予約
 - 課金情報とコストの設定
 - アクションスクリプトとオーケストレーションの設定
 - 電子メールおよびトラップ通知の設定
 - レポートおよびレポートテンプレート
 - データ保存の設定
 - 電子メールサーバーの設定
 - HTTP プロキシの設定
 - SSO の設定
 - ログインレベル
 - サポートオプション

Workload Optimization Manager インスタンスの準備

移行する前に、移行用に Classic インスタンスと XL インスタンスを準備する必要があります。各インスタンスは正しいバージョンに更新する必要があり、セキュアシェルセッション (SSH) 経由でアクセスできる必要があります。

- Classic インスタンスを 2.3.18 以降に更新する
ツールを使用するには、Classic インスタンスが 2.3.18 以降である必要があります。
- Workload Optimization Manager の最新の四半期リリースをインストールする
これが XL インスタンスになります。このインスタンスをネットワーク上の目的の場所にインストールします。移行するバージョンは 3.0.1 以降である必要があります。

注 :

XL インスタンスをインストールするときは、次の点を考慮してください。

- 常に XL の四半期リリースに移行する

シスコでは、Workload Optimization Manager の四半期リリースと隔週リリースを提供していますが、各四半期リリースに対して移行スクリプトをテストしています。

Classic から XL に移行するときは、常に XL の最新の四半期リリースに移行する必要があります。後続の隔週リリースを使用する場合は、最初に四半期リリースに移行し、移行完了後に隔週リリースに更新します。

- 新しくインストールした XL インスタンスに移行することをお勧めします。ツールを使用して、ターゲット、検出済みのエンティティとグループ、またはポリシーがすでにある XL インスタンスに Classic インスタンスを移行する場合は注意が必要です。
- 複数の Classic インスタンスを単一の XL インスタンスに移行する必要がある場合は、サポート担当者に連絡してください。
- Workload Optimization Manager の両方のインスタンスにアクセスするためのアドレスとログイン情報を収集します。次のことを確認してください。
 - SSH (PuTTY) と Classic インスタンスの UI の両方にアクセスするために使用される IP アドレス
 - Classic インスタンスの SSH ログイン情報
 - XL インスタンスの UI へのアクセスに使用される IP アドレス
 - Classic インスタンスと XL インスタンスの両方の UI 管理者のログイン情報
 - Workload Optimization Manager の Classic および XL インスタンスのユーザーインターフェイスを介して、管理者ユーザーアカウントのログイン情報があることを確認する
- XL インスタンスに有効なライセンスがあることを確認してください。

XL バージョンの有効なライセンスがない場合、移行ツールは実行できません。

- Workload Optimization Manager インスタンスへのアクセスを再確認します。次のことができることを確認します。

- Classic インスタンスでセキュアシェルセッション (SSH) にログインする
- Classic インスタンスから XL インスタンスの REST API にアクセスする

このアクセスをテストするには、Classic インスタンスで SSH にログインし、次のコマンドを実行します。

```
curl -k https://XL_INSTANCE_IP/vmturbo/rest/cluster/isXLEnabled && echo
```

XL REST API で接続できる場合、コマンド出力は `true` になります。

- 管理者アカウントを使用して、Workload Optimization Manager の Classic インスタンスと XL インスタンスの両方の UI にログインする
- Classic 環境に Kuberbo を展開した場合は、XL 用の Kuberbo を展開します。

ツールでは、Kuberbo 展開は移行されません。移行プロセスのこの段階は、XL インストールで動作するように Kuberbo を展開するのに最適なタイミングです。詳細については、<https://github.com/turbonomic/kuberbo> にある Kuberbo github リポジトリを参照してください。

Kuberbo が検出するトポロジに依存するグループまたはポリシーを (Classic インスタンスで) 宣言した場合は、XL インスタンスでそれらを手動で作成する必要があります。XL インスタンスでは、コンテナ インフラストラクチャに対してさまざまなエンティティタイプが作成されることに注意してください。関連付けられたグループとポリシーでは、この新しいトポロジモデルが使用されます。

- 使用環境内のすべてのターゲットが Valid 状態であることを確認してください。

移行ツールにより、Classic インスタンスで設定したターゲットと一致するように、XL インスタンスでターゲットが設定されます。移行を実行する前に、すべてのターゲットが検証されている必要があるため、問題がある場合は、問題を修正するか、ターゲットを削除してください。

検証されていないターゲットが見つかった場合、失敗エラーが通知され、そのターゲットは移行されません。

移行前にターゲットを削除した場合、またはターゲットの移行が失敗した場合、そのターゲットをサポートするために Workload Optimization Manager で検出されたグループは移行に含まれません。その結果、それらの検出されたグループのスコープに依存するポリシーがある場合、そのスコープは存在しなくなります。ポリシーが特定のターゲットのスコープをどのように使用するのが理解し、ターゲットが削除されたために空のスコープを持つポリシーをクリーンアップまたは削除する必要があります。

- Classic および XL インスタンスの設定に効果的な凍結を課します。

移行プロセスを開始したら、XL インスタンスや Classic インスタンスの設定に変更を加えないでください。

このルールには、次の 2 つの例外があります。

- 状況によっては、特定の手順を中止して変更を加える必要がある状況が発生する場合があります。ツールから設定の変更を指示されたら、変更できます。
たとえば、XL インスタンスで現在有効になっていないターゲットタイプを移行する場合、移行を中止し、XL インスタンスでターゲットを有効にして、移行を再開するように指示されます。
- 同様に、Classic インスタンスに Kubeturbo を展開した場合は、XL インスタンスに Kubeturbo を手動で展開する必要があります。手順には、その実行プロセスの手順が含まれています。

移行ツールのインストール

移行を実行するには、Workload Optimization Manager の Classic インスタンスにツールをインストールします。

1. ツールをダウンロードします。

アクセスとダウンロードの手順については、<https://github.com/turbonomic/tbmigrate/blob/master/DOWNLOAD.md> を参照してください。[Downloading TbMigrate] セクションまでスクロールします。

注：

移行ツールの最新バージョンを必ずダウンロードしてください。

2. 「PuTTY」または別の SSH クライアントを使用して Classic インスタンスにログインします（まだログインしていない場合）。
3. アーカイブファイルを Workload Optimization Manager インスタンスに展開します。次のコマンドを実行します。
 - `cd $HOME`
 - `tar xvfz /tmp/tbmigrate-VERSION_NUMBER.tgz --no-same-owner`

移行の実行

移行を実行するには、移行ツールの [Interactive] メニューを開き、コマンドを順番に実行します。

注：

Classic から XL への移行ツールを使用する場合は、手順を順番に実行する必要があります。Migrate Targets 手順の後に他の手順を実行し、その後、Migrate Targets 手順を再度実行するために後戻りすると、Migrate Targets の最初の実行後に実行した手順の記録がすべて失われます。

次の手順に進む前に、完了した各手順の結果を注意深く確認することをお勧めします。そうすることで、後戻りする必要がなくなります。

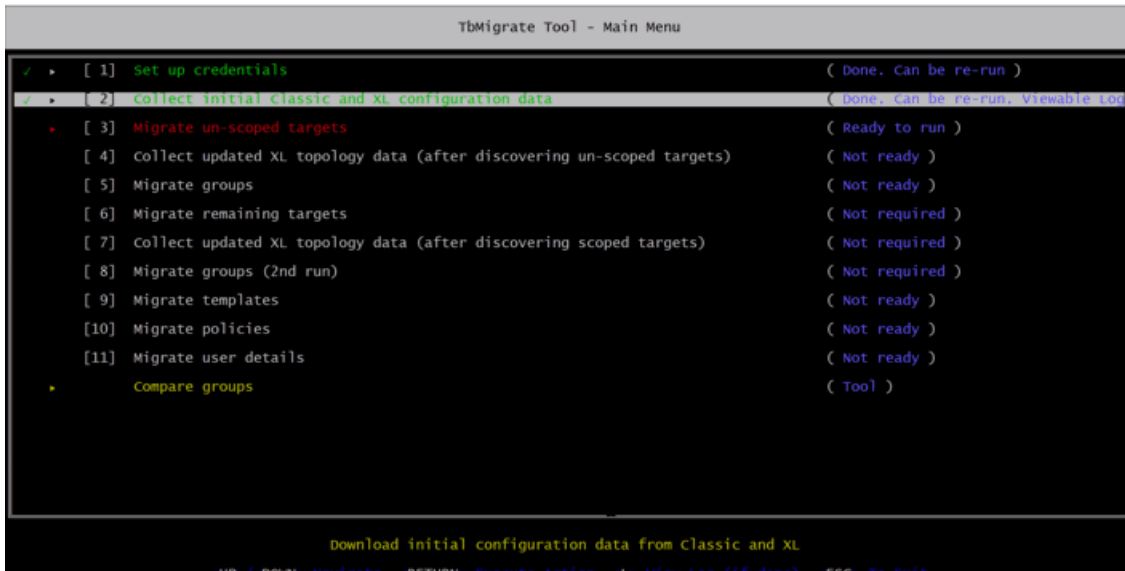
たとえば、移行の初めから終わりまでツールを実行した場合、Migrate Targets を再度実行するために、ターゲットの移行がツールによってどのように計算されたかを確認したいとします。その後、ツールによって移行が完了していないと判断された場合、Migrate Targets 以降の手順を確認することはできません。当該手順を実行した直後に Migrate Targets の結果を詳しく確認していれば、戻って再度実行する必要はありません。

移行ツールの [Interactive] メニューを開く

Classic インスタンスにログインしたまま、移行ツールの場所へ移動し、[Interactive] メニューを起動します。次のコマンドを実行します。

- `cd $HOME/tbmigrate`
- `sh menu.sh`

シェルウィンドウに [Interactive] メニューが表示されます。



```

Tbmigrate Tool - Main Menu
├─ [ 1] Set up credentials ( Done, Can be re-run )
├─ [ 2] Collect initial Classic and XL configuration data ( Done, Can be re-run, Viewable Log )
├─ [ 3] Migrate un-scoped targets ( Ready to run )
├─ [ 4] Collect updated XL topology data (after discovering un-scoped targets) ( Not ready )
├─ [ 5] Migrate groups ( Not ready )
├─ [ 6] Migrate remaining targets ( Not required )
├─ [ 7] Collect updated XL topology data (after discovering scoped targets) ( Not required )
├─ [ 8] Migrate groups (2nd run) ( Not required )
├─ [ 9] Migrate templates ( Not ready )
├─ [10] Migrate policies ( Not ready )
├─ [11] Migrate user details ( Not ready )
└─ [ *] Compare groups ( Tool )

Download initial configuration data from Classic and XL
UP / DOWN: Navigate, RETURN: Execute Action, L: View Log (if done), ESC: To Exit
  
```

メニューには、実行する必要がある移行手順が順番に表示されます。メニューの内容：

- 各手順のステータスは次のとおりです。
 - [Ready to run] : 手順を順番に実行する必要があります。
 - [Not ready] : この手順を実行する前に、前の手順を実行する必要があります。
 - [Done] : 完了した一部の手順を再実行できます。完了した一部の手順については、その手順に移動してログを表示できます。
 - [Not required] : データを再度移行または収集するための任意の手順。
- 上下の矢印を使用して、さまざまな手順に移動します。
- 選択した手順について、次のことができます。
 - メニューコマンドの実行 : Return を押して、選択した手順を実行します。
 - 関連するログの表示 : 「L」キーを押して、完了した手順のログを表示します。
 - 手順の終了 : Esc を押して、手順を終了します。

移行コマンドの実行

注：

移行コマンドを実行すると、\$HOME/tbmigrate/logs ディレクトリの .log ファイルに出力が書き込まれます。

ログを注意深く調べて、ツールで完全に移行できなかった要素による悪影響がないことを確認する必要があります。ログは移行が完了したら確認する必要があり、移行プロセスのさまざまな段階で確認できます。

移行を実行するには、メニューコマンドを順番に実行します。次の手順では、各メニューコマンドについて説明します。

1. ログイン情報を設定します。

この手順では、両方のインスタンスのログイン情報を設定します。コマンドを実行し、ツールに必要な情報を提供します。ログイン情報の設定の一環として、この手順では、[Target Password Migration] オプションが表示されます。

- [YES] : 移行ツールは暗号化されたターゲットパスワードを収集して使用し、XL インスタンスに移行するターゲットを設定および検証します。これらのパスワードは、ターゲット移行フェーズのさまざまな段階で入力する必要はありません。
- [NO] : 移行ツールは暗号化されたパスワードを収集しません。パスワードを移行しない場合、ターゲット移行プロセスのさまざまな段階でパスワードの入力を求めるプロンプトが表示されます。

2. 初期の Classic および XL 設定データを収集します。

Classic インストールから現在の設定を収集します。

この手順では、インスタンスから設定データを収集します。移行プロセスでは、収集されたデータが使用されます。

この手順が完了すると、実行したアクションのログが生成されます。ログを調べて、エラーがないことを確認する必要があります。

3. スコープ外のターゲットを移行します。

この手順では、グループにスコープが設定されていないすべてのターゲットを移行します。

スコープターゲットは、Workload Optimization Manager がスコープを定義するグループを作成した後にのみ移行できます。Workload Optimization Manager は、スコープ外のターゲットが設定および検証され、ターゲットのエンティティが検出されるまで、それらのグループを作成できません。後の手順でスコープターゲットを移行します。

この手順を実行すると、ターゲットの対話型リストが表示されます（少し遅れて表示される場合があります）。このリストには、移行できるターゲット（左側にチェックマーク付き）と、移行できないターゲット（X マーク付き）が表示されます。このリストを使用して、ターゲットを移行します。

注 :

すでにターゲットがインストールされている XL インスタンスにターゲットを移行する場合、重複するターゲットを作成しないようにする必要があります。重複は、XL ターゲットの 1 つが対話型リストのターゲットと同じ場合に発生することがあります。

リストにより重複が識別され、ターゲット名が大文字と小文字を含めて同一である場合、ターゲットの移行が無効になります。ただし、名前前のスペルが何らかの方法で異なる場合、重複を識別できません。

すでにターゲットが実行されている XL インスタンスに移行する場合は、このリストを注意深く調べて、リストに重複するターゲットがないことを確認してください。

■ リスト内を移動します。

上下の矢印キーを使用してターゲット間を移動し、各ターゲットの画面の下部に表示されるメッセージを表示します。

サポートするメディアエーションポッドを XL インスタンスで有効にする必要があるターゲットをメモします。後に XL インスタンスでそれらのプローブを有効にしてから、ターゲットの移行手順を再度実行する必要があります。

移行しないターゲットについては、そのエントリに移動し、スペースキーを押して選択を解除します。

■ ターゲットのリストを終了します。

移行するターゲットの表示と選択が終了したら、Esc キーを押してリストを終了します。

このツールには、移行を続行するための [Yes/No] オプションがあります。ターゲットの移行を続行するか中止するかを決定する必要があります。

- 不足しているプローブを XL インスタンスにインストールする必要がある場合は、今すぐインストールします。移行を中止するには、[No] を選択します。次に、XL インスタンスでセッションを開始して、メモした不足しているプローブをインストールします。手順については、インストールガイドを参照してください。不足しているプローブを XL インスタンスにインストールした後、Classic インスタンスに戻り、Migrate Unscoped Targets コマンドを再度実行します。
- 不足しているプローブを XL インスタンスにインストールする必要がない場合は、移行を続行します。[はい (Yes)] を選択します。リストで選択したターゲットが移行されます。

- ターゲットの移行でエラーが報告されないことを確認します。
- ツールの出力が「There are some scoped targets to be migrated later」メッセージで終了しているかどうかをメモします。

移行されたターゲットが検証されて、ターゲットのトポロジが検出されるのを待ちます。

ユーザーインターフェイスにログインすることで、XL インスタンスでの検出を監視できます。または、Classic インスタンスの SSH から次のコマンドを使用して、XL インスタンスのサプライチェーンとターゲットステータスのレポートを確認できます。

- `bin/tbutil @xl list supplychain`: サプライチェーンを表示する
- `bin/tbutil @xl list targets -l`: ターゲットのステータスを一覧表示する

「watch」ツールを使用して、これらのツールのいずれかを繰り返し実行できます。次に例を示します。

```
watch -n 10 bin/tbutil @xl list supplychain
```

「watch」セッションが終了したら、Ctrl+C を押してコマンドを停止します。

移行が完了するのを待つためのヒント：

- メッセージ `WOOPS! Supplychain object not found` は、エンティティが検出されなかったことを示します。1 つ以上のターゲットが検証されたと仮定すると、これは、検出でまだトポロジが生成されていないことを意味します。
- より多くのターゲットで検出が完了するに伴い、表示される数値は時間の経過とともに増加する可能性があるため、すべての検出が完了するまで待つ必要があります。
- 「Minor」、「Major」、および「Critical」列の数値はすべてゼロである必要があります。これは、アクションが XL インスタンスに対してグローバルに無効化されているため、システムにアクションが存在しないためです。その新しいインスタンスに管理を切り替える際に、XL インスタンスのアクションを有効にします。
- XL インスタンスでターゲット検証の問題が見つかった場合は、XL ユーザーインターフェイスにログインして、ターゲットの設定に関するログイン情報やその他の設定を変更できます。その後、影響を受けるターゲットの再検証と再検出をトリガーできます。

すべてのターゲットが検証され、検出が完了したら、次の手順に進むことができます。

4. 更新された XL トポロジデータを収集します。

元の「収集」手順（手順 2）を実行してから、XL インスタンスのトポロジが変更されているため、ここで「収集」手順を繰り返す必要があります。手順を実行し、完了したらログを調べてエラーがないことを確認します。

5. グループを移行します。

この手順で、カスタムグループを移行します。また、スコープ外のターゲットから作成できるスコープグループも移行します。そうすることで、スコープターゲットの最終的な移行が可能になります。

この手順では、検出されたほとんどのグループは移行されないことに注意してください。XL インスタンスは、独自のルールに従ってグループを検出します。

手順を実行し、完了したらログを調べてエラーがないことを確認します。

6. 残りのターゲットを移行します。

前の「スコープ外のターゲットの移行」手順で「There are some scoped targets to be migrated later」メッセージが表示された場合は、ここで移行する必要があります。「There are no outstanding scoped targets」というメッセージが表示された場合は、次のいくつかの手順をスキップして、「テンプレートの移行」にジャンプできます。

この手順の実行は、スコープターゲットを移行することを除いて、「スコープ外のターゲットの移行」の手順と同じです。移行するターゲットの同じ対話型リストが表示されます。また、XL インスタンスにインストールする必要があるプローブも識別されます。詳細については、前述の「スコープ外のターゲットの移行」手順を参照してください。

注：

すでにターゲットがインストールされている XL インスタンスにターゲットを移行する場合、重複するターゲットを作成しないようにする必要があります。重複は、XL ターゲットの 1 つが対話型リストのターゲットと同じ場合に発生することがあります。

リストにより重複が識別され、ターゲット名が大文字と小文字を含めて同一である場合、ターゲットの移行が無効になります。ただし、名前がスペルが何らかの方法で異なっている場合、重複を識別できません。

すでにターゲットが実行されている XL インスタンスに移行する場合は、このリストを注意深く調べて、リストに重複するターゲットがないことを確認してください。

簡単に言えば、スコープターゲットを移行するには、手順を実行してから、次の手順を実行します。

- ツールからの質問に答えます。上下の矢印キーを使用して、リストに移動します。スペースキーを押して、リスト内のターゲットを選択または選択解除します。
- サポートするメディエーションポッドを XL インスタンスで有効にする必要があるターゲットをメモします。この移行手順を終了し、XL インスタンスでそれらのプローブを有効にしてから、このターゲット移行手順を再度実行する必要があります。この移行手順を中止するには、Esc を押してから、[No] を選択します。
- 不足しているプローブを XL インスタンスにインストールする必要がない場合は、移行を続行します。Esc を押して、[Yes] を選択します。リストで選択したターゲットが移行されます。
- ターゲットの移行でエラーが報告されないことを確認します。

移行されたターゲットが検証されて、ターゲットのトポロジが検出されるのを待ちます。

ユーザーインターフェイスにログインすることで、XL インスタンスでの検出を監視できます。または、Classic インスタンスの SSH から次のコマンドを使用して、XL インスタンスのサプライチェーンとターゲットステータスのレポートを確認できます。

- `bin/tbutil @xl list supplychain`: サプライチェーンを表示する
- `bin/tbutil @xl list targets -l`: ターゲットのステータスを一覧表示する

7. 更新された XL トポロジデータを収集します。

XL インスタンスのトポロジが再び変更されたため、ここで「収集」手順を繰り返す必要があります。手順を実行し、完了したらログを調べてエラーがないことを確認します。

8. グループを移行します。

スコープターゲットに関連する追加のグループを移行します。

この手順により、スコープターゲットによって検出された新しいトポロジを使用するようにグループが更新されます。手順を実行し、完了したらログを調べてエラーがないことを確認します。

9. テンプレートを移行します。

この手順により、Classic インスタンスで定義したテンプレートが移行されます。平均的なクラスタテンプレートなどの生成されたテンプレートは移行されず、検出されたテンプレートも移行されません。手順を実行し、完了したらログを調べてエラーがないことを確認します。

10. ポリシーを移行します。

この手順では、配置ポリシーと、デフォルトおよびカスタムの自動化ポリシーを移行します。

自動化ポリシーの場合、この手順で、次のすべての条件を満たす自動化設定が Classic から XL にコピーされます。

- Classic インスタンスの設定が変更されました。つまり、設定の値はデフォルト値とは異なっています。
- XL インスタンスにも同等の設定があります。Classic のすべての設定が XL に複製されているわけではないことに注意してください。一方、XL では、Classic に存在しない新しい設定が導入されています。

この手順では、配置ポリシーも移行します。

移行中に、XL に移行できない（またはその逆の）設定が Classic で見つかった場合、適切な警告がログに書き込まれます。次に例を示します。

- XL サプライチェーンにマッピングされないエンティティタイプのアクションポリシー。たとえば、Classic エンティティのアプリケーションは、XL のアプリケーション コンポーネント エンティティです。
 - 設定が一致しない。たとえば、Classic ではビジネスアプリケーションの SLA キャパシティを設定できますが、XL にその設定が含まれていません。
 - 別のメカニズムによって指定された設定。たとえば、アクションスクリプトの設定は、XL では異なる方法で指定されます。
- 手順を実行し、完了したらログを調べてエラーがないことを確認します。

11. ユーザーの詳細を移行します。

この手順では、次のものが移行されます。

- ローカルユーザー：移行には、ユーザーのパスワードを除くすべてのユーザーの詳細（ロール、スコープなど）が含まれます。移行により、XL インストールの各ユーザーに対して任意の複雑なパスワードが作成されます。Workload Optimization Manager 管理者は、各ローカルユーザーアカウントを編集し、個々のユーザーと協力してパスワードを手動で更新できます。
- LDAP ユーザー：移行には、すべてのユーザーの詳細が含まれます。
- LDAP ユーザーグループ：移行には、すべてのグループの詳細が含まれます。
- Active Directory 設定：移行には、Classic インストールで指定した AD 設定が含まれます。

これで移行プロセスは完了です。

ツールの出力を調べて、エラーがないことを確認する必要があります。エラーがログに記録されている場合は、エラーを確認して、XL インスタンスで実行できる修正アクションを決定できます。出力は `$HOME/tbmigrate/logs` ディレクトリの `.log` ファイルに書き込まれます。

前述のすべての手順が完了すると、XL インスタンスは Classic インスタンスと可能な限り類似するように設定されますが、設定が期待どおりであることを確認する必要があります。特に、前述の移行手順のいずれかでログに記録された警告とエラーを確認してください。それらの警告やエラーが使用環境に影響を与える可能性があるかどうかを検討する必要があります。必要に応じて、XL ユーザーインターフェイスにログインして、修正できます。

新しいバージョンへの切り替え

移行が完了したら、Workload Optimization Manager の Classic バージョンから XL バージョンへの変更を計画する必要があります。

移行を「落ち着かせる」

移行プロセスを実行すると、XL インスタンスが使用環境を検出し、グループを作成して、市場分析を開始します。ただし、アクションは XL インスタンスに対してグローバルに無効になっています。これは重要な点で、同じ環境を同時にアクティブに管理する 2 つのインスタンスを持つことはできません。XL インスタンスでアクションを無効にすることで、準備が整うまで、移行プロセスでアクションが実行されないようになります。

一定期間、アクションを無効にして XL インスタンスを実行したままにしておくことをお勧めします。アクションは、Classic インスタンスと並行して実行させ、Classic インスタンスで有効にしておく必要があります。このようにして、Classic インスタンスが使用環境を管理し続けながら、XL インスタンスが追いつくようにします。

- XL インスタンス（アクションが無効）：このインスタンスは、Workload Optimization Manager 分析がサイズ変更アクションを生成するために使用する履歴データを作成します。
- Classic インスタンス（アクションが有効）：このインスタンスは、XL インスタンスが「落ち着く」間、引き続き環境を管理します。

この時間を使用して、XL インスタンスが使用率と容量のメトリックを正しく検出していることを確認し、ポリシーやその他のビジネスルールを設定できます。これは、ツールで移行されなかった環境設定に対処する良い機会でもあります。たとえば、前述の「制限事項」セクションを参照して、ツールで移行されなかった項目に対処できます。

注：

一般に、移行が落ち着くまで 30 ~ 90 日かかります。これは、XL インスタンスがクラウドコストとパーセントイルベースの分析を適切に処理するために必要な履歴データを収集するのに十分な日数です。

XL インスタンスへの管理の移行

XL インスタンスが十分な履歴データを収集し、XL インスタンスの設定が正しいことを確認したら、使用環境の管理を Classic から XL に切り替えることができます。

このプロセスには 2 つの手順があり、順序が重要です。

1. Classic インスタンスで [Disable All Actions] をオンにして、すべてのアクションがクリアされるのを待ちます。
[Settings] > [Policies] に移動します。次に、[Policies] リストの [Search] ボックスに *global* と入力して、[Global Defaults] ポリシーを検索します。クリックしてそのポリシーを編集します。[ACTION CONSTRAINTS] で、[Disable All Actions] オプションをオンにします。[SAVE AND APPLY] をクリックします。
アクションを無効にしたら、[Pending Actions] リストがクリアされ、アクションが表示されなくなるまで待ちます。
2. XL インスタンスで [Disable All Actions] をオフにします。
[Global Defaults] ポリシーに移動し、[Disable All Actions] オプションをオフにします。XL インスタンスでアクションの生成が開始され、アクションを使用して環境を管理できます。

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。

リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動 / 変更されている場合がありますことをご了承ください。

あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。