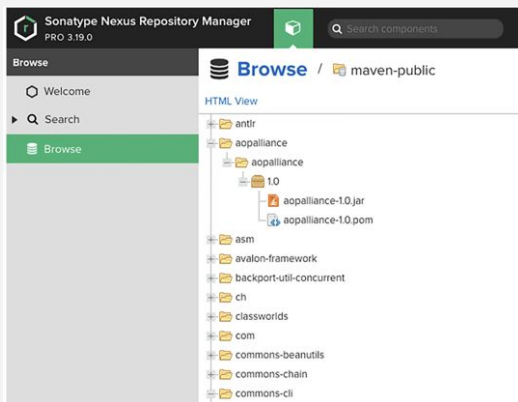


## GETTING STARTED WITH NEXUS REPOSITORY MANAGER

Get started using NXRM by completing the following:

1. [Download, unpack, and run NXRM 3 OSS.](#)
2. Update security defaults, sign in, sign out, and shut down the repository manager.
3. Configure a [Maven](#) proxy and [hosted](#) repository.
4. [Create a repository group for Maven](#) and combine local and cached components into the repository group.

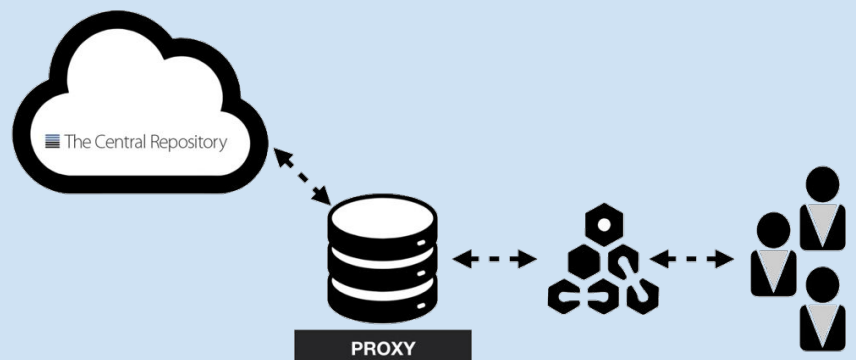


## MODERN FEATURES / CONTINUOUS INNOVATION

- Deploy directly to a desired repository
- Stage and manage releases
- Share binaries, snapshots and releases between groups of developers
- Post a collection of related, staged artifacts which can be easily tested, promoted, or discarded

## MAVEN / CENTRAL REPOSITORY

- Using Maven lets you centralize a location for your project's builds, reporting, and documentation.
- Maven relies on the Central Repository - a public repo for Java components.
- Java components are things like JARs, source and binary distributions, and WARs.
- Maven components are uniquely identified by a group ID, artifact, and version.

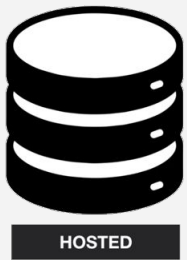


## MAVEN PROXY REPOSITORY

- Proxy repository is linked to **Central** Repository.
- When you search for components, if the request isn't found in your proxy repository, it is **forwarded to the Central Repository**. That component is then retrieved from Central and cached in the repository manager.
- If you search for that same component again, it **will** be found in local storage. This eliminates the need to go to the Central Repository and **reduces bandwidth and time** needed to retrieve the components you need.
- When components are stored in a caching proxy, a **copy of those components stays indefinitely**. In the event a component becomes unavailable in the Central Repository, you'll **still have access** to it in your proxy. This provides more control over the components you need to build your applications.



## MAVEN HOSTED REPOSITORY



- NXRM includes two hosted Maven repositories: **maven-releases** and **maven-snapshots**
- **Release** repositories - the place where your organization publishes internal releases. A release component is created by a specific, versioned release. These components are considered to be solid, stable and perpetual to guarantee the builds that use them are repeatable over time. You can also use this repository for third-party components that are not available in external repositories and can't be retrieved using a configured proxy repository.
- Snapshot components change over time, generated during active development. Continuous development is typically performed with snapshot versions.
- We recommend using the release repository for stable components in production and a snapshot repository for components still in the development phase.

## REPOSITORY GROUP

- Is its own access point, separate from proxy and hosted repositories.
- Created by adding repositories as members of an ordered list.
- A combined repository, of locally published and remotely cached components.
- A unique endpoint that users can access without having to retrieve components from hosted and proxy repositories.
- A repository group is a one-stop location where developers can collaborate on a project with all available components assigned to it. If you're an administrator, you can dynamically add repositories without updating developer settings. If you're a developer, you can access the contents of the group by sending requests directly to the repository manager.

