[Repository Manager 2](https://help.sonatype.com/display/NXRM2/Repository+Manager+2)

[转至元数据结尾](https://help.sonatype.com/display/NXRM2#page-metadata-end)

* 由 [Jeff Wayman](https://help.sonatype.com/display/~jwayman)创建, 最终由 [Dulani Wallace](https://help.sonatype.com/display/~dwallace)修改于 [八月 24, 2017](https://help.sonatype.com/pages/diffpagesbyversion.action?pageId=329737&selectedPageVersions=16&selectedPageVersions=17)

[转至元数据起始](https://help.sonatype.com/display/NXRM2#page-metadata-start)

Welcome to the Help Portal for Nexus Repository Manager (NXRM) 2.

If you are developing software without a repository manager you are likely missing a number of opportunities to reduce some pretty obvious inefficiencies. If everyone on your team has to hit public repositories like the Central Repository to download components（下载组件）, you are missing out on some simple gains in speed and efficiency. If you don't have a local place to deploy components you are forced to share binary components using half-measures and compromises such as storing binaries in source control. Stop developing in the Dark Ages, read this guide, and start using a repository manager. Trust us, once you start using a Nexus Repository Manager, you'll wonder how you ever functioned without it.

 Getting Started with NXRM 2 (Click to expand)

 Installation

 Additional Configuration

 Security

 Synchronization

 Repository Formats

To learn more about Nexus solutions and licenses, see the knowledge base article [Sonatype Product and Solutions Overview](https://support.sonatype.com/hc/en-us/articles/215716807), or contact [support@sonatype.com](mailto:support@sonatype.com).

[Installation and Running](https://help.sonatype.com/display/NXRM2/Installation+and+Running)

[转至元数据结尾](https://help.sonatype.com/display/NXRM2/Installation+and+Running#page-metadata-end)

* 由 [Dulani Wallace](https://help.sonatype.com/display/~dwallace)创建, 最终由 [Peter Lynch](https://help.sonatype.com/display/~plynch)修改于 [八月 16, 2017](https://help.sonatype.com/pages/diffpagesbyversion.action?pageId=329742&selectedPageVersions=8&selectedPageVersions=9)

[转至元数据起始](https://help.sonatype.com/display/NXRM2/Installation+and+Running#page-metadata-start)

* [Prerequisites](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Prerequisites)
* [Downloading](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Downloading)
  + [Downloading Nexus Repository Manager OSS](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-DownloadingNexusRepositoryManagerOSS)
  + [Downloading Nexus Repository Manager Pro](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-DownloadingNexusRepositoryManagerPro)
* [Installing](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Installing)
* [Upgrading](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Upgrading)
* [Running](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Running)
* [Post-Install Checklist](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Post-InstallChecklist)
  + [Step 1: Change the Administrative Password and Email Address](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Step1:ChangetheAdministrativePasswordandEmailAddress)
  + [Step 2: Configure the SMTP Settings](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Step2:ConfiguretheSMTPSettings)
  + [Step 3: Configure Default HTTP and HTTPS Proxy Settings](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Step3:ConfigureDefaultHTTPandHTTPSProxySettings)
  + [Step 4: Enable Remote Index Downloads](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Step4:EnableRemoteIndexDownloads)
  + [Step 5: Change the Deployment Password](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Step5:ChangetheDeploymentPassword)
  + [Step 6: If Necessary, Set the LANG Environment Variable](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Step6:IfNecessary,SettheLANGEnvironmentVariable)
  + [Step 7: Configure Routes](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Step7:ConfigureRoutes)
* [Configuring Nexus Repository Manager as a Service](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-ConfiguringNexusRepositoryManagerasaService)
  + [Running as a Service on Linux](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-RunningasaServiceonLinux)
    - [Run as a Service on Red Hat, Fedora, and CentOS](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-RunasaServiceonRedHat,Fedora,andCentOS)
    - [Runs as a Service on Ubuntu and Debian](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-RunsasaServiceonUbuntuandDebian)
  + [Running as a Service on Mac OS X](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-RunningasaServiceonMacOSX)
  + [Running as a Service on Windows](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-RunningasaServiceonWindows)
* [Running Behind a Reverse Proxy](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-RunningBehindaReverseProxy)
  + [Webapp Context Path](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-WebappContextPath)
  + [Do Not Force Base URL](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-DoNotForceBaseURL)
  + [Example: Reverse Proxy On Restricted Ports](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Example:ReverseProxyOnRestrictedPorts)
  + [Example: Reverse Proxy Virtual Host at Base Path](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Example:ReverseProxyVirtualHostatBasePath)
  + [Example: Reverse Proxy SSL Termination at Base Path](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Example:ReverseProxySSLTerminationatBasePath)
* [Installing a Nexus Repository Manager Pro License](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-InstallingaNexusRepositoryManagerProLicense)
  + [License Expiration](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-LicenseExpiration)
* [Directories](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Directories)
  + [Sonatype Work Directory](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-SonatypeWorkDirectory)
  + [Configuration Directory](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-ConfigurationDirectory)
* [Monitoring](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Monitoring)
  + [General Logging](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-GeneralLogging)
  + [Request Access Logging](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-RequestAccessLogging)
  + [Using Java Management Extension JMX](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-UsingJavaManagementExtensionJMX)
  + [Analytics](https://help.sonatype.com/display/NXRM2/Installation+and+Running#InstallationandRunning-Analytics)

***Available in Nexus Repository OSS and Nexus Repository Pro***

# Prerequisites

Nexus Repository Manager only has one prerequisite, a Java Runtime Environment (JRE) compatible with Java 7 or Java 8（只需要一个先决条件，java 7或8的运行时环境）. Nexus Repository Manager Pro and Nexus Repository Manager OSS are most often run with the JRE that is bundled with a Java Development Kit (JDK) installation. The main supported Java distribution is the Oracle version.

To download the Oracle JDK, go to <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. At a minimum Java 7u2 is required（java的版本至少要是java7u2）, but we recommend to use the latest available version.

# Downloading

There are two distributions of the Nexus Repository Manager: [Nexus Repository Manager OSS](https://www.sonatype.com/nexus-repository-oss) and [Nexus Repository Manager Pro](https://www.sonatype.com/nexus-repository-sonatype). Nexus Repository Manager OSS is a fully-featured repository manager(丰富功能的仓库管理者) which can be freely used, customized, and distributed under the Eclipse Public License (EPL Version 1). Nexus Repository Manager Pro（专业版适用于大型的企业和复杂的仓库管理的公司） is a distribution with features that are relevant to large enterprises and organizations which require complex procurement and staging workflows in addition to more advanced LDAP integration, Atlassian Crowd support, and other development infrastructure.

## Downloading Nexus Repository Manager OSS

To download the latest Nexus Repository Manager OSS distribution, go to [Sonatype’s OSS download page](http://www.sonatype.org/nexus/go/) and choose the .tar.gz file or .zip file that suits your need from the Nexus Repository Manager OSS 2.x section. The downloaded file will match the exact link nexus-2.14.5-03-bundle.tar.gz or nexus-2.14.5-02-bundle.zip respectively.

Older versions can be found [here](https://help.sonatype.com/display/NXRM2/Download+Archives+-+Repository+Manager+2) if needed.

## Downloading Nexus Repository Manager Pro

Nexus Repository Manager Pro can be downloaded as zip or tar.gz archive from the [download page on the support site](https://help.sonatype.com/display/NXRM2/Download).

Use the Nexus Repository Manager Pro trial version for an evaluation.

# Installing

The following instructions are for installing Nexus Repository Manager OSS or Nexus Repository Manager Pro as a stand-alone server(单机服务器). Nexus Repository Manager Pro and Nexus Repository Manager OSS are bundled with a Jetty instance（绑定了一个jetty实例） that listens to all configured IP addresses on a host (0.0.0.0) and runs on port 8081 by default（默认的jetty监听所有绑定的ip,并监听端口8081）.

Installing the repository is straightforward. Unpack the web application archive in a directory. If you are installing the repository manager on a local workstation to give it a test run, you can install it in your home directory or wherever you like. Nexus Repository Manager Pro and Nexus Repository Manager OSS do not have any hard coded directories. It will run from any directory（没有硬编码，可以在任何的文件目录启动）. If you downloaded the ZIP

**（解压）**

|  |
| --- |
| $ unzip nexus-2.14.5-02-bundle.zip |

And, if you download the GZip’d TAR archive, run:

|  |
| --- |
| $ tar xvzf nexus-2.14.5-02-bundle.tar.gz |

 For Nexus Repository Manager Pro the equivalent commands would be

|  |
| --- |
| $ unzip nexus-professional-2.14.5-02-bundle.zip  $ tar xvzf nexus-professional-2.14.5-02-bundle.tar.gz |

There are some known incompatibilities with the version of the tar command provided by Solaris and the GZip TAR format. If you are installing Nexus Repository Manager on Solaris, you must use the GNU tar application, or you will end up with corrupted files.

If you are installing the repository manager on a server, you might want to use a directory other than your home directory. On a Unix machine, this book assumes that it is installed in /usr/local/nexus-2.14.4-03 with a symbolic link（软连接） /usr/local/nexus to the nexus directory. Using a generic symbolic link nexus to a specific version is a common practice which makes it easier to upgrade when a newer version is made available.

|  |
| --- |
| $ sudo cp nexus-2.14.4-03-bundle.tar.gz /usr/local  $ cd /usr/local  $ sudo tar xvzf nexus-2.14.4-03-bundle.tar.gz  $ sudo ln -s nexus-2.14.4-03 nexus |

Although it isn’t required to run, you may want to set an environment variable NEXUS\_HOME in your environment that points to the installation directory. This chapter will refer to this location as $NEXUS\_HOME.

On Windows you should install the repository manage r outside Program Files to avoid problems with Windows file registry virtualization. If you plan to run the repository manager as a specific user you could install into the AppData\Local directory of that users home directory. Otherwise simply go with e.g., C:\nexus or something similar.

The installation directory nexus-2.14.4-03 or nexus-professional-2.14.4-03 has a sibling directory named sonatype-work（在同级目录下还有一个名为sonatype-work的目录）. This directory contains all of the repository and configuration data and is stored outside of the installation directory to make it easier to upgrade to a newer version（这个目录包含所以后的仓库局和数据配置，并且被保存在了安装目录的外边以便于更容易的更新到新的版本）.

By default, this directory is always a sibling to the installation directory（默认的，这个目录总是处在安装目录的同级目录）. If you installed the repository manager in the /usr/local directory it would also contain a sonatype-work subdirectory with a nested nexus directory containing all of the content and configuration. The location of the sonatype-work directory can be customized by altering the nexus-work property in   $NEXUS\_HOME/conf/nexus.properties.

# Upgrading

Since the repository manager separates its configuration and data storage from the application, it is easy to upgrade an existing installation（因为仓库将他的配置和数据存储分别存放，这样使得它容易更新一个存在到的安装）.

To upgrade the repository manager, unpack the archive in the directory that contains the existing installation. Once the archive is unpacked, the new application directory should be a sibling to your existing sonatype-work/ directory（要想更新仓库管理，只需要将文件解压到已安装目录即可）.

If you have defined a symbolic link for the version of the repository manager to use, stop the server and change that to point at the new application directory. When you start the new instance it will read the existing repository configuration from the sonatype-work directory（当启动了新的实例后，它会从sonatype-work目录读取存在的仓库配置和数据）. Depending on the version you upgrade from and to, some maintenance tasks like rebuilding the internal indices can be necessary（一些保留的任务项重构内部索引是必须的，所以在更新版本的时候需要参考好版本和以便仍然能够使用这些功能）. Please refer to the upgrade notes of the new release for more information on this. In addition, a review of the release notes can be very useful to get a better understanding of potential, additional steps required（可以参考一下每个版本的release notes）.

更新的时候要考虑到关键的功能仍然存在，因为某些功能跟版本有关。

If you are using any additional plugins supplied by Sonatype, the new version you downloaded will contain a newer version of the plugin. Be sure to copy the new version from the optional-plugins folder to the plugin-repository folder, as documented in “Managing Plugins”, and restart the repository manager.

更新版本的时候还要考虑到插件的更新。

Externally supplied plugins are updated by simply replacing the folder with the plugin with the new version（外部提供的插件只需要旧的插件目录覆盖即可）.

This automatic upgrade of the repository manager works for nearly all update ranges. All 2.x versions can directly upgrade to the latest version. All 1.x version can upgrade to 2.7.x maximum. If you need to upgrade from 1.x to a newer version, you need to perform an intermediate upgrade step to a 2.x version.

The same upgrade process can be used to change from Nexus Repository Manager OSS to Nexus Repository Manager Pro.

# Running

When you start the repository manager, you are starting a web server on the default port 0.0.0.0:8081（当启动了一个仓库管理的时候，实际上是启动了一个web服务器，绑定ip为0.0.0.0，端口为8081）. It runs within a servlet container called Eclipse Jetty（这个web服务运行在jetty中）, and it is started with a native service wrapper called the Tanuki Java Service Wrapper . This service wrapper can be configured to run the repository manager as a Windows service or a Unix daemon. Nexus Repository Manager Pro and Nexus Repository Manager OSS ship with generic startup scripts for Unix-like platforms called nexus and for Windows platforms called nexus.bat in the $NEXUS\_HOME/bin folder. To start the repository manager on a Unix-like platform like Linux, MacOSX or Solaris use:

|  |
| --- |
| cd /usr/local/nexus  ./bin/nexus console #运行repository manager |

Similarly, starting on Windows can be done with the nexus.bat file. Starting the repository manager with the console command will leave it running in the current shell and display the log output.

On Unix systems, you can start the repository manager detached from the starting shell with the start command even when not yet installed as a service.

|  |
| --- |
| ./bin/nexus start |

When executed you should see a feedback message and then you can follow the startup process viewing the log file logs/wrapper.log   
changes.

|  |
| --- |
| Starting Nexus Repository Manager...  Started Nexus Repository Manager.  $ tail -f logs/wrapper.log |

At this point, the repository manager will be running and listening on all IP addresses (0.0.0.0) that are configured for the current host on port 8081(此时nexus将会监听在端口8081上监听所有的ip访问). To use the user interface, fire up a web browser and type in the URL <http://localhost:8081/nexus> . You should see the user interface as displayed in Figure 3.5, “Application Window” .

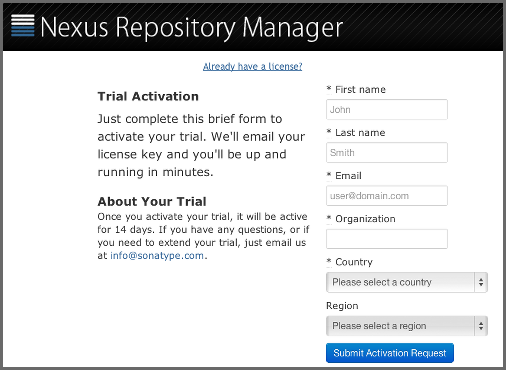
While we use localhost throughout this book, you may need to use the IP Loopback Address of 127.0.0.1, the IP address or the DNS hostname assigned to the machine running the repository manager.

Need more from Nexus?

Download a Pro trial and see what you're missing.

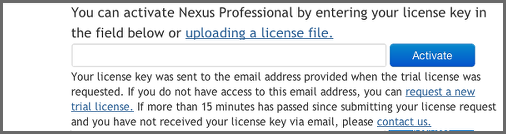
[https://help.sonatype.com/download/attachments/329742/Button-Try-Nexus-Pro-For-Free.png?version=3&modificationDate=1502740700411&api=v2](http://links.sonatype.com/products/nexus/pro/trial)

When first starting Nexus Repository Manager Pro you are presented with a form that allows you to request a trial activation. This page displayed in Figure 3.1, “Trial Activation Form” contains a link to the license activation screen in Figure 3.2, “License Activation”.



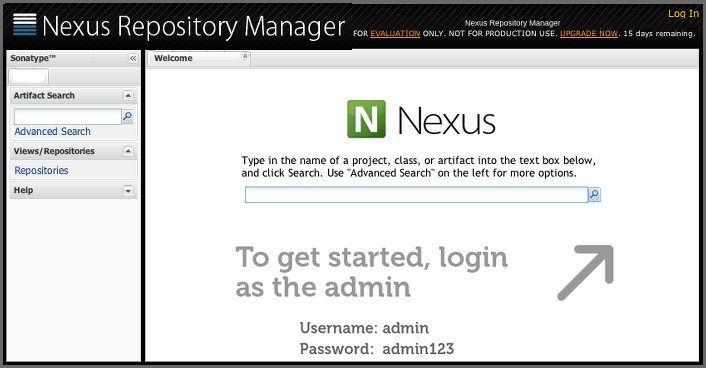
**Figure 3.1. Trial Activation Form**

After submitting the form for your trial activation, you will receive a license key via email that you can use in the license activation screen to activate Nexus Repository Manager Pro. If you already have a license key or license file, you can use the same screen to upload the file and register your license.



**Figure 3.2. License Activation**

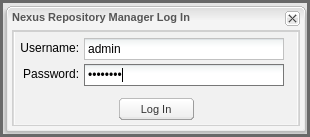
Once you have agreed to the End User License Agreement you will be directed to the Nexus Repository Manager Pro Welcome screen displayed in Figure 3.3, “Nexus Repository Manager Pro Welcome Screen”.



**Figure 3.3. Nexus Repository Manager Pro Welcome Screen**

Click on the Log In link in the upper right-hand corner of the web page, and you should see the login dialog displayed in Figure 3.4, “Log In Dialog (default login/password is admin/admin123)” .

The default administrator username and password combination is admin and admin123 .

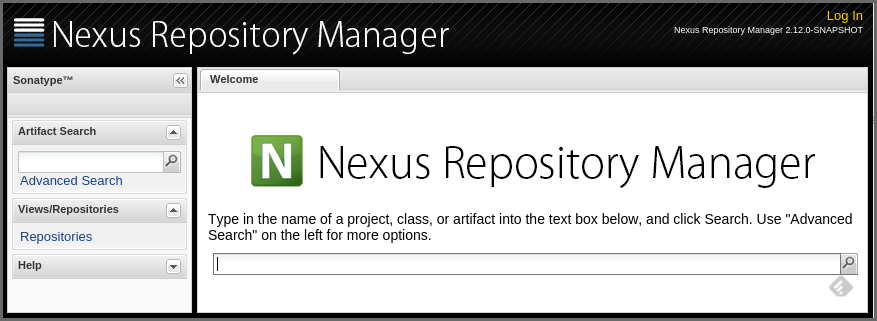


**Figure 3.4. Log In Dialog (default login/password is admin/admin123)**

When you are logged into your evaluation version of Nexus Repository Manager Pro, you will see some helpful links to the Nexus Repository Manager Pro Evaluation Guide, Sample Projects and the Knowledge base below the search input on the Welcome screen.

With a full license for Nexus Repository Manager Pro these links will be removed and you will get the application window displayed in Figure 3.5, “Application Window”.

Nexus Repository Manager OSS will not need to be activated with a license key and will display a number of links to resources and support on the Welcome screen to logged in users.



**Figure 3.5. Application Window**

The files from Java Service Wrapper used for the start up process can be found in $NEXUS\_HOME/bin/jsw and are separated into generic files like the wrapper.conf configuration file in conf and a number of libraries in lib. An optional wrapper.conf include allows you to place further configuration optionally in $NEXUS\_HOME/conf/wrapper-override.conf .

The platform-specific directories are available for backwards compatibility with older versions only and should not be used. A full list of directories follows:

|  |
| --- |
| $ cd /usr/local/nexus/bin/jsw  $ ls -1  conf  lib  license  linux-ppc-64  linux-x86-32  linux-x86-64  macosx-universal-32  macosx-universal-64  solaris-sparc-32  solaris-sparc-64  solaris-x86-32  windows-x86-32  windows-x86-64 |

The wrapper.conf file is the central configuration file for the startup of the Jetty servlet container running the repository manager on a Java virtual machine and therefore includes configuration for things such as the java command to use, Java memory configuration, logging configuration and other settings documented in the configuration file.

Typical modifications include adapting the maximum memory size to your server hardware and usage requirements e.g. 2000 MB up from the default 768 and other JVM related configurations.

|  |
| --- |
| wrapper.java.maxmemory=2000 |

You can configure JSW to use a specific Java installation and not just the Java command found on the PATH by setting JAVA\_HOME in the wrapper.conf file and using it for the startup command.

|  |
| --- |
| set.JAVA\_HOME=/opt/jdk1.8.0\_40/  wrapper.java.command=%JAVA\_HOME%/bin/java |

A typical use case is using a custom installation of the Oracle JDK instead of OpenJDK that is preinstalled as part of the Linux distribution.

Additional configuration in the wrapper.conf file includes activation of further Jetty configuration file for monitoring the repository manager via JMX or using HTTPS .

The startup script nexus supports the common service commands start, stop, restart , status, console, and dump.

# Post-Install Checklist

Nexus Repository Manager Pro and Nexus Repository Manager OSS ship with some default passwords and settings for repository indexing that need to be changed for your installation to be useful (and secure). After installing and running the repository manager, you need to make sure that you complete the following tasks:

## Step 1: Change the Administrative Password and Email Address

The administrative password defaults to admin123. The first thing you should do to your new installation is change this password. To change the administrative password, login as admin with the password admin123, and click on Change Password under the Security menu in the left-hand side of the browser window. For more detailed instructions, see “Working with Your User Profile” .

## Step 2: Configure the SMTP Settings

The repository manager can send username and password recovery emails. To enable this feature, you will need to configure a SMTP Host and Port as well as any necessary authentication parameters that the repository manager needs to connect to the mail server. To configure the SMTP settings, follow the instructions in “SMTP Settings” .

## Step 3: Configure Default HTTP and HTTPS Proxy Settings

In many deployments the internet, and therefore any remote repositories that the repository manager needs to proxy, can only be reached via a HTTP and HTTPS proxy server internal to the deployment company. In these cases the connection details to that proxy server need to be configured, as documented in “Default HTTP and HTTPS Proxy Settings”  in order for the repository manager to be able to proxy remote repositories at all.

## Step 4: Enable Remote Index Downloads

Nexus Repository Manager Pro and Nexus Repository Manager OSS ship with three important proxy repositories for the Maven Central repository, Apache Snapshot repository, and the Codehaus Snapshot repository. Each of these repositories contains thousands (or tens of  
thousands) of components and it would be impractical to download the entire contents of each. To that end, most repositories maintain an index which catalogues the entire contents and provides for fast and efficient searching. The repository manager uses these remote indexes to search for components, but we’ve disabled the index download as a default setting. To download remote indexes:

1. Click on Repositories under the Views/Repositories menu in the left-hand side of the browser window.
2. Select each of the three proxy repositories and change Download Remote Indexes to true in the Configuration tab. You’ll need to load the dialog shown in Figure 6.9, “Repository Configuration Screen for a Proxy Repository” for each of the three repositories.

This will trigger the repository manager to re-index these repositories, during which the remote index files will be downloaded. It might take a few minutes to download the entire index, but once you have it, you’ll be able to search the entire contents of the Maven repository.

Once you’ve enabled remote index downloads, you still will not be able to browse the complete contents of a remote repository. Downloading the remote index allows you to search for components in a repository, but until you download those components from the remote repository they will not show in the repository tree when you are browsing a repository. When browsing a repository, you will only be shown components which have been downloaded from the remote repository.

## Step 5: Change the Deployment Password

The deployment user’s password defaults to deployment123 . Change this password to make sure that only authorized developers can deploy components to your installation. To change the deployment password, log in as an administrator. Click on Security to expand the security menu. When the menu appears, click on Users. A list of users will appear. At that point, right-click on the user named Deployment and select Set Password.

## Step 6: If Necessary, Set the LANG Environment Variable

If your repository manager needs to store configuration and data using an international character set, you should set the LANG environment variable. The Java Runtime will adapt to the value of the LANG environment variable and ensure that configuration data is saved using the appropriate character type. If you are starting the repository manager as a service, place this environment variable in the startup script found in /etc/init.d/nexus.

## Step 7: Configure Routes

A route defines patterns used to define and identify the repositories in which the components are searched for. Typically, internal components are not available in the Central Repository or any other external, public repository. A route, as documented in “Managing Routing” , should be configured so that any requests for internal components do not leak to external repositories.

# Configuring Nexus Repository Manager as a Service

When installing Nexus Repository Manager Pro or Nexus Repository Manager OSS for production usage you should configure it to run as a service, so it starts back up after server reboots. It is good practice to run that service or daemon as a specific user that has only the required access rights. The following sections provide instructions for configuring the repository manager as a service or daemon on various operating systems.

## Running as a Service on Linux

You can configure the repository manager to start automatically by copying the nexus script to the /etc/init.d directory. On a Linux system perform the following operations as the root user:

1. Create a nexus user with sufficient access rights to run the service
2. Copy $NEXUS\_HOME/bin/nexus to /etc/init.d/nexus
3. Make the /etc/init.d/nexus script executable and owned by the root user -

|  |
| --- |
| chmod 755 /etc/init.d/nexus  chown root /etc/init.d/nexus |

1. Edit this script changing the following variables:
   1. Change NEXUS\_HOME to the absolute folder location (e.g., NEXUS\_HOME="/usr/local/nexus")
   2. Set the RUN\_AS\_USER to nexus or any other user with restricted rights that you want to use to run the service. You should not be running the repository manager as root.
   3. Change PIDDIR to a directory where this user has read/write permissions. In most Linux distributions, /var/run is only writable by root. The property you need to add to customize the PID file location is wrapper.pidfile. For more information about this property and how it would be configured in wrapper.conf, see: <http://wrapper.tanukisoftware.com/doc/english/properties.html>.
2. Change the owner and group of the directories used by the repository manager, including nexus-work configured in nexus.properties defaulting to sonatype-work/nexus, to the nexus user that will run the application.
3. If Java is not on the default path for the user running the repository manager, add a JAVA\_HOME variable which points to your local Java installation and add a $JAVA\_HOME/bin to the PATH.

We recommend to avoid running the repository manager as the root user or a similar privileged user, as this practice poses serious security risks to the host operating system unnecessarily. Instead we suggest to follow system administration best practice and use a service specific user with the minimum required access rights only.

### Run as a Service on Red Hat, Fedora, and CentOS

This script has the appropriate chkconfig directives, so all you need to do is to add the repository manager as a service is run the following commands:

|  |
| --- |
| $ cd /etc/init.d  $ chkconfig --add nexus  $ chkconfig --levels 345 nexus on  $ service nexus start  Starting Nexus Repository Manager Pro...  $ tail -f /usr/local/nexus/logs/wrapper.log |

The second command adds nexus as a service to be started and stopped with the service command.

chkconfig manages the symbolic links in /etc/rc[0-6].d which control the services to be started and stopped when the operating system restarts or transitions between run-levels. The third command adds nexus to run-levels 3, 4, and 5. The service command starts the repository manager, and the last command tails the wrapper.log to verify that it has been started successfully. If the repository manager has started successfully, you should see a message notifying you that it is listening for HTTP.

### Runs as a Service on Ubuntu and Debian

The process for setting up the repository manager as a service on Ubuntu differs slightly from the process used on a Red Hat variant. Instead of running chkconfig, you should run the following sequence of commands once you’ve configured the startup script in /etc/init.d.

|  |
| --- |
| $ cd /etc/init.d  $ update-rc.d nexus defaults  $ service nexus start  Starting Nexus Repository Manager Pro...  $ tail -f /usr/local/nexus/logs/wrapper.log |

## Running as a Service on Mac OS X

The standard way to run a service on Mac OS X is by using launchd, which uses plist files for configuration. An example plist file for the repository manager installed in /opt is shown A sample com.sonatype.nexus.plist file.

A sample com.sonatype.nexus.plist file.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"      "[http://www.apple.com/DTDs/PropertyList-1.0.dtd"](http://www.apple.com/DTDs/PropertyList-1.0.dtd)>  <plist version="1.0">  <dict>      <key>Label</key>      <string>com.sonatype.nexus</string>      <key>ProgramArguments</key>      <array>          <string>/opt/nexus/bin/nexus</string>          <string>start</string>      </array>      <key>RunAtLoad</key>      <true/>  </dict>  </plist> |

After saving the file as com.sonatype.nexus.plist in /Library/LaunchDaemons/ you have to change the ownership and access rights.

|  |
| --- |
| sudo chown root:wheel /Library/LaunchDaemons/com.sonatype.nexus.plist  sudo chmod 644 /Library/LaunchDaemons/com.sonatype.nexus.plist |

Consider setting up a different user to run the repository manager and adapt permissions and the RUN\_AS\_USER setting in the nexus startup script.

With this setup the repository managers, starts as a service at boot time. To manually start it after the configuration you can use

|  |
| --- |
| sudo launchctl load /Library/LaunchDaemons/com.sonatype.nexus.plist |

## Running as a Service on Windows

The startup script for the repository manager on Windows platforms is bin/nexus.bat. Besides the standard commands for starting and stopping the service, it has the additional commands install and uninstall. Running these commands with elevated privileges will set up the service for you or remove it as desired. Once installed as a service with the install command, the batch file can be used to start and stop the service. In addition, the service will be available in the usual Windows service management console as a service named nexus.

# Running Behind a Reverse Proxy

Nexus Repository Manager is a sophisticated server application with a web-application user interface, answering HTTP requests using  
the high-performance servlet container [Eclipse Jetty](http://www.eclipse.org/jetty/). Organizations are sometimes required to run applications like Nexus Repository Manager Pro or Nexus Repository Manager OSS behind a [reverse proxy](https://en.wikipedia.org/wiki/Reverse_proxy). Reasoning can include:

* security and auditing concerns
* network administrator familiarity
* organizational policy
* disparate application consolidation
* virtual hosting
* exposing applications on restricted ports
* SSL termination

We provide some general guidance on how to configure common reverse proxy servers to work with Nexus Repository Manager Pro and Nexus Repository Manager OSS. Always consult your reverse proxy administrator to ensure you configuration is secure.

There are two main settings of the repository manager, which can affect how reverse proxies interact.

## Webapp Context Path

The repository manager webapp context path is /nexus by default. This means every URL path used to access the repository manager must begin with /nexus.

In cases where the repository manager needs to be accessed at a different base path, through your reverse proxy or directly, you must change the default path by editing a property value.

For example, to expose the repository manager in the root context ( / ) instead of /nexus/:

1. Edit $NEXUS\_HOME/conf/nexus.properties. Change nexus-webapp-context-path=/nexus to nexus-webapp-context-path=/
2. Restart the repository manager and verify that it is available on http://localhost:8081/ and no longer available at http://localhost:8081/nexus/.
3. Emails triggered by your repository manager may include absolute links back to the originating server. As a matter of courtesy, set the Base URL as shown in Figure 6.4, “Administration Application Server Settings” under Application Server Settings to the URL that will be externally available to your users e.g. http://repo.example.com/ .

## Do Not Force Base URL

The Administration → Server → Application Server Settings configuration to Force Base URL feature. The original use case for forcing base URL is no longer valid.

When enabled, the incoming request host and base path is ignored and the repository manager acts like it is being accessed at the value of base URL.

Do not enable the Figure 6.4, “Administration Application Server Settings” Force Base URL unless explicitly advised by Sonatype - enabling this will most likely cause your repository manager to not work properly through a reverse proxy.

## Example: Reverse Proxy On Restricted Ports

**Scenario:** You need to expose the repository manager on restricted port 80. The repository manager should not be run with the root user . Instead run your reverse proxy on the restricted port 80 and the repository manager on the default port 8081. End users will access the repository manager using the virtual host URL http://www.example.com/nexus instead of http://localhost:8081/nexus.

Ensure your external host name (www.example.com) routes to your reverse proxy server.

**Apache httpd.**

|  |
| --- |
| ProxyRequests Off  ProxyPreserveHost On    <VirtualHost \*:80>    ServerName www.example.com    ServerAdmin admin@example.com    ProxyPass /nexus http://localhost:8081/nexus    ProxyPassReverse /nexus http://localhost:8081/nexus    ErrorLog logs/www.example.com/nexus/error.log    CustomLog logs/www.example.com/nexus/access.log common  </VirtualHost> |

**nginx.**

|  |
| --- |
| http {        proxy\_send\_timeout 120;      proxy\_read\_timeout 300;      proxy\_buffering    off;      keepalive\_timeout  5 5;      tcp\_nodelay        on;        server {          listen   \*:80;          server\_name  www.example.com;            # allow large uploads of files - refer to nginx documentation          client\_max\_body\_size 1G;            # optimize downloading files larger than 1G - refer to nginx doc before adjusting          #proxy\_max\_temp\_file\_size 2G;            location /nexus {              proxy\_pass http://localhost:8081/nexus;              proxy\_set\_header Host $host;              proxy\_set\_header X-Real-IP $remote\_addr;              proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;          }      }  } |

## Example: Reverse Proxy Virtual Host at Base Path

**Scenario:** You need to expose the repository manager using a custom host name of repo.example.com on a restricted port at a base path of slash ( / ).

Ensure your external host name ( repo.example.com ) routes to your reverse proxy server and edit the webapp path to be slash ( / ) .

**Apache httpd.**

|  |
| --- |
| ProxyRequests Off  ProxyPreserveHost On    <VirtualHost \*:80>    ServerName repo.example.com    ServerAdmin admin@example.com    ProxyPass / http://localhost:8081/    ProxyPassReverse / http://localhost:8081/    ErrorLog logs/repo.example.com/nexus/error.log    CustomLog logs/repo.example.com/nexus/access.log common  </VirtualHost> |

**nginx.**

|  |
| --- |
| http {        proxy\_send\_timeout 120;      proxy\_read\_timeout 300;      proxy\_buffering    off;      keepalive\_timeout  5 5;      tcp\_nodelay        on;        server {          listen   \*:80;          server\_name  repo.example.com;            # allow large uploads of files - refer to nginx documentation          client\_max\_body\_size 1G;            # optimize downloading files larger than 1G - refer to nginx doc before adjusting          #proxy\_max\_temp\_file\_size 2G;            location / {              proxy\_pass http://localhost:8081/;              proxy\_set\_header Host $host;              proxy\_set\_header X-Real-IP $remote\_addr;              proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;          }      }  } |

## Example: Reverse Proxy SSL Termination at Base Path

**Scenario:** Your organization has standardized on a reverse proxy to handle SSL certificates and termination. The reverse proxy virtual host will accept HTTPS requests on the standard port 443 and serve content from the repository manager running on the default non-restricted HTTP port 8081 transparently to end users.

Ensure your external host name (repo.example.com) routes to your reverse proxy server and edit the webapp path to be slash ( / ) .

To test your configuration, we offer a [quick reference on how to generate self-signed SSL certificates](https://support.sonatype.com/entries/95353268-SSL-Self-Signed-Certificate-Guide) for reverse proxy servers.

**Apache httpd**. Ensure Apache httpd is loading mod\_ssl and mod\_headers.

|  |
| --- |
| Listen 443    ProxyRequests Off  ProxyPreserveHost On    <VirtualHost \*:443>    SSLEngine on      SSLCertificateFile "example.pem"    SSLCertificateKeyFile "example.key"      ServerName repo.example.com    ServerAdmin admin@example.com    ProxyPass / http://localhost:8081/    ProxyPassReverse / http://localhost:8081/    RequestHeader set X-Forwarded-Proto "https"      ErrorLog logs/repo.example.com/nexus/error.log    CustomLog logs/repo.example.com/nexus/access.log common  </VirtualHost> |

**nginx**. Make sure nginx is compiled using the --with-http\_ssl\_module option.

|  |
| --- |
| http {        proxy\_send\_timeout 120;      proxy\_read\_timeout 300;      proxy\_buffering    off;      keepalive\_timeout  5 5;      tcp\_nodelay        on;        server {          listen   \*:443;          server\_name  repo.example.com;            # allow large uploads of files - refer to nginx documentation          client\_max\_body\_size 1G;            # optimize downloading files larger than 1G - refer to nginx doc before adjusting          #proxy\_max\_temp\_file\_size 2G;            ssl on;          ssl\_certificate      example.pem;          ssl\_certificate\_key  example.key;            location / {              proxy\_pass http://localhost:8081/;              proxy\_set\_header Host $host;              proxy\_set\_header X-Real-IP $remote\_addr;              proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;              proxy\_set\_header X-Forwarded-Proto "https";          }      }  } |

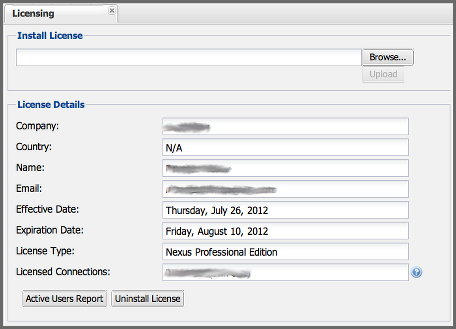
Reverse proxy configuration is going to vary and can get complex. Always consult the specific reverse proxy product documentation. [Apache httpd](http://httpd.apache.org/) ( mod\_proxy , mod\_ssl ), nginx ( ngx\_http\_proxy\_module , ssl compatibility )

# Installing a Nexus Repository Manager Pro License

**Available in Nexus Repository Pro only**

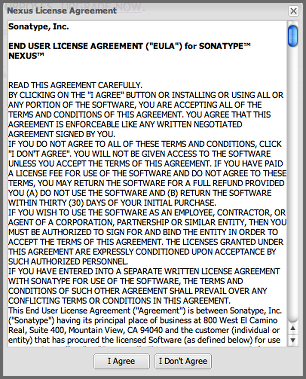
When starting a Nexus Repository Manager Pro trial installation you can upload your license file as described in Section 3.5, “Running” on the license screen visible in Figure 3.2, “License Activation”.

If you are currently using an evaluation license or need to replace your current license with a new one, click on Licensing in the  
Administration menu. This will bring up the panel shown in Figure 3.6, “Nexus Repository Manager Pro Licensing Panel” . To upload your  
Nexus Repository Manager Pro license, click on Browse…, select the file, and click on Upload.



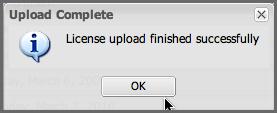
**Figure 3.6. Nexus Repository Manager Pro Licensing Panel**

Once you have selected a license and uploaded it to the repository manager, Nexus Repository Manager Pro will display a dialog box with  
the Nexus Repository Manager Pro End User License Agreement as shown in Figure 3.7, “Nexus Repository Manager Pro End User License Agreement” . If you agree with the terms and conditions, click on "I Agree".



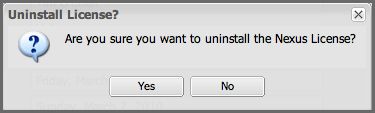
**Figure 3.7. Nexus Repository Manager Pro End User License Agreement**

Once you have agreed to the terms and conditions contained in the End User License Agreement, Nexus Repository Manager Pro will then display a dialog box confirming the installation of a Nexus Repository Manager Pro license, as shown in Figure 3.8, “License Upload Finished Dialog”.



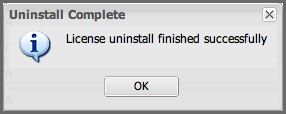
**Figure 3.8. License Upload Finished Dialog**

If you need to remove your Nexus Repository Manager Pro license, you can click on the "Uninstall License" button at the bottom of the Licensing  
Panel. Clicking on this button will show the dialog in Figure 3.9, “Uninstall License Confirmation Dialog”, confirming that you want to uninstall a license.



**Figure 3.9. Uninstall License Confirmation Dialog**

Clicking Yes in this dialog box will uninstall the license from Nexus Repository Manager Pro and display another dialog which confirms that the  
license has been successfully uninstalled.



**Figure 3.10. License Uninstall Completed Dialog**

Clicking on the Active Users Report button shows a list of IP numbers that connected to the repository manager in the last 7 days.

## License Expiration

When a Nexus Repository Manager Pro license expires, the user interface will have all functionality disabled except for the ability to install a new license file.

# Directories

The following sections describe the various directories that are a part of any Nexus Repository Manager Pro and Nexus Repository Manager OSS installation. When you install Nexus Repository Manager OSS or Nexus Repository Manager Pro, you are creating two directories: a directory containing the runtime and application often symlinked as nexus and a directory containing your own configuration and data - sonatype-work/nexus. When you upgrade to a newer version of Nexus Repository Manager, you replace the application directory and retain all of your own custom configuration and repository data in sonatype-work/.

## Sonatype Work Directory

The Sonatype Work directory sonatype-work is created as a sibling to the nexus application directory, and the location of this directory can be configured via the nexus.properties file which is described in “Configuration Directory”.

The Sonatype Work directory sonatype-work/nexus/ contains a number of subdirectories. Depending on the plugins installed and used, some directories may or may be not present in your installation:

**access/**

This directory contains a log of all IP addresses accessing the repository manager. The data can be viewed by clicking on Active Users Report in the Administration - Licensing tab in the user interface.

**aether-local-repository/**or**maven2-local-repository**

This holds temporary files created when running Maven dependency queries in the user interface.

**backup/**

If you have configured a scheduled job to back up configuration, this directory is going to contain a number of ZIP archives that contain snapshots of the configuration. Each ZIP file contains the contents of the conf/ directory. (Automated backups are a feature of Nexus Repository Manager Pro.)

**broker/**

The broker directory and its subdirectories contains the storage backend for the Smart Proxy messaging component.

**conf/**

This directory contains the configuration. Settings that define the list of repositories, the logging configuration, the staging and procurement configuration, and the security settings are all captured in this directory.

**conf/keystore/**

Contains the automatically generated key used to identify this repository manager for Smart Proxy usage

**db/**

Contains the database storing the User Token information, if that feature is enabled.

**error-report-bundles/**

Used to contain the bundled archives of data assembled for problem reporting. Since this feature has been removed this folder can be safely deleted.

**felix-cache/**

This directory holds the cache for the OSGi framework Apache Felix, which is used for the repository manager plugin architecture.

**health-check/**

Holds cached reports from the Repository Health Check plugin.

**indexer/** and **indexer-pro/**

Contains an index for all repositories and repository groups managed by repository manager. An index is a Lucene index which is the standard for indexing and searching a Maven repository. The repository manager maintains a local index for all repositories, and can also download an index from remote repositories.

**logs/**

The nexus.log file that contains information about a running instance of the repository manager. This directory also contains archived copies of log files. Log files are rotated every day. To reclaim disk space, you can delete old log files from the logs directory.

**nuget/**

Contains the database supporting queries against NuGet repositories used for .NET package support.

**p2/**

If you are using the P2 repository management features of Nexus Repository Manager Pro, this directory contains a local cache of P2 repository components.

**plugin-repository/**

This directory contains any additionally installed plugins from third parties as documented in “Managing Plugins” .

**proxy/**

Stores data about the files contained in a remote repository. Each proxy repository has a subdirectory in the proxy/attributes/ directory and every file that the repository manager has interacted with in the remote repository has an XML file that captures the last requested time stamp, the remote URL for a particular file, the length of the file, the digests for a particular file, and others. If you need to backup the local cached contents  
of a proxy repository, you should also back up the contents of the proxy repository’s directory under proxy/attributes/

**storage/**

Stores components and metadata repositories. Each repository is a subdirectory that contains the components in a repository. If the repository is a proxy repository, the storage directory will contain locally cached components from the remote repository. If the repository is a hosted repository, the storage directory will contain all components in the repository. If you need to back-up the contents of a repository, you should back up the contents of the storage directory.

**support/**

The support zip archive documented in “Support Tools” is created and stored in this folder.

**template-store/**

Contains the Maven settings template files documented in detail in "Managing Maven Settings" .

**timeline/**

Contains an index that the repository manager uses to store events and other information to support internal operations. The user interface exposes this data with the system feeds.

**tmp/**

Folder used for temporary storage.

**trash/**

If you have configured scheduled jobs to remove snapshot components or to delete other information from repositories, the deleted data will be stored in this directory. To empty this trash folder, view a list of repositories, and then click on the Trash icon in the user interface.

The conf/ directory contains a number of files which allow for configuration and customization of the repository manager. All of the files contained in this directory are altered by the administrative user interface. While you can change the configuration settings contained in these files with a text editor, Sonatype recommends that you modify the contents of these files using the administrative user interface. Depending on your version of the repository manager and the installed plugins, the complete list of files may differ slightly.

**broker.groovy**

A groovy script for configuring low-level properties for Smart Proxy.

**capabilities.xml**

Further Smart Proxy backend configuration.

**healthcheck.properties**

Configuration for the Repository Health Check.

**logback.properties**, **logback.xml**, and **logback-\*.xml**

Contains logging configuration. If you need to customize the detail of log messages, the frequency of log file rotation, or if you want to connect your own custom logging appenders, you should edit the logback-nexus.xml configuration file as desired. If you find log4j.properties files as well, you can safely remove them since they are remnants from an old version and are not used anymore.

**lvo-plugin.xml**

Contains configuration for the latest version plugin. This XML file contains the location of the properties file that the repository manager queries to check for a newer version.

**nexus.xml**

The bulk of the configuration is contained in this file. This file maintains a list of repositories and all server-wide configuration like the SMTP settings, security realms, repository groups, targets, path mappings and others.

**pgp.xml**

Contains PGP key server configuration.

**nexus-obr-plugin.properties**

Contains configuration for the Nexus OSGi Bundle repository plugin in Nexus Repository Manager Pro.

**procurement.xml**

Contains configuration for the procurement plugin in Nexus Repository Manager Pro.

**security-configuration.xml**

Contains global security configuration.

**security.xml**

Contains security configuration about users and roles.

**staging.xml**

Contains configuration for the Nexus Staging Plugin in Nexus Repository Manager Pro.

## Configuration Directory

After installing the repository manager and creating the nexus symlink as described earlier, your nexus folder contains another conf directory. This directory contains configuration for the Jetty servlet container. You will only need to modify the files in this directory if you are customizing the configuration of Jetty servlet container or the behavior of the scripts that start the repository manager.

The files and folders contained in this directory are:

**nexus.properties**

This file contains configuration variables which control the behavior of the repository manager and the Jetty servlet container. If you are customizing the port and host that the repository manager listens to, you change the application-port and application-host properties defined in this file. If you want to customize the location of the sonatype-work directory, you modify the value of the nexus-work property in this  
configuration file. Changing nexus-webapp-context-path allows you to configure the server context path where repository manager lives.

**jetty.xml** and **jetty-\*.xml**

Configuration files for the Eclipse Jetty servlet container running the repository manager. Jetty users are used to providing a list of jetty XML config files which are merged to form the final configuration. As an advanced configuration option, the repository manager supports this merging concept in its launcher code as of version 2.8.

You can specify additional jetty XML configuration files to load to form the final configuration. For the standard distribution bundle, these files can be specified using special properties located in NEXUS\_HOME/bin/jsw/conf/wrapper.conf.

|  |
| --- |
| wrapper.app.parameter.1=./conf/jetty.xml  wrapper.app.parameter.2=./conf/jetty-requestlog.xml  # add more indexed app parameters... |

Any of the files located at NEXUS\_HOME/conf/jetty-\*.xml can be specified as part of the wrapper.app.parameter.n property, where n is the next highest number not already used. The [Java Service Wrapper](http://wrapper.tanukisoftware.com/doc/english/prop-app-parameter-n.html) documentation contains more information about this property. This setup allows for a simple method to add configuration for https, JMX and others by adjusting a few properties.

Versions of Nexus Repository Manager Pro and Nexus Repository Manager OSS prior to 2.8 loaded all of the Jetty configuration from one jetty.xml file, typically found at NEXUS\_HOME/conf/jetty.xml and required modifications to this file for configuration changes. Examples were available in NEXUS\_HOME/conf/examples. These files cannot be used in version 2.8 or higher, as they were intended to be standalone files that could not be merged into other files.

# Monitoring

Now that your repository manager instance is up and running, you need to ensure that it stays that way. Typically this is done on a number of levels and each organization and system administration team has its own preferences and tools.

In general you can monitor:

* hardware values like CPU, memory or diskspace utilization and many more
* operating system level values like processes running
* Java Virtual Machine specific values
* application specific value

For the hardware and operating system values, a large number of dedicated tools exist. Many of these tools can be configured to work  
with application-specific logs and other events. The following section discusses some of the available information in the repository manager. It can potentially be integrated into the usage of the more generic tools for monitoring, log capturing and analysis.

A host of information from the operating system, the Java Virtual Machine and the application itself is available via the Support Tools , which allow you to inspect the value directly in the user interface.

## General Logging

The repository manager logs events in the sonatype-work/nexus/logs/nexus.log file. In addition a dedicated user interface to configure and inspect the log is available. Further information about this interface can be found in Section 6.10, “Logging” .

## Request Access Logging

Logging all access requests to the repository manager allows you to gain a good understanding of the usage in your organization and the sources of these requests.

For example, you will be able to tell if the main load is due to a CI server cluster or from your developers, based on the IP numbers of the requests. You can also see the spread or requests and load across different time zones. Also available for review are the URLs , API calls, and features that are used in the repository manager.

Requests access logging is enabled by default in version 2.8 or higher and uses a performant and flexible LogBack implementation with built-in log rotation already configured for 90 days of log file retention. The log is written to the file sonatype-work/nexus/logs/request.log and contains all requests and the username for authenticated requests.

The configuration is located in NEXUS\_HOME/conf/logback-access.xml and can be changed to suit your requirements. If you change the file, a restart of the repository manager is required for these changes to take effect.

If you do not want to run access logging, you can disable it by commenting out the line in bin/jsw/conf/wrapper.conf.

|  |
| --- |
| wrapper.app.parameter.2=conf/jetty-requestlog.xml |

Older versions of Nexus Repository Manager Pro and Nexus Repository Manager OSS require different customization of the Jetty configuration files. Instructions for these customizations can be found on the support site.

## Using Java Management Extension JMX

JMX is a common tool for managing and monitoring Java applications with client software like the free VisualVM  and many others available. It can be performed locally on the server as well as remotely.

The repository manager can be configured to support JMX by adding

|  |
| --- |
| wrapper.app.parameter.3=./conf/jetty-jmx.xml |

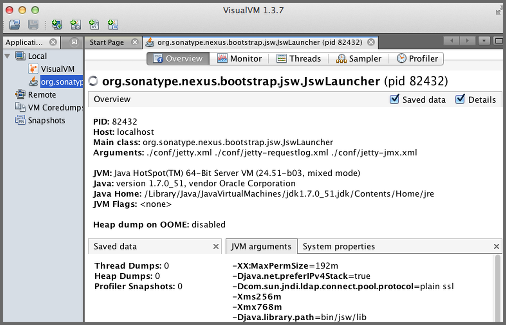
to the list of wrapper.app parameters in NEXUS\_HOME/bin/jsw/conf/wrapper.conf and set the parameters jmx-host and jmx-port in NEXUS\_HOME/conf/nexus.properties.

|  |
| --- |
| jmx-host=192.168.10.12  jmx-port=1099 |

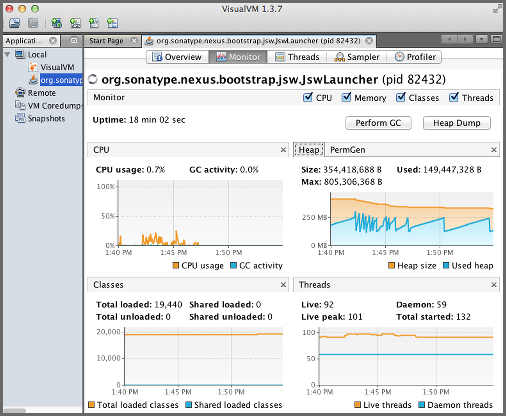
jmx-host is the host name, or commonly the IP address, to remotely monitor the application using JMX from another host and jmx-port is the network port used for the connection. It is important to ensure that the port is not blocked by any network setup, when connecting remotely. The value of 1099 is the default port used for JMX, but any other available port can be used as well.

Versions older than 2.8 require different procedures, depending on the specific version.

Once the repository manager is restarted with JMX enabled you can inspect the running JVM in detail. Figure 3.11, “Overview of Nexus Repository Manager Monitored via JMX in VisualVM” and Figure 3.12, “CPU, Memory and Other Visualizations of Nexus Repository Manager Monitored via JMX in VisualVM” show some example screenshots of VisualVM connected to a repository manager instance running on localhost.



**Figure 3.11. Overview of Nexus Repository Manager Monitored via JMX in VisualVM**



**Figure 3.12. CPU, Memory and Other Visualizations of Nexus Repository Manager Monitored via JMX in VisualVM**

Depending on the tool used to connect, a number of monitoring, analysis and troubleshooting actions can be performed. Please refer to the documentation about your specific tool for more information.

## Analytics

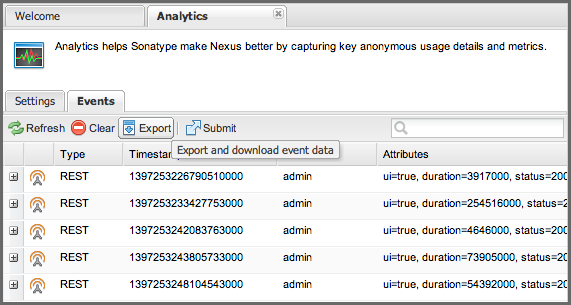
The analytics integration of Nexus Repository Manager allows you to gather a good understanding of your usage, since it enables the  
collection of event data in the repository manager. It collects non-sensitive information about how you are using the repository manager. It is useful to you from a compatibility perspective, since it gathers answers to questions such as what features are most important, where are users having difficulties, and what integrations/APIs are actively in use.

The collected information is limited to the use of the user interface and the REST API, the primary interaction points between your environment and the repository manager. Only the user interface navigation flows and REST endpoints being called are recorded. None of the request specific data (e.g., credentials or otherwise sensitive information) is ever captured.

You can enable the event logging in the Settings section of the Analytics tab available via Analytics menu item in the Administration menu in the left side navigation. Select the checkbox beside Enable analytics event collection and press the Save button.

You can choose to provide this data automatically to Sonatype by selecting the checkbox beside Enable automatic analytics event submission. It enables Sonatype to tailor the ongoing development of the product. Alternatively, you can submit the data manually or just use the gathered data for your own analysis only.

Once enabled all events logged can be inspected in the Events tab in the Analytics section displayed in Figure 3.13, “List of Events in the Analytics Tab” .



**Figure 3.13. List of Events in the Analytics Tab**

The list of events shows the Type and the Timestamp of the event as well as the User that triggered it and any Attributes . Each row has a + symbol in the first column that allows you to expand the row vertically. Each attribute will be expanded into a separate line allowing you to inspect all the information that is potentially submitted to Sonatype. The User value is replaced by a salted hash so that no username information is  
transmitted. The Anonymization Salt is automatically randomly generated by the repository manager and can optionally be configured in the Analytics: Collection capability manually. This administration area can additionally be used to change the random identifier for the repository manager instance.

More information about capabilities can be found in “Accessing and Configuring Capabilities”.

If you desire to further inspect the data that is potentially submitted, you can select to download the file containing the JSON files in a zip archive by clicking the Export button above the events list and downloading the file. The Submit button can be used to manually submit the events to Sonatype.

When you select to automatically submit the analytics data, a scheduled task, named Automatically submit analytics events , is automatically created. This task is preconfigured to run at 1:00 AM every day. If desired the recurrence can be changed in the scheduled tasks administration area documented in “Managing Scheduled Tasks” .

Sonatype values your input greatly and hopes you will activate the analytics feature and the automatic submission to allow us to ensure ongoing development is well aligned with your needs. In addition, we appreciate any further direct contact and feedback in person and look forward to hearing from you.