Workload Optimization Manager 3.4.6
インストールガイド

目次

概要

Workload Optimization Manager、クラウドおよびリモート対応環境用アプリケーションリソース管理（ARM）向けプレミアソリューション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。

よろしくお願いいたします。

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。	■ 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 OptimizationManager 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>-<XXXXXXXXXXXXXX>.ova` ファイルは、

   <version> が Workload Optimization Manager のバージョン番号であり、<XXXXXXXXXXXXXX> が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. [メモリ（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構成手順を実行するには、「全般 構成タスク」 *（16 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」 *（73 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 OptimizationManager 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 を有効にします。

   Workload Optimization Manager インスタンスの構成には、1 つの NIC が含まれますが、これは有効化されていないか、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、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構成手順を実行するには、「全般 構成タスク」 *（16 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」 *（73 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」 *（73 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、展開対象の Kubernetes ホスト名を変更するには、段階的なインストールを実行する必要があります。

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」（16 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」（8 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。

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。
   - turbo ユーザーとして Workload Optimization Manager VM でセキュアセッション（SSH）を開始します。
   - スクリプト ディレクトリに変更します：

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

   - 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。

   コンポーネントが起動したら、この静的 IP アドレスを Web ブラウザに入力して、Workload Optimization Manager ユーザー インターフェイスのログイン ページにアクセスします。

   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。

   ```
   ipsetup スクリプトを実行してネットワークをセットアップしましたか？[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 を構成します。

   プラットフォーム VM の静的 IP をすでに構成している場合は、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。

   ```
   ------------------------------------------------
   古い IP アドレス：10.0.2.15
   新しい IP アドレス：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」（8 ページ）または「VHD：vCenter イメージのインストール」（10 ページ）、これを行うには、すでに ipsetup.sh スクリプトを実行している必要があります。

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。

     ```
     クラスタの問題をさらにデバッグおよび診断するには、「kubectl cluster-info dump」を使用します。
     ```

     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。

注：
セキュリティ上の理由から、これらのアカウントにはパスワードを構成することをお勧めします。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タ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。

```
############################
展開をロールアウトし、開始します
############################
```

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。

```
インストール プロセスが完了し、すべてのコンポーネントが起動するのを待ちます。
** スクリプトは 30 分間待機します。**
```

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。

```
================================================================================
  1 つ以上の展開がまだ起動していません。
  ** 環境が安定するまで、さらに 30 分お待ちください。**
  コンポーネントのステータスを確認するには、次のコマンドを実行します。
  kubectl get pods
  一部のコンポーネントがまだ準備ができていない場合は、サポート担当者に連絡してください
  展開の準備ができていません:。
```

スクリプトは次に、kubectl get pods コマンドのフォーマットされた結果を表示します。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。

```
################################################
展開が完了しました。UI からログインしてください
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。

```
kubectl 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。

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、**[パスワードの再入力（REPEAT PASSWORD）]** を確認します。
- **[アカウントの作成（Create Account）]** 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 アカウントであり、Workload Optimization Manager システム管理者アカウントではありません。

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。

注：
セキュリティ上の理由から、Workload Optimization Manager インストールのメイン管理者として機能する別の管理者ユーザーアカウントを作成してから、デフォルトの管理者 アカウントを削除することができます。ただし、*管理者権限を持つ少なくとも 1 つのユーザー アカウントが常に必要であることを忘れないでください。*

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、次の構成タスクを実行する必要があります。

- （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。
- （重要）MariaDB バージョンを確認してください。
- （オプション）使用可能なディスク容量を増やします。
- （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介し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）データエクスポータを有効にします。
- （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 をインストールしたときに設定したシステム管理者でログインします。

   - ユーザ名：`turbo`
   - ユーザ名：`[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。

   2019 年 2 月 2 日(木) 14:25:45 UTC

   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。

```
          現地時間： Fri 2019-12-06 21:09:26 UTC
          世界時間： Fri 2019-12-06 21:09:26 UTC
         RTC 時間： Fri 2019-12-06 21:09:27
     タイム ゾーン： UTC (UTC, +0000)
        NTP 有効： yes
     同期済み NTP： yes
ローカル TZ の RTC： no
    DST アクティブ： 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構成する必要がある場合、
chrony on the server instance.

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。

   2019 年 12 月 6 日金曜日 21:09:26 UTC

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。

```
          現地時間： Fri 2019-12-06 21:09:26 UTC
          世界時間： Fri 2019-12-06 21:09:26 UTC
           RTC 時間 ：Fri 2019-12-06 21:09:27
      タイム ゾーン： UTC (UTC, +0000)
         NTP 有効： yes
      同期済み NTP： yes
ローカル TZ の RTC：    no
    DST アクティブ： 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で、Workload Optimization Manager は現在、MariaDB バージョン 10.5.16 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 10.5.16 を使用する必要があります。Workload Optimization Manager のバージョンを（初めてインストールするのではなく）更新する場合は、インストールで正しいバージョンの MariaDB を使用していることを確認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 10.5.16 以降を実行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 10.5.16 での動作が完全にテストされています。

*既知の問題のため、MariaDB バージョン 10.5.14、10.5.15、10.6.7、10.7.3、または 10.8.2 は**使用しない**でください。*

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。

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 10.5.16 を含む Workload Optimization Manager の最初のリリースは 3.3.6 です。最初に旧バージョンをインストールしている場合、MariaDB を明示的に 10.5.16 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。

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。そのような展開では、データベースのバージョニングを自分で管理する必要があります。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 10.5.16 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。

（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 10.5.16 を使用することをお勧めします。ダウンロード パッケージは https://archive.mariadb.org/mariadb-10.5.16/yum/centos7-amd64 にあります。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ニ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 をインストールしたときに設定したシステム管理者でログインします。

   - ユーザ名：turbo
   - ユーザ名：[your_private_password]

2. MariaDB のバージョンを確認します。

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

出力は次のようになります。

```
+---------------+-----------------+
| Variable_name | Value           |
+---------------+-----------------+
| version       | 10.5.16-MariaDB |
+---------------+-----------------+
```

バージョンが 10.5.16-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 10.5.16-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 10.5.16 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 をインストールしたときに設定したシステム管理者でログインします。

   - ユーザ名：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」 (66 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、MariaDB 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）。

   ```
   ================================
   mariadb 構成を更新します。
   ================================
   合計メモリ：128773 MB
   Innodb バッファー プール サイズの変更：9216 MB
   最大許容パケット数の変更：1G
   innodb ログ ファイル サイズの変更：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 10.5.16 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=my_pwd -e "SHOW VARIABLES LIKE 'version';"
   ```

   出力は次のようになります。

   ```
   +---------------+-----------------+
   | Variable_name | Value           |
   +---------------+-----------------+
   | version       | 10.5.16-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。

   ```
   ##############################################################################
   mariadb がアップグレードされ、適切に機能していることを確認したら、次を実行します。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」（39 ページ）を参照してください。

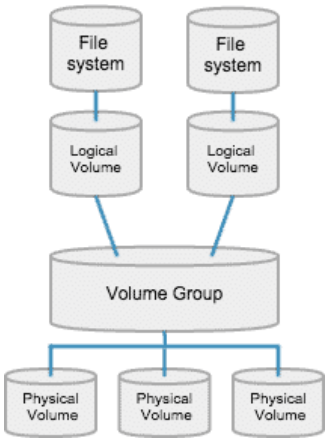
実行する手順の概要は次のとおりです。

- 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 をインストールしたときに設定したシステム管理者でログインします。

- ユーザ名：turbo
- パスワード：[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 のような名前で表示されます。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
   - プライマリ
   - サイズを確認
   - タイプを 8E に変更
   - 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。

   ```
   pvcreate /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。

```
フリー PE / サイズ          128000 / 500.00 GiB
```

この例では、128000 が lv を拡張する量です。たとえば、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 である場合を仮定します。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 をインストールしたときに設定したシステム管理者でログインします。
   - ユーザ名：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+7QAAAAIAAAABAAAAAgAFY2VydDEAAAF5H2lEigAFWC41MDkAAAYQMIIGDDCCBPSgAwIBAgIT
    HAAAARHIFJdLbG90sAAAAAABETANBgkqhkiG9w0BAQUFADBcMRMwEQYKCZImiZPyLGQBGRYDY29t
    MRcwFQYKCZImiZPyLGQBGRYHdm10dXJibzEUMBIGCgmSJomT8ixkARkWBGNvcnAxFjAUBgNVBAMT
    DWNvcnAtREVMTDEtQ0EwHhcNMjEwNDA4MDM0OTEyWhcNMjIwNDA4MDM0OTEyWjAhMR8wHQYDVQQD
    ExZkZWxsMS5jb3JwLnZtdHVyYm8uY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
    sCXuh2MTrFERyU/aKgdbgyjLezNuwF6nmZveZUhDaJDpfLHJlzhwfyYRTGfSSusVo4polJS4WqPZ
    T3Zk8f2IaX04RpfpQErq5N3uY/BxFkATWLMDiquSd0Di798k2diYXAxXvzMmfmIkBBYJta9oztum
    uXyh/42dXOGznQ5fFuxosgAksZ6CnXGDKrTBlb0bHpST1z1Pdg+fJ+f9Tq7IffOYdVbuedFTwsik
    Z0JgDCIRrmmsOJphiHdBqJ6ZLdbSeEzBIbboiQs81pAELw7V0ZZUfKV6y8+zMTACGwpVPJSFv7LX
    RlW1TWcqhXVAOmroe2WcU8KJE6XZTBxp7z7dzwIDAQABo4IDADCCAvwwLwYJKwYBBAGCNxQCBCIe
    IABEAG8AbQBhAGkAbgBDAG8AbgB0AHIAbwBsAGwAZQByMB0GA1UdJQQWMBQGCCsGAQUFBwMCBggr
    BgEFBQcDATAOBgNVHQ8BAf8EBAMCBaAweAYJKoZIhvcNAQkPBGswaTAOBggqhkiG9w0DAgICAIAw
    DgYIKoZIhvcNAwQCAgCAMAsGCWCGSAFlAwQBKjALBglghkgBZQMEAS0wCwYJYIZIAWUDBAECMAsG
    CWCGSAFlAwQBBTAHBgUrDgMCBzAKBggqhkiG9w0DBzBCBgNVHREEOzA5oB8GCSsGAQQBgjcZAaAS
    BBDswjlHut/nQZ0uK2aUglGbghZkZWxsMS5jb3JwLnZtdHVyYm8uY29tMB0GA1UdDgQWBBR6M7Hb
    BiirpjIXQ3PXXScB8LkmRDAfBgNVHSMEGDAWgBRjs9l3el7SuKUDMlrHHRhBkENgaDCB0QYDVR0f
    BIHJMIHGMIHDoIHAoIG9hoG6bGRhcDovLy9DTj1jb3JwLURFTEwxLUNBLENOPWRlbGwxLENOPUNE
    UCxDTj1QdWJsaWMlMjBLZXklMjBTZXJ2aWNlcyxDTj1TZXJ2aWNlcyxDTj1Db25maWd1cmF0aW9u
    LERDPWNvcnAsREM9dm10dXJibyxEQz1jb20/Y2VydGlmaWNhdGVSZXZvY2F0aW9uTGlzdD9iYXNl
    P29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlvblBvaW50MIHHBggrBgEFBQcBAQSBujCBtzCBtAYI
    KwYBBQUHMAKGgadsZGFwOi8vL0NOPWNvcnAtREVMTDEtQ0EsQ049QUlBLENOPVB1YmxpYyUyMEtl
    eSUyMFNlcnZpY2VzLENOPVNlcnZpY2VzLENOPUNvbmZpZ3VyYXRpb24sREM9Y29ycCxEQz12bXR1
    cmJvLERDPWNvbT9jQUNlcnRpZmljYXRlP2Jhc2U/b2JqZWN0Q2xhc3M9Y2VydGlmaWNhdGlvbkF1
    dGhvcml0eTANBgkqhkiG9w0BAQUFAAOCAQEADP6OYLONkZ2j6gaBdfdoIJtvn1g1qXTsRrtFuUcF
    C9mUxL0G5Tudr0VlyEnLH2wtj10CGsIi54+aPGYiElXijThEe1WTHaO2hklRLdNrM8KxUp3tUNb/
    cP4d+EYt297wVWgxpl9MStiND8+7M2+65daoEu5IOLtq4lC7YlCSXay19N5HdiGBHV5L07PTZ261
    qDzShSb0ZWtG7++5VkqveVEIfs3hUYdaItz0Zu6sym90aUcvn5wohV1GPPqGDvVCg5Kf50hsZfmy
    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。

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

```
# JAVA_OPTS を認証 POD に渡して、次のような追加オプションを設定します。
# LDAPS（セキュア LDAP）の AD 証明書のトラストストア
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。

（オプション）Turbonomic UI を保護するための証明書の追加

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 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/ターボノミック

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 openssl genrsa -out myPrivate.key 2048 です。

   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

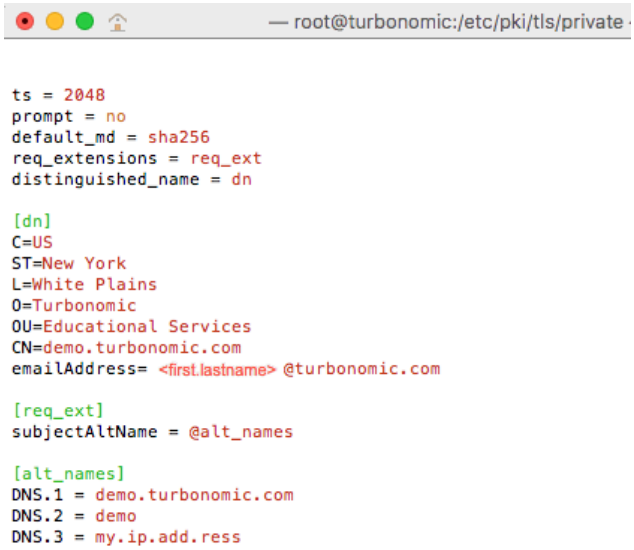
[alt_names]
DNS.1 = <FQDN>
DNS.2 = <server's short name>
DNS.3 = <server's IP address>
```

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` の値はオプションです。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。



6. 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。

7. 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
```

8. 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。

9. 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：

- `qDzShSb0ZWtG7++5VkqveVEIfs3hUYdaItz0Zu6sym90aUcvn5wohV1GPPqGDvVCg5Kf50hsZfmy ltNlaqiiqLMnYVMa93CkpFFjoP9gmGFJky0yTfh6G8HuqbI7guddDsUqMQTT3uv3EBwSYeImOya7 Zye5C4NnsAfnx8kOwXdsVERC`
- `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。

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

   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_xl_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。
      <UID> がポッド インスタンスに対して生成された識別子の場所に次のコマンドを実行します。

```
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。

(オプション) プローブ用の追加の CA 証明書の追加

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。

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、構成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。ここに記載されていないプローブに対してセキュア アクセスを構成する必要がある場合は、サポート担当者にお問合せください。

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。証明書ファイルがない場合は、ネットワーキング チームに作成してもらってください。

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。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。
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. プローブ コンポーネントからトラスト ストアを取得します。

   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 pods
   ```

   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。

   CA 信頼ストアを取得するには、次のコマンドを実行します。**<Probe-Pod-Id>** は記録した識別子です。

   ```
   kubectl cp <Probe-Pod-Id>:etc/pki/ca-trust/extracted/java/cacerts cacerts
   ```
4. 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 フォーマットであることを確認し、プローブの K8s シークレット名を使用して yaml ファイルを作成します。バッシュ セッションにいる間で、次のコマンドを入力します：

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

     MyCertificate.crt は、取得した証明書の名前です。
   - `base64 cacerts > <Secret_Name>-secrets.yaml`

     <Secret_Name>-secrets.yaml がある場所にプローブの K8s シークレット名を使用して作成する yaml ファイルがあります。

     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。

     ```
     dynatrace-secrets.yaml
     ```
5. 作成した <Secret_Name>-secrets yaml ファイルを更新します。

   a. Workload Optimization Manager サーバーの bash セッションにいる間に、前のステップで作成した yaml ファイルを vi エディター セッションで開きます:

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

   b. base64 データを yaml 形式に揃えます。

      : 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。

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

      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。

   c. 次に、次の格納ファイルを Base64 データの上のファイルに追加します：

      ```
      apiVersion: v1
      ```
      ```

```
kind: Secret
metadata:
  name: <Secret_Name>
data:
  cacerts: |
    xxxxxxxxxxxxxxxxxxxxxxxx
    xxxxxxxxxxxxxxxxxxxxxxxx
```

Base64 データは、上の例の xxx 文字の代わりに cacerts セクションにある必要があります。

6. yaml ファイルを 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。**<Probe-Pod-Id>** は記録した ID です。
   - kubectl apply -f **<Secret_Name>**-secrets.yaml
7. 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。

      ```
      vi /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。エントリがファイルに存在しない場合は、global: と同じレベルの spec: セクションに追加できます。プローブ エントリは 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。
   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
      ```

      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
          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識別子です。

```
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

Workload Optimization Manager プラットフォームには、プラットフォームのインストール時に有効化するように選択できる埋め込みレポートコンポーネントが含まれています。埋め込みレポートを使用すると、アプリケーション リソース管理の傾向を把握し、レポートやダッシュボードを介して関係者とインサイトを共有できます。

埋め込みレポートは、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。管理対象環境の履歴を保存し、一連の標準規格のダッシュボードとレポートを介して、この履歴の選択的なスナップショットを表示します。独自のダッシュボードとレポートを作成して、他の懸念事項に焦点を当てることができます。

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」 *(33 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」 *(34 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_xl_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 をインストールしたときに設定したシステム管理者でログインします。

   - ユーザ名：turbo
   - ユーザ名：[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。

```
次の場所に新しい変更を正常に適用しました。 /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alp
ha1_xl_cr.yaml.
次の場所にバックアップが書き込まれています。
/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。

```
CR ファイル /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml の適用
警告：kubectl apply は、いずれかによって作成されたリソースで使用する必要があります
kubectl create --save-config または kubectl apply
 xl.charts.helm.k8s.io/xl-release configured
変更が有効になるのを待っています...
構成の変更を適用するために api ポッドを再起動します。
ポッド「api-65cf47986f-jxszd」を削除
変更が正常に適用されました。埋め込みレポートが有効になりました。
```

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。

```
明らかな埋め込みレポートのインストール エラーは検出されませんでした。
```

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。
- 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 をインストールしたときに設定したシステム管理者でログインします。

   - ユーザ名：`turbo`
   - ユーザ名：`[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.yam` 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て、`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 reporting processes: 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ことが重要です。 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` および `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                                              準備      STATUS    RESTARTS
     extractor-7759dbcb47-vs6hr                        1/1     Running   0
     grafana-84ccb4bfb-l7sp7                           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 サーバー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。

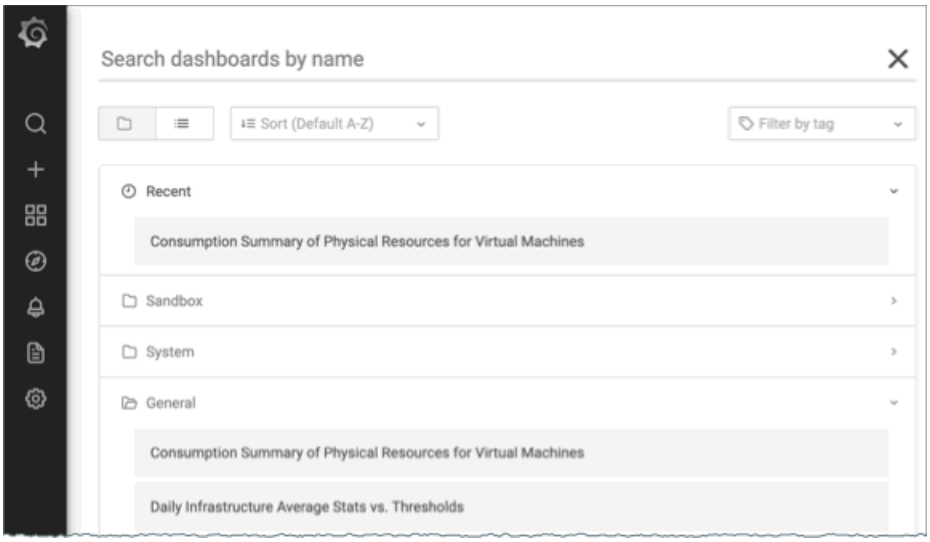
     ```
     postgresql-12.service - PostgreSQL 12 database server
     Loaded: loaded (/usr/lib/systemd/system/postgresql-12.service; enabled; vendor preset: disabled)
     Active: active (running) since Wed 2020-07-29 06:39:43 UTC; 14h ago
         Docs:https://www.postgresql.org/docs/12/static/
     Process: 1536 ExecStartPre=/usr/pgsql-12/bin/postgresql-12-check-db-dir ${PGDATA} (code=exited, s
     tatus=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/pgsql-12/bin/postmaster -D /var/lib/pgsql/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
```

埋め込みレポート ページへの移動

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 ユーザー インターフェイスを開けて、[ナビゲーション バー内（in navigation bar）]の**[レポート（Reports）]**をクリックします。

これにより、新しいブラウザ タブでダッシュボードとグラフが開きます。



特定のダッシュボードを検索するか、フォルダを参照すると、必要なダッシュボードを見つけることができます。カスタムレポートも作成できます。

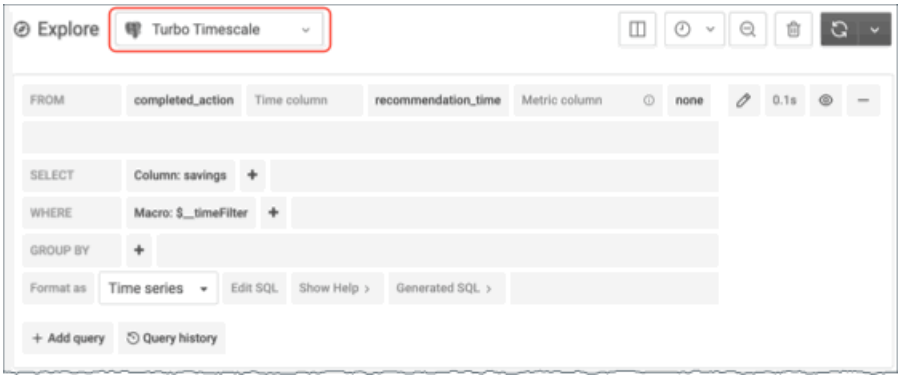
ダッシュボードとチャートは、Grafana® のオブザーバビリティ プラットフォームを使用します。Grafana を使用すると、既存のダッシュボードを簡単にナビゲートし、コーディングせずに独自のチャートやダッシュボードを作成できます。Grafana を初めて使用し、開始するのにヘルプが必要な場合は、次の場所にあるドキュメントを参照してください：

https://grafana.com/docs/grafana/latest/

注：
バージョン 3.4.1 以降、Workload Optimization Manager には Grafana の Enterprise ライセンスを使用するコントラクトがなくなり、代わりにオープンソースの Community ライセンスが出荷されます。以前のバージョンの Workload Optimization Manager をバージョン 3.4.1 以降に更新すると、Grafana は自動的にコミュニティ ライセンスに切り替わり、通常どおり埋め込みレポートを使用できるようになります。ただし、コミュニティ ライセンスは PDF レポートをサポートしていないことに注意してください。PDF レポートのガイダンスについては、Workload Optimization Manager の担当者にお問い合わせください。

カスタム レポートの作成

カスタムレポートを作成するには、埋め込みレポート データベース（`Turbo Timescale`）に対して SQL クエリを実行する必要があります。



データベース スキーマには、クエリを実行できる特定のテーブルが含まれています。

(オプション) レポートの編集

レポートを作成および編集するには、ユーザーはレポート エディタの権限を持っている必要があります。このユーザーは、*共有* ロールと*範囲指定された* ロール以外は全てのロールを持つことができます。Workload Optimization Manager インスタンスごとに 1 人のユーザーのみがこのレポート エディタ権限を持つことができます（デフォルトでは、ローカル**管理者**ユーザー）。これらの権限を別のユーザーに付与できます。

レポート エディタ 権限でユーザーアカウントを作成するには：

1. Workload Optimization Manager ユーザー インターフェイス内で**設定 / ユーザー管理**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* ロールと*範囲指定された*ロール以外は全てのロールを持つことができます。
4. [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。

   注：
   [レポート（Reports ）] ページに Report Editor のユーザー名が表示されるまで、最大 30 分かかることがあります。これは通常、Report Editor を複数回変更した場合に発生します。

   ユーザー インターフェイスからのレポートにアクセスするには、ユーザーに管理者またはサイト管理者ロール、または定義された範囲がない非管理者ロールを付与する必要があります。たとえば、オブザーバ ロールを持つが範囲を持たないユーザーは、レポートにアクセスできます。

   デフォルトの Shared Observer および Shared Adviser ロールには範囲が必要であるため、これらのロールを持つユーザーはレポートにアクセスできません。

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。

	エンティティの数						
保持期間	10k	25k	50k	100k	250k	500k	1000k
6 ヵ月	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。

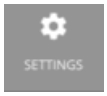
保持期間	エンティティ数
6 ヵ月	55k
1 年	27k
2 年	14k

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」 (21 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、各検出周期のエンティティ数をリストする INFO メッセージを Workload Optimization Manager の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) データエクスポータの有効化

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。

データエクスポータを有効にするには、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

データエクスポータ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 をインストールしたときに設定したシステム管理者でログインします。
   - ユーザ名：`turbo`
   - ユーザ名：`[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_xl_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 エクスポータ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 エクスポータ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                                          準備     STATUS    RESTARTS
...
extractor-5f41dd61c4-4d6lq                    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、Easticsearch 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 yaml 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                                          準備     STATUS    RESTARTS
...
es-kafka-connect-5f41dd61c4-4d6lq             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>
                キー： <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 の出力構成は、ES_HOSTS 環境変数によって識別される Elasticsearch サーバに対するものです。**<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、データエクスポータの 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>
                キー： <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:
```

```
ポート
  - 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：**DHCP を使用しますか、それとも静的 IP を設定しますか...

        static を選択します。
      - **必須：**このマシンの IP アドレスを入力してください
      - **必須：**このマシンのネットワーク マスクを入力してください
      - **必須：**このマシンのゲートウェイ アドレスを入力してください

- **必須：**このマシンの DNS サーバの IP アドレスを入力します

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。サプライチェーン、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。新しいバージョンに更新しても、プローブ構成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` ファイルの格納ファイルを表示します。

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_xl_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、結果が次のように表示されます。
`vcenter` probe:

```
NAME                                          準備     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: properties` 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_xl_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_xl_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                      準備     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」 *(52 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」 *(55 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」 *(69 ページ)*。

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 が構成されており、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。

   手順については、「時刻の同期」 *（16 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」 *（54 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 ノードを構成する `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: chart.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。場所の値は 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/exkexl6xc9MhzqiC30h7/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/exkexl6xc9MhzqiC30h7
  ```

- 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 証明書を入力します。

  IdP から取得したメタデータで、 のエントリを見つけます。<ds:X509Certificate>を使用して無効にすることができます。このタグの内容をコピーします。つまり、<ds:X509Certificate> と の間にある文字をコピーします。</ds:X509Certificate>。

  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
        A1UECAwKQ2FsaWZvcm5pYTEWMBQGA1UEBwwNU2FuIEZyYW5jaXNjbzENMAsGA1UECgwET2t0YTEU
        MBIGA1UECwwLU1NPUHJvdmlkZXIxEzARBgNVBAMMCmRldi03NzEyMDIxHDAaBgkqhkiG9w0BCQEW
        DWluZm9Ab2t0YS5jb20wHhcNMTgwNTAzMTk0MTI4WhcNMjgwNTAzMTk0MjI4WjCBkjELMAkGA1UE
        BhMCVVMxEzARBgNVBAgMCkNhbGlmb3JuaWExFjAUBgNVBAcMDVNhbiBGcmFuY2lzY28xDTALBgNV
        BAoMBE9rdGExFDASBgNVBAsMC1NTT1Byb3ZpZGVyMRMwEQYDVQQDDApkZXYtNzcxMjAyMRwwGgYJ
        KoZIhvcNAQkBFg1pbmZvQG9rdGEuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
        ugxQGqHAXpjVQZwsO9n8l8bFCoEevH3AZbz7568XuQm6MK6h7/O9wB4C5oUYddemt5t2Kc8GRhf3
        BDXX5MVZ8G9AUpG1MSqe1CLV2J96rMnwMIJsKeRXr01LYxv/J4kjnktpOC389wmcy2fE4RbPoJne
        P4u2b32c2/V7xsJ7UEjPPSD4i8l2QG6qsUkkx3AyNsjo89PekMfm+Iu/dFKXkdjwXZXPxaL0HrNW
        PTpzek8NS5M5rvF8yaD+eE1zS0I/HicHbPOVvLal0JZyN/f4bp0XJkxZJz6jF5DvBkwIs8/Lz5GK
        nn4XW9Cqjk3equSCJPo5o1Msj8vlLrJYVarqhwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQC26kYe
        LgqjIkF5rvxB2QzTgcd0LVzXOuiVVTZr8Sh57l4jJqbDoIgvaQQrxRSQzD/X+hcmhuwdp9s8zPHS
        JagtUJXiypwNtrzbf6M7ltrWB9sdNrqc99d1gOVRr0Kt5pLTaLe5kkq7dRaQoOIVIJhX9wgynaAK
        HF/SL3mHUytjXggs88AAQa8JH9hEpwG2srN8EsizX6xwQ/p92hM2oLvK5CSMwTx4VBuGod70EOwp
        6Ta1uRLQh6jCCOCWRuZbbz2T3/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 pod api-<API_POD_ID>` （注：ポッド ID を自動入力するには、`api-` と入力し、TAB を押します。）
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。

      時刻が同期されていない場合、ブラウザで `HTTP ステータス 401 -認証が機能不全` 例外が発生する可能性があります。
   d. 構成に失敗した場合は、 製品ログで `HTTP ステータス 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。

IdP メタデータの例

このセクションでは、メタデータのオプションの属性を調べるときに役立つ IdP メタデータの例を示します。

例に記載されていないオプションの属性タグがメタデータに含まれている場合、それらのオプションの属性タグはサポートされていないため削除する必要があります。

```
<?xml version="1.0" encoding="UTF-8"?>
     <md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
     entityID="http://www.okta.com/exkexl6xc9MhzqiC30h7">
     <md:IDPSSODescriptor WantAuthnRequestsSigned="false"
     protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
     <md:KeyDescriptor use="signing">
     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
     <ds:X509Data>
     <ds:X509Certificate>
     MIIDpDCCAoygAwIBAgIGAWMnhv7cMA0GCSqGSIb3DQEBCwUAMIGSMQswCQYDVQQGEwJVUzETMBEG
     A1UECAwKQ2FsaWZvcm5pYTEWMBQGA1UEBwwNU2FuIEZyYW5jaXNjbzENMAsGA1UECgwET2t0YTEU
     MBIGA1UECwwLU1NPUHJvdmlkZXIxEzARBgNVBAMMCmRldi03NzEyMDIxHDAaBgkqhkiG9w0BCQEW
     DWluZm9Ab2t0YS5jb20wHhcNMTgwNTAzMTk0MTI4WhcNMjgwNTAzMTk0MjI4WjCBkjELMAkGA1UE
     BhMCVVMxEzARBgNVBAgMCkNhbGlmb3JuaWExFjAUBgNVBAcMDVNhbiBGcmFuY2lzY28xDTALBgNV
     BAoMBE9rdGExFDASBgNVBAsMC1NTT1Byb3ZpZGVyMRMwEQYDVQQDDApkZXYtNzcxMjAyMRwwGgYJ
     KoZIhvcNAQkBFg1pbmZvQG9rdGEuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
     ugxQGqHAXpjVQZwsO9n8l8bFCoEevH3AZbz7568XuQm6MK6h7/O9wB4C5oUYddemt5t2Kc8GRhf3
     BDXX5MVZ8G9AUpG1MSqe1CLV2J96rMnwMIJsKeRXr01LYxv/J4kjnktpOC389wmcy2fE4RbPoJne
     P4u2b32c2/V7xsJ7UEjPPSD4i8l2QG6qsUkkx3AyNsjo89PekMfm+Iu/dFKXkdjwXZXPxaL0HrNW
     PTpzek8NS5M5rvF8yaD+eE1zS0I/HicHbPOVvLal0JZyN/f4bp0XJkxZJz6jF5DvBkwIs8/Lz5GK
     nn4XW9Cqjk3equSCJPo5o1Msj8vlLrJYVarqhwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQC26kYe
     LgqjIkF5rvxB2QzTgcd0LVzXOuiVVTZr8Sh57l4jJqbDoIgvaQQrxRSQzD/X+hcmhuwdp9s8zPHS
     JagtUJXiypwNtrzbf6M7ltrWB9sdNrqc99d1gOVRr0Kt5pLTaLe5kkq7dRaQoOIVIJhX9wgynaAK
     HF/SL3mHUytjXggs88AAQa8JH9hEpwG2srN8EsizX6xwQ/p92hM2oLvK5CSMwTx4VBuGod70EOwp
     6Ta1uRLQh6jCCOCWRuZbbz2T3/sOX+sibC4rLIlwfyTkcUopF/bTSdWwknoRskK4dBekFcvN9N+C
     p/qaHYcQd6i2vyor888DLHDPXhSKWhpG
     </ds:X509Certificate>
     </ds:X509Data>
     </ds:KeyInfo>
     </md:KeyDescriptor>
     <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</md:NameIDFormat>
     <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</md:NameIDFormat>
     <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
     Location="https://dev-771202.oktapreview.com/app/ibmdev771202_turbo2_1/exkexl6xc9MhzqiC30h7/sso/
saml"/>
     <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
     Location="https://dev-771202.oktapreview.com/app/ibmdev771202_turbo2_1/exkexl6xc9MhzqiC30h7/sso/

saml"/>
     </md:IDPSSODescriptor>
     </md:EntityDescriptor>
```

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://${hostname}/vmturbo/oauth2/login/code/${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。

注：
この URL は、プロバイダーの OpenID 構成の「承認済みダイレクト URI」セクションでも設定する必要があります。

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/login/code/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 が構成されており、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。

   手順については、「時刻の同期」 *(16 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。
openIdExternalGroupTag	認証に使用するカスタム グループの名前。

- 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。
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 ノードを構成する 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 chart_v1alpha1_xl_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: chart.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: Xl
  metadata:
    name: xl-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: Xl
  metadata:
    name: xl-release
  spec:
    properties:
      api:
        openIdEnabled: true
        openIdClients: google
        openIdClientId: xxxx-4vinrdgllag5p84jjebc6xxxxxx5u.apps.googleusercontent.com
        openIdClientSecret: xxxxxhGcdFEjQa-xxxxxxxx
  ```

  – 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: "xxxxxxvZ2ZscDhtOFVxxxxxxU3d6cXR4cTZhb2xxxxxxRT0K"
      openIdAccessTokenUri: https://icp-console.apps.blue-13.dev.multicloudops.io/idprovider/
v1/auth/token
      openIdUserAuthorizationUri: https://icp-console.apps.blue-13.dev.multicloudops.io/idpro
vider/v1/auth/authorize
      openIdUserInfoUri: https://icp-console.apps.blue-13.dev.multicloudops.io/v1/auth/userI
nfo
      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: xxxxxxxxxh1xhQnSKxxxx
      openIdClientSecret: xxxxxxxxxxtIhVCIRUnhq4xxxxxxxDdhLdqx0
      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 pod api-<API_POD_ID>` （注：ポッド ID を自動入力するには、`api-` と入力し、TAB を押します。）
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 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。

時刻が同期されていない場合、ブラウザで HTTP ステータス 401 -認証が機能不全 例外が発生する可能性があります。

d. 構成に失敗した場合は、 製品ログで HTTP ステータス 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 chart_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: chart.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 pod api-<API_POD_ID>` （注：ポッド ID を自動入力するには、`api-` と入力し、TAB を押します。）
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。シスコは、この製品の新しいバージョンを 2 週間ごとにリリースします。IBM Workload Optimization Manager の資料サイトにアクセスして、新しいバージョンが利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。そして、データベースは仮想環境全体のパフォーマンス データを保持します。
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。

OVA の更新

更新手順を開始する前に、次の手順を実行します。

- ユーザー ガイドの 新機能とはセクションを確認して、このリリースの新機能を確認してください。

  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。新しいバージョンに更新しても、プローブ構成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。

- 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で、Workload Optimization Manager は現在、MariaDB バージョン 10.5.16 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 バージョンの確認を参照します。 *(18 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」 *(62 ページ参照)*）。

- 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」 *(66 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。
      i. 次のシスコ Web ページに移動します。
      https://software.cisco.com/download/home/286328879/type
      ii. **[ソフトウェア タイプの選択（Select a Software Type）]** で、**Workload Optimization Manager** をクリックします。
      iii. 左側のメニューから、目的の Workload Optimization Manager バージョンを選択します。
      iv. 次のファイルのダウンロード ボタンをクリックします。

      ```
      upgrade-precheck-X.X.X.zip
      ```
      v. プロンプトが表示されたら、Cisco アカウントを使用してログインします。
      vi. ダウンロードが完了したら、ダウンロードしたファイルを解凍します。
   d. 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 をインストールしたときに設定したシステム管理者でログインします。 ■ ユーザ名：turbo ■ パスワード：[your_private_password] 次のファイルを編集します。 `/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml`	以下のファイルを編集のために開きます。 `deploy/crds/charts_v1alpha1_xl_cr.yaml`

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
      actionDbPassword: 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 WorkloadOptimization 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ます。

   a. 次のシスコ Web ページに移動します。

   https://software.cisco.com/download/home/286328879/type

   b. **[ソフトウェア タイプの選択（Select a Software Type）]** で、**Workload Optimization Manager** をクリックします。

   c. 左側のメニューから、目的の Workload Optimization Manager バージョンを選択します。

   d. 次のファイルのダウンロード ボタンをクリックします。

   ```
   update64_package-X.X.X.iso
   ```

   e. プロンプトが表示されたら、Cisco アカウントを使用してログインします。

3. Workload Optimization Manager を実行する VM で使用できる場所に ISO イメージを保存します。画像を 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. **[電源オン時に接続する（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 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 をインストールしたときに設定したシステム管理者でログインします。

   - ユーザ名：`turbo`
   - ユーザ名：`[your_private_password]`

5. 更新バージョンの isoUpdate.sh スクリプトを取得します。

   a. 次のシスコ Web ページに移動します。

   https://software.cisco.com/download/home/286328879/type

   b. **[ソフトウェア タイプの選択（Select a Software Type）]** で、**Workload Optimization Manager** をクリックします。

   c. 左側のメニューから、目的の Workload Optimization Manager バージョンを選択します。

   d. 次のファイルのダウンロード ボタンをクリックします。

   ```
   isoUpdate-X.X.X.zip
   ```

   e. プロンプトが表示されたら、Cisco アカウントを使用してログインします。

   f. ダウンロードが完了したら、ダウンロードしたファイルを解凍します。

6. 使用する Workload Optimization Manager インスタンスにスクリプトをアップロードします。

   ローカル マシンから Workload Optimization Manager サーバーへのファイル転送を実行します。スクリプトを、Workload Optimization Manager を実行する VM の `/opt/ local/bin/` に保存します。

7. スクリプトを実行可能にします。

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

8. 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 (ただし、MariaDB バージョンは更新されません)。
   - 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。
9. 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 実行中 と表示されます。

   以下のような出力が表示されます。

```
NAME                                          準備     STATUS    RESTARTS
action-orchestrator-b6454c9c8-mfl85           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-4lzfv        1/1     Running   0
mediation-appdynamics-6d65f8766f-kb44l        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-6dffc4c9b6-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
```

10. 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 10.5.16 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=my_pwd -e "SHOW VARIABLES LIKE 'version';"
    ```

    出力は次のようになります。

    ```
    +---------------+-----------------+
    | Variable_name | Value           |
    +---------------+-----------------+
    | version       | 10.5.16-MariaDB |
    +---------------+-----------------+
    ```

    MariaDB のバージョンが 10.5.16 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」 *(18 ページ)* を参照してください。

11. 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。

12. 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。新しいバージョンに更新しても、プローブ構成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。

    <x id="group-0-START"/>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」（47* ページ）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、`kubectl` を介してこれらの変更を適用します。

13. 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

注：
ここで説明するプロセスは、OVA 展開モデルにのみ適用されます。

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 プロバイダから取得した登録識別子）	`https://<hostname>/vmturbo/saml2/sso/<samlRegistrationID>`
接続先の URL（**<hostname>** は、Workload Optimization Manager が実行されるホスト、**<samlRegistrationID>** は、SSO プロバイダから取得した登録識別子）	`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
設定	例
署名アルゴリズム	`RSA_SHA256`

SAML 設定：全般	
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
シングル ログアウト URL（**<hostname>** <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：FIPS 暗号スイート

注：
ここで説明するプロセスは、OVA 展開モデルにのみ適用されます。

安全な接続の暗号強度を確保するために、連邦情報処理標準（FIPS）が導入されています。デフォルトでは、Workload Optimization Manager は、FIPS 準拠の暗号スイートがすでに有効化された状態で出荷されます。このスイートは、次の暗号で構成されています。

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384

暗号スイートの変更

必要に応じて、内部ポリシーに準拠するように暗号スイートを変更できます。

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. 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_xl_cr.yaml
   ```
3. 暗号スイートを編集します。
   ファイル内の暗号のリストを検索します。ポリシーの必要に応じてリストを変更し、ファイルを保存します。
4. 変更をプラットフォームに適用します。

   ```
   kubectl apply -f \
   ```

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

付録：段階的なプラットフォーム展開

注：
ここで説明するプロセスは、OVA 展開モデルにのみ適用されます。

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」（51 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 実行中 と表示されます。

      以下のような出力が表示されます。

      ```
      NAME                                  準備     STATUS    RESTARTS
      action-orchestrator-b6454c9c8-mfl85   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-4lzfv       1/1     Running   0
mediation-appdynamics-6d65f8766f-kb44l       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-6dffc4c9b6-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」 *(16 ページ)* および「シングル サインオン認証」 *(51 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ーブ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 プローブのセットを変更できます（「プローブ コンポーネントの有効化と無効化」 *(47 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+
  - AppInsights
  - Dynatrace 1.1+

  - IBM WebSphere Application Server 8.5+
  - Instana、リリース 209 以降
  - 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、12c、18c、および 19c
  - Oracle WebLogic 12c
- クラウドネイティブ
  - 準拠した k8s ディストリビューション（Rancher、Tanzu、オープンソースなど）を含む Kubernetes
  - クラウドでホストされる k8s サービス（AKS、EKS、GKE、IBM、Cisco IKS、ROKS、ROSA など）
  - OpenShift 3.11 以降（OCP 4.x）
- 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 8 / 8.1、10、2008 R2、2012 / 2012 R2、2016、2019 および 7
- ハイパーコンバージド
  - Cisco Hyperflex 3.5
  - 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
  - Flexera One
  - ServiceNow
- プライベート クラウド
  - Microsoft System Center 2012/2012 R2 Virtual Machine Manager、System Center 2016 Virtual Machine Manager、および System Center Virtual Machine Manager 2019
- パブリック クラウド
  - Amazon AWS
  - Amazon AWS Billing
  - Google Cloud Platform（GCP）
  - Google Cloud Platform（GCP）課金情報
  - Microsoft Azure Service Principal
  - Microsoft Azure Billing
  - 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 +

– Spectrum Virtualize 8.3.1.2 またはそれ以降（8.4.2.0 またはそれ以降を推奨）で実行する IBM FlashSystem

  - ONTAP 8.0+ を使用した NetApp Cluster Mode（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」（73 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

注：
ここで説明するプロセスは、OVA 展開モデルにのみ適用されます。

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 WorkloadOptimization 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. **[電源オン時に接続する（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/turboload.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。

```
t8c アップグレード iso がマウントされました
イメージ チェック：
============
**************************
すべてのイメージがロードされています
**************************
```

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。

8. 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。

9. 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 10.5.16 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=my_pwd -e "SHOW VARIABLES LIKE 'version';"
   ```

   出力は次のようになります。

   ```
   +---------------+-----------------+
   | Variable_name | Value           |
   +---------------+-----------------+
   | version       | 10.5.16-MariaDB |
   +---------------+-----------------+
   ```

   MariaDB のバージョンが 10.5.16 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」（18 ページ）を参照してください。

10. 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
    ```

11. 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。

12. 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。新しいバージョンに更新しても、プローブ構成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。

    <x id="group-0-START"/>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」（47 ページ）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、`kubectl` を介してこれらの変更を適用します。

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 実行中 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：YAML ファイルの操作

YAML は、kubernetes の技術情報を作成し、構成するための主要なファイル フォーマットで、Workload Optimization Manager プラットフォームに関連するすべてのものを含みます。カスタム技術情報 YAML は、Workload Optimization Manager の構成の詳細の大部分を定義するための便利な単一の場所を提供します。編集のための一般的なルールには、次のものが含まれます。

- すべてのインデントには、タブではなく、常にスペースを使用します。スペースとインデントは重要であり、無効な結果やパラメータが完全にスキップされる可能性があるため、インデントに関連付けられた縦線の使用をサポートするエディタを使用して YAML ファイルを操作し、ミスアライメントを視覚的に見つける必要があります。選択したエディタでこれが難しい場合は、完了したときに Linux 拡張 ユーティリティを使用して、タブを同等のスペースに変更することができます。
- インデントはレベルごとに 2 個のスペースを使用します。
- 指定されたセクションのすべてのプロパティに対して、注意深く同じインデントを維持してください。
- 同じセクションで同じプロパティ名を 2 回使用しないでください。これを行うと、YAML ファイルが無効になりますが、問題の通知は表示されない可能性があります。むしろ、1 つを除くすべてのプロパティ定義が黙って無視されます。

間隔の問題

Workload Optimization Manager カスタム技術情報の場合、インデントによってパラメータが適用される場所（グローバルまたは特定のコンポーネント）が定義されるため、テキストを適切に並べてください。以下の例はすべてのインスタンスにコンテナ イメージ タグを設定するグローバル レベル（{"spec":{"global":[{"tag":"8.6.4"}]}}）で適用される仕様の例を示しています。次にインデントすると、リモート データベース (`{"spec":{"properties":{"global":[{"dbPort":"6033"}]}}}`) に対してグローバルなプロパティで、リモート DB 接続用に設定される dbPort のプロパティを記述します。各行は、上位レベルから 2 個のスペースでインデントされます。

```
spec:
  global:
    repository: turbonomic
    tag: 8.6.4
  properties:
     global:
       dbPort: 6033
  kubeturbo:
    enabled : true
  aws:
    enabled : true
```

プロパティを正しく組み合わせる

YAML ファイルはトップダウン方式で読み込み、同じコンポーネントに適用される異なるパラメータがある場合、それらを組み合わせる必要があります。次の例はイメージ タグの `ui` コンポーネントとメモリ制限リソースのプロパティを表示する YAML を表示します。

```
spec:
  global:
    repository: turbonomic
    tag: 8.6.4
  ui:
    image:
      tag: 8.0.5
  properties:
    global:
      dbPort: 6033
  kubeturbo:
    enabled : true
  aws:
    enabled : true
  ui:
    resources:
      limits:
        memory: 4Gi
```

この YAML は、ui コンポーネントのイメージ タグとメモリ制限技術情報の両方をセットしません。それらは 2 つの異なるセクションでセットされるからです。メモリ制限は最後に読み取られるセクションであるため、メモリ制限が適用され、プロパティの最初のセットがイメージ タグで上書きされます。次の YAML は ui コンポーネントのイメージ タグとメモリ制限リソースの両方をセットします :

```
spec:
  global:
    repository: turbonomic
    tag: 8.6.4
  ui:
    image:
      tag: 8.0.5
    resources:
      limits:
        memory: 4Gi
  properties:
    global:
      dbPort: 6033
  kubeturbo:
    enabled : true
  aws:
    enabled : true
```

翻訳について
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。