

Sonatype nexus 로 docker registry 관리하기



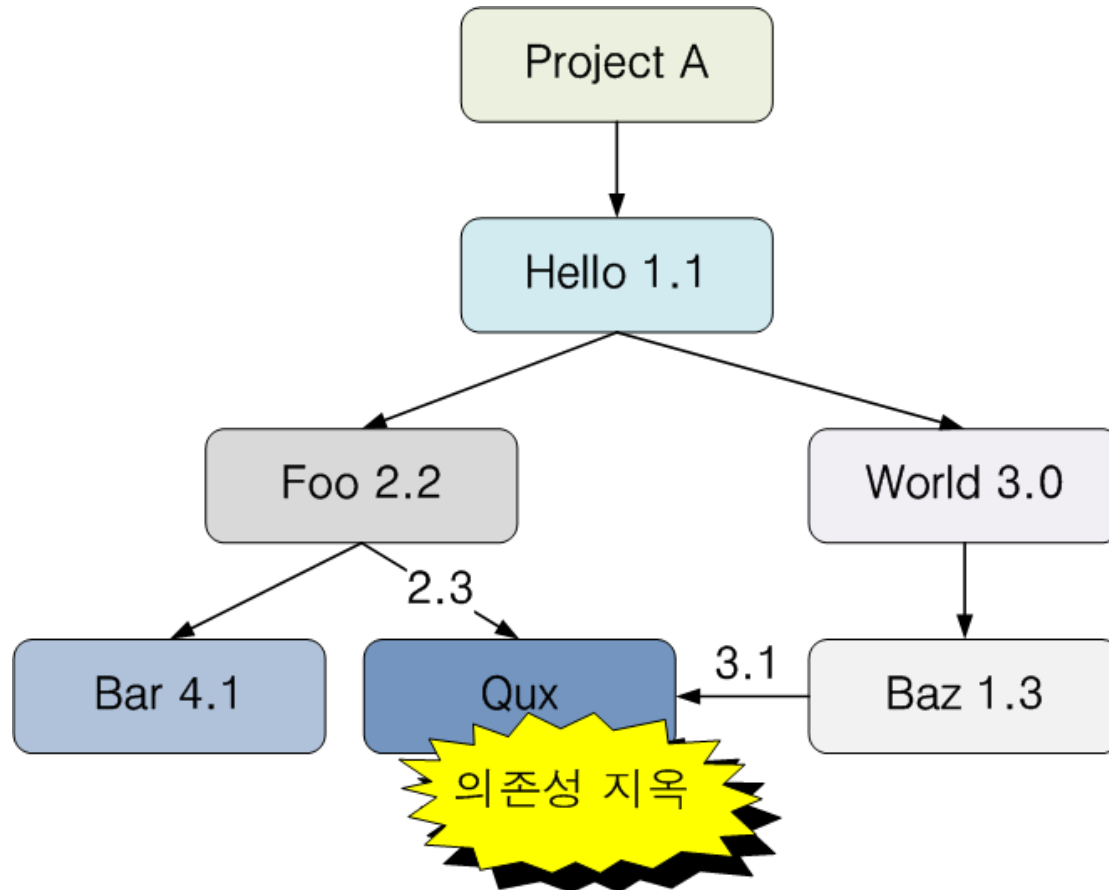
정광섭

Software 개발 패러다임 변화

- 규모가 커지고 요구사항 ↑ 일정 ↓
- 자체 개발에서 외부의 모듈과 라이브러리 사용
- 외부 모듈 조달, 이를 활용한 제품 개발, 유통의 중요성 증대

의존성 지옥(Dependency Hell)

복잡도가 높아지면서 모듈간의 의존성이 충돌



유의적 버전(Semantic Versioning)

- 호환성을 위해 라이브러리 버전 이름 짓는 규칙
- 기존 버전과 호환되지 않게 API가 변경되면 "MAJOR 버전" 을 올린다.
- 기존 버전과 호환되면서 새로운 기능이 추가 되었을 때는 "MINOR 버전"을 올린다.
- 기존 버전과 호환되면서 버그를 수정했을 경우 "PATCH 버전" 을 올린다.

패키지 관리자 (Package Manager)

- 의존성 지옥을 해결하기 위해 패키지 설치/삭제/관리 수행
- Composer, maven, npm, Bunlder 등이 있음

저장소 (Repository)

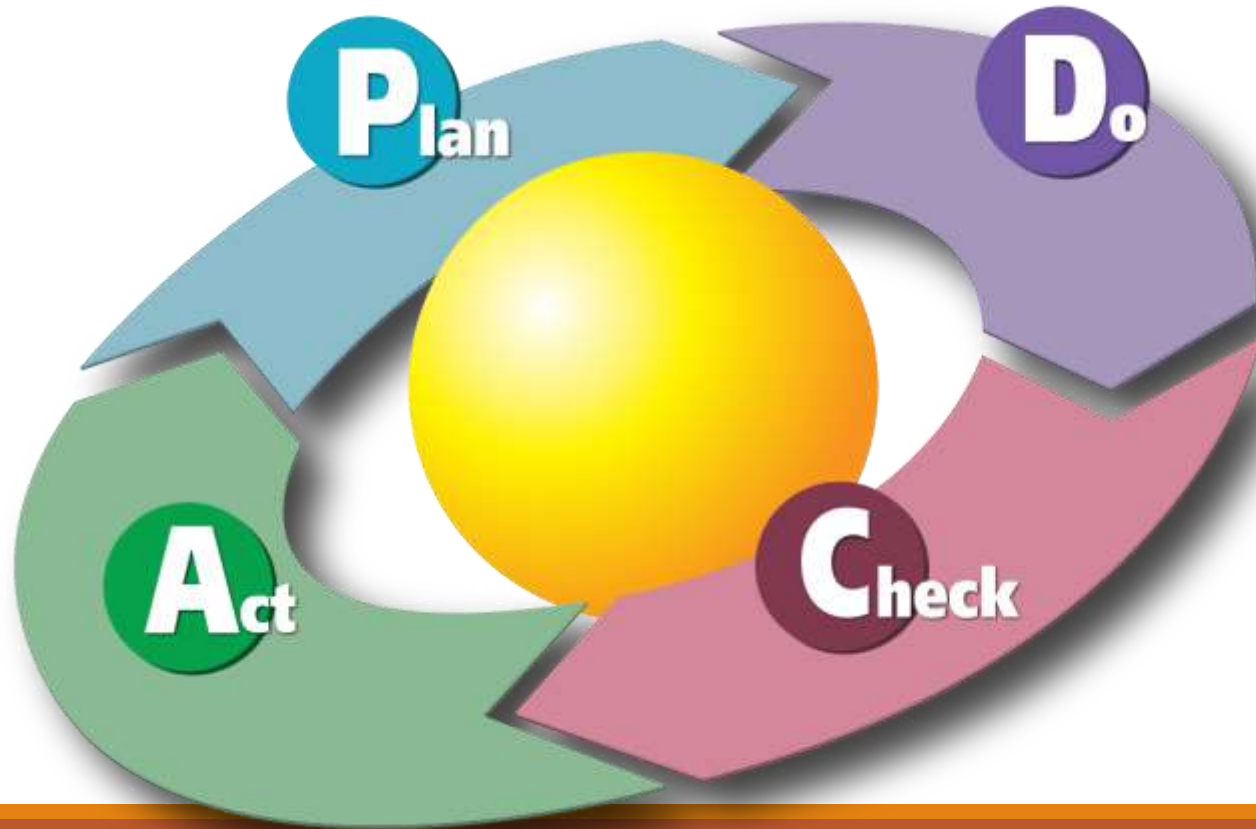
- 패키지 관리자가 패키지를 올리고 다운받고 관리하는 기능을 제공하는 중앙 서비스
- 언어마다 별도의 저장소 제공
- <https://packagist.org/> - PHP
- <https://pypi.python.org> - Python
- <https://repo1.maven.org/> - Java

저장소 관리자(Repository Manager)

- 저장소 관리 기능을 제공하는 SW
- 오늘 소개할 Nexus가 저장소 관리자

Edward Deming 박사(1990 ~ 1993)

- 품질관리의 대가로 PDCA 사이클 개발
- 만들면 팔리던 당시의 미 제조업계에서는 외면



Edward Deming 박사

- 일본에 초청받아 제조업에 품질관리 전파
- 도요타의 간반(看板), 지도카(自動化), JIT(Just-In-Time) 등에 지대한 영향

Deming 의 품질이론

- “품질 향상은 실제 비용을 낮춘다”
- “모든 직원들은 자기가 만든 제품의 품질에 대해서 스스로 책임을 지는 품질 책임체제를 구축해야 한다.”

-
- 생산라인에서 가장 중요한 것은 소비자다.
소비자를 만족시켜 주는 일이 회사의 모든
사람들이 해결해야 할 최우선 과제
 - 품질의 실패는 현장 관리자의 실패가
아니라 **경영진의 책임**

Sonatype

- Software Supply Chain
솔루션 제조사
- 데밍박사가 자동차 업계
수행한 일을 SW 업계에
적용하는 사업 모델

Scale your DevOps Pipeline

Get precise intelligence about open source components and accelerate software innovation, quality and security. **See How.**

Forrester Use DevOps and Supply Chain Principles to Automate Application Delivery Governance

How organizations are applying automated supply chain management practices to improve governance.

[READ THE REPORT](#)

Gartner How to Seamlessly Integrate Security Into DevOps

Information security architects must integrate security at multiple points into DevOps workflows.

[READ THE REPORT](#)

Say Hello to the Software Supply Chain

JS
npm
HD video

Sonatype Nexus

- Sonatype 사가 만든 저장소 관리자가 Nexus
- 검색시에는 꼭 sonatype 을 붙여주세요..(아니면 구글폰만 나와요..)

Nexus 장점 - cache

- 메인 저장소의 **cache** 역할을 수행합니다.
- 그래서 빌드시 매우 매우 빠릅니다.
- packagist 만 느린게 아니라 maven central, npmjs 등 중앙 저장소는 사용자가 많고 해외에 있어서 느립니다.

Nexus 장점 – 자체 모듈 배포

- **사설(Private) 저장소**를 만들 수 있습니다.
- 외부에 공개하고 싶지 않은 자체 개발 모듈은 여기에 올리면 됩니다.

Nexus 장점 – 상용 모듈 조달/관리

- 상용으로 판매하는 모듈은 중앙 저장소에 없습니다. (당연하겠죠)
- 상용을 사서 패키지 관리자에서 사용하려면 사설 저장소가 필요합니다.

Nexus 장점 – 재배포 불가 lib

- 상용은 아니지만 재배포 라이선스 문제가 있는 모듈도 중앙 저장소에 없습니다.
- 오라클 JDBC, MSSQL JDBC 등은 벤더에서 다운 받아서 빌드시 사용해야 하는데 매우매우 귀찮습니다.

Nexus 장점 - 보안

- 외부망 연결이 불가능한 네트워크일 경우 중앙 저장소에 연결을 못해서 빌드가 안 됩니다.
(대부분의 SI)
- nexus 서버만 외부 연결하게 설정하면 빌드가 가능합니다.

Nexus 장점 – 중앙 저장소 장애시

- 중앙 저장소가 장애나도 내부 빌드는 이상무
- [npm-leftpad](#) 삭제등 중앙 저장소에 장애나 이상이 발생해도 우리는 열심히 개발하고 빌드하면 됩니다. (아 장점이 아닌가??)

Nexus 는 대격변중

- 기존 Java 의 저장소 관리자로 잘 팔았지만.....
- Node, python 등 Script 언어의 약진
- 프론트엔드의 중요성 증대
- 이에 맞게 nexus 3 에서는 대격변중

Nexus 장점 – 다양한 저장소 지원

What's New



Expanded Repository Format Support with Nexus Repository Manager 3

Nexus Repository Manager Resources

Documentation	Mailing Lists and Live Chat
Sonatype Support	Knowledge Base
Nexus Community	Training
Email Feedback	

Nexus 설치

- DBMS 등 외부 의존성이 전혀 없음
- 오직 JDK 1.8 만 필요
- 압축 풀고 실행하면 끝이므로 docker 로 설치할 필요가 없음
- Java 만 사용한다면 nexus 2, node, docker 등도 사용할 예정이면 nexus 3 설치(저장소 포맷 호환 안됨)

Repository Type

The screenshot displays the Nexus Repository Manager (OSS 3.10.04) web interface. The left sidebar shows the 'Administration' menu with 'Repositories' selected. The main content area is titled 'Repositories / Select Recipe' and lists various repository types. A red box highlights the 'docker (group)', 'docker (hosted)', and 'docker (proxy)' entries. Another red box highlights the 'npm (group)', 'npm (hosted)', and 'npm (proxy)' entries. A yellow speech bubble points to these two groups with the text 'group, hosted, proxy 3개 저장소 유형'.

Recipe
bower (group)
bower (hosted)
bower (proxy)
docker (group)
docker (hosted)
docker (proxy)
maven2 (group)
maven2 (hosted)
maven2 (proxy)
npm (group)
npm (hosted)
npm (proxy)
nuget (group)
nuget (hosted)
nuget (proxy)
pypi (group)
pypi (hosted)
pypi (proxy)
raw (group)
raw (hosted)
raw (proxy)
rubygems (group)
rubygems (hosted)
rubygems (proxy)

group, hosted, proxy
3개 저장소 유형

Repository Type

- Proxy: 외부(중앙) 저장소 캐싱(registry.npmjs.org)
- Hosted: 자체 모듈 저장소
- Group: Proxy, Hosted 마다 각각 접근 URL 이 생기므로 패키지 매니저에서 불편함.
이 때문에 여러 저장소를 단일 URL 로 접근할 수 있도록 묶은 저장소

Docker Proxy Repository

Administration

Repository

Blob Stores

Repositories

Content Selectors

Security

Privileges

Roles

Users

Anonymous

LDAP

Realms

SSL Certificates

Support

Analytics

Logging

Log Viewer

Metrics

Support ZIP

System Information

System

Bundles

Capabilities

Email Server

HTTP

Licensing

Recent Connections

Tasks

Repositories / Select Recipe / Create Repository: docker (proxy)

Name:

A unique identifier for this repository.
docker-proxy

Online:

☒ If checked, the repository accepts incoming requests

Repository Connectors

Connectors allow Docker clients to connect directly to hosted registries, but are not always required. Consult our [documentation](#) for which connector is appropriate for your use case.

HTTP:

Create an HTTP connector at specified port. Normally used if the server is behind a secure proxy.
☒ 8082

HTTPS:

Create an HTTPS connector at specified port. Normally used if the server is configured for https.
☐

Docker Registry API Support

Enable Docker V1 API:

☒ Allow clients to use the V1 API to interact with this Repository

Proxy

Remote storage:

Location of the remote repository being proxied.
https://registry-1.docker.io

Use the Nexus truststore:

☐ Use certificates stored in the Nexus truststore to connect to external systems ☒ View certificate

Docker Index:

☐ Use proxy registry (specified above)
☒ Use Docker Hub
☐ Custom index

Docker Proxy Repository

- Remote storage
 - <https://registry-1.docker.io>
- Enable Docker V1 API 체크
- Use Docker Hub 체크

Nexus HTTP Connector

- nexus 에 등록된 저장소는 URL 로 접근 가능(<http://localhost:8081/repository/docker-proxy/>)
- URL 패턴은 Repository 밑에 저장소 이름
- 하지만 **docker는 무조건 repository:port/v1** URL이어야 함!!

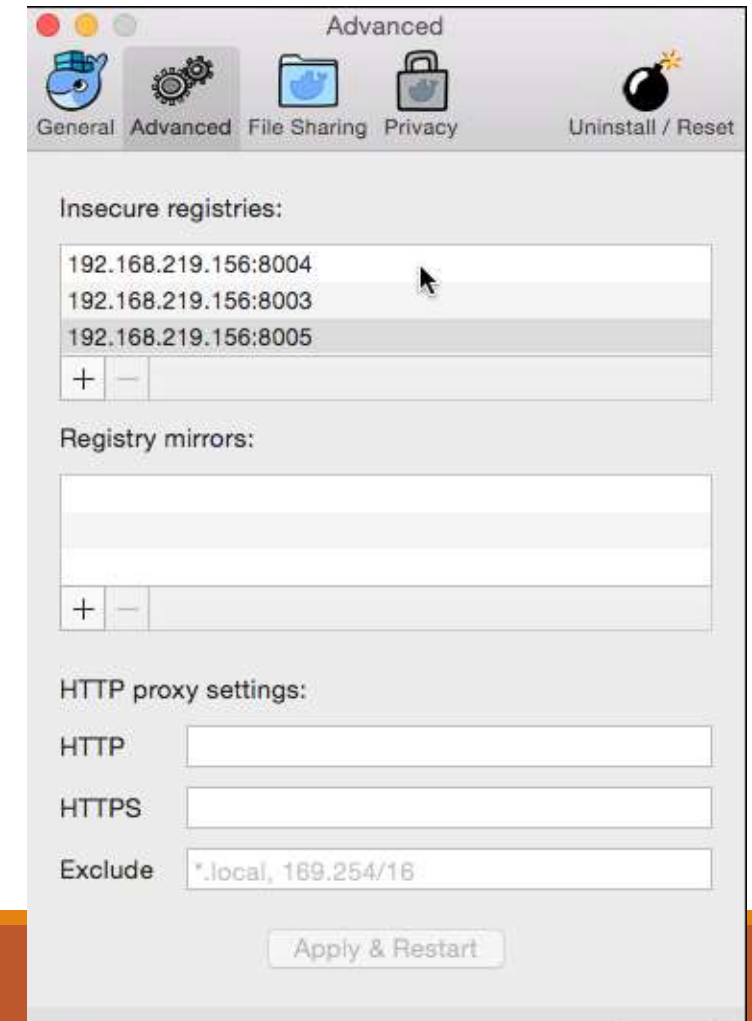
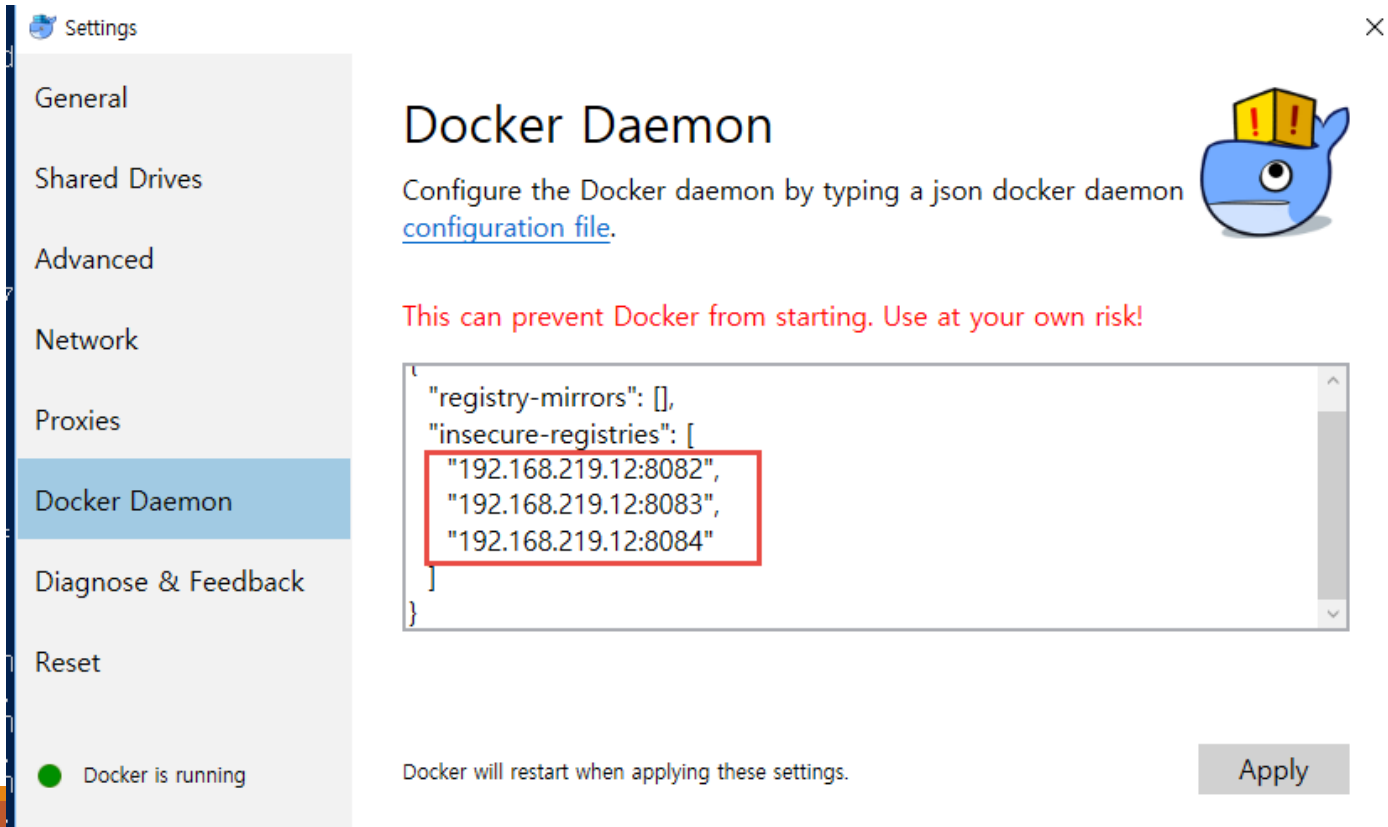
Nexus HTTP Connector

- 이 때문에 docker 저장소는 **HTTP connector** 를 만들고 해당 포트로 연결해야 함



Docker insecure repository

- docker 는 https 만 사용하므로 DOCKER_OPTS="--insecure-registry :8004" 추가 필요



Docker 연결/검색

- docker registry 와 사용법 동일
- `docker login localhost:8444 -u admin -p admin123`
- `docker search localhost:8444/postgres`
-

Docker pull/push

- `docker pull localhost:8444/ubuntu`
- `docker pull localhost:8444/postgresq:9.4`
- `docker push localhost:8444/hello-world:mytag`
- `docker push localhost:8444/ubuntu:lsof`

Docker pull/push

Nexus Repository Manager OSS 3.1.0-04

Search components

lesstif Sign out

Browse

- Welcome
- + Search
- Browse
- Assets
- Components**

Components / docker-group

Filter

	Name ↑	Group	Version	
	library/alpine	⊗	latest	>
	library/postgres	⊗	9.4	>
	library/ubuntu	⊗	14.04	>
	library/ubuntu	⊗	latest	>
	ubuntu	⊗	basic	>
	ubuntu	⊗	Isol	>

Nexus vs Docker Registry

- 동일한 역할 수행(Docker private Registry)
- nexus 는 하나로 다양한 저장소 서비스 가능
- nexus 는 관리 UI 가 있음

감사합니다.