



Artifact Repository Manager with Nexus

Key Takeaways

Introduction to Artifact Repository Manager

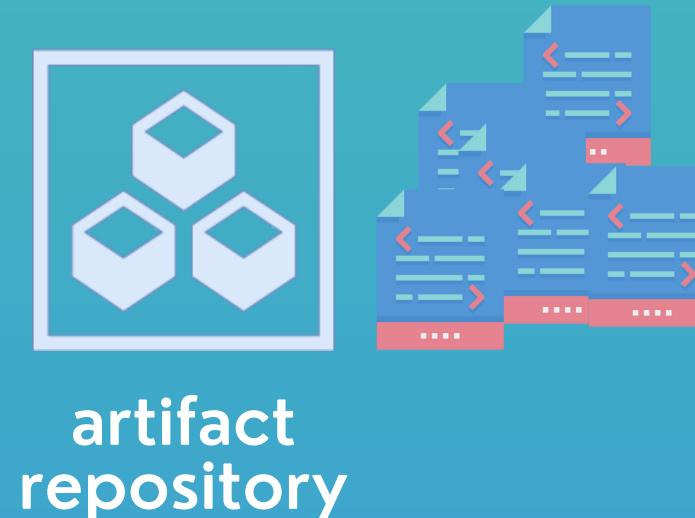
What is an Artifact Repository?

- **Storage of build artifacts** produced by continuous integration and makes them available for automated deployment to different deployment environments
- Provides a **central location**
- Artifacts are applications built into a **single file**
- There are different artifact **formats**:

JAR, WAR, ZIP, TAR,etc.

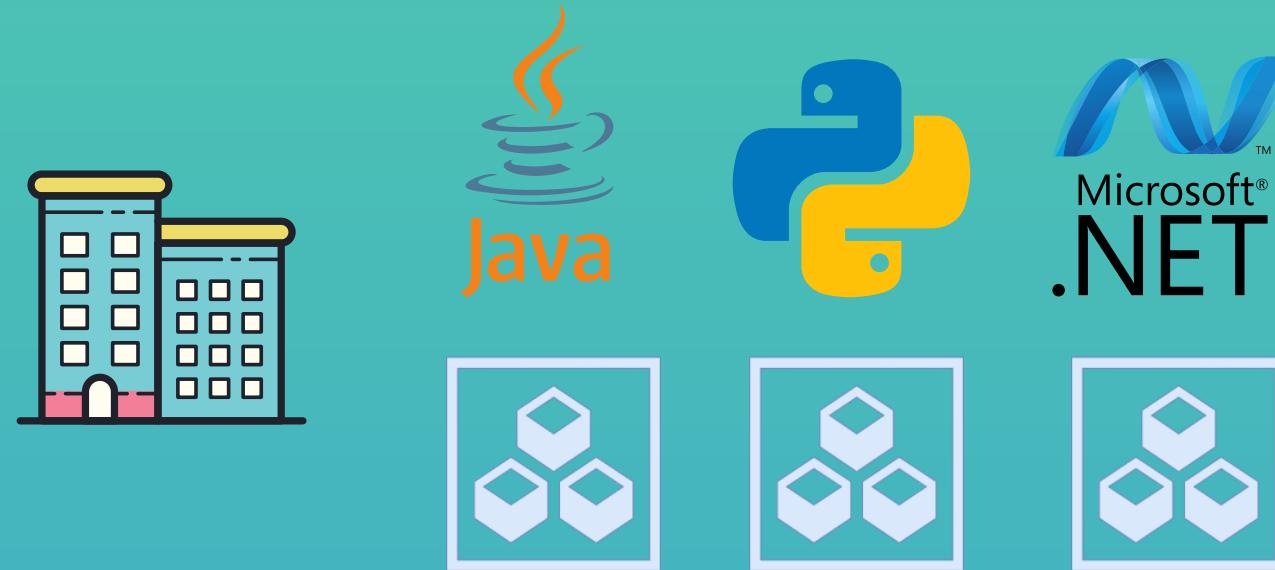


- Artifact repository needs to **support this specific format**
- Repository for each file/artifact type



What is an Artifact Repository Manager?

- Can **store many different artifact types**
- Which is great, because as a company you often produce different types of artifacts



Instead of **having different repositories for each artifact type**

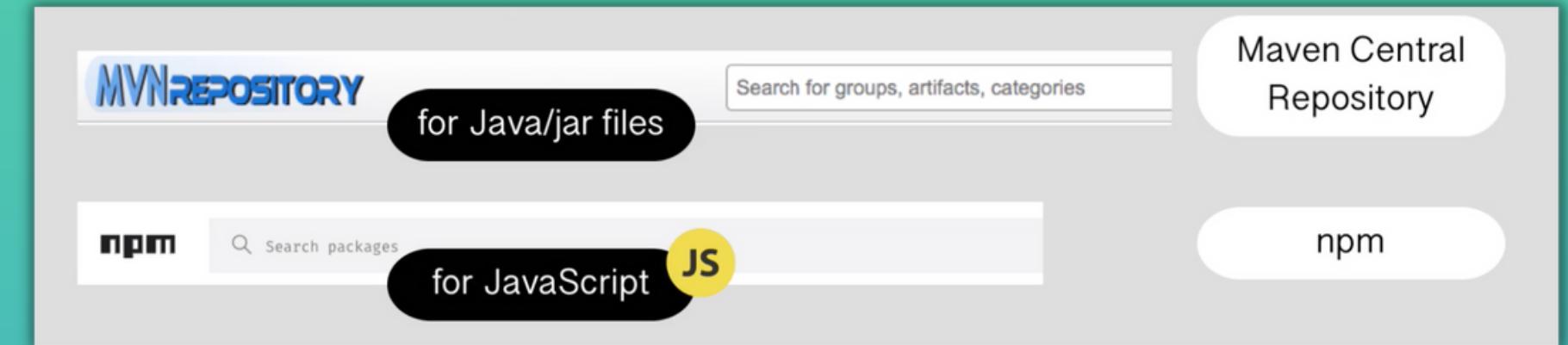


Just 1 repository for managing all your different artifact types

Public vs Private Artifact Repository Managers

Public Repository Managers

- There are public repository managers
- For example libraries/frameworks you use as a dependency
- You can make own project publically available there:



Public

Nexus - one of most popular repository manager

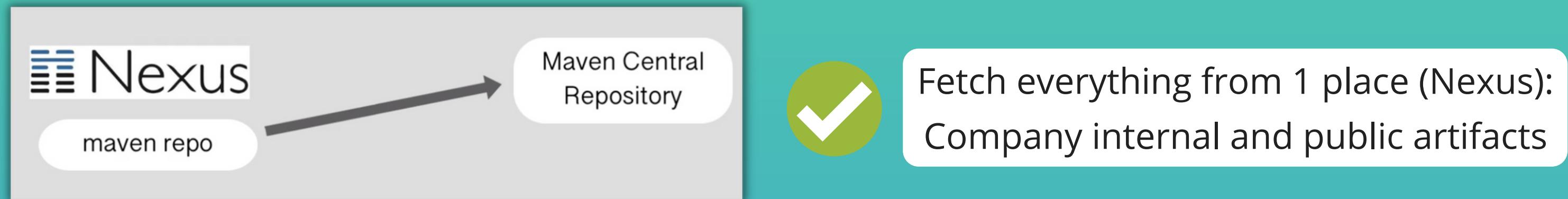
- It's a private central store
- You can **upload** and **store** different built artifacts
- **Retrieve** (download) artifacts later
- For internal usage in your company



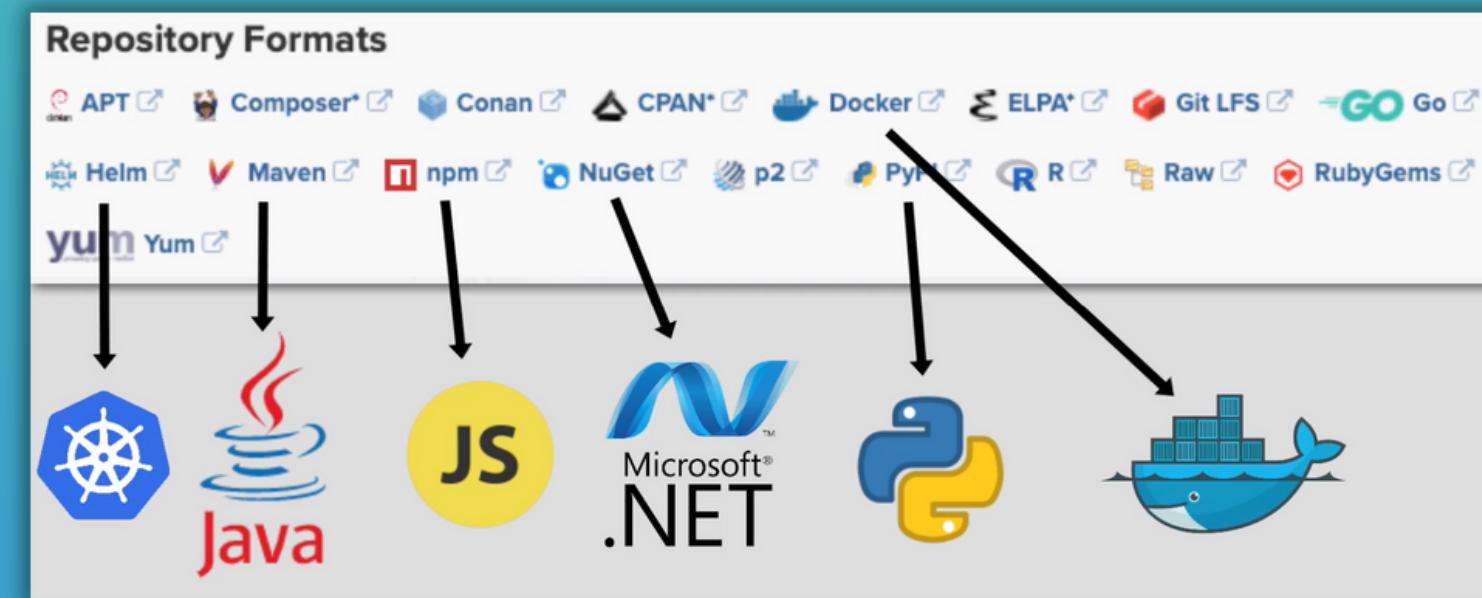
Private

Nexus - Artifact Repository Manager

- Has open source and commercial version
- Host own repositories
- Proxy repositories (intermediary to another repository):



- You can have multiple repositories for different formats or different teams in your company
- **Supported formats** in Nexus:



Features of Repository Manager

- thumb up icon Integration with LDAP
- thumb up icon Flexible and powerful REST API for integration with other tools

Important, because you need to integrate it into your CI/CD Pipeline



- thumb up icon Backup and Restore
- thumb up icon Multi-Format Support (different file types - zip, tar, docker etc)
- thumb up icon Metadata Tagging (labelling and tagging artifacts)
- thumb up icon Cleanup Policies (automatically delete files that match condition)
- thumb up icon Search functionality (across projects, artifact repos etc.)
- thumb up icon User token support for system user authentication

Install and Run Nexus on Cloud Server

Install and Run Nexus on Cloud Server - 1

Summary of Steps - 1st Part:

1. **Create Ubuntu Server (Droplet)** - min 4GB RAM & 2 CPUs
2. **Open SSH port 22**
3. **Install Java 8**
4. **Download and Install Nexus**
5. **Create "nexus" user and group** - Best Practice: Run applications with own user

Install and Run Nexus on Cloud Server - 2

With Nexus installation you get:

```
Bootcamp — root@ubuntu-s-1vcpu-1gb-fra1-01: /opt — ssh root@46.101.157.191 — 102x26
~/Demo-projects/Bootcamp — root@ubuntu-s-1vcpu-1gb-fra1-01: /opt — ssh root@46.101.157.191 ..react-example
root@ubuntu-s-1vcpu-1gb-fra1-01:/opt# ls
latest-unix.tar.gz  nexus-3.28.1-01  sonatype-work
root@ubuntu-s-1vcpu-1gb-fra1-01:/opt# ls nexus-3.28.1-01/
NOTICE.txt  OSS-LICENSE.txt  PRO-LICENSE.txt  bin  deploy  etc  lib  public  system
root@ubuntu-s-1vcpu-1gb-fra1-01:/opt# ls sonatype-work/nexus3/
clean_cache  log  orient  tmp
root@ubuntu-s-1vcpu-1gb-fra1-01:/opt#
```

Sonatype-work:

contains own config for Nexus and data

- subdirectories depending on your Nexus configuration
- IP address that accessed Nexus
- Logs of Nexus App
- Your uploaded files and metadata
- You can use this folder for backup



Nexus folder:

contains runtime and application of Nexus

Install and Run Nexus on Cloud Server - 3

Summary of Steps - 2nd Part:

1. Make Nexus own the folders
2. Start with nexus user
3. Open port 8081 with firewall
4. Access from browser

Starting Nexus...

- Services should not run with root user permissions
- So best practice: create own user for service (like in this case nexus)
- Should have only the permission for that specific service



Install and Run Nexus on Cloud Server - 4

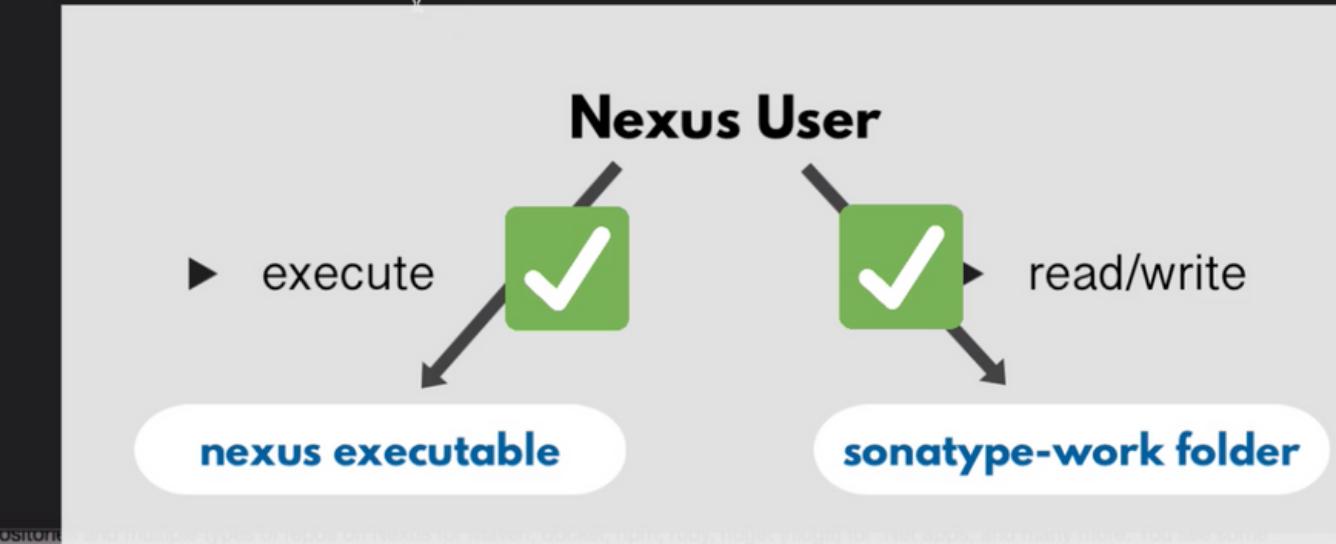
Run Nexus...

1. Make Nexus own the folders
2. Start with nexus user



```
Bootcamp — root@ubuntu-s-1vcpu-1gb-fra1-01:/opt — ssh root@46.101.157.191 — 102x26
~/Demo-projects/Bootcamp — root@ubuntu-s-1vcpu-1gb-fra1-01:/opt — ssh root@46.101.157.191 ..react-example
root@ubuntu-s-1vcpu-1gb-fra1-01:/opt# ls -l
total 162600
-rw-r--r-- 1 root root 166494026 Oct 19 14:35 latest-unix.tar.gz
drwxr-xr-x 9 root root 4096 Oct 25 09:06 nexus-3.28.1-01
drwxr-xr-x 3 root root 4096 Oct 25 09:06 sonatype-work
root@ubuntu-s-1vcpu-1gb-fra1-01:/opt# chown -R nexus:nexus nexus-3.28.1-01
root@ubuntu-s-1vcpu-1gb-fra1-01:/opt# chown -R nexus:nexus sonatype-work
root@ubuntu-s-1vcpu-1gb-fra1-01:/opt# ls -l
total 162600
-rw-r--r-- 1 root root 166494026 Oct 19 14:35 latest-unix.tar.gz
drwxr-xr-x 9 nexus nexus 4096 Oct 25 09:06 nexus-3.28.1-01
drwxr-xr-x 3 nexus nexus 4096 Oct 25 09:06 sonatype-work
root@ubuntu-s-1vcpu-1gb-fra1-01:/opt#
```

```
Bootcamp — nexus@ubuntu-s-4vcpu-8gb-fra1-01: ~ — ssh root@159.89.109.182 — 102x26
~/Demo-projects/Bootcamp — nexus@ubuntu-s-4vcpu-8gb-fra1-01: ~ — ssh root@159.89.109.182 ..react-example
nexus@ubuntu-s-4vcpu-8gb-fra1-01:~$ /opt/nexus-3.28.1-01/bin/nexus start
Starting nexus
nexus@ubuntu-s-4vcpu-8gb-fra1-01:~$ ps aux | grep nexus
root      14850  0.0  0.0 10104 3768 pts/0    S    12:38   0:00 su - nexus
nexus     14851  0.0  0.0 10056 5032 pts/0    S    12:38   0:00 -bash
nexus     15531 281 11.5 6353596 943024 pts/0    Sl   12:39   0:31 /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java -server -Dinstall4j.jvmDir=/usr/lib/jvm/java-8-openjdk-amd64/jre -Dexec.moduleName=/opt/nexus-3.28.1-01/bin/nexus -XX:+UnlockDiagnosticVMOptions -Dinstall4j.launcherId=245 -Dinstall4j.swt=false -Djava.jv=0 -Djava.jv=0 -Djava.jv=0 -Djava.jv=0 -Xms2703m -Xmx2703m -XX:MaxDirectMemorySize=2703m -XX:+UnlockDiagnosticVMOptions -XX:+LogVMOutput -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 -Dkaraf.startLocalConsole=false -Djava.endorsed.dirs=lib/endorsed -Di4j.vpt=true -classpath /opt/nexus-3.28.1-01/install4j/i4jruntime.jar:/opt/nexus-3.28.1-01/lib/boot/nexus-main.jar:/opt/nexus-3.28.1-01/lib/boot/activation-1.1.1.jar:/opt/nexus-3.28.1-01/lib/boot/jakarta.xml.bind-api-2.3.3.jar:/opt/nexus-3.28.1-01/lib/boot/jaxb-runtime-2.3.3.jar:/opt/nexus-3.28.1-01/lib/boot/txw2-2.3.3.jar:/opt/nexus-3.28.1-01/lib/boot/istack-commons-runtime-3.0.10.jar:/opt/nexus-3.28.1-01/lib/boot/org.apache.karaf.main-4.2.9.jar:/opt/nexus-3.28.1-01/lib/boot/osgi.core-6.0.0.jar:/opt/nexus-3.28.1-01/lib/boot/org.apache.karaf.specs.activator-4.2.9.jar:/opt/nexus-3.28.1-01/lib/boot/org.apache.karaf.diagnostic.boot-4.2.9.jar:/opt/nexus-3.28.1-01/lib/boot/org.apache.karaf.jaas.boot-4.2.9.jar com.install4j.runtime.launcher.UnixLauncher start 9d17dc87 0 0 org.sonatype.nexus.karaf.NexusMain
```



Install and Run Nexus on Cloud Server - 5

Run Nexus...

3. Open port 8081 with Firewall



4. Access from browser

Sonatype Nexus Repository Manager OSS 3.28.1-01

Welcome - Nexus Repository Manager

Secure | 159.89.109.182:8081

Browse

Welcome

Search

Browse

Get Started

Configuration Set things up properly

Documentation Visit our help site

Community Ask and answer questions

Repository Formats

```
r:/opt/nexus-3.28.1-01/lib/boot/osgi.cs.activator-4.2.9.jar:/opt/nexus-3.28.1-01/lib/boot/org.apache.karaf.start 9d17dc87 0 0 org.sonatype.nexus 15619 0.0 0.0 11476 352 nexus 15620 0.0 0.0 8144 81 nexus@ubuntu-s-4vcpu-8gb-fra1-01:~$ netstat -an | grep :8081 (Not all processes could be identified. The following protocol addresses will not be shown, you would have to Active Internet connections (only servers) Proto Recv-Q Send-Q Local Address          tcp        0      0 127.0.0.1:36401      tcp        0      0 127.0.0.53:53      tcp        0      0 0.0.0.0:22      tcp6       0      0 :::22 nexus@ubuntu-s-4vcpu-8gb-fra1-01:~$ netstat -an | grep :8081 (Not all processes could be identified. The following protocol addresses will not be shown, you would have to Active Internet connections (only servers) Proto Recv-Q Send-Q Local Address          tcp        0      0 0.0.0.0:8081      tcp        0      0 127.0.0.1:36401
```

Access Nexus from Browser



Open port 8081 !

Overview of Nexus

Login & Default Repositories

Login

- With default "admin" user that automatically gets created to administer Nexus



Screenshot of the Sonatype Nexus Repository Manager web interface. The URL is `167.99.248.163:8081/#admin/repository`. The title bar says "Repository - Nexus Repository". The top navigation bar includes "Sonatype Nexus Repository Manager OSS 3.28.1-01", a gear icon, a search bar ("Search components"), and user information ("admin"). The main menu is "Administration" with a sub-menu "Repository" expanded. Under "Repository", there are links for "Repositories", "Blob Stores", "Content Selectors", "Cleanup Policies", and "Routing Rules". On the right side, there are buttons for "Blob Stores", "Cleanup Policies", "Repositories", "Routing Rules", and "Content Selectors". The status bar at the bottom shows "Not Secure | 167.99.248.163:8081/#admin/repository".

Screenshot of the Sonatype Nexus Repository Manager web interface. The URL is `167.99.248.163:8081/#admin/repository/repositories`. The title bar says "Repositories - Nexus Repository". The top navigation bar includes "Sonatype Nexus Repository Manager OSS 3.28.1-01", a gear icon, a search bar ("Search components"), and user information ("admin"). The main menu is "Administration" with a sub-menu "Repository" expanded. Under "Repository", there are links for "Repositories", "Blob Stores", "Content Selectors", "Cleanup Policies", "Routing Rules", "Security", "Privileges", and "Roles". The main content area is titled "Repositories Manage repositories" and contains a table with the following data:

Name	Type	Format	Status	URL	Health check	IQ Policy Vi...
maven-central	proxy	maven2	Online - Ready to Connect		Analyze	
maven-public	group	maven2	Online			
maven-releases	hosted	maven2	Online			
maven-snapshots	hosted	maven2	Online			
nuget-group	group	nuget	Online			
nuget-hosted	hosted	nuget	Online			
nuget.org-proxy	proxy	nuget	Online - Ready to Connect		Analyze	

Default Repositories

- You get default repositories with different TYPES when you install Nexus

Repository Types & Formats

Repository Types

- **Proxy:** intermediary to another repository
- **Group:** containing a group of repositories
- **Hosted:** for hosting company-internal artifacts



Each technology (python, java, docker) has its artifact type (jar, image), which can be saved in its own artifact repository format

Name ↑	Type	Format
maven-central	proxy	maven2
maven-public	group	maven2
maven-releases	hosted	maven2
maven-snapshots	hosted	maven2
nuget-group	group	nuget
nuget-hosted	hosted	nuget
nuget.org-proxy	proxy	nuget

Repository Formats

- **maven2:** for java jar artifacts
- **nuget:** for .NET packages
- **docker:** for Docker images

Publish artifact to repository

You can push or "PUBLISH" artifacts to Nexus with different tools

- Using built-in commands of build tools, like Maven or Gradle
- For that, you need to **configure** each build tool **with Nexus credentials and address**

X X X X X



Nexus REST API

You can also **interact with Nexus using its REST API** to do multiple things:

- query for different information:
- download an artifact
- upload an artifact

[list of repositories](#)

[list of components](#)

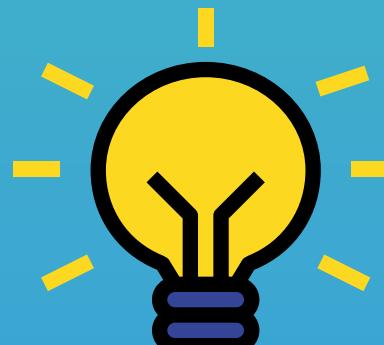
...

How to interact?

- Using a tool like curl or wget to execute http request
- You need to provide user and credential of a Nexus user

```
[\W]$ curl -u user:pwd -X GET 'http://167.99.248.163:8081/service/rest/v1/repositories'
```

```
[\W]$ curl -u nana:Bibilo123 -X GET 'http://167.99.248.163:8081/service/rest/v1/components?repository=maven-snapshots'
```



To use Nexus API or any API:
You should **always reference the documentation**. Not learn by heart,
because 1) API changes 2) there are a lot of endpoints, so you can't learn
that much

Component vs Asset

Components

- **abstract**
- what we are uploading
- term "component" refers to any type or format



Assets

- **actual physical packages/files**
- 1 component = 1 or more assets

Cleanup Policies & Scheduled Tasks

Cleanup Policies

- Clean up old artifacts to make space. You can define logic like:



Delete all artifacts older than 30 days



Delete all artifacts, which haven't been used/downloaded for more than 10 days

- You can create own cleanup policy for each repository



Scheduled Tasks

- Nexus allows you to schedule tasks that will be applied to all or a specific repository on a configurable schedule