Upgrading from Nexus Repository Manager 2

Table of Contents

Nexus Repository 3 is a complete redesign of the repository manager and does not include any legacy code from Nexus Repository 2.

While Nexus Repository 2 was based on a Maven repository design that was dependant on how components were stored on the file system, Nexus Repository 3 uses blob storage, moving away from the idea of a repository being a folder in a file system. This makes Nexus Repository 3 far more scalable to different ecosystems and deployment environments, avoiding the file system limitations that some Nexus Repository 2 deployments encountered with flat directories containing tens of thousands of components common in some repositories.

Upgrading from Nexus Repository 2 to Nexus Repository 3 requires creating an upgrade plan:

1. Begin by [1]Determining and Planning Your Upgrade Path
    a. [2]Built-in Upgrade Wizard (Always Preferred)
    b. [3]Manual Import/Export (Preferred if Cannot Use Wizard)
    c. [4]Manually Proxying Old Repositories
2. Understand [5]Nexus Repository 2 vs. Nexus Repository 3 feature equivalency
3. Consider any [6]IQ Server or [7]Firewall integration impacts, especially if using a manual method
4. Plan to replace custom plugins and scripts using the Nexus Repository 3 API
5. If not already complete, upgrade your Nexus Repository 2 instance to the latest Nexus Repository version using [8]these instructions
6. If using the Upgrade Wizard, [9]decide on a data transfer method
    a. Whether using the Upgrade Wizard or Import, hard linking is always preferred; consider if you can remove any barriers to hard linking

1 https://help.sonatype.com/repomanager3/installation-and-upgrades/upgrading-from-nexus-repository-manager-2/determining-and-planning-your-upgrade-path

2 https://help.sonatype.com/repomanager3/installation-and-upgrades/upgrading-from-nexus-repository-manager-2/nexus-repository-3-upgrade-wizard

3 https://help.sonatype.com/repomanager3/installation-and-upgrades/upgrading-from-nexus-repository-manager-2/manual-upgrade-methods#ManualUpgradeMethods-Export/ImportManualUpgradeMethodBluePROexportimport

4 https://help.sonatype.com/repomanager3/installation-and-upgrades/upgrading-from-nexus-repository-manager-2/manual-upgrade-methods#ManualUpgradeMethods-StepsfortheProxyingRepositoriesManualUpgradeMethod

5 https://help.sonatype.com/repomanager3/installation-and-upgrades/upgrading-from-nexus-repository-manager-2/nexus-repository-2-vs.-nexus-repository-3-feature-equivalency-matrix

6 https://help.sonatype.com/repomanager3/installation-and-upgrades/upgrading-from-nexus-repository-manager-2/determining-and-planning-your-upgrade-path#DeterminingandPlanningYourUpgradePath-AreYouUsingIQServer?IQ

7 https://help.sonatype.com/repomanager3/installation-and-upgrades/upgrading-from-nexus-repository-manager-2/determining-and-planning-your-upgrade-path#DeterminingandPlanningYourUpgradePath-AreYouUsingNexusFirewall?firewall

8 https://support.sonatype.com/hc/en-us/articles/213464138?_ga=2.18566265.1925118275.1667825797-986364695.1656691738&_gac=1.182528468.1667572342.CjwKCAjw8JKbBhBYEiwAs3sxN48e7FmEPe_zEoJ4YRsGq1rT_X6zY5yQxB--1Z-tcSq5FjyTb4PwjxoCZKgQAvD_BwE

9 https://help.sonatype.com/repomanager3/installation-and-upgrades/upgrading-from-nexus-repository-manager-2/nexus-repository-3-upgrade-wizard/deciding-on-a-data-transfer-method

7. If using the Upgrade Wizard, complete [10]optimization steps
8. Ensure you meet all prerequisites
   a. [11]Upgrade Wizard Prerequisites
   b. [12]Manual Method Prerequisites

Once you have made the above decisions and created a plan, you are ready to upgrade!

✓ Nexus Repository Pro customers can also take advantage of support team assistance.

Nexus Repository 2 vs. Nexus Repository 3 Feature Equivalency Matrix

Nexus Repository 3 represents a complete rewrite of architecture and functionality. Some features from Nexus Repository 2 are no longer available in Nexus Repository 3. The matrix below provides details about features that have been replaced or removed in Nexus Repository 3.

Repository 2 Feature	Repository 3 Equivalent	Note
Server Settings	✓ Yes	These are settings under *Administration* → *Server* in Nexus Repository 2
Security	✓ Yes	Please refer to the[13]Security section
Local User Accounts	✓ Yes	
Privileges	✓ Yes	Names may change on upgrade from Nexus Repository 2 to 3. Note that transitive group privileges are implemented differently in Nexus Repository 2 and 3: In Nexus Repository 2, when a user is given a privilege to a group repository, then that user will also have that privilege to all transitive members of that group repository. This

10 https://help.sonatype.com/repomanager3/installation-and-upgrades/upgrading-from-nexus-repository-manager-2/nexus-repository-3-upgrade-wizard/optimizing-upgrade-performance
11 https://help.sonatype.com/repomanager3/installation-and-upgrades/upgrading-from-nexus-repository-manager-2/nexus-repository-3-upgrade-wizard/prerequisites-for-upgrading-from-nexus-repository-2-to-nexus-repository-3
12 https://help.sonatype.com/repomanager3/installation-and-upgrades/upgrading-from-nexus-repository-manager-2/manual-upgrade-methods#ManualUpgradeMethods-Prerequisites
13 https://help.sonatype.com/display/NXRM3/Security

Repository 2 Feature	Repository 3 Equivalent	Note
		permission access applies even if the HTTP requests are sent directly to the member repositories rather than the group repository. As repositories are added to or removed from a group, the user may gain or lose access to those individual repositories. (See more in[14]this Nexus Repository 2 support article .) In Nexus Repository 3, when a user is given a privilege to a group repository, then that user will also have that privilege to all transitive members of that group repository **only when their request is directed to the group repository**. Direct requests to indvidual member repositories will only work if the user is given explicit permission to the individual repository. The Upgrade Wizard does not detect this difference. After upgrading from Nexus Repository 2 to 3, you may need to manually assign privileges for individual repositories to a subset of users.
Roles	✓ Yes	
Maven Repositories	✓ Yes	
Proxy Repository of `maven.oracle.com`	✓ Yes	[15]See KB article to help getting this configured
Browse Remote	✗ No	
Repository Targets	✓ Yes	Maps to[16]content selectors
Manual Routing Rules	✓ Yes	See [17]Routing Rules

14 https://support.sonatype.com/hc/en-us/articles/221279107?_ga=2.108313093.609935302.1662465096-1261099841.1662465096
15 https://support.sonatype.com/hc/en-us/articles/213465728
16 https://help.sonatype.com/display/NXRM3/Repository+Management#RepositoryManagement-ContentSelectors
17 https://help.sonatype.com/repomanager3/nexus-repository-administration/repository-management/routing-rules

Repository 2 Feature	Repository 3 Equivalent	Note
Maven Settings	✗ No	
Automatic Routing	✗ No	We will consider similar optimization features like this in the future as warranted.
Smart Proxy	✗ No	We are planning a set of instance-to-instance replication features.
Procurement	✓ Yes	The Procurement feature, which is limited to Maven in Nexus Repository 2, has been deprecated. Going forward, Sonatype's [18]Nexus Firewall solution provides a far more robust and comprehensive set of features aimed at managing risky components via your repository.
Staging	✓ Yes	Nexus Repository 2.x Staging has been replaced with a more robust solution that relies on component tagging to manage workflow. See our Nexus Repository 3[19]Staging topic for details.
Custom Metadata	✓ Yes	Nexus Repository 3 has the ability to associate arbitrary data with any component in any repository format via it's[20]tagging feature. However, there is no automated way to migrate Nexus Repository 2[21]custom metadata.
YUM Repositories	✓ Yes	[22]YUM proxy support is included in the Nexus Repository 3.5.0 release and hosted support is included in the Nexus Repository 3.8.0 release. Feature was completed in 3.11.0.
p2 and p2 Update Site Repositories	✓ Yes	

18 https://www.sonatype.com/nexus-firewall
19 https://help.sonatype.com/display/NXRM3/Staging
20 https://help.sonatype.com/display/NXRM3/Tagging
21 https://support.sonatype.com/hc/en-us/articles/213465338-How-can-I-add-additional-metadata-to-an-artifact-stored-in-Nexus-?_ga=2.262676298.717644679.1669834604-291822469.1668525051
22 https://help.sonatype.com/display/NXRM3/Yum+Repositories

Repository 2 Feature	Repository 3 Equivalent	Note
RubyGems Repositories	✓ Yes	
PyPI Repositories	✓ Yes	
npm Repositories	✓ Yes	
NuGet Repositories	✓ Yes	
NuGet API Keys	✓ Yes	
Bower Repositories	✓ Yes	
Site Repositories	✓ Yes	This is called Raw repository format in Nexus Repository 3
System Feeds	✗ No	We have no plans to upgrade System Feeds as the data is specific to the instance where it was generated. Near equivalent functionality is available through[23] auditing and[24] webhooks.
Branding	✓ Yes	The Nexus Repository 2 branding configuration will not be upgraded automatically. Nexus Repository 3 uses the[25] Branding capability.
REST API	✓ Yes	See the Nexus Repository 3[26] REST API documentation. ⓘ Note that Nexus Repository 2 REST APIs are not compatible with Nexus Repository 3.
Non-Sonatype Repository Plugins	✗ No	Nexus Repository 2 plugins are not compatible with Nexus Repository 3 OSGI bundle architecture

23 https://help.sonatype.com/display/NXRM3/Auditing
24 https://help.sonatype.com/display/NXRM3/Webhooks
25 https://help.sonatype.com/display/NXRM3/Branding+Capability
26 https://help.sonatype.com/display/NXRM3/REST+and+Integration+API

Repository 2 Feature	Repository 3 Equivalent	Note
SSL Certificates	✓ Yes	SSL certificates trusted in the UI will be upgraded, but not all of them may be applicable to a new server. Trusted certificates should be reviewed after your upgrade. The Upgrade Wizard does not automatically migrate the certificate that Nexus Repository uses for its own HTTPS connection.
Analytics	✗ No	This little-used feature was removed in 3.15.
LDAP	✓ Yes	
Nexus Repository OSS Servers	✓ Yes	
Nexus Repository Pro (Enterprise) Servers	✓ Yes	Individual LDAP Server configurations will be upgraded. Backup Mirror servers will not be upgraded .
Mapped Users and Roles	✓ Yes	
Atlassian Crowd	✓ Yes	After your upgrade, you will need to ensure your Crowd application is configured to accept connections from Nexus Repository 3
Mapped User and Roles	✓ Yes	
User Token	✓ Yes	See section on User Token Upgrade(see page 4)
IQ Server Configuration	✓ Yes	The Upgrade Wizard will clone IQ Server configuration and expects that Nexus Repository 3 will use the same IQ Server as Nexus Repository 2. The data inside IQ Server for Nexus Repository 2 will be cloned to work with the repositories upgraded to Nexus Repository 3.

Repository 2 Feature	Repository 3 Equivalent	Note
Repository Health Check Analysis (RHC)	✓ Yes	As of release 3.55.0, those who have[27]enabled IQ Server under *Administration → IQ Server* and have[28]Sonatype Repository Firewall as a feature on that IQ Server instance will not see Repository Health Check in their Sonatype Nexus Repository instance. Firewall is a much more fully featured tool for identifying security risks in your repositories.[29]Learn more about Sonatype Repository Firewall on sonatype.com.
RUT Auth Realm	✓ Yes	This will NOT be upgraded automatically . This feature can be a security concern when enabled. Configure this manually inside Nexus Repository 3 when you are confident that Nexus Repository 3 access is protected.

Determining and Planning Your Upgrade Path

Upgrading from Nexus Repository 2 to Nexus Repository 3 requires a complete transformation of the repository metadata and storage methods. Because of this, you will need to use one of the following for upgrade:

- **Built-in Upgrade Wizard (Preferred Method)**
  - Best for those upgrading to a fresh Nexus Repository 3 instance
  - Not appropriate when you are already using Nexus Repository 3
- **Manual Methods**
  - **Exporting / Importing Content and Configuration (Preferred Manual Method)**
    - Best for those already using Nexus Repository 3 and wishing to bring Nexus Repository 2 content into the existing Nexus Repository 3 instance
    - Not appropriate for proxy repositories, content cannot be imported into a proxy repository
  - **Proxy Old Repositories**
    - Best for those only wishing to bring over what is actively being used
    - Not appropriate when you do not wish to maintain your Nexus Repository 2 instance
- **Hybrid Model**

27 https://help.sonatype.com/display/NXRM3/Connecting+IQ+Server
28 https://help.sonatype.com/display/FW/Firewall
29 https://www.sonatype.com/products/sonatype-repository-firewall-video

- Due to the complicated nature of this method, we do not recommend using a hybrid method unless you have access to our support and customer success teams.

⚠ We recommend using the built-in Upgrade Wizard whenever possible.

While the built-in Upgrade Wizard is always our top recommendation, we also recognize that it's not always possible to use this tool in every scenario. The questions below will help you determine the best upgrade path for your individual deployment.

Baseline Requirements & Limitations

The following questions are not considerations, but requirements to be able to use the specified upgrade path. You **must** be able to answer yes to all questions below in order to use the specified method:

Path	Requirements	Limitations
Upgrade Wizard (Preferred)	• Are you upgrading to a fresh Nexus Repository 3 instance? • Will you be using the same license type (OSS or Pro) on your Nexus Repository 3 instance as on your Nexus Repository 2 instance? • Can you ensure that your Nexus Repository 2 instance is on the latest Nexus Repository 2 version before upgrading? • Can you ensure that all files in the Nexus Repository work directory are owned by the OS user and that there are no zero length files in the work directory? (See Prerequisites for Upgrading from Nexus Repository 2 to Nexus Repository 3(see page 18) for more details.) • Can you ensure that Nexus Repository 2 repository and repository group Repository IDs differ by more than just case and use distinguishable names?	Cannot be used to upgrade to an existing Nexus Repository 3 instance.
Import/ Export (Preferred	Are you prepared for a potentially lengthy upgrade process?	Only moves repository contents, not configuration.

Path	Requirements	Limitations
Manual Method)		
Proxy Old Repositories	• Are you prepared for a prolonged upgrade process? • Are you prepared to keep your Nexus Repository 2 instance after this upgrade, potentially long-term? • Are you prepared to allow inbound network traffic from Nexus Repository 3 to Nexus Repository 2? • Are you ok with some components in repositories not being upgraded?	Only moves repository contents, not configuration.
Hybrid (Upgrade Wizard + Manual)	A common hybrid scenario would be using the Upgrade Wizard to move user tokens, then using manual methods for everything else. • Are you upgrading to a fresh Nexus Repository 3 instance? • Will you be using the same license type (OSS or Pro) on your Nexus Repository 3 instance as on your Nexus Repository 2 instance? • Can you ensure that your Nexus Repository 2 instance is on the latest Nexus Repository 2 version before upgrading? • Are you prepared for a potentially lengthy upgrade process?	Due to the complicated nature of this method, we do not recommend using a hybrid method unless you have access to our support and customer success teams.

Additional Considerations for Fine Tuning Your Path

RPMs in Maven Repositories

If you have RPMs in Maven repositories in Nexus Repository 2, we highly suggest using the Nexus Repository 3 import task to import these into[30]Yum hosted repositories in Nexus Repository 3.

When doing this, you will need to disable the[31]*Yum: Configuration* capability on any Maven repositories in Nexus Repository 2. If you do not disable this capability, the entire Maven 2 hosted repository will not import into a Maven 2 repository in Nexus Repository 3.

How much data do you need to move?

Whether using the Upgrade Wizard or Import, we always recommend using hard links when possible. Using hard linking will save both time and storage space.

See hard linking requirements(see page 21) when using the Upgrade Wizard or hard linking requirements when using[32]Import depending on which method you are using.

How are you using Nexus Repository 2 APIs, and how will you get that same functionality in Nexus Repository 3?

Custom plugins and scripts written using the Nexus Repository 2 APIs will not work for Nexus Repository 3. You will need to rewrite scripts using the Nexus Repository 3 API.

Nexus Repository 3 does support legacy content URLs; however, you should transition builds to default URLs as soon as possible.

Do You Want to Use Different Architecture (e.g., Resilient Deployment)?

It is best to upgrade to a Nexus Repository 3 instance using the same architecture as your Nexus Repository 2 deployment. If you wish to move to a different architecture (e.g., if you wish to use one of our[33]resilient deployment options), we recommend first upgrading to a Nexus Repository 3 instance that uses the same architecture as your Nexus Repository 2 deployment and then doing a separate upgrade to the new architecture.

---

30 https://help.sonatype.com/display/NXRM3/Yum+Repositories#YumRepositories-HostingYumRepositories
31 https://help.sonatype.com/repomanager2/rpm-packages-and-yum-repositories#RPMPackagesandYUMRepositories-Configuration
32 https://help.sonatype.com/display/NXRM3/Repository+Import
33 https://help.sonatype.com/display/NXRM3/Resiliency+and+High+Availability

Do You Wish to Use Cloud Object Stores?

While possible to do, moving to cloud object stores (e.g., S3 blob stores) takes two to three times longer than moving to other storage options. As a best practice, we recommend importing to a file-based blob store on a low-latency file system and following the instructions in[34]Moving a Blob Store to move the content to the cloud object store.

Are You Using IQ Server?

The Upgrade Wizard will clone IQ Server configuration and expects that Nexus Repository 3 will use the same IQ Server as Nexus Repository 2. The data inside IQ Server for Nexus Repository 2 will be cloned to work with the repositories upgraded to Nexus Repository 3.

Are You Using Nexus Firewall?

The Upgrade Wizard will move Nexus Firewall waivers and approved components to your new Nexus Repository 3 instance.

While you cannot manually move Firewall waivers and approval, you can pre-fetch the approved components before enabling Firewall quarantine on the new repository.

1. Export the Firewall report from the source repository using the[35]IQ Server REST API for Repository Results.
2. Use this report to request components from the new proxy repository.
3. Enable Firewall quarantine on the new repository once all of the allowed components have been proxied.

Nexus Repository 3 Upgrade Wizard

Nexus Repository includes upgrade capabilities which, when enabled, allow you to use an Upgrade Wizard that will walk you through migrating your Nexus Repository 2 configuration and repositories to a fresh Nexus Repository 3 instance.

Upgrading to Nexus Repository 3 using the Upgrade Wizard is always the preferred method.

The following sections will help you plan and perform your migration using this wizard.

1. Understand what is included in, excluded from, and modified during the upgrade(see page 14)

34 https://help.sonatype.com/repomanager3/nexus-repository-administration/repository-management/configuring-blob-stores#ConfiguringBlobStores-MovingaBlobStore
35 https://help.sonatype.com/iqserver/automating/rest-apis/experimental---repository-results-view-rest-api

2. Ensure you meet all prerequisites(see page 18); if you can't, you may need to reevaluate your upgrade path(see page 9)
3. Decide on a data transfer method(see page 21); if possible, we always recommend hard linking
4. Take steps to optimize upgrade performance(see page 23)
5. Configure the upgrade capabilities(see page 25) in your Nexus Repository 2 and 3 instances
6. Follow our step-by-step guide through the wizard itself(see page 27)
7. If you need to start over, reset your upgrade(see page 35)
8. Configure legacy URL paths(see page 37)

Data Included, Excluded, and Changed During Nexus Repository 2 to 3 Upgrade Process

The Upgrade Wizard will help guide you through the upgrade process as outlined in Step-by-Step Nexus Repository 2 to 3 Upgrade Wizard Instructions(see page 27). However, the wizard is only able to include certain configurations in the upgrade.

What is Included in the Upgrade?

The Upgrade capability upgrades the following automatically:

- Configuration
  - Active Security Realms
  - Anonymous Access Settings
  - Atlassian Crowd Settings
  - HTTP Proxy Settings
  - LDAP Settings
  - NuGet API Keys
  - Privileges
  - Repository Health Check and its corresponding SSL trust store configuration
    - For OSS, only *Health Check: Configuration* capability settings are included
    - For Pro, existing *SSL: Health Check* settings are also included
  - Role Mappings for LDAP/Crowd
  - Roles
  - SSL Certificates that have been trusted in SSL Certificates feature
  - Users
  - User Tokens
- IQ Server configuration
  - *IQ Server URL*
  - *Username*

- *Password*
- *Request Timeout*
- Analysis properties
- Trust store usage
- Enabled Nexus IQ Server connection
- Repositories configured with the *Firewall Audit and Quarantine* capability will keep the capabilities as well as the audited and quarantined items

While Nexus Repository 2 can keep running while upgrade occurs, changes to the content will only be upgraded through the *Synchronization* upgrade step. Changes to audited and quarantined items from the IQ Server are included in this caveat.

- Repositories in supported formats:
  - Maven2 (excluding staging repositories)
  - npm
  - NuGet
  - Site (called Raw in Nexus Repository 3)
  - RubyGems
- NuGet API key - The upgrade tool adds all keys from Nexus Repository 2 to Nexus Repository 3 even if the NuGet API Key Realm is not active in Nexus Repository 2
- The following existing Nexus Repository artifact information is maintained during upgrade: **NEW IN 3.37.0**
  - `uploaded_by`
  - `uploaded_by_ip`
  - `uploaded_date`
  - `lastModified`

Changes to Data During Upgrade Process

Nexus Repository 3 uses different architecture from Nexus Repository 2; therefore, some core changes to data occur as part of the upgrade process.

Content Storage

In Nexus Repository 2, all content is stored in a filesystem on disk.

Nexus Repository 3, however, uses a blob store format for storing content. This allows Nexus Repository 3 to work with more formats since they no longer have to map directly to a file system. See[36]Storage Planning for more details about how to work with blob stores.

Nexus Repository 3 must iterate over every component stored in Nexus Repository 2. This takes the bulk of the time required for the upgrade process.

Settings and Metadata

In Nexus Repository 2, settings and some component metadata are stored across many files.

Nexus Repository 3 loads equivalent data into the database.

URLs

Nexus Repository 2 uses a different URL pattern to expose repositories and repository groups than Nexus Repository 3. During upgrade, Nexus Repository 3 adds a *NXRM2 style URLs* capability to allow your automated tools and CI to use the old patterns. Configuring Legacy URL Paths(see page 37) covers this in more detail.

Roles

Roles upgraded from Nexus Repository 2 are assigned a Role ID that starts with `nx2-` in Nexus Repository 3. Role descriptions created during the upgrade process have the word `legacy` in their description.

Repository Targets

Repository targets from Nexus Repository 2 are converted to[37]Content Selectors in Nexus Repository 3. Repository targets were based on regular expressions; these are automatically converted to Content Selector Expression Language (CSEL), a performant subset of JEXL.

Repository IDs

Nexus Repository 3 uses the Repository ID defined in Nexus Repository 2 as the Name for the upgraded repository; they define the access URL in both cases. The user-facing Repository Name from Nexus Repository 2 is dropped during the upgrade since there is no equivalent feature in Nexus Repository 3 .

IDs of repositories and repository groups in Nexus Repository 2 that differ only by case will not be accepted during an upgrade to Nexus Repository 3 (example version 2 IDs: `myrepoid` vs `Myrepoid`). To resolve the ID conflict review and change any IDs in version 2 to distinguishable names.

36 https://help.sonatype.com/repomanager3/planning-your-implementation/storage-guide/storage-planning
37 https://help.sonatype.com/display/NXRM3/Content+Selectors

HTTP(S) Proxy Configuration

Your HTTP or HTTPS proxy settings for Nexus Repository 2 **may not be valid** for your Nexus Repository 3 environment. You must configure your HTTP or HTTPS proxy settings in Nexus Repository 3 in order to upgrade them.

If HTTP or HTTPS proxy settings are enabled in your source Nexus Repository 2, the upgrade to your target Nexus Repository 3 might fail because the target can not communicate with the source repository manager. This could happen if Nexus Repository 3 could not find a Nexus Repository 2 proxy server in place.

If you enabled Nexus Repository 2 HTTP or HTTPS settings during an upgrade, Nexus Repository 3 would use its original HTTP or HTTPS settings and ignore the settings in place for Nexus Repository 2.

Nexus Repository 3 generates a warning is in the log if this error occurrs. It will also log "Skipping HTTP(S) proxy settings, connection to Nexus 2 failed using new settings. Please manually configure the HTTP proxy settings after the upgrade is done." if something else fails.

What is Not Included in the Upgrade?

The following items are not included in the upgrade:

- Unsupported repository formats
  - Yum
    - RPMs stored in Yum-enabled Maven2 repositories on Nexus Repository 2 can only be moved to Maven2 repositories in Nexus Repository 3. Any Yum-related capabilities must be disabled on the Maven2 repository first. There is no automatic option to move RPMs in Nexus Repository 2 Maven2 repositories into Yum format repositories in Nexus Repository 3.
  - Maven1
  - p2
  - OBR
- Capabilities
- Custom branding
- Custom log levels or edits to `logback.xml` configuration files (e.g., custom log file rotation, file naming, log patterns)
- Environment variables affecting values used to control the repository manager
- Java VM settings, including custom system properties or variables
- Jetty server XML configuration files
- Manual edits to other files under the `nexus` installation directory, such as edits to `nexus/WEB-INF/classes/ehcache.xml`
- Operating system `nexus` service scripts
- Operating system optimization, such as increasing allowable open file handles
- Routing rules

- Scheduled tasks
- Staging repositories, settings, or configuration
- The *admin* user password from Nexus Repository 2
- Third-party or custom-developed plugins
- Virtual repositories

Prerequisites for Upgrading from Nexus Repository 2 to Nexus Repository 3

A successful upgrade from Nexus Repository 2 to Nexus Repository 3 requires planning, decision making, and, if possible, testing before attempting the actual upgrade.

Prerequisite	Required/ Recommended	Details
Nexus Repository 2 instance is on the latest Nexus Repository 2 version	Required	Your Nexus Repository 2 instance must be on the latest Nexus Repository 2 version before upgrading to the latest Nexus Repository 3 version. Upgrading to an older Nexus Repository 3 then subsequently updating to the latest Nexus Repository 3 is not equivalent, as it will hit old upgrade bugs.
Both Nexus Repository 2 and Nexus Repository 3 have the same license type (OSS or Pro)	Required	When upgrading from Nexus Repository 2 to 3, both versions must have the same licensing setup. The same license file can be used in Nexus Repository 2 and Nexus Repository 3 or a license can always be applied at a later time. If the Nexus Repository 3 instance is on the same server as the Nexus Repository 2 instance, both instances will automatically share the same license. Otherwise, you will need to install the license on the new Nexus Repository 3 instance.

Prerequisite	Required/ Recommended	Details
The Nexus Repository 3 instance must not yet be in use; it must be a fresh/clean instance	Required	The upgrade tool completely replaces existing configurations in Nexus Repository 3. It is intended for one-time use, and if you need to re-start an upgrade, you must first reset that upgrade(see page 35).
All files in the Nexus Repository work directory must be owned by the OS user, and there should be no zero length files in the work directory	Required	You can use the following Linux command to find files in the work directory that are not owned by the 'nexus' user as well as zero length files: `find . \! -user nexus -print`
Decide on a data transfer method(see page 21); ensure you have adequate storage space for the upgrade	Required	Full details available in Deciding on a Data Transfer Method(see page 21).
Ensure Nexus Repository 2 repository and repository group Repository IDs differ by more than just case. You must use distinguishable names.	Required	IDs of repositories and repository groups in Nexus Repository 2 that differ only by case will not be accepted during an upgrade to Nexus Repository 3 (example version 2 IDs: `myrepoid` vs. `Myrepoid`).
Ensure your Nexus Repository 3 instance is using an external PostgreSQL database **PRO**	Recommended	We highly recommend that you set up your Nexus Repository 3 instance with an external PostgreSQL database. See our[38]documentation on configuring Nexus Repository Pro for an external PostgreSQL database.
Review repository groups' contents in Nexus Repository Manager 2 to ensure all contents are selected for upgrade.	Recommended	Neglecting to select a single repository within a group may prevent the entire group from being upgraded.

38 https://help.sonatype.com/repomanager3/installation-and-upgrades/configuring-nexus-repository-pro-for-h2-or-postgresql#ConfiguringNexusRepositoryProforH2orPostgreSQL-ConfiguringforExternalPostgreSQL

Prerequisite	Required/ Recommended	Details
If necessary, configure your blob stores in Nexus Repository 3. (See our[39]Storage Guide)	Recommended	You will not be able to make configuration changes in Nexus Repository 3 again until the upgrade has successfully completed.
If you wish to use and move user tokens, navigate to the *User Token* tab in the *Administration* menu on the Nexus Repository 2 instance and check to ensure that user tokens are enabled.	Recommended	The upgrade tool only upgrades pre-existing user tokens to Nexus Repository 3 if user token support is enabled in Nexus Repository 2.
Consider optimizing upgrade performance(see page 23) by disabling unnecessary tasks and enabling some helfup ones.	Recommended	Full details available in the Optimizing Upgrade Performance(see page 23) topic.
Back up your infrastructure	Recommended	Establish a safe recovery point should something go wrong.
Test your upgrade with your desired upgrade settings using a similar test environment	Recommended	As a best practice, we recommend trying to create a test version of your current production environment so that you can test upgrading from Nexus Repository 2 to 3 there before performing it in a production environment. This can help you see what errors (such as out of memory errors in Nexus Repository 2) may occur with your selected test upgrade settings. Then, you can adjust your settings so that you do not encounter these errors when upgrading your production instance.

39 https://help.sonatype.com/display/NXRM3/Storage+Guide

Deciding on a Data Transfer Method

When performing the upgrade from Nexus Repository 2 to 3, there are three data transfer methods from which you must choose: HTTP downloading, file system copying, or file system hard linking. Each method has its own pros, cons, and requirements as detailed below.

Data Transfer Method	Description	Pros	Cons	Requirements
File System Hard Linking (Preferred)	Nexus Repository 2 tells Nexus Repository 3 the path of the file content to transfer. Nexus Repository 3 then creates a file system hard link to the same content. Data is not duplicated	• The fastest method • Saves storage space	Nexus Repository 2 and 3 must be installed on the same server	• Nexus Repository 2 and 3 must be configured to access the same storage system on identically named mount points. • The file system must support hard linking. • There must be adequate file handles for both Nexus Repository 2 and 3. • See the[40]System Requirements for Nexus Repository 3. • See[41]this article on how to determine configured open file handles for

40 https://help.sonatype.com/repomanager3/product-information/system-requirements#SystemRequirements-AdequateFileHandleLimits
41 https://support.sonatype.com/hc/en-us/articles/213465218-The-nexus-log-file-is-full-of-too-many-open-files-exceptions-how-can-I-fix-this-

Data Transfer Method	Description	Pros	Cons	Requirements
				Nexus Repository 2.
File System Copying	Nexus Repository 2 provides Nexus Repository 3 with the necessary path to file content, which Nexus Repository 3 then copies.	• Faster than HTTP downloading • Less impact on performance of Nexus Repository 2	Not as fast as hard linking	• Nexus Repository 2 and 3 must be configured to access the same storage system on identically named mount points. • You must ensure you have enough storage space; upwards of double the original storage space will be needed at least temporarily during data duplication.
HTTP Downloading	Nexus Repository 3 makes HTTP requests to Nexus Repository 2 to transfer configuration and data.	The only method that works for situations where Nexus Repository 2 and 3 are on different machines and do not share access to the same file system storage	The slowest option	• You should not synchronize Nexus Repository content to cloud services or on-premises data centers while this method is running. This tool is solely designed to allow for data and configuration

Data Transfer Method	Description	Pros	Cons	Requirements
				transfers between Nexus Repository 2 and 3. • You must ensure you have enough storage space; upwards of double the original storage space will be needed at least temporarily during data duplication.

Optimizing Upgrade Performance

To speed up the upgrade process, you may wish to make some changes and perform some cleanup in Nexus Repository 2. The suggestions below can help you reduce the amount of content you need to move, free up resources to speed up the upgrade, and avoid some issues that could arise regarding old repositories.

Feature	How to Optimize
[42]System Feeds	If your organization does not rely on system feeds (often used for team communication), consider[43]disabling them.
[44]Repair Index Tasks	These tasks support searching components within the user interface and do not need to be rebuilt that often. Consider disabling them across all repositories.

42 https://help.sonatype.com/repomanager2/using-the-user-interface/browsing-system-feeds
43 https://support.sonatype.com/hc/en-us/articles/213464998-How-to-disable-the-System-Feeds-nexus-timeline-plugin-feature-to-improve-Nexus-performance?_ga=2.132181390.14056720.1669046939-291822469.1668525051
44 https://support.sonatype.com/hc/en-us/articles/213465468-What-do-all-of-the-search-index-related-scheduled-tasks-do-and-when-should-I-schedule-them-?_ga=2.132181390.14056720.1669046939-291822469.1668525051

Feature	How to Optimize
	Disabling the[45]*Update Repositories Index*[46]task while using the Upgrade Wizard will enhance the wizard's performance and free up resources that Nexus Repository 2 will need.
Snapshot Removal Tasks	Enable both [47]*Remove Snapshots from Repository* and [48]*Remove Unused Snapshots From Repository*, which deletes old unneeded component versions. Running these task in Nexus Repository 2 before upgrade will reduce the amount of content to move, meaning a faster upgrade.
Deprecated Repositories	Remove any deprecated repositories. For example, any Maven 2 proxy repositories with the domain name [49]"codehaus.org" should be deleted. If Nexus Repository 2 contains proxy repositories that are pointing at sites that don't exist and are not manually blocked for remote contact even though remote contact cannot work, this can slow things down during upgrade as Nexus Repository 2 could get hung up trying to contact bad remotes. You can also elect not to include very old repositories when selecting repositories to upgrade in the Upgrade Wizard.
[50]Rebuild Maven Metadata Files Task	This scheduled task is disabled by default and should only be run if you need to repair a corrupted Maven repository storage on disk.
[51]Staging Rules	If you are a Nexus Repository Pro user that uses the application for staging releases, redefine or reduce the number of configured staging rules and staging profiles.

45 https://help.sonatype.com/repomanager2/configuration/managing-scheduled-tasks#ManagingScheduledTasks-UpdateRepositoriesIndex
46 https://help.sonatype.com/repomanager2/configuration/managing-scheduled-tasks#ManagingScheduledTasks-UpdateRepositoriesIndex
47 https://help.sonatype.com/repomanager2/configuration/managing-scheduled-tasks#ManagingScheduledTasks-RemoveSnapshotsfromRepository
48 https://help.sonatype.com/repomanager2/configuration/managing-scheduled-tasks#ManagingScheduledTasks-RemoveUnusedSnapshotsFromRepository
49 https://support.sonatype.com/hc/en-us/articles/217611787
50 https://help.sonatype.com/nxrm2master/configuration/managing-scheduled-tasks#ManagingScheduledTasks-RebuildMavenMetadataFiles
51 https://help.sonatype.com/nxrm2master/staging-releases/configuring-the-staging-suite

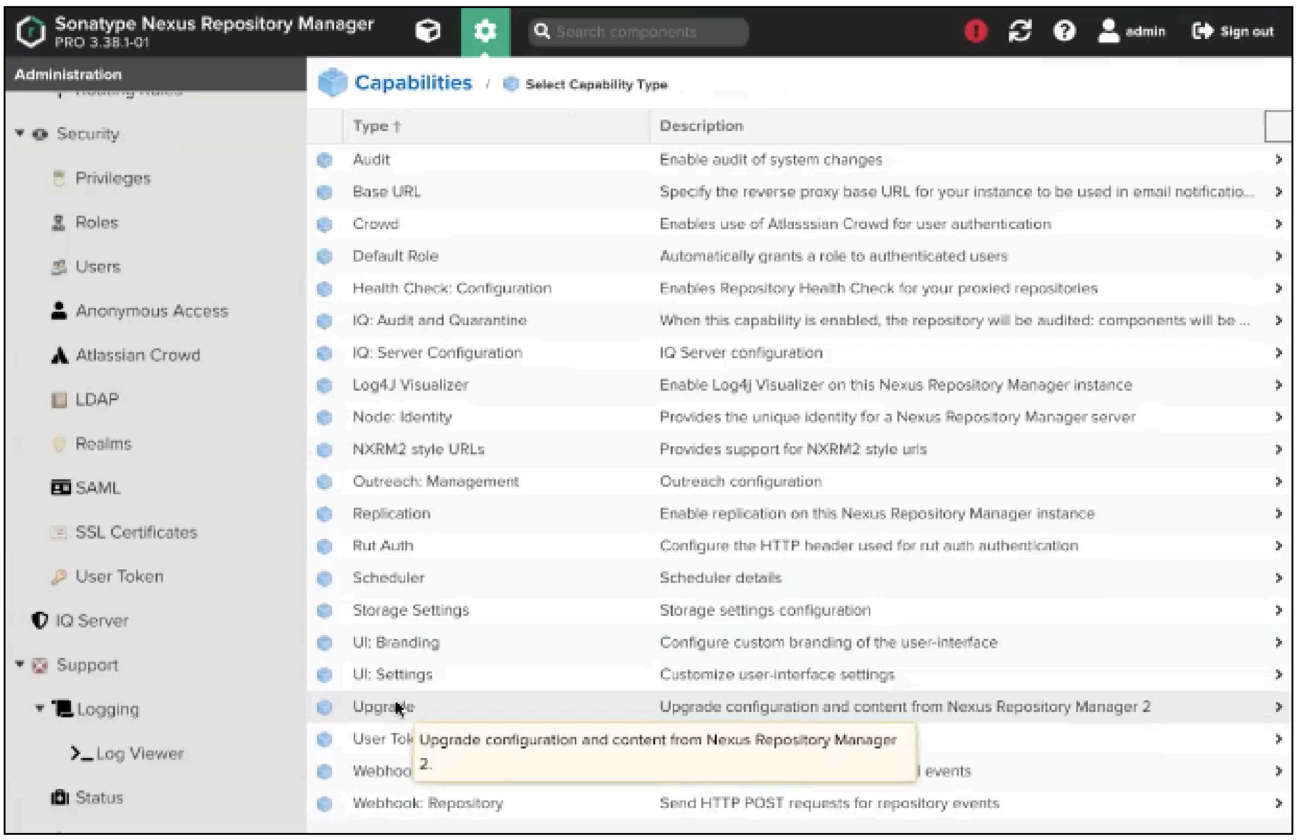
Feature	How to Optimize
Scheduled Tasks for Releases	If you find empty *Use Index* checkboxes under [52]*Task Settings*, use the opportunity to disable or remove those specific tasks for releases.

Configuring Upgrade Capabilities on Nexus Repository 2 and 3

Before you can use the Upgrade Wizard, you will need to configure upgrade capabilities on both the Nexus Repository 2 and 3 instances. The instructions below walk you through this process.

Create the Upgrade Capability on the Nexus Repository 3 Instance

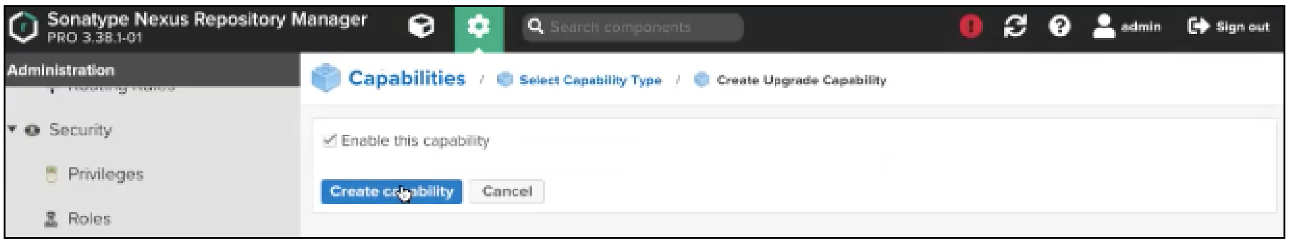
1. Under *Administration* → *System* → *Capabilities*, create an *Upgrade* capability on your Nexus Repository 3 instance.



Type ↑	Description
Audit	Enable audit of system changes
Base URL	Specify the reverse proxy base URL for your instance to be used in email notificatio...
Crowd	Enables use of Atlassian Crowd for user authentication
Default Role	Automatically grants a role to authenticated users
Health Check: Configuration	Enables Repository Health Check for your proxied repositories
IQ: Audit and Quarantine	When this capability is enabled, the repository will be audited: components will be ...
IQ: Server Configuration	IQ Server configuration
Log4J Visualizer	Enable Log4j Visualizer on this Nexus Repository Manager instance
Node: Identity	Provides the unique identity for a Nexus Repository Manager server
NXRM2 style URLs	Provides support for NXRM2 style urls
Outreach: Management	Outreach configuration
Replication	Enable replication on this Nexus Repository Manager instance
Rut Auth	Configure the HTTP header used for rut auth authentication
Scheduler	Scheduler details
Storage Settings	Storage settings configuration
UI: Branding	Configure custom branding of the user-interface
UI: Settings	Customize user-interface settings
Upgrade	Upgrade configuration and content from Nexus Repository Manager 2
User Tok	Upgrade configuration and content from Nexus Repository Manager 2.
Webhoo	events
Webhook: Repository	Send HTTP POST requests for repository events

2. After selecting the *Upgrade* capability type, ensure the *Enable this capability* checkbox is checked, and select Create capability.

52 https://help.sonatype.com/repomanager2/configuration/managing-scheduled-tasks#ManagingScheduledTasks-TaskSettings


Sonatype Nexus Repository Manager
PRO 3.38.1-01
Search components
admin
Sign out
Administration
Security
Privileges
Roles
Capabilities / Select Capability Type / Create Upgrade Capability

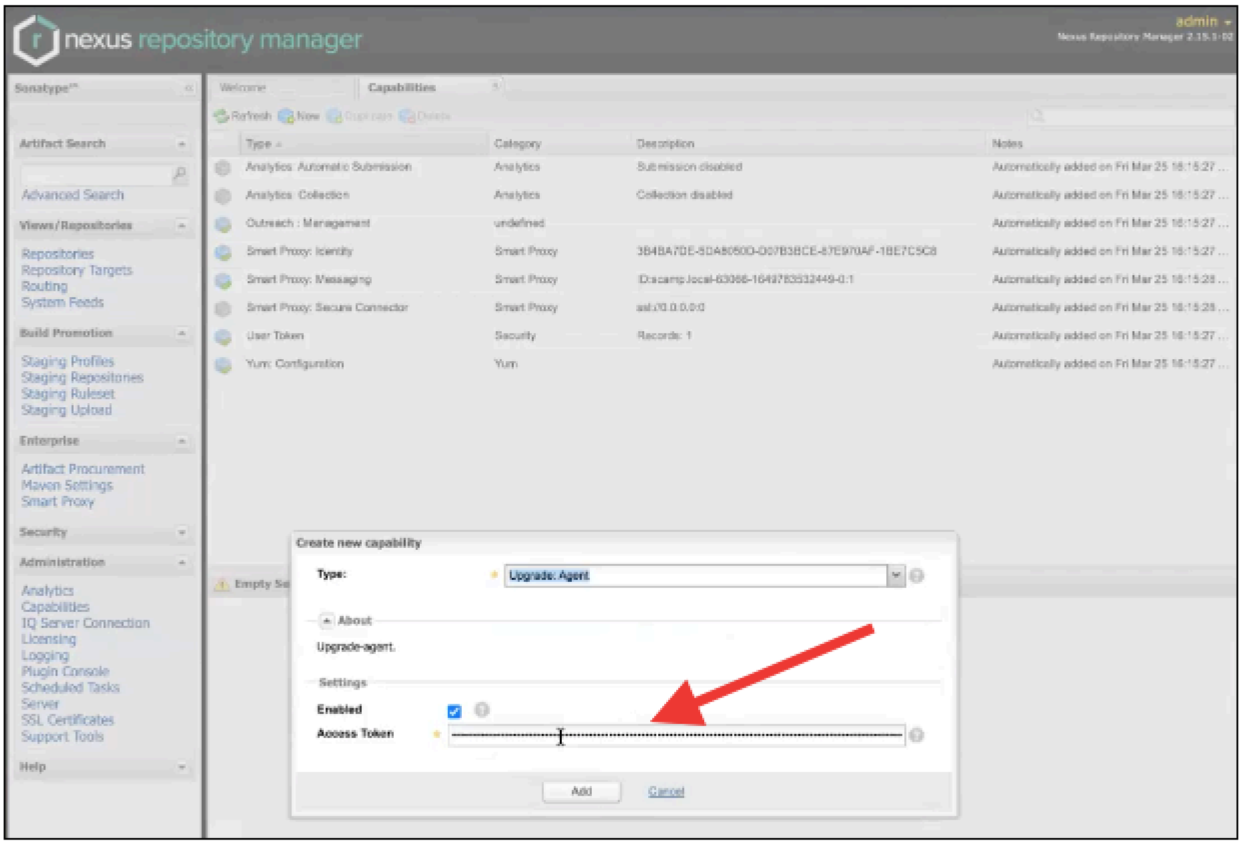
✓ Enable this capability

Create capability | Cancel

3. An *Upgrade* option now appears at the bottom of your main side menu.

Create the Upgrade Agent Capability on the Nexus Repository 2 Instance

1. In your Nexus Repository 2 instance, navigate to *Administration* → *Capabilities*; select the *New* button under the *Capabilities* tab to create a new capability (see the[53] Nexus Repository 2 documentation for more details on capabilities).



Type	Category	Description	Notes
Analytics: Automatic Submission	Analytics	Submission disabled	Automatically added on Fri Mar 25 16:15:27 ...
Analytics: Collection	Analytics	Collection disabled	Automatically added on Fri Mar 25 16:15:27 ...
Outreach : Management	undefined		Automatically added on Fri Mar 25 16:15:27 ...
Smart Proxy: Identity	Smart Proxy	3B4BA7DE-5DA8050D-D07B3BCE-87E970AF-1BE7C5C8	Automatically added on Fri Mar 25 16:15:27 ...
Smart Proxy: Messaging	Smart Proxy	D:acamp.local-63066-1649783532449-0:1	Automatically added on Fri Mar 25 16:15:28 ...
Smart Proxy: Secure Connector	Smart Proxy	ssl://0.0.0.0:0	Automatically added on Fri Mar 25 16:15:28 ...
User Token	Security	Records: 1	Automatically added on Fri Mar 25 16:15:27 ...
Yum: Configuration	Yum		Automatically added on Fri Mar 25 16:15:27 ...

Create new capability

Type: Upgrade: Agent

About

Upgrade-agent.

Settings

Enabled: ✓

Access Token:

Add | Cancel

2. Select *Upgrade Agent* as the capability *Type*.

3. Ensure the *Enabled* checkbox is selected.

4. You can either keep the generated *Access Token* or provide your own as shown in the highlighted box above. You will need this access token later when performing the upgrade.

53 https://help.sonatype.com/repomanager2/configuration/accessing-and-configuring-capabilities

5. Select *Add* to add the capability.

Step-by-Step Nexus Repository 2 to 3 Upgrade Wizard Instructions

As a best practice, we recommend trying to create a test version of your current production environment so that you can test upgrading from Nexus Repository 2 to 3 there before performing it in a production environment. This can help you see what errors (such as out of memory errors in Nexus Repository 2) may occur with your selected test upgrade settings. Then, you can adjust your settings so that you do not encounter these errors when upgrading your production instance.

The wizard is meant to be used once; do not try to transfer server configuration multiple times.

Follow our instructions for resetting your upgrade(see page 35) before attempting another upgrade.

Are You Ready to Upgrade?

Checkpoint: Are You Ready to Upgrade?

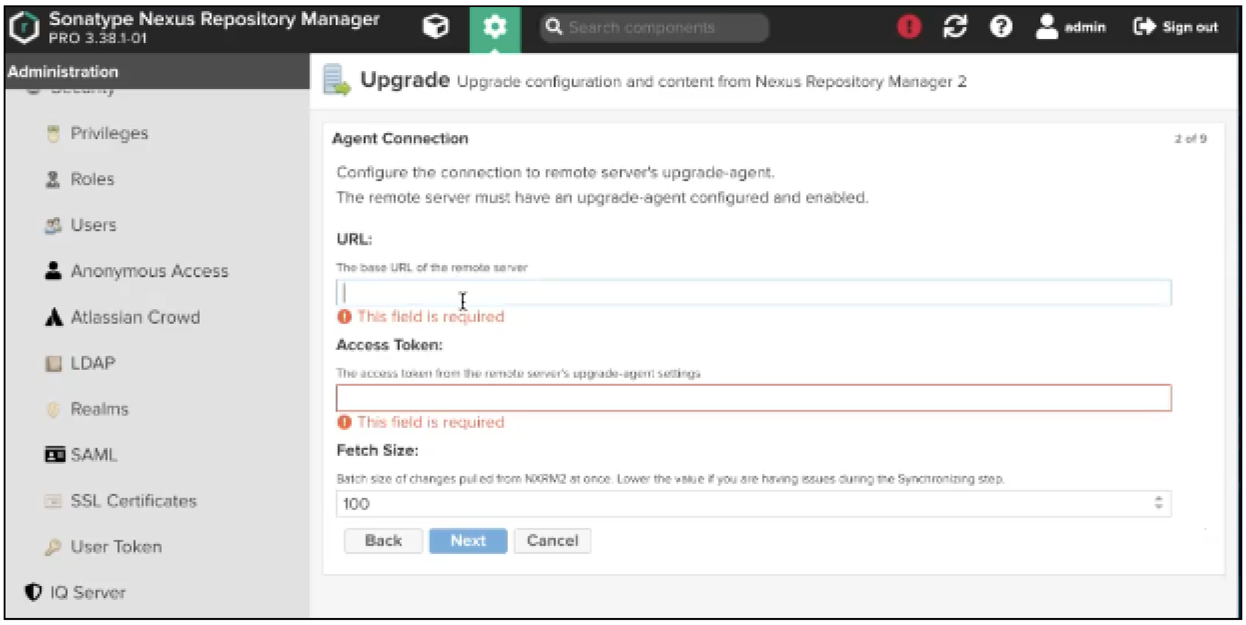
Before proceeding, have you done the following things?

- ☐ Have you configured the upgrade capabilities in Nexus Repository 2 and 3(see page 25)?
- ☐ Have you performed a test upgrade with your desired upgrade settings using a similar test environment to evaluate any potential issues? (Recommended)
- ☐ Have you ensured that nobody is using the Nexus Repository 3 instance? While this upgrade is happening, Nexus Repository 3 will be offline, and nobody should be using it.
- ☐ Have you upgraded your Nexus Repository 2 instance to the latest version of Nexus Repository 2? (Required)
- ☐ Do both your Nexus Repository 2 and 3 instances have the same licensing setup (both OSS or both Pro)? (Required)
- ☐ Do you need to configure your blob stores in Nexus Repository 3? (See our[54]Storage Guide)
- ☐ Have you evaluated and decided on which transfer method you wish to use and ensured you meet all storage requirements? (See Deciding on a Data Transfer Method(see page 21))

54 https://help.sonatype.com/display/NXRM3/Storage+Guide

Step 1 - Configure the Connection Between Nexus Repository 2 and Nexus Repository 3

1. In your Nexus Repository 3 instance, select the *Upgrade* option from the side menu. This launches the Upgrade wizard and takes you to the *Overview* screen, which contains important information. Be sure to read this information before proceeding. Then, select *Next* to navigate to the next screen in the wizard.



Upgrade Upgrade configuration and content from Nexus Repository Manager 2

Agent Connection 2 of 9

Configure the connection to remote server's upgrade-agent.
The remote server must have an upgrade-agent configured and enabled.

URL:
The base URL of the remote server
This field is required

Access Token:
The access token from the remote server's upgrade-agent settings
This field is required

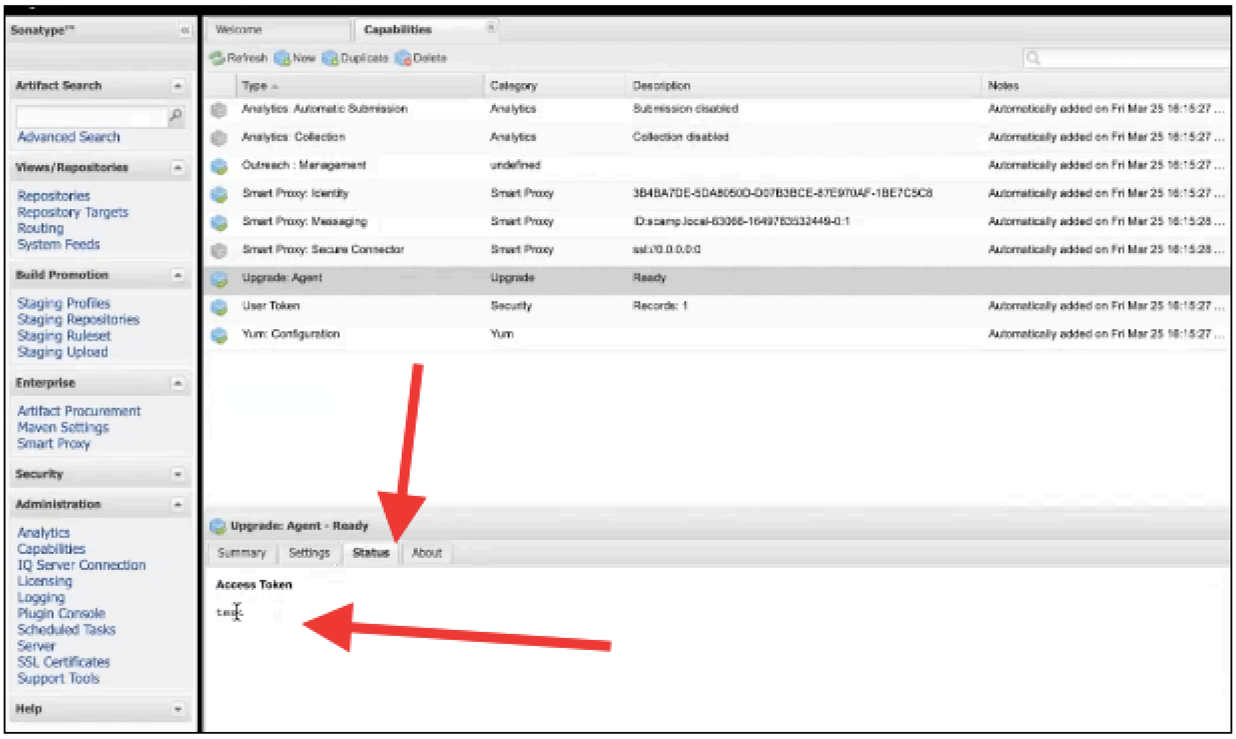
Fetch Size:
Batch size of changes pulled from NXRM2 at once. Lower the value if you are having issues during the Synchronizing step.
100

2. On the *Agent Connection* screen, complete the following information:

- *URL* - Enter the base URL for the Nexus Repository 2 instance (available in your[55]*Application Server Settings*).
  - By default, Nexus Repository 2 ships with a `/nexus` web context that must be included in the URL (e.g., `http://<hostname>:<port>/nexus`)

- *Access Token* - You will need to obtain this from the Nexus Repository 2 instance. This was created when configuring the *Upgrade Agent* capability in Nexus Repository 2(see page 26) and is available in your Nexus Repository 2 instance at *Administration → Capabilities → Upgrade Agent → Status* tab.

55 https://help.sonatype.com/repomanager2/configuration/customizing-server-configuration#CustomizingServerConfiguration-ApplicationServerSettings



Type	Category	Description	Notes
Analytics: Automatic Submission	Analytics	Submission disabled	Automatically added on Fri Mar 25 16:15:27 ...
Analytics: Collection	Analytics	Collection disabled	Automatically added on Fri Mar 25 16:15:27 ...
Outreach : Management	undefined		Automatically added on Fri Mar 25 16:15:27 ...
Smart Proxy: Identity	Smart Proxy	3B4BA7DE-5DA8050D-D07B3BCE-87E970AF-1BE7C5C8	Automatically added on Fri Mar 25 16:15:27 ...
Smart Proxy: Messaging	Smart Proxy	ID:scamp.local-63066-1649783532449-0:1	Automatically added on Fri Mar 25 16:15:28 ...
Smart Proxy: Secure Connector	Smart Proxy	ssl://0.0.0.0:0	Automatically added on Fri Mar 25 16:15:28 ...
Upgrade: Agent	Upgrade	Ready	
User Token	Security	Records: 1	Automatically added on Fri Mar 25 16:15:27 ...
Yum: Configuration	Yum		Automatically added on Fri Mar 25 16:15:27 ...

Access Token

[illegible]

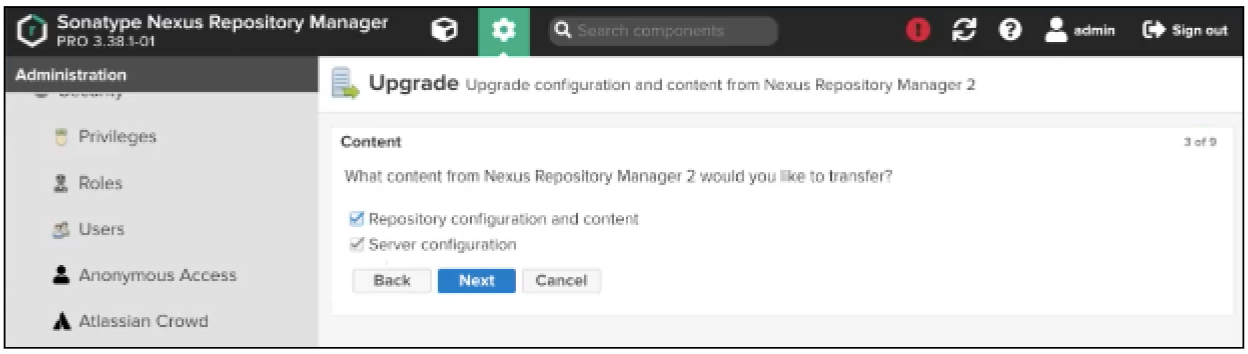
- *Fetch Size* - This dictates the number of components that Nexus Repository 3 requests from Nexus Repository 2 at one time. It is set to 100 by default.

Keeping your *Fetch Size* too high can cause out of memory errors with Nexus Repository 2 and is especially common for NPM and NuGet repositories, which have a lot of package metadata. Consider reducing your fetch size to a smaller value (e.g., 20).

3. Select *Next* to continue.

Step 2 - Decide What Content to Transfer

1. On the *Content* screen, select the appropriate checkboxes for what you wish to transfer from your Nexus Repository 2 instance: *Repository configuration and content* and/or *Server configuration.* (See Data Included, Excluded, and Changed During Nexus Repository 2 to 3 Upgrade Process(see page 14).)



Upgrade configuration and content from Nexus Repository Manager 2

Content

3 of 9

What content from Nexus Repository Manager 2 would you like to transfer?

☑ Repository configuration and content
☑ Server configuration

This tool is intended for single use on both repositories and server configuration.

Upgrading repositories in batches or more than once is not supported.

Do not try to transfer server configuration multiple times.

Follow our instructions for resetting your upgrade(see page 35) before attempting another upgrade.

2. Select *Next* to continue.

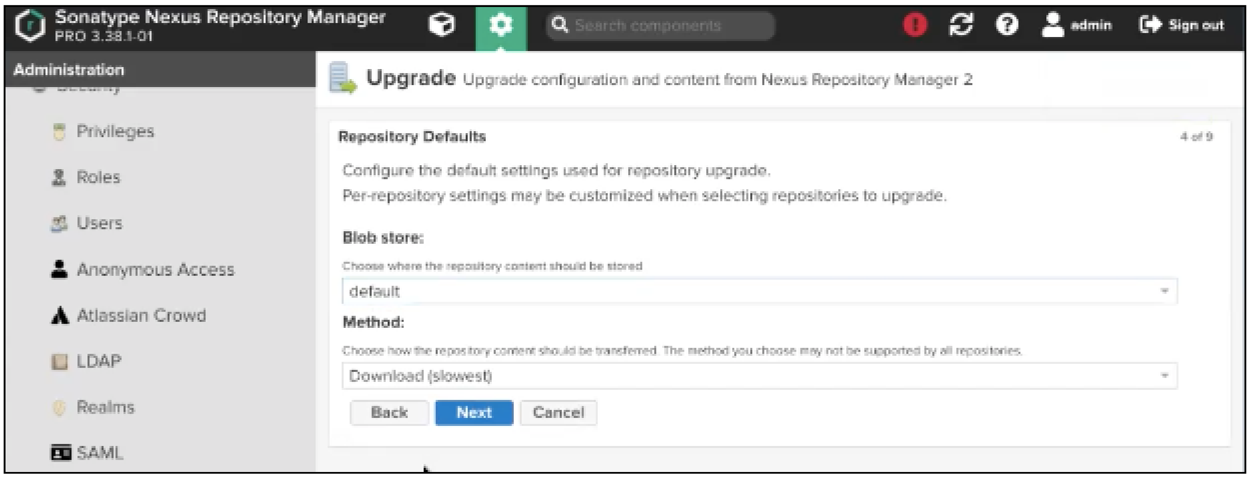
Step 3 - Configure and Select Repositories for Upgrade

1. On the *Repository Defaults* screen, configure the default settings to use for your upgrade:

- *Blob store* - Choose the blob store where the repository content should be stored.
- *Method* - Choose how the repository content should be transferred (See Deciding on a Data Transfer Method(see page 21)).
  - *Hard Link (fastest)* (Recommended)
  - *Filesystem copy (slow)*
  - *Download (slowest)*

If your Nexus Repository 2 and 3 instances are running on different machines and do not share access to the same file system storage, you must use the HTTP download method.

When running the HTTP download method, we discourage you from synchronizing Nexus Repository content to cloud services or on-premises data centers. This tool is solely designed to allow for data and configuration transfers between Nexus Repository 2 to 3.



Upgrade Upgrade configuration and content from Nexus Repository Manager 2

Repository Defaults 4 of 9

Configure the default settings used for repository upgrade.
Per-repository settings may be customized when selecting repositories to upgrade.

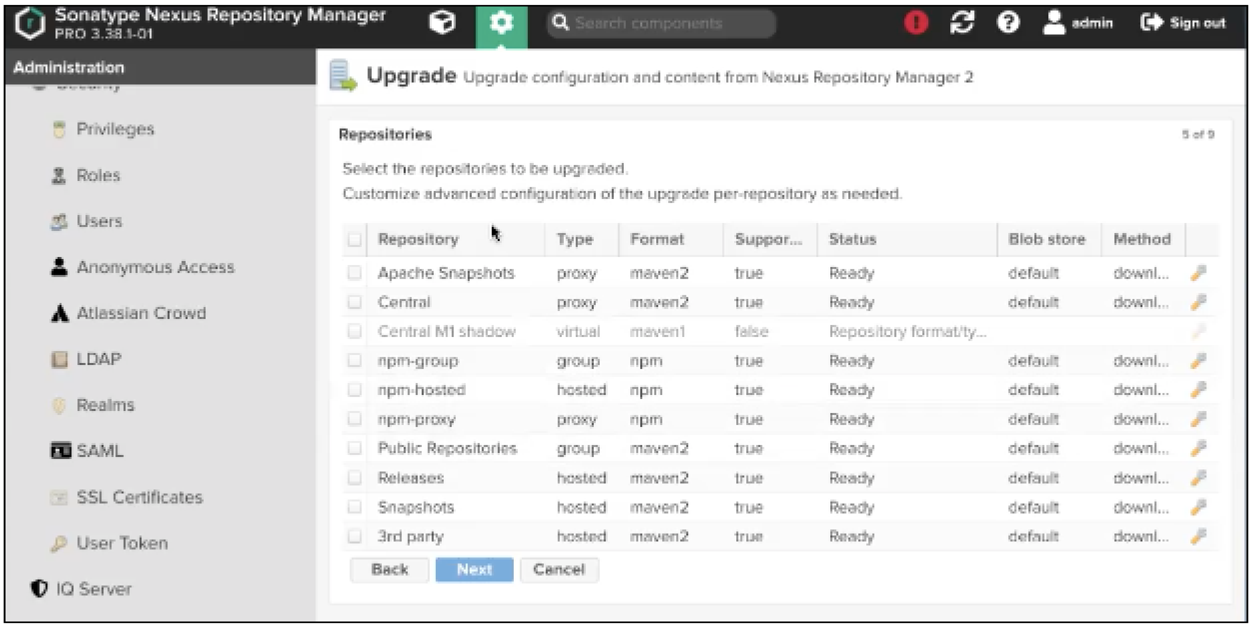
Blob store:
Choose where the repository content should be stored
default

Method:
Choose how the repository content should be transferred. The method you choose may not be supported by all repositories.
Download (slowest)

2. Select *Next* to continue to select the repositories you wish to upgrade.

3. On the *Repositories* screen, select the repositories you wish to upgrade by selecting the checkbox next to each repository; or, select the checkbox in the header row to select all repositories.

Repositories that you cannot upgrade to Nexus Repository 3 will appear greyed out and will not be selectable in the list.



Upgrade Upgrade configuration and content from Nexus Repository Manager 2

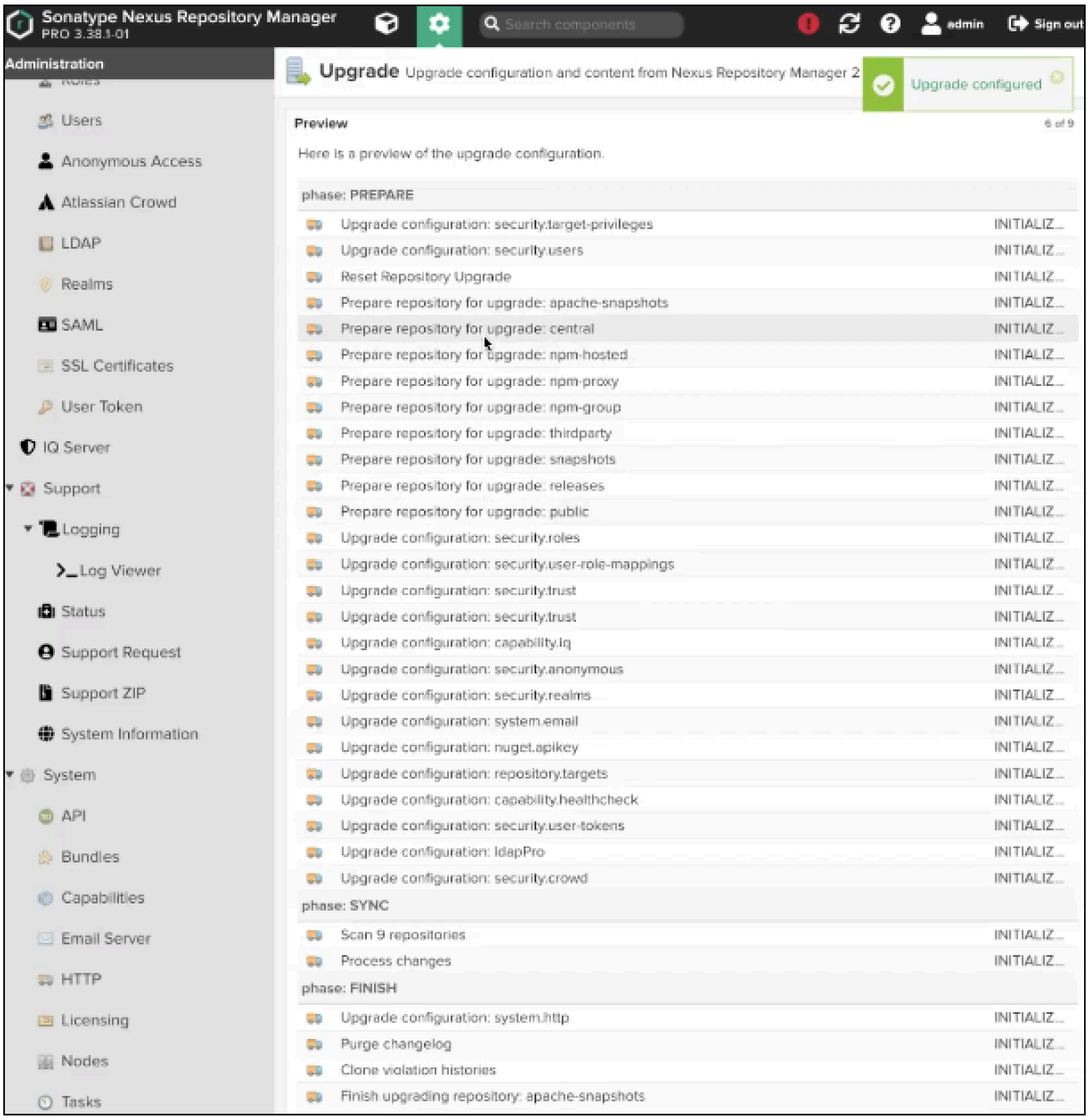
Repositories 5 of 9

Select the repositories to be upgraded.
Customize advanced configuration of the upgrade per-repository as needed.

Repository	Type	Format	Suppor...	Status	Blob store	Method
Apache Snapshots	proxy	maven2	true	Ready	default	downl...
Central	proxy	maven2	true	Ready	default	downl...
Central M1 shadow	virtual	maven1	false	Repository format/ty...		
npm-group	group	npm	true	Ready	default	downl...
npm-hosted	hosted	npm	true	Ready	default	downl...
npm-proxy	proxy	npm	true	Ready	default	downl...
Public Repositories	group	maven2	true	Ready	default	downl...
Releases	hosted	maven2	true	Ready	default	downl...
Snapshots	hosted	maven2	true	Ready	default	downl...
3rd party	hosted	maven2	true	Ready	default	downl...

4. You can also select the wrench icon in the last column to customize advanced configuration of the upgrade per-repository as needed (e.g., to select a different blob store).

5. Review the preview of the upgrade configuration. These are all the steps that will take place during the upgrade after you select the *Begin* button at the bottom of the screen.



Upgrade Upgrade configuration and content from Nexus Repository Manager 2

Preview 6 of 9

Here is a preview of the upgrade configuration.

phase: PREPARE	
Upgrade configuration: security.target-privileges	INITIALIZ...
Upgrade configuration: security.users	INITIALIZ...
Reset Repository Upgrade	INITIALIZ...
Prepare repository for upgrade: apache-snapshots	INITIALIZ...
Prepare repository for upgrade: central	INITIALIZ...
Prepare repository for upgrade: npm-hosted	INITIALIZ...
Prepare repository for upgrade: npm-proxy	INITIALIZ...
Prepare repository for upgrade: npm-group	INITIALIZ...
Prepare repository for upgrade: thirdparty	INITIALIZ...
Prepare repository for upgrade: snapshots	INITIALIZ...
Prepare repository for upgrade: releases	INITIALIZ...
Prepare repository for upgrade: public	INITIALIZ...
Upgrade configuration: security.roles	INITIALIZ...
Upgrade configuration: security.user-role-mappings	INITIALIZ...
Upgrade configuration: security.trust	INITIALIZ...
Upgrade configuration: security.trust	INITIALIZ...
Upgrade configuration: capability.iq	INITIALIZ...
Upgrade configuration: security.anonymous	INITIALIZ...
Upgrade configuration: security.realms	INITIALIZ...
Upgrade configuration: system.email	INITIALIZ...
Upgrade configuration: nuget.apikey	INITIALIZ...
Upgrade configuration: repository.targets	INITIALIZ...
Upgrade configuration: capability.healthcheck	INITIALIZ...
Upgrade configuration: security.user-tokens	INITIALIZ...
Upgrade configuration: ldapPro	INITIALIZ...
Upgrade configuration: security.crowd	INITIALIZ...
phase: SYNC	
Scan 9 repositories	INITIALIZ...
Process changes	INITIALIZ...
phase: FINISH	
Upgrade configuration: system.http	INITIALIZ...
Purge changelog	INITIALIZ...
Clone violation histories	INITIALIZ...
Finish upgrading repository: apache-snapshots	INITIALIZ...

Step 4 - Run the Upgrade

The upgrade is broken into three phases:

- *Prepare* - Prepares the transfer and creation of all configuration and components.
- *Sync* - Counts and processes all components set to upgrade and upgrades all other configurations.
- *Finish* - Performs the final clean up then closes the process.

1. To begin the upgrade from the *Preview* screen, select *Begin*; then, select *Yes* when asked if you wish to proceed. This will initiate the *Prepare* phase and take you to a progress screen where you can watch the progress of this phase.

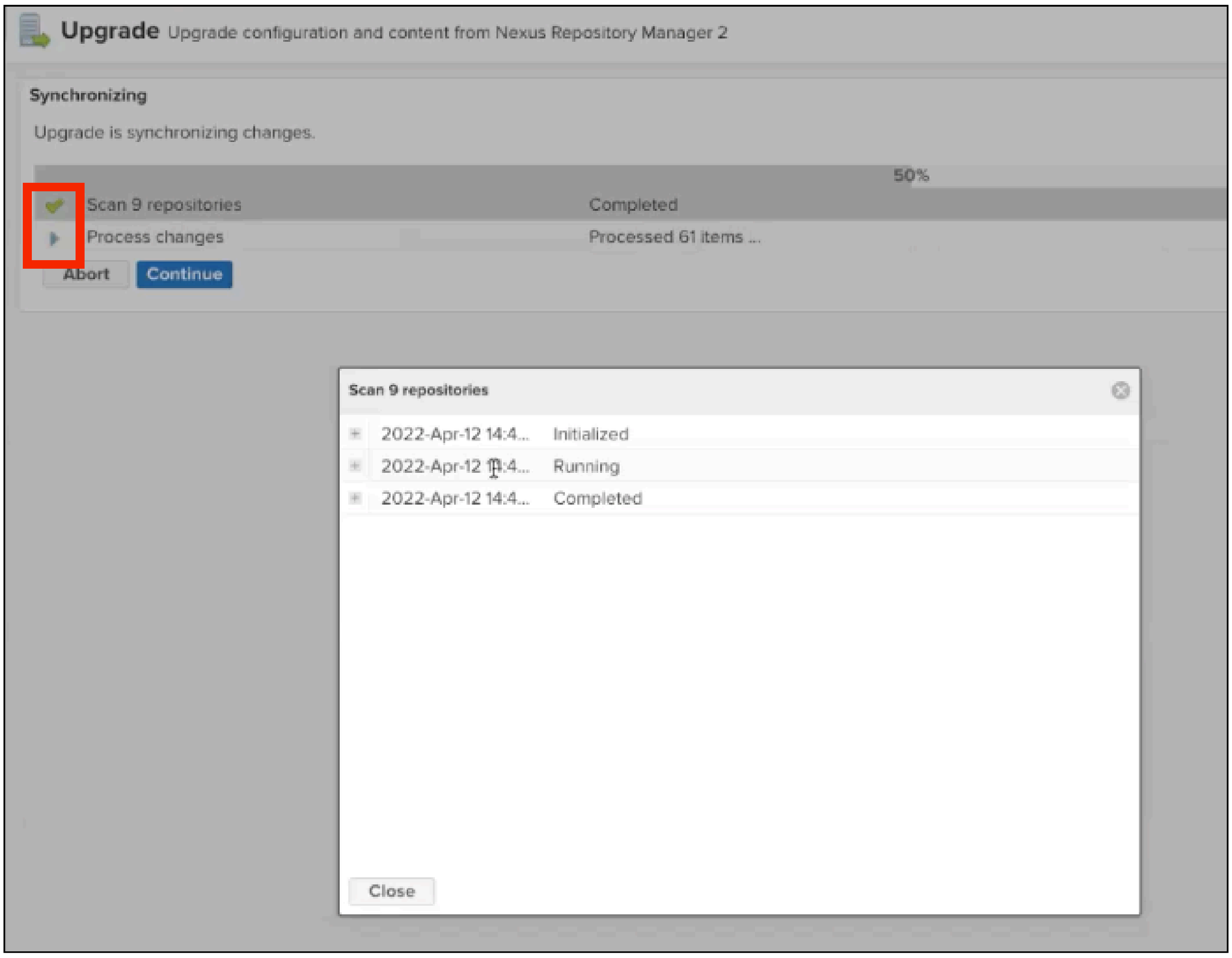
The invoked upgrade process destroys any existing configuration in the target Nexus Repository 3 server and replaces it with the configuration from the Nexus Repository 2 instance.

ⓘ From this point on, do not make further configuration changes in Nexus Repository 2. Any changes made will not be moved to Nexus Repository 3.

ⓘ Note that the progress percentage above the table represents the number of completed steps in the phase (e.g., it may read 54% when 14 of 26 steps are complete).

2. When all steps in the *Preparing* phase are complete, select *Continue* to move to the *Synchronizing* phase.

3. Monitor the *Synchronizing* phase screen to observe progress; you can select the arrows in the first column to see more details. Any content changes to the version 2 repositories continue to be upgraded during the *Synchronizing* step (e.g., new proxied components or new deployed components in version 2 are transferred to version 3).



Upgrade Upgrade configuration and content from Nexus Repository Manager 2

Synchronizing

Upgrade is synchronizing changes.

50%

✓	Scan 9 repositories	Completed
▸	Process changes	Processed 61 items ...

Scan 9 repositories

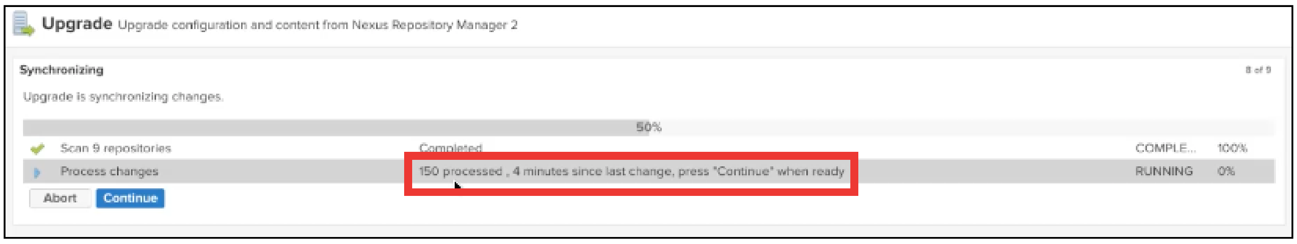
+	2022-Apr-12 14:4...	Initialized
+	2022-Apr-12 14:4...	Running
+	2022-Apr-12 14:4...	Completed

During the transfer process, you can already view and access your content in Nexus Repository 3 by using the component search or browsing in repositories or repository groups. However, the repositories will be offline until the process is fully complete.

Step 5 - Finalize the Upgrade

Nexus Repository 3 continues to ask Nexus Repository 2 for new content.

1. Once the *Process changes* step shows a sufficient amount of time "*since last change*," press *Continue* and then *Yes* when prompted to finalize the upgrade and move to the *Finishing* screen. Note that any future changes in Nexus Repository 2 will not be synchronized.



Upgrade Upgrade configuration and content from Nexus Repository Manager 2

Synchronizing 8 of 9

Upgrade is synchronizing changes.

50%

Scan 9 repositories	Completed	COMPLE...	100%
Process changes	150 processed , 4 minutes since last change, press "Continue" when ready	RUNNING	0%

2. Select *Done* on the *Finishing* screen to complete.

- Repository content is available immediately for direct download.
- [56]*Repair - Rebuild repository browse* and [57]*Repair - Rebuild repository search* tasks run automatically after an upgrade completes to build component Search UI and Search REST APIs as well as component Browse UI and HTML views. You do not need to manually schedule tasks to rebuild search indexes or browse nodes.
- Components will not be visible in the user interface (browse or search), HTML views of content, or Search REST API results until tasks named "Repo 2 Migration" finish. You can monitor this status in the [58]*Tasks* section or by examining the associated task log.

Licensing Note: If you upgraded your versions on the same server, no additional work need be done. If you upgraded your Nexus Repository 3 to a different server than Nexus Repository 2, then you must reinstall the license on the new server. You can use the same license file for Nexus Repository 3 as for Nexus Repository 2.

Aborting an Upgrade

If you select the *Abort* button during the upgrade process, the upgrade stops, and Nexus Repository 3 deletes all of the repositories it has created during this upgrade process.

Everything you've done up to this point is removed, and you must start over.

If you need to attempt this upgrade again, you should follow our instructions for resetting your upgrade(see page 35) and start from scratch.

Resetting a Nexus Repository 2 to 3 Upgrade

If you need to restart a Nexus Repository 2 to 3 upgrade for any reason, you will need to reset the upgrade and start it over from the beginning.

1. To end the upgrade process explicitly, use the *Abort* button in the upgrade wizard in Nexus Repository 3 if available.

56 https://help.sonatype.com/repomanager3/system-configuration/tasks#Tasks-Repair-Rebuildrepositorybrowse
57 https://help.sonatype.com/repomanager3/system-configuration/tasks#Tasks-Repair-Rebuildrepositorysearch
58 https://help.sonatype.com/display/NXRM3/Tasks

Complete all remaining steps regardless of whether or not you are able to use the *Abort* button.

2. Shut down Nexus Repository 3

3. Remove the entire Nexus Repository 3[59]data directory (i.e., $data-dir directory). If needed, see our[60]support article on how to determine your Nexus Repository 3's data directory location.

4. Shut down Nexus Repository 2.

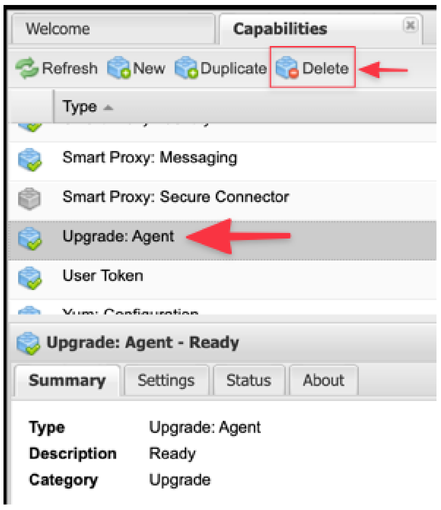
If your Nexus Repository 2 instance is a production instance, a brief production outage will be required.

5. Remove the entire Nexus Repository 2 directory at[61] $work-dir/db/migrationagent.

6. Restart Nexus Repository 2.

7. In Nexus Repository 2, go to *Administration* → *Capabilities*.

8. Select the *Upgrade: Agent* capability; then, select the *Delete* button to remove the capability.



Type Upgrade: Agent
Description Ready
Category Upgrade

This step will remove any internal RepositoryMigrationTask scheduled tasks that pertain to repository upgrade. These tasks will not be visible in the Scheduled Tasks view to an administrator, so this is the only way to remove these if they remain.

You can now start a new upgrade when ready by re-configuring the upgrade capabilities(see page 25).

59 https://help.sonatype.com/display/NXRM3/Directories
60 https://support.sonatype.com/hc/en-us/articles/218707647-How-to-Determine-the-Location-of-the-Nexus-3-Data-Directory
61 https://support.sonatype.com/hc/en-us/articles/213464308-Understanding-Nexus-Repository-Manager-2-Basic-Directories#WorkDir

Configuring Legacy URL Paths

Nexus Repository 2 uses a different URL pattern than Nexus Repository 3 to expose repositories and repository groups. Nexus Repository 3 creates an *NXRM2 style URLs* capability during upgrade so that it automatically supports the old patterns and your automated tools and CI continue to work. This allows you to access the example of `http://localhost:8081/nexus/repository/sample` by using `http://localhost:8081/nexus/content/repositories/sample`.

This example assumes your hostname (`localhost`), port (`8081`) and context path (`nexus`) match between your Nexus Repository 2 and Nexus Repository 3 installations. If not, you must use the ones from version 3 or reconfigure as stated in [62]Changing the Context Path section.

The *NXRM2 style URLs* capability has no effect on REST API calls, which were totally rewritten in Nexus Repository 3. You will need to reevaluate REST API calls and rewrite them against the Nexus Repository 3 [63]REST API.

Any automated tooling that uses direct repository browsing will need to be reconfigured for Nexus Repository 3 endpoints. See [64]Nexus Repository 3 HTML View documentation for more information on how to obtain this endpoint.

Manual Upgrade Methods

When either upgrading to a new Nexus Repository 3 instance or consolidating to an existing instance, you can manually upgrade repository content into Nexus Repository 3 Pro using any of the following methods:

- Using the Repository Export/Import features
- Including external proxies in a group repository
- Running scripts using the REST API

Manual upgrade methods have a few considerations:

- Expect to spend a fair amount of time in planning and executing any upgrade
- You cannot import or export repository configuration
- Some data duplication is required

62 https://help.sonatype.com/repomanager3/installation-and-upgrades/configuring-the-runtime-environment#ConfiguringtheRuntimeEnvironment-ChangingtheContextPath

63 https://help.sonatype.com/display/NXRM3/REST+and+Integration+API

64 https://help.sonatype.com/display/NXRM3/Browsing+Repositories+and+Repository+Groups#BrowsingRepositoriesandRepositoryGroups-HTMLView

Prerequisites

Before proceeding with any upgrade, we recommend taking the following steps:

Prerequisite	Required/ Recommended	Details
Access to Repository Storage	Required	• You must export content to a folder that the Nexus Repository 3 instance can access. • You can only import content from a folder that the Nexus Repository 3 instance can access.
Ensure your Nexus Repository 3 instance is using an external PostgreSQL database **PRO**	Recommended	We highly recommend that you set up your Nexus Repository 3 instance with an external PostgreSQL database. See our [65]documentation on configuring Nexus Repository Pro for an external PostgreSQL database or on[66]migrating an existing Nexus Repository 3 instance to a PostgreSQL database.
Back Up Infrastructure	Recommended	
Do a Test Run	Recommended	• Do a test run of the import process to identify issues and approximately how long the process could take.
Use Import Hard Links	Recommended	• Hard linking is far faster and does not duplicate the data on disk. • Review the[67]Import Hard Links documentation for details.
Avoid Importing Directly to S3	Recommended	• Use File-based blobstores when possible. • Importing directly to an S3 blobstore can be 3x slower. • Use the[68]move blobstore functionality to migrate to S3.

65 https://help.sonatype.com/repomanager3/installation-and-upgrades/configuring-nexus-repository-pro-for-h2-or-postgresql#ConfiguringNexusRepositoryProforH2orPostgreSQL-ConfiguringforExternalPostgreSQL
66 https://help.sonatype.com/repomanager3/installation-and-upgrades/migrating-to-a-new-database
67 https://help.sonatype.com/repomanager3/nexus-repository-administration/tasks/repository-import
68 https://help.sonatype.com/repomanager3/nexus-repository-administration/repository-management/configuring-blob-stores#ConfiguringBlobStores-MovingaBlobStore

Prerequisite	Required/Recommended	Details
Firewall Repositories	Optional	• You cannot migrate Firewall waivers. • Prefetch the approved components through the proxy before enabling the Firewall quarantine. • Export the Firewall report to obtain a list of approved components.

Export/Import Manual Upgrade Method PRO

Using[69]Repository Export and[70]Repository Import (both Pro-only) is the easiest and fastest manual method to copy a repository's contents from one Nexus Repository 3 instance to another.

Considerations

- You can directly import Nexus Repository 2 repository content from where it is stored on disk.
- The import task will maintain the created date, last updated date, and the last downloaded date attributes from Nexus Repository 2.
- In cases where the content in the source directory changes or the process is interrupted, you can use these tasks to only add the missing/additional content.
- The tasks must be manually created; you cannot use scripts or the API.
- You will need to create and run individual tasks for each repository.
- Multiple tasks will not run at the same time.

Steps for Nexus Repository 2 to 3 Export/Import Upgrade

1. Ensure that your Nexus Repository 3 instance has file system access to the Nexus Repository 2 storage; consider using a backup if Nexus Repository 2 is located on a different server.
2. On your Nexus Repository 3 instance, create the target repository to which the content will be imported.

69 https://help.sonatype.com/repomanager3/nexus-repository-administration/tasks/repository-export
70 https://help.sonatype.com/repomanager3/nexus-repository-administration/tasks/repository-import

As a best practice, ensure that the blobstore for the target repository is on the same storage as the source repository; this will allow you to use hard linking (recommended).

3. Under *Administration* → *System* → *Tasks* on the Nexus Repository 3 instance, create a new[71] *Repository - Import external files* task.
    a. Set the source directory to the folder where the Nexus Repository 2 repository is stored.
    b. Optionally, select *Enable Hard Links* (Recommended).
4. Run the task.
    a. For testing, you may stop the task before it completes.
    b. Review the[72] importing documentation on how to[73] reset the import proces.
5. When finished use the[74] Move Blobstore steps to use a different storage disk or S3 object store

If you have a lot of data to copy into the repository, this upgrade may take some time. Consider using the Proxying Repositories(see page 40) method below to make the source content available while waiting for the import tasks to complete.

Proxying Repositories Manual Upgrade Method

One common use case for proxy repositories is to use them to make content immediately available on a new Nexus Repository instance while slowly moving content as it is requested.

Considerations

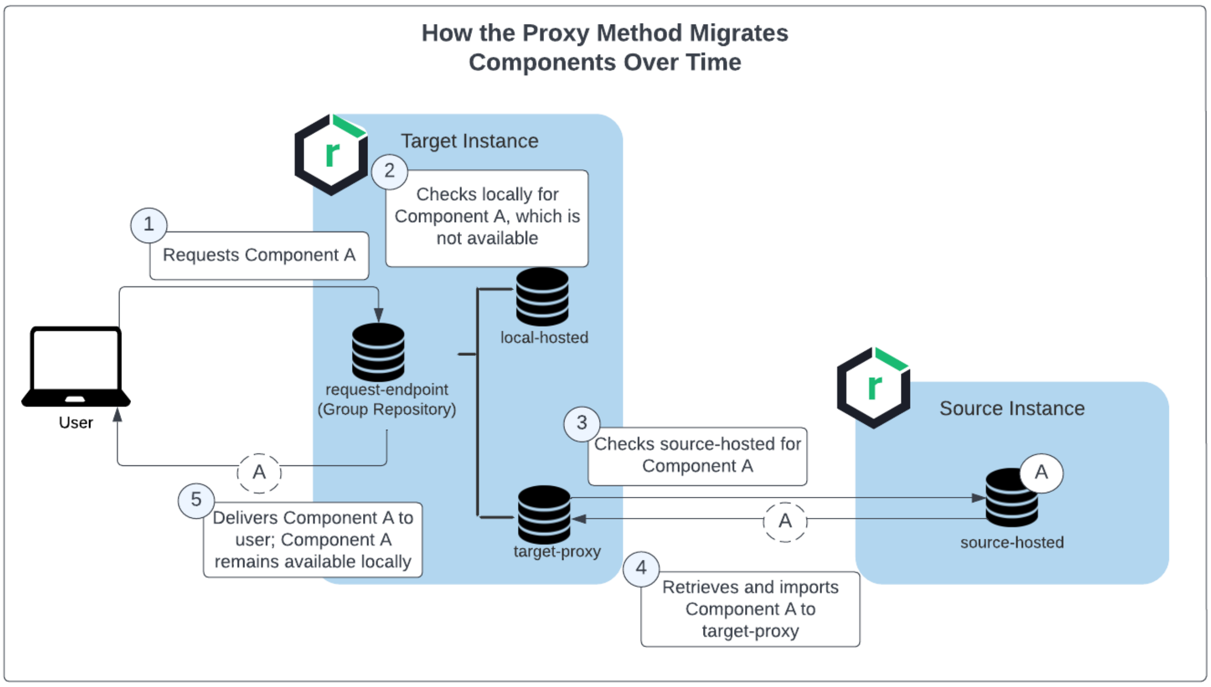
- This upgrade will only copy content that is actively being requested by development and production. This may be an effective way to avoid moving unused content.
- Nexus Repository fetches imported components through the proxy over HTTP(S), which can be very slow.
- You will need to maintain both repository instances until your new instance contains all needed content.
- You cannot convert a proxy repository to a hosted repository when the process is complete.

71 https://help.sonatype.com/repomanager3/nexus-repository-administration/tasks/repository-import
72 https://help.sonatype.com/display/NXRM3/Repository+Import
73 https://help.sonatype.com/repomanager3/nexus-repository-administration/tasks/repository-import#RepositoryImport-ResettingData(.dat)Files
74 https://help.sonatype.com/display/NXRM3/Configuring+Blob+Stores#ConfiguringBlobStores-MovingaBlobStore

Steps for the Proxying Repositories Manual Upgrade Method

This method involves creating a proxy repository (called `target-proxy` in the example below) on the new Nexus Repository 3 instance and having that proxy repository use the URL of hosted repository on a source instance (called `source-hosted` in the example below).

When a component is requested from but not found on the target repository group (called `request-endpoint` in the example below), Nexus Repository downloads the component from the hosted repository on the source instance through the proxy repository. This is demonstrated in the diagram below:


How the Proxy Method Migrates Components Over Time
Target Instance
1 Requests Component A
2 Checks locally for Component A, which is not available
local-hosted
request-endpoint (Group Repository)
User
3 Checks source-hosted for Component A
Source Instance
A
source-hosted
target-proxy
5 Delivers Component A to user; Component A remains available locally
4 Retrieves and imports Component A to target-proxy

This configuration can be used until the upgrade is complete.

Upgrading Staging

This topic provides Nexus Repository administrators valuable information about the change in staging implementation between Nexus Repository 2 and 3 as well as guidance on migrating your data and processes. For detailed information on using the staging feature in Nexus Repository 3, see the[75]Staging topic.

75 https://help.sonatype.com/repomanager3/nexus-repository-administration/staging

Staging in Nexus Repository 2 vs. Nexus Repository 3

In Nexus Repository 2, staging is done through the staging suite; in Nexus Repository 3, staging is done through tagging and REST API calls. This difference means that upgrading from Nexus Repository 2 to 3 lets you go from a dynamic repo-based model to a tag-based model.

When designing staging in Nexus Repository 3, the Sonatype Nexus Repository team had a chance to look for areas of improvement in Nexus Repository 2 and then develop the new staging feature to mitigate these issues.

Some of the issues identified in Nexus Repository 2 staging were as follows:

- It only works for the Maven2 format.
- Deployment/approval workflows are increasingly external to Nexus Repository.
- It doesn't scale. Failing to apply proper housekeeping controls means there can be 1000's of repositories, leading to significant performance problems in Nexus Repository 2.
- It's easy to lose track of build identity after promotion (the staging repo goes away).

The Nexus Repository team worked to resolve these issues and develop features in Nexus Repository 3 that let you do the following:

- Administer access control to in-house components at different phases of the SDLC. Components can be restricted until fully tested.
- Store components in a single hosted repository for each stage, while identifying individual builds and managing retention periods by stage.
- Move components between release stages using the REST API.
- Associate custom metadata to builds as components are uploaded to the repository.
- Manage workflows with external CI/CD tools.

A side by side comparison:

Staging in Nexus Repository 2	Staging in Nexus Repository 3
Workflow defined in Nexus Repository 2	Workflow defined externally (i.e., Jenkins)
Each build is a separate repository	Each build is a tag
Hundreds of repositories accumulate	Small, fixed number of repositories
Builds are absorbed into release repo	Build metadata outlives promotion
UI-driven	Powered by REST, no UI

Staging in Nexus Repository 2	Staging in Nexus Repository 3
Maven2 only	Maven2, raw, Docker, npm, Nuget, YUM
Rules defined in Nexus Repository 2	Rules invoked externally by CI/CD
“Staging” is a feature	Toolkit APIs used to build staging

Due to the extent to which we redesigned staging in Nexus Repository 3, it is not compatible with Nexus Repository 2 staging. Nexus Repository 3 staging provides a set of flexible building block capabilities that can be combined and arranged as needed to accommodate your organization’s software build and release pipeline. The set of REST endpoints that expose these capabilities allow for easy integration into CI/CD systems, and are easily customized to the workflows or client tooling and technologies needed to interact with them.

Benefits of Tagging and Staging in Nexus Repository 3

There are several advantages to upgrading to Nexus Repository 3:

Flexible Workflow

Staging is a set of powerful REST endpoints that are highly customizable to your environment. This new implementation makes build deployment configuration more flexible for teams as they expand beyond CI server usage to the staging repository and the ability to dynamically select builds for promotion.

Staging for Nexus Repository 3 also helps empower teams to take control of the build deployment process using advanced metadata capabilities. This includes deciding how the metadata changes along the release workflow, as well as how to isolate the right builds for the right teams.

Tags

Staging also works closely together with our tagging feature. In Nexus Repository Pro, staging works by moving components across hosted repositories. For instance, you could have hosted repositories for development, UAT, and production. You can then promote software components matching your organization’s software development lifecycle phases by moving components between these hosted repositories. This is where the tagging feature comes in. Tagging in Nexus Repository 3 is more powerful because you can easily group multiple builds and promote them as if they were a single unit.

For example, a complex project like Nexus Repository has thousands of files and all sorts of components that go into a build. The Nexus Repository teams needs a way to take all of these components that span multiple different coordinates and group them together in a way that lets us know it’s a single build. You can use the tagging REST endpoints to create and delete tags, list all tags, and attach metadata to tags. This is the Nexus Repository 3 equivalent to custom metadata.

As you can see from the example above, using tags provides a way to group and move your artifacts. Nexus Repository 3 provides several means for tagging, which usually happens on upload, but can also be done after upload as well. You can tag components with your automated CI/CD processes using the Jenkins Platform plugin. Alternatively, you can also tag components during upload within the UI.

Path from Nexus Repository 2 to 3

Now that you understand how staging in Nexus Repository 3 works, the next step is redefining your process from Nexus Repository 2 to the new instance. Upgrading to Nexus Repository 3 will give you a staging feature with a set of extremely configurable building blocks that can be adapted to your organization's needs.

Although there is no direct upgrade path from Nexus Repository 2 to 3, this section will go over some steps you can take to get started.

A basic setup workflow for staging in Nexus Repository 3 might look something like this:

1. Ensure that your upgrade from Nexus Repository 2 to Nexus Repository 3 is complete.
2. Identify the staging workflow from your Nexus Repository 2 configuration.
3. Create matching hosted repos in Nexus Repository 3.
4. Incorporate the stages of your agile software development lifecycle into the Nexus Platform Plugin for Staging.
5. Add a tagging call to the build.
6. Update the Nexus Repository 2 Maven calls to the Nexus Repository 3 REST calls.
7. Associate custom metadata to tags. Custom information is an extension of tagging. See[76]Viewing Tags for more info.
8. Use the Move REST API endpoint to promote tagged components among the hosted repos.
9. Use the Delete REST API endpoint to delete tagged components among the hosted repos.
10. Run a cleanup task to delete tags you no longer need.

The above steps represent a basic scenario. Our REST APIs give you a lot of flexibility to make this more robust and well integrated into your CI/CD pipeline.

For more information on our staging workflow, please see the staging and tagging topics in our help documentation.

76 https://help.sonatype.com/repomanager3/user-interface/viewing-tags

Best Practices and Troubleshooting

Although migration from Nexus Repository 2 staging to Nexus Repository 3 staging is not supported, there are a few things you can do to make the move easier. Here are a few ideas to help you make the switch:

- Keep in mind that this is an opportunity to start fresh with a new set of features that are compatible with modern CI/CD pipelines.
- Review what worked in Nexus Repository 2 staging and what didn't. This can help you determine what needs to be implemented similarly and where there are areas for improvement.
  - For example, you may need to keep your Nexus Repository 2 staging repos for data retention purposes. These types of requirements should be taken into account.
- When you upgrade, any "in flight" staging repositories in Nexus Repository 2 are not migrated. This means upgraded components in a build promotion repo or staging repo are not moved. This is an important setup step that should be thought about prior to upgrade.
- Get familiar with concepts and terminology, such as the following:
  - Tag names are a string (or label) that are the primary key for what gets "staged."
  - Metadata is extra information.
  - You cannot stage or create permissions based on metadata.
- Invest your time in establishing a strategy for naming tags and your management approach for tags. For example, think about things like the following:
  - Will tags be used? (This may depend on your organization's approach to naming builds and repos.)
  - How will tags be routinely aged off?
  - What is the best way to implement tag creation and assignment to your artifacts?
- Keep in mind the benefits of staging in Nexus Repository 3:
  - You can control who can see in-house components at different parts of the SDLC, so they aren't used before they're ready.
  - You can batch components together and easily identify builds.
  - Stagings REST APIs are powerful and flexible.
  - You can control the workflow from your CI/CD pipeline.
  - The "toolkit" style implementation means it supports other use cases.
  - Maven and Jenkins integration (e.g., automatically tested, manually tested, signed, approved, etc.).

Optimization, Performance, and Tuning

When considering upgrade time and speed, take into account all enabled features on your Nexus Repository Manager 2 instance that you may not need. You can optimize the performance of your upgrade by disabling specific features. As discussed in this article about [77]performance and tuning, identify and then reduce your list of used features to improve the performance of your repository manager. See some highlights, below:

System feeds

If your organization does not rely on system feeds, often used for team communication, learn how to disable them.

Repair index tasks

These tasks support searching components within the user interface, and do not need to be rebuilt that often, consider disabling them across all repositories.

Snapshot removal tasks

Enable both *Remove Snapshots from Repository* and *Remove Unused Snapshots From Repository*, which deletes old component versions no longer needed.

Repositories no longer supported

Remove any deprecated repositories. For example, any Maven 2 proxy repositories with the domain name [78]"codehaus.org" should be deleted.

Rebuild Maven Metadata Files

This scheduled task should only be run if you need to repair a corrupted Maven repository storage on disk.

Staging rules

If you are a Nexus Repository Manager Pro user that uses the application for staging releases, redefine or reduce the number of configured staging rules and staging profiles.

Scheduled tasks for releases

If you find empty *Use Index* checkboxes under *Task Settings*, use the opportunity to disable or remove those specific tasks for releases.

To help you decide how to reduce scheduled tasks, improving the performance of your upgrade, review the [79]knowledge base article.

77 https://support.sonatype.com/hc/en-us/articles/213465138
78 https://support.sonatype.com/hc/en-us/articles/217611787
79 https://support.sonatype.com/hc/en-us/articles/213465208