Course Prologue

In this course, you will have a glimpse on the concepts of Nexus Artifact Repository.

- ***Introduction to Nexus Repository***
- ***Installation and Configuration of Nexus***
- ***User Interface Tour***
- ***Integration of Jenkins with Nexus***
- ***Demo on integration of Nexus***

What is Repository?

A Base place where an organization's aggregated data is stored and maintained in an organized way, for easy retrieval.

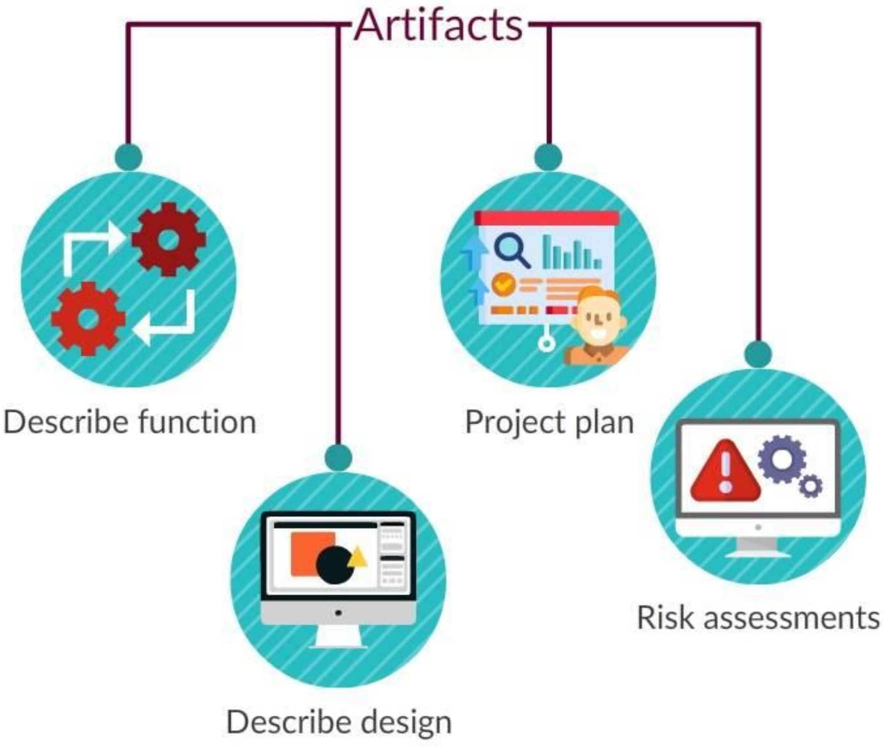
Easy distribution over the network.

Direct accessibility when needed without having to travel across the network.

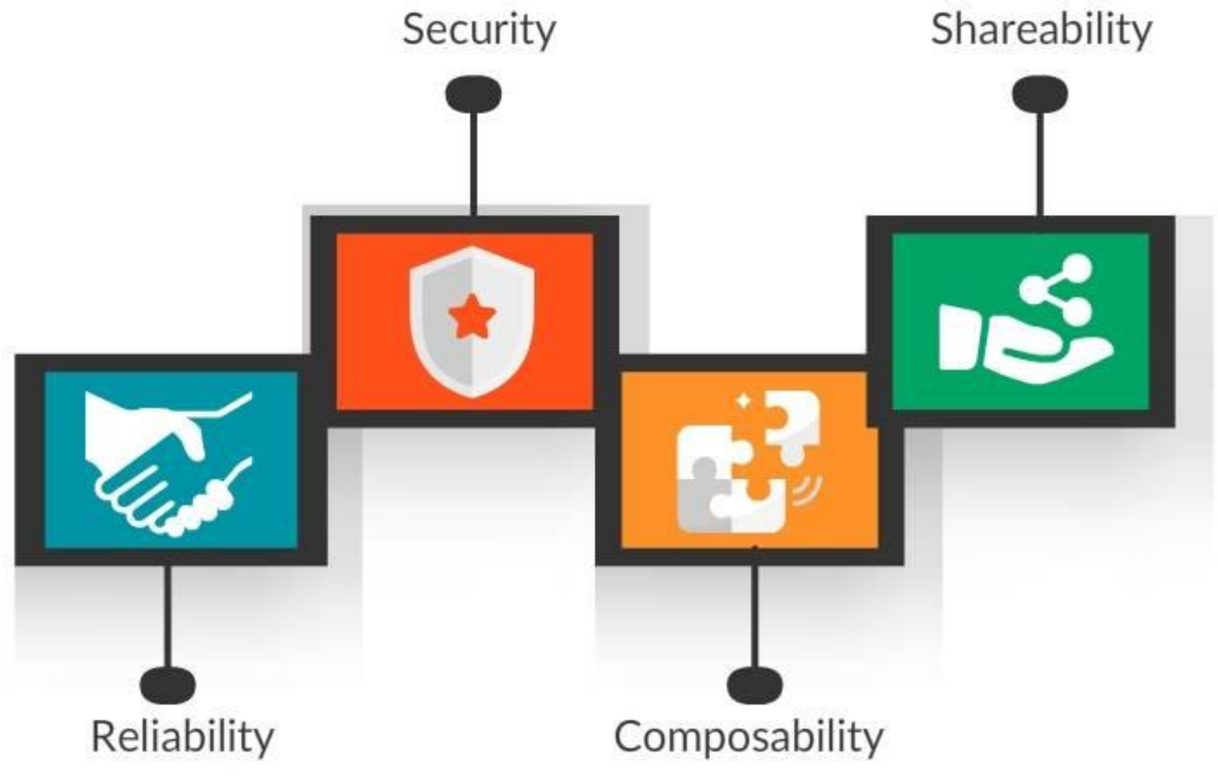
Repository Manager

A repository manager allows to store and retrieve build artifacts.

"Repository" is derived from the Latin word **Repositorium** meaning a vessel or chamber in which things can be placed.



- Artifacts are the End Products of the development lifecycle.
- Artifacts are used in different contexts like
  - Describes function
  - Describes design
  - Project plans
  - Risk assessments.



The advantages of artifacts are:

- Reliability
- Composability
- Security
- Shareability

Nexus Artifact Repository

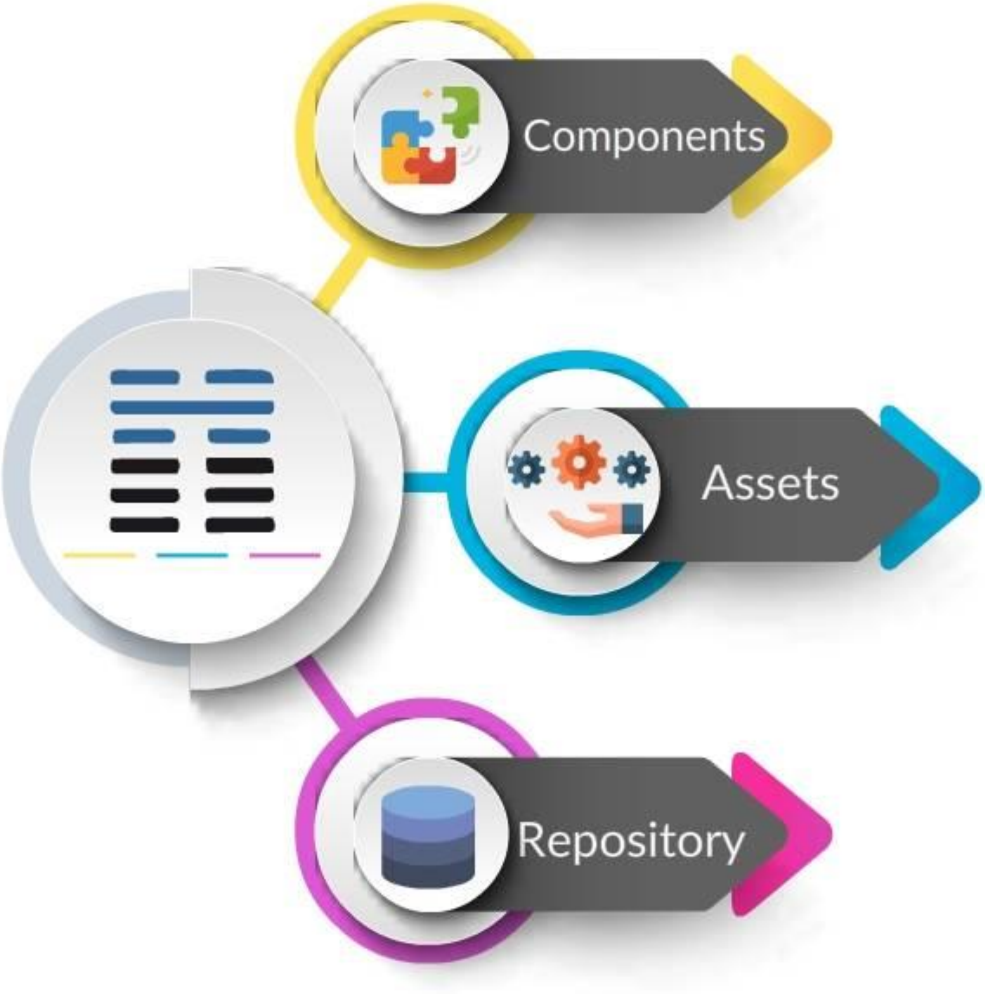
Nexus Artifact Repository is a Repository Manager that helps to proxy, collect and manage dependencies.

The two main advantages are:

Easy distribution of Software

Collaboration with other developers made easy.

Basic Elements of Nexus



The Basic Elements of Nexus are listed below as portrayed in the picture.

- Component

- Assets
- Repository

Components

- A Component resource is embraced as a part of our application at run-time, testing time or as a part of the build process.
- It can be an image (static resource) or an entire application.

Components are the archives of different files namely:

- Java bytecode in class files
- C object files
- Text files (For example: PDF, music files)

Component Archives Format

As we have seen earlier, components are the archives of different files. Archives are supported in different formats like:

Java JAR, WAR, EAR formats

Plain ZIP or .tar.gz files

Other package formats such as NuGet packages, Ruby gems, NPM packages

Executable formats such as .exe or .sh files, Android APK files, various installer formats.

Storage of Components

Artifacts are stored in the form coordinate structure namely:

- GroupId - Group identifier groups the set of components into one logical group.
- ArtifactId - Identifier for a single software component.
- Version - Version must follow established semantic versioning

It is called GAV Coordinate.

Confused about what is JAR? A JAR file is a product that is generated after the build stage of the project in the open source tool like Jenkins.

To know more about Jenkins, refer to the course Jenkins.

Assets

- The actual archive file is an asset associated with the component.
- One-to-many relationship with the component. A Component can have numerous assets associated with it.

Example:

JAR component of Maven contains:

- POM file
- JAR files of project

Both of which constitute as individual assets to the same component.

Components can be called as an artifact, package, bundle, archive and other terms in different tools.

Central Repository

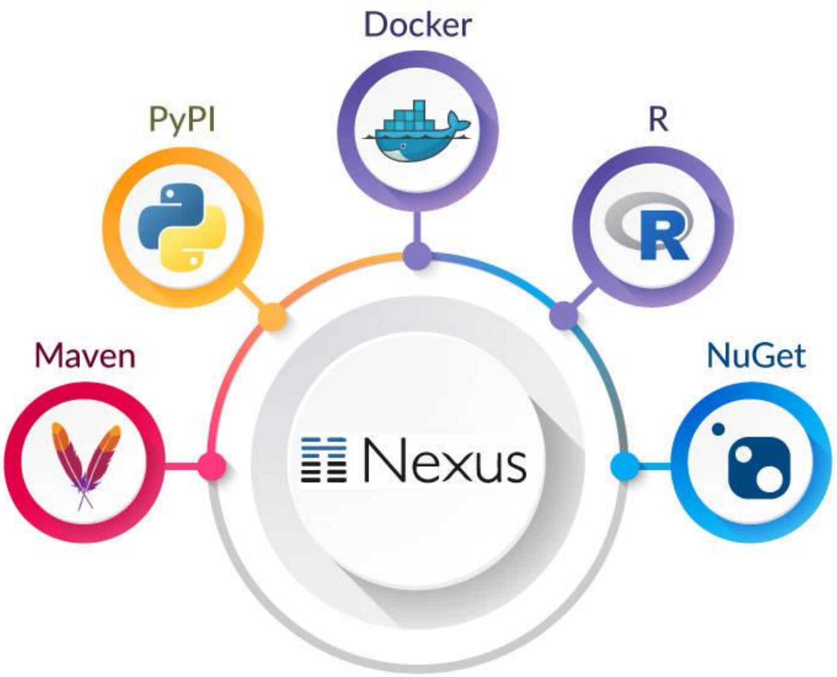
- A collection of Components.

Examples:

- Maven Central
- The NuGet Gallery
- RubyGems.org
- npmjs.org

Numerous tools access components in these repositories:

- Other package formats such as NuGet packages, Ruby gems, NPM packages
- Executable formats such as .exe or .sh files, Android APK files, various installer formats."



Repository Management

Repository Manager

1. Manages all the repositories that the team of developers can interact with, supported by the dedicated server.
2. Enable greater *collaboration* between developers.
3. Wider *distribution of software*, by facilitating the exchange and usage of binary components.

Core Features of the Repository Manager:

- ***Proxy***: It helps us in proxying in components located in remote which in turn saves the factors of bandwidth and time.
- ***Host***: Hosts a repository by providing a deployment target of internal software components to the organization.

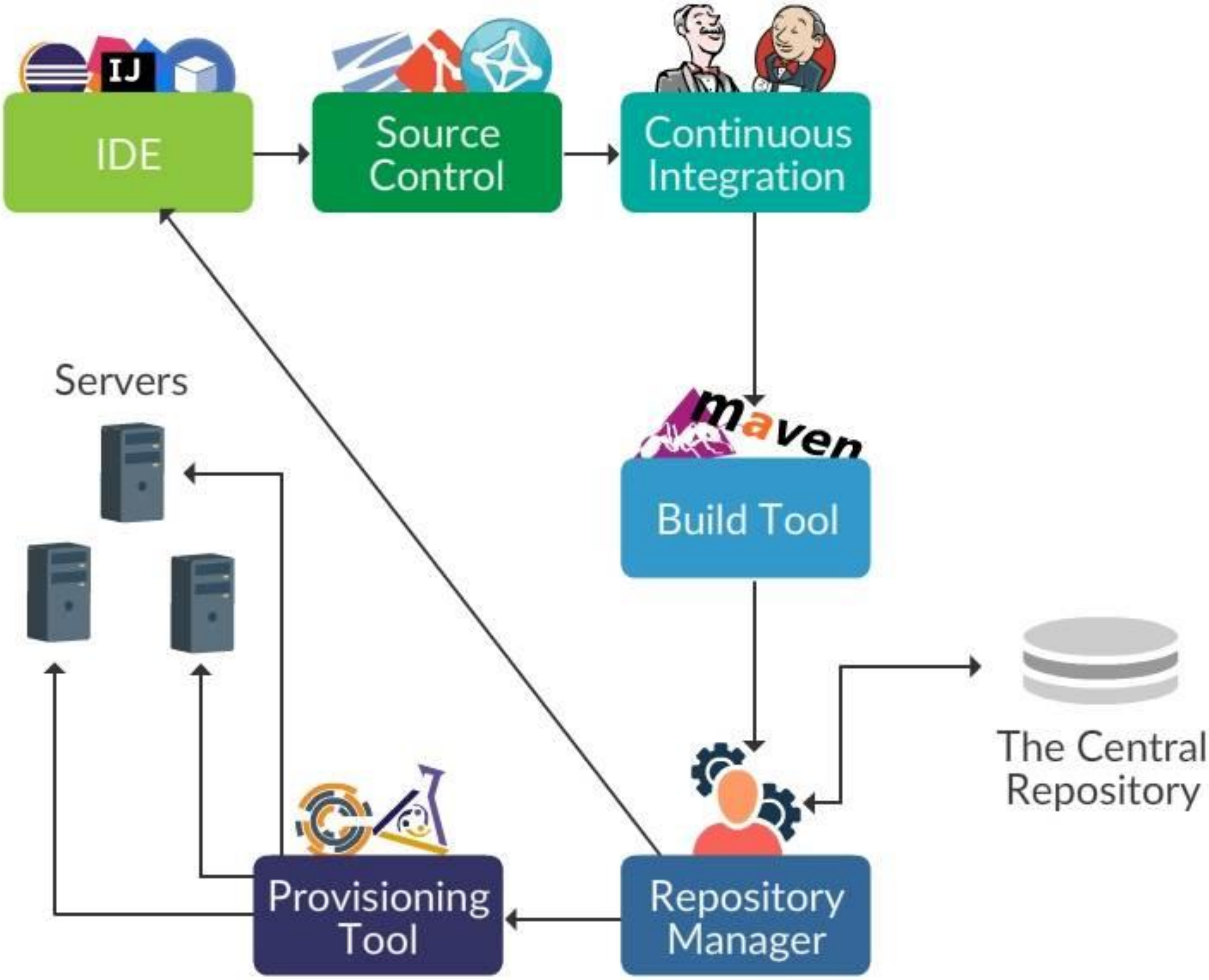
Capabilities and Advantages

Capabilities of Repository Manager

- Efficient search of components
- Integration with security systems like LDAP
- Hosts external components
- Controlled access to the components

Advantages

- Improved software build
- Improved collaboration
- Reduced Bandwidth and Time
- Scalability



The project flow is depicted in the form of a diagram here.

- ***IDE***: Place where developers write the source code.
- ***Source Control***: Helps in version control of the code.
- ***Build tool***: Builds up a project to artifacts it in the form of WAR, JAR. Example Maven
- ***Repository Manager***: The artifacts are stored in the Repository for collaborative development.

Prerequisite

Install Java 7 or higher version for Nexus version greater than 2.0

For more details, refer Sonatype Help.

Installation

- Nexus Repository Manager can be downloaded from Sonatype.

Extraction

OSX or Linux the downloaded GZip’d TAR archive can be extracted with these commands in terminal.

```
tar xvzf nexus-<version>.<tar file extension>
```

Example:

```
$ tar xvzf nexus-2.14.11-01-mac.tgz

$ tar xvzf nexus-2.14.11-01-unix.tar.gz
```

In Windows, extraction can be done in command prompt by:

```
$ 7za.exe e nexus-2.14.11-01-win64.zip
```

Nexus Platform Plugin for Jenkins is only compatible with Jenkins versions 2.x. Support for Jenkins 1.x is available using a now deprecated plugin.

Directories

On Extracting two directories get created namely:

- Application Directory
- Sonatype Work Directory (Data)

Application Directory

This directory contains the application and other additional components like Java libraries and Configuration files.

Nested Directories include: bin, deploy, etc, lib

Data Directory

This directory is the container of all components, repositories and other data that are managed by the repository manager.

Nested Directories: blob, cache, DB, health check

Starting Nexus

To start the repository manager from the application directory in the bin folder, type the following commands:

Unix and Mac OSX

```
cd bin

./nexus start
```

Windows

```
$ cd bin
$ nexus.exe /start
```

- To ensure that the repository manager is fully started, you can see a message similar to the following in the log depending on the operating system you use

Started Sonatype Nexus OSS

- Open your browser and type in the URL http://localhost:8081/ (Nexus 3) and http://localhost:8081/nexus (Nexus 2). The Nexus Repository Manager Welcome screen displays. ￼ ￼

Configuring as a service

- Configuring the Repository to run as service is necessary if it's been installed for the production usage environment.

The instructions for configuring the repository manager as a service on the various operating system is given below:

Running as a service in Window

1. Startup script on Windows platforms is bin/nexus.bat.
2. Install - Command with elevated privileges will set up the service.
3. uninstall - Command with elevated privileges will remove the service when desired.

- After the installation of service is done, the start and the stop service can be done using a batch file.
- Service is present in Windows Service Management console named Nexus.

Running as a Service in Mac OSX

launchd is a standard way of running a service in Mac OSX.

plist file is used for the configuration.

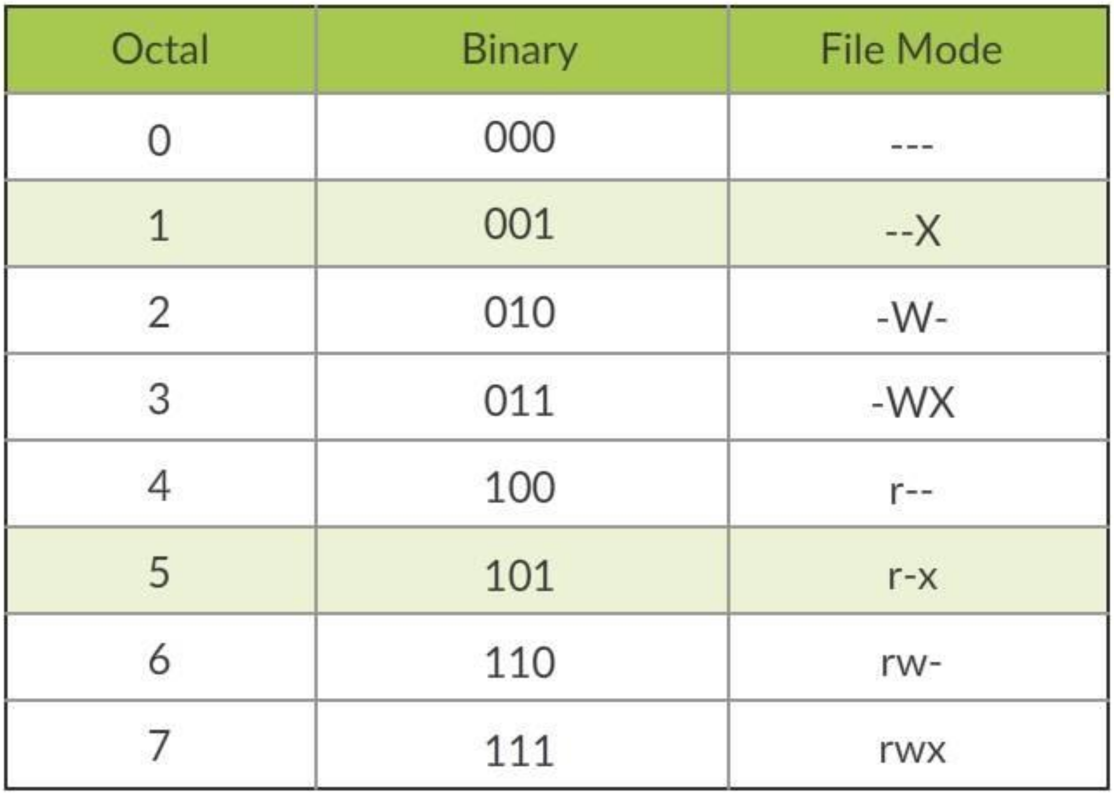
Save the plist file in /Library/LaunchDaemons/ as com.sonatype.nexus.plist

Change the ownership and access rights:

```
sudo chown root:wheel /Library/LaunchDaemons/com.sonatype.nexus.plist
sudo chmod 644 /Library/LaunchDaemons/com.sonatype.nexus.plist
```

Start the configuration by

```
sudo launchctl load /Library/LaunchDaemons/com.sonatype.nexus.plist
```



Octal	Binary	File Mode
0	000	---
1	001	--X
2	010	-W-
3	011	-WX
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwx

644 permision indicates:

1. Owner - read and write
2. Group - read
3. Other - read

The picture portrays the welcome screen of the Nexus Repository 3. (Nexus 2 differs by slight design changes)

Default Username and Password

Username: admin

Password: admin123.

Sign in with the credential to fall in the ocean of Nexus!

Admin Features

- ***Repository***

Manages all Repositories and related configurations.

- ***IQ Server***

Configuring of the connection of Nexus Repository Manager to Nexus IQ Server.

- ***Security***

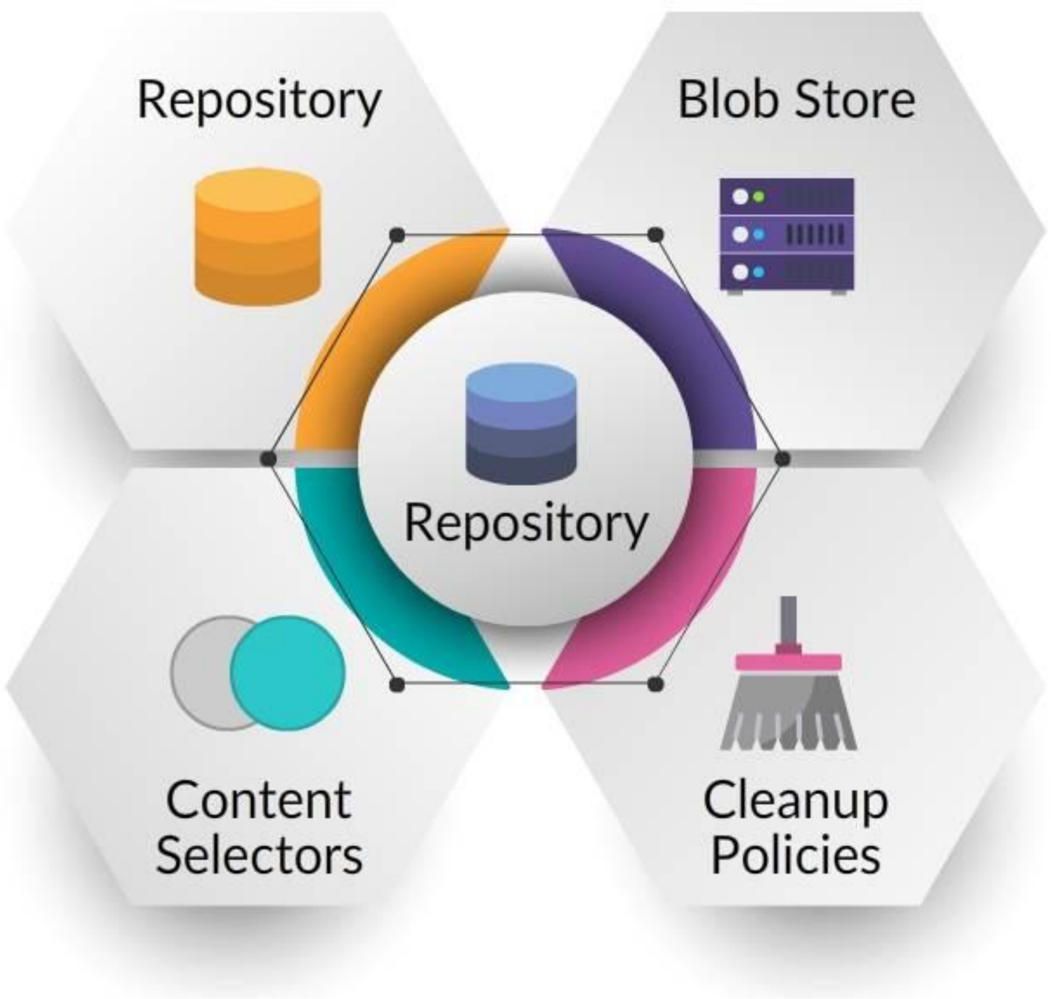
Authentication and authorization features which include Privileges, Roles, LDAP.

- ***Support***

Administers and monitor the repository manager.

- ***System***

The general configuration for getting started and running the repository manager



- ***Blob store***: A storage mechanism for the binary parts of the components and their assets.

- ***Proxy Repository***: Repository that is linked with the remote repository. (E.g. maven-central)
- ***Hosted Repository***: Repository storing components as the authoritative location for these components. (E.g. Maven-releases, Maven-snapshots)
- ***Repository Group***: Combine multiple repositories and other repository groups in a single repository. (E.g. Maven-Public)

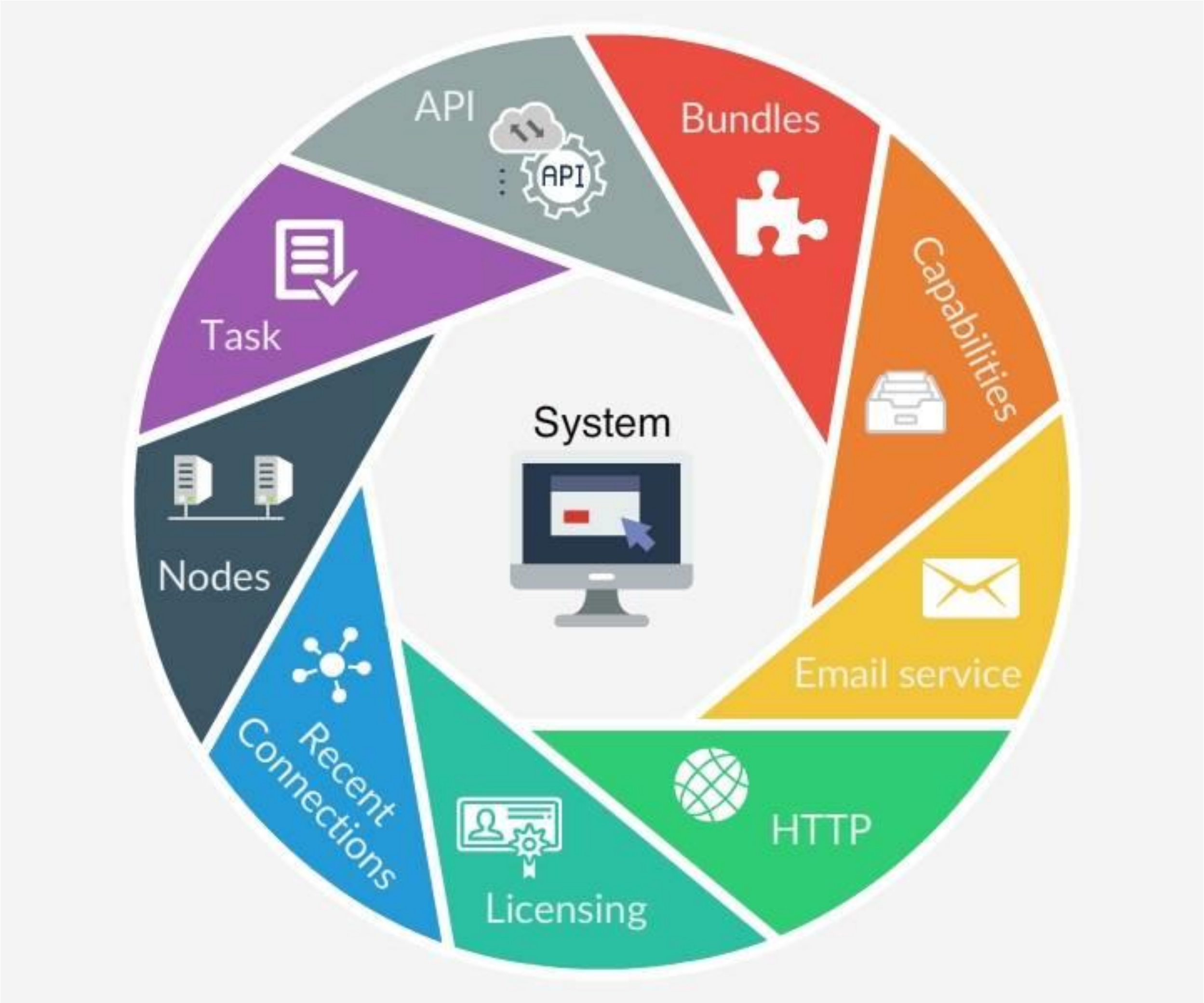
Repository Policy (Artifact Type)

Releases

- Specific Point in-time Releases.
- Solid and Stable
- PGP signatures and checksums verify the authenticity and integrity.
- Central Maven stores Release Artifacts.

Snapshots

- Captures the work in Progress.
- Named with version and Timestamp.



System gives access to many configuration features like:

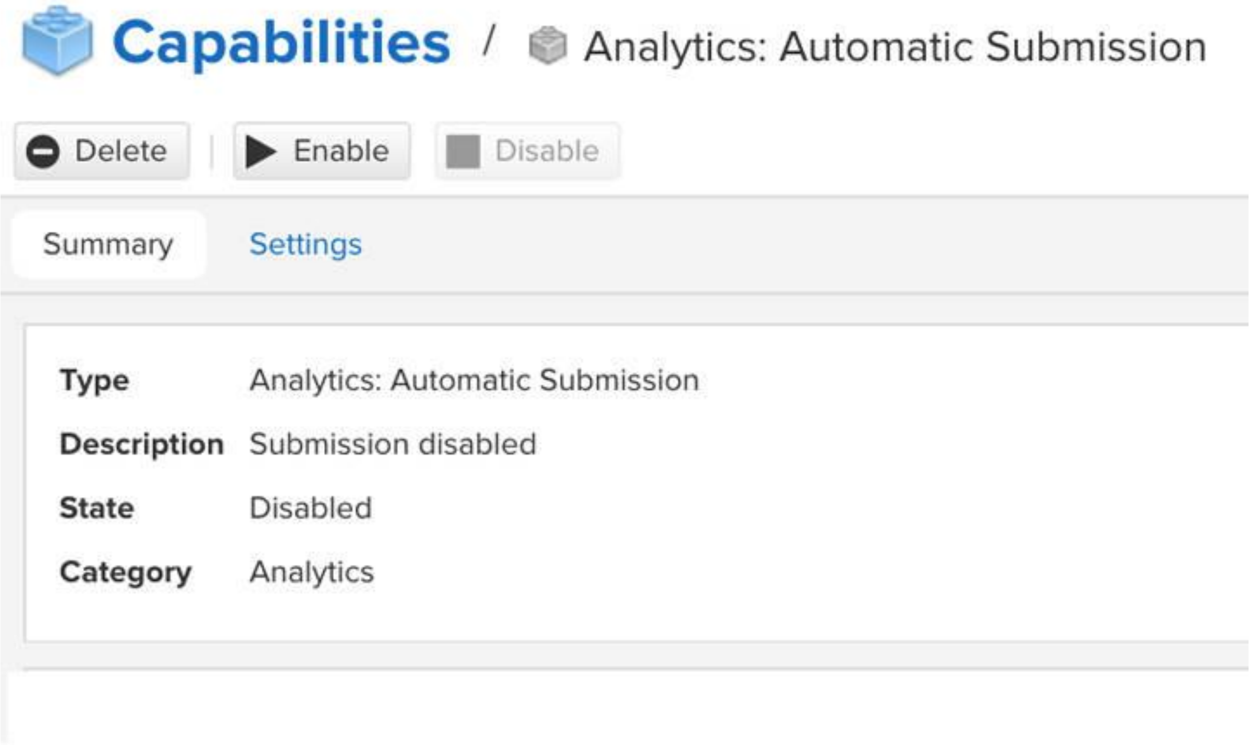
- Bundles
- Capabilities
- Email/SMTP server configuration
- Nodes for high availability
- HTTP/HTTPS proxy server configuration
- Base URL for an application and repository manager
- Tasks

System - Bundles

The Nexus Repository Manager application runs on the OSGi container (Open Services Gateway initiative), Apache Felix.

Features are maintained as OSGi Bundles.

Bundles allow us to inspect the list of OSGi bundles to have access to the detailed information.



Type Analytics: Automatic Submission
Description Submission disabled
State Disabled
Category Analytics

The repository manager gets in with the number of capabilities which has been installed beforehand and this allows us to enable/disable them.

System - Email Server

- The Nexus may send out messages for a number of reasons like the trigger in the build.
- For the successful delivery of messages, we need to configure the connection to the SMTP server.

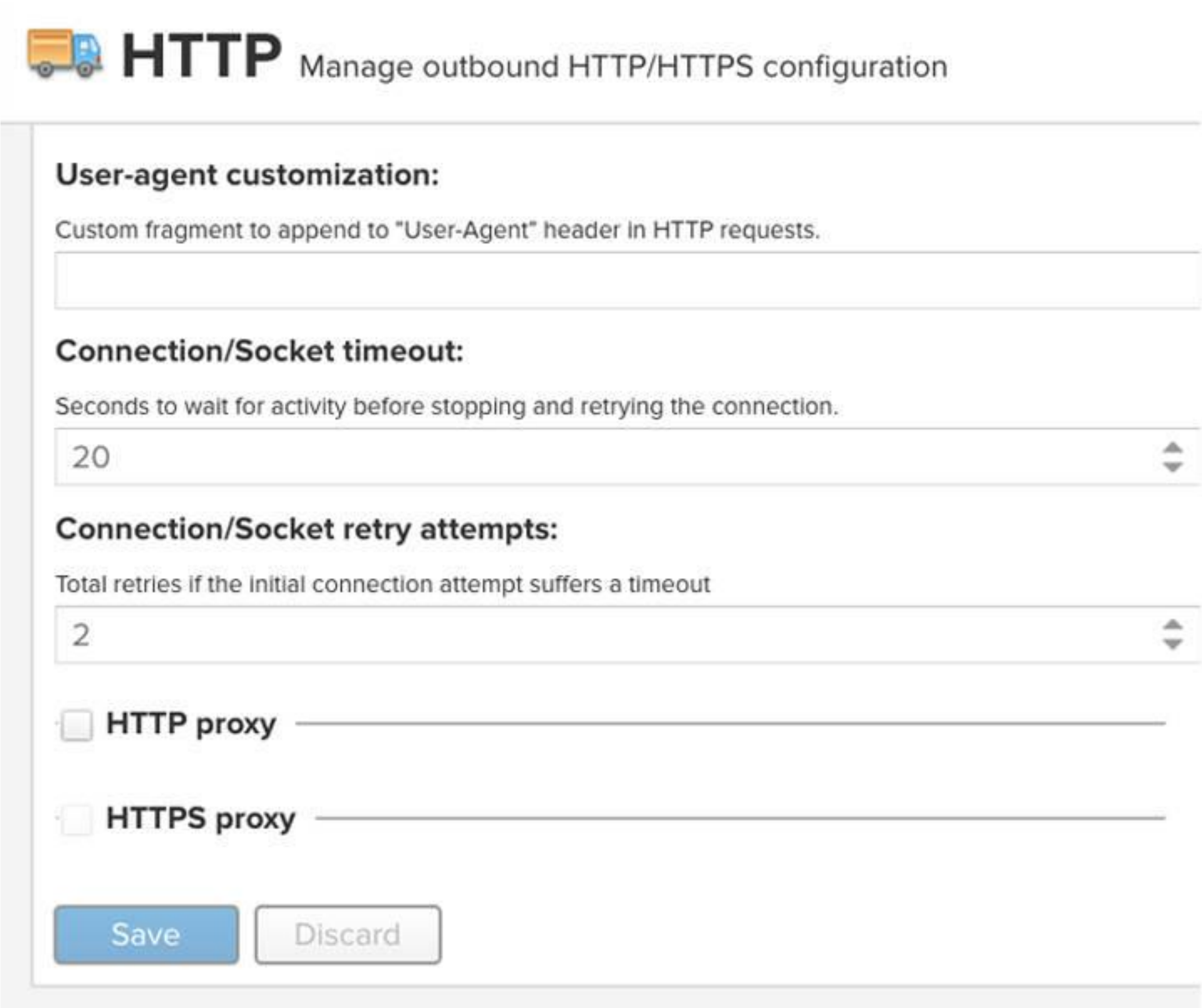
System - Nodes and Base URL

Nodes

The Nodes screen summarizes all active nodes.

Base URL

The Base URL helps the users in accessing the user interface.



HTTP Manage outbound HTTP/HTTPS configuration

User-agent customization:

Custom fragment to append to "User-Agent" header in HTTP requests.

Connection/Socket timeout:

Seconds to wait for activity before stopping and retrying the connection.

20

Connection/Socket retry attempts:

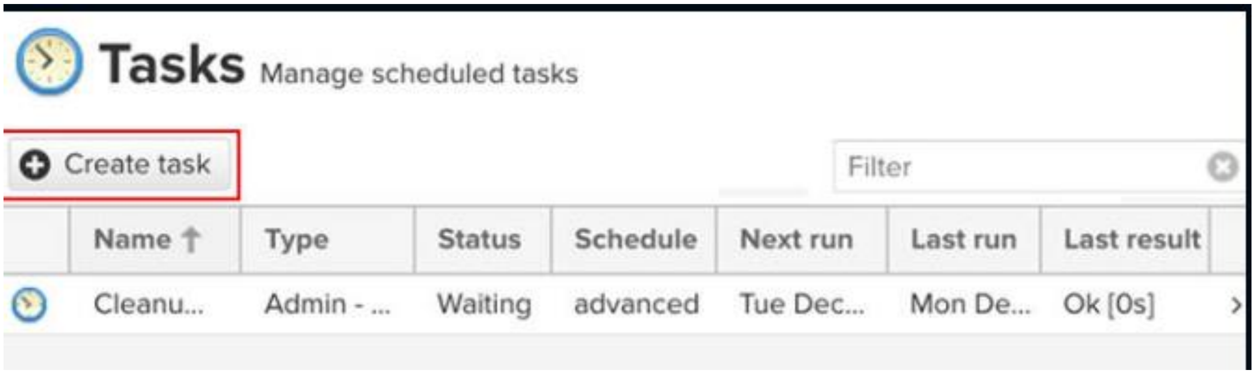
Total retries if the initial connection attempt suffers a timeout

2

□ HTTP proxy

□ HTTPS proxy

This feature of nexus helps in fetching the content from the remote servers, supported by HTTP requests.



Tasks Manage scheduled tasks

Name ↑	Type	Status	Schedule	Next run	Last run	Last result
Cleanu...	Admin - ...	Waiting	advanced	Tue Dec...	Mon De...	Ok [0s]

- Tasks help in scheduling the execution of maintenance tasks.

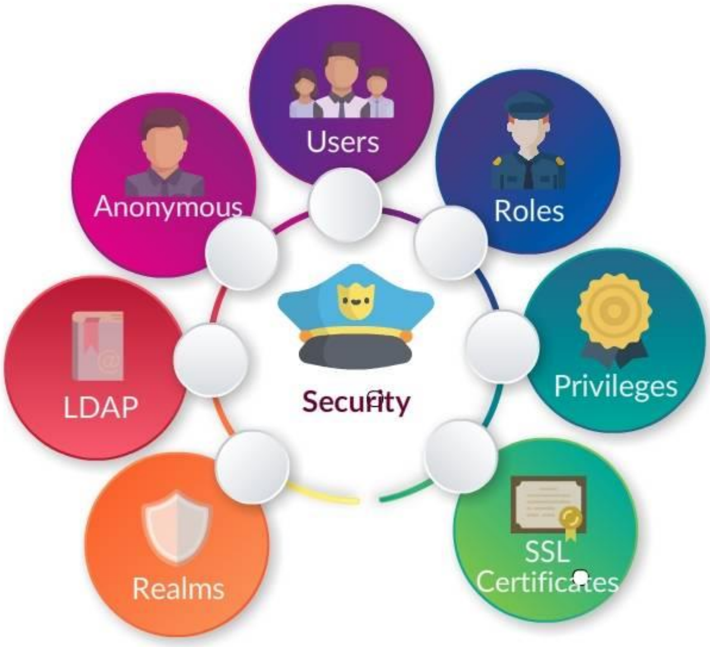
When you want to run a task every day at 4 PM, you need not wait till the clock strikes 4. System task does it for you!

Support

- Ensures the server is configured correctly.
- Provides tools to investigate configuration details.
- Useful for troubleshooting.

Features of Support

1. Analytics
2. Logging and Log Viewer
3. Metrics
4. Support ZIP
5. System Information



Security is designed around the following concepts.

- ***Privileges***: Rights to read, write and execute
- ***Roles***: Privileges, when grouped into collections, is called roles.
- ***Users***: Users can be assigned one or more roles.

- ***LDAP***: They authenticate by Lightweight Directory Access Protocol via the external system which has the support

- ***Realms***

IQ Server

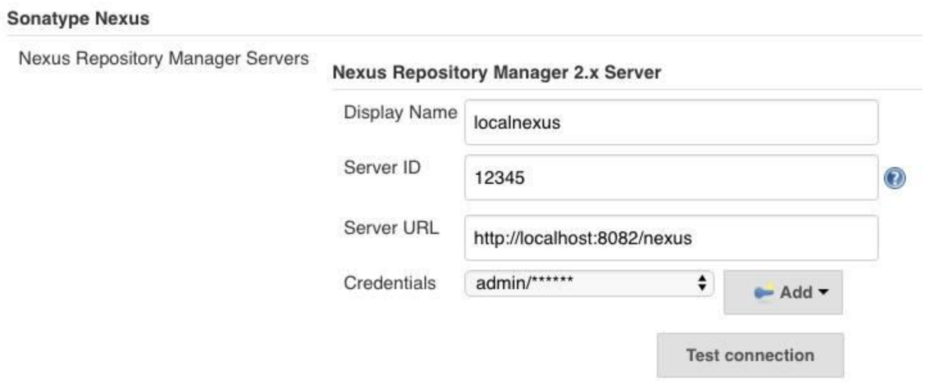
They help in configuring the connection of Nexus Repository Manager to Nexus IQ Server.

Time for the Nexus

To blend Nexus with Jenkins, Plugin has to be installed in Jenkins.

Follow the sequence for the magic installation of Nexus in Jenkins.

- Choose Manage Jenkins tile.
- Select Manage plugin tab which shows the various colors of plugins like Updates, Available and Installed.
    - Updates: Notifies the latest releases
    - Available: Plugins that are not installed yet.
    - Installed: Plugins that are installed.
- Go to Available and install the plugin named **Nexus Platform** to integrate Sonatype Nexus into Jenkins.
- Finally, choose Install without Restart option to help Jenkins embrace Nexus plugin.



Sonatype Nexus

Nexus Repository Manager Servers

Nexus Repository Manager 2.x Server

Display Name: localnexus

Server ID: 12345

Server URL: http://localhost:8082/nexus

Credentials: admin/******

The following sequence of the steps will lead you to the Configuration of Nexus server details in Jenkins.

- Select Manage Jenkins from the left navigation bar and choose Configure System.
- Scroll down the page till you can find ***Sonatype Nexus*** section.
- Select Add Nexus Repository Manager Server.

To complete the perfect configuration two main steps have to be done namely:

1. Fill out Server Details
2. Adding Credentials
3. Test connection

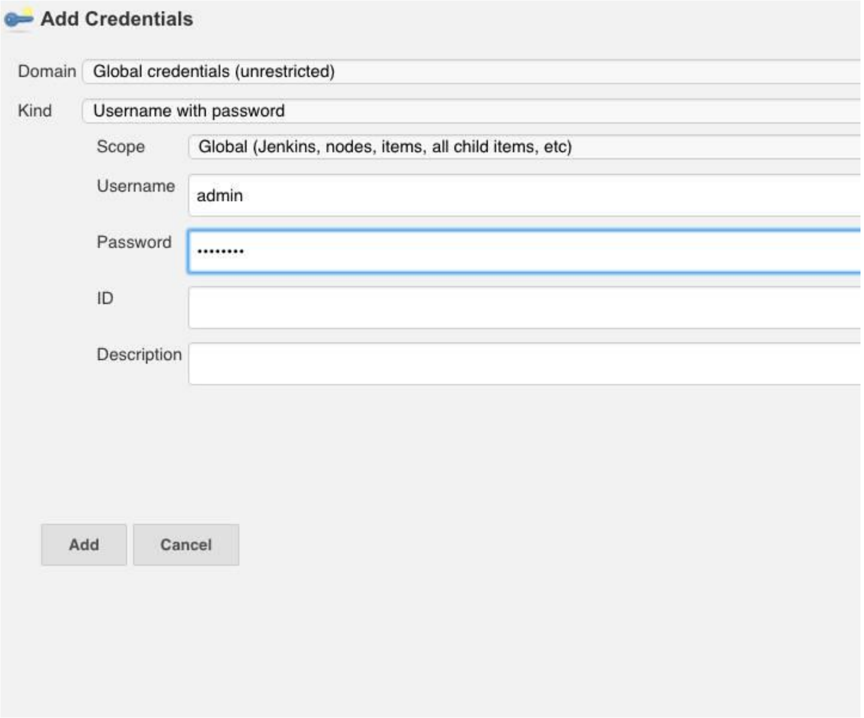
Fill Out the Server Details

Details of the Nexus Server must be filled out to get connected with Jenkins.

1. ***Display name***: Name of server to be displayed in Jenkins (User Preference
2. ***Server Id***: Id of the server for reference (User Preference)
3. ***Server URL***: URL in which our Nexus server runs.

- Nexus 2: http://localhost:8081/nexus
- Nexus 3: http://localhost:8081

4. Credentials: Username and password of Nexus Server



Add Credentials

Domain: Global credentials (unrestricted)

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: admin

Password: ••••••••

ID:

Description:

The picture portrays the page of Add Credentials. In order to add the credentials of Nexus Server into Jenkins, we configure the credentials of Nexus.

- Choose Add.
- Give the details of Username and Password and click on Add button.
- Choose the newly added credentials from the drop-down menu of credentials to get Jenkins logged in with that credential.

In our case, we have used the default **Username: admin and Password: admin123**

- Click the Test Connection button.
- Ensure the status of connection by Success message as depicted in the picture.

Now, it's all done for the part of blending Nexus with Jenkins.

We will use jenkins to trigger a code using maven integrated build and push to nexus .

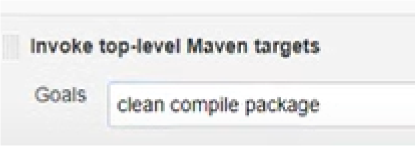
Integration :

Install maven on jenkins

Create repo in nexus where we can upload .

We will create pipeline , use git for source code and we will use maven to compile code .

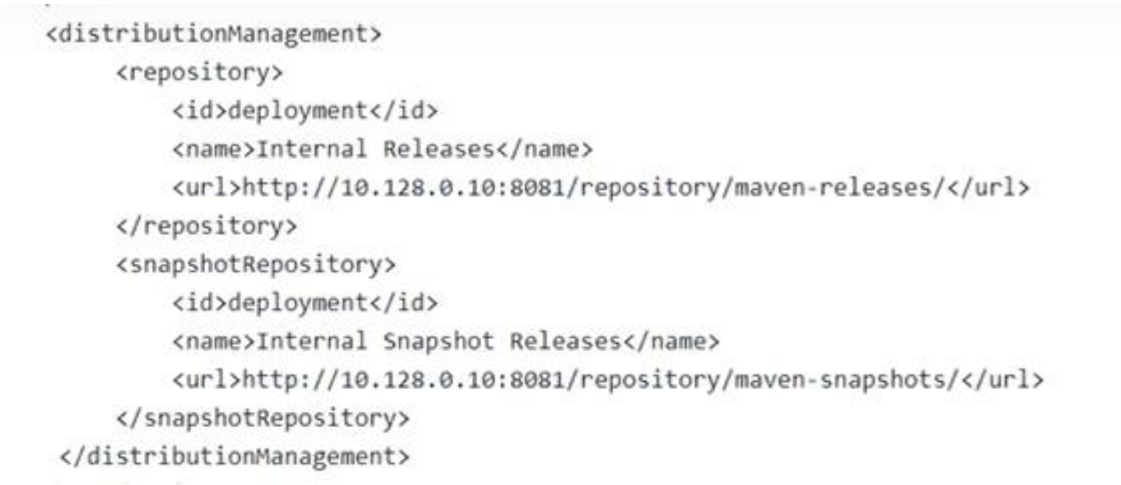
We can use maven goals in pipeline



□ Invoke top-level Maven targets

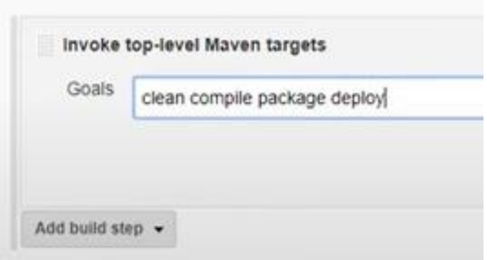
Goals: clean compile package

Once build is successful we can upload artifact to nexus . IN pomxml file we can update the



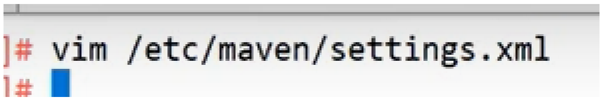
```
 <distributionManagement>
      <repository>
          <id>deployment</id>
          <name>Internal Releases</name>
          <url>http://10.128.0.10:8081/repository/maven-releases/</url>
      </repository>
      <snapshotRepository>
          <id>deployment</id>
          <name>Internal Snapshot Releases</name>
          <url>http://10.128.0.10:8081/repository/maven-snapshots/</url>
      </snapshotRepository>
  </distributionManagement>
```

Repo urls where we want to store our artifacts .

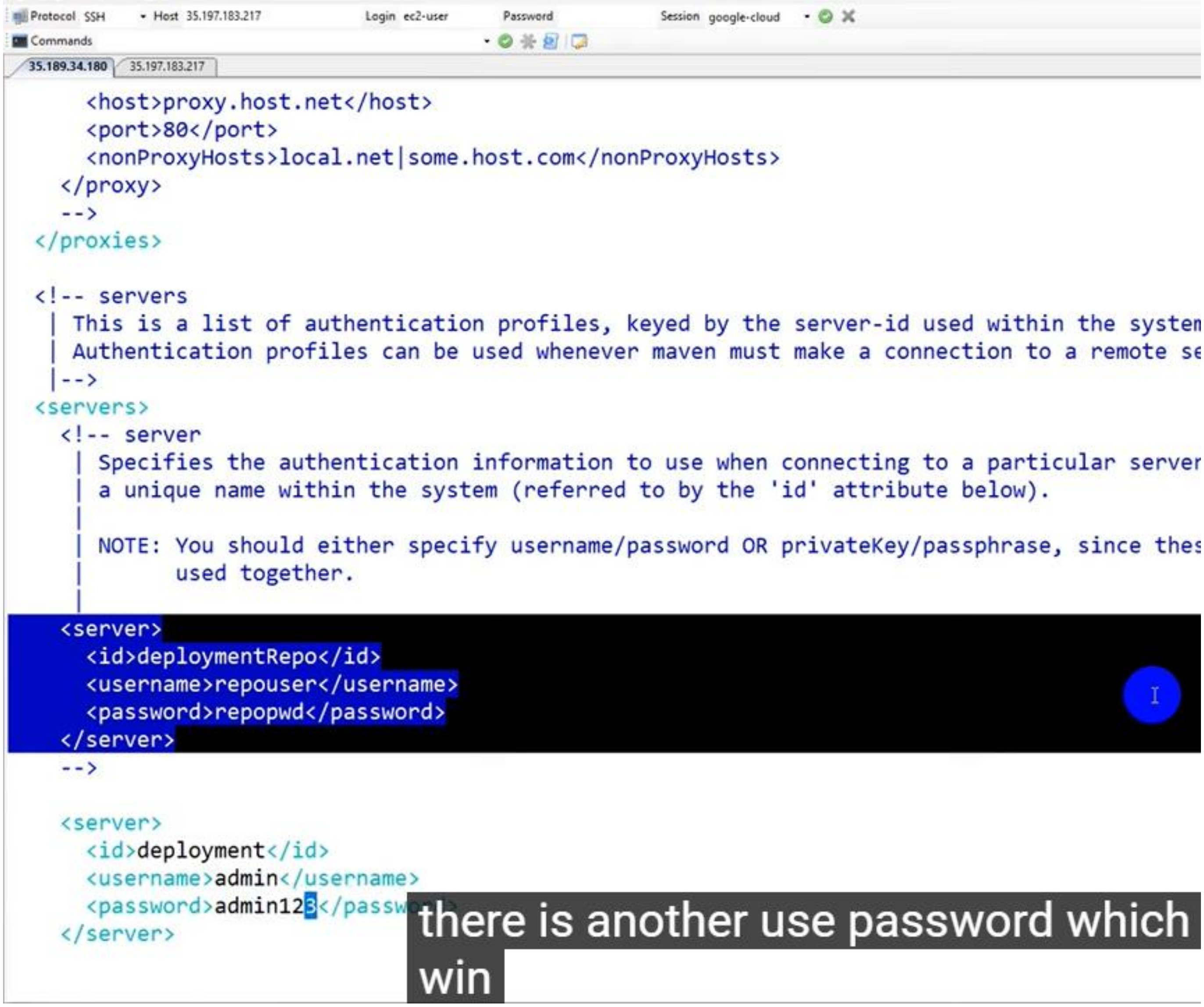


Goals: clean compile package deploy

We can add deploy using above goal in nexus . We can update creds for nexus in file .



```
]# vim /etc/maven/settings.xml
]#
```



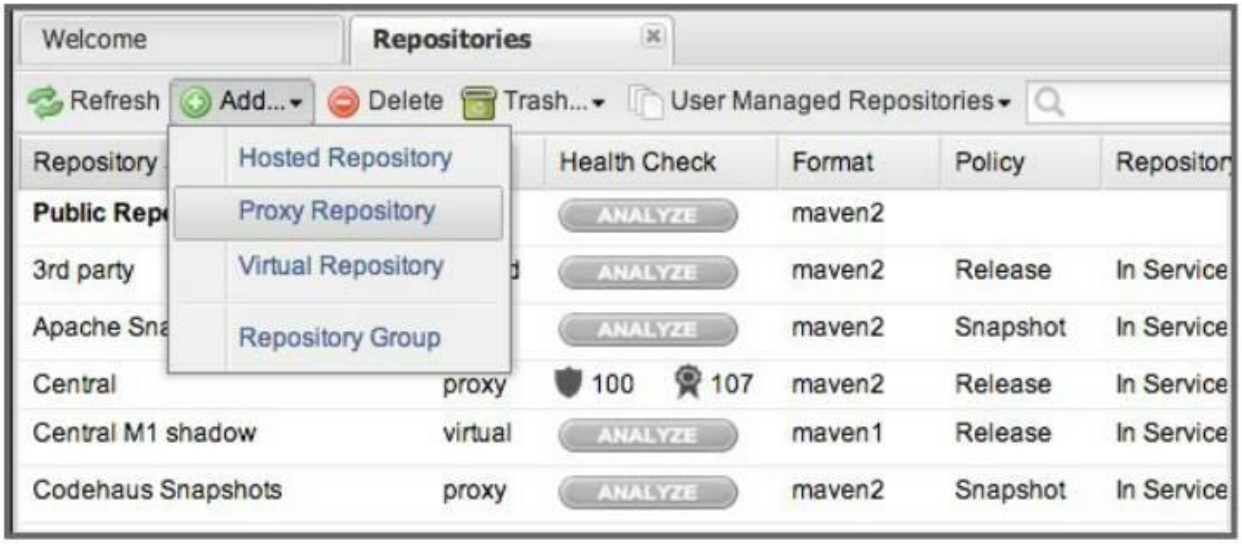
```
      <host>proxy.host.net</host>
      <port>80</port>
      <nonProxyHosts>local.net|some.host.com</nonProxyHosts>
    </proxy>
    -->
  </proxies>

  <!-- servers
   | This is a list of authentication profiles, keyed by the server-id used within the systen
   | Authentication profiles can be used whenever maven must make a connection to a remote se
   |-->
  <servers>
    <!-- server
     | Specifies the authentication information to use when connecting to a particular server
     | a unique name within the system (referred to by the 'id' attribute below).
     |
     | NOTE: You should either specify username/password OR privateKey/passphrase, since thes
     |       used together.
     |
    <server>
      <id>deploymentRepo</id>
      <username>repouser</username>
      <password>repopwd</password>
    </server>
    -->

    <server>
      <id>deployment</id>
      <username>admin</username>
      <password>admin123</passw
    </server>
```

there is another use password which win

And now when we run job artifact will be deployed to repo .

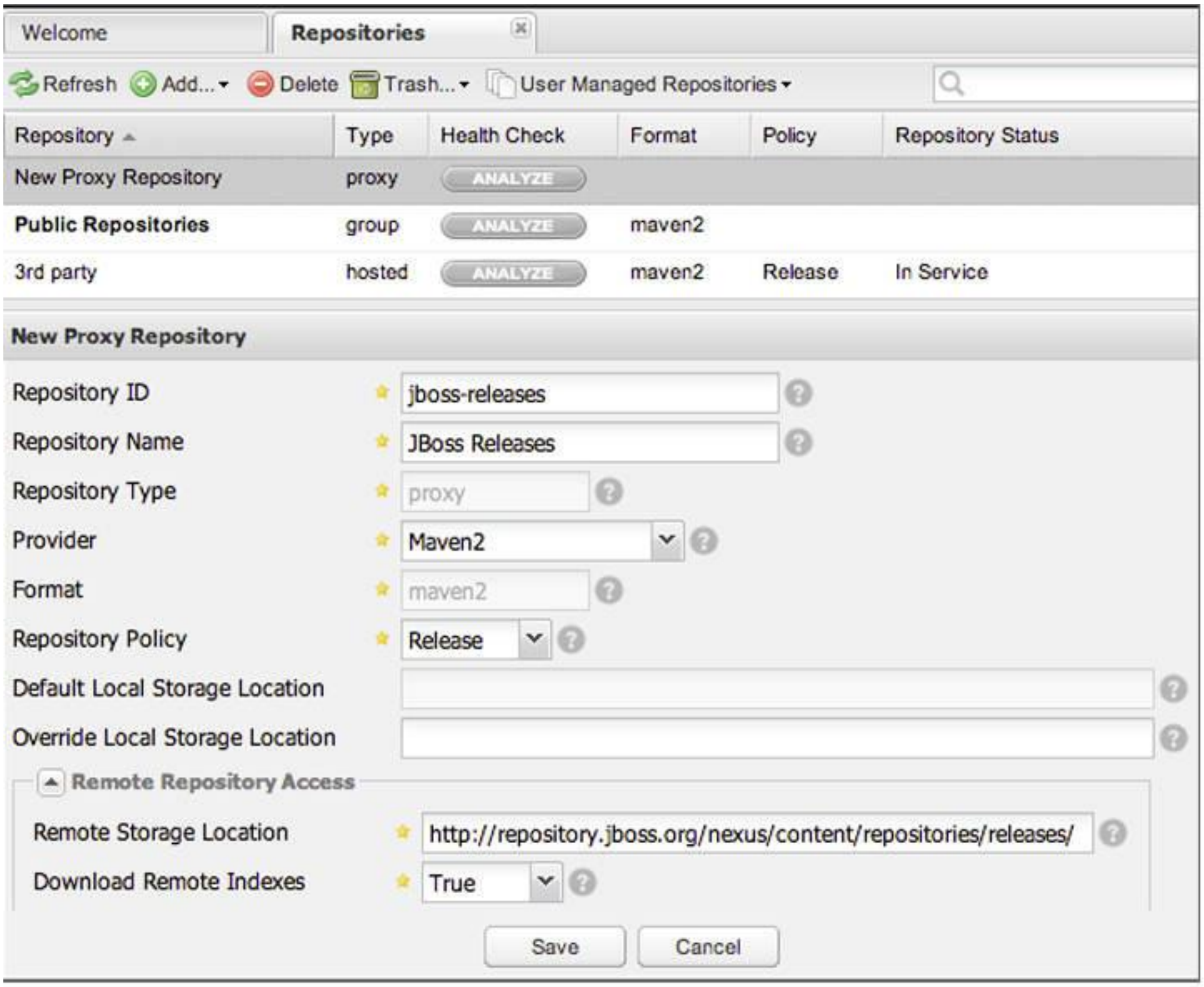


Repository		Health Check	Format	Policy	Repositor
Public Rep		ANALYZE	maven2		
3rd party	d	ANALYZE	maven2	Release	In Service
Apache Sna		ANALYZE	maven2	Snapshot	In Service
Central	proxy	100 107	maven2	Release	In Service
Central M1 shadow	virtual	ANALYZE	maven1	Release	In Service
Codehaus Snapshots	proxy	ANALYZE	maven2	Snapshot	In Service

Finally, its time for our Nexus to get its hands-on.

Creating a Repository

- Now let's log in to the Nexus Repository 2 with the credentials. (For easy use, we have used the default admin credentials in all cases)
- Choose Views\Repositories tile.
- Click the +Add button to select the type of Repository that is needed.
- Configuring Repository page appears where the details of the repository have to be populated.



Repository	Type	Health Check	Format	Policy	Repository Status
New Proxy Repository	proxy	ANALYZE			
Public Repositories	group	ANALYZE	maven2		
3rd party	hosted	ANALYZE	maven2	Release	In Service

New Proxy Repository

Repository ID	jboss-releases
Repository Name	JBoss Releases
Repository Type	proxy
Provider	Maven2
Format	maven2
Repository Policy	Release
Default Local Storage Location	
Override Local Storage Location	
Remote Storage Location	http://repository.jboss.org/nexus/content/repositories/releases/
Download Remote Indexes	True

Required fields

Repository ID and Repository Name.

- ***Repository ID***: It will be part of the URL used to access the repository,
- ***Repository Policy***: Release or Snapshots

Click Save button.

Let's try Creating our own Repository

- Having Installed Nexus in previous hands-on, let's try creating our own Repository!!
- Perform ***Step 4 and 5*** in addition to the previous steps in the hands-on environment and explore.