

# System Requirements

Created by Dulani Wallace, last modified by Rich Seddon on Feb 23, 2018

- [Supported Versions](#)
- [Host Operating System](#)
- [Dedicated Operating System User Account](#)
- [Adequate File Handle Limits](#)
  - [Linux](#)
  - [Mac OSX](#)
  - [Windows](#)
- [CPU](#)
- [Memory](#)
  - [General Memory Guidelines](#)
  - [Instance Sizing Profiles](#)
  - [Example Maximum Memory Configurations](#)
  - [Advanced Database Memory Tuning](#)
- [Disk](#)
  - [File Systems to avoid](#)
- [Web Browser](#)

## Supported Versions

Sonatype fully supports versions of repository manager for one year after the release date. Older releases are supported on a best effort basis and the release dates are listed in our download archives. The terms of support are explained in section 3 of the [End User License Agreement](#).


## Host Operating System


Any Windows, Linux or Macintosh operating system that can run a supported Sun/Oracle Java version will work. Other operating systems may work, but they are not tested by Sonatype. See [here](#) for the list of Oracle certified operating systems.

The most widely used operating system for Nexus Repository Manager (NXRM) is Linux and therefore customers should consider it the best tested platform.

## Dedicated Operating System User Account

Unless you are just testing the repository manager or running it only for personal use, a dedicated operating system user account is strongly recommended to run each unique process on a given host.

 The NXRM process user is typically named 'nexus' and must be able to create a valid shell.

 **Important**  
As a security precaution, do not run Nexus Repository Manager 3 as the root user.

## Adequate File Handle Limits

NXRM3 will most likely want to consume more file handles than the per user default value allowed by your Linux or OSX operating system.

Running out of file descriptors can be disastrous and will most probably lead to data loss. Make sure to increase the limit on the number of open files descriptors for the user running Nexus Repository Manager permanently to 65,536 or higher prior to starting.

See <https://issues.sonatype.org/browse/NEXUS-12041> for additional background.

### Linux

On most Linux systems, persistent limits can be set for a particular user by editing the `/etc/security/limits.conf` file. To set the maximum number of open files for both soft and hard limits for the nexus user to 65536, add the following line to the `/etc/security/limits.conf` file, where "nexus" should be replaced with the user ID that is being used to run the repository manager:

```
nexus - nofile 65536
```

This change will only take effect the next time the nexus process user opens a new session. Which essentially means that you will need to restart NXRM.

On Ubuntu systems there is a caveat: Ubuntu ignores the `/etc/security/limits.conf` file for processes started by `init.d`.

So if NXRM is started using `init.d` there, edit `/etc/pam.d/common-session` and uncomment the following line ( remove the hash # and space at the beginning of the line):

```
# session    required    pam_limits.so
```

For more information refer to your specific operating system documentation.

If you're using `systemd` to launch the server the above won't work. Instead, modify the configuration file to add a `LimitNOFILE` line:

```
[Unit]
Description=nexus service
After=network.target

[Service]
Type=forking
LimitNOFILE=65536
ExecStart=/opt/nexus/bin/nexus start
ExecStop=/opt/nexus/bin/nexus stop
User=nexus
Restart=on-abort
```

```
[Install]
WantedBy=multi-user.target
```

## Mac OSX

The method to modify the file descriptor limits on OSX has changed a few times over the years. Please note your OS X version and follow the appropriate instructions.

### For OS X Yosemite (10.10) and newer

1. Create the file: `/Library/LaunchDaemons/limit.maxfiles.plist`

```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Label</key>
    <string>limit.maxfiles</string>
    <key>ProgramArguments</key>
    <array>
      <string>launchctl</string>
      <string>limit</string>
      <string>maxfiles</string>
      <string>65536</string>
      <string>65536</string>
    </array>
    <key>RunAtLoad</key>
    <true/>
    <key>ServiceIPC</key>
    <false/>
  </dict>
</plist>
```

If this file already exists, then ensure the value is **at least** 65536 as shown.

The file must be owned by `root:wheel` and have permissions `-rw-r--r--`.

```
sudo chmod 644 /Library/LaunchDaemons/limit.maxfiles.plist
sudo chown root:wheel /Library/LaunchDaemons/limit.maxfiles.plist
```

**Reboot the operating system to activate the change.**

2. Add a new line to `$install-dir/bin/nexus.vmoptions` containing:

```
-XX:-MaxFDLimit
```

**Restart NXRM to activate the change.**

### For OS X Lion (10.7) up to OS X Mavericks (10.9)

1. Create and edit the system file `/etc/launchd.conf` using this command:

```
sudo sh -c 'echo "limit maxfiles 65536 65536" >> /etc/launchd.conf'
```

**Reboot the operating system to activate the change.**

2. Add a new line to `$install-dir/bin/nexus.vmoptions` containing:

```
-XX:-MaxFDLimit
```


**Restart NXRM to activate the change.**

## Windows

Windows operating systems do not need file handle limit adjustments.

## Java

Nexus Repository Manager 3 is a Java server application that requires specific versions to operate.

 We strongly suggest using the latest Java 8 release version of Java available from Oracle.  
**Support for Java 9 has not been verified - do not use it.**

OpenJDK has been known to work in most cases, but we do not actively test it. GCJ will not work and is not supported.

IBM JDK can be used on systems that do not have a Sun/Oracle JRE available, but our support of this is done on a best effort basis.

## CPU

NXRM performance is primarily bounded by IO (disk and network) rather than CPU. So any reasonably modern 4 core (or better) CPU will generally be sufficient for normal uses of NXRM.

## Memory

The default JRE min and max heap size of NXRM3 is pre-configured to be 1200MB, which should be considered an absolute minimum. The codebase will consume approximately another 1GB. So factoring in operating system overhead you will need at least 4GB of RAM on a dedicated NXRM host, assuming no other large applications are running on the machine.

Increasing the JVM heap memory larger than recommended values in an attempt to improve performance is **not recommended**. This actually can have the opposite effect, causing the operating system to thrash needlessly.

Unless you have evidence that a max heap of 4GB is consistently utilized and/or there are frequent lengthy garbage collection pauses that cannot be explained by software bugs, then **do not set max heap size larger than 4GB**.

General Memory Guidelines

- set minimum heap should always equal set maximum heap
- minimum heap size 1200MB
- maximum heap size <= 4GB
- minimum MaxDirectMemory size 2GB
- minimum unallocated physical memory should be no less than 1/3 of total physical RAM to allow for virtual memory swap
- max heap + max direct memory <= host physical RAM \* 2/3

Instance Sizing Profiles

Here are instance profiles you can use to gauge the typical physical memory requirements needed for a dedicated server host running repository manager. *Due to the inherent complexities of use cases, one size does not fit all and this should only be interpreted as a guideline.*

Profile Use Case	Physical Memory RAM
small, personal repositories < 20 total blobstore size < 20GB single repository format type	4GB minimum
medium, team repositories < 50 total blobstore size < 200GB a few repository formats	8GB
large, enterprise repositories > 50 total blobstore size > 200GB diverse set of repository formats	16GB+

Example Maximum Memory Configurations

Physical Memory	Example Maximum Memory Configuration
4GB	-Xms1200M -Xmx1200M -XX:MaxDirectMemorySize=2G
8GB	-Xms2703M -Xmx2703M -XX:MaxDirectMemorySize=2703M
12GB	-Xms4G -Xmx4G -XX:MaxDirectMemorySize=4014M
16GB	-Xms4G -Xmx4G -XX:MaxDirectMemorySize=6717M
32GB	-Xms4G -Xmx4G -XX:MaxDirectMemorySize=17530M
64GB	-Xms4G -Xmx4G -XX:MaxDirectMemorySize=39158M

Advanced Database Memory Tuning

Refer to another article which outlines [additional memory tuning procedures](#).

Disk

Nexus Repository Manager 3 installed consumes around 500 MB. The bulk of disk space will be held by your deployed and proxied artifacts, as well as any search indexes. This is highly installation specific, and will be dependent on the repository formats used, the number of artifacts stored, the size of your teams and projects, etc. It's best to plan for a lot though, formats like Docker and Maven can use very large amounts of storage (500Gb easily).

File Systems to avoid

File systems known to be unreliable are:

1. glusterfs
2. FUSE based user space filesystems

Additionally, we strongly recommend to avoid using NFS and Amazon EFS for anything other than blob storage in Nexus Repository 3.x, especially in large installations, as this can cause severe performance degradation. If NFS is used for blob storage we recommend to only use NFSv4, NFSv3 is known to provide inadequate performance. We also have some [optimization suggestions to use at your discretion](#). Also consider the `noatime` option for your Nexus Repository work directory mounts and limit the symbolic links used as this will cause increased overhead whenever paths need to be resolved to an absolute file path.

Web Browser

Our general policy is to support the most recent modern browser version for your supported OS at time of NXRM release date. This table is updated for the most recent NXRM release.

Vendor	Versions
--------	----------

Vendor	Versions
Google Chrome	latest at NXRM release
Firefox	latest and ESR at NXRM release
Safari	<b>on OSX only</b> , latest at NXRM release
Opera	<b>untested and not supported</b>
Internet Explorer	9, 10, 11 - <a href="#">Check this article for known issues</a>
Microsoft Edge	latest at NXRM release

No labels

Copyright© 2008-present, Sonatype Inc. All rights reserved. Includes the third-party code listed here. Sonatype and Sonatype Nexus are trademarks of Sonatype, Inc. Sonatype Nexus Repository Manager OSS™, Nexus Repository Manager Pro™, Nexus Lifecycle™, Nexus Auditor™, Nexus Firewall™, IQ Server™, and all Nexus-related logos as well as Sonatype CLM are trademarks or registered trademarks of Sonatype, Inc., in the United States and other countries.

Powered by a free **Atlassian Confluence Open Source Project License** granted to Sonatype. Evaluate Confluence today.  
This Confluence installation runs a Free Gliffy License - Evaluate the Gliffy Confluence Plugin for your Wiki!