```
====================
Nexus Repository
====================

=> Nexus is called as Artifactory Server.

=> It is used to store project artifacts (jar/war files) as backup.

=> It is also used as Maven Remote Repository.

=> Remote Repositories are used to store shared libraries (jars).

=> Shared libs means the jars which are required for several projects in the same company

Note: In realtime, every company will have their own remote repository.

=================
Nexus vs Git Hub
=================

=> Git Hub is used as version control system
=> Git Hub is used for storing project source code

=> Nexus is used as artifactory server
=> Nexus is used for storing project build artifacts & shared libs
=> Nexus runs on port no 8081

==========================
Nexus Setup using Docker
==========================

URL : https://github.com/ashokitschool/DevOps-Documents/blob/main/07-Nexus-Setup-Docker.md

========================================
Configure Nexus Server details in maven
========================================

-> Add nexus server details in maven settings.xml file

                <server>
                        <id>nexus</id>
                        <username>admin</username>
                        <password>admin</password>
                </server>

============================================
How to upload project artifact into Nexus ?
============================================

=> When we package our maven application it will generate a jar / war file. That jar/war is called as
project build artifact.

=> Project build artifacts we will store into nexus for backup purpose.

=> To store project artifacts we will create nexus repositories.

=> We have 2 types repositories in nexus to store build artifacts.

                1) snapshot repository

                2) release repository

-> If project is under development then that project build artifacts will be stored into snapshot
```

repository.

-> If project development completed and released to production then that project build artifacts will be stored to release repository.

=> Create Two Repositories one as snapshot and another one as release.

	Snapshot Repo URL : http://3.109.184.254:8081/repository/ashokit-snapshot-repo/

	Release Repo URL : http://3.109.184.254:8081/repository/ashokit-release-repo/

-> Nexus Repository details we will configure in project pom.xml file like below

```
<distributionManagement>

	<repository>
		<id>nexus</id>
		<name>Ashok IT Releases Nexus Repo</name>
		<url>http://3.109.184.254:8081/repository/ashokit-release-repo/</url>
	</repository>

	<snapshotRepository>
		<id>nexus</id>
		<name>Ashok IT Snapshots Nexus Repo</name>
		<url>http://3.109.184.254:8081/repository/ashokit-snapshot-repo/</url>
	</snapshotRepository>

</distributionManagement>
```

-> Once these details are configured then we can run below maven goal to upload build artifacts to Nexus Server.

	$ mvn clean deploy

Note: When we execute maven deploy goal, internally it will execute 'compile + test + package + install + deploy' goals.

Note: Based on <version/> name available in project pom.xml file it will decide artifacts should be stored to which repository.

<version>0.0.1-SNAPSHOT</version> : Upload to snapshot repository

<version>0.0.1-RELEASE</version> : Upload to release repository

==========
Remote Repository
==========

-> Remote repository used for maintaining shared libraries (common jars required for multiple projects)

-> If we want to use few jar files in multiple projects in the company then we will use Remote Repository to store those jars (libraries).

-> Remote repository is specific to our company projects.

==========
Create Remote Repo and Upload Jar file
==========

-> Go to Settings
-> Go to Repositories

-> Create New Repository
-> Choose Maven (Hosted) Repository
-> Give a name for Repository (Ex: ashokit-remote-repository) & Complete the process

Note: With above steps Remote Repository got created.

Remote Repo URL : http://3.109.184.254:8081/repository/ashokit-remote-repo/

=> Now upload jar file into remote repository

        -> Go to BrowseSection
        -> Select Remote Repository (By default it is empty)
        -> Click on Upload Component
        -> Upload Jar file and give groupId, artifactId and Version

                groupId : in.ashokit
                artifactId : pwd-utils
                version : 1.0

-> Take dependency details of uploaded jar file and add in project pom.xml as a dependency like below

```
<dependency>
  <groupId>in.ashokit</groupId>
  <artifactId>pwd-utils</artifactId>
  <version>1.0</version>
</dependency>
```

> We need to add Remote Repository Details in pom.xml above <dependencies/> tag

```
        <repositories>
                <repository>
                        <id>nexus</id>
                        <url>http://3.109.184.254:8081/repository/ashokit-remote-repo/</url>
                </repository>
        </repositories>
```

-> After adding the remote repository details in pom.xml then execute maven package goal and see dependency is downloading from nexus repo or not.

```
                $ mvn clean package
```

==========
How to resolve HTTP Mirror Block Issue ?
==========

=> Make below change in maven settings.xml

```
<mirror>
      <id>maven-default-http-blocker</id>
      <mirrorOf>dummy</mirrorOf>
      <name>Pseudo repository to mirror external repositories initially using HTTP.</name>
      <url>http://0.0.0.0/</url>
      <blocked>false</blocked>
</mirror>
```

==========
Nexus Summary
==========

1) What is Nexus and Why we need to go for Nexus ?

2) How to setup Nexus Server using Docker

3) How to create Repositories in Nexus (snapshot & release)

4) How to upload build artifacts into Nexus Repositories

5) What are Shared Libraries ?

6) How to create Remote Repository ?

7) How to upload Shared Libraries into remote repository

8) How to configure remote repository in pom.xml file

9) Download shared libraries from Remote Repository.

Steps to Create a Repository in Nexus Server
==========
Go to settings=>create repository=>Select maven2(hosted) => Select(snapshot/releas)=>Create Repository=>Then copy the URL we need to put <dependencyManagement></dependencyManagement> this code inside pom.xml file i.e above <dependencies> tag. Our project artifact will be stored in which repository that depends on out project version(<version>0.0.1-SNAPSHOT</version>

To connect our Maven with Nexus Repository we need to add the below code inside settings.xml file inside <servers></servers>

```
<server>
  <id>nexus</id>   // This id should be match with the id inside <distributionManagement>
  <username>admin</username>
  <password>Jilucse@2002</password>
</server>
```