**OTM Repository Installation & System Administration Guide**

S. Livezey

Updated June 2022

Table of Contents

[1 Introduction 3](#_Toc28619719)

[2 System Requirements 3](#_Toc28619720)

[3 Architectural Overview 3](#_Toc28619721)

[4 Ansible Inventory Setup 5](#_Toc28619722)

[4.1 Inventory Directory Structure and Host Files 5](#_Toc28619723)

[4.2 OTM Repository Variables File 6](#_Toc28619724)

[4.3 Monitoring Server Variables File 8](#_Toc28619725)

[5 System Administration Tasks 9](#_Toc28619726)

[5.1 Preparing the Ansible Controller 9](#_Toc28619727)

[5.2 OTM Repository Installation 10](#_Toc28619728)

[5.3 Nagios Monitoring Installation 10](#_Toc28619729)

[5.4 Start, Shutdown & Restart the OTM Repository 11](#_Toc28619730)

[5.5 Backup the OTM Repository 12](#_Toc28619731)

[5.6 Restore an OTM Repository from Backup 12](#_Toc28619732)

[5.7 Upgrade the OTM Repository 13](#_Toc28619733)

[5.8 Update Nagios Monitoring Configuration 13](#_Toc28619734)

[6 Troubleshooting 14](#_Toc28619735)

# Introduction

The OTM Repository provides an easy-to-use method for publishing and sharing OTM models among model designers and other consumers. This guide covers the setup and maintenance of the OTM Repository itself, as well as an optional [Nagios Core](https://www.nagios.com/products/nagios-core/) installation that can remotely monitor one or more OTM Repositories for operational support. It also provides instructions for several of the most common system administration tasks.

Installation and maintenance of an OTM Repository is done using Ansible, an open source tool for IT automation that is supported for virtually every Linux platform. This guide assumes readers are familiar with Ansible and several of its key concepts such as controllers, hosts, and basic inventory setup. For more information about Ansible, see <http://www.ansible.org> and <https://docs.ansible.com/ansible/latest/index.html>.

# System Requirements

The recommended system requirements for Linux servers to host an OTM Repository and the accompanying Nagios Core monitoring platform are as follows.

**OTM Repository Server**

|  |  |
| --- | --- |
| CPU | 2 x 2+ GHz Processor (x86 64bit architecture) |
| Memory | 8GB RAM (4GB minimum) |
| Storage | 30GB HD |
| Operating System | Supported Linux OS’s include Red Hat Enterprise Linux (RHEL), Debian, and Ubuntu.[[1]](#footnote-1) |

**Nagios Core Monitoring Server**

|  |  |
| --- | --- |
| CPU | 1 x 1+ GHz Processor (x86 64bit architecture) |
| Memory | 1GB RAM |
| Storage | 8GB HD |
| Operating System | Supported Linux OS’s include Red Hat Enterprise Linux (RHEL), Debian, and Ubuntu. |

# Architectural Overview

The OTM Repository is designed to operate in a single-server configuration. Figure 1 and the table that follows provide an overview of the various system components. At this time, multi-server high availability configurations are not supported for an OTM Repository.

Figure 1: OTM Repository Architecture

|  |  |
| --- | --- |
| Component | Description |
| Tomcat Application Server | Application server used to host the OTM Repository API's and web console. |
| OTM Repository Web Service | Provides REST API’s used by the OTM-DE and build plugins, as well as a web console for administration and online browsing of the OTM models. |
| Repository Indexing Manager | Performs two primary functions for OTM repository indexing:  1) Host an ActiveMQ broker used for sending and receiving indexing messages from the OTM Repository Web Service.  2) Launch the Repository Indexing Agent process. If the agent process crashes for any reason, the Indexing Manager will relaunch it. |
| Repository Indexing Agent | Performs the actual search indexing for the OTM repository. Indexing messages are received from the Repository Web Service (via the Indexing Manager’s Active MQ broker) when OTM libraries are created, modified, or deleted. Once the indexing is complete, notifications are sent back to the Web Service via JMS messaging. |
| RabbitMQ Broker | AMQP messaging server that handles communications between the OTM Repository and the Repository Indexing Agent. |
| Subversion (SVN) Repository Server | Server process daemon (svnserve) for the Subversion repository. |

# Ansible Inventory Setup

The setup of Ansible inventory for the OTM repository requires a host file to define the key attributes used to communicate with the server and a variables file that describes the characteristics of the server setup. This file configuration applies for the setup of both the OTM Repository and Nagios Core monitoring server.

For the purposes of providing a meaningful example, this section will define an OTM Repository using the host ID ‘repo-host’ and a monitoring server with the ID ‘monitor-host’. Ansible currently supports both YAML and INI file formats for host files, while variable files may be in either YAML or JSON format. All files presented in these examples are shown in YAML format.

## Inventory Directory Structure and Host Files

By Ansible convention, the variable file is located in a /host\_vars subfolder of the directory where the Ansible host file is stored as shown in Figure 2 below. The name of the variables file for each host

|  |
| --- |
| ~/inventory  |- /host\_vars  | |- repo-host.yml  | |- monitor-host.yml  |- hosts |

Figure 2: Ansible Inventory File Structure

Host files for both the OTM Repository and the Nagios Core monitoring server follow standard Ansible conventions. The examples shown in Figure 3 assume that the servers have been configured for SSH logins via an RSA key (hence, no password is necessary) and whose users have been added to the server’s sudoers file so a password is not required to su as the root user. Readers can refer to the Ansible documentation titled ‘[How to build your inventory](https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html)’ for more information.

|  |
| --- |
| all:  hosts:  repo-host:  ansible\_host: myrepo.opentravel.org  ansible\_user: repouser  ansible\_become: True  ansible\_become\_user: root  monitor-host:  ansible\_host: repomonitor.opentravel.org  ansible\_user: monitoruser  ansible\_become: True  ansible\_become\_user: root |

Figure 3: OTM Repository & Monitoring Server Host Files (repohosts + monitorhosts)

It is also possible to specify configurations for multiple repository and monitoring servers in a single host file. This is often useful in cases where OTM repositories are needed in multiple environments such as integration, QA, training, and production.

## OTM Repository Variables File

The variables file for the OTM Repository specifies the key configuration settings for the server on which the software will be installed. In addition to the initial setup, these variables are used for each of the other system administration tasks such as backup, restoration, and upgrading of the repository software.

Figure 4 provides an example of a typical OTM Repository variables file.

|  |
| --- |
| ---  #  # Variables: localhost  #  # Commented variables provided below only need to be uncommented if you plan on changing the default value.  #  otmrepo\_id: my-repository  otmrepo\_display\_name: My OTM Repository  otmrepo\_root\_ns: http://www.myrepo.opentravel.org/models  #base\_dir: /opt/otm-repository  #tomcat\_port: "8080"  #tomcat\_dir: "{{base\_dir}}/tomcat"  # tomcat\_jmx\_port: "11097"  #manager\_jmx\_port: "11098"  #agent\_jmx\_port: "11099"  #indexing\_activemq\_port: "61616"  #indexsvc\_dir: "{{base\_dir}}/ota2-indexing-service"  #svnrepo\_dir: "{{base\_dir}}/otm/otm\_svnrepo"  #otmrepo\_dir: "{{base\_dir}}/otm/otm-db/repository"  #searchindex\_dir: "{{base\_dir}}/otm/otm-db/search-index"  #uploads\_dir: "{{base\_dir}}/uploads"  #backups\_dir: "{{base\_dir}}/otm/backups"  #controller\_backup\_dir: "~/otm-backups/{{inventory\_hostname}}"  #remote\_svn\_url: svn://localhost:5690/testrepo-1  #svn\_port: "4690"  #svn\_user: otmuser  #svn\_password: otmpA55w0rd  #rabbitmq\_port: "5672"  #otm\_user: otm  #otm\_group: otm  #nagios\_user: nagios  #nagios\_group: nagios |

Figure 4: Example OTM Repository Variables File (repo-host-vars.yml)

The following table provides more information on the optional and required variables. Default values for optional variables are provided in the above example as values for the variables that have been commented out (#).

| Variable Name | Required | Description |
| --- | --- | --- |
| otmrepo\_id | Yes | Unique identifier for the OTM Repository (no white space allowed) |
| otmrepo\_display\_name | Yes | Human-readable display name of the OTM Repository. |
| otmrepo\_root\_ns | Yes | Initial base namespace for the OTM Repository upon installation. Additional base namespaces can be created or deleted post-install. |
| base\_dir | No | The base directory location on the server under which all system components will typically be installed. |
| tomcat\_port | No | The port that will be used for HTTP communications with the Tomcat application server. |
| tomcat\_dir | No | The directory location on the server where the Tomcat application server will be installed. |
| tomcat\_jmx\_port | No | The port that will be used for JMX monitoring of the Tomcat application server. |
| manager\_jmx\_port | No | The port that will be used for JMX monitoring of the Indexing Manager process. |
| agent\_jmx\_port | No | The port that will be used for JMX monitoring of the Indexing Agent process. |
| indexing\_activemq\_port | No | The port that will provide access to the Active MQ broker maintained by the Indexing Manager process. |
| indexsvc\_dir | No | The directory where the indexing services of the OTM Repository will be installed. |
| svnrepo\_dir | No | The directory where the SVN repository storage will be created. |
| otmrepo\_dir | No | The directory where the OTM Repository data files will be stored. |
| searchindex\_dir | No | The directory where the OTM Repository search index files will be stored. |
| uploads\_dir | No | Server directory for files uploaded during the installation and upgrade processes. |
| backups\_dir | No | Server directory where data backup files will be stored (note – only the latest backup is typically stored on the repository server). |
| controller\_backup\_dir | No | Directory on the Ansible controller where repository backups are downloaded and archived. |
| remote\_svn\_url | No | URL for the SVN server if it is not maintained on the OTM repository server itself. |
| svn\_port | No | The port used to communicate with the SVN server process. |
| svn\_user | No | The user ID of the SVN repository user. |
| svn\_password | No | The password of the SVN repository user. |
| rabbitmq\_port | No | The port where the RabbitMQ broker will listen |
| otm\_user | No | The system user for OTM processes |
| otm\_group | No | The group for the OTM system user |
| nagios\_user | No | The system user for Nagios monitoring processes |
| nagios\_group | No | The group for the Nagios system user |

## Monitoring Server Variables File

The variables file for the Nagios Core server provides the information for each OTM Repository host that will be monitored. It is possible to configure monitoring for multiple OTM repositories from a single Nagios Core host, as shown in Figure 5 below.

|  |
| --- |
| otm\_repositories:  otm-production:  hostname: myrepo.opentravel.org  #tomcat\_jmx\_port: "11097"  #manager\_jmx\_port: "11098"  #agent\_jmx\_port: "11099"  otm-test:  hostname: test-repo.opentravel.org  #tomcat\_jmx\_port: "11097"  #manager\_jmx\_port: "11098"  #agent\_jmx\_port: "11099"  otm-training:  hostname: training-repo.opentravel.org  #tomcat\_jmx\_port: "11097"  #manager\_jmx\_port: "11098"  #agent\_jmx\_port: "11099"  otm\_admin\_email: admin@opentravel.org |

Figure 5: Example Monitoring Server Variables File (monitor-host-vars.yml)

The following table provides more information on the optional and required variables under each OTM Repository host ID. Default values for optional variables are provided in the above example as values for the variables that have been commented out (#).

| Variable Name | Required | Description |
| --- | --- | --- |
| <host-id> | Yes | The identifier that will be used to represent the OTM Repository server in the Nagios monitoring console (e.g. ‘otm-production’ from the example in Figure 5). |
| hostname | Yes | The DNS name or IP address of the OTM Repository server to be monitored. |
| tomcat\_jmx\_port | No | The port that will be used for JMX monitoring of the repository’s Tomcat application server. |
| manager\_jmx\_port | No | The port that will be used for JMX monitoring of the repository’s Indexing Manager process. |
| agent\_jmx\_port | No | The port that will be used for JMX monitoring of the repository’s Indexing Agent process. |

# System Administration Tasks

Once the Ansible inventory files have been properly configured, the OTM Repository and Nagios Core monitoring servers can be installed and configured using the playbooks provided in the OTM Repository Playbooks provided in the OTM-DE open source distribution. This chapter provides instructions for the most common system administration tasks used to install, operate, and maintain an OTM Repository server.

Note that all instructions provided in this chapter assume that commands are performed on a Linux command-line (terminal window) from the user’s home directory.

## Preparing the Ansible Controller

To prepare the controller (the system from which you will run Ansible), you must first download the OTM repository playbooks from the OTM-DE open source distribution on GitHub. This step requires Git to be installed on the controller (see [here](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git) for instructions if Git is not installed on the controller).

Once Git is installed, enter the following command to clone the Git repository:

|  |
| --- |
| # git clone https://github.com/OpenTravel/ota2-repository-playbooks.git |

Next, we must verify SSH connectivity between the Ansible controller and the remote host(s). This is accomplished using the Ansible ping command to both the OTM Repository and the Nagios Core servers as follows:

|  |
| --- |
| # ansible all -m ping  repo-host | SUCCESS => {  "changed": false,  "ping": "pong"  }  monitor-host | SUCCESS => {  "changed": false,  "ping": "pong"  } |

When the above steps have been accomplished successfully, the Ansible controller is ready to perform all of the system administration tasks described in this document.

For the remainder of this chapter, the ansible-playbook commands assume that inventory files are located in the default locations and that commands are executed from the directory that was created by the ‘git clone’ command above. Readers should note that all playbooks require an extra parameter named ‘target’ that will specify the host (or host group) for which the playbook will be executed.

## OTM Repository Installation

The OTM Repository server can be setup by running the following commands:

|  |
| --- |
| # ansible-playbook -e “target=repo-host” install-otm-repository.yml |

The installation process will take several minutes. After it has completed successfully, you can verify the operation of the new OTM Repository by opening a web browser and opening a page at the URL of the server’s DNS name or IP address (e.g. http://myrepo.opentravel.org).

OTM Repository web console home page.If you see the following page, the repository is up and running:

Figure 6: OTM Repository Home Page

The initial administrator’s account for the repository is ‘admin’ with a default password of ‘password’. This should be changed immediately after installation for production repositories.

## Nagios Monitoring Installation

The Nagios Core monitoring server can be setup by running the following commands:

|  |
| --- |
| # ansible-playbook -e “target=monitor-host” install-nagioscore-server.yml |

**IMPORTANT NOTE**: The installation playbook for the Nagios Core Monitoring server does not setup the server’s email configuration. It is the responsibility of the system administrator to configure the server’s mail system as either a mail relay (more common) or a local email broker (less common). This is necessary if email notifications of host and service alerts are needed.

As with the OTM Repository server, the installation process will take several minutes. After it has completed successfully, you can verify the operation of the monitoring server by opening a web browser and opening a page at the URL of the server’s DNS name or IP address followed by /nagios (e.g. http:/repomonitor.opentravel.org/nagios). When prompted for a login, enter the default Nagios administration credentials (user ID: ‘nagiosadmin’, password: ‘admin’).

Nagios Core monitoring server home page.If you see the following page, the monitoring is up and running:

Figure 7: Nagios Core Monitoring Server Home Page

Refer to the OTM Repository Operational Support Guide or more information regarding the operational support and monitoring of OTM Repositories.

## Start, Shutdown & Restart the OTM Repository

The start, stop, and restart actions for an OTM Repository are performed by the following commands:

Start the Repository services:

|  |
| --- |
| # ansible-playbook -e “target=repo-host” start-otm-repository.yml |

Shut down the Repository services:

|  |
| --- |
| # ansible-playbook -e “target=repo-host” shutdown-otm-repository.yml |

Restart the Repository services:

|  |
| --- |
| # ansible-playbook -e “target=repo-host” restart-otm-repository.yml |

It is also possible to start, stop, and restart the services from the Linux command-line while logged on to the OTM Repository server:[[2]](#footnote-2)

Start the Repository services:

|  |
| --- |
| # systemctl start ota2-indexing  # systemctl start ota2-tomcat |

Shutdown the Repository services:

|  |
| --- |
| # systemctl stop ota2-tomcat  # systemctl stop ota2-indexing |

Restart the Repository services:

|  |
| --- |
| # systemctl restart ota2-indexing  # systemctl restart ota2-tomcat |

## Backup the OTM Repository

To back up one or more OTM repositories, run the following Ansible playbook:

|  |
| --- |
| # ansible-playbook -e “target=repo-host” backup-otm-repository.yml |

The resulting backup archive will be in the form of a zip file whose name includes the date and time that the backup was created. To conserve disk space, only the latest backup archive is maintained on the OTM Repository server. On the Ansible controller (where the backup playbook is executed), all of the backups are uploaded after the archival process is complete. By default, backup archives are stored in the /otm-backups folder of the user’s home directory.

It should be noted that the OTM Repository is fully available during the backup process, and that no outages or downtime needs to be scheduled.

## Restore an OTM Repository from Backup

In the event of a server crash or some other form of data loss, it is possible to restore the data for an OTM Repository from a previously created backup archive. To restore an OTM repository from backup, run the following Ansible playbook:

|  |
| --- |
| # ansible-playbook -e "target=repo-host backup\_archive=<archive-file>" restore-otm-repository.yml |

Be advised that during the restoration process the existing SVN repository folder is renamed but not deleted. This means that the existing persistent data store for the repository is inactive, but not deleted outright. This is not typically a concern when restoring from a previous backup, but it does provide system administrators an opportunity to archive or otherwise preserve the previous content of the repository.

## Upgrade the OTM Repository

Run the following Ansible playbook to upgrade an OTM repository to the latest release from the OTM-DE open source distribution on GitHub:

|  |
| --- |
| # ansible-playbook -e “target=repo-host” upgrade-otm-repository.yml |

Since the OTM Repository does not currently support a high-availability configuration, administrators should be aware that the repository will be unavailable for a short time during the upgrade process. For this reason, it is advisable to conduct upgrades during off-hours or periods of low usage. If a Nagios Core monitoring server is in use, system administrators can schedule a downtime window for the repository by following [these instructions](https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/downtime.html).

Administrators should be aware that an upgrade only modifies the executable code of the OTM Repository Web Service and Indexing Services. Any configuration files or launch scripts remain unchanged in order to preserve any manual edits that might have been made. In some cases, this may cause functional or other operational issues, so it is strongly advised to perform any upgrades on a test server prior to modifying any production deployments. If manual edits to a repository’s configuration are required, they should be done before running the upgrade playbook to avoid any extended downtime or outage.

Finally, it is not necessary to create a backup prior to an upgrade since the creation of a backup archive is the first step of the automated upgrade process.

***IMPORTANT NOTE:*** *When upgrading the OTM repository from version 4.x to 5.x, it will be necessary to perform a fresh installation and restore the repository from a backup of the old 4.x repository. This is due to numerous system dependencies that have changed since the 4.x generation of the OTM architecture.*

## Update Nagios Monitoring Configuration

The configuration of the Nagios Core monitoring console can be updated at any time by changing the configuration of the Monitoring Server Variables File (see section 4.3). Typically, this is done to add or remove servers from operational monitoring, but it may also occur if any of the ports used to monitor the OTM Repository server have been modified.

Once the Monitoring Server Variables File has been updated, run the following Ansible playbook command to update the Nagios Core configuration:

|  |
| --- |
| # ansible-playbook -e “target=monitor-host” configure-nagios-monitoring.yml |

# Troubleshooting

This chapter is intended to provide solutions to common problems that arise during the installation process of the OTM Repository and Nagios Core monitoring server.

**PROBLEM**: Playbook execution fails due to an error related to Python

**LIKELY CAUSE**: The correct version of Python may not be installed on the remote server (Ansible requires Python v3 or higher).

**SOLUTION #1**: Using the host’s package installer (apt or yum), install the python3 package.

**SOLUTION #2**: If multiple versions of Python are installed, configure the system to use the latest version. [See here](https://linuxconfig.org/how-to-change-default-python-version-on-debian-9-stretch-linux) for an article of how to accomplish this on Debian Linux.

**PROBLEM**: Cannot login to the server via SSH after the OTM Repository and/or Nagios Core monitoring server installation completes successfully.

**LIKELY CAUSE**: Installation playbooks assume that firewalls are active and need to be configured, but firewalls were not active before the install.

**SOLUTION**: Re-disable the firewall or allow inbound access to port 22 (default for SSH).

**PROBLEM**: Package installation fails when running an installation on Debian or Ubuntu.

**LIKELY CAUSE**: The correct package distributions have not been configured on the server.

**SOLUTION**:

1. Before installation, make sure the following lines are in the /etc/apt/sources.list file:

|  |
| --- |
| deb http://deb.debian.org/debian stretch main  deb-src http://deb.debian.org/debian stretch main  deb http://deb.debian.org/debian stretch-updates main  deb-src http://deb.debian.org/debian stretch-updates main |

1. Run the following commands:

|  |
| --- |
| # apt update  # apt upgrade |

1. A good web site reference that describes this process in greater detail:

<https://linoxide.com/debian/configure-sources-list-debian-9/>

**PROBLEM**: After configuring Nagios Core monitoring, most of the services for the repository host show critical alerts. Each error message shows the same error related to JMX connections.

**LIKELY CAUSE**: Either the firewall ports are misconfigured on the OTM Repository server, or there is a problem with the server’s host name configuration.

**SOLUTION #1**: Allow inbound access for each of the required JMX monitoring ports on the OTM Repository server. The default ports are 11097, 11098, and 11099.

**SOLUTION #2:**

1. Check to see that the hostname assigned on the server matches either the server's IP address or the DNS name by running the following command:

|  |
| --- |
| # hostname -i |

1. If needed, change the hostname to match the DNS name or IP address of the server with the following command:

|  |
| --- |
| # hostnamectl set-hostname <ip/dns-name> |

**PROBLEM**: Playbook execution fails while downloading an artifact from GitHub or some other external web site.

**LIKELY CAUSE**: Either a network glitch occurred, or the Ansible controller does not have access to the public internet to download the required files.

**SOLUTION #1**: Retry the playbook execution. If a network glitch occurred, it is probably that the downloads will succeed on subsequent attempts. Otherwise, the Ansible controller needs to be configured for access to the public internet (possibly via a network proxy).

**PROBLEM**: In very rare cases, the following error may be reported on CentOS or RHEL servers: FAILED! => {"changed": false, "msg": "yum lockfile is held by another process"}

**LIKELY CAUSE**: Unknown

**SOLUTION #1**: Retry the playbook execution (things should work fine the next time). If the error continues, the server may need to be rebooted.

1. While other configurations are expected to function properly, verification testing was conducted with specific versions of each OS including RHEL v8, Debian v11, and Ubuntu v22. Supported service managers are SystemD and InetD, with firewall configuration support for FirewallD, UFW, and IPTables. [↑](#footnote-ref-1)
2. The examples provided are based upon the SystemD service manager. For InetD, the command syntax is as follows: ‘service ota2-tomcat start’. [↑](#footnote-ref-2)