Nexus :

What is Artifactory in DevOps ?
In CI/CD process when we build our project we obtain an artifactory after a build.So further in deployment phase we deploy these artifactories on our production / pre-production server.

So now if we assume what if we don't use artifacts…It will be really very difficult over time.hat is why Artifacts are important to hold onto throughout the development process and also after that.

What is Nexus Repository Manager ?
Artifact Repository: Artifact repository is a location where you can store your all artifacts which are needed for the projects.

Nexus Repository Manager: It allows developer to collect, retrieve, manage our artifacts.

Basically Nexus Repository Manager helps us to host our repositories.

For eg- "Maven Central Repository" so we can use it to retrieve all dependencies needed for a Maven build.

Prerequisites
Open JDK 8
Minimum CPU's: 4
Ubuntu Server with User sudo privileges.
Set User limits
Web Browser
Firewall/Inbound port: 22, 8081
you can go through Nexus artifactory official page to know more about system requirement for Nexus.

update the system packages

sudo apt-get update
#1: Install OpenJDK 1.8 on Ubuntu 20.04 LTS
sudo apt install openjdk-8-jre-headless
#2: Download Nexus Repository Manager setup on Ubuntu 20.04 LTS
Download the latest Nexus Repository Manager Setup from official nexus page.

Navigate to /opt directory

cd /opt
Download the SonaType Nexus on Ubuntu using wget

sudo wget https://download.sonatype.com/nexus/3/latest-unix.tar.gz
#3: Install Nexus Repository on Ubuntu 20.04 LTS
Extract the Nexus repository setup in /opt directory

tar -zxvf latest-unix.tar.gz
Rename the extracted Nexus setup folder to nexus

sudo mv /opt/nexus-3.30.1-01 /opt/nexus
As security practice, not to run nexus service using root user, so lets create new user named nexus to run nexus service

sudo adduser nexus
To set no password for nexus user open the visudo file in ubuntu

sudo visudo
Add below line into it , save and exit

nexus ALL=(ALL) NOPASSWD: ALL
Give permission to nexus files and nexus directory to nexus user

sudo chown -R nexus:nexus /opt/nexus
sudo chown -R nexus:nexus /opt/sonatype-work
To run nexus as service at boot time, open /opt/nexus/bin/nexus.rc file, uncomment it and add nexus user as shown below

sudo nano /opt/nexus/bin/nexus.rc
run_as_user="nexus"
To Increase the nexus JVM heap size, open the /opt/nexus/bin/nexus.vmoptions file, you can modify the size as shown below

In the below settings, the directory is changed from ../sonatype-work to ./sonatype-work

-Xms1024m
-Xmx1024m
-XX:MaxDirectMemorySize=1024m

-XX:LogFile=./sonatype-work/nexus3/log/jvm.log
-XX:-OmitStackTraceInFastThrow
-Djava.net.preferIPv4Stack=true
-Dkaraf.home=.
-Dkaraf.base=.
-Dkaraf.etc=etc/karaf
-Djava.util.logging.config.file=/etc/karaf/java.util.logging.properties
-Dkaraf.data=./sonatype-work/nexus3
-Dkaraf.log=./sonatype-work/nexus3/log
-Djava.io.tmpdir=./sonatype-work/nexus3/tmp
#4: Run Nexus as a service using Systemd
To run nexus as service using Systemd

sudo nano /etc/systemd/system/nexus.service
paste the below lines into it.

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

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

[Install]
WantedBy=multi-user.target
To start nexus service using systemctl

sudo systemctl start nexus
To enable nexus service at system startup

sudo systemctl enable nexus
To check nexus service status

sudo systemctl status nexus
To stop Nexus service

sudo systemctl stop nexus
if the nexus service is not started, you can the nexus logs using below command

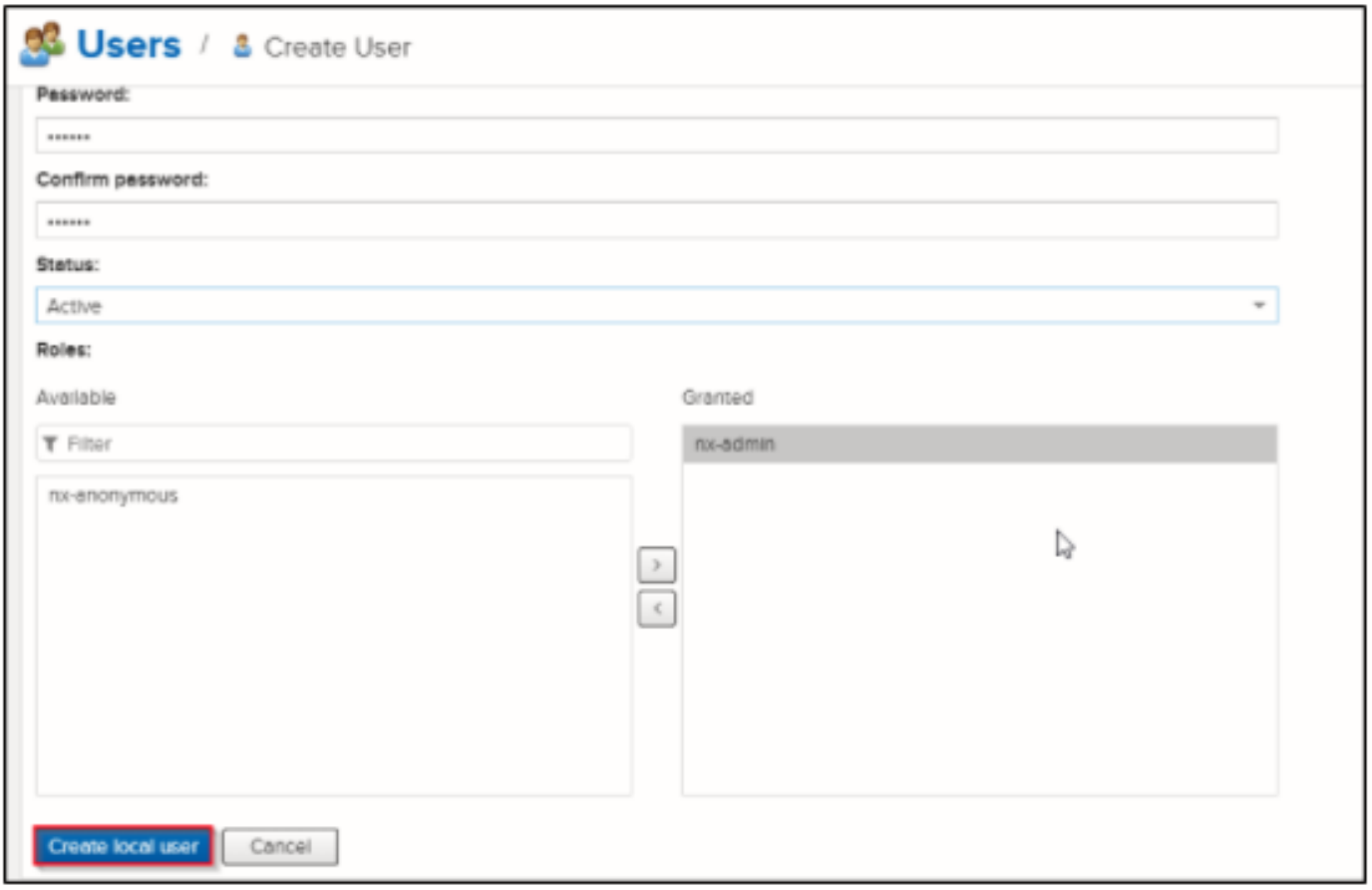
tail -f /opt/sonatype-work/nexus3/log/nexus.log
We have covered How to Install Nexus Repository on Ubuntu 20.04 LTS.

#5: Access Nexus Repository Web Interface
To access Nexus repository web interface , open your favorite browser

if you are running UFW firewall on Ubuntu, open the firewall port 8081 using below command

ufw allow 8081/tcp
http://server_IP:8081

CREATING USER
To create a user, click on the **Users** tab. Then fill out the details such as ID, first and last name, email, password, the status of the user (active/disabled), and the roles of that user. Then click on the **Create local user** button.



You can see the list of all the users under the **Users** tab.

Creating Repository
To create a repository, click on the **Repository** tab, and under the **Repositories** section, click the **Create repository** button.

Then you will have to choose the recipe of the repository. This decides what kind of file format you will be uploading and whether it's a hosted or a proxy repository.

Then fill out the details for the repository. For example, if you choose Advanced Packaging Tool (APT), you will have to fill in details such as name, APT distribution, and signing key. If you choose PyPI, you will have to fill in details such as name, blob details, and deployment policy.

After filling in all the details, click on the **Create repository** button.

Nexus format :

Format	Description	In-product or Community	HA-C
APK	Proxy Alpine OS apk packages and cache them in repository manager.		no
APT	Use Advanced Package Tool (APT) tools such as apt-get to access hosted Debian, Ubuntu and other Linux software packages.		no
Bower	Keep track of all your web site frontend development packages using Bower registries. Note that Bower format is not compatible with H2 or PostgreSQL databases. See Feature Availability for PostgreSQL and H2 Databases for more information.		yes
Cargo	Package repositories for the Rust package manager Cargo.		no
Chef	Provision Chef Cookbooks using tools like Knife and Berkshelf.		no
CocoaPods	Dependency manager for Swift and Objective-C Cocoa projects.		no
Composer	PHP Composer dependency managment for the PHP programming language.		no
Conan	Share your C/C++ packages in a central repository.		no
Conda	Proxy Conda packages for languages such as Python, R, Ruby, Lua, Scala, Java, JavaScript, C/C++ and FORTRAN.		no

Git LFS	Store large files such as audio samples, videos, datasets, and graphics inside repositories and use simple text pointers to these inside your Git project.		yes
Go	Modernize your development process and reuse your Go (golang) modules by sharing them in repositories.		no
Helm	Manage packages for Kubernetes by accessing Helm Charts in Helm repositories.		no
Maven	Leverage the most experienced Maven repository format product to host your private Java components and proxy defacto public repositories like Central using tooling such as Maven, Ant and Gradle.		yes
npm	Publish your javascript node.js projects to hosted registries and integrate dependencies from external JavaScript package registries.		yes
NuGet	Use NuGet client compatible tooling to push and install .Net packages. Development is easier using consolidated hosted and proxy registries. Note that NuGet v2 format is not compatible with H2 or PostgreSQL databases. See Feature Availability for PostgreSQL and H2 Databases for more information.		yes
p2	Proxy p2 format repositories for your Eclipse IDE and other Equinox based application dependencies.		no
Puppet	Securely serve your Puppet Modules in repository manager by proxying repositories like Puppet Forge.		no

Proxy Repository

A repository with the type *proxy*, also known as a proxy repository, is a repository that is linked to a remote repository. Any request for a component is verified against the local content of the proxy repository. If no local component is found, the request is forwarded to the remote repository. The component is then retrieved and stored locally in the repository manager, which acts as a cache. Subsequent requests for the same component are then fulfilled from the local storage, therefore eliminating the network bandwidth and time overhead of retrieving the component from the remote repository again.

By default, the repository manager ships with the following configured proxy repositories:

maven-central

> This proxy repository accesses the Central Repository, formerly known as Maven Central. It is the default component repository built into Apache Maven and is well-supported by other build tools like Gradle, SBT or Ant/Ivy.

nuget.org-proxy

> This proxy repository accesses the NuGet Gallery. It is the default component repository used by the nuget package management tool used for .Net development.

Hosted Repository

A repository with the type *hosted*, also known as a hosted repository, is a repository that stores components in the repository manager as the authoritative location for these components.

By default, the repository manager ships with the following configured hosted repositories:

maven-releases

This hosted repository uses the *maven2* repository format with a release version policy. It is intended to be the repository where your organization publishes internal releases. You can also use this repository for third-party components that are not available in external repositories and can therefore not be retrieved via a configured proxy repository. Examples of these components could be commercial, proprietary libraries such as an Oracle JDBC driver that may be referenced by your organization.

maven-snapshots

This hosted repository uses the *maven2* repository format with a snapshot version policy. It is intended to be the repository where your organization publishes internal development versions, also known as snapshots.

nuget-hosted

This hosted repository is where your organization can publish internal releases in repository using the *nuget* repository format. You can also use this repository for third-party components that are not available in external repositories, that could potentially be proxied to gain access to the components.

Repository Group

A repository with the type *group*, also known as repository group, represents a powerful feature of Nexus Repository Manager. They allow you to combine multiple repositories and other repository groups in a single repository. This in turn means that your users can rely on a single URL for their configuration needs, while the administrators can add more repositories and therefore components to the repository group.

The repository manager ships with the following groups:

maven-public

The maven-public group is a repository group of *maven2* formatted repositories and combines the important external proxy repository for the Central Repository with the hosted repositories *maven-releases* and *maven-snapshots*. This allows you to expose the components of the Central Repository as well as your internal components in one single, simple-to-use repository and therefore URL.

nuget-group

This group combines the *nuget* formatted repositories *nuget-hosted* and *nuget.org-proxy* into a single repository for your .Net development with NuGet.

Managing Repositories and Repository Groups

The administration user interface for repositories and repository groups is available via the *Repositories* item in the *Repository* sub menu of the *Administration* menu. It allows you to create and configure repositories as well as delete them and perform various maintenance operations. To access this section, the user must have the *nx-all* or *nx-repository-admin* privileges. The initial view displayed in *Figure: "List of Repositories"* features a list of all configured repositories and repository groups.

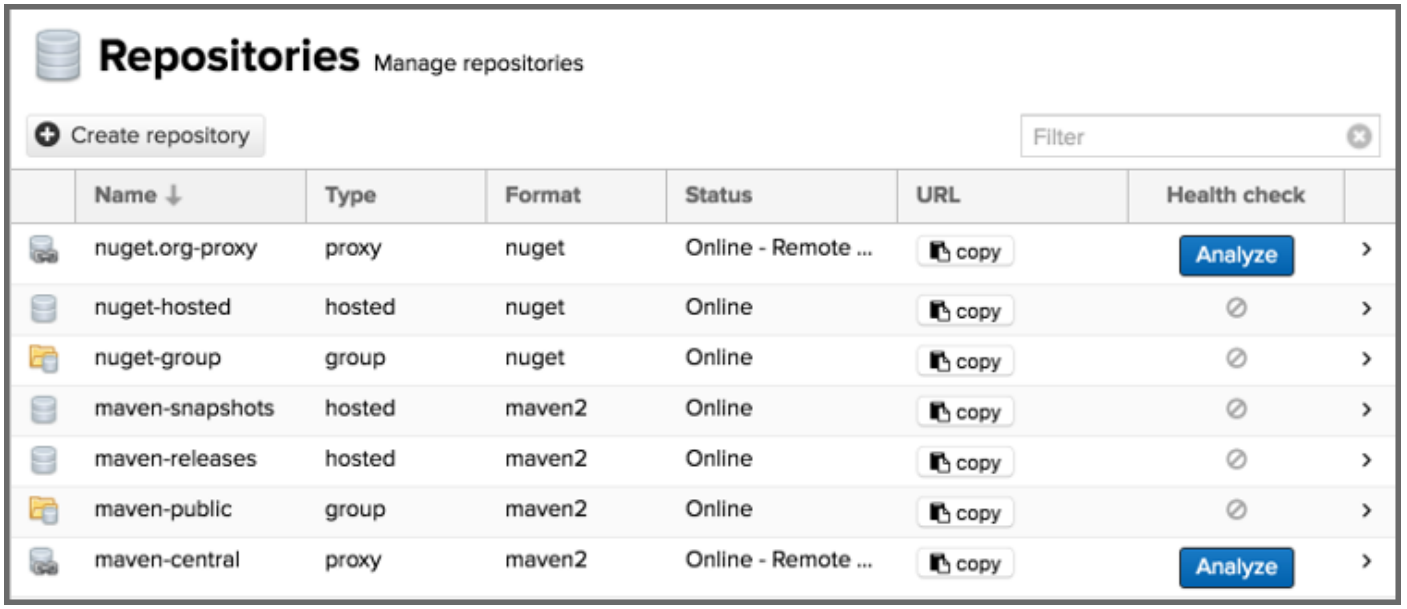


Figure: List of Repositories

The list of repositories displays some information for each repository in the following columns:

Name

the unique name of the repository or repository group

Type

the type of the repository with values of *proxy* or *hosted* for repositories or *group* for a repository group

Format

the repository format used for the storage in the repository with values such as *maven2*, *nuget* or others

Status

the status of the repository as well as further information about the status. A functioning repository would show the status to be *Online*. Additional information can e.g., be about SSL certification problems or the status of the remote repository for a currently disabled proxy repository.

URL

the *copy* button prompts a dialog containing a direct URL path exposing the repository

Health Check

displays the repository health statistics from a previously run Repository Health Check or a button to start the analysis