Nexus

- description : nexus관련 내용 기술
- author : heon
- email : khlee@repia.com
- lastupdate : 2023-09-22

Nexus

현재 설치 위치

* URL: http://nexus.repia.com:8081
* IP: 125.132.25.172
* PORT: 8081
* 위치: /opt/nexus
* admin 계정: admin / repia12#$

Nexus 란?

- Nexus는 메이븐에서 사용할 수 있는 Repository다.
- 외부에서 dependency를 수고를 덜고, local nexus(cache)로 사용함으로써 빠르게 라이브러리를 끌어올 수 있고, 개발팀에서 사용하는 공용 라이브러리를 local nexus에 배포해서 팀간에 공유할 수 있다.

> 출처

- IBM Nexus

설치

1. 다운로드 및 설치

opt 디렉터리에 설치하는 것을 기준으로 진행합니다.

```
sudo mkdir -p /opt/nexus
cd /opt/nexus
```

tar 파일을 다운받고 압축 해제합니다.

```
sudo wget https://download.sonatype.com/nexus/3/latest-unix.tar.gz
sudo tar -xvzf latest-unix.tar.gz
```

Nexus 서비스를 생성합니다.
Nexus를 서비스로 관리하기 위해 systemd 서비스 유닛 파일을 생성합니다.

```
sudo vi /etc/systemd/system/nexus.service
```

그런 다음 아래의 내용을 추가하고 저장합니다.

```
[Unit]
Description=Nexus Repository Manager
After=network.target

[Service]
Type=forking
ExecStart=/opt/nexus/nexus-3.x.x-xx/bin/nexus start
ExecStop=/opt/nexus/nexus-3.x.x-xx/bin/nexus stop
User=nexus
Restart=on-abort

[Install]
WantedBy=multi-user.target
```

/opt/nexus/nexus-3.x.x-xx/ 부분을 Nexus 설치 디렉터리로 변경해야합니다.
또한 필요한 경우 User를 Nexus를 실행할 사용자로 변경합니다.

Nexus를 실행할 사용자를 만들고 해당 사용자에게 권한을 부여

```
sudo useradd -M -r -s /sbin/nologin nexus
sudo chown -R nexus:nexus /opt/nexus
```

서비스를 시작하고 부팅 시 자동으로 실행되도록 활성화합니다.

```
sudo systemctl start nexus
sudo systemctl enable nexus
```

Nexus가 성공적으로 실행 중인지 확인하기 위해 로그를 확인할 수 있습니다.

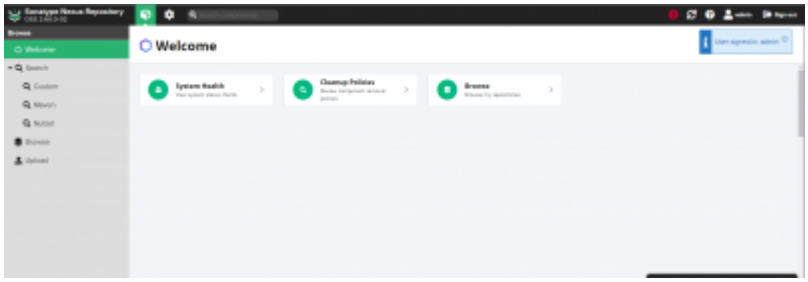
```
sudo tail -f /opt/nexus/sonatype-work/nexus3/log/wrapper.log
```

웹 브라우저에서 다음 URL을 사용하여 Nexus 웹 인터페이스에 액세스할 수 있습니다.

```
http://125.132.25.172:8081/
```

Nexus의 기본 port는 8081입니다.

Nexus 구동에 성공하면 아래와 같은 화면을 볼 수 있습니다.

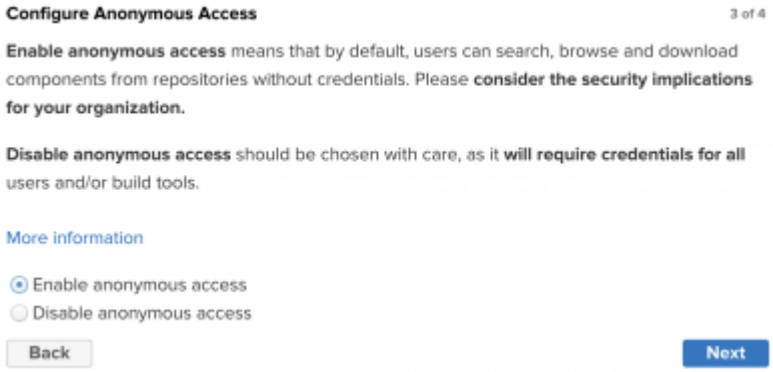


2. 최초 계정 및 설정

최초 계정의 비밀번호는 nexus-data/admin.password 파일 안에 기입되어 있습니다.
아이디: admin
비밀번호: 위의 사항 참조
* Nexus 버전이 3.17.0 버전 이전이라면 비밀번호는 admin123입니다.

로그인에 성공하면 아래와 같이 비밀번호 설정화면이 팝업됩니다.

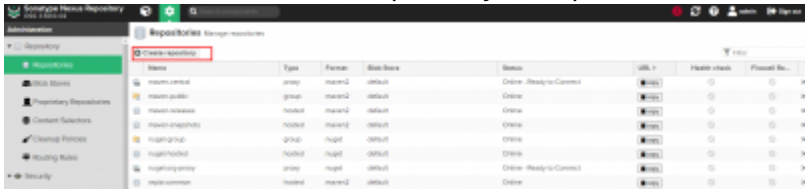
비밀번호 설정 완료 후, 익명에 대한 접근 설정 여부를 물어봅니다.



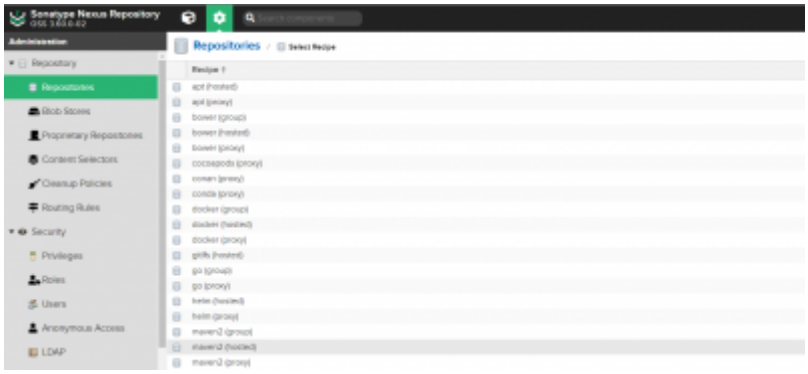
보안을 위해서 익명 사용자는 접근을 허용하지 않도록 하겠습니다.

3. 레포지토리 생성

상위의 톱나바퀴 아이콘 > Repository > Repositories > create repository를 클릭합니다.

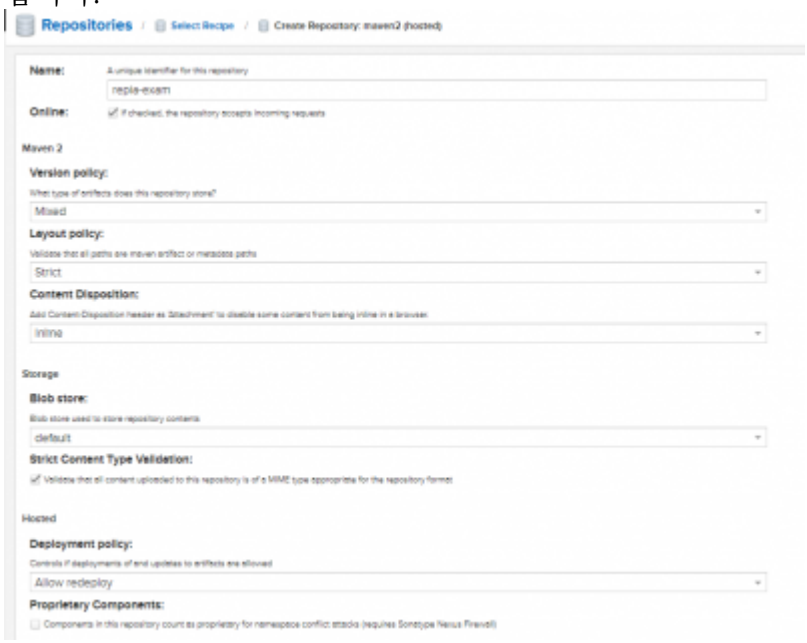


타입은 maven2(hosted)를 선택합니다.



* `proxy:` 외부 리포지토리를 프록시 해준다. 즉, 내부 시스템에서 외부에 직접 접근하지 못하는 환경을 구성할때 프록시를 주로 사용한다.
* `group:` 여러 리포지토리를 그루핑 한다.
* `hosted:` 실제 현재 nexus에서 라이브러리를 제공한다. 주로 자체 개발한 라이브러리를 서비스할 때 `hosted` 리포지토리를 설정한다.

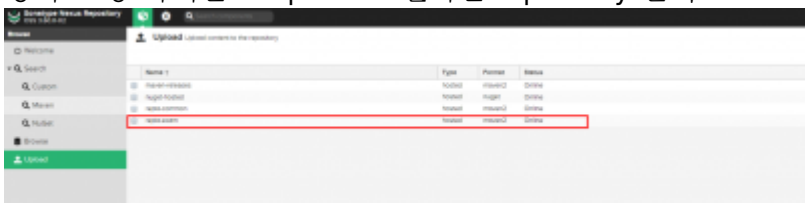
repository 설정은 아래와 같이 진행하겠습니다.
*해당 이미지는 예시일 뿐입니다. 실제로는 Version.Policy별로 구분해서 레포지토리를 생성 및 사용해야 합니다.



4. 레포지토리에 라이브러리 배포

* 해당 방법은 구동된 Nexus 웹페이지에서 바로 진행하는 방법입니다.
* 배포할 라이브러리의 pom.xml에 등록하여 바로 repository에 배포하는 방법은 아래의 ref > Nexus에 Spring Maven 라이브러리 배포 및 사용하기 링크 참조.

상자 모양 아이콘 > Upload > 원하는 repository 선택



repository에 올릴 라이브러리를 선택하고, 필드를 입력해줍니다.

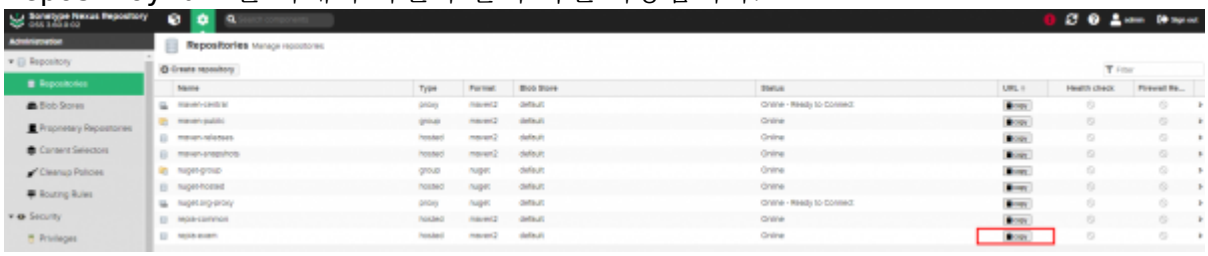
완료된 화면은 아래와 같습니다.

5. 라이브러리를 build.gralde에 적용하는 방법

build.gradle에 아래와 같이 기입합니다.

```
repositories {
    maven {
        url "repositoy url"
        credentials {
            username 'your-username'
            password 'your-password'
        }
    }
}
dependencies {
    implementation group: 'your-group-id', name: 'your-artifact-id',
version: 'version'
}
```

`repositoy url`은 아래의 사진과 같이 확인 가능합니다.



넥서스 계정의 별도 관리가 필요하다면 gradle.properties에 아래와 같이 입력한 후에 build.gradle에서 사용 가능합니다.
gradle.properties

```
version = 1.0.0
nexusUsername = repia-khlee
nexusPassword = 1234
```

```
nexusSnapshotUrl = http://125.132.25.172:8081/repository/repia-snapshot/
```

build.gradle

```
repositories {
        maven {
            url project.findProperty("nexusSnapshotUrl")
            credentials {
                username project.findProperty("nexusUsername")
                password project.findProperty("nexusPassword")
            }
        }
//    jcenter()
}
```

Tip

- Nexus에서 각 역할과, 계정을 생성하여 관리하는 Repository를 분류할 수 있습니다. 방법은 아래의 Ref > Nexus Repository 설치 및 레포지토리 생성과 역할,계정 생성 링크 참조.
- release, snapshot repository에 따라서 개별 관리가 필요합니다.

Troubleshooting

- FileDescriptors 에러가 발생한경우

Ref

- Nexus Repository 설치 및 레포지토리 생성과 역할,계정 생성
- Nexus에 Spring Maven 라이브러리 배포 및 사용하기

heon, nexus, 넥서스

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:3party:nexus&rev=1701143027

Last update: **2023/11/28 12:43**