



Software Version: 2.0.0

Installation Guide v2.0.0

Document Release Date: July 2017
Software Release Date: July 2017



저작권 및 면책조항

본 매뉴얼은 기술상의 부정확함 또는 글자상의 오류를 포함하고 있으며, 저자는 여기에 관한 어떠한 책임도 지지 않습니다.

본 매뉴얼의 내용은 정기적으로 변경 또는 추가 될 수 있으며, 그에 따른 변경본은 본 매뉴얼에 "New Edition"의 형태로 순차적으로 추가될 것입니다. 사전 공지 없이 언제든지 본 매뉴얼에 기재된 제품 및 프로그램을 변경 또는 수정할 수 있습니다.

nCube:Lavender 단말 플랫폼 소스코드는 하기의 라이선스 정책에 따라 배포됩니다.

- OCEAN 기반 오픈 소스코드는 3-clause BSD-style 라이선스를 기반으로 배포한다. 3-clause BSD-style 라이선스에 따라, 소스코드 파일에 저작권 헤더를 유지하는 동안 OCEAN 기반 오픈 소스 코드는 상업용 또는 비상업용 시스템에 자유롭게 사용할 수 있다.
- OCEAN 기반 오픈 소스 라이선스는 개발된 소스코드를 다른 사람과 공유하는 것을 요구하지는 않는다. OCEAN기반 오픈 소스를 활용하여 개발한 소스코드는 개발자의 것이고 누구와 공유할 의무는 없다.
- 누구나 OCEAN의 오픈 소스 발전에 공헌할 수 있다. 그러나, 개발된 소스코드가 OCEAN에 공헌되기를 원한다면 이 코드는 OCEAN의 라이선스 정책을 따르도록 요구된다.

Copyright (c) 2017, OCEAN
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

내 용

1. nCube:Lavender 소개.....	5
1.1. 개요.....	5
1.2. nCube:Lavender 디렉토리 구조	6
2. nCube:Lavender 설치	오류! 책갈피가 정의되어 있지 않습니다.
2.1. 설치 개요	오류! 책갈피가 정의되어 있지 않습니다.
2.2. nCube:Lavender 구동환경 구축 (Windows).....	오류! 책갈피가 정의되어 있지 않습니다.
2.2.1. MySQL 설치	오류! 책갈피가 정의되어 있지 않습니다.
2.2.2. Node JS 설치	오류! 책갈피가 정의되어 있지 않습니다.
2.3. nCube:Lavender 설치	오류! 책갈피가 정의되어 있지 않습니다.
3. nCube:Lavender 구동	31

1. nCube:Lavender

1.1. 개요

nCube:Lavender는 oneM2M에서 독립 단말 장치로 정의되고 있으며 실환경에서는 디바이스의 standalone 장치로써 리소스를 독립적으로 저장할 수 있으나 차일드 리소스(remoteCSE)를 가질 수 없는 단말용 플랫폼이다. oneM2M에서는 ASN-CSE 기능을 수행한다. nCube:Lavender는 CSE를 포함하고 있기 때문에 nCube이지만 서버로써의 CSE 기능을 수행하므로 Mobius와 소스를 공유한다. 즉, nCube:Lavender와 Mobius와의 차이는 소스를 공유하기 때문에 물리적 HW 차이 외에는 CSE 동작을 위한 기능 상의 차이는 없다. 단지 nCube:Lavender는 ASN-CSE이기 때문에 IN-CSE(Mobius)에 자신(Lavender)을 등록하는 절차와 Child로 remoteCSE를 추가할 수 없을 뿐이다. MQTT 지원을 위해 nCube:Lavender는 MQTT broker로 IN-CSE의 Broker 서버를 그대로 이용하기 때문에 nCube:Lavender 설치 환경을 구축할 때 Broker 설치는 제외된다.

다음 그림은 nCube:Lavender를 사용할 때의 연동 구조를 보여준다. 실제 적용에서 Lavender를 사용할 것인지 Thyme을 사용할 것인지는 적용 분야에 따라 리소스를 로컬에서 관리를 할 필요가 있느냐 아니면 단순 전달이냐에 따라 선택하면 될 것이다.

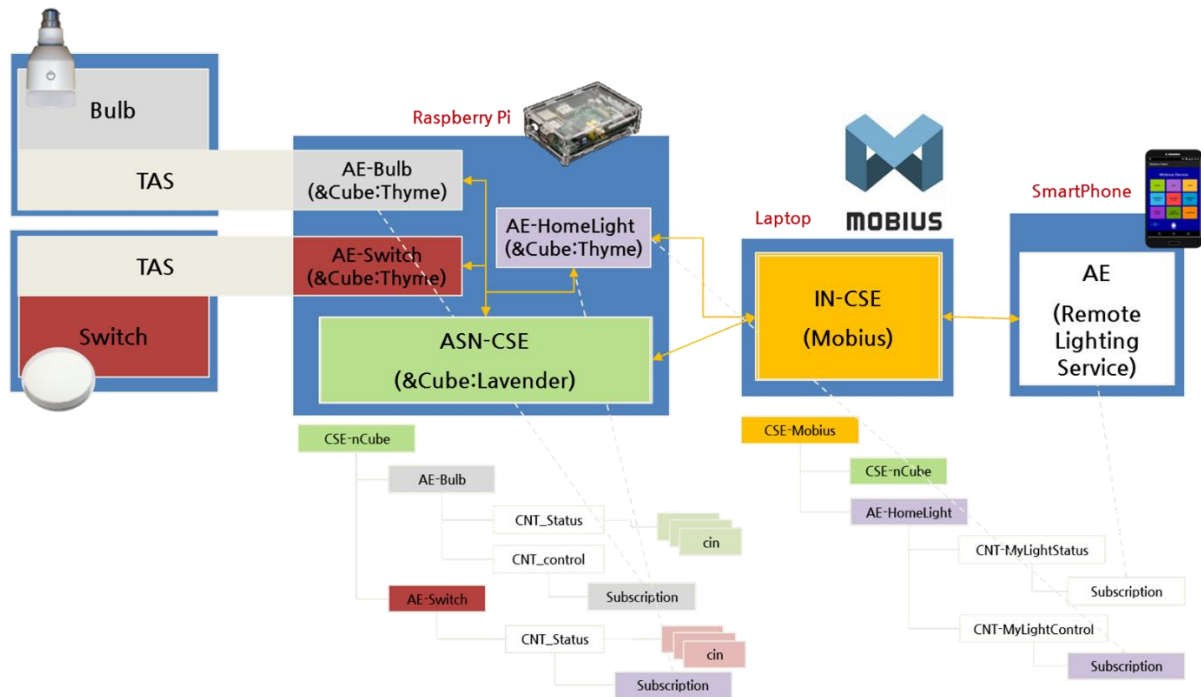
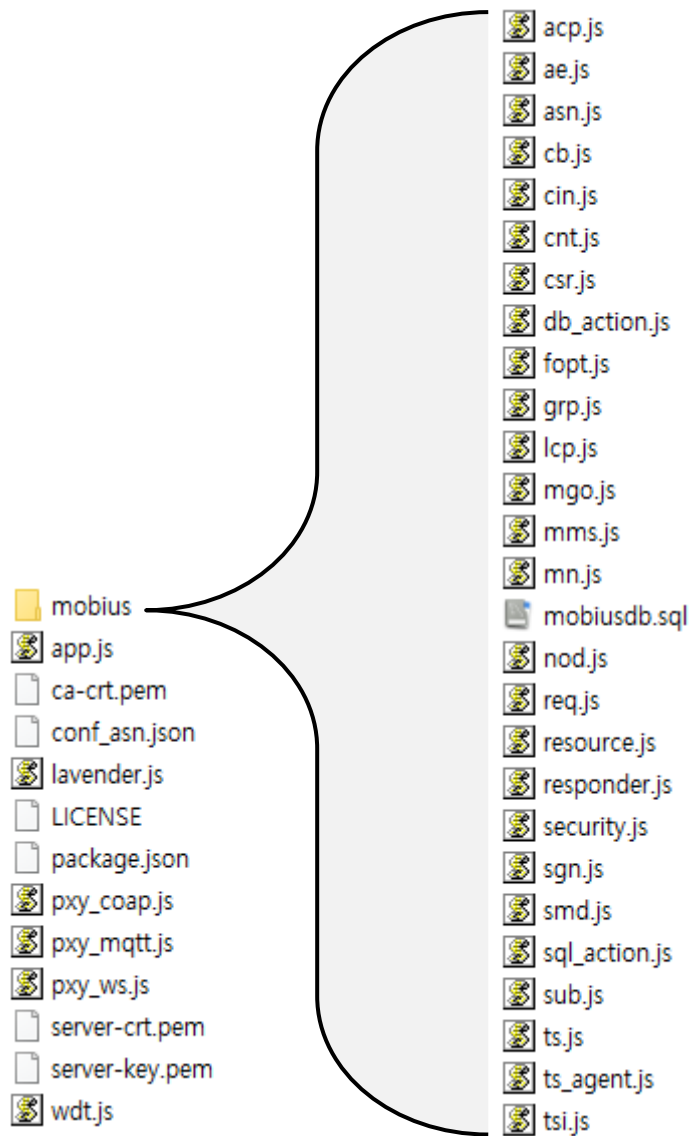


그림 1 nCube:Lavender를 사용한 연동 구조

nCube:Lavender와 Mobius의 연동은 nCube:Rosemary의 연동 구조와 거의 동일하다. 다른 점은 nCube:Lavender가 하부에 CSE를 가질 수 없는 것이 다르다. 위의 예제에서는 remoteCSE를 사용하지 않으므로 그림이 동일하다. 그림 1의 경우에는 Lavender 내에 리소스 트리를 가지고 있으며 여기에 nCube:Thyme인 AE가 등록이 된다. IN-CSE인 Mobius와의 연동은 또 다른 nCube:Thyme이 IN-CSE에 해당 리소스를 등록하고 데이터를 업로드 하는 것을 볼 수 있다. 여기에서 Lavender는 단말 플랫폼의 역할을 수행하고 nCube:Thyme인 AE에 의해 필요한 정보 및 가공을 통해 데이터를 IN-CSE인 Mobius로 전달하고 있다.

1.2. nCube:Lavender 디렉토리 구조

다음 그림은 nCube:Lavender 플랫폼의 디렉토리 구조를 나타낸다.



Lavender 폴더 구조는 크게 mobius 폴더와 lavender.js, app.js, pxy_mqtt.js, pxy_coap, pxy_ws, wdt.js, package.json으로 구성된다. mobius 폴더는 다시 여러 개의 소스 파일로 구성되어 있고 특정 기능을 수행하는 db_action.js, fopt.js, resource.js, responder.js, security.js, sgn.js, sql_action.js, ts_agent.js 소스 파일과 이를 제외한 나머지 파일인 리소스를 나타내는 소스파일로 구성되어 있다. 각 파일에 대한 구체적 기능 및 역할은 아래 표와 같다.

소스 파일	기능 및 역할
Lavender.js	Lavender를 실행하는 파일로써 Lavender 구동을 위한 설정 값들을 수정할 수 있다. 이 파일을 통해 Lavender의 메인 코드를 로드한다.
app.js	Lavender의 main 역할을 수행하는 파일로 Lavender로 들어오는 모든 패킷은 이 파일에서 제일 처음 처리한다. 이 파일에서 http 서버를 실행하고 Listen 하고 대기 한다. 이후 패킷이 들어오면 패킷의 타겟 url을 파싱하고 패킷의 body를 분석하여 정상적인 패킷인지 아닌지를 판단하고, 정상적인 패킷으로 판단되면 해당 동작을 수행하는 resource.js로 전달한다. 오류 패킷일 경우 responder.js를 통해 오류를 리턴한다. 이 파일에서 http 서버는 서버가 실행될 때 서버 성능을 개선하기 위해 클러스터링을 통해 서버를 실행하도록 구현한 것이다.
pxy_mqtt.js	이 파일은 mqtt to http binding 하는 파일로써 mqtt 관련 데이터를 처리하는 파일이다. 시작될 때 Lavender 정보를 가져와서 oneM2M mqtt topic을 만들고 subscribe한다. 이후 mqtt로 데이터가 수신되면 이를 http 패킷으로 만들어서 Lavender로 전달하고 response를 받으면 이를 다시 mqtt 패킷으로 만들어 mqtt response 한다.
pxy_coap.js	이 파일은 coap to http binding 하는 파일로써 coap 데이터를 처리하는 파일이다. 시작될 때 coap 서버를 생성하고 이후 coap 데이터가 수신되면 이를 http 패킷으로 만들어서 Lavender로 전달하고 response를 받으면 이를 다시 coap 패킷으로 만들어 coap 으로 response 한다.
pxy_ws.js	이 파일은 websocket to http binding 하는 파일로써 websocket 데이터를 처리하는 파일이다. 시작될 때 websocket 서버를 생성하고 이후 websocket 데이터가 수신되면 이를 http 패킷으로 변환해서 Lavender로 전달하고 Lavender로부터 response를 받으면 이를 다시 websocket 패킷으로 만들어 websocket으로 response 한다.
wdt.js	Lavender 자체 watchdogtimer로 구축한 것이고, 모비우스 자체 타이머로 사용된다.

mobius/resource.js	이 파일은 Lavender가 지원하는 모든 리소스에 대해 create, retrieve, update, delete, notify, discovery를 수행하는 코어 파일이다. app.js에서 검증된 url, body에 대해 해당되는 method에 따라 내부적으로 처리하는 포맷으로 변환하고 실제 데이터베이스 연동작업을 수행한다. 데이터 베이스 동작은 db_action.js와 sql_action.js를 통해 수행된다.
mobius/responder.js	이 파일은 요청받은 request 패킷에 대해 app.js, resource.js에 의해 처리된 결과를 response 하는 역할을 수행한다. response 할 때 응답할 데이터를 json, xml 포맷에 맞게 변형한 후 response 한다.
mobius/db_action.js	실제 데이터베이스와 연결하고 데이터베이스에 접근하여 결과를 리턴하는 함수를 포함하는 파일이다.
mobius/sql_action.js	이 파일은 데이터베이스를 Lavender 동작에 필요한 데이터를 얻기 위한 sql문을 만들고 이를 db_action.js의 함수를 통해 데이터 베데이터부터 데이터를 리턴받아 전달하는 역할을 수행한다.
mobius/sgn.js	이 파일은 타겟 리소스에 subscription이 차일드로 존재하는지 체크하고 subscription이 존재하면 해당 이벤트를 체크하고 subscription의 notificationUri attribute 정보를 바탕으로 request 메시지를 만들어 전송하는 (notification 전송) 역할을 수행한다.
mobius/security.js	이 파일은 타겟 리소스에 대한 accessControlPolicy를 체크하여 request 요청에 대해 reject를 할 것인지 허용할 것인지를 판단한다. request 패킷의 X-M2M-Origin 헤더에 포함되어 있는 id 값을 가지고 타겟 리소스에 연결되어 있는 accessControlPolicy 리소스를 확인하여 이 id에 접근 권한이 있는지 확인한다.
mobius/fopt.js	group 리소스에 대한 fanOutPoint 리소스에 대한 처리를 담당한다. fanOutPoint 리소스는 virtual 리소스으로써 group 리소스에 포함된 멤버들 모두에게 요청받은 패킷을 전달하고 그 결과를 모아서 요청자에게 response 한다.
mobius/ts_agent.js	이 파일은 timeSeriesInstance 리소스를 관리하는 역할로써 timeSeriesInstance 중에서 missing 되는 데이터들을 주기적으로 모니터링하여 missing 데이터들을 저장한다.
pxymqtt.js	이 파일은 mqtt to http binding 하는 파일로써 mqtt 관련 데이터를 처리하는 파일이다. 시작될 때 Lavender 정보를 가져와서 oneM2M mqtt topic을 만들고 subscribe한다. 이후 mqtt로 데이터가 수신되면 이를 http 패킷으로 만들어서 Lavender로 전달하

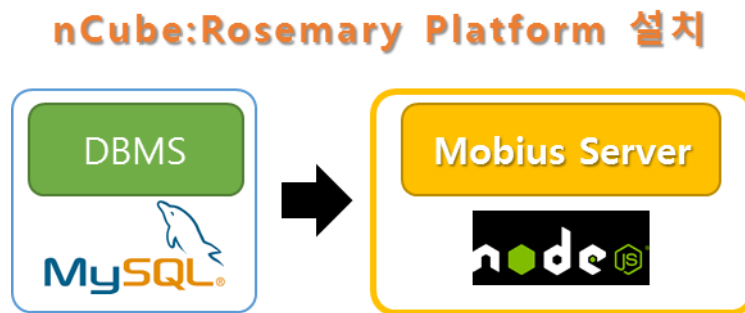
	고 response을 받으면 이를 다시 mqtt 패킷으로 만들어 mqtt response 한다.
mobius/acp.js	accessControlPolicy 리소스 파일 – 접근제어 관련 리소스
mobius/ae.js	AE 리소스 파일
mobius/cb.js	CSEBase 리소스 파일 - Lavender 자신의 정보를 가진다.
mobius/cin.js	contentInstance 리소스 파일 – 실제 데이터가 여기에 저장된다.
mobius/cnt.js	container 리소스 파일
mobius/csr.js	remoteCSE 리소스 파일
mobius/grp.js	group 리소스 파일
mobius/lcp.js	locationPolicy 리소스 파일
mobius/mms.js	multimediaSession 리소스 파일
mobius/mn.js	MN-CSE 리소스 파일 – IN-CSE에 remoteCSE 등록하는 기능을 포함하고 있다. nCube:Lavender 구동시 사용된다.
mobius/smd.js	semanticDescriptor 리소스 파일
mobius/sub.js	subscription 리소스 파일
mobius/ts.js	timeSeries 리소스 파일
mobius/tsi.js	timeSeriesInstance 리소스 파일
mobius/req.js	request 리소스 파일 – nonblocking 기능 지원
mobius/nod.js	node 리소스 파일
mobius/mgo.js	mgmtObj 리소스 파일

2. nCube:Lavender 설치

nCube:Lavender 설치 과정은 Mobius 설치 과정과 동일하게 진행된다. 물리적 디바이스의 차이가 있을 수 있지만 본 문서는 Windows OS를 기준으로 작성된다.

2.1. 설치 개요

nCube:Lavender는 MySQL DBMS를 사용한다. 먼저 MySQL을 설치하고 Mobius 서버 순으로 설치를 하면 nCube:Lavender 서버 플랫폼 구축이 완성된다. Mobius와는 다르게 MQTT 서버의 경우 Mobius의 MQTT 서버를 사용하므로 MQTT 서버는 설치하지 않는다.



[Lavender 플랫폼 설치 개요]

2.2. nCube:Lavender 구동환경 구축 (Windows)

2.2.1. MySQL 설치

nCube:Lavender은 데이터 저장 및 관리를 위해 MySQL DB를 기반으로 동작한다. 여기서는 오픈소스 RDBMS인 MySQL을 다운로드 받고 설치하는 방법에 대해 설명한다.

<http://www.mysql.com>



Generally Available (GA) Releases | Development Releases

MySQL Community Server 5.6.22

Select Platform:

Looking for previous GA versions?

Recommended Download:

MySQL Installer 5.6 for Windows

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the server-only MSI packages.

Windows (x86, 64-bit), MySQL Installer MSI

[Download](#)

Other Downloads:

Windows (x86, 32-bit), ZIP Archive (mysql-5.6.22-win32.zip)	5.6.22	342.0M	Download
MD5: 00abcb99a71708d372ff073f870deabd Signature			
Windows (x86, 64-bit), ZIP Archive (mysql-5.6.22-winx64.zip)	5.6.22	347.5M	Download
MD5: 8810b875ff1651e3c91473faa7ed6509 Signature			

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

<http://dev.mysql.com/downloads/mysql>

MySQL에 들어가 보면 MySQL의 다양한 버전이 있는데 위 링크를 클릭해서 들어가면 무료로 사용할 수 있는 GPL버전인 MySQL Community Server를 다운받을 수 있다. 여기서는 윈도우에 MySQL을 설치하기 위해 Windows (x86, 64-bit), MySQL Installer MSI라고 적힌 Download 버튼을 클릭한다. (MySQL에는 MySQL Enterprise Edition이나 MySQL Cluster CGE처럼 유료로 사용해야 하는(commercial) 것도 있으니 참고하길 바란다.)

웹 설치가 아닌, 설치파일 전부를 다운로드 받아 설치하기 위해서는 Zip Archive 다운로드 버튼을 클릭한다. 이 버전은 웹 설치가 어려운 곳에서 MySQL을 설치할 때 이 버전을 활용할 수 있다.

Begin Your Download - mysql-installer-community-5.6.21

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system
- Comment in the MySQL Documentation

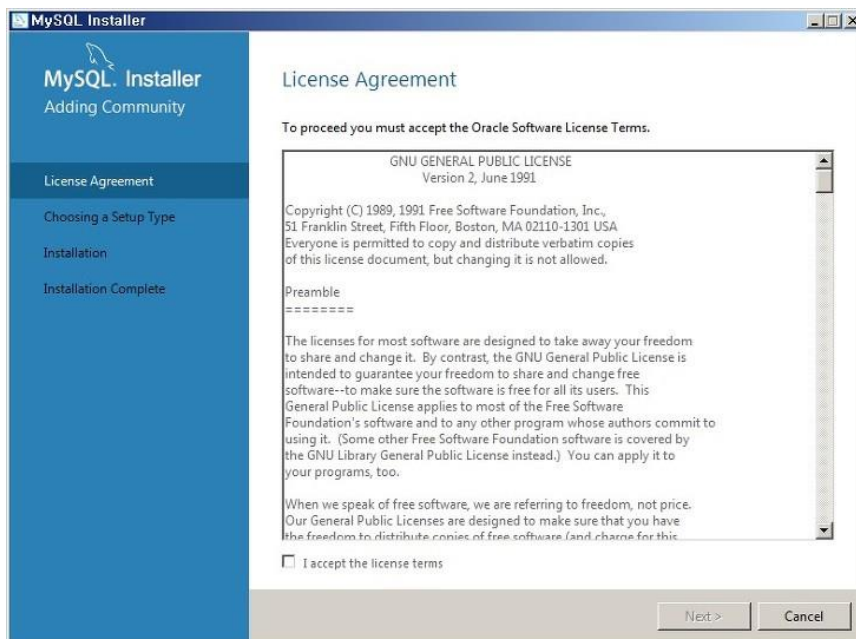
Login »
using my Oracle Web account

Sign Up »
for an Oracle Web account

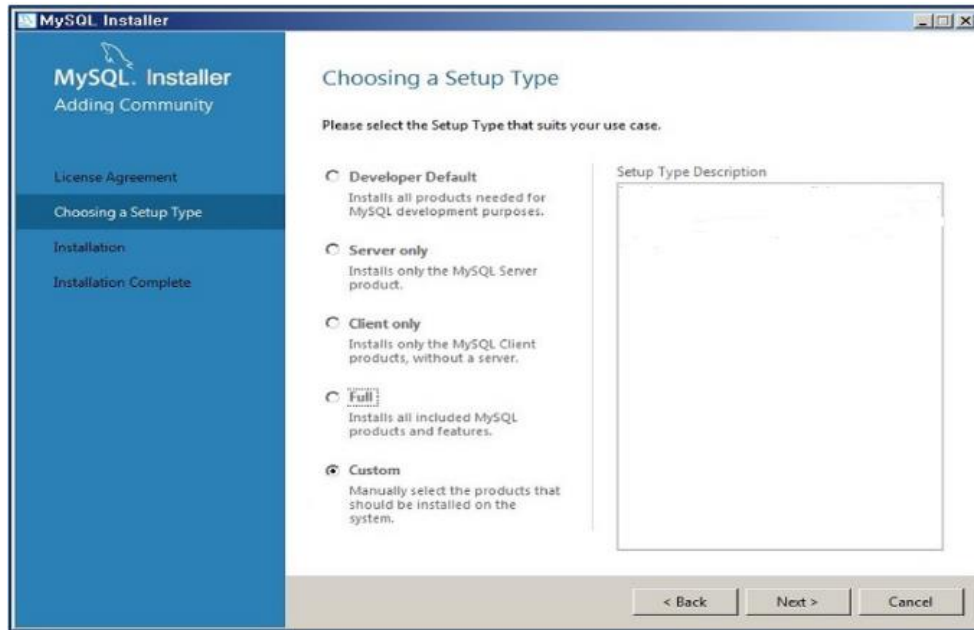
MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account click the Login link. Otherwise, you can sign up for a free account by clicking the Sign Up link and following the instructions.

[No thanks, just start my download.](#)

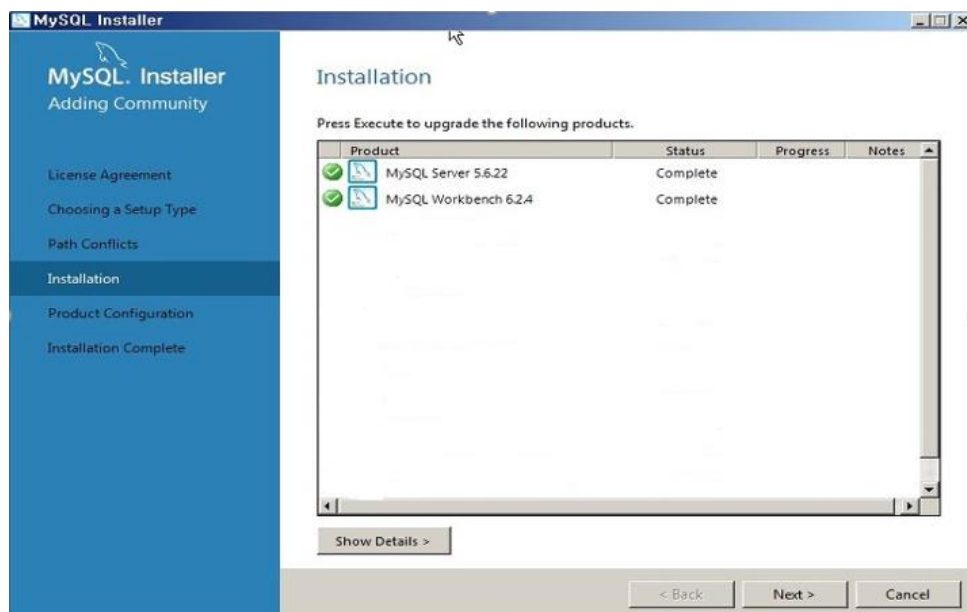
로그인 화면이 나오면 회원가입을 하거나 아래에 있는 No thanks, just start my download 링크를 클릭해서 다운로드 받는다.



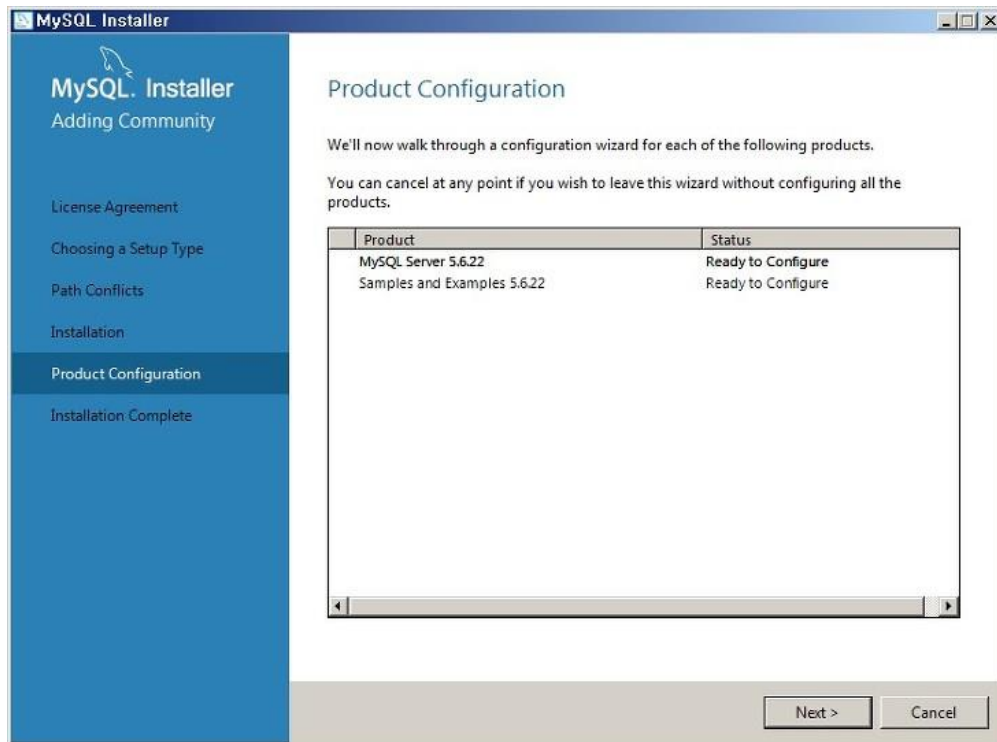
MySQL 설치파일을 다운받아서 실행시켜보면 위와 같은 화면이 나오면 라이선스에 동의하고 Next 버튼을 누른다.



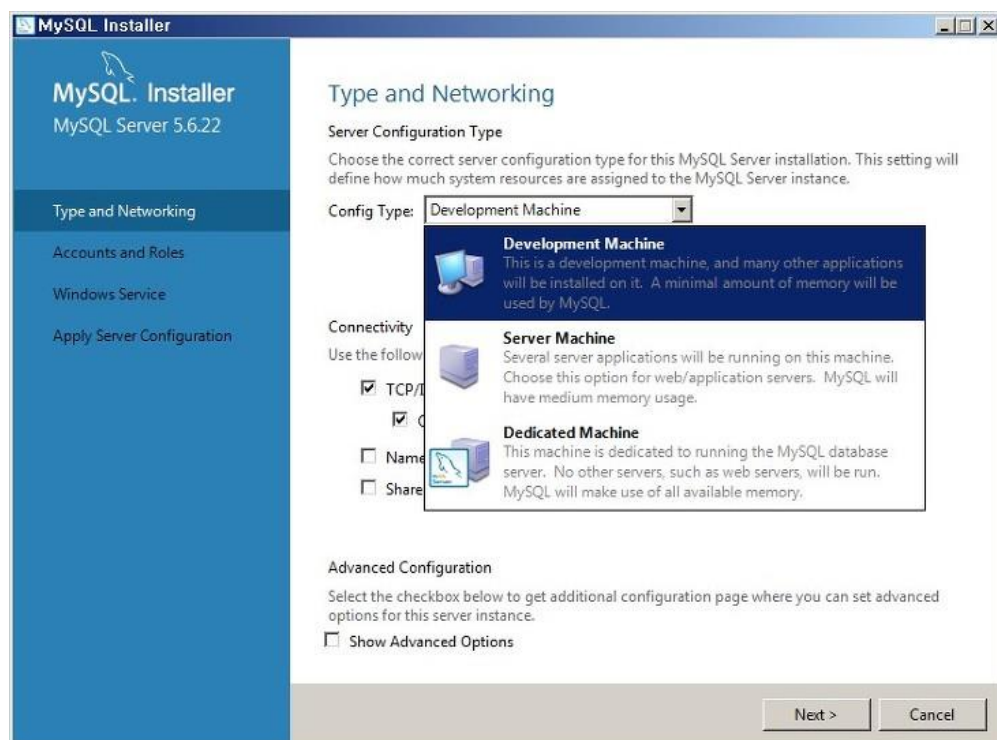
설치 스타일에는 Developer Default, Server only, Client only, Full, Custom 이렇게 5가지가 있는데, Lavender를 위해서는 MySQL Server와 MySQL Workbench가 필요하다. 여기서 Custom을 선택하고 Next 버튼을 누르면서 설치를 진행한다.



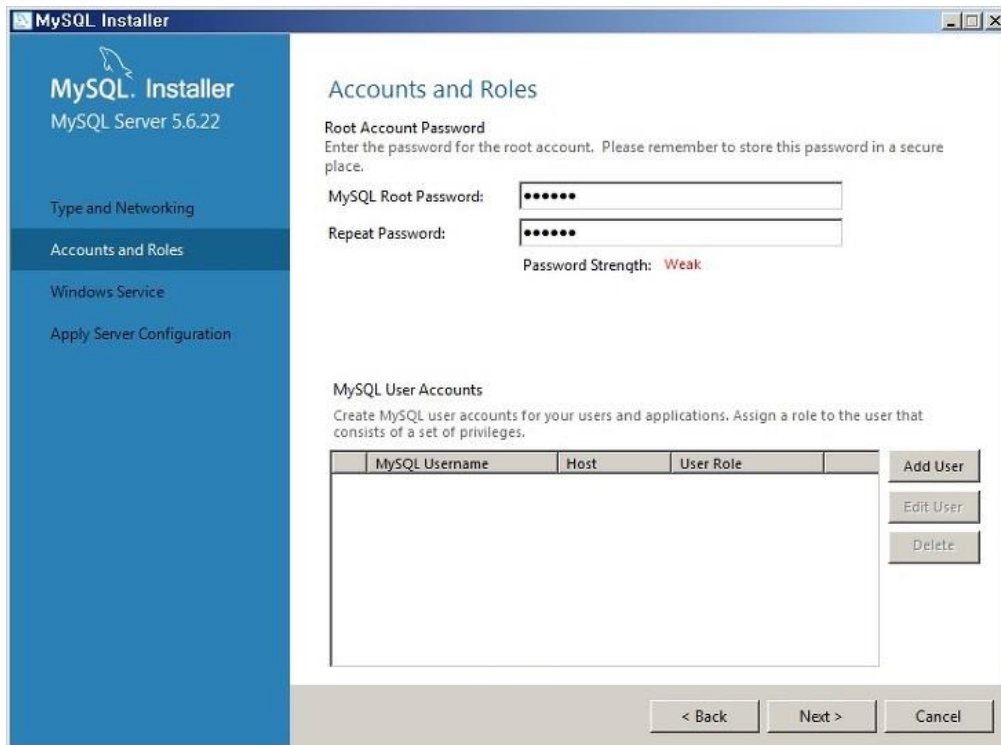
설치가 완료되었으면 Next



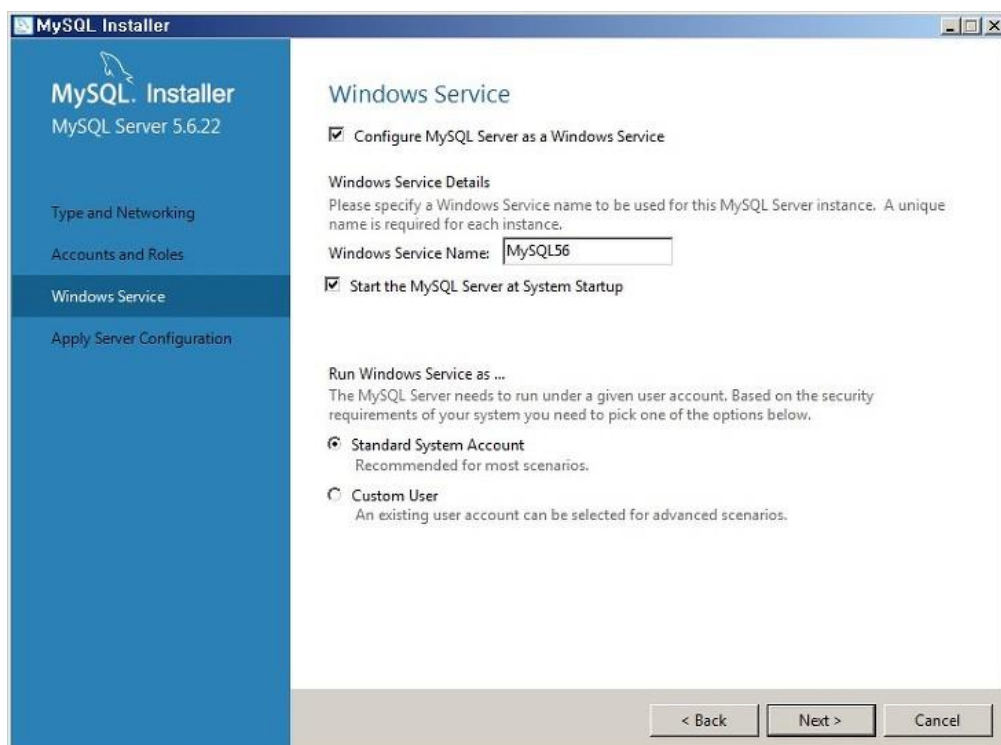
MySQL 환경설정 작업을 위해 Next



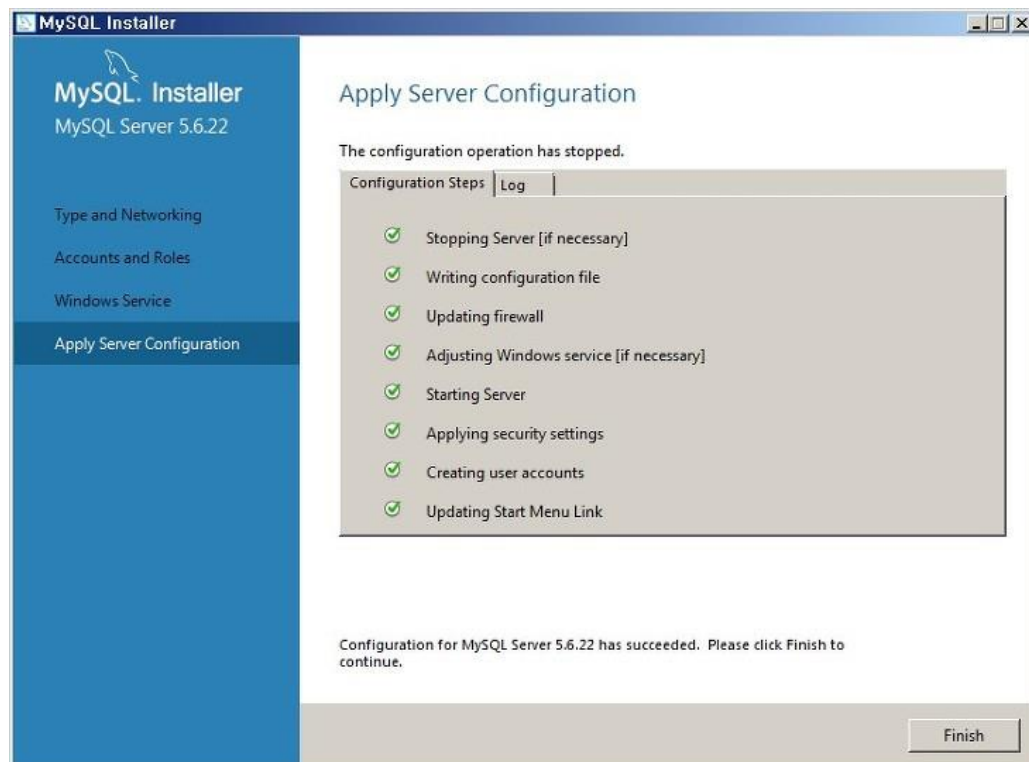
환경설정 타입에는 Development Machine, Server Machine, Dedicated Machine이 있는데 필요한 것을 선택하면 되고 여기서는 개발자라고 가정하고 Development Machine을 선택한다. Next



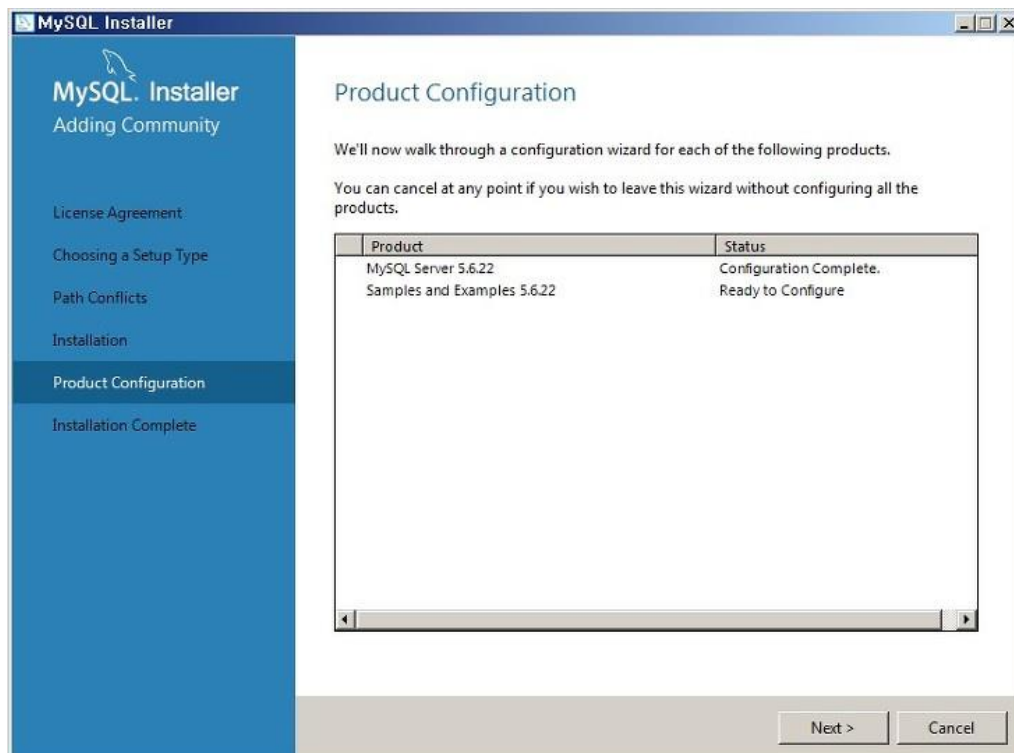
루트암호(Root Password)를 지정한다.



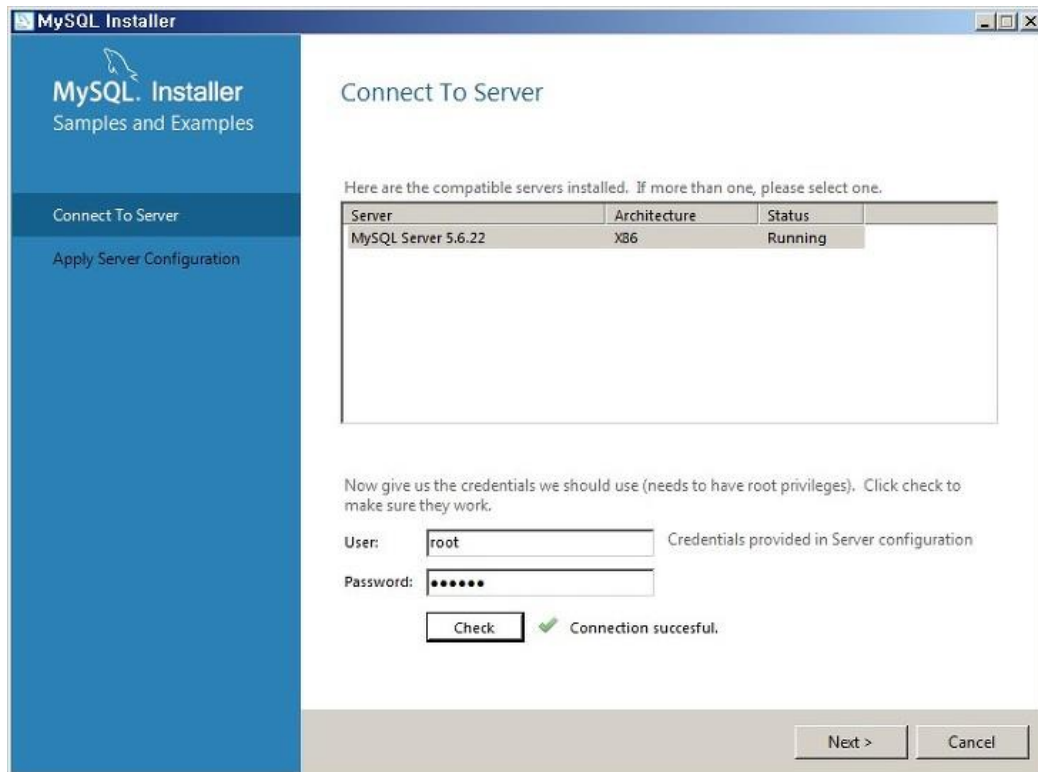
윈도우 부팅시 MySQL 자동 실행 옵션을 설정 할 수 있는 화면으로 윈도우 서비스 이름 지정한다. Next



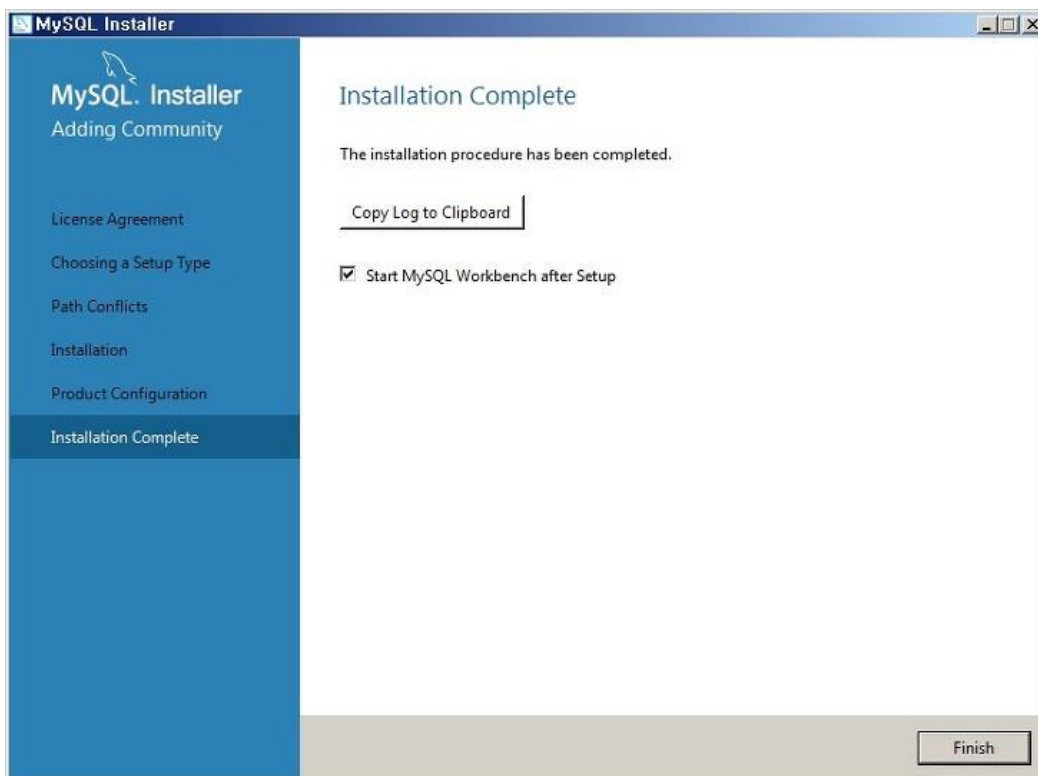
Execute를 누르고 MySQL Server 환경설정이 제대로 설정된 것을 확인하고 Finish를 누른다.



기타 환경설정을 위해 Next.



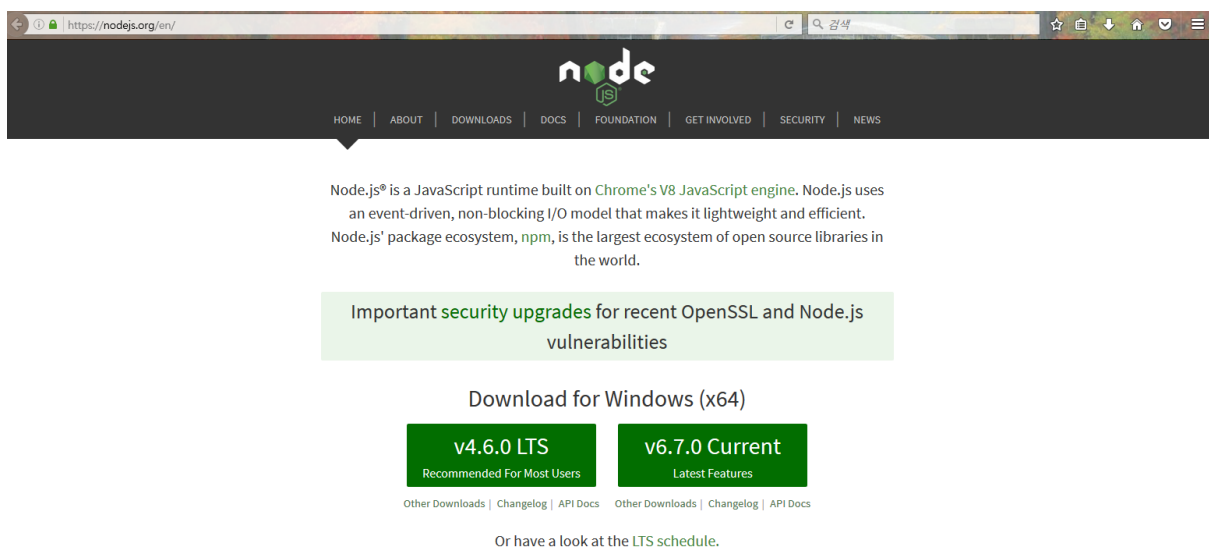
루트 ID와 암호를 입력하고 체크 후 계속 Next, Execute, Finish 버튼을 누른다.



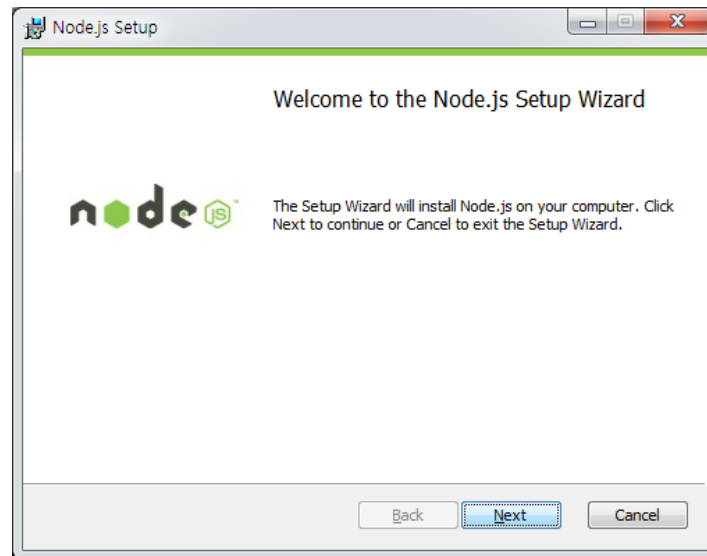
MySQL 설치가 완료되었다. 마지막 Finish 버튼을 누른다.

2.2.2. Node JS 설치

node.js는 고성능의 비동기 IO (Async/Non-blocking IO)를 지원하는 single thread 기반 네트워크 서버이다. 2009년 Ryan Dahl에 의해 개발이 시작되었고 현재 수많은 지원 모듈을 가지고 있는 오픈 소스 프로젝트 중 하나이다. Node.js는 Google Chrome V8 엔진으로 개발되어 있고 Event 기반의 프로그래밍 모델로써 프로그래밍 언어로는 Java script를 사용한다. 근래에 들어서 많은 인터넷 기업들이 node.js를 도입하고 있다.

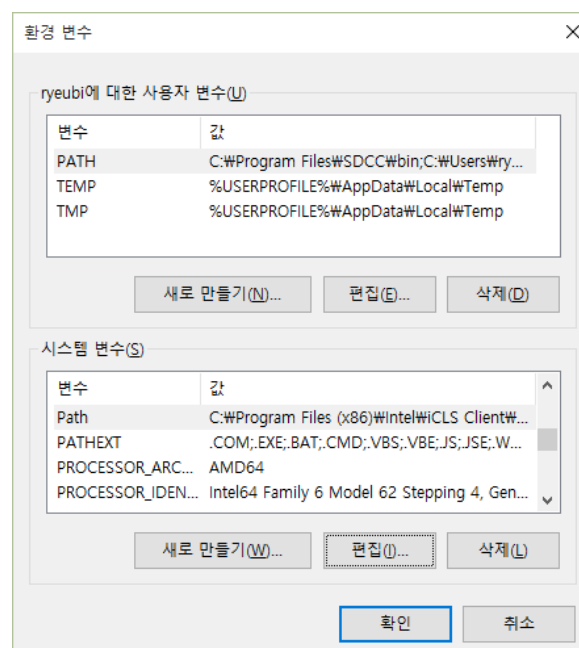


<http://www.nodejs.org> 페이지에서 Install 버튼을 누르면 OS에 맞는 인스톨러를 다운로드해 준다.

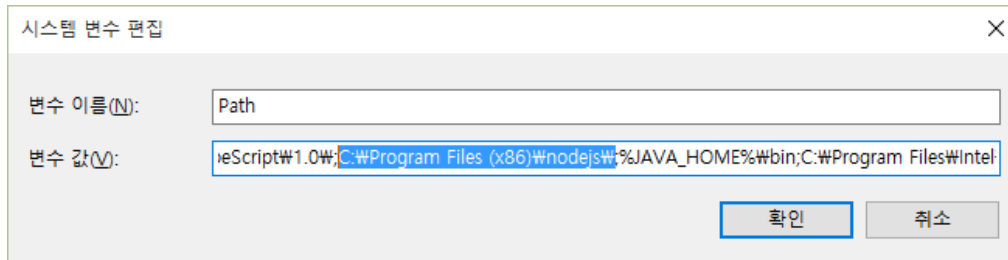


다음으로 위와 같이 다운로드한 인스톨러를 실행한다.

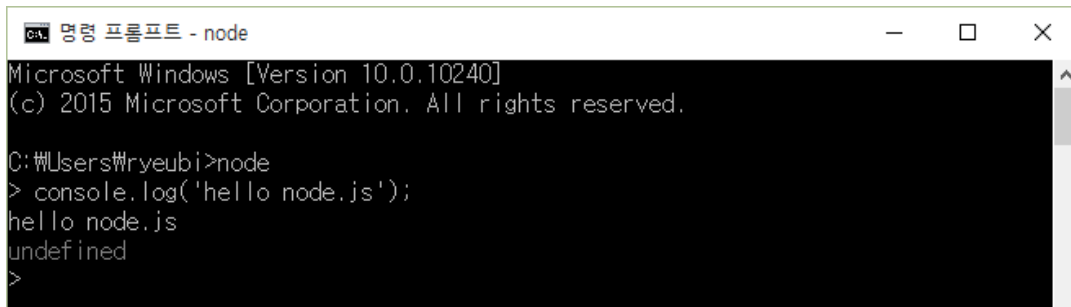
디폴트로 Next 버튼을 누르면 설치가 완료되고, 설치가 되었으면 설치된 디렉토리를 PATH에 추가한다.



Path를 선택하고 아래와 같이 C:\Program Files (x86)\nodejs\ 를 추가하고 확인 버튼을 누른다.



노드가 제대로 설치가 되었는지 확인한다. Node.js는 대화형 cli를 제공한다.



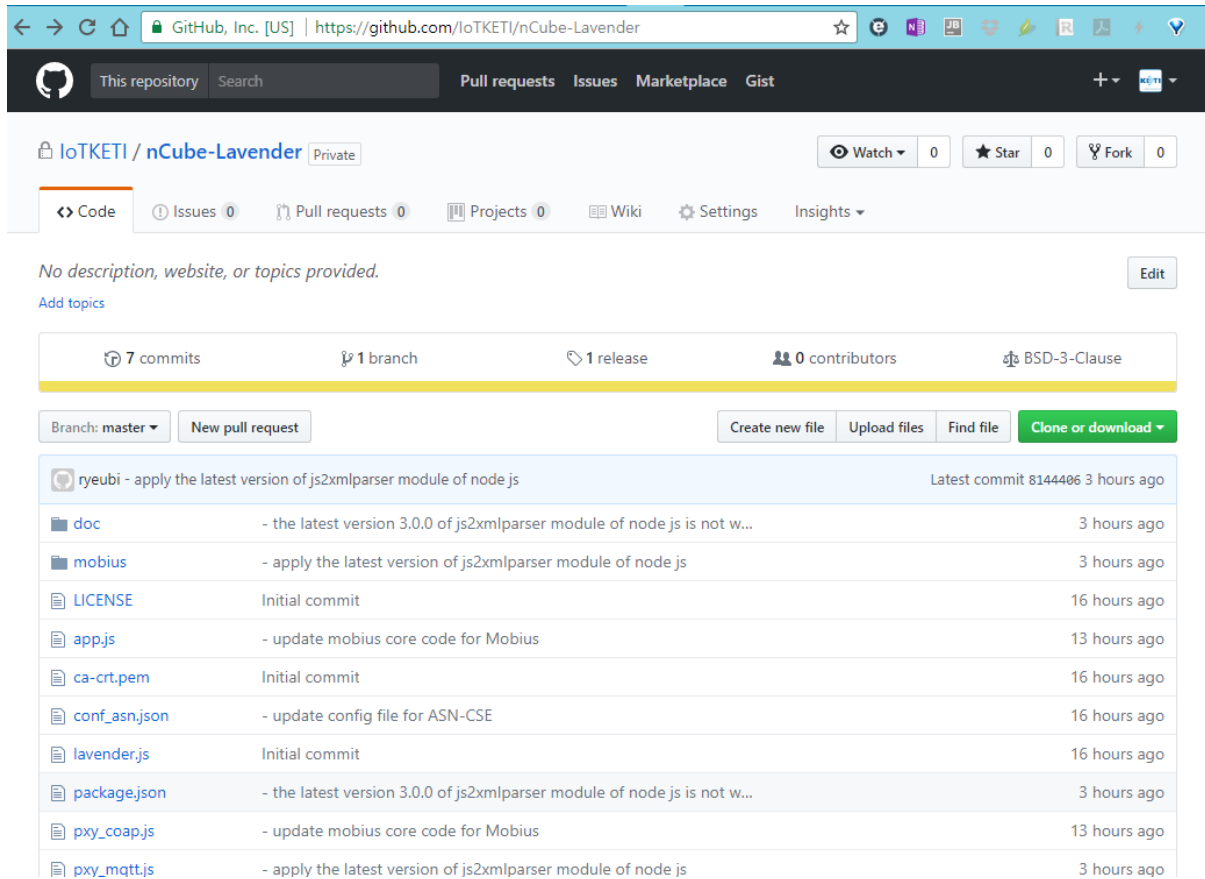
위와 같이 실행이 되면 node.js가 제대로 설치가 된 것이다.

2.3. nCube:Lavender 설치

nCube:Lavender는 GitHub를 통해 Lavender 서버 플랫폼 소스를 공개하고 있다. nCube:Lavender 서버 플랫폼 압축 파일을 다운로드 받은 후, 제공된 매뉴얼에 따라 Port 정보 등의 Configuration 정보를 설정한다.

2.3.1. 다운로드 및 모듈 설치

nCube:Lavender 설치 파일은 OCEAN Alliance에서 제공하는 GitHub (<https://github.com/loTKETI/nCube-Lavender>) 접속하여 아래와 같이 nCube-Lavender 서버 파일을 다운로드한다.



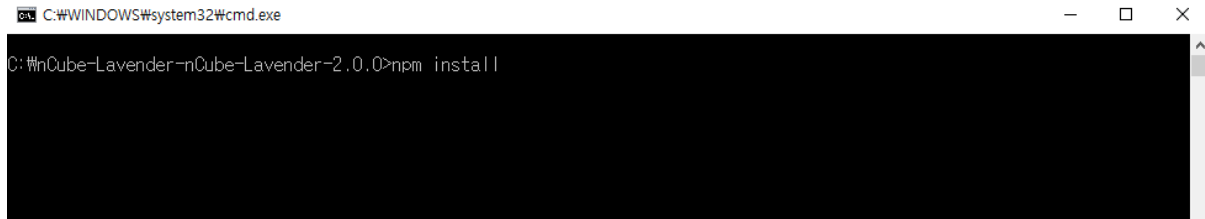
다운로드한 파일을 적당한 곳에서 압축을 풀면 다음과 같이 node.js 버전의 Lavender 파일들을 볼 수 있다. Node.js는 별도의 컴파일 과정이 필요하지 않으므로 바로 압축을 풀 후 실행할 수 있다. 이전의 MySQL과 Node.js가 제대로 설치가 되어 있다면 nCube:Lavender 에 사용된 node.js 모듈을 설치하면 별도의 작업없이 실행이 가능하다.

- mobius
- app.js
- ca-crt.pem
- conf_asn.json
- lavender.js
- LICENSE
- package.json
- pxy_coap.js
- pxy_mqtt.js
- pxy_ws.js
- server-crt.pem
- server-key.pem
- wdt.js

Lavender.js가 Lavender의 메인 코드를 실행하는 파일이다. app.js는 대문 역할과 라우팅 역할을 수행하는 메인 코드이다. pxy_mqtt는 MQTT proxy를 구현한 코드이고, pxy_coap은 CoAP, pxy_ws는 Websocket을 구현한 코드이다. mobius 폴더에 있는 파일들이 실제 패킷을 처리하고 DB 액세스와 응답을 수행하는 코드들이다.

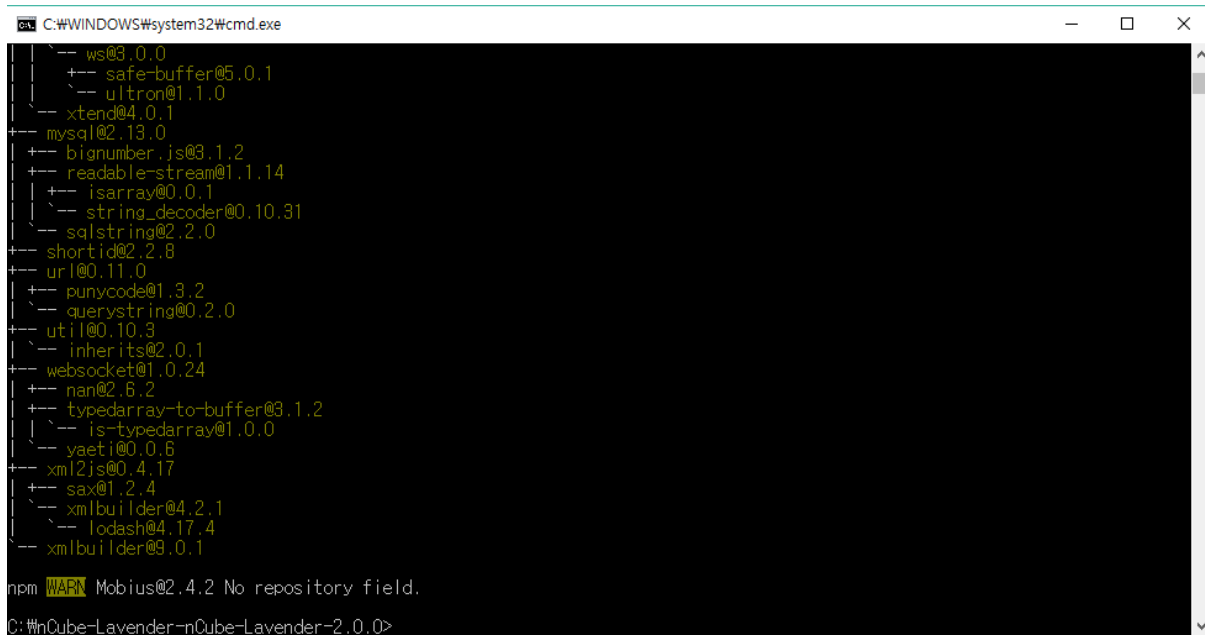
현재는 추가적인 node.js 모듈들이 설치가 되어있지 않아 에러가 날 것이다. 다음과 같이 명령 프롬프트를 실행시키고 npm 툴을 이용해 필요한 모듈

을 설치해야 한다. package.json 파일에 필요한 모듈 리스트를 가지고 있다.



```
C:\WINDOWS\system32\cmd.exe
C:\nCube-Lavender-nCube-Lavender-2.0.0>npm install
```

위와 같이 npm install 을 입력하고 엔터를 치면 아래와 같이 필요한 모듈들이 설치가 된다.



```
C:\WINDOWS\system32\cmd.exe
C:\nCube-Lavender-nCube-Lavender-2.0.0>npm install
-- ws@3.0.0
-- safe-buffer@5.0.1
-- ultron@1.1.0
-- xtend@4.0.1
-- mysql@2.13.0
-- bignumber.js@3.1.2
-- readable-stream@1.1.14
-- isarray@0.0.1
-- string_decoder@0.10.31
-- sqlstring@2.2.0
-- shortid@2.2.8
-- url@0.11.0
-- punycode@1.3.2
-- querystring@0.2.0
-- util@0.10.3
-- inherits@2.0.1
-- websocket@1.0.24
-- nan@2.6.2
-- typedarray-to-buffer@3.1.2
-- is-typedarray@1.0.0
-- yaeti@0.0.6
-- xmldom@0.4.17
-- sax@1.2.4
-- xmlbuilder@4.2.1
-- lodash@4.17.4
-- xmlbuilder@3.0.1
npm WARN Mobius@2.4.2 No repository field.
C:\nCube-Lavender-nCube-Lavender-2.0.0>
```

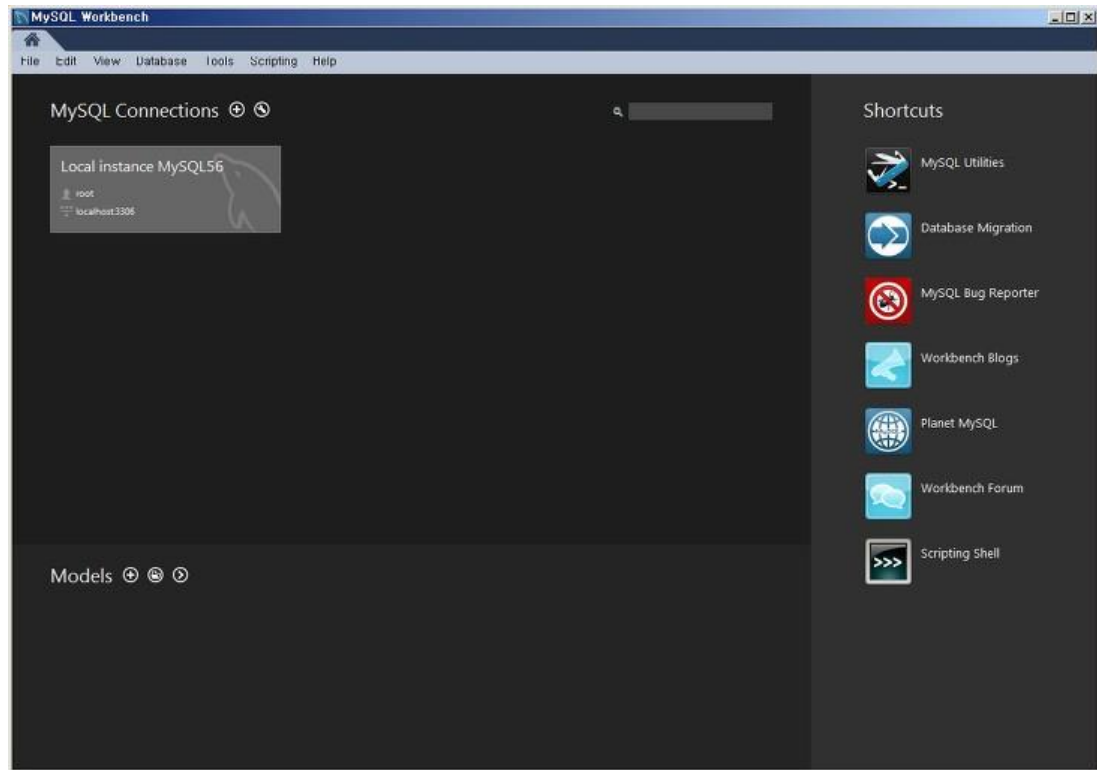
- mobius
- node_modules
- app.js
- ca-crt.pem
- conf_asn.json
- lavender.js
- LICENSE
- package.json
- pxy_coap.js
- pxy_mqtt.js
- pxy_ws.js
- server-crt.pem
- server-key.pem
- wdt.js

이렇게 설치가 되면 다음 그림과 같이 node_modules 폴더가 생긴 것을 볼 수 있다.

이제 모듈 설치가 끝났으니 다음과 같이 node Lavender.js를 실행하면 nCube:Lavender 서버 플랫폼 설치가 완료된다.

2.3.2. mobiusdb 데이터베이스 생성

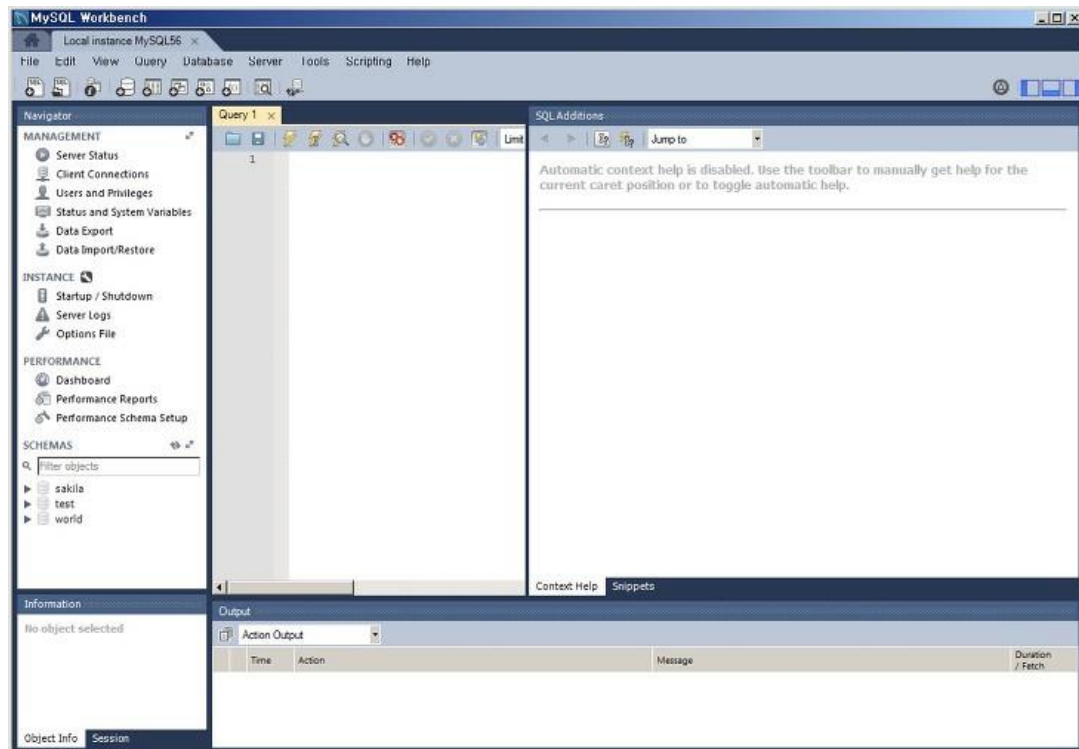
위의 Lavender 구동 개발환경에서 MySQL을 설치했으므로 MySQL Workbench가 설치되어 있을 것이다.



MySQL Workbench를 실행하고 Connections를 선택한다.

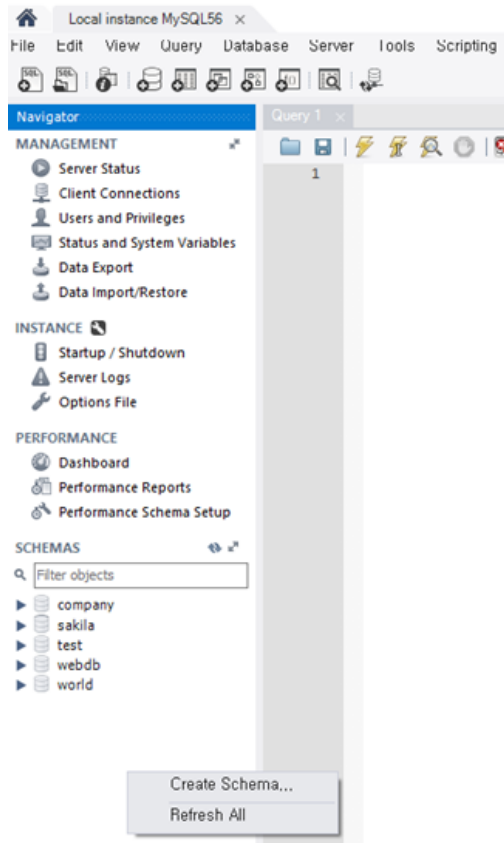


MySQL을 설치할 때 설정한 암호를 입력하면 사용할 수 있다.

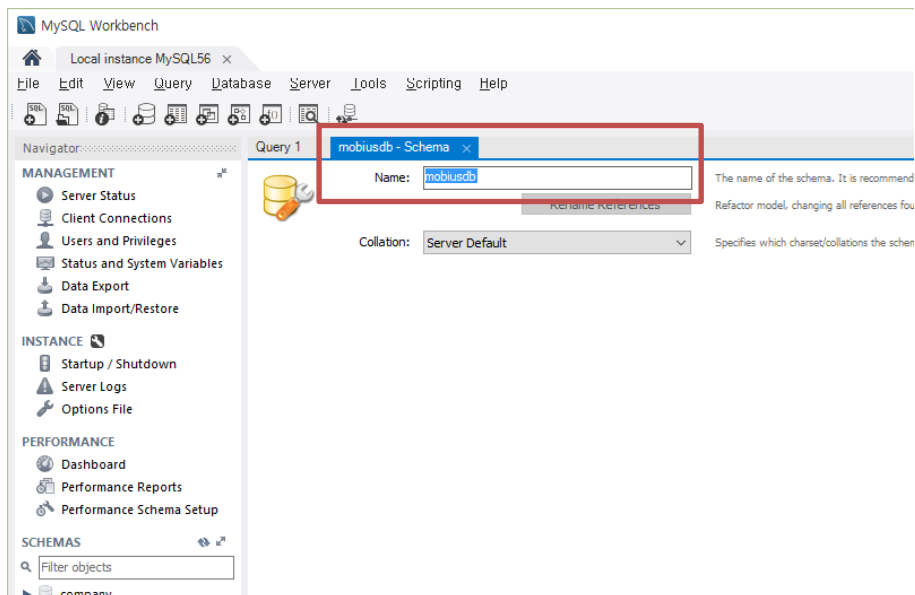


Workbench는 GUI환경에서 MySQL을 다룰 수 있는 프로그램으로 SQL 쿼리를 날릴 수 있는 것은 물론 데이터베이스, 테이블 생성, Import, Export 등 다양한 MySQL 관리 작업을 할 수 있다.

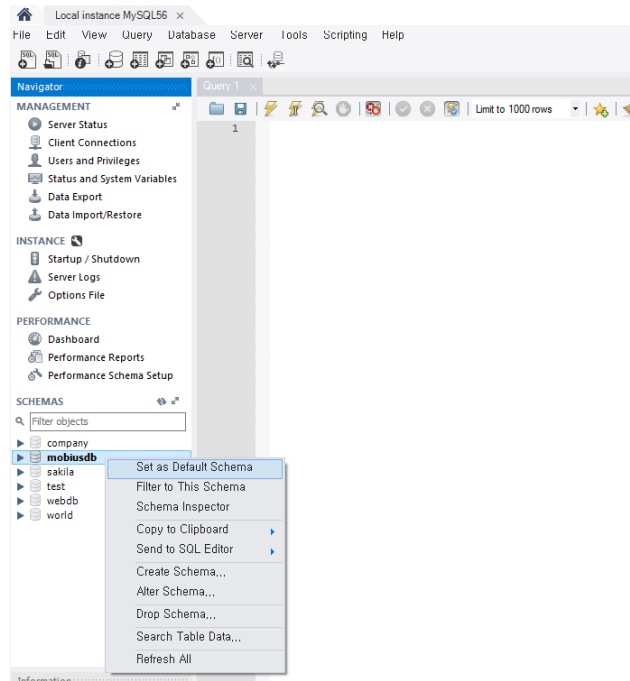
- 데이터베이스 생성



위의 그림처럼 SCHEMAS 빈 곳에서 마우스 오른쪽 버튼을 클릭하면 뜨는 메뉴에서 “Create Schema...” 선택한다.



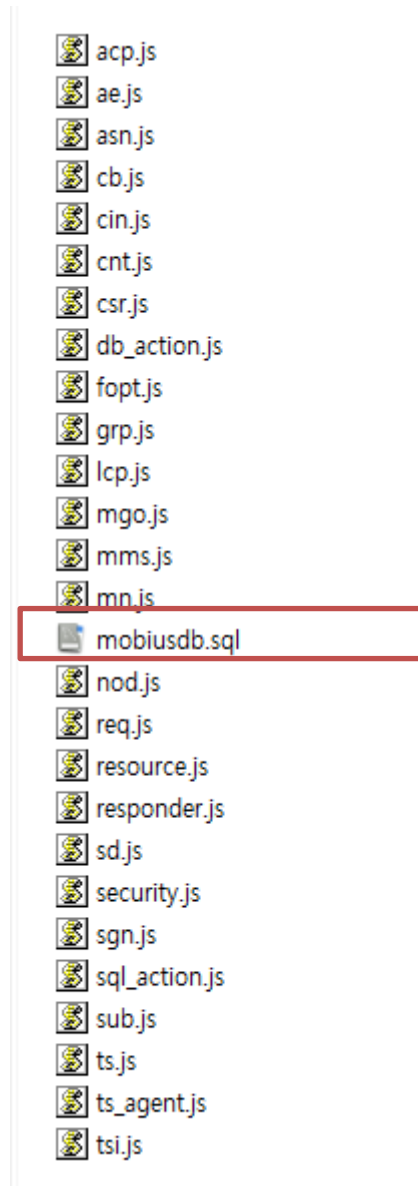
스키마 생성 창이 뜨면 데이터베이스 이름을 “mobiusdb” 입력하고 Apply 버튼을 누르면 데이터베이스가 생성된다.



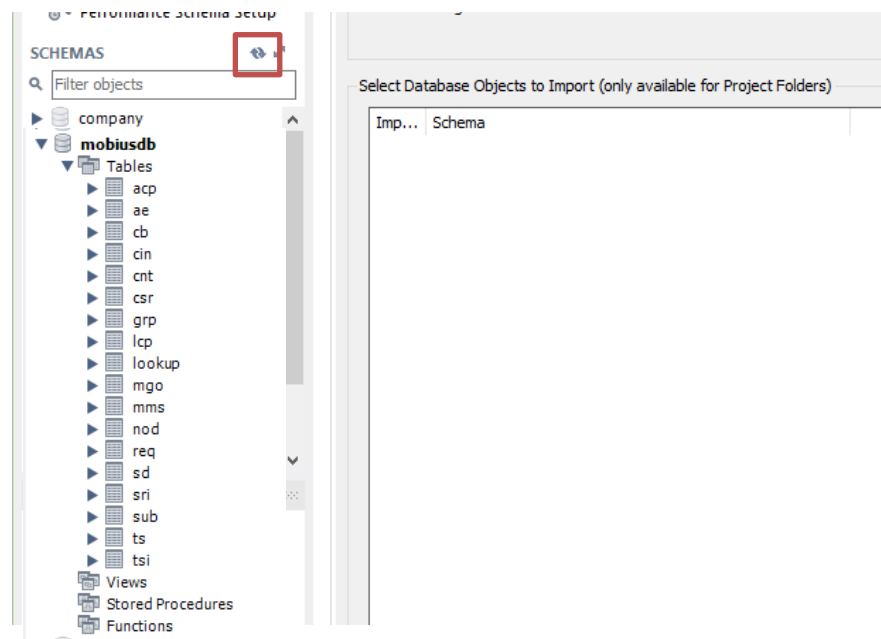
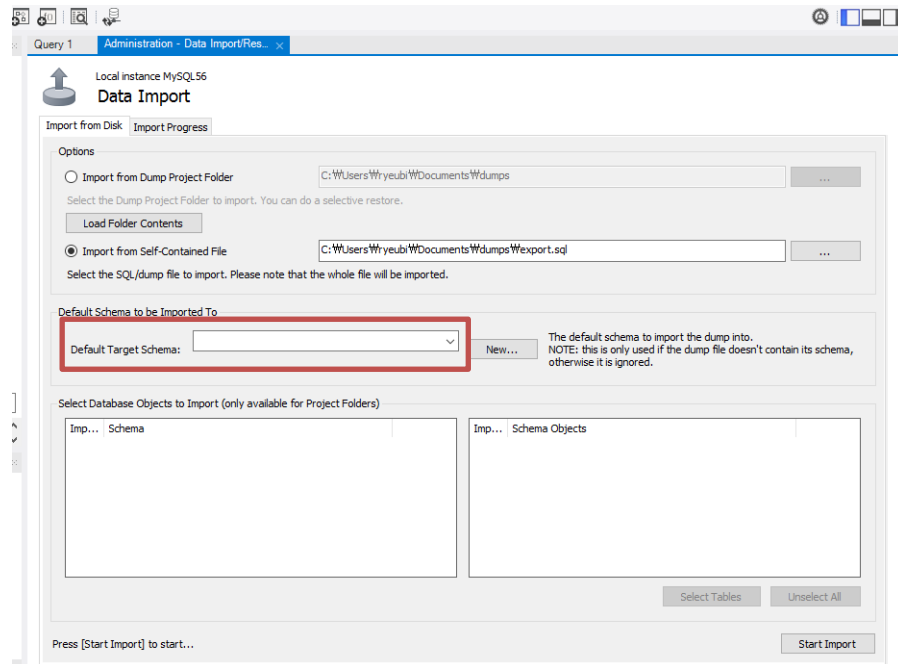
"mobiusdb" 데이터가 생성이 되면 위의 그림과 같이 왼쪽 창에 데이터베이스가 생성됨을 확인할 수 있고 마우스 오른쪽 버튼을 눌러 Set as Default Schema로 설정을 해 준다.

- mobiusdb로 테이블 가져오기

Import할 Lavender DB 테이블에 대한 sql 파일은 앞서 다운로드한 Lavender가 설치된 폴더내의 mobius폴더에 위치해 있다.



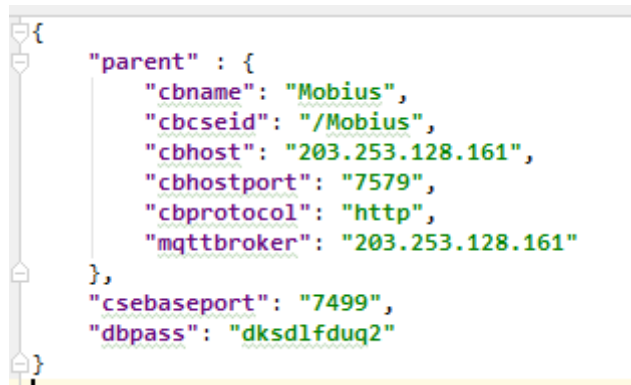
mobiusdb로 테이블을 가져오기 위해 Workbench의 Server 메뉴에서 Data Import 를 선택한다. Data Import 창에서 Import from Self-Contained File을 선택하고 위의 그림에 있는 mobiusdb.sql 파일을 선택한다. 이후 Default Target Schema 로 앞서 만든 mobiusdb 를 선택하고 Start Import를 누르면 다음 그림과 같이 Lavender에서 사용되는 테이블이 생성된다.



Data Import를 수행한 후에는 테이블이 보이지 않으면 위에 있는 새로고침 아이콘을 클릭해야 Import 된 테이블들을 볼 수 있다.

3. nCube:Lavender 구동

nCube:Lavender 설치까지 정상적으로 설치된 후 Lavender 를 구동하기 전에 환경 설정을 해야 정상적으로 구동할 수 있다. 설정 정보는 conf_mn.json을 파일을 통해 변경할 수 있다.



```
{
  "parent" : {
    "cbname": "Mobius",
    "cbcseid": "/Mobius",
    "cbhost": "203.253.128.161",
    "cbhostport": "7579",
    "cbprotocol": "http",
    "mqttbroker": "203.253.128.161"
  },
  "csebaseport": "7499",
  "dbpass": "dksdlfduq2"
}
```

Lavender 구동 시 필요한 설정은 위와 같이 아래부분에 Lavender가 오픈하는 포트 번호와 환경구축 시 설치한 MySQL 데이터베이스 접근 암호를 설정해줘야 한다.

parent 부분은 Lavender가 등록하는 IN-CSE의 정보를 입력한다.

- cbname : 등록할 IN-CSE 의 이름
- cbcseid: 등록할 IN-CSE 의 CSE-ID
- cbhost: 등록할 IN-CSE 의 Host 주소
- cbhostport: 등록할 IN-CSE 의 포트번호
- cbprotocal: 등록할 IN-CSE 의 프로토콜 (http or mqtt)
- mqttbroker: 등록할 IN-CSE 의 mqtt broker 주소

환경 설정이 완료되면 다음 그림과 같이 command prompt 창에서 node Lavender.js 를 실행한다. asn-cse setting success 라는 출력이 나오면 IN-CSE에 정상적으로 등록이 된 것이고, 이로써 nCube:Lavender의 구동이 완료가 된 것이다.

```

C:\WINDOWS\system32\cmd.exe - node lavender.js
select_direct /lavender (ryMLi0JLr-): 10.582ms
select_direct /lavender (rk7Lsu1USW): 12.042ms
select_resource cb /lavender (BJHJjdkLBb): 6.078ms
resource_retrieve /lavender (S1VUsdk8Hb): 6.281ms
select_resource cb /lavender (SyvLsd1UBZ): 7.231ms
resource_retrieve /lavender (ryL8odJ8S-): 7.748ms
<----- [pxy_coap]
{"m2m:cb":{"ty":5,"ct":"20170707T132849","ri":"S1q9nZaN-","rn":"lavender","lt":"20170707T132849","et":"20270707T132849","bl":["lavender"],"mni":3153600000,"cst":1,"csi":"/lavender","srt":["1","2","3","4","9","10","16","17","23","24","29","30"],"poa":["http://192.168.0.28:7499"]}}
[pxy_coap] coap ready
select_direct /lavender (H1uodk8BW): 99.885ms
subscribe req_topic as /oneM2M/req+/lavender/#
subscribe reg_req_topic as /oneM2M/reg_req+/lavender/#
ts_missing agent server (192.168.0.28) running at 7472 port
X-M2M-Origin: /lavender
select_direct /lavender (H1dsuJ1BW): 1.328ms
search_parents_lookup /lavender: 0.938ms
X-M2M-Origin: /lavender
select_direct /lavender (HybdsdJLSW): 1.014ms
select_direct /lavender/Mobius (BkzuiuJUH-): 2.022ms
"resource (Mobius) is already exist"
[[["register_remoteCSE"]]] {"rsc":"2000","ri":"/lavender","dbg":"asn-cse setting success"}
search_lookup (HJ1juJ8HZ): 48.601ms
init_TS - 4004

```

Lavender에 대한 설정을 추가적으로 변경하고 싶다면 Lavender.js 파일을 직접 수정해서 설정을 변경할 수 있다. Lavender.js 파일을 보면 다음과 같다.

```

// my CSE information
global.usecsetype      = 'asn'; // select 'in' or 'mn' or 'asn'
global.usecsebase      = 'lavender';
global.usecseid        = '/lavender';
global.usecsebaseport  = conf.csebaseport;

global.usedbhost       = 'localhost';
global.usedbpass       = conf.dbpass;

global.usepxywsport    = '7473';
global.usepxymqttport  = '7474';

global.usetsagentport  = '7472';

global.usemqttbroker   = parent_mqttbroker; // mobius to mqttbroker
global.usesecure       = 'disable';

```


Lavender.js 서버의 정보 설정

- usecsetype: 서버 타입, Lavender 는 ASN-CSE 로써 'asn'으로 입력해야 한다.
- usecsebase: Lavender 에 대한 CSEBase 이름을 나타낸다. Lavender 로 입력한다.
- usecseid: Lavender 의 cseid 를 입력한다. cseid 는 '/'로 시작하여야 하며, 디폴트로 csebase 이름 앞에 '/'를 붙여서 사용한다.
- usedbhost: DB 서버 주소, 디폴트로 localhost 를 사용한다.
- usemqttbroker: IN-CSE 가 사용하는 mqtt broker 를 사용한다. parent_mqttbroker 로 입력하여야 한다.