





# Nexus repository

 Created By	 ba thanh
 Last Edited By	 ba thanh

## Summary

Nexus repository thường được dùng để làm private repository để có thể lưu trữ những dependencies, docker image private của công ty, phần này sẽ cài nexus sử dụng docker.

## Setup Sonatype Nexus3 Docker.

- Có thể lên docker hub để lựa chọn version mong muốn cho sonatype nexus. ở đây mình sẽ cài phiên bản latest:

1. Sử dụng docker volume, để lưu lại dữ liệu cho container chạy sonatype nexus, vì khi container bị tắt hoặc xóa thì dữ liệu sẽ bị xóa → sử dụng docker volume để có thể lưu lại dữ liệu trong trường hợp container bị tắt hoặc xóa thì sẽ không mất dữ liệu.

```
docker volume create --name nexus-data
```

```
docker run -d -p 8081:8081 --name nexus -v nexus-data:/nexus-data sonatype/nexus3
```

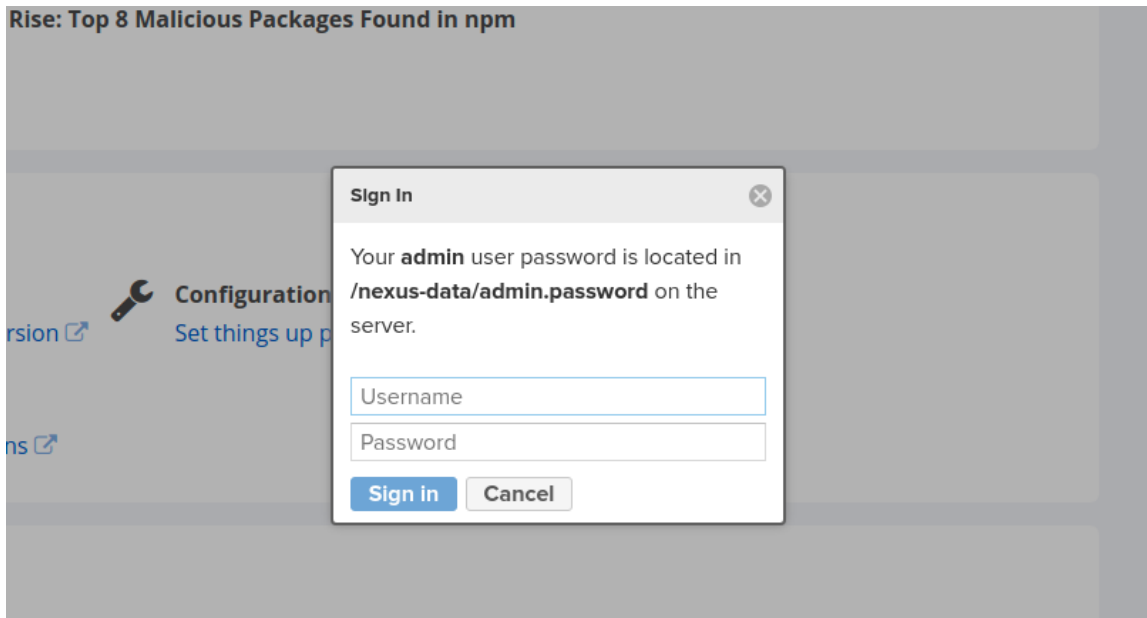
2. Tạo một thư mục ở máy host giống như là volume.

```
$ mkdir /some/dir/nexus-data && chown -R 200 /some/dir/nexus-data
```

```
$ docker run -d -p 8081:8081 --name nexus -v /some/dir/nexus-data:/nexus-data sonatype/nexus3
```

- **Login sonatype nexus3**

1. Truy cập vào: <IP>:8081



```
docker exec -it nexus cat /nexus-data/admin.password
```

```
// đây là password default của tài khoản admin, sau khi đăng nhập vào nexus  
// thì sẽ đổi tài khoản  
//password: 8693609d-9f1b-40d0-a7f8-f179d67ec6ec
```

- Ở project muốn đẩy dependency lên nexus thì sẽ cần phải tạo 1 file settings.xml, trong này chứa thông tin tài khoản nexus

```
<?xml version="1.0" encoding="UTF-8"?>  
<servers>  
  <server>  
    <id>nexus-snapshots</id>  
    <username>admin</username>  
    <password>thanhnb</password>  
  </server>  
  <server>  
    <id>nexus-releases</id>  
    <username>admin</username>  
    <password>thanhnb</password>  
  </server>  
</servers>
```

- Phần "<id>nexus-snapshots</id>" và "<id>nexus-releases</id>" sẽ dùng để mapping tài khoản và mật khẩu của được định nghĩa ở trong file pom.xml

```

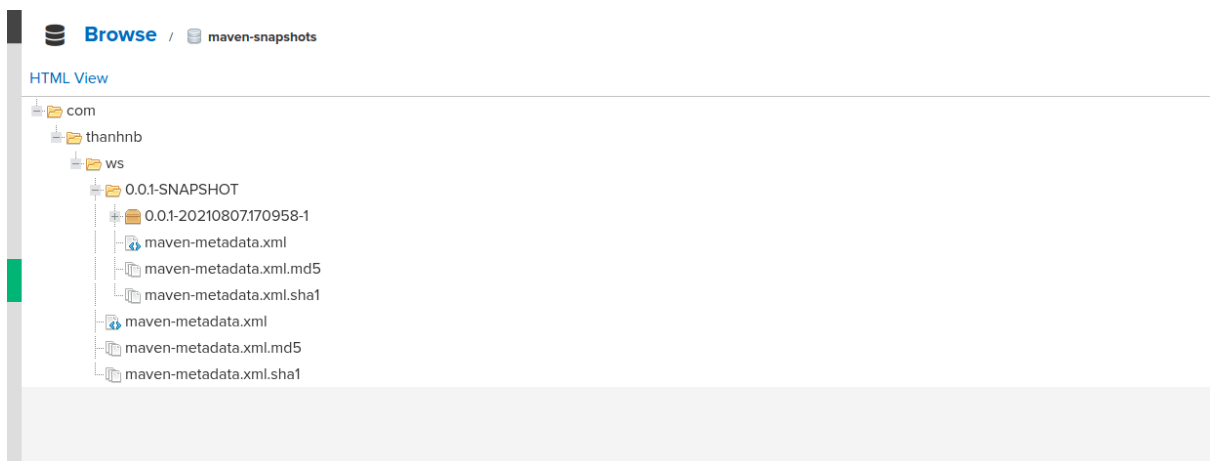
<distributionManagement>
  <snapshotRepository>
    <id>nexus-snapshots</id>
    <url>[YOUR-NEXUS-IP]:[YOUR-NEXUS-PORT]/repository/maven-snapshots/</url>
  </snapshotRepository>
  <repository>
    <id>nexus-releases</id>
    <url>[YOUR-NEXUS-IP]:[YOUR-NEXUS-PORT]/repository/maven-releases/</url>
  </repository>
</distributionManagement>

```

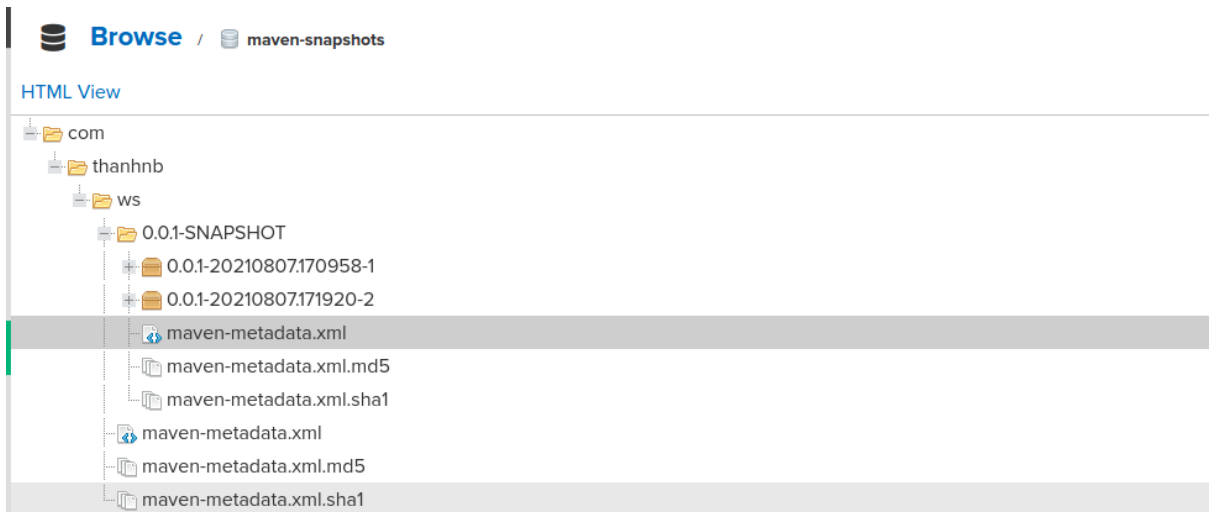
// 2 repositories "maven-snapshots" và "maven-releases" là 2 repository được nexus tạo sẵn, có thể sử dụng luôn hoặc là tạo mới 1 repositories khác.

// Ở đây có đang sử dụng 2 loại repository là "maven-snapshots" và "maven-releases", "maven-snapshots" là repository có tính thay đổi thường xuyên, vậy nên mỗi lần re-deploy dependency với loại repository này thì sẽ lại tạo ra 1 version mới được đánh theo thời gian.

// Sử dụng command "mvn clean package deploy -s settings.xml" để có thể deploy dependency lên nexus



như ảnh trên là đã đẩy được dependency lên nexus, version ở đây là "0.0.1-SNAPSHOT", ở trên cũng có thể thấy "0.0.1-20210807.170958-1", đây là những version được tạo ra khi thực hiện re-deploy, nhưng đây chỉ là đối với snapshot-repository thôi.



Khi re-deploy thì đã lại tạo thêm 1 version mới, ở đây khi được re-deploy dù không thay đổi code thì vẫn sẽ tạo ra 1 version mới "0.0.1-20210807.171920-2"

## #Sử dụng các dependencies trên nexus:

Ở project cần download dependencies từ nexus thì cần thêm link của loại repositories, ảnh dưới đây là sử dụng gradle, thì sẽ thêm ở file "build.gradle"

```
repositories {  
    maven {  
        url "http://localhost:8081/repository/maven-snapshots/"  
    }  
    mavenCentral()  
    mavenLocal()  
}
```

// Thêm dependencies:

Ảnh dưới là thêm 1 version cụ thể, được đánh version theo thời gian.

```
implementation group: 'com.thanhnb', name: 'ws', version: '0.0.1-20210807.170958-1'
```

Ảnh dưới được thêm 1 version "0.0.1-SNAPSHOT", với kiểu này thì sẽ lấy được những version theo thời gian gần nhất (lần re-deploy gần nhất)

```
implementation group: 'com.thanhnb', name: 'ws', version: '0.0.1-SNAPSHOT'
```

