

# SONATYPE NEXUS PROFESSIONAL

## Deployment Guidelines

Many organizations have successfully deployed the Sonatype *Nexus™ Professional (Nexus Pro)* repository manager. While the system design and architecture might seem difficult at first, it's quite straightforward so long as you follow a few basic guidelines.

This document provides guidance and tips we've learned after years of helping customers successfully deploy *Nexus Pro*. Follow these guidelines and you'll be successful too.

**If you need additional help, please don't hesitate to contact us at [www.sonatype.com/contact-us](http://www.sonatype.com/contact-us).**

### Summary of Deployment Guidelines

1. Have at least one repository manager in each location that has a CI system, a release manager or more than a handful of developers.
2. Locate the hosted repository for each project near the artifact producers.
3. Consider the underlying networking infrastructure when deciding how to configure proxy repositories.
4. You may need to make a trade off between repository control and performance.
5. Implement a highly available configuration when you need to ensure the repository is always available.
6. Utilize the enhanced proxy feature in Nexus 2.x to scale your proxy architecture.
7. Utilize either physical or virtual machines as both work well with Nexus.

### Detailed Deployment Guidelines

#### Where to Locate Repository Managers

The first thing to decide is where to locate repository managers. The answer depends on several factors as outlined in this section.

#### The Types and Number of Users at Each Location:

This is the primary criterion for determining repository locations. We strongly recommend locating a repository manager in each office that has a CI system, release manager or more than a handful of developers.

#### Artifact Production

The geographic dispersion of producers is important because it dictates whether you should use a federated or star pattern of hosted repositories. Each project should have one master hosted repository where artifacts are staged and deployed. Other repositories will be configured as read-only proxies of this master. The master

repository for each project is typically co-located with the producers for that project. Typically release managers and CI servers produce artifacts.

### **Star Pattern**

We recommend a star pattern when all of the producers in your organization are at the same location. You'll deploy a single master repository for the entire enterprise collocated with the producers, and proxy repositories at other locations.

### **Federated Pattern**

We recommend a federated pattern when you have producers at multiple locations. You'll deploy a master hosted repository at each location with producers and proxy repositories at locations with just artifact consumers.

## **Configuring Proxy Repositories**

Once you've decided where to locate repositories, the next step is to determine how to configure your proxies. This is influenced by the network infrastructure and Internet connectivity, as well as the degree of control you need to exercise over component usage.

### **Networking Infrastructure**

The underlying data networking infrastructure will impact the proxy configuration. You need to be sure that it can handle the expected data traffic both between repositories and between repositories and users.

If you have a star networking topology that requires all data traffic to go through a central location, then you probably will want to create a proxy repository in the central location. Remote offices will not proxy directly with each other, but rather through the intermediate proxy.

On the other hand, if you have fast network connections between all your offices, then you can set up direct proxies between locations as needed without worrying about the underlying infrastructure.

### **Internet Connectivity**

For maximum performance, configure the local repository manager to proxy for *The Central Repository*

directly if the location has a direct Internet connection. On the other hand, if the location accesses the Internet over an enterprise WAN, then you will probably want to proxy *The Central Repository* indirectly through another repository that does have a direct Internet connection.

### **Component Management vs. Performance**

The relative importance of component management versus performance will impact the proxy configuration.

You'll get the best performance by locating hosted repositories closest to producers, and locally proxying all other repositories, including *The Central Repository*. For maximum performance, each location with an adequate Internet connection should proxy *The Central Repository* directly because it is globally load balanced and artifacts will be fetched from the fastest *The Central Repository* server based on the proxy repository's location.

However, if you need to control which open source components are available to developers then you'll want to create a centrally managed proxy repository for *The Central Repository*. You can use the Procurement Suite along with Insight for Nexus to ensure this repository only has approved components that are free of license or security issues. Each location should have a local proxy of this controlled repository which developers will use to acquire approved open source components.

## **High Availability**

When your development teams are working on mission critical projects that cannot suffer a delay, you'll want to design the architecture to ensure high availability.

The architectures described in this section work best with the enhanced proxy capability available in *Nexus Professional 2.x* and later releases.

### **Maven Configuration for High Availability**

Deploying a highly available (HA) configuration requires the use of separate URLs for reading and writing artifacts. Whether or not you plan to deploy an HA configuration immediately, we recommend configuring Maven to use different host names for the deployment URL (set in the

POM), and the repository read URL (set in either the Maven settings file or the POM). If you are not yet using an HA configuration, then you can create an alias for your repository with a CNAME record entry in your DNS. By doing this, you'll be able to move to an HA configuration later without requiring changes to your developer's Maven environment.

### High Availability for Reading Artifacts

The following configuration ensures that components are always available for consumption using Active/ Active "reads" with load balancing. The release manager or CI Server continues to publish components directly to the master repository.

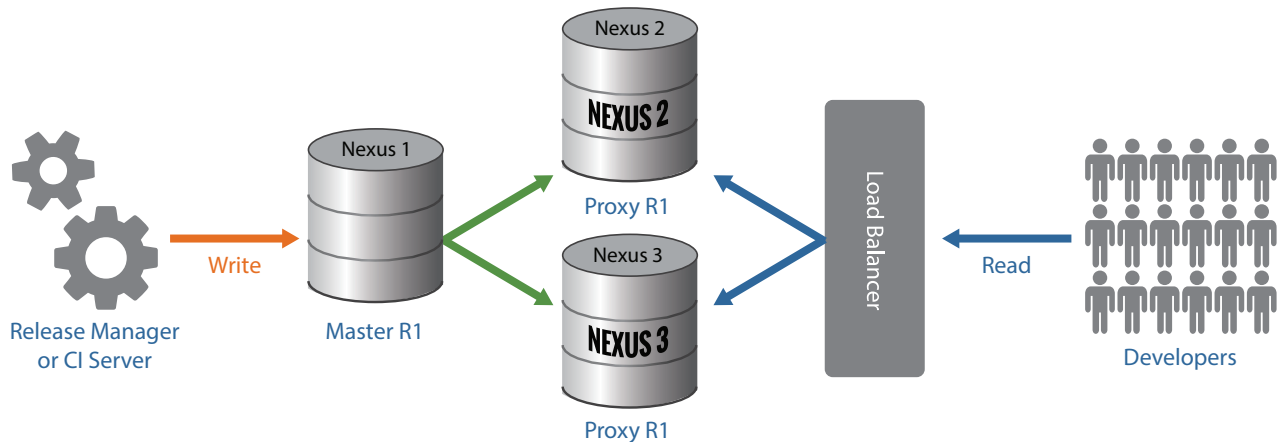


Figure 1. The highly available read configuration ensures artifacts are always available for consumption.

- You'll need two or more Nexus repository managers and a load balancer. Most users deploy 3 such servers. (Please remember that your Nexus license includes unlimited servers at no additional cost)
- The Nexus master contains the hosted repositories.
- The other two (or more) Nexus repository managers are configured as proxy repositories of the master.
- The load balancer sits in front of the proxy repository managers and balances the read load between them.
- Deployed artifacts are written directly to the hosted master repository.

This configuration provides a robust solution for ensuring that artifacts are always available for reading. This is the most important function to protect as the read load typically outweighs the write load by many orders of magnitude (thousands of developers are typically reading, only a very small set are writing).

### High Availability for Creating Artifacts

The following configuration will increase the availability of repository "writes" by using redundant, stand-by servers backed by a highly available file system.

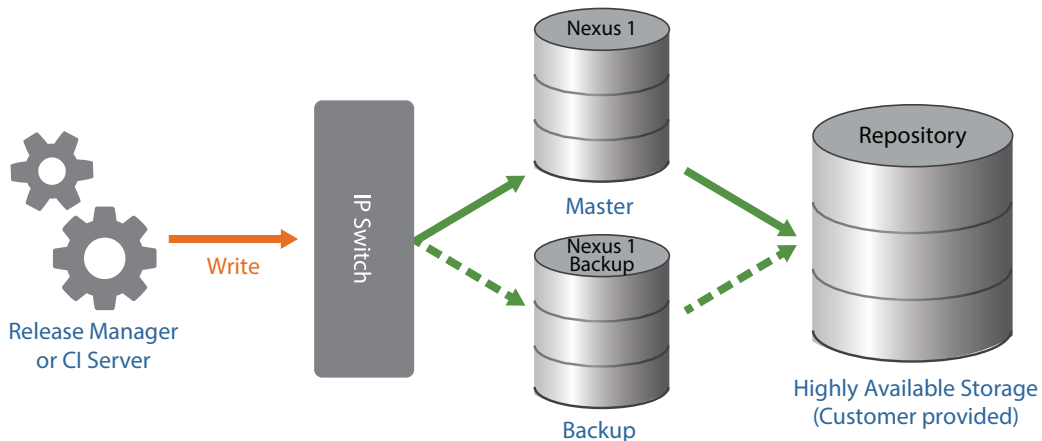


Figure 2. The highly available write configuration ensures the repository is always available for deploying artifacts.

- You'll need to configure two (or more) Nexus servers identically, sharing the same file system. Only one Nexus repository manager can be active at a time. The backup will utilize the same configuration files and point to the same repositories as the master.
- The file system should be highly available using an off-the-shelf solution.
- When Nexus 1 fails, Nexus 1 backup must be activated through a configuration change.
- An IP switch (or similar device) enables clients to continue using the same Name/IP address for the Nexus server.
- You can combine the two high availability approaches to create a system that provides high availability for both reads and writes.

### Scale Your Proxy Infrastructure with Smart Proxy

Large-scale, multi-site deployments of proxy repositories may overload master repositories using the traditional proxy mechanism. Nexus Pro 2.x includes an enhanced proxy that scales to support the largest deployments by pushing component update notifications from the master repository.

We recommend enabling the Smart Proxy feature when utilizing snapshot repositories. You should also ensure the time-outs are set at the default value of 1440 minutes (24 hours). This will reduce the load on your master repository while still ensuring that the most recent snapshots are available immediately around your organization.

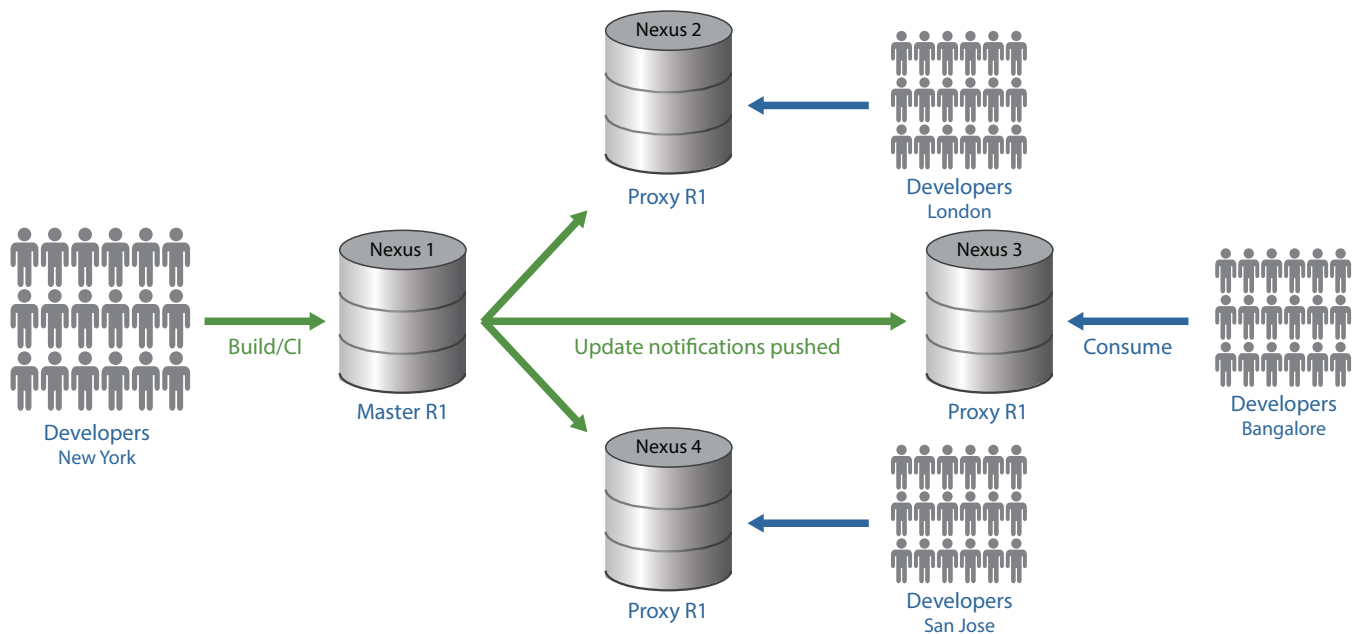


Figure 3. The smart proxy functionality in Nexus Pro 2.x scales for the largest proxy architectures by pushing update notifications from the master.

## Virtual or Physical Machine

Nexus supports both physical and virtual machines equally well as it doesn't require a lot of CPU or RAM to work effectively. At Sonatype, we've moved all of our managed forges over to virtual machines with the following specifications:

- 2 CPUs
- 3GB RAM
- 400GB disk (this is completely dependent on your repository contents)
- RHEL 5.6 x64 (Contegix, our managed hosting service, recommends using this OS)
- Java 1.6 x64 with 1GB Heap
- The virtual disk is located on a SAN connected with iSCSI over 1GBE

- For I/O performance, we recommend a redundant solution that maximizes disk spindles, while maintaining fault tolerance. We use RAID 50 in our SAN. A RAID 50 combines the straight block-level striping of RAID 0 with the distributed parity of RAID 5. It is a RAID 0 array striped across RAID 5 elements. It requires at least 6 drives.

These systems serve requests on the order of 1,400-2,500 requests per minute. Above that, the system typically needs to scale up in terms of network and IO optimization. Increasing the number of CPUs and amount of RAM can typically help as well. We do not recommend using NFS to mount a virtual disk or the working folder as many customers have had trouble with locking and corrupted indexes. iSCSI is working very well for us and for many of our customers.

## Additional assistance available in the Nexus Community at [www.Sonatype.org/Nexus](http://www.Sonatype.org/Nexus)

Please visit our community for easy access to support from us or your peers, product updates, insights from Nexus experts, free training and books, forums and much more.

---

Every day, developers rely on millions of third party and open source building blocks—known as components—to build the software that runs our world. Sonatype ensures that only the best components are used throughout the software development lifecycle so that organizations don't have to make the tradeoff between going fast and being secure. Policy automation, ongoing monitoring and proactive alerts makes it easy to have full visibility and control of components throughout the software supply chain so that applications start secure and remain that way over time. Sonatype is privately held with investments from New Enterprise Associates (NEA), Accel Partners, Bay Partners, Hummer Winblad Venture Partners and Morgenthaler Ventures. Visit: [www.sonatype.com](http://www.sonatype.com)