Workload Optimization Manager 3.5.5 Installation Guide

Contents

Introduction

Thank you for choosing Workload Optimization Manager, the premier solution for Application Resource Management (ARM) of cloud and virtual environments. This guide gives you information you need to install Workload Optimization Manager in your virtual environment, install your license, and get started managing your resources.

If you have any questions, please contact Cisco support.

Sincerely:

The Workload Optimization Manager Team

Minimum Requirements

License Requirements

To run Workload Optimization Manager on your environment, you must install the appropriate license. Licenses enable different sets of Workload Optimization Manager features, and they support a specified number of workloads in your environment.

User Interface Requirements

To display the Workload Optimization Manager user interface, you must log into the platform with a browser that can display HTML5 pages. Workload Optimization Manager currently supports the following browsers:

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

Network Addressing Requirements

Workload Optimization Manager requires static IP addressing. Static IP setup is covered as a step when installing the Workload Optimization Manager VM image.

Compute and Storage Requirements

The requirements for running a Workload Optimization Manager instance depend on the size of the environment you are managing. Workload Optimization Manager keeps a real-time representation of your environment in memory. The greater the number of entities to manage, and the more extensive the relationships between them, the more resources you need for the VM that runs Workload Optimization Manager. And as the VM requirements increase, so do the requirements for the physical machine that hosts the VM.

The requirements listed here are recommendations that you should keep in mind as you plan your Workload Optimization Manager deployment. After deploying, if you find that you need to change memory capacity, CPU capacity, or both for the VM, you can shut it down, make changes, and then power it up again to use the new capacity.

NOTE:
The machine that hosts the Workload Optimization Manager platform must support the SSE4.2 instruction set. Support for this instruction set was introduced at different times for different chip manufacturers:

- Intel: November 2008
- AMD: October 2011

The machine you use to host Workload Optimization Manager should be newer than these dates. On a Linux system, you can run the following command to check for this support:

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

For more information, see the glossary entry at http://www.cpu-world.com/Glossary/S/SSE4.html.

In most cases you can run Workload Optimization Manager on a host that meets the following minimum requirements:

Supported VM Image Technology		Storage Requirements	Memory	CPUs
VMware	vCenter versions 5.5, 6.0, 6.5, 6.7, and 7.0	1.25 TB or greater. **NOTE:** Can be thin provisioned depending on the storage requirements.	■ Default: 128 GB ■ For 10,000 VMs or less, 64 GB	8 vCPUs
Microsoft	Hyper-V Server 2012 R2 or later			

Cisco provides a VM image (an OVA or VHD file) which is preconfigured with two hard drives. A minimum of 1.25 TB is necessary to ensure that the drives have the proper amount of space for storage.

Installing on a Virtual Machine Image

You can get a download of the Workload Optimization Manager platform as a:

- VMware OVA `1.0` image
- Microsoft Hyper-V image

NOTE:
For minimum requirements, we recommend 128 GB of memory for the VM that hosts Workload Optimization Manager. However, if you plan to manage a smaller environment (10,000 VMs or less), you can install on a VM that provides 64 GB of memory. (See Minimum Requirements *(on page 6)*).

If you plan to install a VM with 64 GB of memory, then you must modify the default for VM memory. (See Deploy the Workload Optimization Manager VM *(on page 9)*).

You will install the platform in two main steps:

1. Install the Workload Optimization Manager VM image on your network.

   This installs and starts up the VM that will host your instance of the Workload Optimization Manager platform.
2. Deploy the Workload Optimization Manager components on the VM.

About the Workload Optimization Manager VM Image

Workload Optimization Manager installs as a VM that runs the CentOS Linux OS. For each new version, we deliver a VM image (OVA or VHD) that you install to run the product. Typically you install this image once, and for subsequent updates to Workload Optimization Manager you will execute product updates on that installed VM. This means two things:

- Product updates patch new components of the Workload Optimization Manager application stack onto the same CentOS platform that you got when you originally installed the VM image. Product updates do not affect the underlying OS.
- Over time, you might learn of important security patches for the CentOS distribution. It is your responsibility to keep the OS up to date. You can install these patches on your Workload Optimization Manager VM whenever necessary.

NOTE:
We currently release the VM image with the CentOS Linux OS. We have found it to meet overall security requirements. We intend to continue with CentOS for as long as that platform remains viable and secure.

OVA: Installing the vCenter Image for On-prem Environments

The first step to installing Workload Optimization Manager is to deploy the VM that will host the platform.

For vCenter Server environments, we deliver an OVA image for each quarterly release. If you want to run Workload Optimization Manager on vCenter Server, you can install the Quarterly Release, and then update to a later point release if necessary.

NOTE:
For minimum requirements, we recommend 128 GB of memory for the VM that hosts Workload Optimization Manager. However, if you plan to manage a smaller environment (10,000 VMs or less), you can install on a VM that provides 64 GB of memory. (See Minimum Requirements *(on page 6)*).

If you plan to install a VM with 64 GB of memory, then you must modify the default for VM memory. (See Deploy the Workload Optimization Manager VM *(on page 9)*).

To install the Workload Optimization Manager OVA:

1. Download the Workload Optimization Manager installation package.

   Navigate to the Workload Optimization Manager Software Download page (https://software.cisco.com/download/home/286328879/type/286317011/release) for links to the latest OVA image.

   The installation package includes the `cisco_cwom-<version>-<XXXXXXXXXXXXXX>.ova` file

   where <version> is the Workload Optimization Manager version number and <XXXXXXXXXXXXXX> is the timestamp.

   For example: `cisco_cwom-3.0.0-20190916164429000.ova`

   The OVA file deploys as a VM with the Workload Optimization Manager components ready for installation.
2. Import the OVA file into your datacenter.

   Use the vCenter Server client to import the OVA into your environment.
3. Deploy the Workload Optimization Manager VM.

   Configure the VM that was deployed from the OVA file.

   For minimum requirements, we recommend 128 GB of memory for the VM that hosts Workload Optimization Manager. However, if you plan to manage a smaller environment (10,000 VMs or less), you can install on a VM that provides 64 GB of memory. (See Minimum Requirements *(on page 6)*).

   If you want to deploy a VM with 64 GB of memory, manually modify the default value for Memory:

   a. Right-click the VM and choose **Edit Settings**.
   b. Type **64** for Memory.
   c. Click **OK** to save the settings
   d. Power on the VM.
4. Open the remote console.

   For the Workload Optimization Manager VM that you just deployed:

   a. Choose the **Summary** tab.
   b. Click **Launch Remote Console**.
5. Set up the Workload Optimization Manager System Administrator account.

   a. In the remote console, log in with the following default credentials:

      - Username: `turbo`

        Do not use the account name, `root`.
      - Password: `vmturbo`

        Then, you will be prompted to enter a new password.

   b. Enter your new password.

      The new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.

      NOTE:
      Be sure to save the changes account credentials in a safe place. For security reasons, this is the only account that can access and configure the Workload Optimization Manager VM.

   c. Enter your new password again to verify it.
6. Update the root password.

The platform uses the `root` account for certain processes, such as rolling up log messages in `/var/log/messages`. To ensure the account credentials are current, you must change the password:

a. Open a SuperUser session.
   - In the remote console, enter `su -`
   - At the password prompt, enter the default password: `vmturbo`

b. Reset a new password.

   After you log in as root with the default password, the system prompts you for a `New password`. This new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.

   NOTE:
   Be sure to save the root account credentials in a safe place.

c. Exit the SuperUser session.

   Enter `exit`.

7. Perform other necessary configuration steps, and then install the Workload Optimization Manager components.

   To perform the required and important configuration steps for the Workload Optimization Manager instance, see General Configuration Tasks *(on page 16)*.

   To install the Workload Optimization Manager components, see Deploying the Workload Optimization Manager Components *(on page 73)*.

VHD: Installing the Microsoft Hyper-V Image

The first step to installing Workload Optimization Manager is to deploy the VM that will host the platform.

For Hyper-V environments, we deliver a Hyper-V image for each quarterly release. If you want to run Workload Optimization Manager on a Hyper-V VM, you can install the Quarterly Release, and then update to a later point release if necessary.

NOTE:
For minimum requirements, we recommend 128 GB of memory for the VM that hosts Workload Optimization Manager. However, if you plan to manage a smaller environment (10,000 VMs or less), you can install on a VM that provides 64 GB of memory. (See Minimum Requirements *(on page 6)*).

If you plan to install a VM with 64 GB of memory, then you must modify the default for VM memory. (See Deploy the Workload Optimization Manager VM *(on page 9)*).

To install Workload Optimization Manager:

1. Download the Workload Optimization Manager installation package.

   Navigate to the Workload Optimization Manager Software Download page (https://software.cisco.com/download/home/286328879/type/286317011/release) for links to the latest Hyper-V image.
2. Expand the .zip file and copy the contents, which includes the Virtual Machine image, to your Hyper-V server (either to your cluster shared volume or to a local hard drive).
3. Use the Import Virtual Machine Wizard in the Hyper-V Manager to import the Virtual Machine into your environment.
4. Make sure your virtual network adapter is connected to the correct virtual network.
5. Ensure the Workload Optimization Manager instance will have sufficient memory.

   Cisco recommends that you use static memory for your Workload Optimization Manager instance. However, you can specify static or dynamic memory for the instance. By default, the installation sets static memory to128 GB.
6. Start the Workload Optimization Manager appliance and record its IP address.
7. Set up the Workload Optimization Manager System Administrator account.
   a. Log into the VM's Hyper-V console with the following default credentials:
      - Username: `turbo`

        Do not use the account name, `root`.
      - Password: `vmturbo`

Then, you will be prompted to enter a new password.

b. Enter your new password.

The new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.

NOTE:
Be sure to save the changed account credentials in a safe place. For security reasons, this is the only account that can access and configure the Workload Optimization Manager VM.

c. Enter your new password again to verify it.

8. Update the root password.

The platform uses the `root` account for certain processes, such as rolling up log messages in `/var/log/messages`. To ensure the account credentials are current, you must change the password:

a. Open a SuperUser session.

- In the remote console, enter `su -`
- At the password prompt, enter the default password: `vmturbo`

b. Reset a new password.

After you log in as root with the default password, the system prompts you for a `New password`. This new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.

NOTE:
Be sure to save the root account credentials in a safe place.

c. Exit the SuperUser session.

Enter `exit`.

9. Enable the NIC for the installed VM.

The Workload Optimization Manager instance configuration includes one NIC, but it is not enabled or connected to a network. Display the NIC in the Hyper-V Manager and enable it.

10. Perform other necessary configuration steps, and then install the Workload Optimization Manager components.

To perform the required and important configuration steps for the Workload Optimization Manager instance, see General Configuration Tasks *(on page 16)*.

To install the Workload Optimization Manager components, see Deploying the Workload Optimization Manager Components *(on page 73)*.

Deploying the Workload Optimization Manager Components

NOTE:
This section describes the default installation process. If you want to customize your installation, then you should consider taking the steps in Appendix: Stepwise Platform Deployment *(on page 73)*. For example, to change the Kubernetes host name for the deployment, you must perform a stepwise installation.

Starting with Workload Optimization Manager version 3.5.5, Cisco Container Registry is used for all Workload Optimization Manager images for online upgrades and new installs. Additionally, all new OVA installs will use Kubernetes v1.24.6. Docker commands will no longer work; crictl and ctr commands can be used instead. For more information, see:

- Container runtime changes in Kubernetes 1.24 and beyond on the Kubernetes documentation site
- Mapping from dockercli to crictl on the Kubernetes documentation site
- crictl command reference and information on GitHub
- ctr command reference and information on GitHub

After you have installed the Workload Optimization Manager VM that will host the platform, you can install the platform components, as follows:

First, gather the information you will need to run the installation:

- Network Time Source for Time Synchronization (optional)

  You can perform this step during installation, or at a later date. If you want to synchronize the VM's clock now, you will be prompted for the Network Time Source. For more information about synchronizing the VM's clock, see Synchronizing Time *(on page 16)*.
- Your updated `root` password

  The installation script requires that you have updated the `root` password for the VM. If you followed the instructions in OVA: Installing the vCenter Image *(on page 8)* or in VHD: Installing the vCenter Image *(on page 10)*, then you should have already performed this step.

When you are ready with the necessary information, you can run the installation script.

1. Start up the installation script.
   - Start a secure session (SSH) on your Workload Optimization Manager VM as the `turbo` user.
   - Execute the installation script:

     ```
     sudo /opt/local/bin/t8cInstall.sh
     ```
2. Verify that you have configured a static IP address for the Workload Optimization Manager VM.

   After the components start up, you will type this static IP address into a web browser to access the login page for the Workload Optimization Manager user interface.

   As a first step, the script prompts you with:

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

   If you have not configured a static IP for the platform VM, enter `n` to exit the installation script now, and configure a static IP.

   If you have already configured a static IP for the platform VM, enter `y` to continue the installation. The script output displays the IP address that it recognizes for the VM, for example:

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

   NOTE:
   Because of dependencies between Workload Optimization Manager and the Kubernetes installation, it is not recommended to change the IP address after the Workload Optimization Manager installation. For a production installation of Workload Optimization Manager, the VM must run with a static IP. For a testing or evaluation installation, you can use DHCP. However, if you plan to later use such an installation in a production environment, you should be sure to configure a static IP.

   If you followed the instructions in OVA: Installing the vCenter Image *(on page 8)* or in VHD: Installing the vCenter Image *(on page 10)*, then you should have already run the `ipsetup` script to do this.
3. Wait while the script performs the installation.

   As the installation process continues, the script:
   - Configures the platform environment with the necessary certificates
   - Configures the Kubernetes cluster on the VM

     This can take a few tries before it succeeds. For each try that does not succeed, you will see messages similar to:

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

     When the connection succeeds, the script advances to the next steps.
   - Establishes local storage for the platform
   - Creates the kubernetes namespace for the platform as `turbonomic`
   - Configures authorization to access the required Kubernetes secrets

- Initializes the MariaDB database server to manage historical data for the platform

  The script creates two accounts on the MariaDB that have full privileges:

  - `root@localhost`

    This account does not use a password. To connect via this account the user must be `system root`.
  - `mysql@localhost`

    This account does not use a password. To connect via this account the user must be `system mysql`.

  NOTE:
  For security reasons, Cisco recommends that you configure passwords for these accounts. You can connect with these accounts via sudo. For example, `sudo mysql`. After you connect, you can then set passwords to these accounts. For more information, see the MariaDB Knowledgebase at https://mariadb.com/kb.
- Installs the Timescale database for Embedded Reports and the Data Exporter
- Deploys and starts up the platform components

As the deployment begins, the script prints out the following:

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

After it deploys the components, it waits for the components to start up:

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

If the components all start up within 30 minutes, then the installation is complete and successful.

If the components do not all start up within 30 minutes, the script displays the following and then exits:

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

The script then displays the formatted result of the `kubectl get pods` command. This shows you the current status of the pods in the Workload Optimization Manager platform.

NOTE:
If the script exits before the components have all started up, we recommend that you give the platform another 30 minutes. To periodically check the component status, execute `kubectl get pods`. If the components do not all start up after you have waited another 30 minutes, contact your support representative.

If the installation is successful and the components have all started up, the script displays a message similar to the following, where it gives the VM's static IP address:

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

You can move on to the next steps.

4. Save a copy of the platform's Master Key secret.

   The installation procedure creates a Master Key secret in the Kubernetes cluster. Workload Optimization Manager uses this secret to provide access for the platform components. You should save the key data to a safe location. If for some reason

the key data gets corrupted or is otherwise unusable, Workload Optimization Manager will fail to operate. If this happens, you can contact your support representative and use this saved data to recover your platform.

To save the data:

a. List the platform secrets.

Execute the command:

```
kubectl get secrets
```

The results should include the Master Key secret, similar to the following:

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

b. Display the Master Key data.

Once you find the Master Key name, you can then display the key data:

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

The command result should be similar to the following:

```
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
```

The important key data to save is the `primary_key...` data. In the above example, you should save the line:

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

c. Save the data to a safe place.

Write this data to a file and save it in a safe backup location. If you ever need to recover the Master Key, your support representative will use this data to perform the recovery.

5. Log in to the Workload Optimization Manager user interface and set the administrator user account password.

Workload Optimization Manager includes a default user account named `administrator` which has an `ADMINISTRATOR` role. As you log in for the first time, you must set your own password for that account. You can create or delete other accounts with the `ADMINISTRATOR` role, but your installation of Workload Optimization Manager must always have at least one account with that role.

In the login page, enter the information as required, and make a note of it.

- Use the default credential for **USERNAME**: `administrator`.
- Type a password for **PASSWORD**.

  The new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.
- Type the password again to verify it for **REPEAT PASSWORD**.
- Click **Create Account**.

This is the account you will use to access the Workload Optimization Manager user interface with administrator permissions. *Be sure to save the user interface administrator account credentials in a safe place.*

NOTE:
The initial login is always for the `administrator` account. This is an administration *user* account, not a Workload Optimization Manager System Administrator account.

6. After you have logged in as `administrator`, you can create other user accounts, and you can give them various roles. For more information about user accounts and roles, see the *Workload Optimization Manager User Guide*.

NOTE:
For security reasons, you can create a different administrator user account to serve as the main administrator of your Workload Optimization Manager installation, and then delete the default `administrator` account. But remember, *you must always have at least one user account with administrator privileges*.

General Configuration Tasks

After you install the Workload Optimization Manager instance, you should perform the following configuration tasks:

- (Required) Synchronize the system clock and configure your time servers.
- (Important) Verify your MariaDB version.
- (Optional) Increase available disk space.
- (Optional) Enforce secure access via LDAP.
- (Optional) Enforce secure access via trusted certificate.
- (Optional) Enable secure access for probes.
- (Optional) Modify the certificates for Cluster Manager.
- (Optional) Enable embedded reports.
- (Optional) Enable the Data Exporter.
- (Optional) Change the IP address of the platform node.
- (Optional) Enable and disable probe components.

(Required) Synchronizing Time

It is important that you synchronize the clock on the Workload Optimization Manager instance with the other devices on the same network. By default, the Workload Optimization Manager server is configured to synchronize with any one of the following time servers:

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

To synchronize with these servers, your installation of Workload Optimization Manager must have access to the internet. If your environment restricts internet access, then you have to configure synchronization with a time server on your network.

In all cases, you should verify that the Workload Optimization Manager clock is properly synchronized. To check the system clock:

1. Open an SSH terminal session to your Workload Optimization Manager instance.

   Log in with the System Administrator that you set up when you installed Workload Optimization Manager:

   - Username: `turbo`
   - Username: `[your_private_password]`

2. Verify your time settings.

   Execute the `date` command. You should see results similar to:

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

   To verify the time, you can execute the command, `timedatectl`. The output should be similar to:

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

   This tells you whether you have NTP enabled, and whether it is currently synchronized, along with other time synchronization information.

If the output is correct *and* your environment has access to the internet, you can assume the system clock is synchronized.

If the output is incorrect, or if you need to configure synchronization with a time server on your network, you must configure `chrony` on the server instance.

To set up `chrony` on your Workload Optimization Manager instance:

1. Open an SSH terminal session to your Workload Optimization Manager instance.
2. Open the `chrony` configuration file.

   For example, execute the command: `sudo vi /etc/chrony.conf`
3. Specify the time servers that you want to use in your environment.

   The `chrony` file includes the following statements to configure time servers:

   ```
   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
   ```

   Enter statements for the servers you want to use. Then delete or comment out the statements that you do not want to use.

   Specify a time server via the following command syntax:

   ```
   server My_Time_Server_Name iburst
   ```

4. Save the file.
5. Restart the chrony service.

   Execute the command: `sudo systemctl restart chronyd`
6. Verify that your time is correct.

   Execute the `date` command. You should see results similar to:

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

To verify the time has been synchronized, you can execute the command, `timedatectl`. The output should be similar to:

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

To verify the time, compare the `date` output with the output from a known UTC time server.

If the output is correct you can assume the system clock is synchronized.

If the output is incorrect, contact your support representative.

(Important) Verifying your MariaDB Version

For its default historical database, Workload Optimization Manager currently supports MariaDB version 10.5.16. This support includes comprehensive testing and quality control for Workload Optimization Manager usage of the historical database.

If you are running Workload Optimization Manager installed as a VM image (OVA or VHD), and using the database that is included in that image installation, then you must use version 10.5.16. If you are updating your version of Workload Optimization Manager (instead of installing it for the first time), then you should make sure you are using the correct version of MariaDB with your installation.

This section shows you how to check the version of MariaDB on your VM image installation of Workload Optimization Manager. Also, if you have used the update script to updated your Workload Optimization Manager to version 3.1.5 or later, you can use the steps in this section to update your MariaDB.

IMPORTANT:
It is a requirement that you run MariaDB version 10.5.16 or later. Workload Optimization Manager can operate with other versions of MariaDB. However, it is fully tested to operate with MariaDB version 10.5.16.

Because of a known issue, ***you must never use*** *MariaDB versions 10.5.14, 10.5.15, 10.6.7, 10.7.3, or 10.8.2.*

Workload Optimization Manager also supports MySQL 5.7.x, deployed as a custom installation.

When you initially installed Workload Optimization Manager, that installation included MariaDB running a specific version. As you update your Workload Optimization Manager version, the MariaDB version remains the same. The first release of Workload Optimization Manager that included MariaDB 10.5.16 is 3.3.6. If you initially installed an earlier version, and you have not explicitly updated your MariaDB to 10.5.16, then you must do it now.

For VM image installations, it is possible to configure the installation to use a remote database (external to the VM). For such deployments, you must manage the database versioning yourself. If you are using a remote MariaDB instance, we recommend that you use version 10.5.16. For a remote MySQL, you should use version 5.7.x.

For installations on a Kubernetes cluster (not deployed as a Workload Optimization Manager VM image), if you are using MariaDB we recommend that you use version 10.5.16. You can find a download package at: `https://archive.mariadb.org/mariadb-10.5.16/yum/centos7-amd64`. For MySQL, you should use version 5.7.x. For such deployments, you must manage the database versioning yourself.

Checking your MariaDB Version

To check the version of MariaDB running on your Workload Optimization Manager OVA:

1. Open an SSH terminal session to your Workload Optimization Manager instance.

   Log in with the System Administrator that you set up when you installed Workload Optimization Manager:

   - Username: `turbo`
   - Username: `[your_private_password]`

2. Check the MariaDB version.

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

The output should be similar to:

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

If the version is lower than `10.5.16-MariaDB`, then you must update your database.

If your version is equal to or higher than `10.5.16-MariaDB` you should not perform the update steps below.

Updating your MariaDB

If you are using Workload Optimization Manager installed as a VM image, and you are using the default MariaDB that was installed with that image, you must run MariaDB version 10.5.16.

To update your MariaDB on your Workload Optimization Manager VM:

1. Open an SSH terminal session to your Workload Optimization Manager instance.

   Log in with the System Administrator that you set up when you installed Workload Optimization Manager:

   - Username: `turbo`
   - Password: `[your_private_password]`

2. Ensure the VM is mounted on the Workload Optimization Manager update ISO image.

   NOTE:
   When you complete a Workload Optimization Manager update, the system automatically unmounts the ISO image. To perform the MariaDB update, your Workload Optimization Manager instance ***must be mounted*** on the same ISO image that you used to update it to version 3.1.5 or later.

   For information about offline updates and mounting the ISO image, see Offline Update *(on page 66)*.

3. Execute the MariaDB update script.

   Before you execute the script, you will need to know the MariaDB password. By default, this password is `vmturbo`.

   a. Make the script executable.

      NOTE:
      If you perform offline updates and have already run the offline upgrade script, the updated `mariadbUpgrade.sh` is copied from the ISO image to the `/opt/local/bin` directory with the executable flag enabled. You can skip this step and execute the script as noted below.

      ```
      sudo chmod +x /opt/local/bin/mariadbUpgrade.sh
      ```

   b. Execute the database update script:

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

   The script updates the version of MariaDB. It also increases size limits for the allowed packets, and buffer and log sizes for the innodb. The script output should include the following (where `Total Memory` and `buffer pool size` can vary depending on your VM configuration):

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

4. Verify the updated MariaDB version.

   When the script completes, you should be running version 10.5.16. To check the version, execute the following command:

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

   The output should be:

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

5. Scale up the Workload Optimization Manager platform's pods.

   To update the database, the script scales down your platform pods. When it completes, the script displays the following prompt:

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

   After you verify that the correct version of MariaDB is running, scale up the platform:

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

Increasing Available Disk Space

A standard installation of Workload Optimization Manager on a VM image includes a MariaDB database server for historical data. If you enable Embedded Reports, the platform also uses TimescaleDB Postgres database to manage the reports data. For various reasons, you might find that the default storage capacity for your database services is not sufficient. In that case, you need to increase the available storage capacity.

A common reason to increase this capacity is to accommodate estimated needs for Embedded Reports. The storage requirements for Embedded Reports can change over time as your environment changes, or as you increase the number of targets you configure your your Workload Optimization Manager installation. For information about estimating Embedded Reports requirements, see Embedded Reports Storage Requirement Estimates *(on page 39)*.

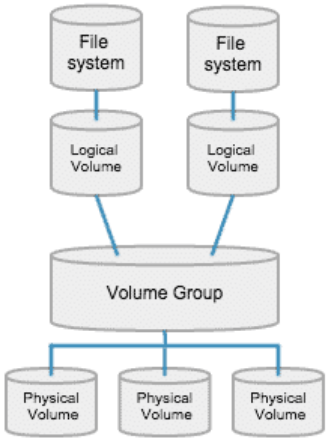
A summary of the steps you will perform is:

- Add a new disk to the VM
- Rescan the scsi devices
- Create a new LVM partition
- Create a physical volume (pv)
- Add the pv to the existing volume group (vg)
- Extend the logical volume (lv)
- Extend the file system to use the new lv
- To increase storage for Embedded Reports, increase the XFS quota

  To increase space for MariaDB, you do not need to perform this step.

Logical Volume Management for Workload Optimization Manager Storage

The platform uses Logical Volume Management (LVM) to manage the VM disks. To increase database storage, you should add a new disk to the VM, and then use it to extend the LVM logical volume, `/dev/turbo/var_lib_mysql`. This logical volume serves both the historical database and the Embedded Reports database.



Increasing Storage – Procedure

To increase the storage space available to your databases:

1. Add a new disk to the VM.

   Use the steps for your VM datacenter to add a new disk to the VM. Workload Optimization Manager installs as a VMware or a Hyper-V VM. Refer to the documentation for your hypervisor for the steps to add a new disk.
2. Open an SSH terminal session to your Workload Optimization Manager instance.

   Log in with the System Administrator that you set up when you installed Workload Optimization Manager:

   - Username: `turbo`
   - Password: `[your_private_password]`

3. Rescan the scsi devices.

   To make sure the new disk is available, rescan the scsi devices and then list your block devices.

   To scan the devices, execute:

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

   To check for the new disk, execute:

   ```
   lsblk
   ```

   The new disk should appear with a name similar to `/dev/sdc`. If you don't see the new disk, try this alternative to force a rescan:

   - Check the number of scsi host devices that are on your VM:

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

     You should see a list of devices, such as `host0, host1, host2... hostn`
   - Scan each device

     For each device execute the command (where **<hostn>** is a numbered host device such as `host0` or `host1`):

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

   - List the block devices

     Execute `lsblk` again to list the block devices.
4. Create a new LVM partition.

   Assuming the new disk is named `/dev/sdc1`, execute the command:

   ```
   cfdisk /dev/sdc1
   ```

   Then execute the operations:

   - `new`
   - `primary`
   - `confirm size`
   - `change type to 8E`
   - `write`
   - `quit`
5. Create the Physical Volume (pv).

   Assuming the new disk is named `/dev/sdc1`, execute the command:

   ```
   pvcreate /dev/sdc1
   ```
6. Add the new pv to the existing Volume Group.

   Assuming the new disk is named `/dev/sdc1`, execute the command:

   ```
   vgextend /dev/turbo /dev/sdc1
   ```
7. Extend the Logical Volume (lv) to use the free space in the new pv.

   First list the physical extents (PE) that are available. Execute the command:

```
vgdisplay
```

You should see results similar to:

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

In this example, `128000` is the amount to extend the lv. For this example, execute the command:

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

8. Extend the XFS file system to use all the current lv space.

   Before you extend the XFS, view the free disk space and record the number. To verify that you have increased the available space, you will compare this value to the free space after you have extended XFS. Execute the command:

   ```
   df -h
   ```

   Then extend the XFS capacity:

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

   Then list the updated free disk space and compare it to your original number:

   ```
   df -h
   ```

9. If you are increasing capacity for Embedded Reports, extend the XFS quota for the TimescaleDB.

   To increase space for MariaDB, you do not need to perform this step.

   To increase capacity for the Timescale DB, you need to increase the quota for that process by the amount you want. The quota name is `Postgresql`.

   For example, assume you added a 400 GB volume, and the current `Postgresql` quota is 400 GB. In that case, you could increase the quota to 800 GB. Following this example, execute the command:

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

   To see the current quotas set for `/var/lib/dbs`, execute the command:

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

(Optional) Enforcing Secure Access Via LDAP

If your company policy requires secure access, you can use a certificate with your LDAP service to set up secure access for your users. For example, you can configure Active Directory (AD) accounts to manage *External Authentication* for users or user groups. The user interface to enable AD includes a **Secure** option, which enforces certificate-based security. For more information, see "Managing User Accounts" in the *Workload Optimization Manager User Guide*.

If your LDAP service uses a Certificate Authority (CA), then the certificate signed by that CA should support this feature as it is. Simply turn on the **Secure** option when you are setting up your AD connection.

If your LDAP service uses a self-signed certificate, then you must install that certificate on the Workload Optimization Manager authorization pod. The steps you will perform include:

- Get the certificate from your LDAP server
- Import the certificate to the platform's TrustStore
- Add the certificate to the Workload Optimization Manager platform's authorization pod
- Enable the TrustStore in the Workload Optimization Manager platform's Operator chart

Installing a Self-signed Certificate

To set up secure access:

1. Open an SSH terminal session to your Workload Optimization Manager instance.

   Log in with the System Administrator that you set up when you installed Workload Optimization Manager:

   - Username: `turbo`
   - Password: `[your_private_password]`

2. Download your LDAP Server certificate to the Workload Optimization Manager instance.

   Acquire a certificate from your LDAP administrator, and download it to the Workload Optimization Manager platform. For example, you can download it to the file `/tmp/ldapserver.crt`:

3. Import the `.crt` file to the Workload Optimization Manager TrustStore.

   This step modifies the `cacerts` file on the Workload Optimization Manager platform.

   NOTE:
   To import a certificate to the Workload Optimization Manager TrustStore, you must use the `keytool` utility. To install this utility, execute the command:

   ```
   sudo yum install java-1.8.0-openjdk
   ```

   This installs the utility in `/usr/bin/keytool`.

   If an alias for an LDAP certificate already exists, delete that certificate. For example, assuming the alias `ldapcert1`, execute the following command:

   ```
   keytool -delete -alias ldapcert1 -keystore cacerts -storepass changeit
   ```

   Then use the following command to import your new certificate to the TrustStore:

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

4. Create an `auth` secret from the `cacerts` file.

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

5. Open the secrets file for editing.

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

6. Edit the file to make it a valid yaml file.

   a. Indent every line of the certificate by four spaces.

      When you created the file, you concatenated the contents of the certificate. The first step is to indent the certificate by four spaces. For example, in a `vi` editor, execute the following command:

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

   b. Add data fields to the secrets file.

      Add the following text to the top of the file:

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

   c. Save your changes.

   The completed file should be similar to:

```
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. Apply this secrets file to the platform environment.

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

8. Update the platform's Operator Chart to use the `cacerts` certificate that you created in the secrets file.
   a. Open the chart file for editing.

      Open the file, `/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml`.

   b. Add the certification secret as an authorization spec for the component options.

      In the chart file, find the `spec:` section. Within that section, find the `auth:` subsection.

      This should be the second subsection in `spec:`, after `global:`. If there is no `auth:` subsection, you can add it to `spec:`.

c. Add the certification secret to the file:

You will add the secret's path to a `javaComponentOptions:` statement within the `auth:` subsection. Add the path as a `-D` option. The `auth:` subsection should be similar to the following, with `auth` indented by two spaces and `javaComponentOptions` indented by four spaces:

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

d. Apply your Operator Chart changes to the Workload Optimization Manager platform.

Execute the following command:

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

This restarts the authorization component so it can use the new setting.

(Optional) Adding a Certificate for Securing the Turbonomic UI

If your company policy requires SSL connections via trusted certificate, Workload Optimization Manager enables you to install a trusted certificate from a known certificate authority.

Requesting a Certificate

The first step is to acquire a certificate. The following steps describe how to generate a certificate request.

1. Open a shell terminal session.

   Open an SSH terminal session on your Workload Optimization Manager instance. Log in as `turbo`, and use the password that you created for the administration account in the installation steps above. For information, see the installation step, Set up the Workload Optimization Manager System Administrator account *(on page 9)*.
2. Change to the directory where you want to store the private key file.

   If your shell session is on your Workload Optimization Manager instance, you should use the `/opt/turbonomic` directory:

   ```
   cd /opt/turbonomic
   ```
3. Create and save the private key file.

   Execute the command to create a private key file.

   For this example, the private key file is named `myPrivate.key`

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

   You will need this file later. If you are in a session on your Workload Optimization Manager instance, you might want to copy the file to your local machine.
4. Create a file to contain the information that will generate the signed certificate request (CSR).

   ```
   vi certsignreq.cfg
   ```
5. Add the request data to the `certsignreq.cfg` file.

   In the file, insert the following code. For any fields marked by angle brackets (for example `<city>`), provide the indicated value. For example, your country, city, company, etc.

   ```
   [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>
```

NOTE:
For the CN field, specify the fully-qualified domain name of the Workload Optimization Manager instance.

Alternate names are other ways to access the Workload Optimization Manager instance. In the `[alt_names]` section, the value for the `DNS.1` field is required. For `DNS.1`, specify the fully-qualified domain name of the Workload Optimization Manager instance. Values for the `DNS.2` and `DNS.3` are optional. You can add more DNS.<n> fields if needed.

For example:


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

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

[req_ext]
subjectAltName = @alt_names

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

6. Write and quit the file.

   Press **esc**, type `:wq!`, and press **Enter**.

7. Generate the certificate request file.

   In this example, we name the file `myRequest.csr`.

   Execute the command:

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

8. Send the generated request file to your certificate authority.

   If you generated the file on your Workload Optimization Manager instance, you should transfer the file to your local machine. The path to the certificate request file on your remote machine is `/opt/turbonomic/myRequest.csr`.

   Your certificate authority will use this file to create the certificate for you.

   If your certificate authority gives you an encoding choice between DER and Base 64, choose **Base 64**.

9. When you receive the certificate, save it to disk.

   If you did not receive the certificate encoded in Base 64, you must convert it from DER to Base 64. Execute the following command, assuming the certificate is named `MyCertificate.crt`:

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

Installing the Signed Certificate in Workload Optimization Manager

Once you have obtained the signed certificate, you can install it on your Workload Optimization Manager instance. You will use the private key and certificate files you obtained when requesting the signed certificate:

- `myPrivate.key`
- `MyCertificate.crt`

To install the signed certificate:

1. Open an SSH terminal session on your Workload Optimization Manager instance.
2. Add the key and certificate data to your Workload Optimization Manager charts.yaml file.

   Open the file: `/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml`

   Find the section for `global` parameters. Under the `global` parameters, create the `ingress:secrets` section, and then create entries for `certificate`, `key`, and `name`.

   Your global parameters should be similar to the following:

   ```
   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
   ```

   For the fields you added:

   - `certificate:` This field holds the content of your `MyCertificate.crt` file. Open that file to copy its contents and paste them here.
   - `key:` This field holds the content of your `myPrivate.key` file. Open that file to copy its contents and paste them here.
   - `name:` This field is required, and the name must be `nginx-ingressgateway-certs`.

3. Apply the changes you made to the CR file.

   Execute the command:

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

4. Restart the `nginx` pod.

   To require a certificate for HTTPS access, you must restart the `nginx` pod:

   a. Get the full name of the pod.

      Execute the command `kubectl get pods -n turbonomic`. In the output, look for the entry for `nginx`. You should find an entry similar to:

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

   b. Restart the pod.

      Execute the following command, where **<UID>** is the generated ID for the pod instance:

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

   This should restart the nginx pod. After restart, Workload Optimization Manager will then require a certificate for HTTPS access.

(Optional) Adding Additional CA Certificates for Probes

If your targets require SSL connections via trusted certificate, Workload Optimization Manager enables you to install a trusted certificate on the associated probe component.

The Workload Optimization Manager platform includes a number of probe components that it uses to connect to targets and discover their data. This procedure assumes setup for one component, the *Dynatrace* probe. You can use the same steps for other probes, providing a different Kubernetes Secret Name for each.

To install a certificate on a probe component, you must know the Kubernetes secret name for the given probe. This table lists the probes that you can configure, plus their secret names. If you must configure secure access for any probes not listed here, contact your support representative.

Probe	K8s Secret Name
mediation-appinsights	appinsights
mediation-aws mediation-awsbilling mediation-awscost mediation-awslambda	aws
mediation-azure mediation-azurecost mediation-azuresp mediation-azurevolumes	azure
mediation-azurebilling	azurebilling
mediation-azureea	azureea
mediation-dynatrace	dynatrace

Probe	K8s Secret Name
mediation-newrelic	newrelic

Installing the Signed Certificate on the Probe Component

This procedure assumes you already have a valid `.crt` file. If you do not have the certificate file, ask your networking team to generate one for you.

Once you have obtained the signed certificate, you can install it on your probe instance. You will use the certificate file you obtained:

```
MyCertificate.crt
```

To install the signed certificate on a probe:

1. Copy the certificate from your local machine to the Workload Optimization Manager instance.

   Use SCP to copy the `MyCertificate.crt` from your local machine to the `/tmp` directory on the instance.
2. Open an SSH terminal session on your Workload Optimization Manager instance, using the `turbo` user account.
3. Obtain the trust store from the probe component.

   First, get the ID for the pod that runs the probe. To get the ID, execute the command:

   ```
   kubectl get pods
   ```

   This lists the pods running in the Workload Optimization Manager platform, including their IDs. Record the ID of the pod you want to configure.

   To get the CA trust store, execute the following command, where **<Probe-Pod-Id>** is the ID you recorded:

   ```
   kubectl cp <Probe-Pod-Id>:etc/pki/ca-trust/extracted/java/cacerts cacerts
   ```
4. Import the certificate into the pod's keystore.

   As part of this step, you will ensure that the certificate is in Base64 format and you will create a yaml file using the K8s Secret Name for the probe. While still in the bash session, execute the following commands:

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

     Where **`MyCertificate.crt`** is the name of the certificate that you acquired.
   - `base64 cacerts > <Secret_Name>-secrets.yaml`
     Where **`<Secret_Name>-secrets.yaml`** is the yaml file you will create using the K8s Secret Name for the probe.

     For example, assume you are enabling SSL for the Dynatrace probe. In that case, the secret name is `dynatrace`, and you would create the yaml file:

     ```
     dynatrace-secrets.yaml
     ```
5. Update the `<Secret_Name>-secrets` yaml file you just created.

   a. While still in the bash session on the Workload Optimization Manager server, open the yaml file created in the previous step in a vi editor session:

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

   b. Align the base64 data to the yaml format.

      Type `:` to enter the command mode. For the command, type the following, where the whitespace token is four space characters:

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

Press RETURN to execute the command. Then save and exit the vi editor.

c. Then add the following content to the file above your Base64 data:

```
apiVersion: v1
kind: Secret
metadata:
  name: <Secret_Name>
data:
  cacerts: |
    xxxxxxxxxxxxxxxxxxxxxxxx
    xxxxxxxxxxxxxxxxxxxxxxxx
```

Your Base64 data should be in the `cacerts` section in place of the `xxx` characters in the example above.

6. Apply the yaml file to the Workload Optimization Manager platform.

   Execute the following commands, where **<Probe-Pod-Id>** is the ID you recorded, and :

   - `kubectl apply -f <Secret_Name>-secrets.yaml`

7. For each probe that you configure with a SSL certificate, add an entry in the chart_v1alpha1_cl_cr.yaml file.

   a. With a shell session running on the Workload Optimization Manager platform, open the following file in a text editor:

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

   b. Search the file for the entry for the probe that you are configuring. Use the probe names listed in the table above. For example, if you are configuring the Dynatrace probe, find the entry for `mediation-dynatrace`. If the entry does not exist in the file, you can add it to the `spec:` section at the same level as `global:`, with the probe entry indented by two spaces and `javaComponentOptions` indented by four spaces.

   c. Underneath the probe entry, add the following entry for `javaComponentOptions:`

   ```
   javaComponentOptions: -Djavax.net.ssl.trustStore=/etc/targets/cacerts
   ```

   For example, if you are configuring the Dynatrace probe, the entry should be similar to:

   ```
   mediation-dynatrace:
       javaComponentOptions: -Djavax.net.ssl.trustStore=/etc/targets/cacerts
       resources:
         limits:
           memory: 2Gi
   ```

   d. Save and exit the chart_v1alpha1_cl_cr.yaml file.

   e. Apply the changed file to your Workload Optimization Manager platform.

   Execute the command:

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

(Optional) Modifying the Certificates for Cluster Manager

For installations behind a firewall, to upload diagnostics from the `clustermgr` component, you must modify its certificates.

These steps to modify the certificates on `clustermgr` assume you have already generated the certificates that you want to add to the cluster manager.

1. Open an SSH terminal session on your Workload Optimization Manager instance.

   Log in with the System Administrator that you set up when you installed Workload Optimization Manager:

   - Username:

     ```
     turbo
     ```

   - Password:

     ```
     [your_private_password]
     ```

2. Get the full name of the clustermgr pod.

   Execute the command:

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

   The result should be similar to:

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

   In this example, `clustermgr-5f487f58f-tf84b` is the full name of the pod, and `5f487f58f-tf84b` is the POD_ID.

3. Save a copy of the pod's current ca-bundle.crt file to /tmp.

   Execute the following command, where <POD_ID> is the ID you get from the pod's full name.

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

4. Add your certificates to the bundle.

   Repeat this command for each certificate, where <MY_CERT> is your certificate file.

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

5. Create a Kubernetes secret for the modified certificates.

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

6. Open the cr.yaml file for editing.

   For example:

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

7. Modify the cr.yaml file to use this secret.

   Add the following to the file:

   ```
   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. Save your changes and apply the cr.yaml file.

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

If you watch the log with `grep ^clustermgr`, then you should see the appropriate curl command execute whenever you send diagnostics.

You can also check the .crt file in the cluster manager pod via the following commands, where <POD_ID> is the ID you got from the pod's full name:

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

(Optional) Enabling Embedded Reports

The Workload Optimization Manager platform includes an Embedded Reporting component that you can choose to enable when you install the platform. Use Embedded Reporting to understand application resource management trends, and to share insights with stakeholders via reports and dashboards.

Embedded Reporting runs as its own component, as part of the Workload Optimization Manager platform. This architecture enhances performance and reduces storage requirements. It stores a history of your managed environment and then presents selective snapshots of this history via a set of standard dashboards and reports. You can create your own dashboards and reports to focus on other areas of concern.

The method you use to enable embedded reports depends on the version status of your Workload Optimization Manager instance, as follows:

- Script Interface *(on page 33)*

  You have installed Workload Optimization Manager version 3.0.0 or later, as a new VM image (OVA or VHD; see Installing on a Virtual Machine Image *(on page 8)*). In this case, you can execute the `enable_reporting.py` script to set up embedded reporting.
- Editing the Workload Optimization Manager cr.yaml File *(on page 35)*

  You have installed Workload Optimization Manager as a Kubernetes cluster, not using the VM image for your Workload Optimization Manager installation.

  In these cases, you manually edit the `charts_v1alpha1_xl_cr.yaml` file for your installation of Workload Optimization Manager.

Script Interface

If you have installed Workload Optimization Manager as a VM image for version 3.0.0 or later, the script to enable Embedded Reports is already installed on your installation at:

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

To execute this script:

1. Open an SSH terminal session to your Workload Optimization Manager instance.

   Log in with the System Administrator that you set up when you installed Workload Optimization Manager:

   - Username: `turbo`
   - Username: `[your_private_password]`

2. Navigate to the script directory.

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

3. Execute the script.

   ```
   ./enable_reporting.py
   ```

   The script prompts you for two passwords:

   - The Grafana admin password.

     This password enables access to Grafana from external URLs and also from the extractor component that feeds data to Grafana.

     Do not use special characters.

     IMPORTANT:
     This is the only time that you should change the Grafana Admin password.

     If you change the Grafana Admin password subsequent to completing this step, the Embedded Reporting components cannot communicate properly with the other components in the platform. If you have made a subsequent change to this password, contact your support representative.

   - The Grafana database password.

     This password enables communication between Grafana and the Postgres database that stores the reporting data.

   After you supply the passwords, the script displays a confirmation message similar to:

   ```
   Succesfully applied new changes to /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alp
   ha1_xl_cr.yaml.
   Backup written to /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml.bak
   ```

   This indicates that the script successfully updated the Workload Optimization Manager configuration. The script then applies the changed configuration to enable the Embedded Reports feature. It should display messages similar to:

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

4. Verify your installation.

   Execute the command:

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

   If Embedded Reports are successfully enabled, the script output should be:

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

Editing the Workload Optimization Manager cr.yaml File

These instructions describe how to locate and edit the `charts_v1alpha1_xl_cr.yaml` for the VM image installation. If you installed on a Kubernetes node cluster, then the file can be in a different location.

To enable Embedded Reports, you will:

- Enable the processes that implement the embedded reporting.
- Update the API pod to enable new search and data ingestion capabilities.
- Double-check the installation.
- Enable email subscriptions (optional).

You must enable the Grafana Exporter, TimescaleDB, and data extraction processes. To do this, edit the `charts_v1alpha1_xl_cr.yaml` file.

1. Open an SSH terminal session to your Workload Optimization Manager instance.

   Log in with the System Administrator that you set up when you installed Workload Optimization Manager:

   - Username: `turbo`
   - Username: `[your_private_password]`
2. Open the following file in a text editor:

   ```
   /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
   ```
3. Specify the IP address of the Workload Optimization Manager instance for external access to the TimescaleDB database.

   In the `global:` section of the file, add the following line, where **`<Platform_IP>`** is the IP address of your instance:

   ```
   global:
     externalTimescaleDBIP: <Platform_IP>
   ```
4. Enable the Grafana process.

   Find the `grafana:` section in the `crds/charts_v1alpha1_xl_cr.yaml` file, and uncomment the line, `enabled: true`.
5. Enable Postgres as the database type.

   Enabling Postgres sets persistent storage of historical data for Embedded Reports. In the `grafana:` section, find the subsection for `grafana.ini: database:` and uncomment the line, `type: postgres`.

   The changes you have made so far should be similar to:

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

   grafana:
       enabled: true
       adminPassword: admin
       grafana.ini:
         database:
           type: postgres
           ...
   ```
6. Change the admin and database passwords.

   It is good practice to change any passwords, and not keep their default values.

   IMPORTANT:
   Use only alpha-numeric characters for these passwords.

   These passwords enable communication between the various Embedded Reports components. Some of the components only accept alpha-numeric characters. If you use special characters, then the components will not be able to communicate. Further, the steps to correct these passwords require assistance from your Support engineer.

To set the passwords:

- Set the Grafana admin password.

  This password enables access to Grafana from external URLs and also from the extractor component that feeds data to Grafana. In the `grafana:` section, change the value of `adminPassword`.

  Do not use special characters.

  Assume your password is `MyNewGrafanaPassword`. Then you would set `adminPassword:` **`MyNewGrafanaPassword`**

  IMPORTANT:
  This is the only time that you should change the Grafana Admin password.

  If you change the Grafana Admin password subsequent to completing this step, the Embedded Reporting components cannot communicate properly with the other components in the platform. If you have made a subsequent change to this password, contact your support representative.
- Set the Grafana database password.

  This password enables communication between Grafana and the Postgres database that stores the reporting data. In the `grafana:` section, find the subsection for `grafana.ini: database: password:` and change the password value.

7. Enable the three Embedded Reports processes.

   Just after the `properties:` section that you added, and at the same level to it, add the following entries to enable the reporting processes:

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

   It is important that you align these entries with the indentation for the `grafana:` section and the `properties:` section. The changes you have made should now be similar to:

```
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. When you are done editing the `charts_v1alpha1_xl_cr.yaml` file, save and apply your changes.
   - Save your changes and quit the text editor.
   - Apply the changes.

     Execute the command:

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

   - Delete the api and extractor pods.

     Deleting these pods triggers them to restart, which loads the changes you made.

     To get the full pod names, execute the command, `kubectl get pods -n turbonomic`. Then find the two entries for the pods that begin with `api` and `extractor`. For example, assume the entries are:

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

     Then you would execute the commands:

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

9. Verify your installation.

   To double-check the installation:

   - Verify that the Embedded Reports pods are running.

     To verify that the pods are running, execute `kubectl get pods -n turbonomic`. The output should include entries similar to:

     ```
     NAME                                        READY   STATUS    RESTARTS
     extractor-7759dbcb47-vs6hr                  1/1     Running   0
     grafana-84ccb4bfb-l7sp7                     1/1     Running   0
     ```

   - Verify that Postgres is running.

     The Postgres database should be running as a daemon on the Workload Optimization Manager server machine. To check the status, execute the command:

     `sudo systemctl status postgresql-12.service`.

     You should see output similar to:

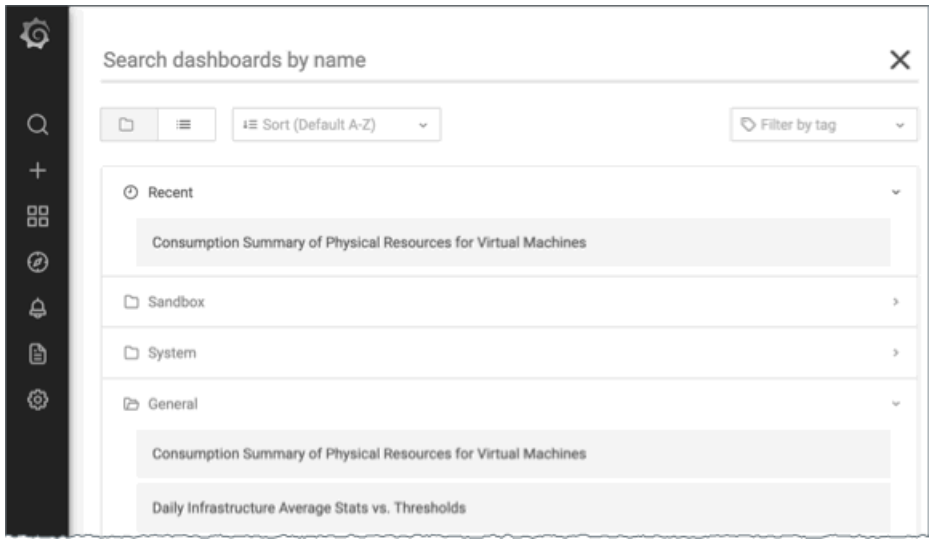
     ```
     postgresql-12.service - PostgreSQL 12 database server
     Loaded: loaded (/usr/lib/systemd/system/postgresql-12.service; enabled; vendor preset: disabled)
     Active: active (running) since Wed 2020-07-29 06:39:43 UTC; 14h ago
          Docs: https://www.postgresql.org/docs/12/static/
     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
```

Navigating to the Embedded Reports Page

After you have completed the steps to enable Embedded Reports, open the Workload Optimization Manager user interface and click **Reports** in navigation bar.

This opens dashboards and charts in a new browser tab.



You can search for specific dashboards or browse folders to find the dashboards you want. You can also create custom reports.

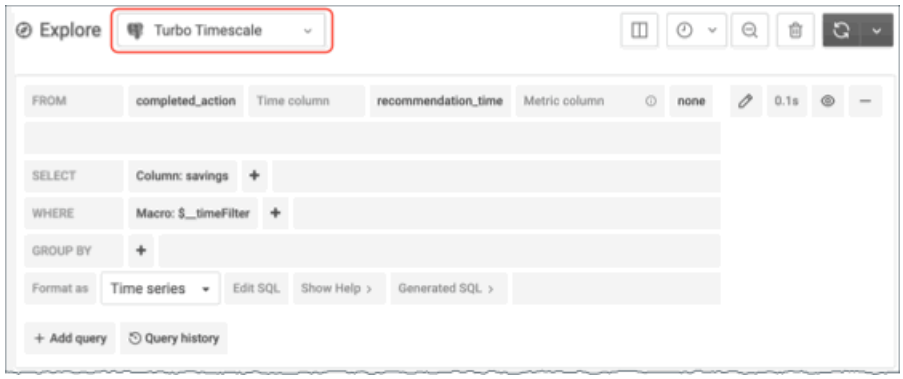
Dashboards and charts are powered by the Grafana® observability platform. With Grafana, it's easy to navigate the existing dashboards, and to make your own charts and dashboards with no coding required. If you are new to Grafana and need help getting started, read the documentation available at:

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

NOTE:
Starting with version 3.4.1, Workload Optimization Manager no longer has an agreement to use the Enterprise license of Grafana, and ships the open-source Community license instead. After you update an earlier version of Workload Optimization Manager to version 3.4.1 or later, Grafana automatically switches to the Community license and you should be able to use Embedded Reporting as usual. However, be aware that the Community license does not support PDF reports. Please contact your Workload Optimization Manager representative for guidance on PDF reports.

Creating Custom Reports

To create custom reports, you must run SQL queries against the Embedded Reports database (`Turbo Timescale`).



The database schema includes certain tables against which you can run queries.

(Optional) Report Editing

To create and edit reports, a user must have Report Editor privileges. This user can have any role, as long as it is not a *shared* or *scoped* role. Only one user per Workload Optimization Manager instance is allowed to have Report Editor privileges (by default, the local **administrator** user). You can grant these privileges to another user.

To create a user account with Report Editor privileges:

1. In the Workload Optimization Manager user interface, navigate to **Settings / User Management**.
2. Choose the user account that you want to configure as a Report Editor. You can either edit an existing account or create a new one.
3. Choose a role for the user. The user can have any role, as long as it is not a *shared* or *scoped* role.
4. Under **Options**, choose **DESIGNATE AS REPORT EDITOR**.
5. Set any other properties for the user account that you want, and save the user account.

NOTE:
It can take up to 30 minutes before the Reports page shows the Report Editor username. This usually occurs if you have changed the Report Editor multiple times.

To access reports from the user interface, a user must have the Administrator or Site Administrator role, or a non-administrator role without a defined scope. For example, a user with the Observer role but without a scope can access reports.

The default Shared Observer and Shared Adviser roles require scopes, so users with these roles cannot access reports.

Embedded Reports Storage Requirement Estimates

The Embedded Reports feature uses a TimescaleDB server to manage the chart data. This is a PostgreSQL server running with the TimescaleDB extension. You must configure the datastore for your Workload Optimization Manager instance so it has enough space to support the TimescaleDB requirements.

When you initially enable Embedded Reports, you should estimate the storage you will need, and configure the platform storage accordingly. If you have already enabled Embedded Reports, you should check your current storage configuration and decide whether it meets your needs now and into the future.

The storage that your TimescaleDB requires depends on:

- Data retention period

  How long to store the TimescaleDB data.
- The size of your environment

  The count of entities Workload Optimization Manager manages in your environment. This count changes over time. You should think of it as the average number of entities in your environment over the given data retention period.

Also note that increased entity count increases the data requirement, as does other activity. Storage requirements can increase over time for reasons such as:

- You add entities such as workloads, application components, storage, or hosts to your environment.
- You configure new targets.

Storage Estimates Lookup Tables

We have investigated the TimescaleDB storage requirements for different topologies and retention periods. The following table lists the estimates that we have calculated. Please be aware that your environment could have different requirements.

	Number of Entities						
Retention Period	10k	25k	50k	100k	250k	500k	1000k
6 months	36GB	91GB	182GB	364GB	910GB	1.8TB	3.6TB
1 year	72GB	181GB	361GB	723GB	1.8TB	3.5TB	7TB
2 years	144GB	361GB	721GB	1.4TB	3.5TB	7.2TB	14TB

Note that the default installation grants a disk quota of 200GB to the TimescaleDB. For the default installation, we estimate that the database can support the following entity counts:

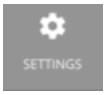
Retention Period	Entity Count
6 months	55k
1 year	27k
2 years	14k

Setting the Data Retention Period

By default, Workload Optimization Manager sets the retention period for Embedded Reports to be 365 days. You can see the currently set retention period, and change it in the Workload Optimization Manager user interface.

To execute these actions, navigate to the Maintenance Options page:

1. Navigate to the Settings Page.

   Click to navigate to the Settings Page.
2. Choose Maintenance Options.

   Maintenance Options
3. Set the data retention period for Embedded Reports.

   In the **Data Retention** group of controls, find the field for **Saved Reporting Data**. This displays the current data retention period for Embedded Reports, in days. The default is 365 days.

   To change the retention period, enter a different number of days, and then click **Apply Settings**.

Increasing Storage Capacity for TimescaleDB

If you estimate the storage requirements for Embedded Reports *after* you have installed Workload Optimization Manager, you might learn that you need to increase the storage capacity that is available to the TimescaleDB.

The platform uses Logical Volume Management (LVM) to manage the VM disks. To increase database storage, you should add a new disk to the VM, and then use it to extend the LVM logical volume, `/dev/turbo/var_lib_mysql`. This logical volume

serves both the historical database and the Embedded Reports database. When you have done that, you will increase the quota for the TimescaleDB.

For more information, see Increasing Available Disk Space *(on page 21)*

Estimating Entity Count

To get a sense of entity count in your environment, search the Workload Optimization Manager log file for an INFO message that lists the number of entities for each discovery cycle. You can search for the string `INFO [Stages$BroadcastStage]`. The INFO string should be similar to:

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

This example shows that you have 1505 entities in the topology. You should consider how your inventory changes over time. For example, you can check the entity count over time to see whether it increases regularly.

(Optional) Enabling the Data Exporter

To support Data Export, Workload Optimization Manager provides an extractor component that can stream data to a standard format. You can load that data into search and analytics services such as Elasticsearch.

To enable the Data Exporter, you must:

- Enable the extractor component.

  The extractor is a component that runs as part your Workload Optimization Manager installation. The extractor is not enabled by default.
- Deploy a connector that delivers the extractor's stream to your data service.

  The extractor publishes Workload Optimization Manager data as Kafka topics. The connector enables your data service to consume the data topic. This document includes a deployment file for a sample Elasticsearch connector.

Enabling the Extractor Component

The first step to enabling the Data Exporter is to enable the extractor component. To enable the extractor:

1. Open an SSH terminal session to your Workload Optimization Manager instance.

   Log in with the System Administrator that you set up when you installed Workload Optimization Manager:
   - Username: `turbo`
   - Username: `[your_private_password]`
2. Edit the cr.yaml file to enable the extractor component.

   In the same SSH session, open the cr.yaml file for editing. For example:

   ```
   vi /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
   ```
3. Edit the entry for the extractor component.

   NOTE:
   If you have enabled Embedded Reporting, then the extractor component will already be enabled (set to `true`).

   You should understand that it is possible to enable the Data Exporter without enabling Embedded Reports, just as it is possible to enable Embedded Reports without enabling the Data Exporter.

   Search for the extractor entry in the cr.yaml file. It should appear as:

   ```
   extractor:
       enabled: false
   ```

   Change the entry to `true`.
4. Edit the entry for the extractor properties.

   Search for the extractor entry in the cr.yaml file. It should appear as:

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

Change the entry to `true`.

5. Save and apply your changes to the platform.

   After you save your changes, use `kubectl` to apply the changes:

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

6. Verify that the extractor component is running.

   Give the platform enough time to restart the components. Then execute the command:

   `kubectl get pods -n turbonomic`

   You should see output similar to the following:

   ```
   NAME                                        READY   STATUS    RESTARTS
   ...
   extractor-5f41dd61c4-4d6lq                  1/1     Running   0
   ...
   ```

   Look for an entry for the `extractor` component. If the entry is present, then the extractor component is installed and running.

Deploying a Connector

The extractor publishes Workload Optimization Manager data as Kafka topics. To load this data into a search and analysis service, you must deploy a connector to that service. For example, if you want to load the data into Elasticsearch, then you must deploy an Easticsearch connector.

You deploy the connector in the same Kubernetes node that runs the Workload Optimization Manager platform. To do this, create a Kubernetes *Deployment* that declares the pods you need for the connector. Below, you can see a sample deployment of a connector to Elasticsearch.

To deploy the connector, you create a deployment yaml file on the same host that is running the extractor component, and execute the command:

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

Where **`<MyConnectorDeployment.yaml>`** is the name of the deployment file.

Assume the name of the deployed pod is `es-kafka-connect`. To verify that the connector is running, execute `kubectl get pods -n turbonomic`. You should see output similar to:

```
NAME                                        READY   STATUS    RESTARTS
...
es-kafka-connect-5f41dd61c4-4d6lq           1/1     Running   0
...
```

After you deploy the connector, wait for a cycle of Workload Optimization Manager analysis (approximately ten minutes). Then you should be able to see the entities and actions from your Workload Optimization Manager environment, loaded as JSON in your data service.

Connector Deployment Sample

Assume that you want to deploy a connector to Elasticsearch so that service can process the exported data. For example, you could use Kibana with Elasticsearch to display data dashboards. Let's say you have:

- Deployed Elasticsearch to a VM on the network where you are running Workload Optimization Manager. The Elasticsearch host is visible from the Workload Optimization Manager Kubernetes node. You will specify this host address in the connector deployment.
- Set up an Elasticsearch index to load the Workload Optimization Manager data. You will specify this index in the connector deployment.

The following listing is a deployment that uses a Logstash image to collect the extractor data and pipe it to the Elasticsearch host. The deployment also sets up storage volumes, configures the input from the extractor, and configures output to the Elasticsearch instance.

As you go over the listing, pay attention to the following:

- The location of the Elasticsearch host and the login credentials:

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

  Logstash will use the following environment variables:

  - ES_HOSTS: to identify where to pipe the exported data.
  - ES_USER: to identify the user account on Elasticsearch.
  - ES_PASSWORD: for the account login. This connector example assumes that you have stored the Elasticsearch password as a Kubernetes Secret.

  Logstash uses the ES_HOSTS environment variable to identify where to pipe the exported data.
- The name of the Kafka topic:

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

  The Logstash input configuration expects a single topic named `turbonomic.exporter`.

- The Logstash output configuration is to the Elasticsearch server that is identified by the ES_HOSTS environment variable. You specify your own Elasticsearch index in place of **<MyElasticsearchIndex>**

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

Sample Listing: Elasticsearch Connector

This listing is a sample of a deployment file that can work to create an Elasticsearch connector for the Data Exporter. Note that you will need to change some settings, such as username and password. You also might need to specify ports and other settings to make the connector comply with your specific environment.

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

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

```
ports:
  - name: "25826"
    port: 25826
    targetPort: 25826
selector:
  app: elasticsearch-kafka-connect
```

(Optional) Changing the IP Address of the Platform Node

For standard installations of Workload Optimization Manager (installed as a VM image), you might need to change the platform's IP address. For example, if you have to move the VM then you might need to assign it a different address. If you must change the IP address of the platform, you can use the supplied scripts.

NOTE:
You should change the IP address of your Workload Optimization Manager installation as seldom as possible. This is a sensitive action that can impact unforeseen dependencies.

You should use these steps to change the IP address only for Workload Optimization Manager version 3.0.0 or later. If you must change your IP address and you cannot update to version 3.0.0 or later, contact your support representative.

To change the IP address of the Workload Optimization Manager VM:

1. Get your information ready.

   Identify both the current IP address for your platform, and the new IP address you will use.

   You must also know the credentials to open a shell session on the VM and run commands.
2. Create a full snapshot of the VM.

   It is important to make a full snapshot of your installation before you try to modify its IP address.
3. Change the VM's IP address.

   The Workload Optimization Manager VM includes the `ipsetup` script to perform this task.

   a. Open an SSH terminal session to your Workload Optimization Manager VM.

      Use the following credentials:

      - Username: `turbo`
      - Password: Give the password that you assigned the `turbo` account when you first installed the platform.

   b. Once the session is open, execute the `ipsetup` script:

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

      When the script runs it requests the following inputs.

      NOTE:
      You must provide values for these required fields. Otherwise the installation can fail or your VM can be unreachable:

      - **Required:** `Do you want to use DHCP or set a static IP...`

        Choose `static`
      - **Required:** `Please enter the IP Address for this machine`
      - **Required:** `Please enter the network mask for this machine`
      - **Required:** `Please enter the Gateway address for this machine`
      - **Required:** `Enter DNS Server(s) IP Address for this machine`

      You should make a note of the IP address that you provide.

c. Propagate your IP change through to the Kubernetes cluster on the VM.

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

d. Verify that the change is successful.

Log into the Workload Optimization Manager user interface for the newly located installation, and ensure that it displays correctly. You should review the Supply Chain, your groups, and your policies. You should also ensure that charts show data correctly.

When you are sure that the change is successful, you can remove the snapshot you made of the VM in its old location.

(Optional) Enabling and Disabling Probe Components

In Workload Optimization Manager, a probe is a platform component that connects to a target. It discovers the target's entities and loads them into the Workload Optimization Manager supply chain, and it can execute actions on the devices in the target environment. Workload Optimization Manager ships with a large number of probe components that you can use to connect Workload Optimization Manager with your environment.

When you first install Workload Optimization Manager, it enables a certain set of probes by default, and leaves other disabled. Each probe consumes resources in your Workload Optimization Manager installation. If there are any probes that you do not need, then you should consider disabling them. On the other hand, if there are disabled probes that you do need, you must enable them to put them into service.

NOTE:
As Workload Optimization Manager evolves, the set of delivered probes change. Also, from one version to the next, the set of probes that are enabled by default can change. When you update to a new version, the update does not change your probe configuration. An update to a newer version does not automatically enable any new probes in your deployment. If you want to take advantage of new probes in an update, then you must enable them manually.

Viewing the Current List of Available Probes

As you update your version of Workload Optimization Manager, more probes can come available with the update. However, the update does *not* modify your current configuration of enabled or disabled probes. This means that any new probes that come with an update will not be available to you by default.

To enable any new probes, you must first know the internal name for the probe. To get a list of probes that are *available* to your current version, you can view the contents of the `values.yaml` file.

1. Open an SSH terminal session on your Workload Optimization Manager instance.

   Log in with the System Administrator that you set up when you installed Workload Optimization Manager:

   - Username:

     ```
     turbo
     ```

   - Password:

     ```
     [your_private_password]
     ```

2. Display the list of available probes.

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

   The results should be similar to:

   ```
   customdata:
     enabled: false
   dynatrace:
   ```

```
  enabled: false
gcp:
  enabled: false
hpe3par:
  enabled: false
...
```

This list gives the internal names of the probes. If you want to add a new probe to your list of configured probes, you must use the internal name, and set `enabled: true`.

Viewing the Current List of Configured Probes

Your current installation of Workload Optimization Manager has a certain set of available probes. Some of these will be enabled, and it is likely that some probes are disabled. To View the current configuration of probes that are currently available, open the cr.yaml file for your Workload Optimization Manager installation and review the probe entries:

1. In the same SSH session, open the cr.yaml file for editing. For example:

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

2. Search for the list of probes

   This will include all the probes that are configured for your current installation. The list will be similar to:

   ```
   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
   ...
   ```

   This list identifies all the probes that are currently configured for your installation, and shows whether they are enabled (`true`) or disabled (`false`).

NOTE:
This list of probes is not identical to the list of probe *pods* that are running in your installation. Some probes use multiple pods. Probe pod names use the following convention, where **`{ProbeName}`** is the probe internal name (in the lists above), and **`{NameExtension}`** is an optional extension to that name in case there are multiple pods for this probe:

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

For example, if you execute `kubectl get pods -n turbonomic`, the results can show the following for the `vcenter` probe:

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

Enabling/Disabling Probes

To enable or disable probes in Workload Optimization Manager, you will edit the cr.yaml file to add new probes and to change the values of the `enabled:` properties. Then you will apply those changes to reload the platform components.

1. Follow the steps above to log into a SSH terminal session for your Workload Optimization Manager instance and display the list of available probes.
2. In the SSH session, open the cr.yaml file for editing. For example:

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

3. Edit the probe entries.

   To enable or disable currently configured probes, find the probes you want to edit and change the settings to enable or disable them.

   To add new probes to the list, copy the probe entry you want from the output when you used `cat` to view the available probes. Then paste that entry into the cr.yaml file and set `enabled: true`.
4. Save and apply your changes to the platform.

   NOTE:
   During the online or offline upgrade process, you should not use `kubectl` to apply these changes at this time.

   After you save your changes, use `kubectl` to apply the changes:

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

5. Verify that the probes have installed correctly and all the Workload Optimization Manager pods have started.

   Execute `kubectl get pods -n turbonomic` and review the list for the mediation pods that implement your probes. Note that all pods should display READY and STATUS states similar to:

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

6. View the new probe configuration in the user interface.

   Refresh your browser and navigate to the Target Management page. You should now see the target categories and types to match your configuration changes.

License Installation and First-time Login

Before you begin, make sure you have your full or trial license key file that was sent to you in a separate email. Save the license file on your local machine so you can upload it to your Workload Optimization Manager installation.

To use Workload Optimization Manager for the first time, perform the following steps:

1. Type the IP address of your installed Workload Optimization Manager instance in a Web browser to connect to it.
2. Log in to Workload Optimization Manager.
   - Use the default credential for **USERNAME**: `administrator`.
   - Type a password for **PASSWORD**.
   - Type the password again to verify it for **REPEAT PASSWORD**.
   - Click **CONFIGURE**.
3. Continue setting up your Workload Optimization Manager installation.

   Click **LET'S GO**.
4. Open the **Enter License** fly-out.

   Click **IMPORT LICENSE**.
5. Upload your license key file.
   a. In the Enter License fly-out, you can upload the license in one of the following ways:
      - Drag and drop the license key file into the Enter License fly-out.
      - Browse to the license key file.

      Be sure to upload only .xml, .lic, or .jwt files.
   b. Click **SAVE**.

Single Sign-On Authentication

If your company policy supports Single Sign-On (SSO) authentication, you can configure Workload Optimization Manager to support SSO authentication via either Security Assertion Markup Language (SAML) 2.0 or OpenID Connect 1.0.

At a high-level, to do this you will:

- Create external groups or at least one external user for SSO. See "Managing User Accounts" in the *Workload Optimization Manager User Guide*.
- Configure Workload Optimization Manager to use SSO authentication.

  You will configure one of:

  - SSO via a SAML Identity Provider (IdP). See Setting Up SAML Authentication *(on page 52)*.
  - SSO via an OpenID Identity Provider. See Setting Up OpenID Authentication *(on page 55)*.

This section describes how to configure Workload Optimization Manager to use either SAML or OpenID to support SSO.

When SSO is enabled, users will provide their SSO credentials to log in to the Workload Optimization Manager instance. Once SSO is enabled, users cannot give local or Active Directory (AD) credentials for to login. The Identity Provider (IdP) will perform the authentication.

Prerequisites

Before you begin, make sure the IdP is set up for SSO. You can use a proprietary or public IdP. For examples of settings for a public Okta IdP, see What Are the Typical Settings for an IdP? *(on page 69)*.

Setting Up SAML Authentication

Security Assertion Markup Language (SAML) is an XML-based open standard for exchanging authentication and authorization data between parties. To configure Workload Optimization Manager to authenticate via SAML:

1. (Required) Create external groups or at least one external user for SSO.

   IMPORTANT:
   When SSO is enabled, Workload Optimization Manager only permits logins via the SSO IdP. Whenever you navigate to your Workload Optimization Manager installation, it redirects you to the SSO Identity Provider (IdP) for authentication before displaying the Workload Optimization Manager user interface.

   Before you enable SSO for your Workload Optimization Manager installation, *you must configure at least one SSO user with Workload Optimization Manager administrator privileges*. If you do not, then once you enable SSO you will not be able to configure any SSO users in Workload Optimization Manager. To authorize an SSO user as an administrator, use **EXTERNAL AUTHENTICATION** to do one of the following:

   - Configure a single SSO user with administrator authorization.

     Add an external user. The username must match an account that is managed by the IdP.
   - Configure an SSO user group with administrator authorization.

     Add an external group. The group name must match a user group on the IdP, and that group must have at least one member.

   For information about creating external groups or external users for SSO, see "Managing User Accounts" in the *Workload Optimization Manager User Guide*.
2. (Required) Ensure that chrony is configured and the system time on your Workload Optimization Manager instance is correct.

   For instructions, see Synchronizing Time *(on page 16)*.
3. Obtain the metadata from your IdP.

   You will use this metadata to configure SSO in the Workload Optimization Manager CR file located at:

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

   To get the metadata:

   a. Contact your security administrator to obtain the metadata from IdP.
   b. Save the metadata file in a directory on your local machine. For example, save the file to:

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

   c. Compare your metadata to the sample provided in Example of IdP Metadata *(on page 54)*.

      Cat out the file you just saved. It should be similar to the provided sample.
4. Obtain a certificate from IdP.

   Contact your security administrator to obtain a certificate from IdP.
5. Update the CR file with your SAML configuration.

   You now have the data that you need to configure SSO via SAML. You will edit the `cr.yaml` file that configures your Workload Optimization Manager node, and then deploy or restart the node.

   - Display the contents of your downloaded SAML metadata.

     For example, assuming you saved the file to this location on your local machine, execute the command:

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

   - Open the CR file for editing.

     In a shell, cd to the deploy/crds directory in the Workload Optimization Manager VM:

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

     Then open the CR file for editing. For example, to open the file in VI:

```
vi charts_v1alpha1_xl_cr.yaml
```

As you edit this file, you will refer to the metadata that you obtained from your IdP.

- In the CR file, navigate to the entry for the API component.

  In the CR file search for or scroll to the entry:

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

  You will make changes to this component spec, under `spec:properties:api:`
- Turn on the SAML feature.

  For the first API property, set the following:

  ```
  samlEnabled: true
  ```

- Set the SSO endpoint

  In the SAML metadata, find the entry for `md:SingleSignOnService`. Within that element, find the `Location` attribute. The value of `Location` is the SSO endpoint. Using the sample metadata we have provided, you would make the following setting in your CR file:

  ```
  samlWebSsoEndpoint: https://dev-771202.oktapreview.com/app/ibmdev771202_turbo2_1/exkexl6xc9MhzqiC30h7/sso/saml
  ```

- Set the SAML entity ID

  In the SAML metadata, find the entry for `md:EntityDescriptor`. Within that element, find the `entityID` attribute. Using the sample metadata we have provided, you would make the following setting in your CR file:

  ```
  samlEntityId: http://www.okta.com/exkexl6xc9MhzqiC30h7
  ```

- Set the SAML registration

  Set the following property:

  `samlRegistrationId: simplesamlphp`
- Set the SAML SP entity ID

  Set the following property:

  `samlSpEntityId: turbo`
- Enter the SAML certificate

  In the metadata that you got from your IdP, find the entry for `<ds:X509Certificate>`. Copy the contents of this tag – copy the characters that are between `<ds:X509Certificate>` and `</ds:X509Certificate>`.

  Create an entry for the certificate in the API properties section of the CR file. On a new line, enter:

  ```
  samlIdpCertificate: |
  ```

  Then open a new line after the entry you just created, and paste the certificate content that you copied from your metadata file.

The finished API section of the CR file should be similar to the following:

```
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. Save your changes to the CR file.
7. Apply the modified cr.yaml file.

   Execute the command:

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

8. Restart the API component to load the new spec.
   a. Open an SSH terminal session to your Workload Optimization Manager instance.
   b. Restart the API component.

      `kubectl delete pod api-<API_POD_ID>` (NOTE: To auto-fill the pod ID, type `api-` and then press TAB.)
9. Verify that the configuration is successful.
   a. Navigate to the Workload Optimization Manager User Interface.

      You will be automatically redirected to your IdP for authentication.
   b. Log in with the username that is a member of the external group or external user that you previously configured.
   c. Verify that the system time on your Workload Optimization Manager instance is correct.

      If the time is not synchronized, this might cause an HTTP `Status 401 -authentication failed` exception in the browser.
   d. If the configuration is not successful, look for an HTTP `Status 500` exception in the product log. If this exception exists, review your CR file for invalid entries.

Example of IdP Metadata

This section provides an example of IdP metadata which may be useful when you are examining the optional attributes in your metadata.

If your metadata includes optional attribute tags that are not listed in the example, remove those optional attribute tags since they are not supported.

```
<?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>
```

Setting Up OpenID Authentication

According to the OpenID Foundation, "OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol". OpenID Connect enables clients to verify user identity via a given authentication server. Workload Optimization Manager supports OpenID authentication through the following providers:

- Google
- IBM-MCM
- Okta

Logging in to Workload Optimization Manager with OpenID

When you configure OpenID in Workload Optimization Manager, the platform registers the OpenID clients that you specify. To log in through of these OpenID clients, you manually navigate to a URL that tells Workload Optimization Manager which client to use. It then redirects to the OpenID login screen for that given client.

The URL that you provide is in the form:

```
https://${hostname}/vmturbo/oauth2/login/code/${openIdClients}
```

Where:

- `${hostname}` is the host address for your installation of Workload Optimization Manager
- `${openIdClients}` is the is the client name of the OpenID provider you want to use

  You specify this as the `openIdClients` property when you configure OpenID in Turbo.

NOTE:
This URL must also be set in the "Authorized direct URIs" section of your provider's OpenID configuration.

For example, assume your Workload Optimization Manager host address is 10.10.12.34, and you configured an Okta OpenID client. In that case, when the Workload Optimization Manager login screen appears, you would navigate to:

```
https://10.10.12.34/vmturbo/oauth2/login/code/okta
```

After you navigate to that URL, the browser redirects to the OpenID login screen, where you can enter credentials for a single user or a user group.

NOTE:
To authenticate a user group, the group must be configured on the OpenID provider, and also on Workload Optimization Manager. The group name must be identical in both configurations.

On the OpenID provider, the client that you are using should include groups scope values that give specific names for user groups. Contact your OpenID administrator to get the group names. Then, on Workload Optimization Manager you should create user groups that use the same names.

For example, assume the OpenID ID token includes the following groups claim:

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

To use the group `My_Special_User_Group` for authentication, you must create a user group in Workload Optimization Manager with the name `My_Special_User_Group`. Any members of that group will then get the role you have assigned to that user group.

Configuring OpenID on Workload Optimization Manager

To configure Workload Optimization Manager to authenticate via OpenID:

1. (Required) Ensure that chrony is configured and the system time on your Workload Optimization Manager instance is correct.

   For instructions, see Synchronizing Time *(on page 16)*.
2. Obtain the necessary data from your OpenID provider.

   Contact your security administrator to obtain the data from the provider. You will use this data to configure SSO in the Workload Optimization Manager CR file located at:

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

The data you need and the properties you declare in the CR file will differ depending on the OpenID provider you want to use:

- Google:

CR Fields:	Description:
`openIdClients`	`google` The name of the OpenID client you are using to perform authentication.
`openIdClientId`	The OAuth2 Client Identifier for the OpenID client that you are using.
`openIdClientSecret`	The OAuth2 Client Secret for the OpenID client that you are using.

- IBM-MCM:

CR Fields:	Description:
`openIdClients`	`ibm` The name of the OpenID client you are using to perform authentication.
`openIdClientAuthentication`	`post` The client authentication method.
`openIdUserAuthentication`	`form` The user authentication method.
`openIdClientId`	The OAuth2 Client Identifier for the OpenID client that you are using.
`openIdClientSecret`	The OAuth2 Client Secret for the OpenID client that you are using.
`openIdAccessTokenUri`	The URI the login process will use to get an Access Token.
`openIdUserAuthorizationUri`	The URI to the Authorization Endpoint for OpenID Connect.
`openIdUserInfoUri`	The URI to the OpenID Connect UserInfo endpoint.
`openIdJwkSetUri`	The URI to get the JSON Web Key set that can verify the Access Token.
`openIdExternalGroupTag`	The name of a custom group to use for authentication.

- Okta:

CR Fields:	Description:
`openIdClients`	`okta` The name of the OpenID client you are using to perform authentication.
`openIdClientId`	The OAuth2 Client Identifier for the OpenID client that you are using.
`openIdClientSecret`	The OAuth2 Client Secret for the OpenID client that you are using.
`openIdAccessTokenUri`	The URI the login process will use to get an Access Token.
`openIdUserAuthorizationUri`	The URI to the Authorization Endpoint for OpenID Connect.
`openIdUserInfoUri`	The URI to the OpenID Connect UserInfo endpoint.
`openIdJwkSetUri`	The URI to get the JSON Web Key set that can verify the Access Token.

3. Update the Workload Optimization Manager CR file with your configuration data.

   You now have the data that you need to configure SSO via OpenID. You will edit the `cr.yaml` file that configures your Workload Optimization Manager node, and then deploy or restart the node.

   - Open the CR file for editing.

     In a shell, cd to the deploy/crds directory in the Workload Optimization Manager VM:

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

Then open the CR file for editing. For example, to open the file in VI:

```
vi charts_v1alpha1_xl_cr.yaml
```

As you edit this file, you will refer to the dat that you obtained from your authentication provider.

- In the CR file, navigate to the entry for the API component.

  In the CR file search for or scroll to the entry:

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

  You will make changes to this component spec, under `spec:properties:api:`
- Turn on the OpenID feature.

  For the first API property, set the following:

  ```
  openIdEnabled: true
  ```

  The file should be similar to:

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

- Enter the relevant OpenId data for your authentication provider. The CR file should be similar to these examples, depending on which provider you use:
  - Google:

    The file should be similar to:

    ```
    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:

    The file should be similar to:

    ```
    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

  The file should be similar to:

```
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. Save your changes to the CR file.
5. Apply the modified cr.yaml file.

   Execute the command:

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

6. Restart the API component to load the new spec.
   a. Open an SSH terminal session to your Workload Optimization Manager instance.
   b. Restart the API component.

      `kubectl delete pod api-`<API_POD_ID> (NOTE: To auto-fill the pod ID, type `api-` and then press TAB.)
7. Verify that the configuration is successful.
   a. Navigate to the Workload Optimization Manager User Interface.

      You will be automatically redirected to your authentication provider for authentication.
   b. Log in with the username that is a member of the external group or external user that you previously configured.
   c. Verify that the system time on your Workload Optimization Manager instance is correct.

If the time is not synchronized, this might cause an `HTTP Status 401 -authentication failed` exception in the browser.

d. If the configuration is not successful, look for an `HTTP Status 500` exception in the product log. If this exception exists, review your CR file for invalid entries.

Disabling Single Sign-On

If for some reason you no longer want to use SSO, you can disable it for your Workload Optimization Manager installation. To disable Single Sign-On, perform these steps:

1. Update the SSO configuration to disable it.
   a. Open an SSH terminal session to your Workload Optimization Manager instance.
   b. Open the CR file for editing.

      In a shell, cd to the deploy/crds directory in the Workload Optimization Manager VM:

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

      Then open the CR file for editing. For example, to open the file in VI:

      ```
      vi charts_v1alpha1_xl_cr.yaml
      ```

   c. In the CR file, navigate to the entry for the API component.

      In the CR file search for or scroll to the entry:

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

      You will make changes to this component spec, under `spec:properties:api:`

   d. Turn off the SSO feature.

      The entry to set to false is different depending on whether you use SAML or OpenID authentication:

      - SAML Authentication:

        Find the `samlEnabled:` property to `false`. It should appear as follows:

        ```
        samlEnabled: false
        ```

      - OpenID Authentication:

        Find the `openIdEnabled:` property to `false`. It should appear as follows:

        ```
        openIdEnabled: false
        ```

   e. Save your changes to the CR file.
2. Restart the API component.

   In the same SSH terminal session that you opened to edit the CR file:

   a. Use sudo as root.

      `sudo bash`

   b. Restart your API component.

      `kubectl delete pod api-<API_POD_ID>` (NOTE: To auto-fill the pod ID, type `api-` and then press TAB.)
3. Verify that the configuration is successful.
   a. Navigate to the Workload Optimization Manager User Interface.

      You will no longer be redirected to your IdP for authentication. You will be redirected to the default Workload Optimization Manager login screen.

   b. Log in with a local account or an Active Directory (AD) account.

Updating Workload Optimization Manager to a New Version

Cisco continually and rapidly innovates and improves all aspects of this product. Cisco releases newer versions of this product every two weeks. You should check regularly to see if a new version is available by visiting the IBM Workload Optimization Manager documentation site.

When a new version is available, it is important to properly update your existing installed instance. When you first installed Workload Optimization Manager, you put into place sophisticated data collection and analysis processes, and your database retains performance data from across your virtual environment. Workload Optimization Manager uses this historical data for right-sizing, projecting trends, and other analysis. This means that the database is important to Workload Optimization Manager and becomes more so over time. Properly updating your installation of Workload Optimization Manager preserves the database for continued use.

OVA updates

Before you begin the update procedure:

- Review the What's New and Fixed Issues sections of the documentation to see what is new for this release.

  NOTE:
  As Workload Optimization Manager evolves, the set of delivered probes change. Also, from one version to the next, the set of probes that are enabled by default can change. When you update to a new version, the update does not change your probe configuration. An update to a newer version does not automatically enable any new probes in your deployment. If you want to take advantage of new probes in an update, then you must enable them manually.
- Make sure you have the email that Cisco sent to you with links to the Workload Optimization Manager 0VA file and to the ISO image.
- For on-prem installations, make sure that the physical machine hosting the VM meets the minimum requirements (see Minimum Requirements *(on page 6)*).
- Ensure you are running the correct version of the historical database.

  For its default historical database, Workload Optimization Manager currently supports MariaDB version 10.5.16. This support includes comprehensive testing and quality control for Workload Optimization Manager usage of the historical database.

  For more information, see Verifying your MariaDB Version. *(on page 18)*
- Execute the `upgrade-precheck.sh` script.

  You can use this script to make sure that your current installation of Workload Optimization Manager is ready to update. We strongly recommend that you run this script before going on to update your installation (see Checking Before Updating *(on page 62)*).
- Execute an offline update, via a downloaded ISO image (see Offline Update *(on page 66)*).

Checking Before Updating

Before you perform an update of your Workload Optimization Manager instance, you should execute the script, `upgrade-precheck.sh`. This script inspects your installation to check for the following:

- Sufficient free disk space
- For online updates, access to required endpoints (icr.io, github.com, etc.)
- The MariaDB service is running

  Note that this check is for the default installation of the MariaDB service, only. For example, the script does not check an external installation of MySQL or MariaDB, if that is the historical database you have configured. In that case, the script will indicate that your MariaDB service is not running. For an external database deployment, this is a normal result.
- The Kubernetes service is running
- The necessary Kubernetes certificates are valid

  If the certificates are not valid, you can run the `kubeNodeCertUpdate.sh` script to correct the issue. This script should be located on your installation at `/opt/local/bin`. For more information, contact your support representative.
- Root password is not set to expire
- Time sync is enabled, and current if running
- All Workload Optimization Manager pods are running

To execute this script:

1. Download the latest version of the script.
   a. Log in to the Workload Optimization Manager VM.

      Use SSH to log in to the Workload Optimization Manager VM using the turbo account and password.
   b. Change to the scripts directory.

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

   c. Get the latest version of the script.
      i. Navigate to the following Cisco web page:

         https://software.cisco.com/download/home/286328879/type
      ii. Under **Select a Software Type**, click **Workload Optimization Manager**.
      iii. From the menu on the left, select the desired Workload Optimization Manager version.
      iv. Click the download button for the following file:

         ```
         upgrade-precheck-X.X.X.zip
         ```

      v. When prompted, login using your Cisco account.
      vi. After the download completes, unzip the downloaded file.
   d. Make the script executable.

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

2. Execute the script.

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

   As the script executes, it identifies any issues that you should address before you execute an update.

External DBs and Workload Optimization Manager Updates

If you have deployed Workload Optimization Manager with an external database server, for some updates you might need to manually create a new database and user for that deployment. This is important if your external database server is multi-tenant, or if your deployment does not grant administrative privileges to Workload Optimization Manager.

NOTE:
If your external database server is multi-tenant, or if your database server does not grant administrative privileges to Workload Optimization Manager, then you must continue with this configuration requirement.

Azure database services are multi-tenant. If you deployed an external database on Azure, this configuration requirement applies to you.

If you deployed your database server in a way that grants Workload Optimization Manager privileges to create new databases and new users, then a product update will automatically create the required database. This configuration requirement does not apply to you and you do not need to take any action.

For some Workload Optimization Manager updates, the updated version includes new databases on the historical database server. If you are updating to one of these versions, then you must *first* create the new database, and a user account with privileges to access that database.

This table lists the Workload Optimization Manager versions that required new databases. If you are updating from a version earlier than one of these, you must create the indicated new databases. For example, if you are updating from version 3.0.1 to 3.0.5, then you must create the `api` database.

Workload Optimization Manager Version:	New Databases:	Notes:
3.0.5	`api`	If you are updating from a version earlier than 3.0.5, you must create a new database named `api`, and a user account named `api`.

NOTE:
If you have already updated to one of these versions of Workload Optimization Manager, and you did not perform the steps to update your external DB, please contact your support representative.

To create the databases and users, you will:

- Manually create each required database

  This includes creating the database in your DB instance, creating a user to access the database, and granting privileges to the user.
- Manually add the each required database to your cr.yaml file

  The cr.yaml file declares entries for each component database. Each entry names the component, and gives the user and password that the component can use to access that database. You must add a new entry for each new database.

To create a new database:

1. Connect to your external DB using a global account.

   The account must have privileges to create databases and users. If you have specified `dbRootUsername` in the cr.yaml file, you can use that account.
2. Create the database, where **`<New_Database>`** matches the database name in the table above:

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

   For example, to create a new `api` database, execute:

   ```
   create database api;
   ```

3. Create the account that Workload Optimization Manager will use to access the database where **<New_Database>** matches the database name in the table above:

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

   For example, to create a user for the `api` database, execute:

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

   NOTE:
   The value **vmturbo** is the default password that Workload Optimization Manager uses for all component database accounts. If you have manually created accounts with different credentials, you can do so for this database as well.

4. Set the user account privileges for the new user account, where **<New_Database>** matches the database name in the table above:

   ```
   GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE TEMPORARY T
   ABLES, LOCK TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EVENT, TRIGGER
    ON <New_Database>.* TO '<New_Database>'@'%';
   ```

   For example, to set account privileges for the `api` user, execute:

   ```
   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. Flush privileges to make the privileges take effect:

   ```
   flush privileges;
   ```

Now that the new database is created in your external DB service, you must declare access to it the Workload Optimization Manager cr.yaml resource.

1. Open the .cr file for editing. The location of the file depends on the type of Workload Optimization Manager installation you are configuring.

VM Image installation of Workload Optimization Manager:	Workload Optimization Manager on a Kubernetes node or node cluster:
Open a SSH terminal session on your Workload Optimization Manager instance Log in with the System Administrator that you set up when you installed Workload Optimization Manager: ■ Username: `turbo` ■ Password: `[your_private_password]` Then edit the file: `/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml`	Open the following file for editing: `deploy/crds/charts_v1alpha1_xl_cr.yaml`

2. Add the credentials for the matching pod to access the new database.

   Add the entry to the `properties:` section of the cr.yaml file, where **vmturbo** is the password that you assigned to that user account, **yourDB** is the qualified name of your external DB or your multi-tenant DB partition, and **<New_Database>** is the name of your new database. Declare the following in the entry you add:

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

   For example, if you added the `api` database, the resulting cr.yaml file should be similar to:

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

After you have done this, you can update to the latest version of Workload Optimization Manager. (Note that upgrading applies changes to the version information in this file.)

Offline Update

To perform an offline update of your Workload Optimization Manager installation:

1. Save a snapshot of your current Workload Optimization Manager VM.

   Before updating, you should properly shut down (not power off) the Workload Optimization Manager VM. To do so, type:

   ```
   sudo init 0
   ```

   Then, perform a snapshot (or clone the VM). This provides a reliable restore point you can turn to in the event that trouble occurs during the update. After you have the snapshot, bring the VM back online.
2. Optionally, enable new probes in your environment.

   NOTE:
   As Workload Optimization Manager evolves, the set of delivered probes change. Also, from one version to the next, the set of probes that are enabled by default can change. When you update to a new version, the update does not change your probe configuration. An update to a newer version does not automatically enable any new probes in your deployment. If you want to take advantage of new probes in an update, then you must enable them manually.

   For steps to enable new probes in your updated version, see Enabling and Disabling Probe Components *(on page 47)*. Use these steps to edit the platform's cr.yaml file.

   NOTE:
   During the online or offline upgrade process, you should not use `kubectl` to apply these changes at this time.
3. Download the ISO image.
   a. Navigate to the following Cisco web page:

      https://software.cisco.com/download/home/286328879/type
   b. Under **Select a Software Type**, click **Workload Optimization Manager**.
   c. From the menu on the left, select the desired Workload Optimization Manager version.
   d. Click the download button for the following file:

      ```
      update64_package-X.X.X.iso
      ```
   e. When prompted, login using your Cisco account.
4. Save the ISO image to a location that is available to the VM that runs Workload Optimization Manager. Then mount the image as a CD drive.

   For example, if you run the Workload Optimization Manager VM in vCenter Server:
   a. In vCenter, navigate to the Workload Optimization Manager VM.
   b. Right-click the VM and choose **Edit Settings**.
   c. In the CD/DVD Drive drop-down menu:
      i. Choose **Datastore ISO**.
      ii. Browse to the Workload Optimization Manager update ISO image and choose it.
   d. Ensure that the **Connect at power on** checkbox is selected.
5. Open an SSH terminal session to your Workload Optimization Manager instance.

   After you have made a snapshot or clone of your current Workload Optimization Manager VM, open an SSH session. Log in with the System Administrator that you set up when you installed Workload Optimization Manager:
   - Username: `turbo`
   - Username: `[your_private_password]`

6. Get the `isoUpdate.sh` script for your update version.
   a. Navigate to the following Cisco web page:

      https://software.cisco.com/download/home/286328879/type
   b. Under **Select a Software Type**, click **Workload Optimization Manager**.
   c. From the menu on the left, select the desired Workload Optimization Manager version.
   d. Click the download button for the following file:

      ```
      isoUpdate-X.X.X.zip
      ```
   e. When prompted, login using your Cisco account.
   f. After the download completes, unzip the downloaded file.
7. Upload the script to your Workload Optimization Manager instance.

   Execute a file transfer from your local machine to the Workload Optimization Manager server. Save the script to `/opt/local/bin/` on the VM that runs Workload Optimization Manager.
8. Make the script executable.

   ```
   chmod +x /opt/local/bin/isoUpdate.sh
   ```
9. Execute the offline installation.

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

   As the script executes, it:
   - Backs up the old scripts in your installation
   - Updates the configuration and code assets in your installation
   - Updates the platform to the new version
   - Updates custom resources
   - Updates the MariaDB configuration (but this does *not* update the MariaDB version)
   - If you have enabled Embedded Reports or Data Export, installs the Embedded Reports and Data Export database (Postgres and TimescaleDB)
   - Scales down the t8c-operator and the Workload Optimization Manager components
   - Executes the final updates for this version
   - Scales up the t8c-operator, which then restarts the Workload Optimization Manager components
10. Verify that the Workload Optimization Manager application installed correctly.

    After the script is finished updating your platform, you should give it enough time for all the components to start up again.

    To verify the installation of the application, execute the command:

    `kubectl get pods -n turbonomic`

    After all of the pods start up, the READY column should read `1/1, 2/2`, and so on and the STATUS column should read `Running` for each pod.

    You should see output similar to the following:

    ```
    NAME                                    READY   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
```

11. Verify that you are running the correct version of MariaDB.

    For this version of the product, Workload Optimization Manager supports MariaDB, version 10.5.16. Even after updating to this Workload Optimization Manager version, it is possible that your installation is running an earlier version of MariaDB.

    While still in the SSH session, check the MariaDB version:

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

    The output should be similar to:

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

    If the MariaDB version is earlier than 10.5.16, you should update your MariaDB. For complete instructions and information, see Verifying your MariaDB Version *(on page 18)*.

12. Clear your browser data and refresh your browser.

    After clearing the browser data and refreshing your browser, you have full access to Workload Optimization Manager features. However, features that rely on current analysis data will not be available until after a full market cycle — usually 10 minutes. For example, the Pending Actions charts will not show any actions until after a full market cycle.

13. Notify other users to clear their browser data and refresh their Workload Optimization Manager browser sessions.

Appendix: What Are the Typical Settings for an IdP?

NOTE:
The process described here is applicable only to the OVA deployment model.

Before you begin configuring Single Sign-On (SSO), you need to make sure the IdP is set up for SSO.

Here are typical settings for a public Okta IdP which may be useful when you set up your IdP.

SAML Settings: GENERAL	
Setting	Example
Single Sign On URL (where **<hostname>** is the host that Workload Optimization Manager runs on, and **<samlRegistrationID>** is the Registration ID that you got from your SSO provider)	`https://<hostname>/vmturbo/saml2/sso/<samlRegistrationID>`
Recipient URL (where **<hostname>** is the host that Workload Optimization Manager runs on, and **<samlRegistrationID>** is the Registration ID that you got from your SSO provider)	`https://<hostname>/vmturbo/saml2/sso/<samlRegistrationID>`
Destination URL (where **<hostname>** is the host that Workload Optimization Manager runs on, and **<samlRegistrationID>** is the Registration ID that you got from your SSO provider)	`https://<hostname>/vmturbo/saml2/sso/<samlRegistrationID>`
Audience Restriction	`urn:test:turbo:markharm`
Default Relay State	
Name ID Format	`Unspecified`
Application username	The username for the account that is managed by Okta
Response	`Signed`
Assertion Signature	`Signed`

SAML Settings: GENERAL	
Setting	Example
Signature Algorithm	RSA_SHA256
Digital Algorithm	SHA256
Assertion Encryption	Unencrypted
SAML Single Logout	Enabled
Single Logout URL (where **<hostname>** is the host that Workload Optimization Manager runs on)	https://**<hostname>**/vmturbo/rest/logout
SP Issuer	turbo
Signature Certificate	Example.cer (CN=apollo)
authnContextClassRef	PasswordProtectedTransport
Honor Force Authentication	Yes
SAML Issuer ID	http://www.okta.com/$(org.externalKey)

Appendix: FIPS Cipher Suites

NOTE:
The process described here is applicable only to the OVA deployment model.

The Federal Information Processing Standard (FIPS) is in place to ensure the cryptographic strength of secure connections. By default, Workload Optimization Manager ships with a FIPS-compliant cipher suite already enabled. The suite comprises the following ciphers:

- 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

Modifying the Cipher Suite

If necessary, you can modify the cipher suite to comply with your internal policies.

1. Open an SSH terminal session on your Workload Optimization Manager instance.

   Log in with the System Administrator that you set up when you installed Workload Optimization Manager:

   - Username:

     ```
     turbo
     ```

   - Password:

     ```
     [your_private_password]
     ```

2. In the SSH session, open the cr.yaml file for editing. For example:

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

3. Edit the cipher suite.

   Search for the list of ciphers in the file. Change the list as your policies require, and then save the file.

4. Apply your changes to the platform.

   ```
   kubectl apply -f \
   ```

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

Appendix: Step-wise Platform Deployment

NOTE:
The process described here is applicable only to the OVA deployment model.

After you have installed the Workload Optimization Manager VM that will host the platform, you can install the platform components, as follows:

1. Optionally, configure Single Sign-On Authentication (SSO) for this installation.

   If you plan to use SSO to authenticate your Workload Optimization Manager users, you can configure it now. To configure SSO you will edit the `charts_v1alpha1_xl_cr.yaml` file. You can edit it now, before you complete the installation, or you can edit it later and restart the affected components. For more information, see Single Sign-On Authentication *(on page 51)*.
2. Deploy Workload Optimization Manager Kubernetes nodes.

   When you deploy Workload Optimization Manager on Kubernetes, you deploy one Kubernetes node as a VM that will host pods to run the Workload Optimization Manager components. The script to deploy and initialize the Kubernetes node also deploys the Kubernetes pods that make up the Workload Optimization Manager application.

   Start a secure session (SSH) on your Workload Optimization Manager VM as the turbo user and perform the following steps:

   a. Initialize the Kubernetes node and deploy the pods.

      Execute the script: `sudo /opt/local/bin/t8cInstall.sh`

      The script should take up to 20 minutes to complete.

   b. Verify that the deployment succeeded.

      At the end of the script output, in the summary section, verify that no errors are reported. If any errors are reported, contact Workload Optimization Manager Support.

   c. Verify that the Workload Optimization Manager application installed correctly.

      To verify the installation of the application, execute the command:

      `kubectl get pods -n turbonomic`

      After all of the pods start up, the READY column should read `1/1, 2/2`, and so on and the STATUS column should read `Running` for each pod.

      You should see output similar to the following:

```
NAME                                    READY   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. Synchronize the system clock.

   To ensure correct display of data, and to support Single Sign-On (SSO) authentication, you need to synchronize the system clock.

   For information, see Synchronizing Time *(on page 16)* and Single Sign-On Authentication *(on page 51)*.

e. Verify that the Load Balancer has installed correctly.

   To verify the presence of the Load Balancer, execute the command:

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

   You should see output similar to the following:

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

f. Configure mediation.

   The installation script automatically enables a default set of mediation probes. After installation completes, you can change the set of enabled mediation probes (see Enabling and Disabling Probe Components *(on page 47)*).

   For Workload Optimization Manager to manage your IT environment, it must attach to targets in your environment so it can perform discovery and execute actions. The combination of the processes of discovery and action execution is *mediation*. This release of Workload Optimization Manager supports mediation through the following targets. If you need to use additional targets that are not in this list, contact Workload Optimization Manager Support.

   - Applications and Databases
     - Apache Tomcat 7.x, 8.x, and 8.5.x
     - AppDynamics 4.1+
     - AppInsights
     - Dynatrace 1.1+

  - IBM WebSphere Application Server 8.5+
  - Instana, release-209 or later
  - JBoss Application Server 6.3+
  - JVM 6.0+
  - Microsoft SQL Server 2012, 2014, 2016, 2017, and 2019
  - MySQL 5.6.x and 5.7.x
  - NewRelic
  - Oracle 11g R2, 12c, 18c, and 19c
  - Oracle WebLogic 12c
- Cloud Native
  - Kubernetes, including any compliant k8s distribution (Rancher, Tanzu, open source, etc.)
  - Cloud-hosted k8s services (AKS, EKS, GKE, IBM, Cisco IKS, ROKS, ROSA, etc.)
  - OpenShift 3.11 and higher (OCP 4.x)
- Fabric and Network
  - Cisco UCS Manager 3.1+
  - HPE OneView 3.00.04
- Guest OS Processes
  - SNMP
  - WMI: Windows versions 8 / 8.1, 10, 2008 R2, 2012 / 2012 R2, 2016, 2019 and 7
- Hyperconverged
  - Cisco HyperFlex 3.5
  - Nutanix Community Edition
  - VMware vSAN
- Hypervisors
  - Microsoft Hyper-V 2008 R2, Hyper-V 2012/2012 R2, Hyper-V 2016, Hyper-V 2019
  - VMware vCenter 6.0, 6.5, 6.7, and 7.0+
- Orchestrator
  - Action Script
  - Flexera One
  - ServiceNow
- Private Cloud
  - Microsoft System Center 2012/2012 R2 Virtual Machine Manager, System Center 2016 Virtual Machine Manager, and System Center Virtual Machine Manager 2019
- Public Cloud
  - Amazon AWS
  - Amazon AWS Billing
  - Google Cloud Platform (GCP)
  - Google Cloud Platform (GCP) Billing
  - Microsoft Azure Service Principal
  - Microsoft Azure Billing
  - Microsoft Enterprise Agreement
- Storage
  - EMC ScaleIO 2.x and 3.x
  - EMC VMAX using SMI-S 8.1+
  - EMC VPLEX Local Architecture with 1:1 mapping of virtual volumes and LUNs
  - EMC XtremIO XMS 4.0+
  - HPE 3PAR InForm OS 3.2.2+, 3PAR SMI-S, 3PAR WSAPI
  - IBM FlashSystem running on Spectrum Virtualize 8.3.1.2 or later (8.4.2.0 or later recommended)

  - NetApp Cluster Mode using ONTAP 8.0+ (excluding AFF and SolidFire)
  - Pure Storage F-series and M-series arrays
- Virtual Desktop Infrastructure
  - VMware Horizon

For information about these targets, see the *Workload Optimization Manager Target Configuration Guide*.

3. Log in to the Workload Optimization Manager user interface and set the administrator user account password.

IMPORTANT:
You should wait until all the platform components have started up, are running, and are fully ready before your first login. If you try to add a license or add a target to the platform before the components are all ready, the platform can fail to initialize correctly. For more information, see Verify that the Workload Optimization Manager application installed correctly *(on page 73)*.

After the components start up, in your Web browser, type the static IP address of your Workload Optimization Manager VM. Your browser redirects the login page for Workload Optimization Manager users.

Workload Optimization Manager includes a default user account named `administrator` which has an `ADMINISTRATOR` role. As you log in for the first time, you must set your own password for that account. You can create or delete other accounts with the `ADMINISTRATOR` role, but your installation of Workload Optimization Manager must always have at least one account with that role.

In the login page, enter the information as required, and make a note of it.

- Use the default credential for **USERNAME**: `administrator`.
- Type a password for **PASSWORD**.

  The new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.
- Type the password again to verify it for **REPEAT PASSWORD**.
- Click **CONFIGURE**.

This is the account you will use to access the Workload Optimization Manager user interface with administrator permissions. *Be sure to save the user interface administrator account credentials in a safe place.*

NOTE:
The initial login is always for the `administrator` account. This is an administration *user* account. Do not confuse this with the Workload Optimization Manager System Administrator account that you previously set up to log into shell sessions on the VM itself.

4. After you have logged in as `administrator`, you can create other user accounts, and you can give them various roles. For more information about user accounts and roles, see the *Workload Optimization Manager User Guide*.

Appendix: Step-wise Offline Update

NOTE:
The process described here is applicable only to the OVA deployment model.

To perform a stepwise offline update of your Workload Optimization Manager installation:

1. Save a snapshot of your current Workload Optimization Manager VM.

   Before updating, you should properly shut down (not power off) the Workload Optimization Manager VM. To do so, type:

   ```
   sudo init 0
   ```

   Then, perform a snapshot (or clone the VM). This provides a reliable restore point you can turn to in the event that trouble occurs during the update. After you have the snapshot, bring the VM back online.
2. Download and attach the ISO image to the VM that runs Workload Optimization Manager.

   Refer to the email you received from Cisco for links to the Workload Optimization Manager 0VA file and to the ISO image.
3. Mount the ISO image by logging in to vCenter.
   a. In vCenter, navigate to the Workload Optimization Manager VM.
   b. Right-click the VM and choose **Edit Settings**.
   c. In the CD/DVD Drive drop-down menu:
      i. Choose **Datastore ISO**.
      ii. Browse to the Workload Optimization Manager update ISO image and choose it.
   d. Ensure that the **Connect at power on** checkbox is selected.
4. Log in to the Workload Optimization Manager VM.

   Use SSH to log in to the Workload Optimization Manager VM using the turbo account and password.
5. Mount the ISO image.

   Type:

   ```
   sudo mount /dev/cdrom /mnt/iso
   ```

6. Verify the correct version of the ISO image is mounted.

   Type: `ls /mnt/iso`

   Verify that the ISO image contains the correct version for your update.
7. Load the latest Docker images.

   Type: `sudo /mnt/iso/turboload.sh`

   This script loads all the images to the Workload Optimization Manager instance. If the load is successful, it displays a message similar to:

```
The t8c upgrade iso has been mounted
Image check:
============
**************************
All images have been loaded
**************************
```

If the load does not succeed, the script will list any images that did not load, along with instructions to load them manually.

8. Execute these commands to update Workload Optimization Manager.

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

   Wait until the script is finished.

9. Verify that you are running the correct version of MariaDB.

   For this version of the product, Workload Optimization Manager supports MariaDB, version 10.5.16. Even after updating to this Workload Optimization Manager version, it is possible that your installation is running an earlier version of MariaDB.

   While still in the SSH session, check the MariaDB version:

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

   The output should be similar to:

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

   If the MariaDB version is earlier than 10.5.16, you must update your MariaDB. For complete instructions and information, see Verifying your MariaDB Version *(on page 18)*.

10. Unmount the ISO image.

    Enter the command:

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

11. Clear your browser data and refresh your browser.

    After clearing the browser data and refreshing your browser, you have full access to Workload Optimization Manager features. However, features that rely on current analysis data will not be available until after a full market cycle — usually 10 minutes. For example, the Pending Actions charts will not show any actions until after a full market cycle.

12. Optionally, enable new probes in your environment.

    NOTE:
    As Workload Optimization Manager evolves, the set of delivered probes change. Also, from one version to the next, the set of probes that are enabled by default can change. When you update to a new version, the update does not change your probe configuration. An update to a newer version does not automatically enable any new probes in your deployment. If you want to take advantage of new probes in an update, then you must enable them manually.

    For steps to enable new probes in your updated version, see Enabling and Disabling Probe Components *(on page 47)*. Use these steps to edit the platform's cr.yaml file.

    NOTE:
    During the online or offline upgrade process, you should not use `kubectl` to apply these changes at this time.

13. Verify that the Workload Optimization Manager application installed correctly.

    To verify the installation of the application, execute the command:

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

After all of the pods start up, the READY column should read `1/1, 2/2`, and so on and the STATUS column should read `Running` for each pod.

14. Notify other users to clear their browser data and refresh their Workload Optimization Manager browser sessions.

Appendix: Working with YAML Files

YAML is the primary file format to create and configure resources on kubernetes, including everything to do with the Workload Optimization Manager platform. The Custom Resource YAML provides a convenient, single place to define the majority of configuration details for Workload Optimization Manager. General rules for edits include:

- Always uses spaces, not tabs, for all indentation. Since spacing and indentations matter, and can yield an invalid result or a parameter completely skipped over, you should work with YAML files with an editor that supports using vertical lines associated with indentations to visually spot a misalignment. If your editor of choice makes this difficult, you can use the linux `expand` utility when you're done, to convert tabs to equivalent spaces.
- Indention uses two spaces per level.
- Be careful to keep the same indentation for all properties in a given section.
- Never use the same property name twice in the same section. Doing this will render the YAML file invalid, though you will likely not see any notification of a problem. Rather, all but one of the property definitions will be silently ignored.

Spacing matters

For the Workload Optimization Manager Custom Resource, indentation defines where parameters are applied (globally or to specific components), so ensure you line up the text appropriately. The following example shows examples of specifications that are applied at a global level (`{“spec”:{“global”:[{“tag”:”8.6.4”}]}}`) which sets the container image tag for all instances. Then indented we see properties that are global for the remote database (`{“spec”:{“properties”:{"global":[{“dbPort”:”6033”}]}}}`) describes a property of dbPort that would be set for remote DB connections. Each line is indented two spaces from the higher level.

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

Combine properties correctly

A YAML file is read top down, and if there are different parameters that apply to the same component, they need to be combined. The following example shows a YAML where properties for the `ui` component of image tag *and* memory limit resources.

```
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
```

This YAML will not set both image tag and memory limit resources for the `ui` component because they are set in two different sections. The memory limits would be applied since that is the last section to be read, and it would overwrite the first set of properties with the image tag. The following YAML will set both image tag and memory limit resources for the `ui` component:

```
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
```