Nexus practical task

> ## Practical task
>
> 1. Install Nexus
> 2. Configure admin user
> 3. Configure proxy repo
>   a. For Maven: Create setting.xml, maven should use it as mirror
>   b. For Gradle: Define proxy repo in gradle settings
> 4. Upload spring-petclinic Artifact to Nexus
>   a. For Maven: Use the deploy plugin in your pom.xml. You should be able to upload your artifacts to Nexus using Maven.
>   b. For Gradle: Define artifact upload as a separate Gradle task
> 5. Search artifacts in Nexus

1. Install Nexus

Download Nexus archive to `/opt` directory.

```
cd /opt/
sudo wget https://download.sonatype.com/nexus/3/nexus-3.74.0-05-
unix.tar.gz
sudo tar xzf nexus-3.74.0-05-unix.tar.gz
```

```
04:25:24 adrwal@olek-desktop-pc opt → sudo tar xzf nexus-3.74.0-05-unix.tar.gz
04:26:29 adrwal@olek-desktop-pc opt → ls
google  nexus-3.74.0-05  nexus-3.74.0-05-unix.tar.gz  sonatype-work  tomcat
04:26:31 adrwal@olek-desktop-pc opt →
```

Now change ownership from root to our user for unpacked Nexus directories (`nexus-3.74.0-05` and `sonatype-work`).

```
04:26:31 adrwal@olek-desktop-pc opt → sudo chown adrwal:adrwal sonatype-work/ -R
04:27:54 adrwal@olek-desktop-pc opt → sudo chown adrwal:adrwal nexus-3.74.0-05/ -R
04:27:58 adrwal@olek-desktop-pc opt → la -al
total 226M
drwxr-xr-x  6 root   root   4.0K Nov 12 16:26 .
drwxr-xr-x 23 root   root   4.0K Nov  6 22:17 ..
drwxr-xr-x  3 root   root   4.0K Jul 20 20:10 google
drwxr-xr-x 10 adrwal adrwal 4.0K Nov 12 16:26 nexus-3.74.0-05
-rw-r--r--  1 root   root   226M Nov  5 07:17 nexus-3.74.0-05-unix.tar.gz
drwxr-xr-x  3 adrwal adrwal 4.0K Nov 12 16:26 sonatype-work
drwxr-xr-x  9 root   root   4.0K Nov  5 18:30 tomcat
04:28:01 adrwal@olek-desktop-pc opt →
```

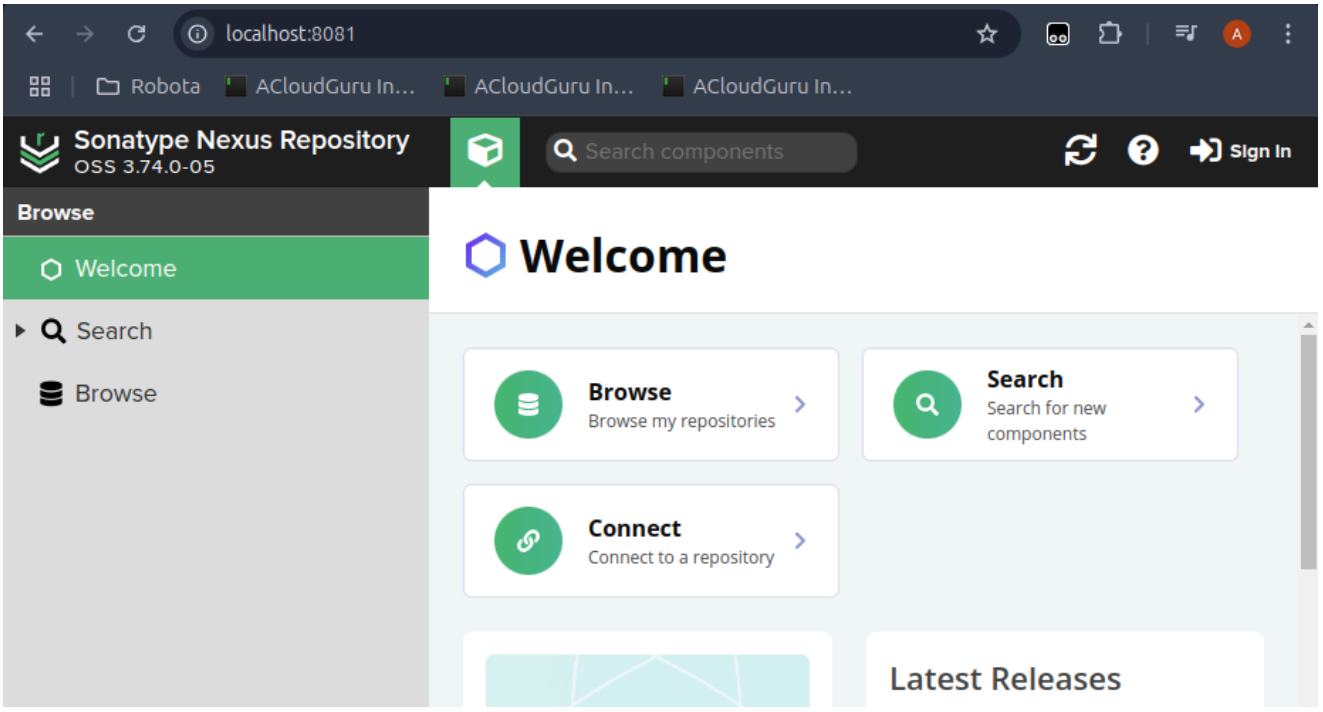
We need to have correct Java version installed (up to Java 17). I had Java 21, so I needed to install Java 17.

```
sudo apt install openjdk-17-jdk openjdk-17-jre
sudo update-java-alternatives --set <jvm path>
```

```
03:15:05 adrwal@olek-desktop-pc bin → java --version
openjdk 21.0.5 2024-10-15
OpenJDK Runtime Environment (build 21.0.5+11-Ubuntu-1ubuntu124.04)
OpenJDK 64-Bit Server VM (build 21.0.5+11-Ubuntu-1ubuntu124.04, mixed mode, sharing)
03:15:07 adrwal@olek-desktop-pc bin → update-java-alternatives --list
java-1.17.0-openjdk-amd64      1711       /usr/lib/jvm/java-1.17.0-openjdk-amd64
java-1.21.0-openjdk-amd64      2111       /usr/lib/jvm/java-1.21.0-openjdk-amd64
03:15:12 adrwal@olek-desktop-pc bin → sudo update-java-alternatives --set /usr/lib/jvm/java-1.17.0-openjdk-amd64
03:15:45 adrwal@olek-desktop-pc bin → java --version
openjdk 17.0.13 2024-10-15
OpenJDK Runtime Environment (build 17.0.13+11-Ubuntu-2ubuntu124.04)
OpenJDK 64-Bit Server VM (build 17.0.13+11-Ubuntu-2ubuntu124.04, mixed mode, sharing)
```

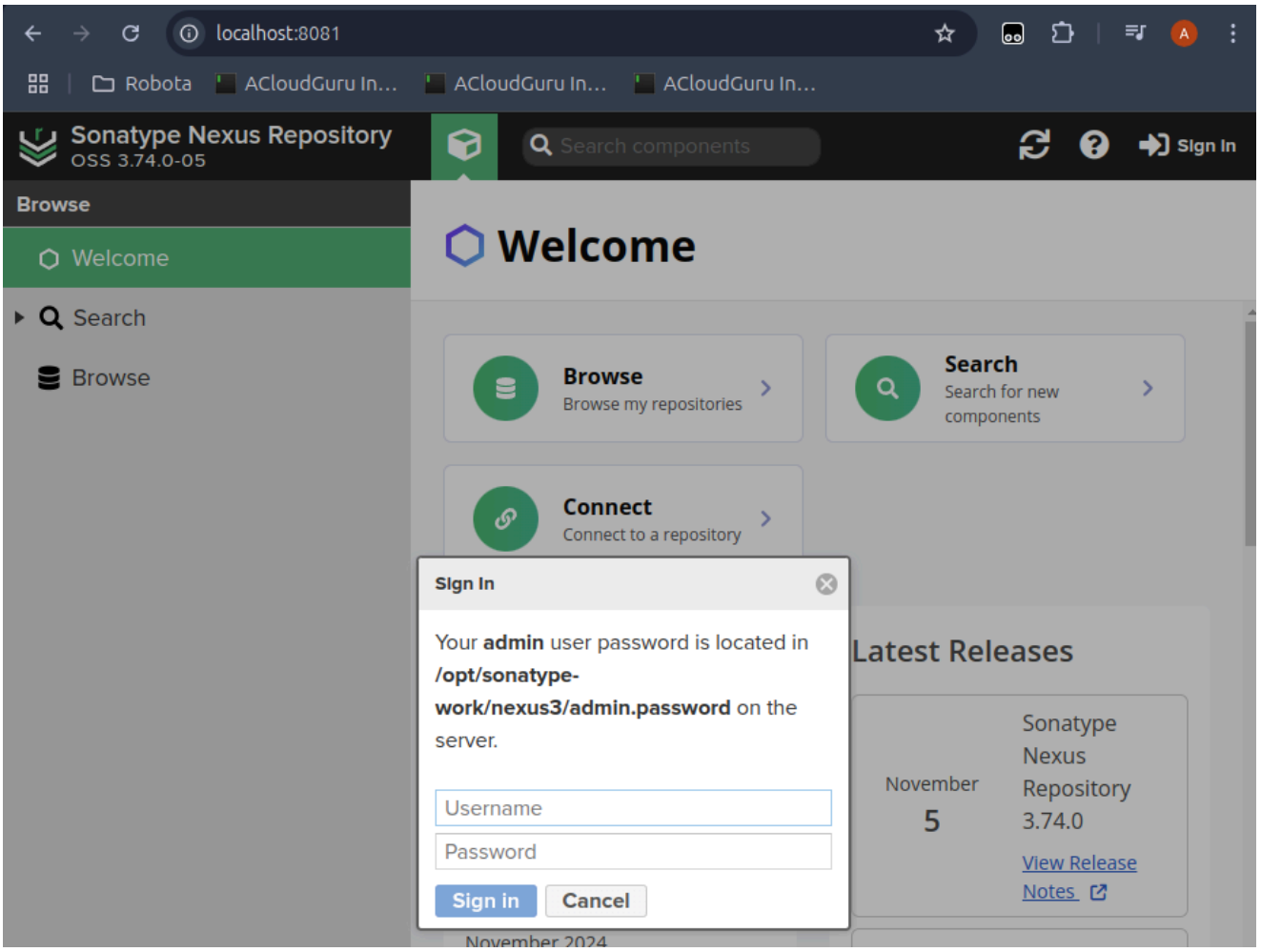
Now we can run nexus.

```
./nexus-3.74.0-05/bin/nexus run
```



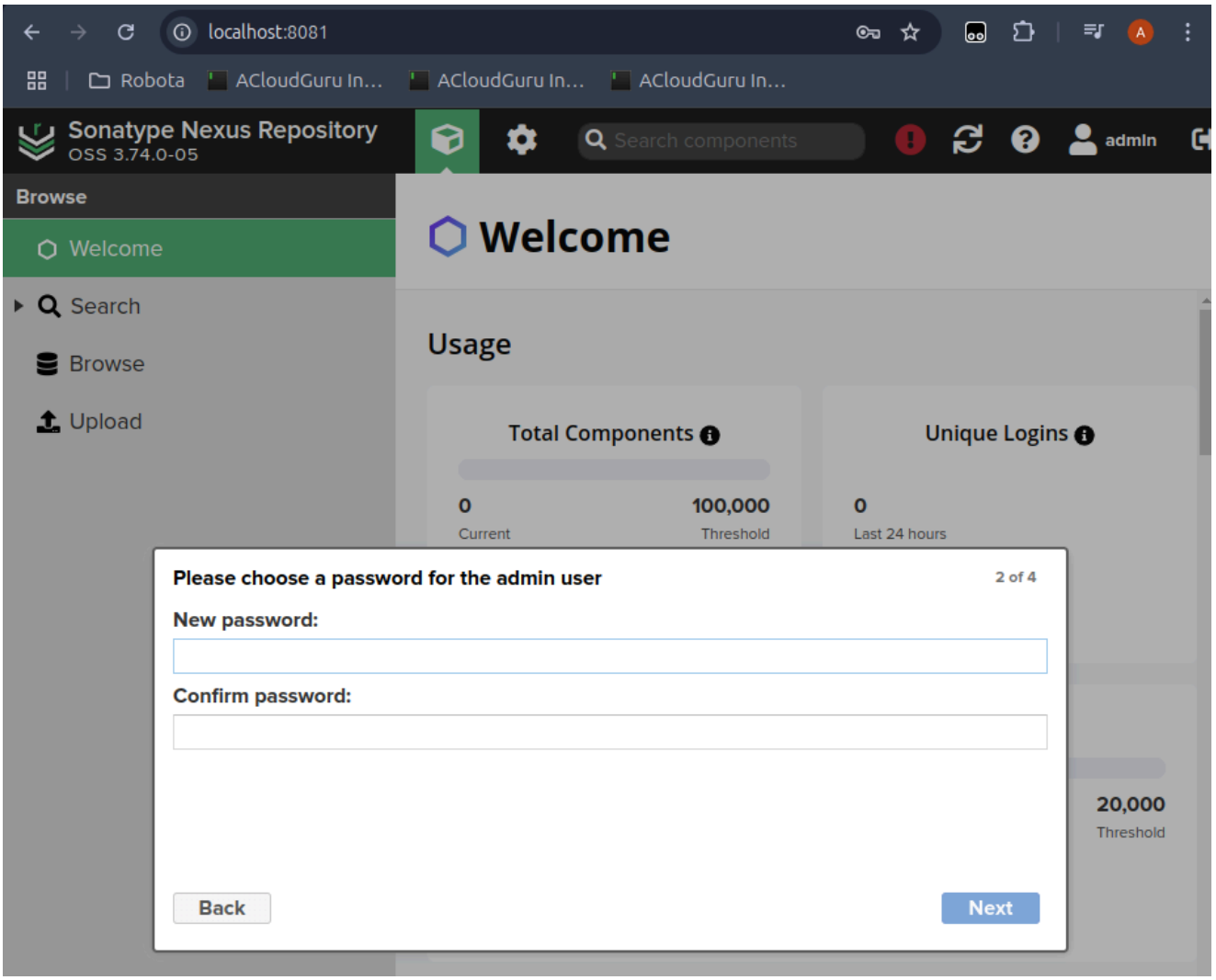
2. Configure admin user

Now click `sign in` to configure admin user.

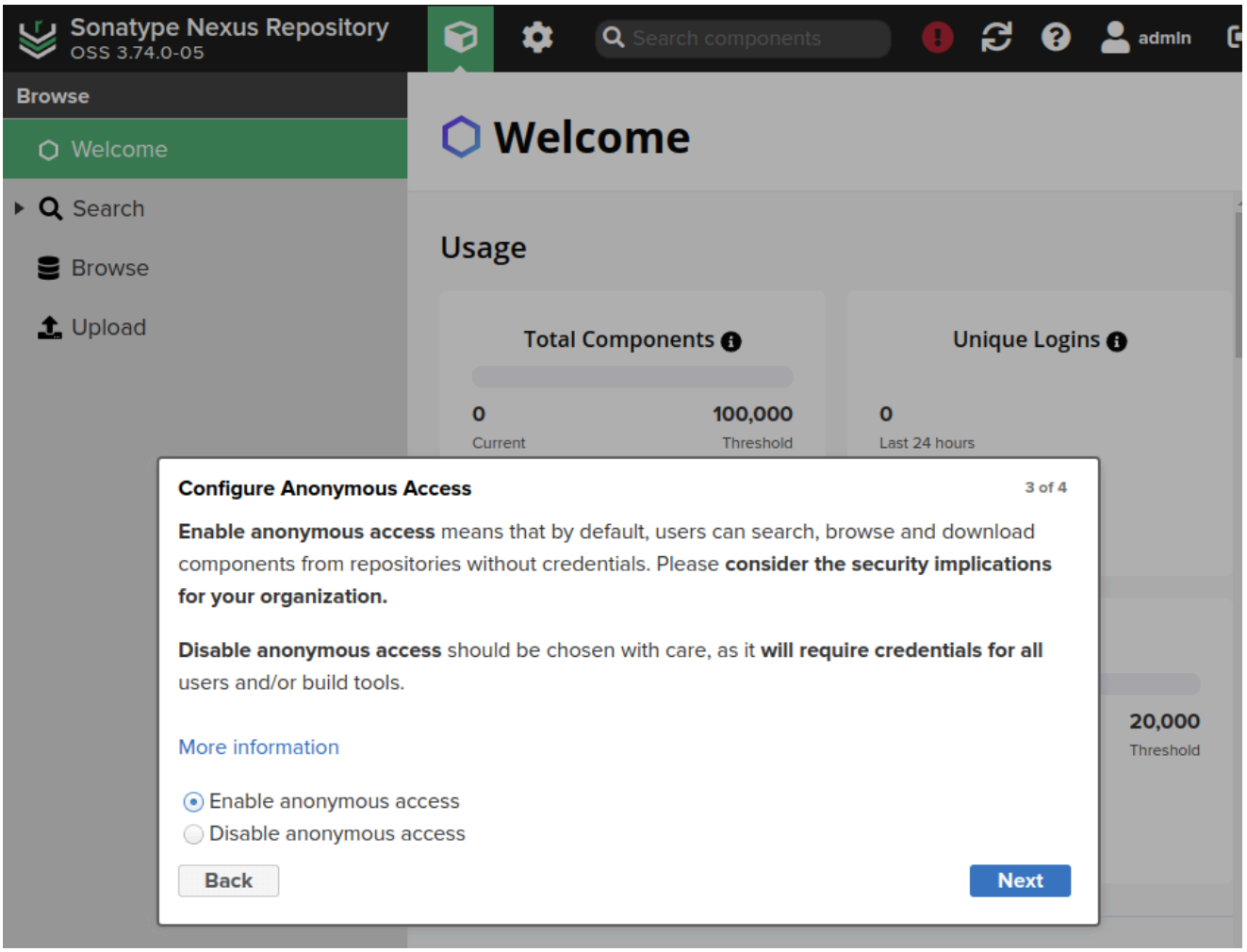


Copy password from `/opt/sonatype-work/nexus3/admin.password` and login with admin username.

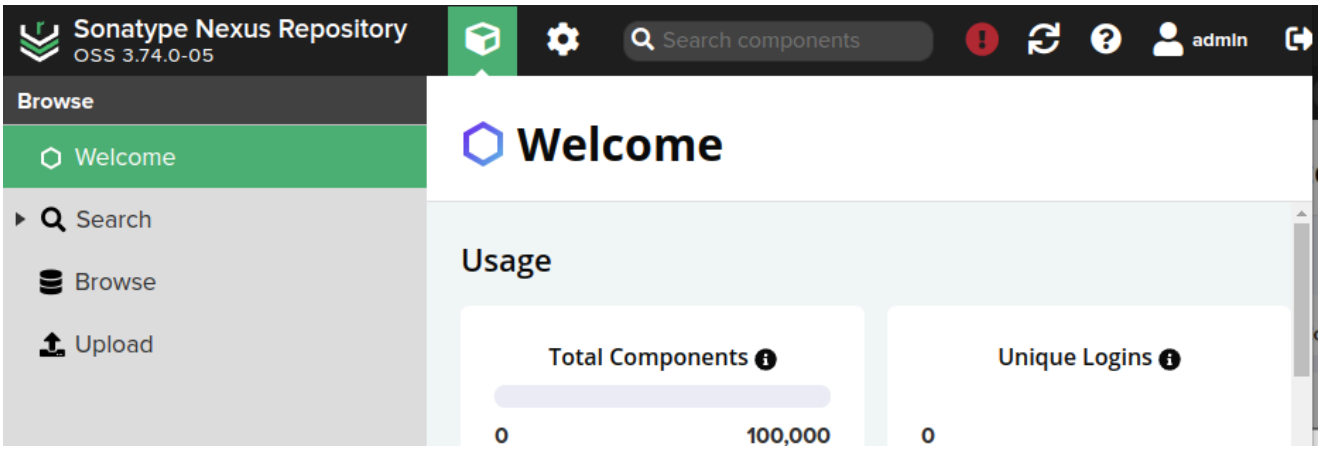
Now we have to follow setup wizard.
First, setup new password for admin.



Lastly, configure anonymous access.



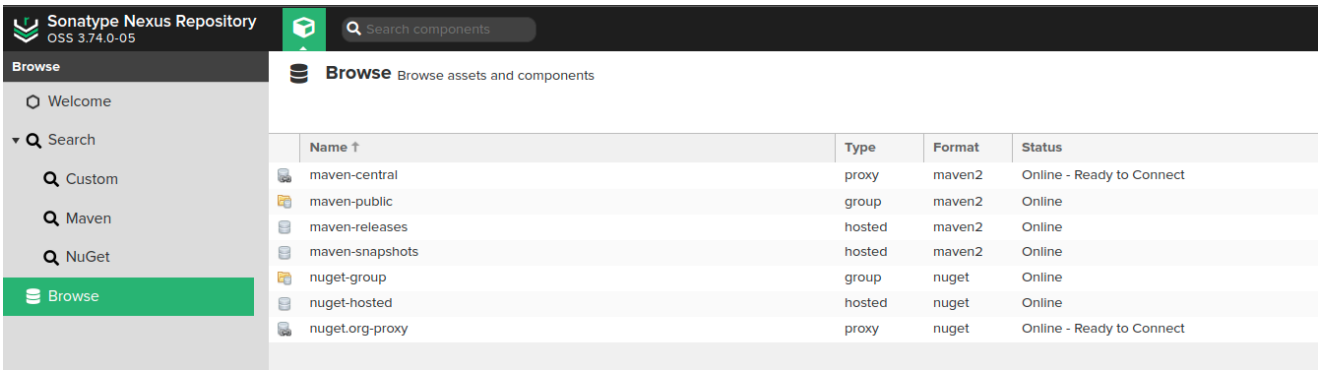
And all done, we are now logged in as admin user with newly assigned password.



3. Configure proxy repo

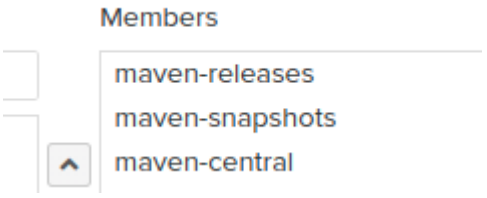
a. For Maven: Create setting.xml, maven should use it as mirror

First we need to create repositories for maven in Nexus. Good thing is that they have already been created automatically by nexus on install.



Add mirror in `~/.m2/settings.xml` pointing to our nexus repository. We use `maven-public` repository, since it is configured as a group with `maven-releases`, `maven-snapshots` and `maven-central`, meaning that it will search for packages in all those repositories starting at our private `maven-releases` and `maven-snapshots` repositories.

```
  GNU nano 7.2
<settings>
  <mirrors>
    <mirror>
      <id>nexus</id>
      <mirrorOf>*</mirrorOf>
      <url>http://localhost:8081/repository/maven-public/</url>
    </mirror>
  </mirrors>
</settings>
```



We have also specified `<mirrorOf>*</mirrorOf>`, meaning that all requests should be handled by our nexus repository.

4. Upload spring-petclinic Artifact to Nexus

a. For Maven: Use the deploy plugin in your pom.xml. You should be able to upload your artifacts to Nexus using Maven.

First we fork spring-petclinic to new repository. After we have done that we can check if we have correctly configured our mirror by running `mvn test` which will need to download some packages.

```
Downloaded from nexus: http://localhost:8081/repository/maven-public/net/java/jvnet-parent/1/jvnet-parent-1.pom
 (4.7 kB at 63 kB/s)
Downloading from nexus: http://localhost:8081/repository/maven-public/com/puppycrawl/tools/checkstyle/10.20.0/c
heckstyle-10.20.0.jar
Downloading from nexus: http://localhost:8081/repository/maven-public/info/picocli/picocli/4.7.6/picocli-4.7.6.
jar
Downloading from nexus: http://localhost:8081/repository/maven-public/org/antlr/antlr4-runtime/4.13.2/antlr4-ru
ntime-4.13.2.jar
Downloading from nexus: http://localhost:8081/repository/maven-public/commons-beanutils/commons-beanutils/1.9.4
/commons-beanutils-1.9.4.jar
Downloading from nexus: http://localhost:8081/repository/maven-public/commons-logging/commons-logging/1.2/commo
ns-logging-1.2.jar
Downloaded from nexus: http://localhost:8081/repository/maven-public/commons-logging/commons-logging/1.2/common
s-logging-1.2.jar (62 kB at 394 kB/s)
Downloading from nexus: http://localhost:8081/repository/maven-public/commons-collections/commons-collections/3
.2.2/commons-collections-3.2.2.jar
Downloaded from nexus: http://localhost:8081/repository/maven-public/org/antlr/antlr4-runtime/4.13.2/antlr4-run
time-4.13.2.jar (326 kB at 1.7 MB/s)
```

```
[WARNING] Tests run: 50, Failures: 0, Errors: 0, Skipped: 2
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:30 min
[INFO] Finished at: 2024-11-12T17:01:28+01:00
[INFO] ------------------------------------------------------------------------
05:01:28 adrwal@olek-desktop-pc spring-petclinic ±|main|→
```

We can see that packages were downloaded from our local nexus instance.

Now to deploy our artifacts we need to add `<distributionManagement>` to `pom.xml`.

```
 GNU nano 7.2
<distributionManagement>
   <repository>
     <id>nexus.releases</id>
     <name>Locally hosted repo for releases</name>
     <url>http://localhost:8081/repository/maven-releases/</url>
   </repository>
   <snapshotRepository>
     <id>nexus.snapshots</id>
     <name>Locally hosted repo for snapshots</name>
     <url>http://localhost:8081/repository/maven-snapshots/</url>
   </snapshotRepository>
 </distributionManagement>
```

In here we have added two repositories, both hosted, however one will be used for storing snapshots and other releases. Maven will automatically choose correct repository based on project version name.

We also have to give maven credentials to nexus repository. In order to do that we need to edit `~/.m2/settings.xml`

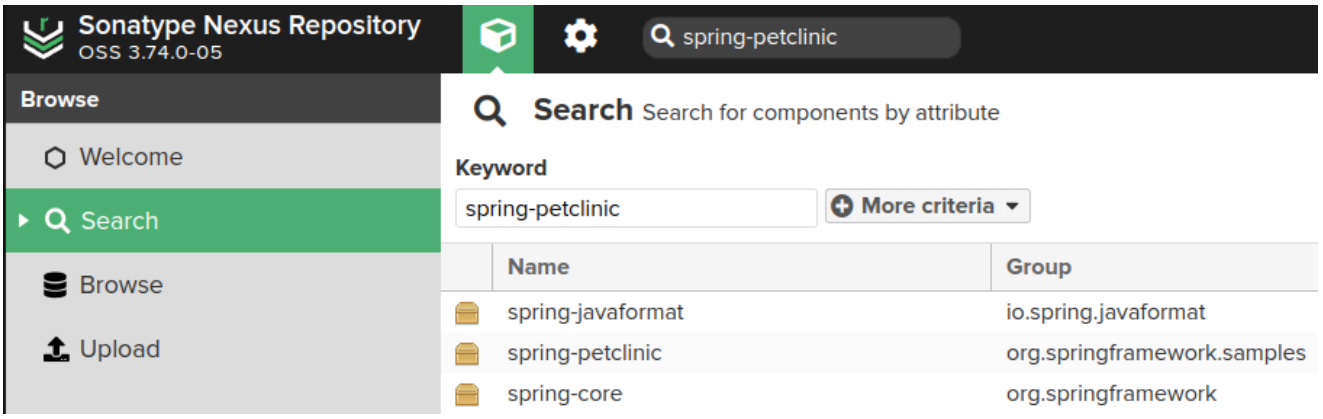
```
  GNU nano 7.2
<settings>
  <mirrors>
    <mirror>
      <id>nexus</id>
      <mirrorOf>*</mirrorOf>
      <url>http://localhost:8081/repository/maven-public/</url>
    </mirror>
  </mirrors>
  <servers>
    <server>
      <id>nexus.releases</id>
      <username>admin</username>
      <password>admin</password>
    </server>
    <server>
      <id>nexus.snapshots</id>
      <username>admin</username>
      <password>admin</password>
    </server>
  </servers>
</settings>
```

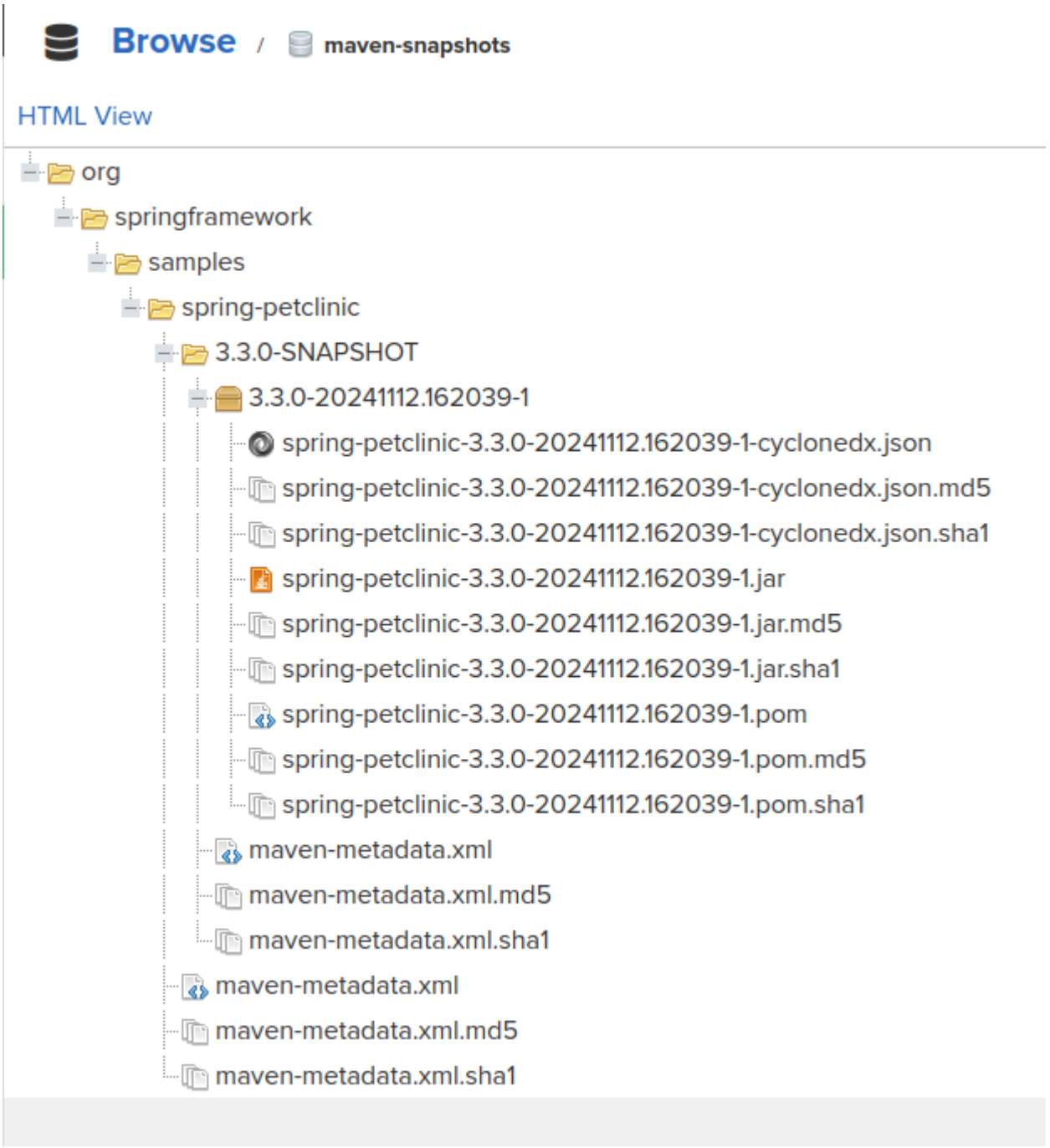
Now we can build the application and upload artifacts with `mvn deploy`.
And our artifacts have been uploaded successfully.

```
Uploading to nexus.snapshots: http://localhost:8081/repository/maven-snapshots/org/springframework/s
Uploaded to nexus.snapshots: http://localhost:8081/repository/maven-snapshots/org/springframework/sa
Uploading to nexus.snapshots: http://localhost:8081/repository/maven-snapshots/org/springframework/s
Uploaded to nexus.snapshots: http://localhost:8081/repository/maven-snapshots/org/springframework/sa
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  18.958 s
[INFO] Finished at: 2024-11-12T17:20:58+01:00
[INFO] ------------------------------------------------------------------------
05:20:58 adrwal@olek-desktop-pc spring-petclinic ±|main ✗|→
```

5. Search artifacts in Nexus

We can view uploaded artifact by searching search bar on nexus or navigating to maven-snapshots repository.


Sonatype Nexus Repository
OSS 3.74.0-05
spring-petclinic
Browse
Welcome
Search
Browse
Upload
Search Search for components by attribute
Keyword
spring-petclinic
More criteria
Name
Group
spring-javaformat
io.spring.javaformat
spring-petclinic
org.springframework.samples
spring-core
org.springframework


Browse / maven-snapshots
HTML View
org
springframework
samples
spring-petclinic
3.3.0-SNAPSHOT
3.3.0-20241112.162039-1
spring-petclinic-3.3.0-20241112.162039-1-cyclonedx.json
spring-petclinic-3.3.0-20241112.162039-1-cyclonedx.json.md5
spring-petclinic-3.3.0-20241112.162039-1-cyclonedx.json.sha1
spring-petclinic-3.3.0-20241112.162039-1.jar
spring-petclinic-3.3.0-20241112.162039-1.jar.md5
spring-petclinic-3.3.0-20241112.162039-1.jar.sha1
spring-petclinic-3.3.0-20241112.162039-1.pom
spring-petclinic-3.3.0-20241112.162039-1.pom.md5
spring-petclinic-3.3.0-20241112.162039-1.pom.sha1
maven-metadata.xml
maven-metadata.xml.md5
maven-metadata.xml.sha1
maven-metadata.xml
maven-metadata.xml.md5
maven-metadata.xml.sha1