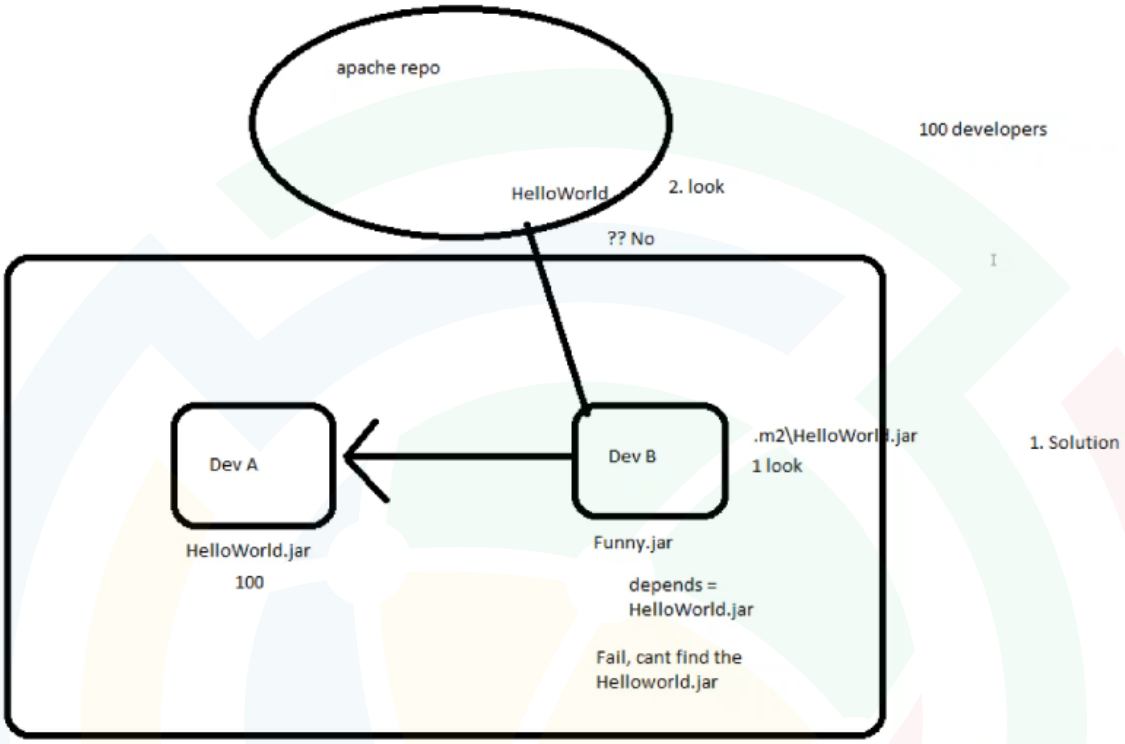
NEXUS

⇒ Nexus is a Binary Repository Manager



Now the simple solution for this problem is, Dev B should ask Dev A to provide the HelloWorld.jar, so that DevB can keep the Hello.jar in DevB machines .m2 directory,

This works coz the maven will look first in .m2 directory.[Local Repo]
If not then it maven central [Apache Repo]
Now imagine DevA, keeps on changing the Hello.jar code, let's say DevA changed 100 times now DevB should ask DevA 100 times which doesn't make sense.

Tomorrow in your project, you have 100 Developers, now they have to exchange their libraries, now it won't be that easy to exchange the libraries.

It's really cumbersome process, Nobody understands which version is there with whom.

The solution is there, but this is a complex and time taking solution.
This may workout if there are only 2-3 developers, but not more then that.

This is where Binary Repository Concept comes into picture.

Now we will introduce a new server within our organization. This server we call it as Remote Repository.

Just like how apache is maintaining a MavenCentral, similarly we will maintain our own remote repository.

Now DevA instead of sharing his Hello.jar with DevB, DevC etc
He will push it to Our Remote Repository and now DevB, DevC everyone who needs that Hello.jar will pull it from the Remote Repository.

I mean they[DevB, DevC] will add that info in the pom.xml, now pom will take care of downloading it from remote repo.
ow we got totally three kinds of Repo's.

1. Local
2. Public Repo
3. Private Repo

Now even the libraries are secured, coz they are present within your organization.

We got different tools for that Artifactory/Nexus/Archiva.

Sonatype Nexus
==========

Now we are going to set up this within our server.

Install Nexus Server.

Search for Apache Maven Nexus Repository in google,

Snapshot ⇒ Development progress build [Partial Completed Jars]

Release ⇒ Ready to release build [Official proper release]

Installation of Nexus
==========

type download nexus in google
go with version 2.x as 3.x is not yet supported with jenkins
copy the link address for 2.x and do wget
wget http://www.sonatype.org/downloads/nexus-latest-bundle.tar.gz
mkdir -p tools/nexus
tar xvzf nexus.tar.gz -C tools/nexus
Nexus by default starts on the port number 8081 and un & pw is admin & admin123
cd nexus/nexus-2.x/bin
./nexus start { if any problem change ownership to devops to both dirs of nexus}
netstat -ntpl | grep 8081 { be patient it takes some time to start }
http://ip-addr:8081/nexus { allow port 8081 through firewall }
login and give default creds admin & admin123

ERROR: Change the permissions of two nexus directories with logged in username.

http://ip-addr:8081/nexus

Once the nexus is up we will create two repositories, Snapshot & Release.

Search for distribution management tag in google for maven & nexus, and paste it after </dependencies> tag.

If your version contains a string SNAPSHOT, by default it goes to SNAPSHOT repo.
If your version contains only version 1.0, it goes to RELEASE repo.

Now we will, deploy the artifacts to remote repo by deploy phase.

Login to nexus using admin & admin123, now we already we have some default repos, we have something Central, this is Apache Maven Central.

But we will go with our own repo's, using HOSTED repo's.

Add → Hosted Repo → Repo Id: releaseRepo → Repo Name: releaseRepo → Repo Policy: Release → Deployment Policy: Disable Redeploy → Save

Add → Hosted Repo → Repo Id: snapshotRepo → Repo Name: snapshotRepo → Repo Policy: Snapshot → Deployment Policy: Allow Redeploy → Save

Goto the maven project and do # mvn deploy

Search for maven distributionmanagement nexus in google,

Goto pom.xml and update <distributionManagement> below </dependencies>

<distributionManagement>

<repository>
<id>releaseRepo</id>
<name>releaseRepo</name>
<url>http://192.168.56.101:8081/nexus/content/repositories/releaseRepo/</url>
</repository>

<snapshotRepository>
<id>snapshotRepo</id>
<name>snapshotRepo</name>
<url>http://192.168.56.101:8081/nexus/content/repositories/snapshotRepo/</url>
</snapshotRepository>

</distributionManagement>

Goto the maven project and do # mvn deploy again, now we get new error like 401.

Nexus is strictly authenticated, you cannot deploy until and unless you login,

We don't provide usernames and passwords in pom.xml, coz pom.xml files are stored in VCS, which will be shared to other[most] developers as well.

For this maven provides solution in local repo,

```
# cd ~/.m2
# vi settings.xml      {search for maven settings.xml nexus username and pass }
```

```
<settings>
    <servers>
      <server>
        <id>releaseRepo</id>
        <username>admin</username>
        <password>admin123</password>
    </server>
    <servers>
      <server>
        <id>snapshotRepo</id>
        <username>admin</username>
        <password>admin123</password>
    </server>
</settings>
```

Now change the pom.xml <version> to 1.0-SNAPSHOT and redeploy again.

```
# mvn deploy[After one min, again run mvn deploy]
# mvn deploy
```

Snapshot is generally for other developers to get the dependency.

Nexus Assignment

==========

Follow this repo to get the code

https://github.com/ravi2krishna/CalculatorTestCases.git

Package the application and implement the add(), substract() and multiply() using the JAR which will be generated from the above repository.