목차

[1. 설치정보 요약 3](#_Toc70607630)

[ 운영체제 3](#_Toc70607631)

[ 하드웨어 3](#_Toc70607632)

[ 소프트웨어 3](#_Toc70607633)

[2. OS 설치 4](#_Toc70607634)

[3. 설치 순서 5](#_Toc70607635)

[ 도커 컨테이너 접속정보 5](#_Toc70607636)

[4. 로컬 Backend 개발환경 설치 5](#_Toc70607637)

[ 설치항목 5](#_Toc70607638)

[ 설치 5](#_Toc70607639)

[ 설치확인 6](#_Toc70607640)

[5. 로컬 Frontend 개발환경 설치 6](#_Toc70607641)

[ 설치항목 6](#_Toc70607642)

[ 설치 6](#_Toc70607643)

[ 설치확인 7](#_Toc70607644)

[6. 도커 설치와 도커 컨테이너 이미지 임포트 7](#_Toc70607645)

[ 설치항목 7](#_Toc70607646)

[ 설치 8](#_Toc70607647)

[ 설치확인 8](#_Toc70607648)

[7. 컨테이너 실행 8](#_Toc70607649)

[ 설치항목 8](#_Toc70607650)

[ 실행 8](#_Toc70607651)

[ IP설정 9](#_Toc70607652)

[8. 도커 nexus레지스트리 연동 설정 10](#_Toc70607653)

[9. 로컬환경에서 프로젝트 빌드 11](#_Toc70607654)

[ 백엔드 11](#_Toc70607655)

[ 프론트엔드 12](#_Toc70607656)

[10. 도커 컨테이너 빌드/배포 14](#_Toc70607657)

[ 젠킨스 실행방법 14](#_Toc70607658)

[ 배포포트 15](#_Toc70607659)

[11. DB (Redis cluster) 15](#_Toc70607660)

[ 파일 15](#_Toc70607661)

[ 설치 15](#_Toc70607662)

[ 설정 (6379,6380,6381) 16](#_Toc70607663)

[12. DB (Oracle - docker) 17](#_Toc70607664)

[ 설정 17](#_Toc70607665)

[ 접속 정보 17](#_Toc70607666)

[13. DB (mysql 5.6) 17](#_Toc70607667)

[ 파일 17](#_Toc70607668)

[ 설치 17](#_Toc70607669)

[ 설정 (my.cnf) 18](#_Toc70607670)

[ 비밀번호 변경 18](#_Toc70607671)

# 설치정보 요약

## 운영체제

* 리눅스 centos7.8[[1]](#footnote-1) ((groupinstall) Development Tools)

## 하드웨어

* CPU: 4core이상
* Memory: 10GB이상

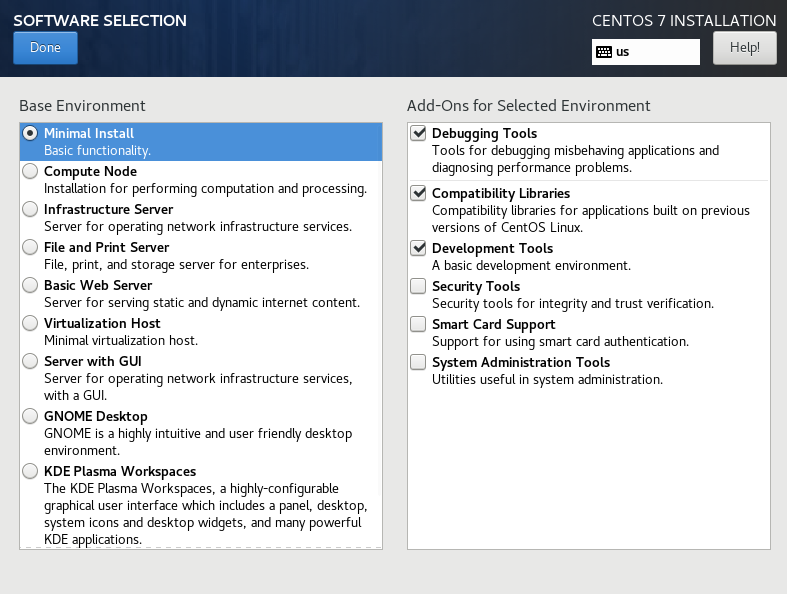
## 소프트웨어

|  |  |  |
| --- | --- | --- |
| **구분** | **이름** | **버전** |
| frontend | node | 10.16.3 |
|  | npm | 6.9.0 |
|  | yarn | 1.19.2 |
|  | nginx | 1.16.1 |
| backend | maven | 3.6.3 |
|  | java | 1.8.0\_292 |
| 도커 컨테이너 | docker 서비스 | 18.09.1 |
|  | nexus 컨테이너 | 3.20.1 |
|  | jenkins 컨테이너 | 2.263.4 |
|  | gitlab 컨테이너 | 13.11.0-ce-0 |
|  | tomcat 컨테이너 | 9.0.41 |
|  | nginx 컨테이너 | 1.16.1 |
| DB | Oracle | Xe-11g |
|  | Redis | 5.0.12 |
|  | Mysql | 5.6 |

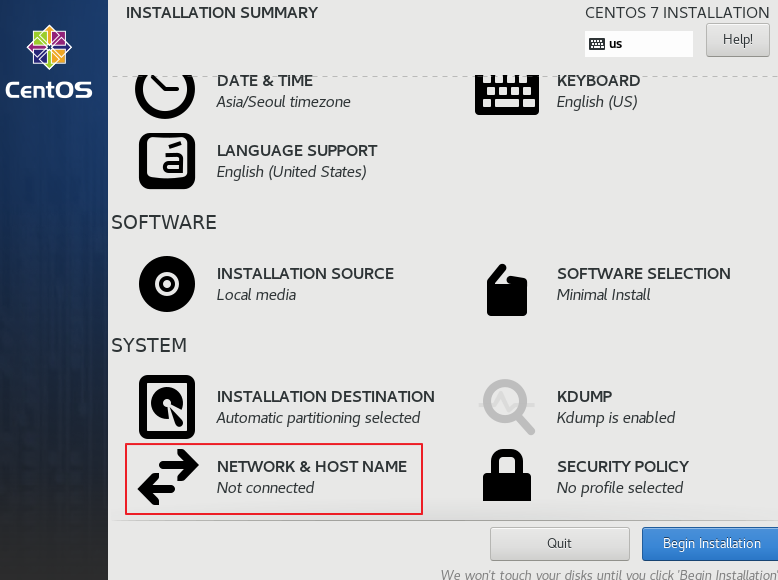
표 소프트웨어 설치전체 목록

# OS 설치

* 버전: centos 7.8[[2]](#footnote-2)
* 소프트웨어는 아래그림처럼 설정



* 인터넷은 비연결



* 나머지 옵션은 해당 기업 규정이나 설치자가 결정

# 설치 순서

1. 로컬 [backend](#_로컬_Backend_개발환경), [frontend](#_로컬_Frontend_개발환경) 환경
2. [도커 서비스 설치](#_도커_설치와_도커)
3. [도커 컨테이너 임포트](#_도커_설치와_도커)
4. [도커 서비스 실행](#_실행)
5. [도커 서비스 IP변경](#_IP설정)
6. [도커 nexus연동레지스트리 설정](#_도커_nexus레지스트리_연동)

## 도커 컨테이너 접속정보

|  |  |  |
| --- | --- | --- |
| **서비스 이름** | **접속주소** | **ID/PW** |
| gitlab | <IP>:12010 | admin / kbsys1234 |
| nexus | <IP>:10000, 5000(도커포트) | admin / kbsys1234 |
| Jenkins | <IP>:10233 | admin / admin123 |

표 도커 컨테이너 접속정보와 인증정보

# 로컬 Backend 개발환경 설치

## 설치항목

|  |  |
| --- | --- |
| **이름** | **버전** |
| java jdk | 1.8.0\_292 |
| maven | 3.6.3 |

표 백엔드 설치항목

## 설치

install.sh스크립트를 실행하여 간단히 설치할 수 있습니다. install.sh스크립트는 maven, java를 설치합니다.

|  |
| --- |
| $ sudo su  # bash quick\_install.sh  # 4번 입력 |

표 backend도구 설치 명령어

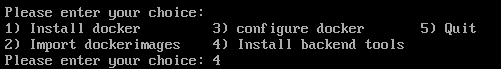


그림 backend 도구 설치 명령어 실행

## 설치확인

설치 후 maven환경변수를 로드하기 위해 로그아웃 또는 source /etc/profile.d/maven.sh 명령어를 입력하시면 됩니다.

설치가 정상적으로 되었다면 maven, java명령어가 실행됩니다.

# 로컬 Frontend 개발환경 설치

## 설치항목

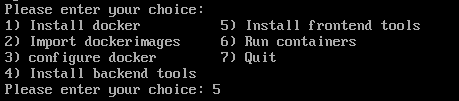
|  |  |
| --- | --- |
| **이름** | **버전** |
| nodejs | node 10.16.3 |
| npm | 6.9.0 |
| yarn | 1.19.2 |
| nginx | 1.16.1 |

표 프론트엔드 설치항목

## 설치

|  |
| --- |
| $ sudo su  # bash quick\_install.sh  # 5번 입력 |

표 Frontend 설치



## 설치확인

설치 후 nodejs환경변수를 로드하기 위해 로그아웃 또는 source /etc/profile.d/nodejs.sh 명령어를 입력하시면 됩니다.

설치가 정상적으로 되었다면 node, npm, yarn, systemctl status nginx명령어가 실행됩니다.

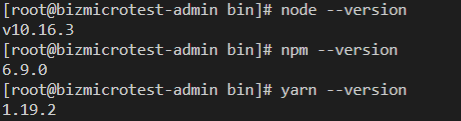


그림 node패키지 설치 성공

# 도커 설치와 도커 컨테이너 이미지 임포트

도커 설치와 도커 컨테이너 이미지를 임포트합니다.

## 설치항목

|  |  |
| --- | --- |
| **이름** | **버전** |
| docker 서비스 | 18.09.1 |
| nexus 컨테이너 | 3.20.1 |
| jenkins 컨테이너 | 2.263.4 |
| gitlab 컨테이너 | 13.11.0-ce-0 |
| tomcat 컨테이너 | 9.0.41 |
| nginx 컨테이너 | 1.16.1 |

## 설치

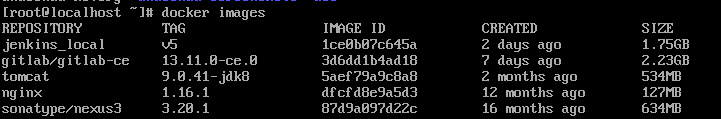
$ sudo su

# bash quick\_install.sh

# 1번 입력 후 2번 입력 ; 1번은 도커 서비스 설치, 2번은 도커이미지 임포트

## 설치확인

|  |
| --- |
| $ sudo su  # systemctl status docker  # docker images |



# 컨테이너 실행

## 설치항목

모든 초기 설정이 전부되어 있는 컨테이너를 실행합니다. 단, IP가 유동이므로 컨테이너 실행 후 IP 설정이 필요합니다.

|  |  |  |
| --- | --- | --- |
| **서비스 이름** | **버전** | **외부 포트** |
| gitlab | node 10.16.3 | 12010(HTTP) |
| nexus | 6.9.0 | 10000(HTTP), 5000(도커 레지스트리) |
| jenkins | 1.19.2 | 10233(HTTP) |

표 도커 컨테이너 설치 목록와 외부접속 포트

## 실행

컨테이너 실행 스크립트는 컨테이너가 참조하는 데이터를 압축을 풀고, 컨테이너를 실행합니다.

|  |
| --- |
| $ sudo su  # bash quick\_install.sh  # 6번입력 |

표 컨테이너 실행 명령어

[표 7]을 참고하여 nexus, gitlab, jenkins에 접속합니다 gitlab을 IP설정(표 10참고)을 하지 않으면 접속이 되지 않습니다.

|  |  |
| --- | --- |
| **서비스 이름** | **ID/PW** |
| nexus | admin / kbsys1234 |
| gitlab | admin / kbsys1234 |
| Jenkins | admin / admin123 |

표 서비스 인증정보

## IP설정

현재 컨테이너는 기존 IP로 설정되어 있어 IP변경이 필요합니다.

### gitlab IP

|  |
| --- |
| $ sudo su  # cd /data/gitlab/config  # vi ./gitlab.rb  external\_url **[오류! 하이퍼링크 참조가 잘못되었습니다.](http://<ip>:12010)**  # docker restart kbsys-gitlab ; gitlab 도커 컨테이너 재실행하여 설정 적용 |

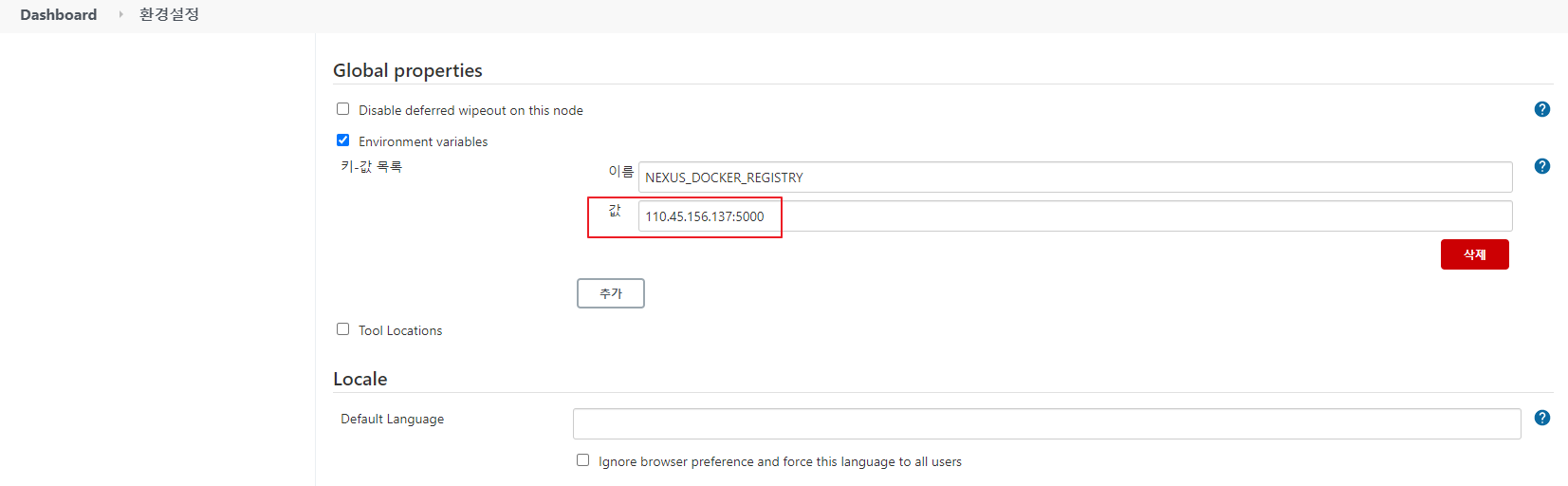
표 gitlab IP설정

### 젠킨스 IP

nexus에 있는 도커 저장소를 사용하기 위해 접속정보를 설정합니다. 접속정보는 젠킨스 전역환경변수에 설정되어 있습니다.



nexus 컨테이너 <IP>:5000으로 변경합니다.



# 도커 nexus레지스트리 연동 설정

nexus레지스트리와 통신하기 위해 도커설정(/etc/docker/daemon.json)이 필요합니다.

$ sudo su

# vi /etc/docker/daemon.json

# 로컬환경에서 프로젝트 빌드

## 백엔드

### pom.xml에서 nexus IP를 수정

|  |
| --- |
| <distributionManagement>  <repository>  <id>nexus-repo-public</id>  <url>**http://110.45.156.137:8081**/repository/maven-public/</url>  </repository>  <snapshotRepository>  <id>nexus-repo</id>  <name>private-repo</name>  <url>http://110.45.156.137:8081/repository/maven-snapshots/</url>  </snapshotRepository>  </distributionManagement> |

### bizmicro-api-pom.xml에서 nexus IP를 수정

|  |
| --- |
| <distributionManagement>  <snapshotRepository>  <id>nexus-repo</id>  <name>private-repo</name>  <url>**http://110.45.156.137:8081**/repository/maven-snapshots/</url>  </snapshotRepository>  </distributionManagement>  <repositories>  <repository>  <id>nexus</id>  <name>maven repository</name>  <url>http://110.45.156.137:8081/repository/maven-public/</url>  </repository>  </repositories>  <pluginRepositories>  <pluginRepository>  <id>nexus</id>  <url>**http://110.45.156.137:8081**/repository/maven-public/</url>  </pluginRepository>  </pluginRepositories> |

### bizmicro-daemon-pom.xml에서 nexus IP를 수정

|  |
| --- |
| <distributionManagement>  <snapshotRepository>  <id>nexus-repo</id>  <name>private-repo</name>  <url>http://110.45.156.137:8081/repository/maven-snapshots/</url>  </snapshotRepository>  </distributionManagement> |

### bizmicro-sockjs-pom.xml에서 nexus IP를 수정

|  |
| --- |
| <distributionManagement>  <snapshotRepository>  <id>nexus-repo</id>  <name>private-repo</name>  <url>**http://110.45.156.137:8081**/repository/maven-snapshots/</url>  </snapshotRepository>  </distributionManagement> |

## 프론트엔드

yarn run build명령어로 프로젝트(uFnt, aFnt)를 빌드합니다.

|  |
| --- |
| $ sudo su  # cd uFnt 또는 aFnt  # yarn run build |

표 reactjs 프로젝트 빌드

설치가 성공적으로 끝나면 Done메시지가 나옵니다.

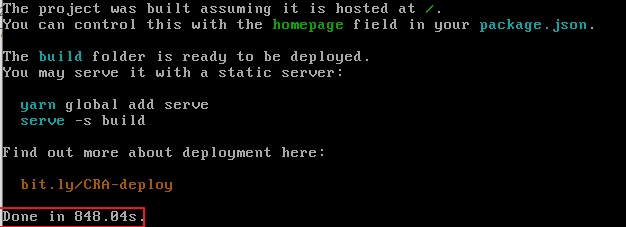


그림 react빌드 성공

heap오류가 발생한 경우 heap메모리 할당 크기를 증가시켜줍니다.

|  |
| --- |
| "scripts": {  "start": "set port=3300&& react-scripts --max-old-space-size=4096 start",  "build": "react-scripts **--max\_old\_space\_size=10240** build",  "test": "react-scripts test",  "eject": "react-scripts eject"  }, |

표 package.json에서 heap메모리 할당크기 확장

# 도커 컨테이너 빌드/배포

## 젠킨스 실행방법

젠킨스 컨테이너를 사용하여 개발자의 소스코드를 자동배포합니다. 젠킨스에 로그인 한 후 bizmicro젠킨스 폴더를 클릭합니다.

* 아이디: admin
* 비빌번호: admin123

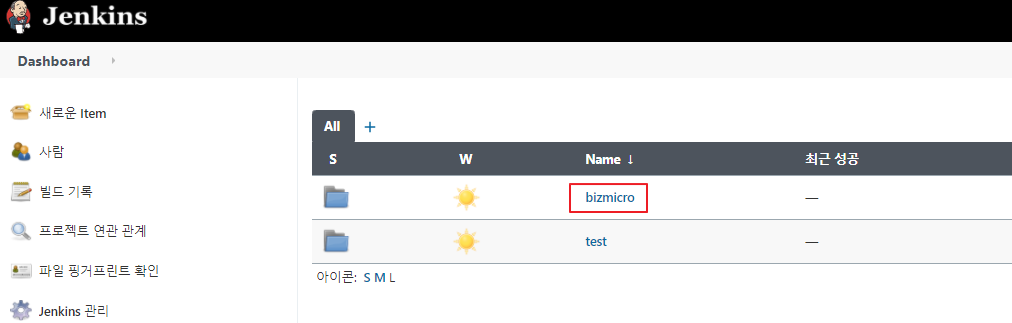


그림 프로젝트가 있는 젠킨스 잡

bizmicro폴더에는 was, web폴더가 있습니다. was폴더는 백엔드프로젝트, web폴더는 프론트엔드 프로젝트를 빌드/배포합니다

**주의:** 백엔드잡이 성공적으로 빌드되기 위해서 0-root는 한번 이상 성공적으로 빌드되어야 한다.

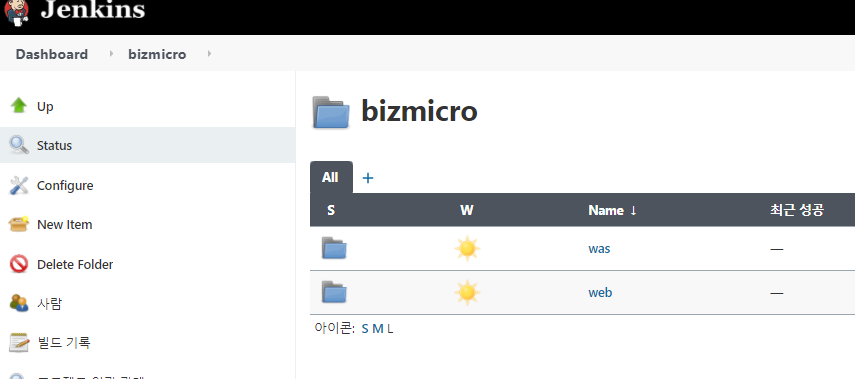


그림 bizmicro폴더 구조

## 배포포트

|  |  |  |
| --- | --- | --- |
| 구분 | 서비스 | 접속포트 |
| backend | api | 11400 -> 8080  11401 -> 9911 |
| daemon | 11410 -> 8080  11411 -> 9912 |
| sockjs | 11420 -> 8080  11421 -> 9910 |
| frontend | user | 11510 -> 80 |
| admin | 11610 -> 80 |

# DB (Redis cluster)

## 파일

|  |  |
| --- | --- |
| **이름** | **역할** |
| redis-5.0.12.tar.gz | * Redis 설치 파일 |

## 설치

$ mkdir /data/kbsys/redis

$ cd /data/kbsys/redis

$ cp redis-5.0.12.tar.gz /data/kbsys/redis

$ tar xvf redis-5.0.12.tar.gz

$ cd redis-5.0.12

$ cd deps

$ make hiredis jemalloc linenoise lua

$ cd ..

$ make install

## 설정 (6379,6380,6381)

Vi /data/kbsys/redis/conf/redis\_(포트정보).conf

port 6379

cluster-enabled yes

cluster-config-file nodes.conf

cluster-node-timeout 5000

appendonly no

requirepass kbsys1234

$ vi redis\_start.sh

#!/bin/sh

redis-server /data/kbsys/redis/conf/redis\_6379.conf

redis-server /data/kbsys/redis/conf/redis\_6380.conf

redis-server /data/kbsys/redis/conf/redis\_6381.conf

$ vi redis\_stop.sh

#!/bin/sh

redis-cli -c -p 6379 shutdown

redis-cli -c -p 6380 shutdown

redis-cli -c -p 6381 shutdown

redis 클러스터 구성 명령어

redis-cli --cluster create <서버 아이피>:6379 <서버 아이피>:6380 <서버 아이피>:6381

# DB (Oracle - docker)

## 설정

$ docker load -I oracle-xe-11g.tar

$ docker run --name oracle11g -d -p 1521:1521 -v /u01/app/oracle:/<oracle 데이터 저장 디렉토리명> jaspeen/oracle-xe-11g:latest

$ docker exec -it oracle11g sqlplus

User-name: system

Password: oracle

## 접속 정보

사용자이름: system

비밀번호: oracle

호스트이름: localhost

포트: 1521

SID: xe

# DB (mysql 5.6)

## 파일

|  |  |
| --- | --- |
| **이름** | **역할** |
| mysql5.6-for-bizmicro.tar | * Mysql (ver 5.6) 설치 파일 |

## 설치

$ tar xvf mysql5.6-for-bizmicro.tar

$ cd mysql-5.6/

$ rpm -ivh \*

$ systemctl start mysqld

$ mysql -u root -p (비번 초기 엔터)

## 설정 (my.cnf)

[client]

default-character-set=utf8

[mysql]

default-character-set=utf8

[mysqld]

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock

character-set-server=utf8

collation-server=utf8\_general\_ci

init\_connect=SET collation\_connection = utf8\_general\_ci

init\_connect=SET NAMES utf8

character-set-client-handshake = FALSE

skip-character-set-client-handshake

[mysqldump]

default-character-set=utf8

# Disabling symbolic-links is recommended to prevent assorted security risks

symbolic-links=0

# Recommended in standard MySQL setup

sql\_mode=NO\_ENGINE\_SUBSTITUTION,STRICT\_TRANS\_TABLES

[mysqld\_safe]

log-error=/var/log/mysqld.log

pid-file=/var/run/mysqld/mysqld.pid

## 비밀번호 변경

$ Mysql -u root -p

$ use mysql;

$ update user set password=password(‘<비밀번호>’) where user=’root’;

$ flush privileges;

1. centos 다운로드 링크: http://mirrors.oit.uci.edu/centos/7.8.2003/isos/x86\_64/CentOS-7-x86\_64-Everything-2003.iso [↑](#footnote-ref-1)
2. centsOS 다운로드링크: http://mirrors.oit.uci.edu/centos/7.8.2003/isos/x86\_64/CentOS-7-x86\_64-Everything-2003.iso [↑](#footnote-ref-2)