

Instructor: Mehdi Abdollahei

Nexus Repository Manager

Implementing and maintaining a repository manager for Linux images, Docker and Kubernetes, and developers.



 [Mehdi.DevOps.Pro](#)

 [mehdi-abdollahei](#)

 [mehdi_devops_pro](#)

 [mehdi_devops_pro](#)

The content of this course

Introduce Nexus:

- What is Nexus Repository Manager
- Why We Need Nexus
- Nexus Architecture
- Nexus Key Feature

Implementing Nexus:

- Installation on (Standalone, Docker, Kubernetes)
- Systemd Configuration
- Open the Nexus browser on the web
- Add Redhat Base and Debian Base Repo
- Test Repository using ubuntu and Rockylinux

Nexus As a Docker Image Repository

- Create Docker repository As a DockerHub
- Create Docker Repository for own Image
- Hands-on

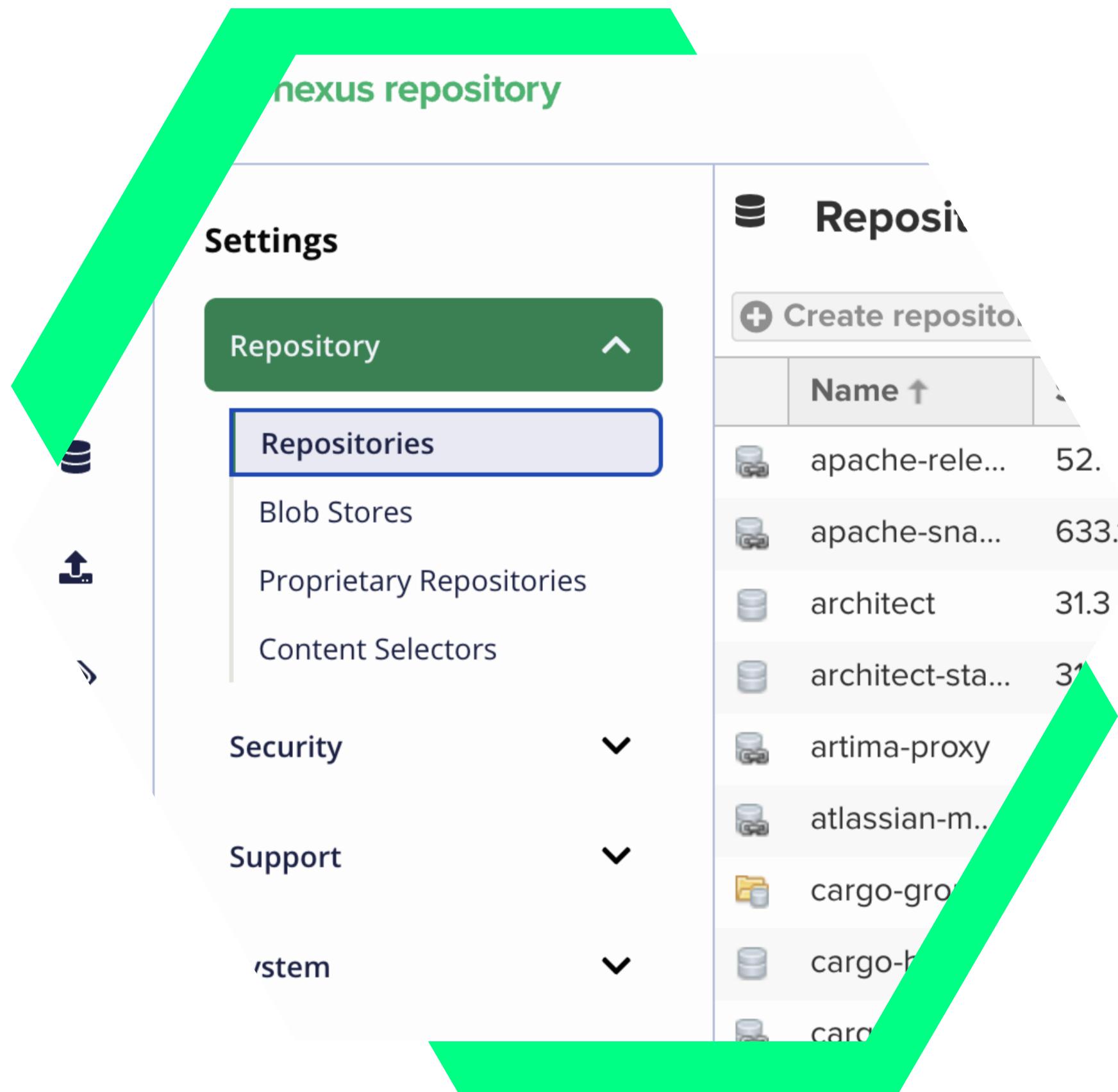
Nexus For Developer

- Add Repository of developer
- Hands-on

High Availability

- Backup
- HA





What is Nexus?

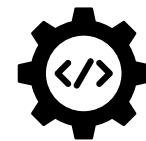
Nexus Repository is a **repository manager** that stores, organizes, and distributes **software components**, **binaries**, and **packages**. It's commonly used in **DevOps** and **CI/CD pipelines** to manage **dependencies** and **artifacts** across different stages of software development

Key Features

It's commonly used in **DevOps** and **CI/CD pipelines** to **manage dependencies** and **artifacts** across different stages of software development. There are **two main versions**:

Nexus Repository OSS (Open Source) – Free version.

Nexus Repository Pro – Paid version with additional features like advanced security, staging, and support for more repository formats.



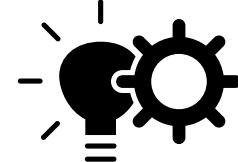
Integration with CI/CD tools

Works with Jenkins, GitLab CI, GitHub Actions, etc.



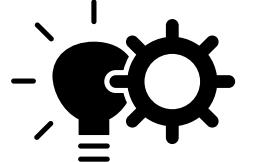
Centralized storage

Keeps all your build artifacts (like JARs, Docker images, npm packages) in one place.



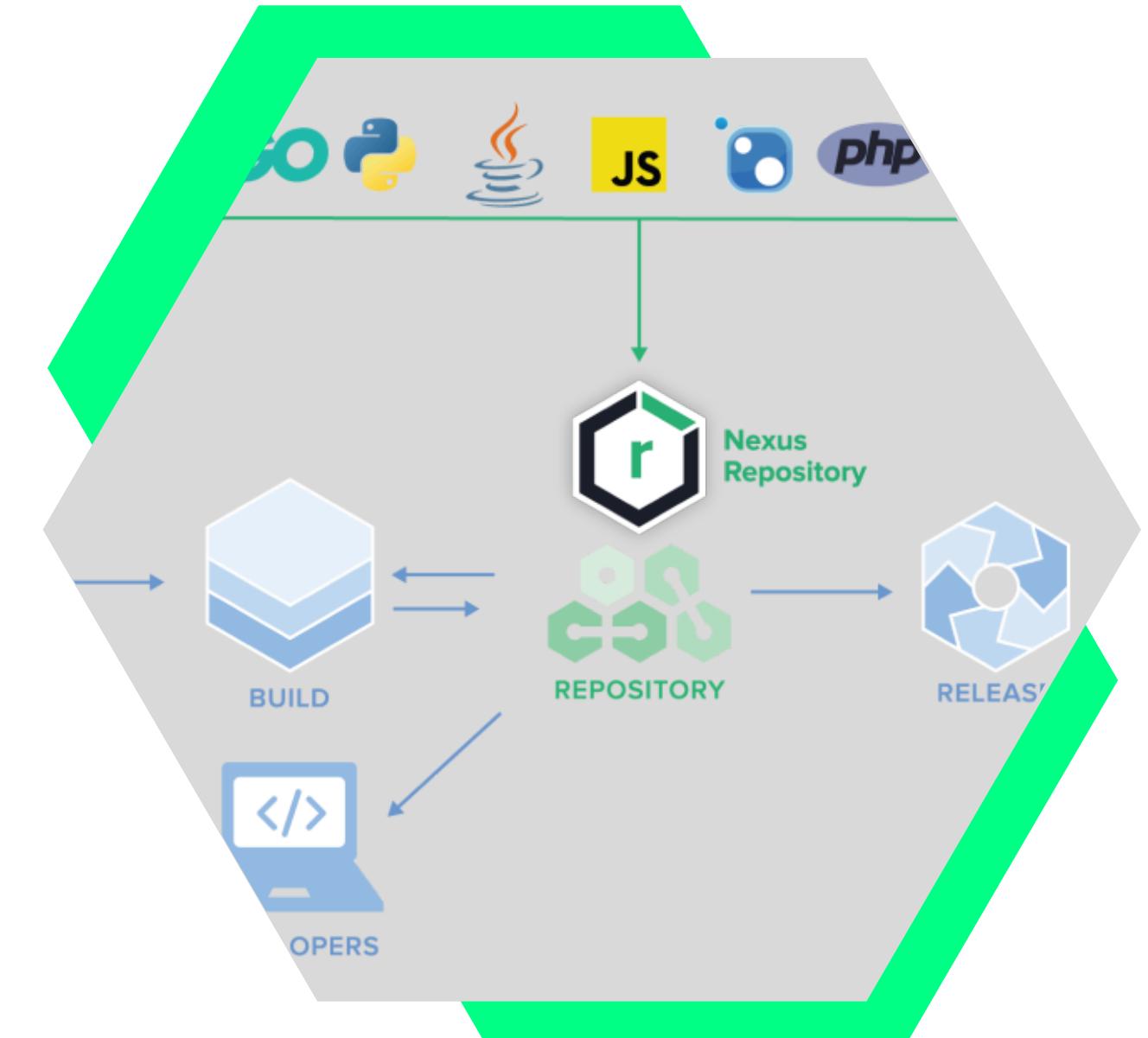
Proxy and Internal repositories

Can **cache external dependencies** to improve build speed and reliability (**Proxy**). Store your own company's artifacts securely (**Internal**).



Supports multiple formats

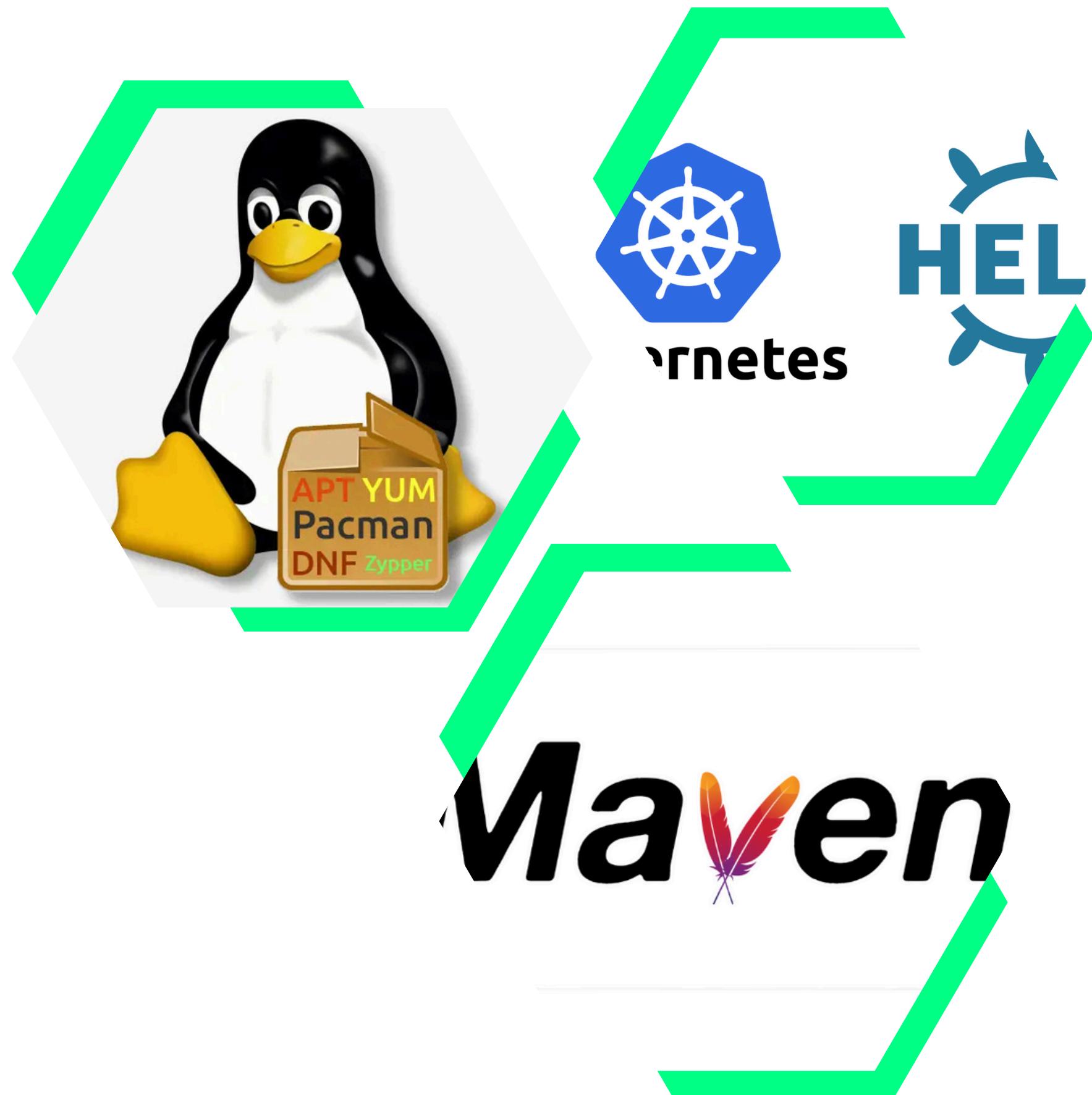
NPM, Maven, Docker, PyPI, Nuget, And more



Use Case

Nexus Repository Manager can be used for various purposes.

- **Docker:** Developers or CI/CD pipelines push Docker images to Nexus and pull them for deployment.
- **Kubernetes:** Teams can pull Helm charts from Nexus to deploy applications on Kubernetes clusters
- **Linux:** Developers or servers can pull packages from Nexus instead of public repos
- **Developer:** Maven pulls dependencies from Nexus (or external repositories via Nexus proxy).



Why do we choose Nexus?

If you want **reliable**, **multi-format**, **lightweight**, and **mostly free repository management**, especially in environments like Iran with restricted external access, **Nexus OSS** is the better choice. JFrog is great for large enterprises needing replication, HA, and advanced security but it comes at higher cost and complexity.

Factor	Nexus OSS	JFrog Artifactory
Free Features	Broad support for many formats	Limited in CE, Pro is paid
Setup & Maintenance	Simple	More complex
Resource Usage	Lightweight	Heavier
Security Features	Good (OSS basic, Pro advanced)	CE limited, Pro advanced

Nexus Repository

Architecture

Nexus Repository is designed around a **centralized artifact storage** and **management** system. Its architecture includes:

Web Application / UI

- Runs on Jetty (embedded Java web server)
- Accessible via browser (<http://<host>:8081>)

Repository Manager

- Hosted – your own artifacts
- Proxy – cached external repositories
- Group – combination of multiple repositories

Storage Layer

- Default: File system (can be mounted to local disk, network storage, or Docker volumes)
- Stores all artifacts, metadata, and configuration.

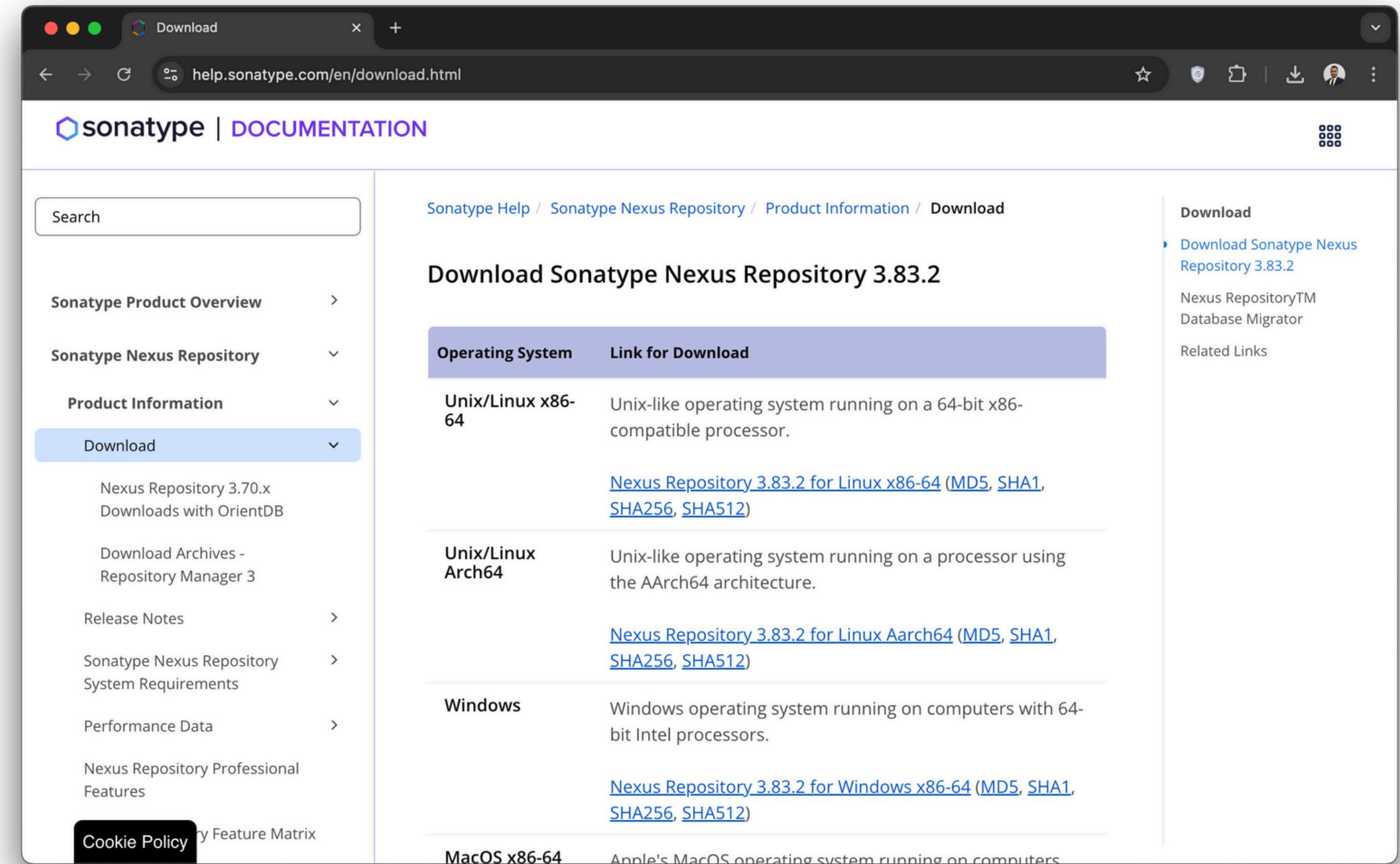
Security Layer

- Users, roles, privileges.
- Optional SSL, LDAP, SSO integration (Pro features for enterprise).



single-node standalone installation

Download Nexus Repository Manager according to your operating system and CPU architecture.



The screenshot shows a web browser window with the URL help.sonatype.com/en/download.html. The page is titled "sonatype | DOCUMENTATION". On the left, there's a navigation sidebar with links like "Sonatype Product Overview", "Sonatype Nexus Repository", "Product Information", "Download" (which is currently selected), and "Release Notes". The main content area is titled "Download Sonatype Nexus Repository 3.83.2". It features a table with columns for "Operating System" and "Link for Download". The table includes rows for "Unix/Linux x86-64", "Unix/Linux Arch64", "Windows", and "MacOS x86-64". Each row provides a description and a link to download the software. A sidebar on the right is titled "Download" and lists "Download Sonatype Nexus Repository 3.83.2", "Nexus RepositoryTM Database Migrator", and "Related Links".

Operating System	Link for Download
Unix/Linux x86-64	Unix-like operating system running on a 64-bit x86-compatible processor. Nexus Repository 3.83.2 for Linux x86-64 (MD5, SHA1, SHA256, SHA512)
Unix/Linux Arch64	Unix-like operating system running on a processor using the AArch64 architecture. Nexus Repository 3.83.2 for Linux AArch64 (MD5, SHA1, SHA256, SHA512)
Windows	Windows operating system running on computers with 64-bit Intel processors. Nexus Repository 3.83.2 for Windows x86-64 (MD5, SHA1, SHA256, SHA512)
MacOS x86-64	Apple's MacOS operating system running on computers.

single-node standalone installation

Create a **new directory** named **sonatype**. Then, **untar** and decompress the **nexus tar file** into the newly created sonatype directory.



```
[root@nexus-repository sonatype]# pwd
/opt/sonatype
[root@nexus-repository sonatype]# mv /opt/nexus-3.83.2-01-linux-aarch_64.tar.gz /opt/sonatype/
[root@nexus-repository sonatype]# tar -xavf nexus-3.83.2-01-linux-aarch_64.tar.gz
nexus-3.83.2-01/bin/nexus
nexus-3.83.2-01/bin/nexus.vmoptions
nexus-3.83.2-01/
nexus-3.83.2-01/etc/
nexus-3.83.2-01/etc/ssl/
nexus-3.83.2-01/etc/fabric/
nexus-3.83.2-01/etc/jetty/
nexus-3.83.2-01/etc/spring/
nexus-3.83.2-01/etc/logback/
nexus-3.83.2-01/bin/
nexus-3.83.2-01/deploy/
sonatype-work/
sonatype-work/nexus3/
sonatype-work/nexus3/tmp/
sonatype-work/nexus3/log/
nexus-3.83.2-01/etc/ssl/.placeholder
nexus-3.83.2-01/etc/fabric/mybatis.xml
nexus-3.83.2-01/etc/fabric/ehcache.xml
nexus-3.83.2-01/etc/fabric/elasticsearch.yml
nexus-3.83.2-01/etc/nexus-default.properties
```

single-node standalone installation

Create a symbolic link for the nexus-3.83.2-01 directory to make it easily accessible. Then, create a nexus user and set it as the owner of the sonatype directory.



```
mehdiabdollahei — root@nexus-repository:/opt/sonatype — ssh root@192.168.83.132 — 94x24
[root@nexus-repository sonatype]# ln -s /opt/sonatype/nexus-3.83.2-01 /opt/sonatype/nexus
[root@nexus-repository sonatype]# useradd -r -s /bin/false nexus
[root@nexus-repository sonatype]# chown -R nexus:nexus /opt/sonatype
[root@nexus-repository sonatype]#
```

single-node standalone installation

Configure Nexus to **run with the nexus user** and **listen on my server's IP addresses**. In my example, the IP address is 192.168.83.132.

```
mehdabdollahei — root@nexus-repository:/opt/sonatype — ssh root@192.168.83.132 — 106x9  
[root@nexus-repository sonatype]# vim /opt/sonatype/nexus/etc/nexus-default.properties  
[root@nexus-repository sonatype]# grep -i 192.168.83.132 /opt/sonatype/nexus/etc/nexus-default.properties  
application-host=192.168.83.132  
[root@nexus-repository sonatype]#  
[root@nexus-repository sonatype]# vim /opt/sonatype/nexus/bin/nexus  
[root@nexus-repository sonatype]# grep -i "run_as_user='nexus'" /opt/sonatype/nexus/bin/nexus  
run_as_user='nexus'  
[root@nexus-repository sonatype]#
```

single-node standalone installation

Configure **Systemd** to manage the **Nexus service**. We should create a **service file** for it.

```
mehdiabdollahei — root@nexus-repository:/opt/sonatype — ssh root@192.168.83.132 — 91x24
[[root@nexus-repository sonatype]# touch /etc/systemd/system/nexus.service
[[root@nexus-repository sonatype]# vim /etc/systemd/system/nexus.service
[[root@nexus-repository sonatype]# cat /etc/systemd/system/nexus.service
[Unit]
Description=nexus service
After=network.target

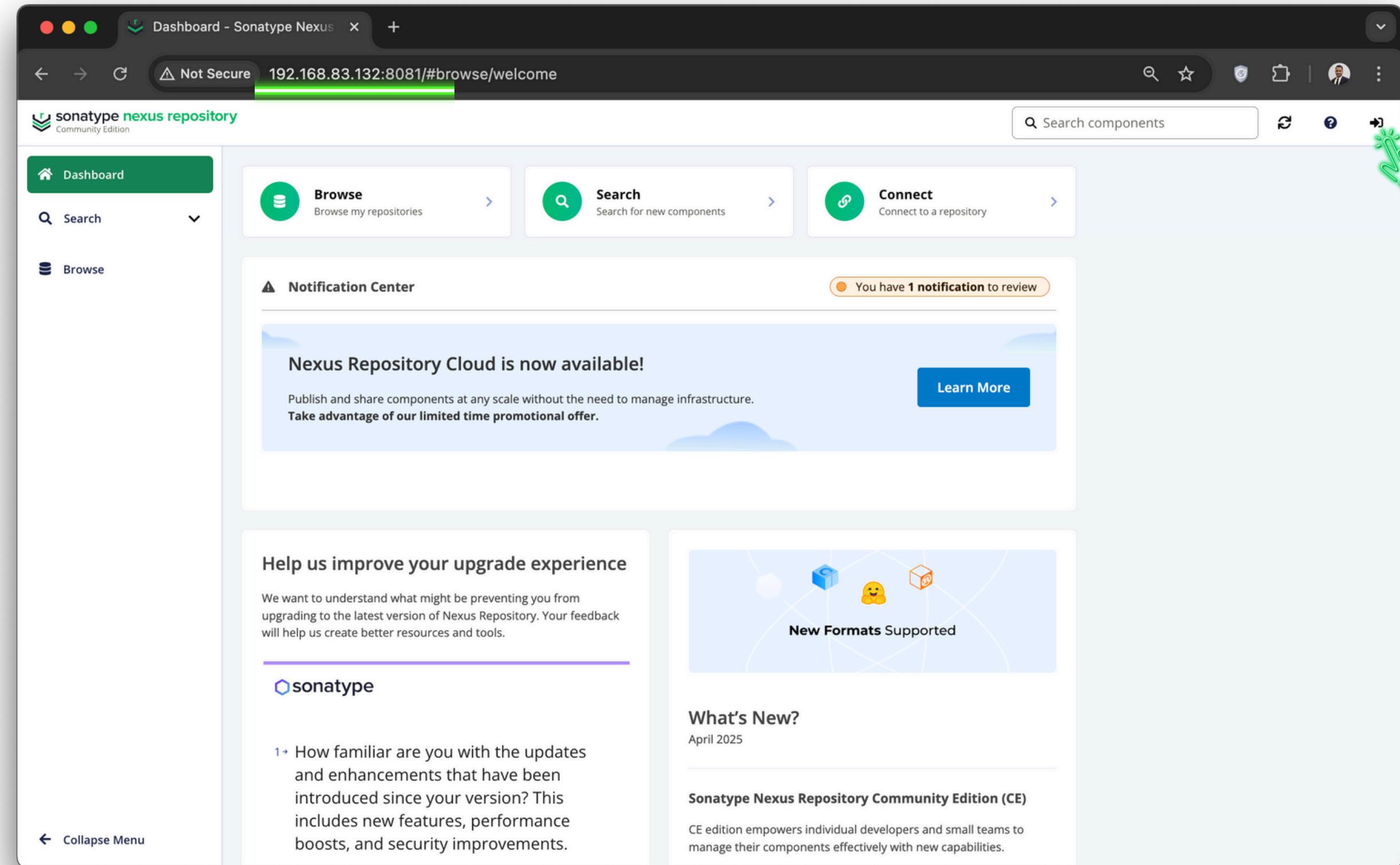
[Service]
Type=forking
LimitNOFILE=65536
ExecStart=/opt/sonatype/nexus/bin/nexus start
ExecStop=/opt/sonatype/nexus/bin/nexus stop

User=nexus
Restart=on-abort
TimeoutSec=600

[Install]
WantedBy=multi-user.target
[[root@nexus-repository sonatype]# systemctl daemon-reload
[[root@nexus-repository sonatype]# systemctl restart nexus && systemctl enable nexus
Created symlink /etc/systemd/system/multi-user.target.wants/nexus.service → /etc/systemd/sys
tem/nexus.service.
[[root@nexus-repository sonatype]# ]]
```

single-node standalone installation

Use the **browser Nexus** to access the **server's IP address** on **port 8081**.



single-node standalone installation

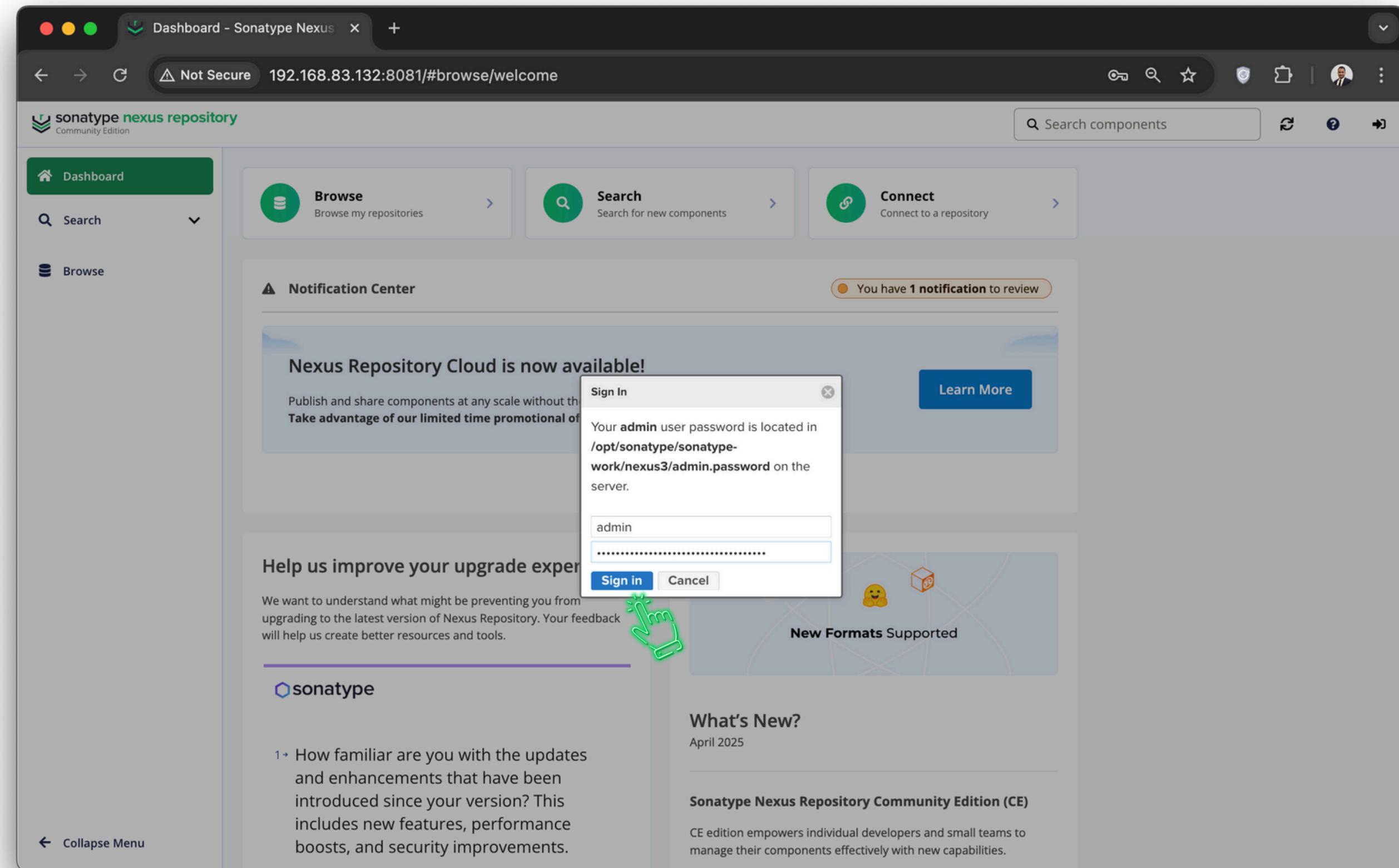
Obtain the **admin's password** to access the Nexus dashboard.



```
mehdiabdollahei — root@nexus-repository:/opt/sonatype — ssh root@192.168.83.132 — 92x17
[root@nexus-repository sonatype]# cat /opt/sonatype/sonatype-work/nexus3/admin.password
1b9d4d94-0b25-408f-b462-104be47db0c5[root@nexus-repository sonatype]#
```

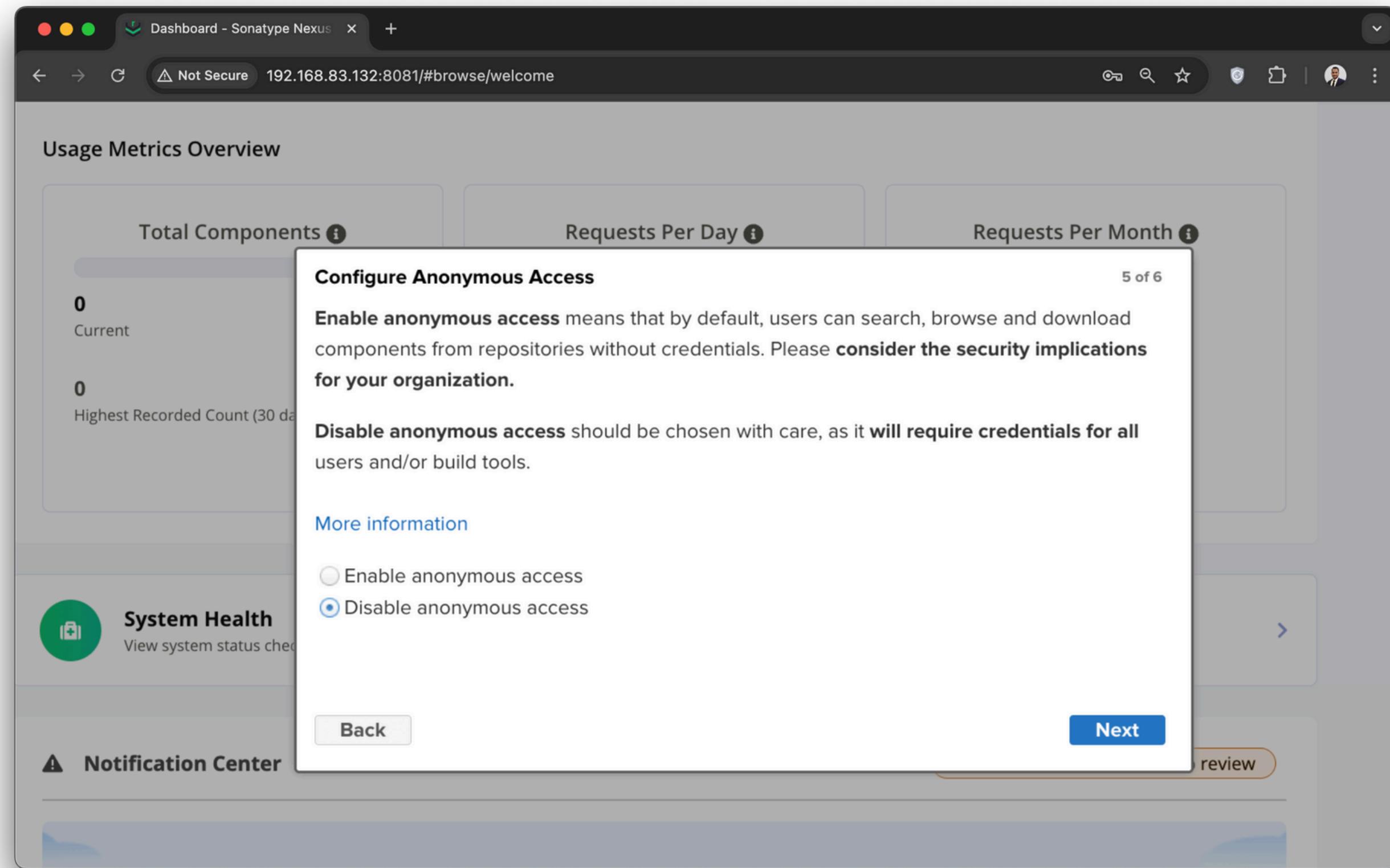
single-node standalone installation

Username: **admin** and password obtained in last slide



single-node standalone installation

Change admin's password and set disable anonymous access



single-node standalone installation

Create a partition for each blob store.

```
mehdiabdollahei - root@nexus-repository:~ - ssh root@192.168.83.132 - 94x30
[root@nexus-repository ~]# fdisk /dev/sdb << EOF >> /dev/null
g
n
1
+2.5G
w
EOF
[root@nexus-repository ~]# mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1          isize=512    agcount=4, agsize=163840 blks
                           =           sectsz=512   attr=2, projid32bit=1
                           =           crc=1      finobt=1, sparse=1, rmapbt=0
                           =           reflink=1  bigtime=1 inobtcount=1 nrext64=0
data        =           bsize=4096   blocks=655360, imaxpct=25
                           =           sunit=0     swidth=0 blks
naming      =version 2       bsize=4096   ascii-ci=0, ftype=1
log         =internal log    bsize=4096   blocks=16384, version=2
                           =           sectsz=512   sunit=0 blks, lazy-count=1
realtime    =none            extsz=4096   blocks=0, rtextents=0
[root@nexus-repository ~]# mkdir /opt/sonatype/sonatype-work/nexus3/rockylinux-blobstore
[root@nexus-repository ~]# echo "/dev/sdb1 /opt/sonatype/sonatype-work/nexus3/rockylinux-blobs
tore xfs defaults 0 0" | sudo tee -a /etc/fstab
/dev/sdb1 /opt/sonatype/sonatype-work/nexus3/rockylinux-blobstore xfs defaults 0 0
[root@nexus-repository ~]#
[root@nexus-repository ~]# systemctl daemon-reload && mount -a
[root@nexus-repository ~]# df -h | grep -i sdb1
/dev/sdb1      2.5G  50M  2.4G  2% /opt/sonatype/sonatype-work/nexus3/rockylinux-blobs
tore
[root@nexus-repository ~]# chown -R nexus:nexus /opt/sonatype/
[root@nexus-repository ~]#
```

single-node standalone installation

Mount your partition



```
mehdiabdollahei — root@nexus-repository:~ — ssh root@192.168.83.132 — 103x13
[root@nexus-repository ~]# mkdir /opt/sonatype/sonatype-work/nexus3/rockylinux-blobstore
[root@nexus-repository ~]# echo "/dev/sdb1 /opt/sonatype/sonatype-work/nexus3/rockylinux-blobstore xfs
defaults 0 0" | sudo tee -a /etc/fstab
/dev/sdb1 /opt/sonatype/sonatype-work/nexus3/rockylinux-blobstore xfs defaults 0 0
[root@nexus-repository ~]#
[root@nexus-repository ~]# sudo chown -R nexus:nexus /opt/sonatype
[root@nexus-repository ~]#
[root@nexus-repository ~]# df -h | grep -i sdb1
[root@nexus-repository ~]# mount -a
[root@nexus-repository ~]# df -h | grep -i sdb1
/dev/sdb1          2.5G   50M  2.4G  2% /opt/sonatype/sonatype-work/nexus3/rockylinux-blobstore
[root@nexus-repository ~]#
```

single-node standalone installation

Set **nexus** as a ownership of new directory



```
mehdiabdollahei — root@nexus-repository:~ — ssh root@192.168.83.132 — 95x13

[root@nexus-repository ~]# ls -ld /opt/sonatype/sonatype-work/nexus3/rockylinux-blobstore
drwxr-xr-x 2 root root 6 Sep  8 08:01 /opt/sonatype/sonatype-work/nexus3/rockylinux-blobstore
[root@nexus-repository ~]#
[root@nexus-repository ~]# chown -R nexus:nexus /opt/sonatype/
[root@nexus-repository ~]#
[root@nexus-repository ~]# ls -ld /opt/sonatype/sonatype-work/nexus3/rockylinux-blobstore
drwxr-xr-x 2 nexus nexus 6 Sep  8 08:01 /opt/sonatype/sonatype-work/nexus3/rockylinux-blobstore
[root@nexus-repository ~]#
```

single-node standalone installation

Create **blob store per repository**

The screenshot shows the Sonatype Nexus Repository Manager interface. A large green checkmark is visible in the top right corner.

- 1** Click the gear icon in the bottom left corner of the sidebar.
- 2** Click **Repository** in the sidebar.
- 3** Click **Blob Stores** in the sidebar.
- 4** Click the **Create Blob Store** button.

The main page displays the **Blob Stores** configuration. A table lists one existing blob store:

NAME	PATH	TYPE	STATE	BLOB COUNT	TOTAL SIZE	AVAILABLE SPACE
default	default	File	Started	40	786.26 KB (805128)	13.83 GB

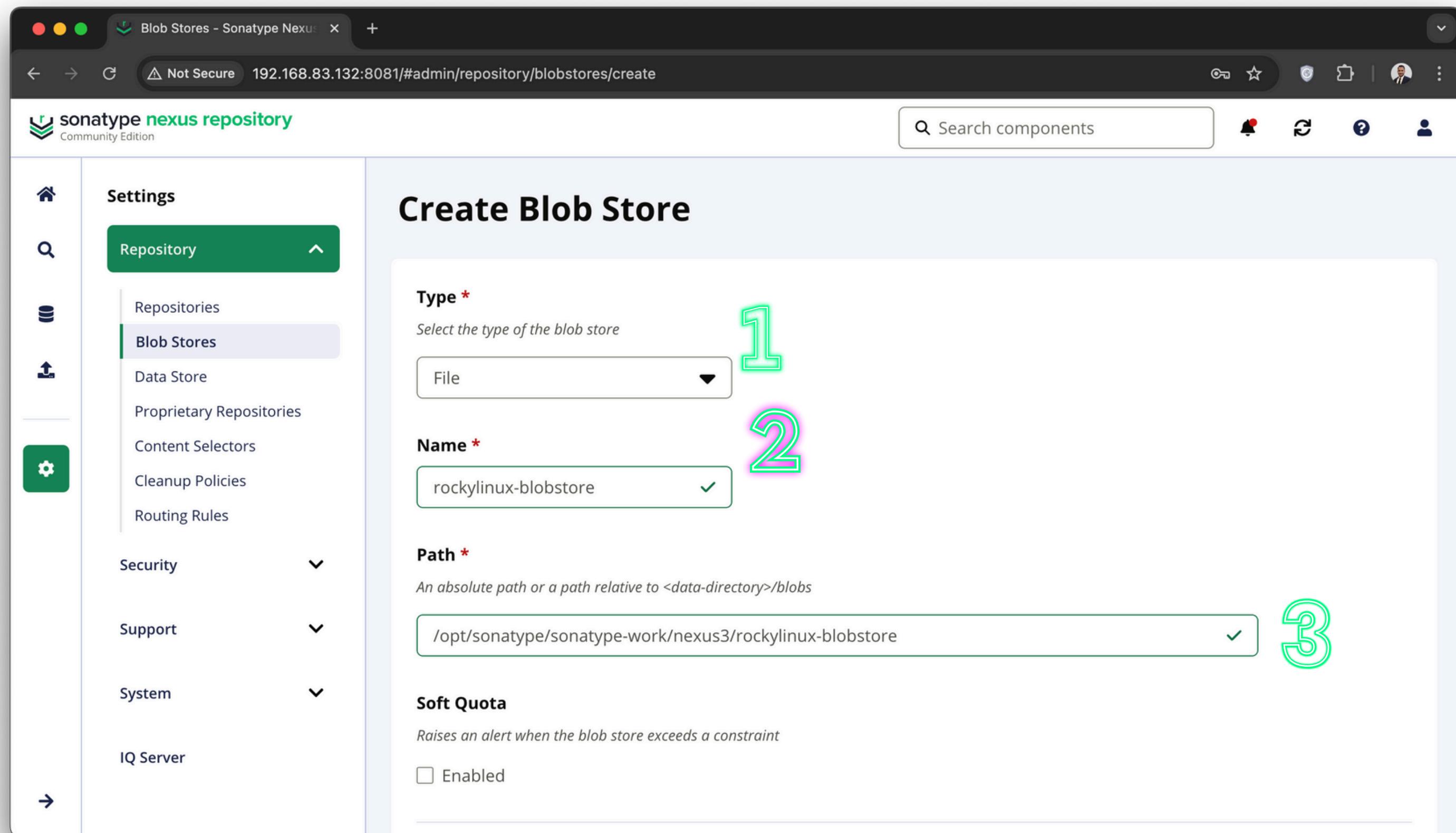
A callout box provides information about what a blob store is:

i What is a blob store?

The binary assets you download via proxy repositories, or publish to hosted repositories, are stored in the blob store attached to those repositories. In traditional, single node NXRM deployments, blob stores are typically associated with a local filesystem directory, usually within the sonatype-work directory. For more information, check [the documentation](#).

single-node standalone installation

Create **blob store per repository**



single-node standalone installation

Add **Rocky Linux's repository** as a **proxy repository**.

The screenshot shows the Sonatype Nexus Repository Manager interface. A large green checkmark is visible in the top right corner.

The interface is divided into two main sections:

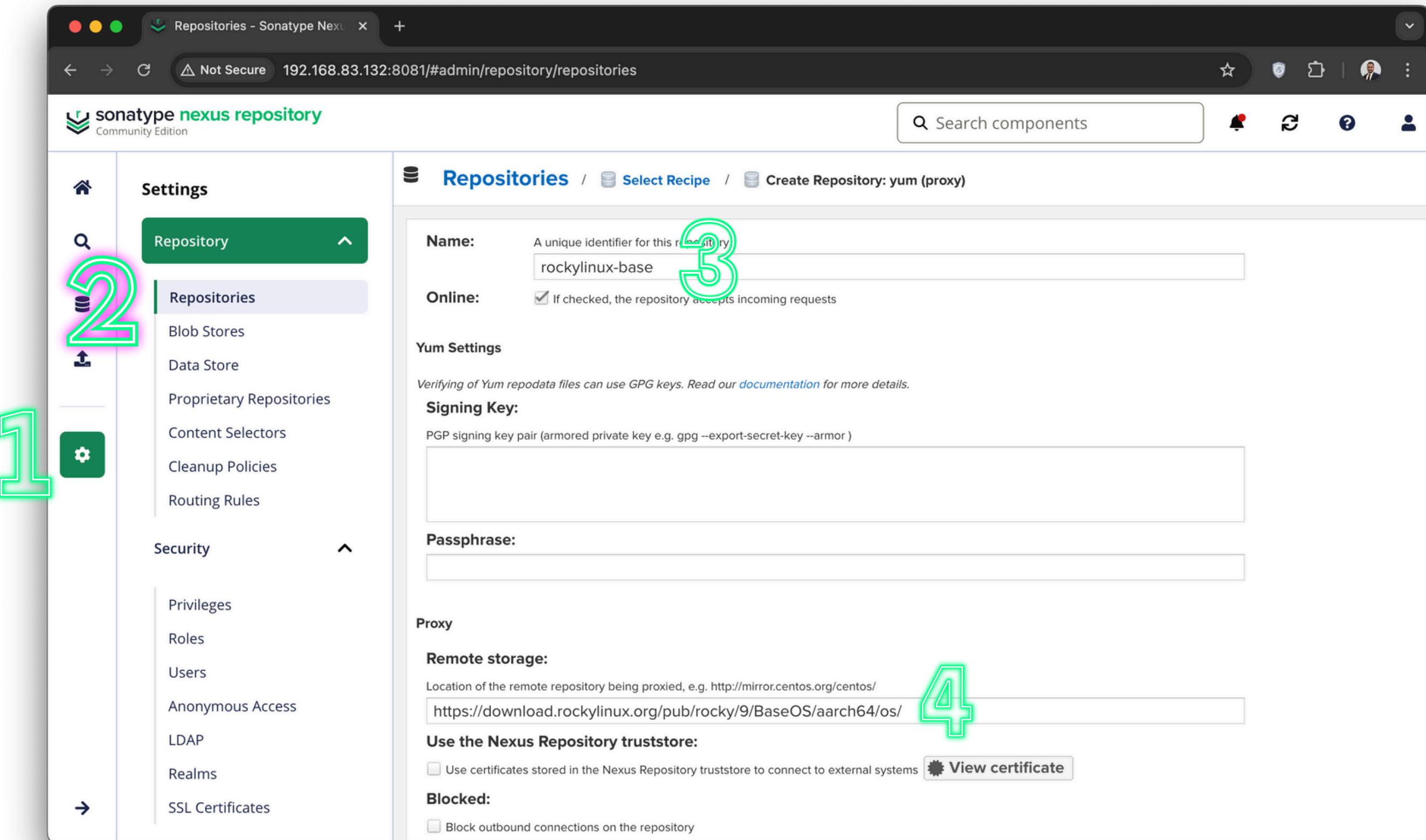
- Left Sidebar (Settings):** Contains links for **Repository**, **Repositories**, **Blob Stores**, **Data Store**, **Proprietary Repositories**, **Content Selectors**, **Cleanup Policies**, and **Routing Rules**. A green number **1** highlights the **Repositories** link.
- Right Main Area (Repositories):** Shows a table of existing repositories. A green number **2** highlights the **Create repository** button at the top left of the table area. A green number **3** highlights the **Name** column header in the table.

The table displays the following repository details:

Name	Type	Format	Blob Store	Status	URL	Health check	Firewall Re...
maven-central	proxy	maven2	default	Online - Read...	copy	0 0	
maven-public	group	maven2	default	Online	copy		
maven-releas...	hosted	maven2	default	Online	copy		
maven-snaps...	hosted	maven2	default	Online	copy		
nuget-group	group	nuget	default	Online	copy		
nuget-hosted	hosted	nuget	default	Online	copy		
nuget.org-pro...	proxy	nuget	default	Online - Read...	copy	0 0	

single-node standalone installation

Config **yum(proxy)** for **base repository** of **rocky linux**



single-node standalone installation

Config **yum(proxy)** for **base** repository of **rocky linux**

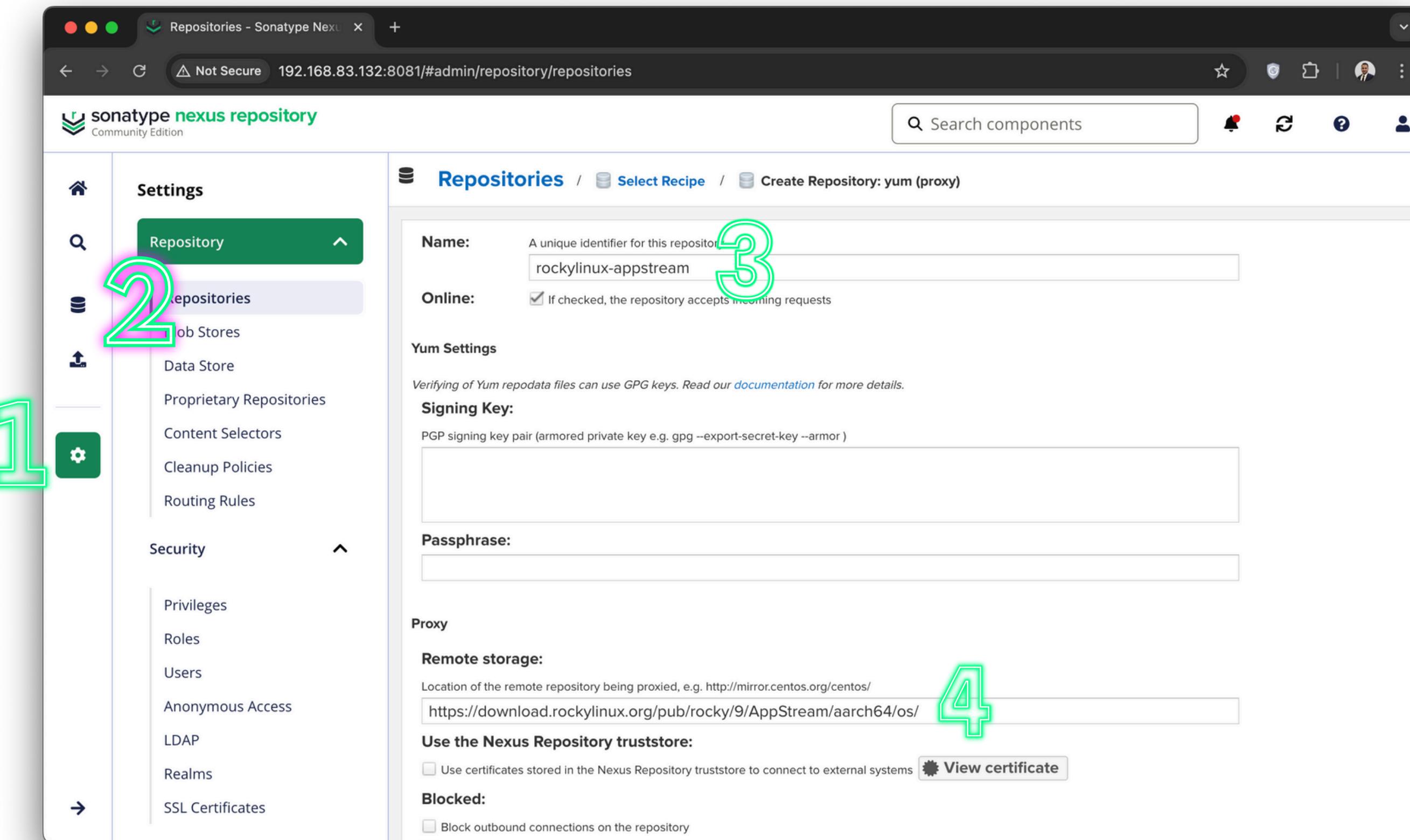
The screenshot shows the Sonatype Nexus Repository Manager interface. On the left, a sidebar titled "Settings" is open, with "Repository" selected. Under "Repository", "Repositories" is also selected. The main content area is titled "Repositories / Select Recipe / Create Repository: yum (proxy)".

Configuration details:

- Cache metadata before rechecking the remote repository: 1440 minutes.
- Storage:
 - Blob store: rockylinux-blobstore
- Strict Content Type Validation: Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format.
- Routing Rule: None.
- Negative Cache:
 - Not found cache enabled: Cache responses for content not present in the proxied repository.
 - Not found cache TTL: 1440 minutes.

single-node standalone installation

Config **yum(proxy)** for **AppStream** repository of **rocky linux**



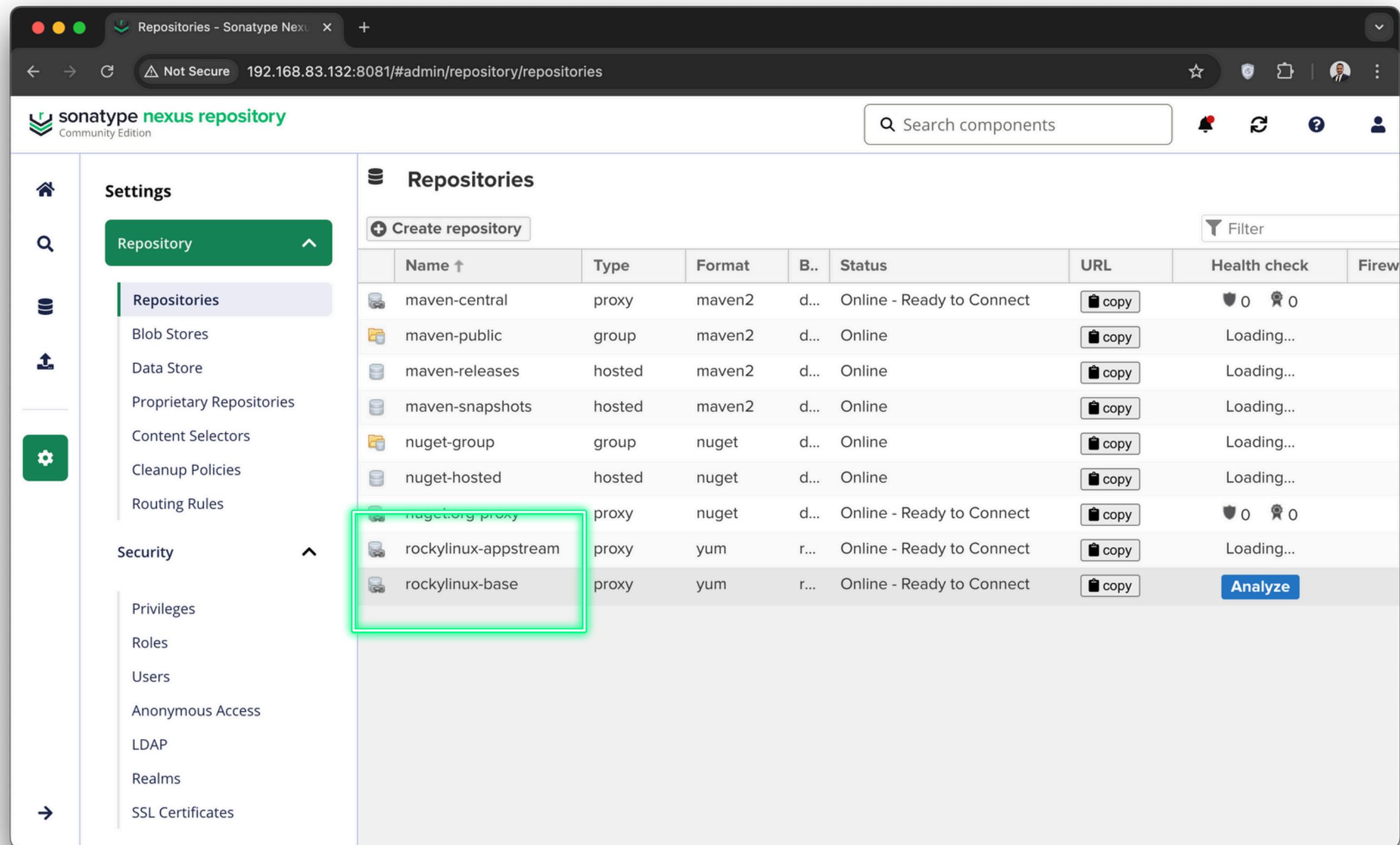
single-node standalone installation

Config **yum(proxy)** for **AppStream** repository of **rocky linux**

The screenshot shows the Sonatype Nexus Repository Manager interface. The left sidebar has a 'Settings' section with 'Repository' selected, and a 'Repositories' sub-section containing 'Blob Stores', 'Data Store', 'Proprietary Repositories', 'Content Selectors', 'Cleanup Policies', and 'Routing Rules'. The main content area is titled 'Repositories / Select Recipe / Create Repository: yum (proxy)'. It includes sections for 'Maximum component age' (set to -1), 'Maximum metadata age' (set to 1440), 'Storage' (blob store set to 'rockylinux-blobstore'), 'Strict Content Type Validation...' (checkbox checked), 'Routing Rule' (set to 'None'), 'Negative Cache' (checkbox checked), 'Not found cache enabled' (checkbox checked), and 'Not found cache TTL' (checkbox checked). A large green number '5' is overlaid on the 'Strict Content Type Validation...' field.

single-node standalone installation

Config yum(proxy)

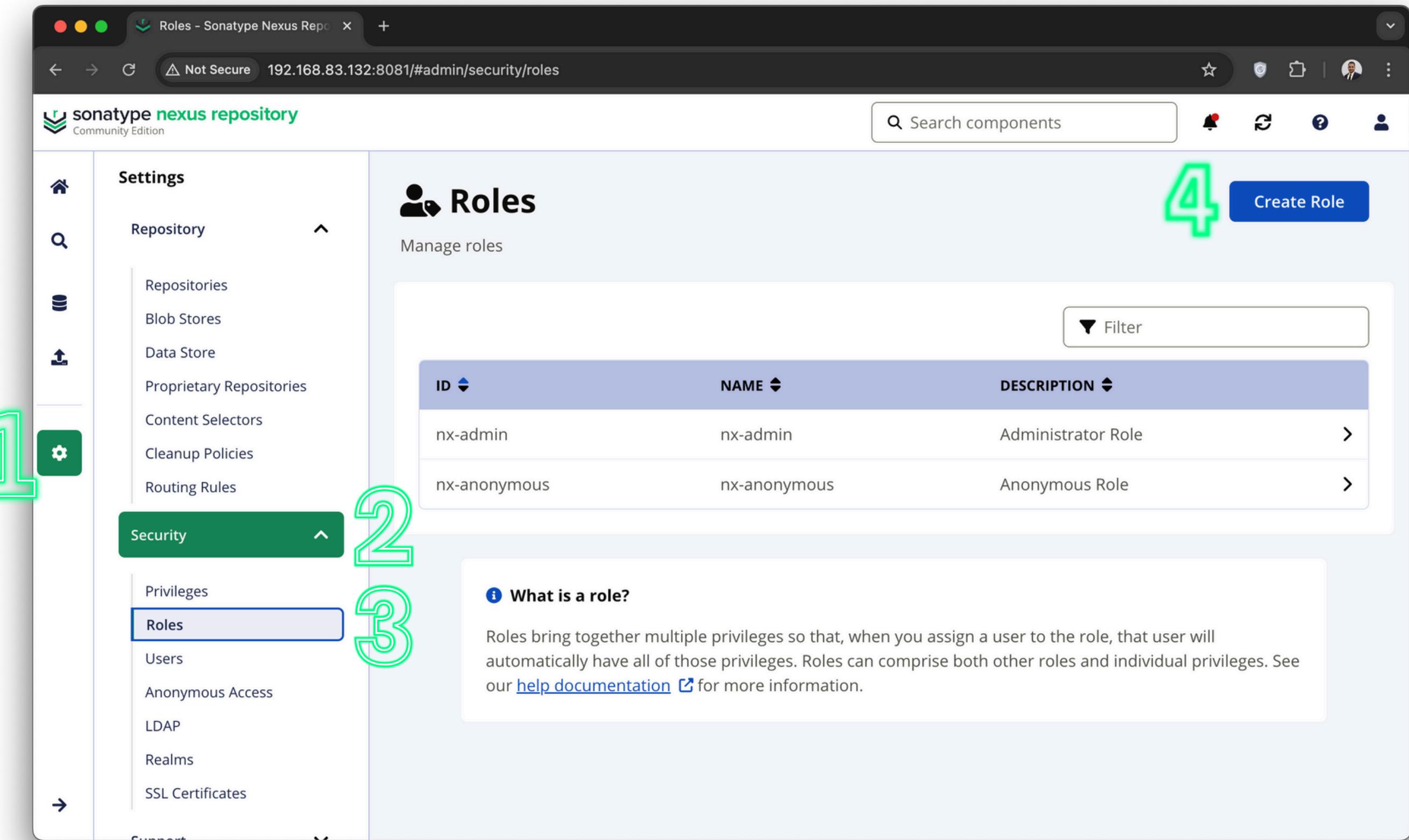


The screenshot shows the Sonatype Nexus Repository Manager interface. The left sidebar is titled 'Settings' and contains sections for 'Repository' (selected), 'Repositories' (highlighted in green), 'Blob Stores', 'Data Store', 'Proprietary Repositories', 'Content Selectors', 'Cleanup Policies', and 'Routing Rules'. Below this is another 'Repository' section for 'Security' with 'Privileges', 'Roles', 'Users', 'Anonymous Access', 'LDAP', 'Realms', and 'SSL Certificates'. The main content area is titled 'Repositories' and includes a 'Create repository' button. A table lists the following repositories:

Name ↑	Type	Format	B..	Status	URL	Health check	Firew...
maven-central	proxy	maven2	d...	Online - Ready to Connect	<button>copy</button>	0 0	
maven-public	group	maven2	d...	Online	<button>copy</button>	Loading...	
maven-releases	hosted	maven2	d...	Online	<button>copy</button>	Loading...	
maven-snapshots	hosted	maven2	d...	Online	<button>copy</button>	Loading...	
nuget-group	group	nuget	d...	Online	<button>copy</button>	Loading...	
nuget-hosted	hosted	nuget	d...	Online	<button>copy</button>	Loading...	
nugget.org-proxy	proxy	nuget	d...	Online - Ready to Connect	<button>copy</button>	0 0	
rockylinux-appstream	proxy	yum	r...	Online - Ready to Connect	<button>copy</button>	Loading...	
rockylinux-base	proxy	yum	r...	Online - Ready to Connect	<button>copy</button>	Analyze	

single-node standalone installation

Create a user and a read role to prevent anonymous access to the repository



The screenshot shows the 'Roles' management screen in the Sonatype Nexus Repository Manager. The interface is organized into a sidebar and a main content area.

1 In the sidebar, under the 'Settings' section, the 'Security' tab is selected. The 'Roles' option is highlighted with a blue border.

2 Under the 'Security' tab, the 'Roles' option is selected.

3 The main content area displays a table titled 'Roles' with two entries:

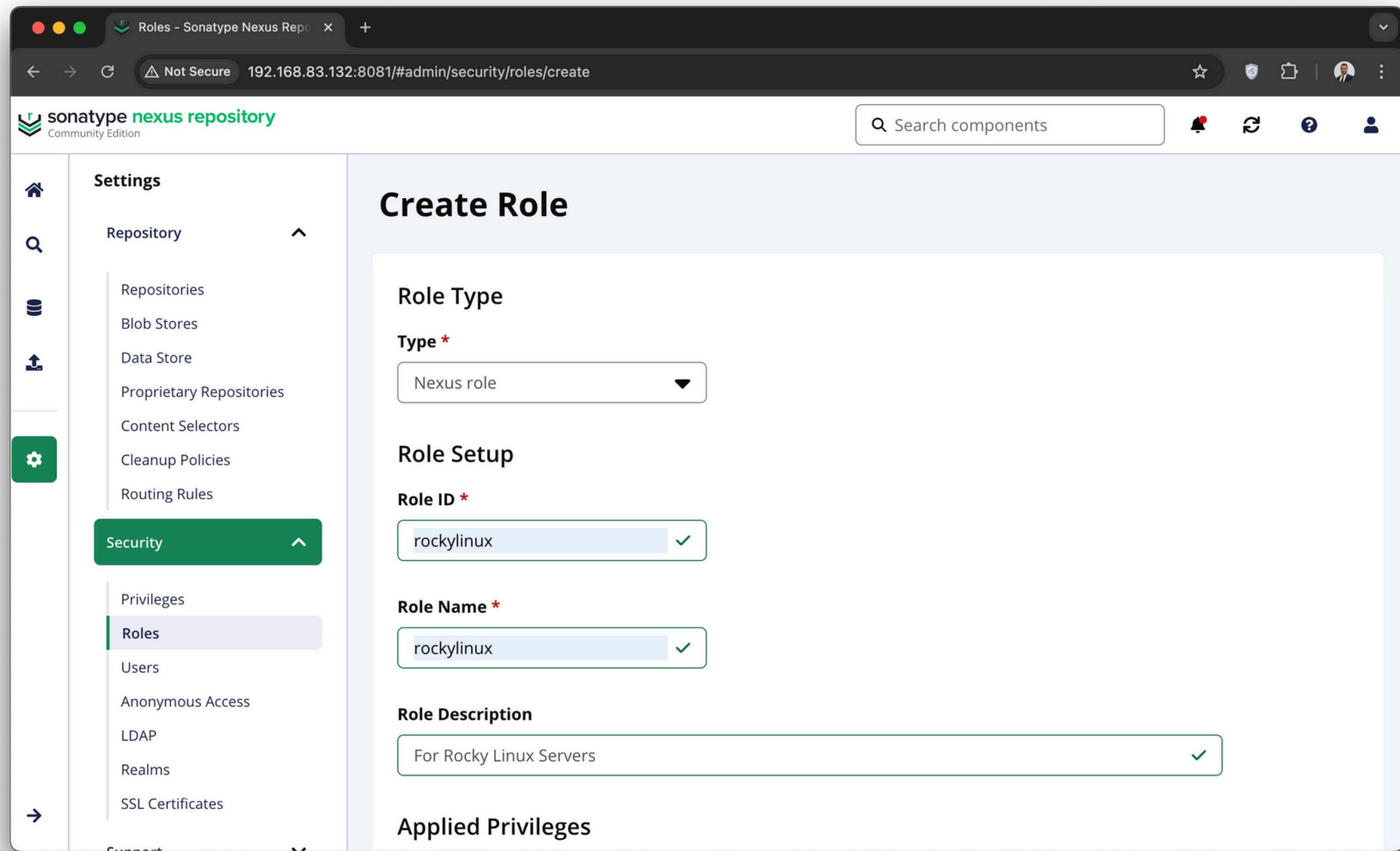
ID	NAME	DESCRIPTION
nx-admin	nx-admin	Administrator Role
nx-anonymous	nx-anonymous	Anonymous Role

4 In the top right corner of the main content area, there is a blue button labeled 'Create Role'.

What is a role?
Roles bring together multiple privileges so that, when you assign a user to the role, that user will automatically have all of those privileges. Roles can comprise both other roles and individual privileges. See our [help documentation](#) for more information.

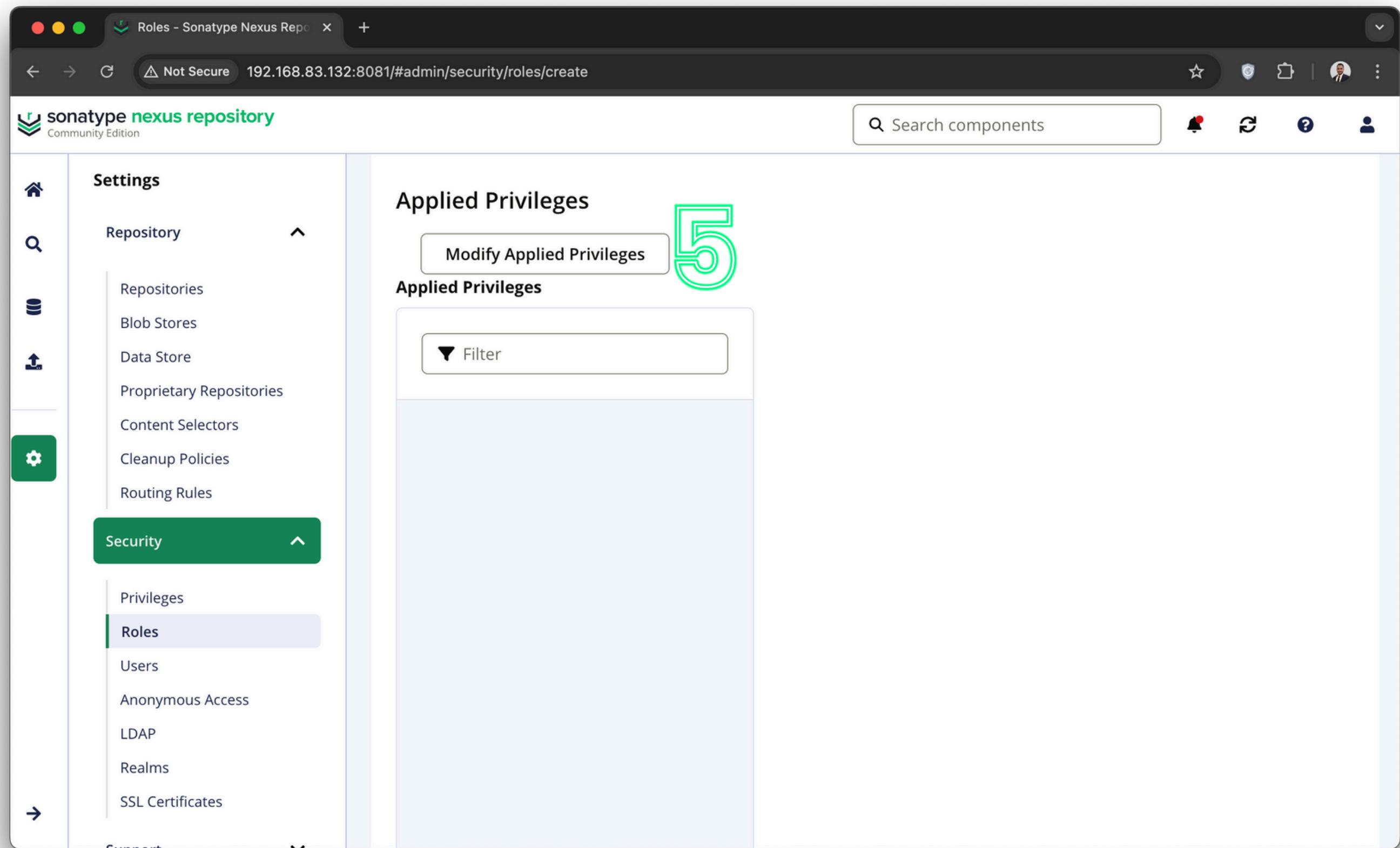
single-node standalone installation

Create a **user** and a **read role** to prevent anonymous access to the **repository**



single-node standalone installation

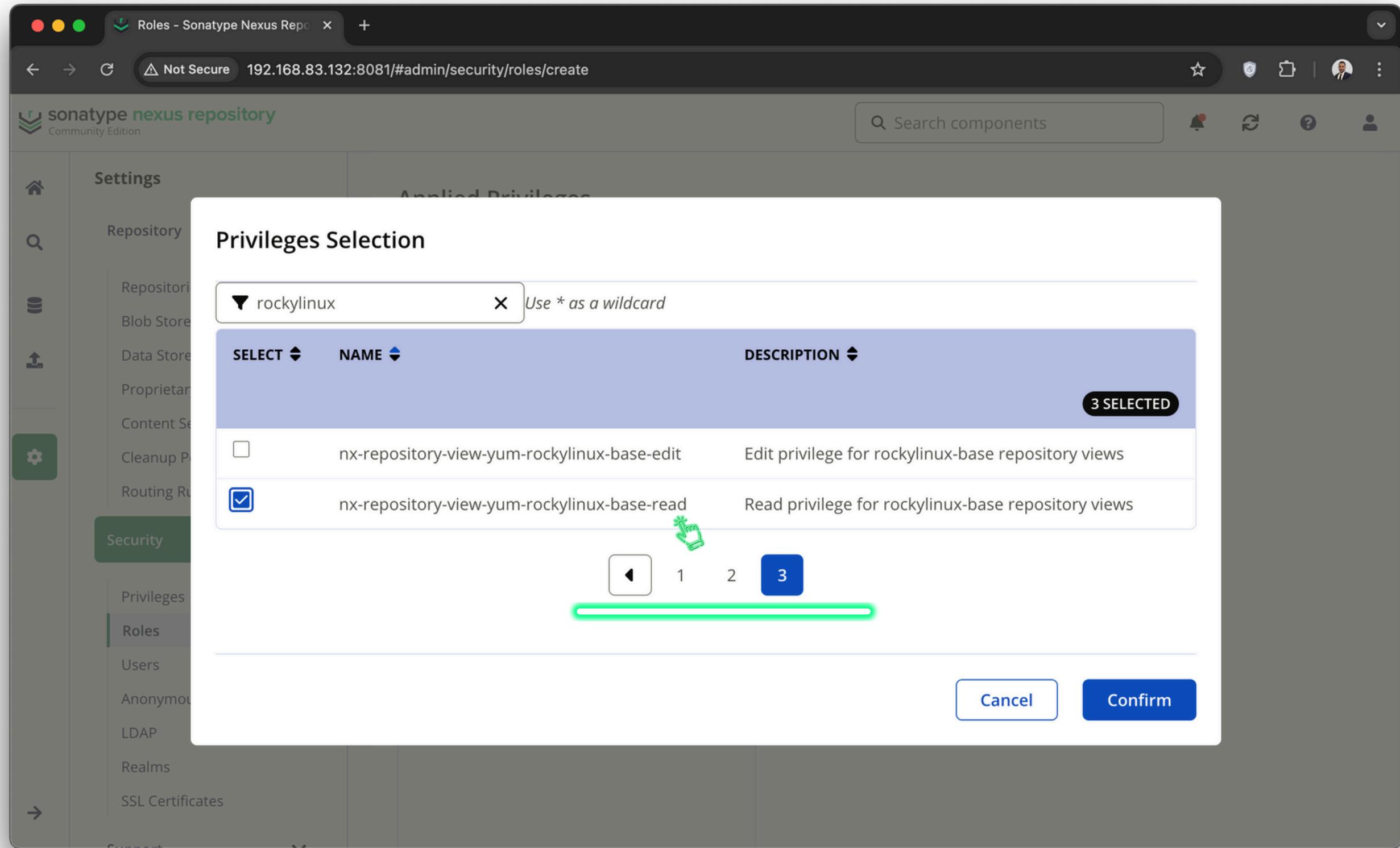
Create a **user** and a **read role** to **prevent anonymous access** to the **repository**



single-node standalone installation

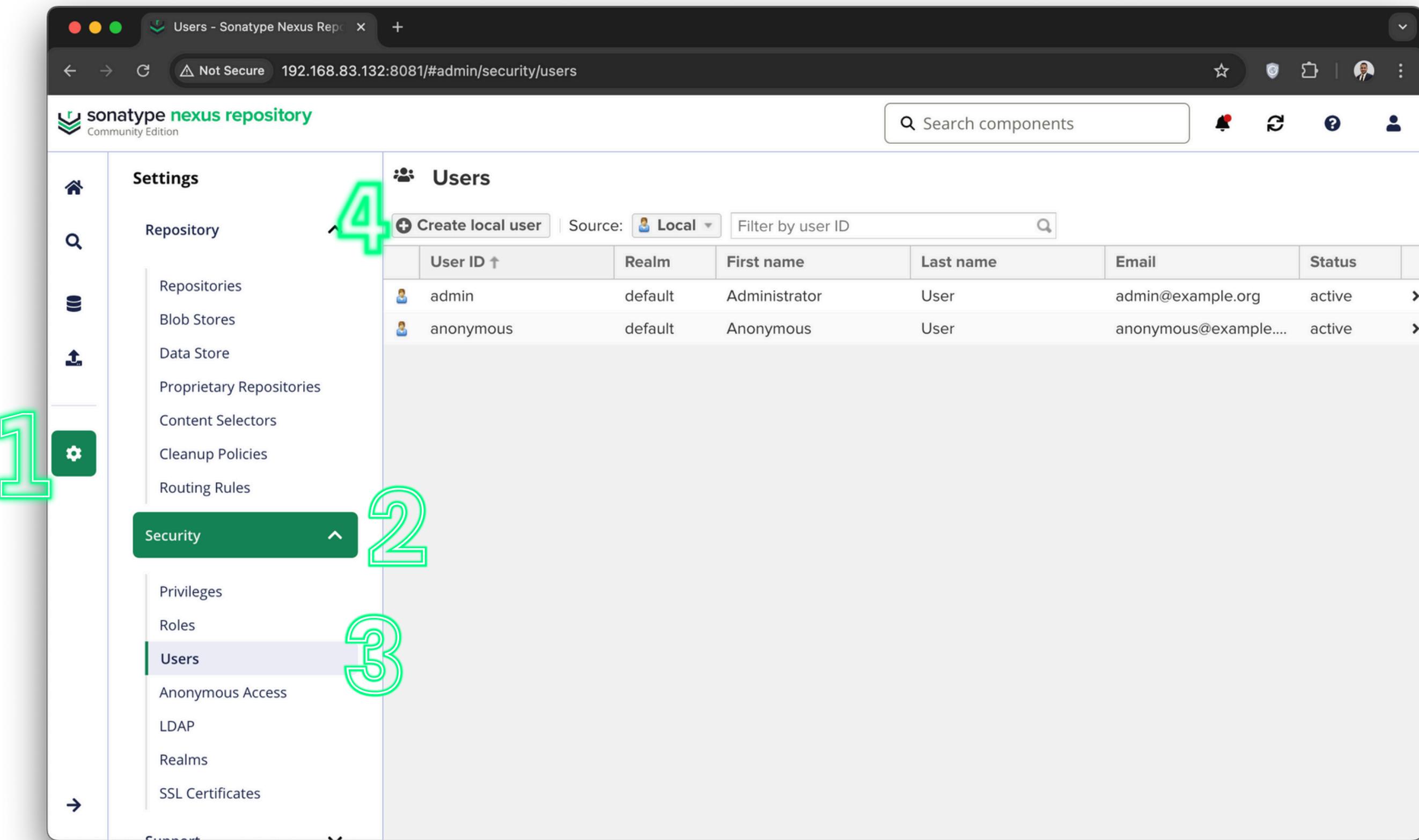
Create a **user** and a **read role** to prevent anonymous access to the **repository**

Take **read access** every three pages



single-node standalone installation

Create a **user** and a **read role** to prevent anonymous access to the **repository**



single-node standalone installation

Create a **user** and a **read role** to prevent anonymous access to the **repository**

The screenshot shows the Sonatype Nexus Repository Manager web interface. The URL in the browser is `192.168.83.132:8081/#admin/security/users`. The left sidebar is titled "Settings" and includes sections for "Repository", "Security", and "Privileges". The "Security" section is currently active and has a sub-menu with "Users" selected, which is highlighted with a green bar. Other options in the sub-menu include "Anonymous Access", "LDAP", "Realms", and "SSL Certificates". The main content area is titled "Users / Create User". It contains fields for "ID" (set to "rockylinux"), "First name" (set to "rockylinux"), "Last name" (set to "rockylinux"), "Email" (set to "test@test.local"), "Password" (a masked password), "Confirm password" (a masked password), and "Status" (set to "Active"). Below these fields is a "Roles:" section. Under "Available" roles, "nx-admin" and "nx-anonymous" are listed. Under "Granted" roles, "rockylinux" is selected. A "Filter" input field is also present in the "Available" section.

single-node standalone installation

Configure **Rocky Linux** to use a **local repository**.



```
[root@rockylinux-server ~]# mkdir /etc/yum.repos.d/backup
[root@rockylinux-server ~]# mv /etc/yum.repos.d/* /etc/yum.repos.d/backup
mv: cannot move '/etc/yum.repos.d/backup' to a subdirectory of itself, '/etc/yum.repos.d/backup/backup'
[root@rockylinux-server ~]# touch /etc/yum.repos.d/nexus.repo
[root@rockylinux-server ~]# vim /etc/yum.repos.d/nexus.repo
[root@rockylinux-server ~]# cat /etc/yum.repos.d/nexus.repo
[rockylinux-base]
name=Rocky Linux BaseOS
baseurl=http://rockylinux:qazwsx@192.168.83.132:8081/repository/rockylinux-base/
enabled=1
gpgcheck=0

[rockylinux-appstream]
name=Rocky Linux AppStream
baseurl=http://rockylinux:qazwsx@192.168.83.132:8081/repository/rockylinux-appstream/
enabled=1
gpgcheck=0
[root@rockylinux-server ~]#
```

single-node standalone installation

Configure **Rocky Linux** to use a **local repository** – TEST

```
mehdiabdollahei - root@rockylinux-server:~ - ssh root@192.168.83.150 - 94x20

[[root@rockylinux-server ~]# dnf clean all
6 files removed
[[root@rockylinux-server ~]# dnf repolist
repo id                                repo name
rockylinux-appstream                      Rocky Linux AppStream
rockylinux-base                            Rocky Linux BaseOS
[[root@rockylinux-server ~]# dnf update
Rocky Linux BaseOS                         332 kB/s | 2.6 MB   00:07
Rocky Linux AppStream                      797 kB/s | 8.2 MB   00:10
Last metadata expiration check: 0:00:01 ago on Tue 09 Sep 2025 05:53:14 AM EDT.
Dependencies resolved.

=====
          Package        Arch      Version       Repository      Size
=====
Installing:
  kernel           aarch64  5.14.0-570.37.1.el9_6    rockylinux-base  1.8 M
Upgrading:
  NetworkManager   aarch64  1:1.52.0-5.el9_6     rockylinux-base  2.2 M
  NetworkManager-libnm   aarch64  1:1.52.0-5.el9_6     rockylinux-base  1.8 M
  NetworkManager-team   aarch64  1:1.52.0-5.el9_6     rockylinux-base  27 k
```

single-node standalone installation

Create a **Yum(group)** for better management

The screenshot shows the Sonatype Nexus Repository Manager interface. A large green checkmark is visible in the top right corner.

The left sidebar has a green outline around the 'Settings' section, with a large green number '1' to its left. Under 'Repository', the 'Repositories' item is highlighted with a blue outline and a large green number '2' to its left. The main content area is titled 'Repositories' and shows a table of existing repositories. A large green number '3' is overlaid on the 'Create repository' button. The table columns are: Name ↑, Type, Format, Blob Store, Status, URL, Health check, and Firewall Re... . The repositories listed are:

Name ↑	Type	Format	Blob Store	Status	URL	Health check	Firewall Re...
maven-central	proxy	maven2	default	Online - Read...	copy	0 0	
maven-public	group	maven2	default	Online	copy		
maven-releas...	hosted	maven2	default	Online	copy		
maven-snaps...	hosted	maven2	default	Online	copy		
nuget-group	group	nuget	default	Online	copy		
nuget-hosted	hosted	nuget	default	Online	copy		
nuget.org-pro...	proxy	nuget	default	Online - Read...	copy	0 0	

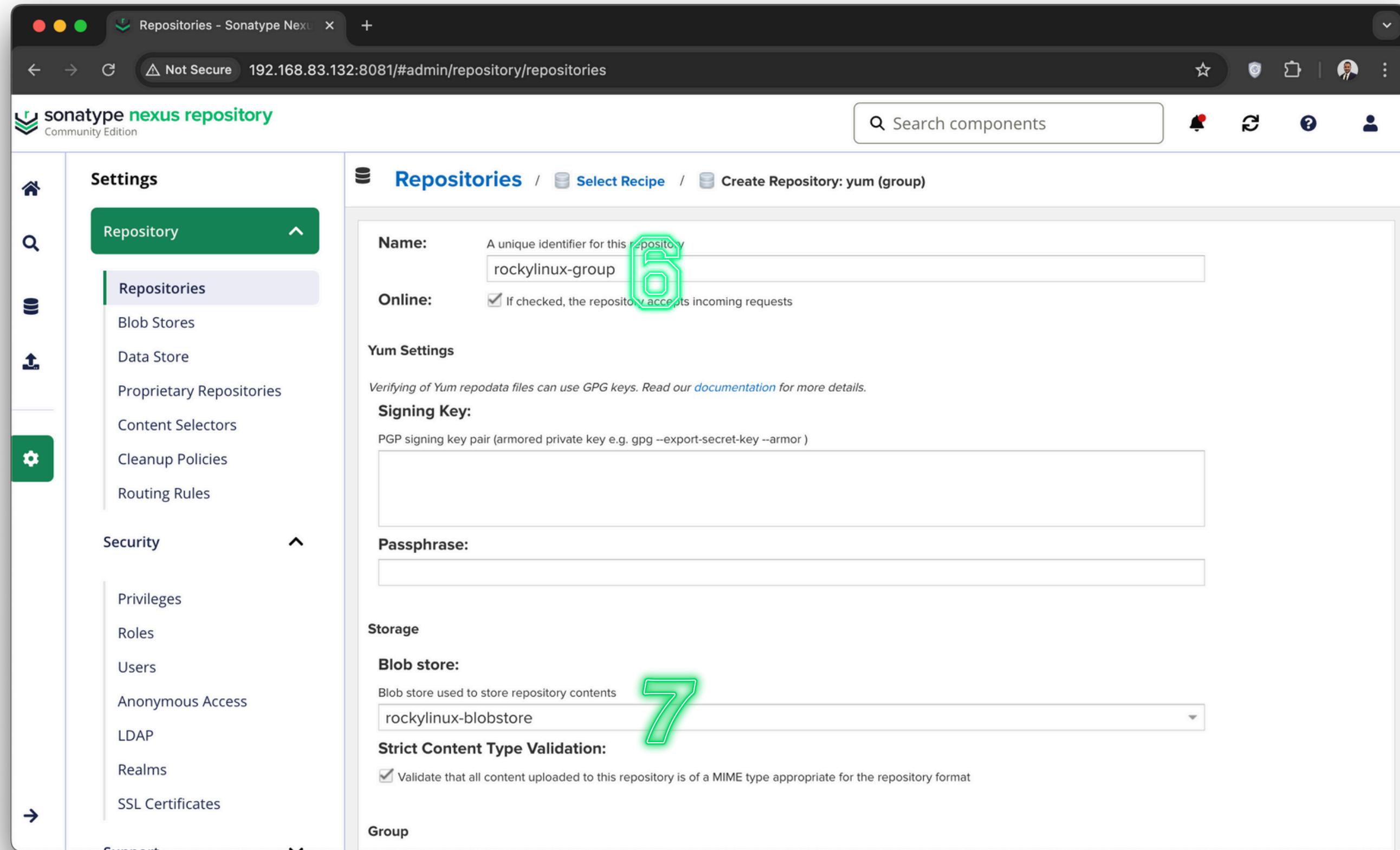
single-node standalone installation

Create a **Yum(group)** for better management

The screenshot shows the Sonatype Nexus Repository Manager interface. The left sidebar has a green header "Repository" and a sub-menu under "Repositories" with "Repositories" selected. The main content area is titled "Repositories / Select Recipe". A list of repository types is shown, with "yum (group)" highlighted in grey and a large green number "4" overlaid on it. Other listed repositories include npm (hosted), npm (proxy), nuget (group), nuget (hosted), nuget (proxy), p2 (proxy), pypi (group), pypi (hosted), pypi (proxy), r (group), r (hosted), r (proxy), raw (group), raw (hosted), raw (proxy), rubygems (group), rubygems (hosted), rubygems (proxy), yum (hosted), and yum (proxy). The top navigation bar shows the URL "192.168.83.132:8081/#admin/repository/repositories" and the status "Not Secure".

single-node standalone installation

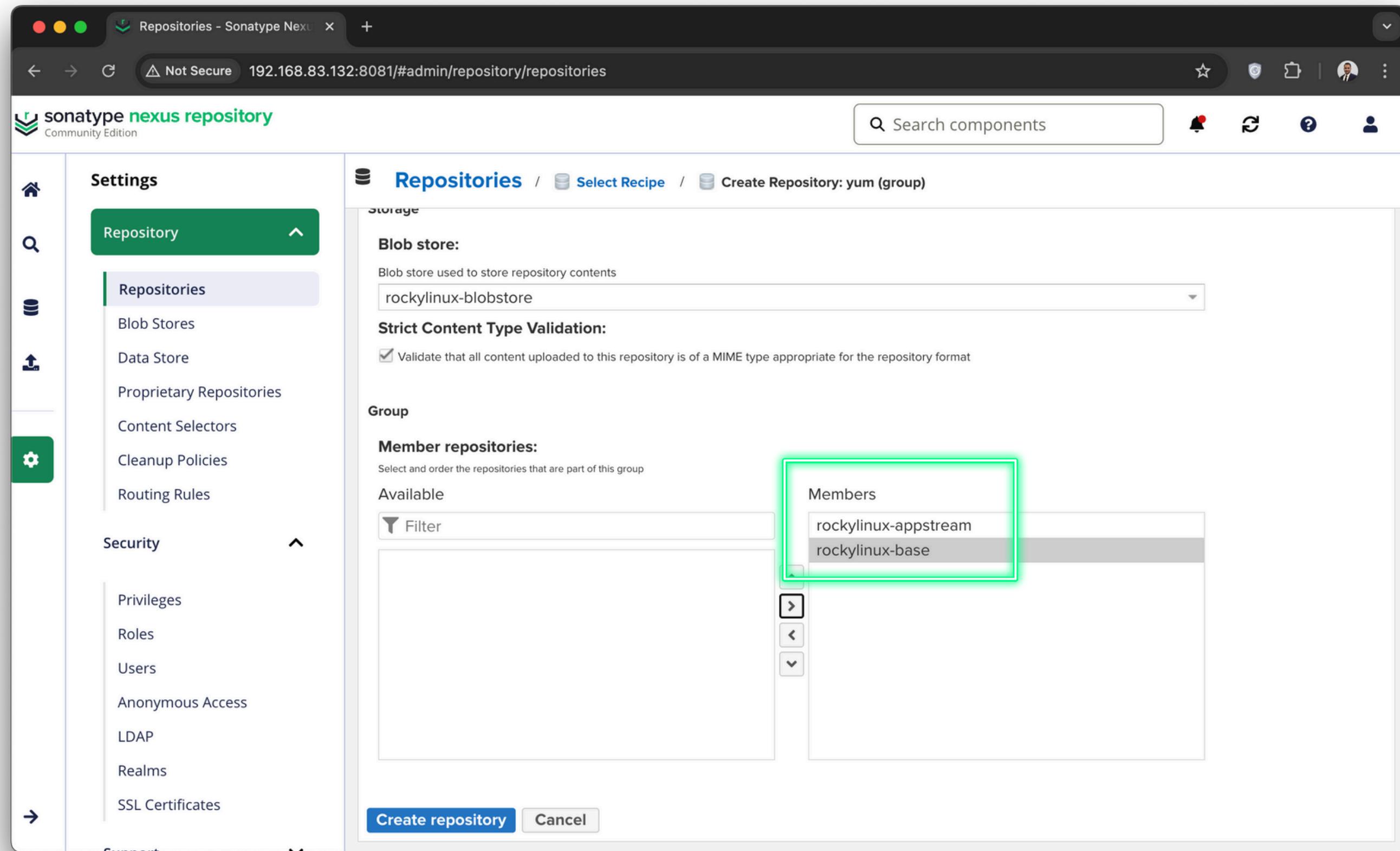
Create a **Yum(group)** for better management



The screenshot shows the Sonatype Nexus Repository Manager interface. On the left, a sidebar titled 'Settings' has 'Repository' selected. Under 'Repository', 'Repositories' is also selected. The main content area is titled 'Repositories / Select Recipe / Create Repository: yum (group)'. The 'Name' field contains 'rockylinux-group'. The 'Online' checkbox is checked. In the 'Yum Settings' section, there's a note about GPG keys and a 'Signing Key' input field. Below it is a 'Passphrase' input field. The 'Storage' section shows 'Blob store:' set to 'rockylinux-blobstore'. A 'Strict Content Type Validation' checkbox is checked. A large green number '6' is overlaid on the 'Name' field, and a large green number '7' is overlaid on the 'Blob store' dropdown.

single-node standalone installation

Create a **Yum(group)** for better management



single-node standalone installation

Create a **Yum(group)** for better management – Client Configuration

```
mehdiabdollahei - root@rockylinux-server:~ - ssh root@192.168.83.150 - 94x27
[ro...@rockylinux-server ~]# vim nexus.repo
[ro...@rockylinux-server ~]# cat nexus.repo
[rockylinux-group]
name=Rocky Linux BaseOS
baseurl=http://rockylinux:qazwsx@192.168.83.132:8081/repository/rockylinux-group/
enabled=1
gpgcheck=0
[ro...@rockylinux-server ~]# dnf install httpd
Last metadata expiration check: 0:10:45 ago on Tue 09 Sep 2025 05:53:14 AM EDT.
Package httpd-2.4.57-11.el9_4.1.aarch64 is already installed.
Dependencies resolved.

=====
      Package           Architecture   Version       Repository    Size
=====
Upgrading:
  httpd              aarch64        2.4.62-4.el9  rockylinux-appstream  44 k
  httpd-core         aarch64        2.4.62-4.el9  rockylinux-appstream  1.4 M
  httpd-filesystem  noarch        2.4.62-4.el9  rockylinux-appstream  12 k
  httpd-tools        aarch64        2.4.62-4.el9  rockylinux-appstream  77 k
  mod_lua            aarch64        2.4.62-4.el9  rockylinux-appstream  56 k

Transaction Summary
=====
Upgrade 5 Packages

Total download size: 1.5 M
Is this ok [y/N]:
```