



**commit**

[DoJonghee](#) authored 4 hours ago

d1c34713

Name	Last commit	Last update
..		
image	install cm	1 day ago
readme.md	commit	4 hours ago

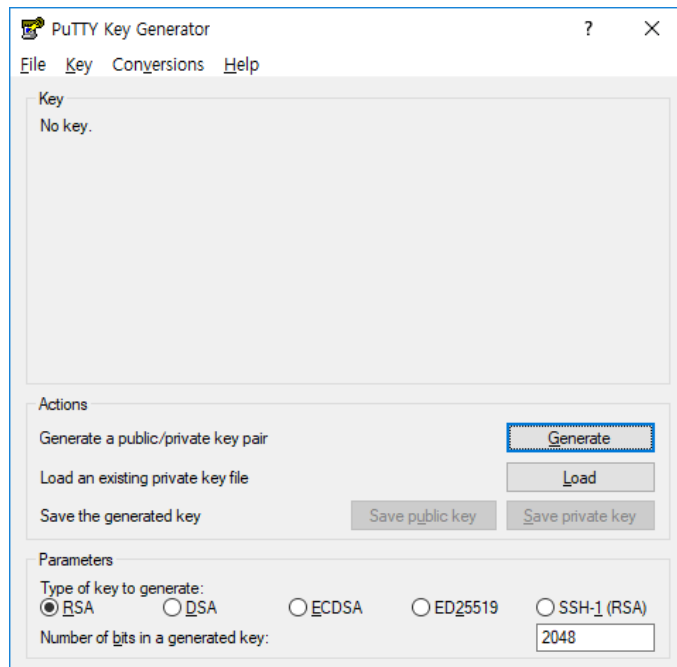
readme.md

## Intsall Cloudera Manager

### [PuTTY] private key(ppk,pem)를 이용하여 ssh접근하기

윈도우에서는 보통 PuTTY를 사용하여 ssh원격접속을한다.

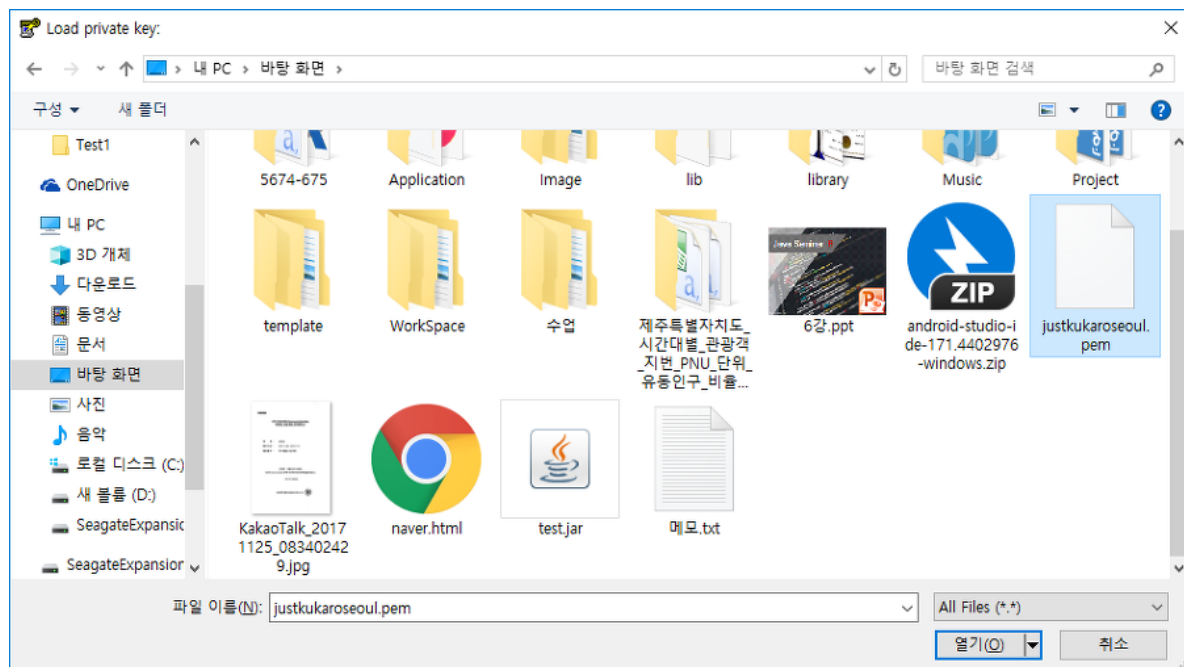
그런데 ec2같은 인스턴스를 사용할경우 일반적으로는 ppk를 사용한다.



만약 Key의 확장자가 처음부터 .ppk면 상관없지만 .pem이면 변형 시켜줘야한다.

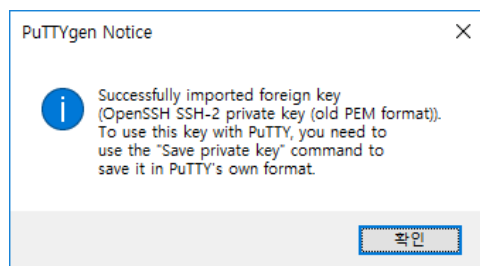
이때 사용하는건 PuTTY Key Generator를 사용해야한다.

먼저 Load를 눌러준다.



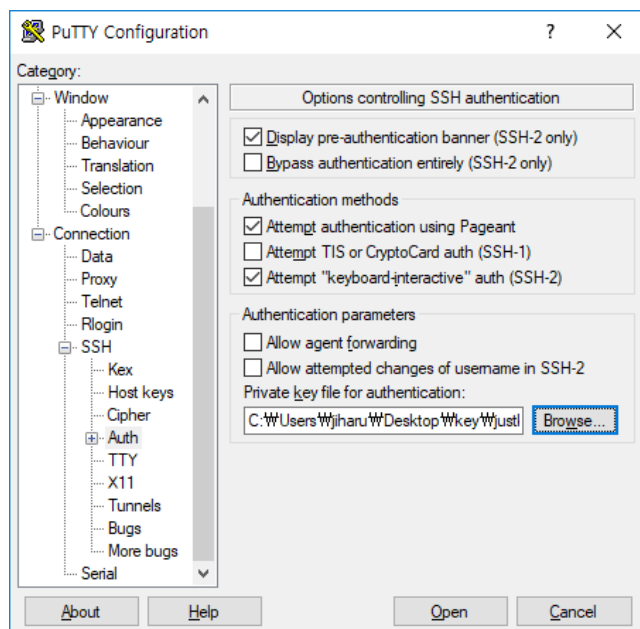
그 중에서 pem파일을 선택한다.

그 후 Save private key를 눌러준다.

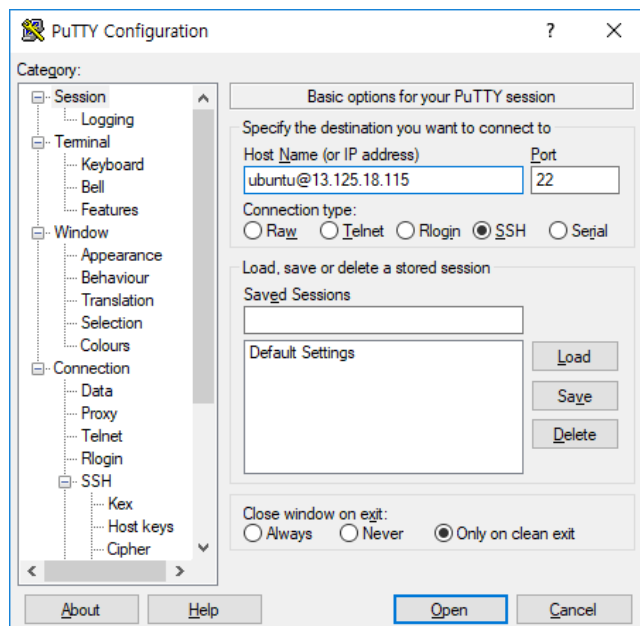


확인을 누르면 .pem파일이 .ppk파일로 변환된다.

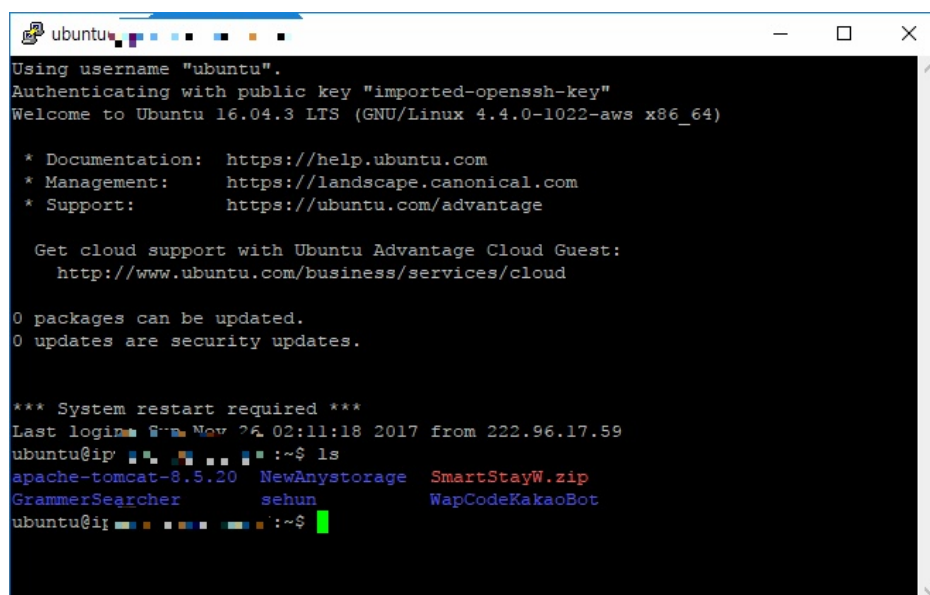
이제 준비는 끝났다.



일반적인 ssh와는 다르게 접속하기 전에 Auth에 들어가서 ppk를 Browse를 눌러서 선택해준다.



그다음 정상적으로 ssh를 사용한다.



이제 정상적으로 접근되는걸 알 수 있다.

#### Disable IPV6 (all server)

```
sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1
sudo sysctl -w net.ipv6.conf.default.disable_ipv6=1
```

#### sudo vi /etc/hosts (all server)

```
hostnamectl set-hostname 'cm.com'
hostnamectl set-hostname 'm1.com'
hostnamectl set-hostname 'd1.com'
hostnamectl set-hostname 'd2.com'
hostnamectl set-hostname 'd3.com'
```

```
172.31.7.144    cm.com  cm
172.31.7.241    m1.com  m1
172.31.7.112    d1.com  d1
172.31.3.217    d2.com  d2
172.31.14.4     d3.com  d3
```

#### Check hostname resolution (all server)

```
getent hosts cm.com
getent hosts m1.com
```

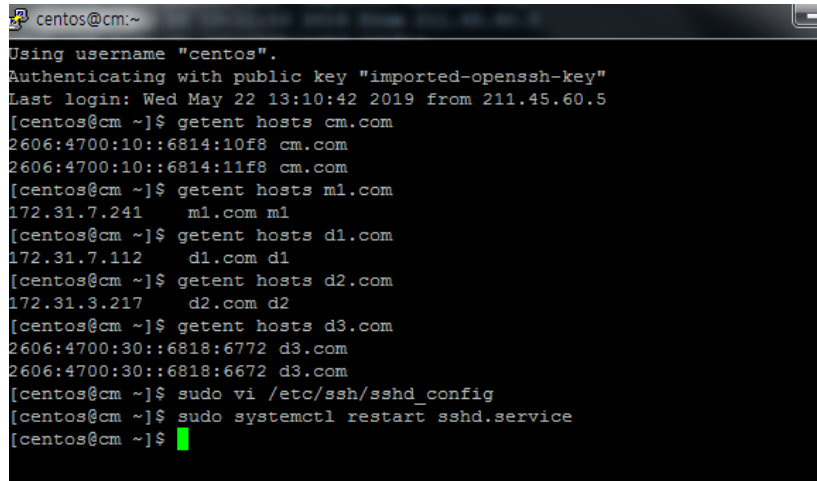
```
getent hosts d1.com
getent hosts d2.com
getent hosts d3.com
```

### reboot server (all server)

```
shutdown -r now
```

### Setup a password for centos (all server)

```
sudo passwd centos
sudo vi /etc/ssh/sshd_config
    change -> PasswordAuthentication yes
sudo systemctl restart sshd.service
```

A terminal window titled 'centos@cm:~' showing the process of setting up SSH. It starts with a login prompt for 'centos' using a public key. The user then runs 'getent hosts' for 'cm.com', 'm1.com', 'd1.com', 'd2.com', and 'd3.com', displaying IP addresses and hostnames. Finally, the user runs 'sudo vi /etc/ssh/sshd\_config' and 'sudo systemctl restart sshd.service' to configure password authentication and restart the service. The prompt returns to the user's shell.

```
centos@cm:~
Using username "centos".
Authenticating with public key "imported-openssh-key"
Last login: Wed May 22 13:10:42 2019 from 211.45.60.5
[centos@cm ~]$ getent hosts cm.com
2606:4700:10::6814:10f8 cm.com
2606:4700:10::6814:11f8 cm.com
[centos@cm ~]$ getent hosts m1.com
172.31.7.241 m1.com m1
[centos@cm ~]$ getent hosts d1.com
172.31.7.112 d1.com d1
[centos@cm ~]$ getent hosts d2.com
172.31.3.217 d2.com d2
[centos@cm ~]$ getent hosts d3.com
2606:4700:30::6818:6772 d3.com
2606:4700:30::6818:6672 d3.com
[centos@cm ~]$ sudo vi /etc/ssh/sshd_config
[centos@cm ~]$ sudo systemctl restart sshd.service
[centos@cm ~]$
```

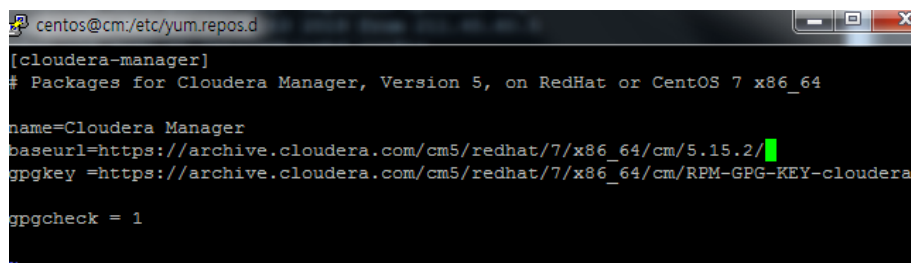
### Configure repository (cm)

```
sudo yum install -y wget
sudo wget https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/cloudera-manager.repo \
-P /etc/yum.repos.d/

** change the baseurl within cloudera-manager
https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/5.15.2/
```

```
sudo rpm --import https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/RPM-GPG-KEY-cloudera
```

```
sudo yum install -y cloudera-manager-daemons cloudera-manager-server
```

A terminal window titled 'centos@cm:/etc/yum.repos.d' showing the configuration of the Cloudera Manager repository. The user is editing the 'cloudera-manager.repo' file. The configuration includes the name 'Cloudera Manager', the base URL 'https://archive.cloudera.com/cm5/redhat/7/x86\_64/cm/5.15.2/', the GPG key URL 'https://archive.cloudera.com/cm5/redhat/7/x86\_64/cm/RPM-GPG-KEY-cloudera', and 'gpgcheck = 1'. The prompt returns to the user's shell.

```
centos@cm:/etc/yum.repos.d
[cloudera-manager]
# Packages for Cloudera Manager, Version 5, on RedHat or CentOS 7 x86_64

name=Cloudera Manager
baseurl=https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/5.15.2/
gpgkey=https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/RPM-GPG-KEY-cloudera
gpgcheck = 1
```

```
centos@cm:/etc/yum.repos.d
Install 2 Packages

Total download size: 752 M
Installed size: 933 M
Downloading packages:
(1/2): cloudera-manager-server-5.15.2-1.cm5152.p0.2.el7.x86_64 | 8.5 kB 00:00
(2/2): cloudera-manager-daemons-5.15.2-1.cm5152.p0.2.el7.x86_64 | 752 MB 00:17
-----
Total                                     43 MB/s | 752 MB 00:17
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : cloudera-manager-daemons-5.15.2-1.cm5152.p0.2.el7.x86_64 1/2
  Installing : cloudera-manager-server-5.15.2-1.cm5152.p0.2.el7.x86_64 2/2
  Verifying   : cloudera-manager-daemons-5.15.2-1.cm5152.p0.2.el7.x86_64 1/2
  Verifying   : cloudera-manager-server-5.15.2-1.cm5152.p0.2.el7.x86_64 2/2

Installed:
  cloudera-manager-daemons.x86_64 0:5.15.2-1.cm5152.p0.2.el7
  cloudera-manager-server.x86_64 0:5.15.2-1.cm5152.p0.2.el7

Complete!
[centos@cm yum.repos.d]$
```

### Installing the JDK Using Cloudera Manager (cm)

```
sudo yum install -y oracle-j2sdk1.7
```

```
centos@cm:/etc/yum.repos.d
=====
Installing:
  oracle-j2sdk1.7      x86_64      1.7.0+update67-1      cloudera-manager      135 M

Transaction Summary
=====
Install 1 Package

Total download size: 135 M
Installed size: 279 M
Downloading packages:
oracle-j2sdk1.7-1.7.0+update67-1.x86_64.rpm | 135 MB 00:03
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : oracle-j2sdk1.7-1.7.0+update67-1.x86_64 1/1
  Verifying   : oracle-j2sdk1.7-1.7.0+update67-1.x86_64 1/1

Installed:
  oracle-j2sdk1.7.x86_64 0:1.7.0+update67-1

Complete!
[centos@cm yum.repos.d]$
```

### Installing MariaDB Server (cm)

```
sudo yum install -y mariadb-server
```

```
centos@cm:/etc/yum.repos.d
Verifying : perl-IO-Compress-2.061-2.el7.noarch 6/11
Verifying : 1:perl-Compress-Raw-Zlib-2.061-4.el7.x86_64 7/11
Verifying : 1:mariadb-5.5.60-1.el7_5.x86_64 8/11
Verifying : perl-DBI-1.627-4.el7.x86_64 9/11
Verifying : libaio-0.3.109-13.el7.x86_64 10/11
Verifying : perl-PlRPC-0.2020-14.el7.noarch 11/11

Installed:
  mariadb-server.x86_64 1:5.5.60-1.el7_5

Dependency Installed:
  libaio.x86_64 0:0.3.109-13.el7
  mariadb.x86_64 1:5.5.60-1.el7_5
  perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.el7
  perl-Compress-Raw-Zlib.x86_64 1:2.061-4.el7
  perl-DBD-MySQL.x86_64 0:4.023-6.el7
  perl-DBI.x86_64 0:1.627-4.el7
  perl-Data-Dumper.x86_64 0:2.145-3.el7
  perl-IO-Compress.noarch 0:2.061-2.el7
  perl-Net-Daemon.noarch 0:0.48-5.el7
  perl-PlRPC.noarch 0:0.2020-14.el7

Complete!
[centos@cm yum.repos.d]$
```

### Configuring and Starting the MariaDB Server (cm)

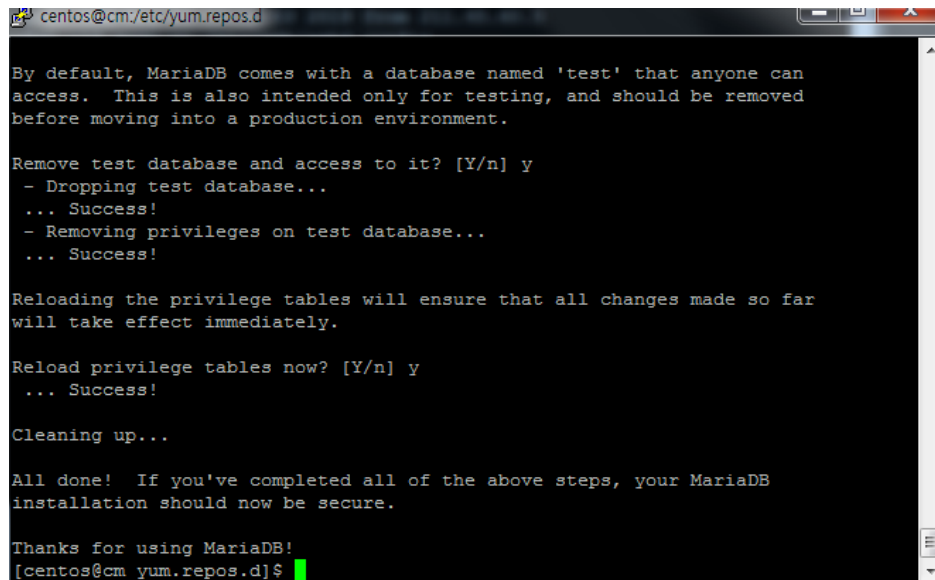
```
sudo systemctl stop mariadb
sudo vi /etc/my.cnf
```

```
sudo systemctl enable mariadb
```

```
sudo systemctl start mariadb
```

### Maria DB Password Setting (cm)

```
sudo /usr/bin/mysql_secure_installation
```



```
centos@cm:/etc/yum.repos.d

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

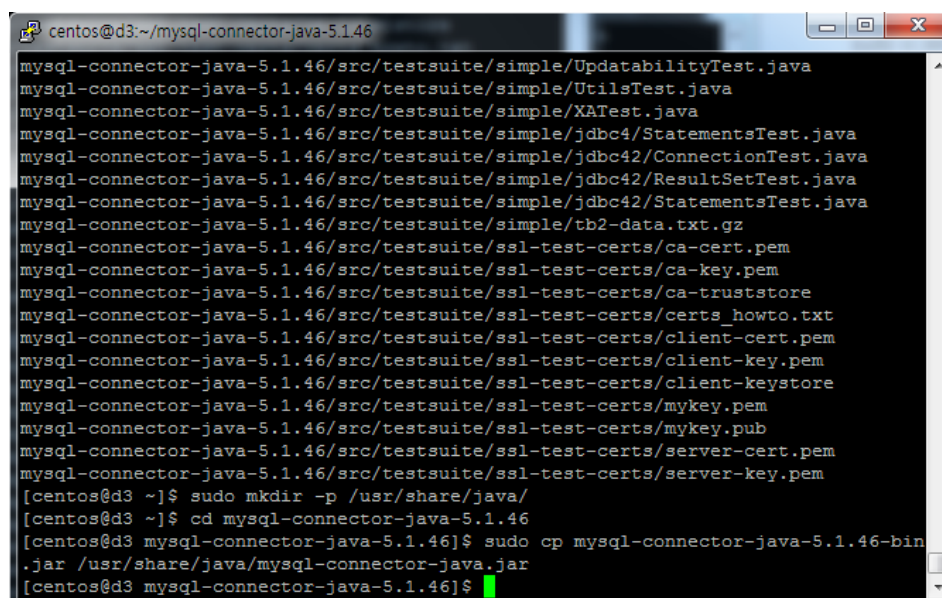
Thanks for using MariaDB!
[centos@cm yum.repos.d]$
```

### Installing the MySQL JDBC Driver for MariaDB (all server)

install all node

```
sudo wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.46.tar.gz

tar zxvf mysql-connector-java-5.1.46.tar.gz
sudo mkdir -p /usr/share/java/
cd mysql-connector-java-5.1.46
sudo cp mysql-connector-java-5.1.46-bin.jar /usr/share/java/mysql-connector-java.jar
```



```
centos@d3:~/mysql-connector-java-5.1.46

mysql-connector-java-5.1.46/src/testsuite/simple/UpdatabilityTest.java
mysql-connector-java-5.1.46/src/testsuite/simple/UtilsTest.java
mysql-connector-java-5.1.46/src/testsuite/simple/XATest.java
mysql-connector-java-5.1.46/src/testsuite/simple/jdbc4/StatementsTest.java
mysql-connector-java-5.1.46/src/testsuite/simple/jdbc42/ConnectionTest.java
mysql-connector-java-5.1.46/src/testsuite/simple/jdbc42/ResultSetTest.java
mysql-connector-java-5.1.46/src/testsuite/simple/jdbc42/StatementsTest.java
mysql-connector-java-5.1.46/src/testsuite/simple/tb2-data.txt.gz
mysql-connector-java-5.1.46/src/testsuite/ssl-test-certs/ca-cert.pem
mysql-connector-java-5.1.46/src/testsuite/ssl-test-certs/ca-key.pem
mysql-connector-java-5.1.46/src/testsuite/ssl-test-certs/ca-truststore
mysql-connector-java-5.1.46/src/testsuite/ssl-test-certs/certs_howto.txt
mysql-connector-java-5.1.46/src/testsuite/ssl-test-certs/client-cert.pem
mysql-connector-java-5.1.46/src/testsuite/ssl-test-certs/client-key.pem
mysql-connector-java-5.1.46/src/testsuite/ssl-test-certs/client-keystore
mysql-connector-java-5.1.46/src/testsuite/ssl-test-certs/mykey.pem
mysql-connector-java-5.1.46/src/testsuite/ssl-test-certs/mykey.pub
mysql-connector-java-5.1.46/src/testsuite/ssl-test-certs/server-cert.pem
mysql-connector-java-5.1.46/src/testsuite/ssl-test-certs/server-key.pem
[centos@d3 ~]$ sudo mkdir -p /usr/share/java/
[centos@d3 ~]$ cd mysql-connector-java-5.1.46
[centos@d3 mysql-connector-java-5.1.46]$ sudo cp mysql-connector-java-5.1.46-bin
.jar /usr/share/java/mysql-connector-java.jar
[centos@d3 mysql-connector-java-5.1.46]$
```

### 계정 및 Database 생성

```
mysql -u root -p

CREATE DATABASE scm DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON scm.* TO 'scm-user'@'%' IDENTIFIED BY 'somepassword';
```

```

CREATE DATABASE amon DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON amon.* TO 'amon-user'@'%' IDENTIFIED BY 'somepassword';

CREATE DATABASE rmon DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON rmon.* TO 'rmon-user'@'%' IDENTIFIED BY 'somepassword';

CREATE DATABASE hue DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON hue.* TO 'hue-user'@'%' IDENTIFIED BY 'somepassword';

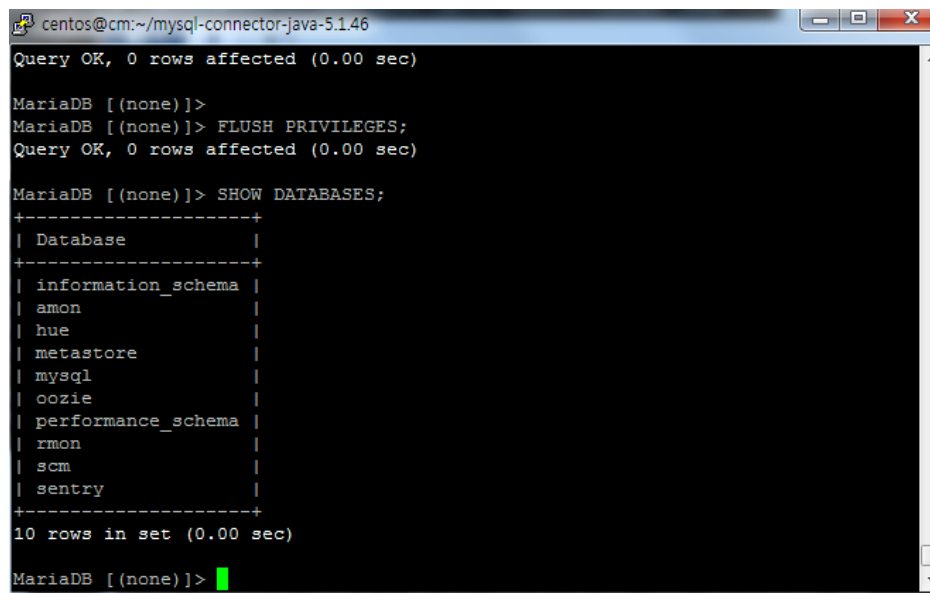
CREATE DATABASE metastore DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON metastore.* TO 'metastore-user'@'%' IDENTIFIED BY 'somepassword';

CREATE DATABASE sentry DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON sentry.* TO 'sentry-user'@'%' IDENTIFIED BY 'somepassword';

CREATE DATABASE oozie DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
GRANT ALL ON oozie.* TO 'oozie-user'@'%' IDENTIFIED BY 'somepassword';

FLUSH PRIVILEGES;
SHOW DATABASES;
EXIT;

```



```

centos@cm:~/mysql-connector-java-5.1.46
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| amon |
| hue |
| metastore |
| mysql |
| oozie |
| performance_schema |
| rmon |
| scm |
| sentry |
+-----+
10 rows in set (0.00 sec)

MariaDB [(none)]>

```

## Setup the CM database (cm)

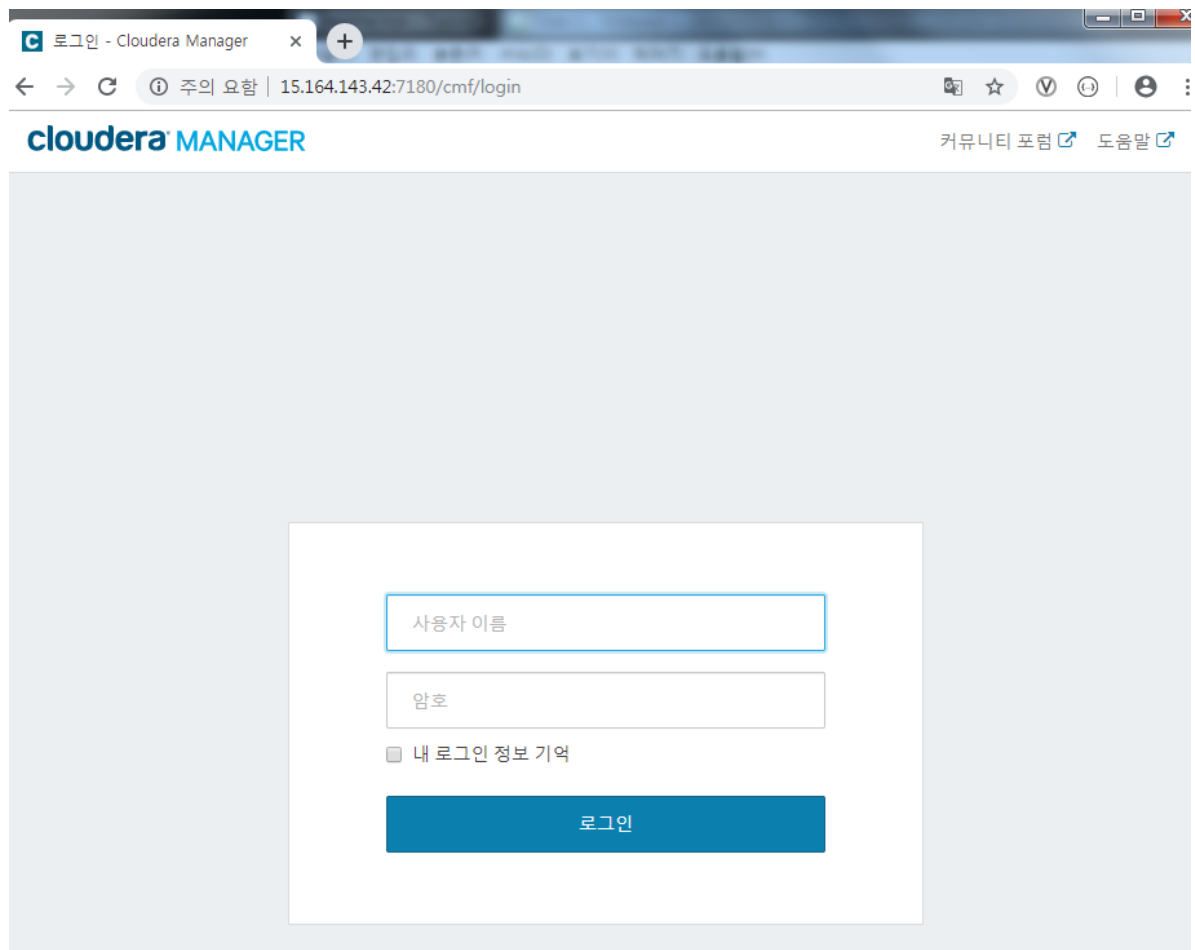
```

sudo /usr/share/cmf/schema/scm_prepare_database.sh mysql scm scm-user somepassword
sudo rm /etc/cloudera-scm-server/db.mgmt.properties
sudo systemctl start cloudera-scm-server

```

## Connect CM

```
http://15.164.143.42:7180
```



admin/ admin



자바관련 체크박스 모두 체크

단일사용자 모드 체크하지 않음

**CM** 설치작업



## 클러스터 설치

SSH 로그인 정보를 제공합니다.

Cloudera 패키지를 설치하려면 호스트에 대한 루트 액세스가 필요합니다. 이 설치 관리자는 SSH를 통해 호스트에 연결하고 루트로 직접 로그인하거나 암호 없이 `sudo/pbrun` 권한을 가진 다른 사용자로 로그인하여 루트가 됩니다.

모든 호스트를 다음으로 ☐ root

로그인:

☒ 다른 사용자

(암호 없는 sudo/pbrun을 사용하여 루트 액세스)

위에서 선택한 사용자에 대한 암호 또는 공용 키 인증을 통해 연결할 수 있습니다.

인증 방법: ☒ 모든 호스트가 동일한 암호 허용

☐ 모든 호스트가 동일한 개인 키 허용

암호 입력:

암호 확인:

SSH 포트:

동시에 진행하는 설치 수:

(많은 설치를 한꺼번에 실행하면 많은 양의 네트워크 대역폭 및 다른 시스템 리소스가 소모됩니다.)

## 클러스터 설정

### Select Services

설치할 서비스 조합을 선택하십시오.

☐ 코어 Hadoop

HDFS, YARN(MapReduce 2 포함), ZooKeeper, Oozie, Hive 및 Hue

☐ HBase가 있는 코어

HDFS, YARN(MapReduce 2 포함), ZooKeeper, Oozie, Hive, Hue 및 HBase

☒ Impala가 있는 코어

HDFS, YARN(MapReduce 2 포함), ZooKeeper, Oozie, Hive, Hue 및 Impala

참고: Cloudera Impala에 대한 적절한 라이선스가 있는지 확인하거나 도움이 필요할 경우 Cloudera에 문의하십시오.

☐ Search가 있는 코어

HDFS, YARN(MapReduce 2 포함), ZooKeeper, Oozie, Hive, Hue 및 Solr

☐ Spark가 있는 코어

HDFS, YARN(MapReduce 2 포함), ZooKeeper, Oozie, Hive, Hue 및 Spark

☐ 모든 서비스

HDFS, YARN(MapReduce 2 포함), ZooKeeper, Oozie, Hive, Hue, HBase, Impala, Solr, Spark 및 Key-Value Store Indexer

☐ 사용자 지정 서비스

보유한 서비스를 선택하십시오. 선택한 서비스에 필요한 서비스가 자동으로 포함됩니다. Flume은 초기 클러스터를 설정한 후에 추가할 수 있습니다.

이 방법사에서는 Cloudera Management Service도 설치합니다. 이 서비스는 모니터링, 보고, 이벤트 및 알림을 설정하는 컴포넌트 집합입니다. 이러한 컴포넌트를 사용하려면 정보를 저장할 데이터베이스(다음 페이지에서 구성할)가 필요합니다.

☐ Cloudera Navigator 포함

## 클러스터 설치

### Install Agents

설치가 완료되었습니다.

5개 중 5개의 호스트가 완료되었습니다.

호스트 이름	IP 주소	진행률	상태
cm.com	172.31.7.144	<div></div>	✓ 설치가 완료되었습니다.
d1.com	172.31.7.112	<div></div>	✓ 설치가 완료되었습니다.
d2.com	172.31.3.217	<div></div>	✓ 설치가 완료되었습니다.
d3.com	172.31.14.4	<div></div>	✓ 설치가 완료되었습니다.
m1.com	172.31.7.241	<div></div>	✓ 설치가 완료되었습니다.

## 클러스터 설치

### 선택한 Parcel을 설치하는 중

선택한 Parcel을 다운로드하여 클러스터의 모든 호스트에 설치하는 중입니다.

✓ CDH 5.15.2-1.cdh5.15.2.p0.3

다운로드됨: 100%

배포됨: 5/5 (227.8 M..)

압축 해제됨: 5/5

활성화됨: 5/5

#### HDFS

NameNode x 1 새로 만들기

cm.com

SecondaryNameNode x 1 새로 만들기

cm.com

Balancer x 1 새로 만들기

cm.com

HttpFS

호스트 선택

NFS Gateway

호스트 선택

DataNode x 3 새로 만들기

d[1-3].com

#### Hive

Gateway x 5 새로 만들기

cm.com; d[1-3].com; m1.com

Hive Metastore Server x 1 새로 만들기

cm.com

WebHCat Server

호스트 선택

HiveServer2 x 1 새로 만들기

cm.com

#### Hue

Hue Server x 1 새로 만들기

cm.com

Load Balancer x 1 새로 만들기

cm.com

#### Impala

Impala Catalog Server x 1 새로 만들기

cm.com

Impala StateStore x 1 새로 만들기

cm.com

Impala Daemon x 3 새로 만들기

d[1-3].com

#### Cloudera Management Service

Service Monitor x 1 새로 만들기

cm.com

Activity Monitor x 1 새로 만들기

cm.com

Host Monitor x 1 새로 만들기

cm.com

Reports Manager x 1 새로 만들기

cm.com

Event Server x 1 새로 만들기

cm.com

Alert Publisher x 1 새로 만들기

cm.com

Telemetry Publisher

호스트 선택

#### Oozie

Oozie Server x 1 새로 만들기

cm.com

## YARN (MR2 Included)

ResourceManager x 1 새로 만들기

m1.com ▼

JobHistory Server x 1 새로 만들기

m1.com ▼

NodeManager x 3 새로 만들기

d[1-3].com ▼

## ZooKeeper

Server x 2 새로 만들기

cm.com; m1.com ▼

데이터베이스 연결을 구성 및 테스트할 수 있습니다. [설치 가이드](#)의 [Installing and Configuring an External Database](#) 섹션에 설명된 대로 데이터베이스를 먼저 생성하십시오.

### Hive

✓ Successful

Database 호스트 이름: \*

cm.com

데이터베이스 유형:

MySQL ▼

데이터베이스 이름: \*

metastore

사용자 이름: \*

metastore

암호:

somepass

### Hue

✓ Successful

Database 호스트 이름: \*

cm.com

데이터베이스 유형:

MySQL ▼

데이터베이스 이름: \*

hue

사용자 이름: \*

hue-user

암호:

somepass

### Activity Monitor

✓ Successful

현재 cm.com에서 실행하도록 할당되었습니다.

Database 호스트 이름: \*

cm.com

데이터베이스 유형:

MySQL ▼

데이터베이스 이름: \*

amon

사용자 이름: \*

amon-user

암호:

somepass

### Reports Manager

✓ Successful

현재 cm.com에서 실행하도록 할당되었습니다.

Database 호스트 이름: \*

cm.com

데이터베이스 유형:

MySQL ▼

데이터베이스 이름: \*

rmon

사용자 이름: \*

rmon-user

암호:

somepass

### Oozie Server

✓ Successful

현재 cm.com에서 실행하도록 할당되었습니다.

Database 호스트 이름: \*

cm.com

데이터베이스 유형:

MySQL ▼

데이터베이스 이름: \*

oozie

사용자 이름: \*

oozie-user

암호:

somepass

☒ 암호 표시

테스트 연결

## 클러스터 설정

### 첫 번째 실행 명령

상태 ✓ 완료됨 May 22, 2:19:58 PM 3.6m

Finished First Run of the following services successfully: ZooKeeper, HDFS, YARN (MR2 Included), Hive, Impala, Oozie, Hue, Cloudera Management Service.

✓ 8/8단계가 완료되었습니다.

☒ Show All Steps

☐ Show Only Failed Steps

☐ Show Running Steps

➤ ✓ Ensuring that the expected software releases are installed on hosts.

May 22, 2:19:58 PM

➤ ✓ 클라이언트 구성 배포\*\*

[Cluster 1](#)

May 22, 2:19:58 PM

➤ ✓ Cloudera Management Service, ZooKeeper 시작

May 22, 2:20:13 PM

➤ ✓ HDFS 시작

May 22, 2:20:38 PM

➤ ✓ YARN (MR2 Included) 시작

May 22, 2:21:20 PM

➤ ✓ Hive 시작

May 22, 2:21:47 PM

➤ ✓ Impala, Oozie 시작

May 22, 2:22:29 PM

➤ ✓ Hue 시작

May 22, 2:23:11 PM

설치완료

## 클러스터 설정

축하합니다!

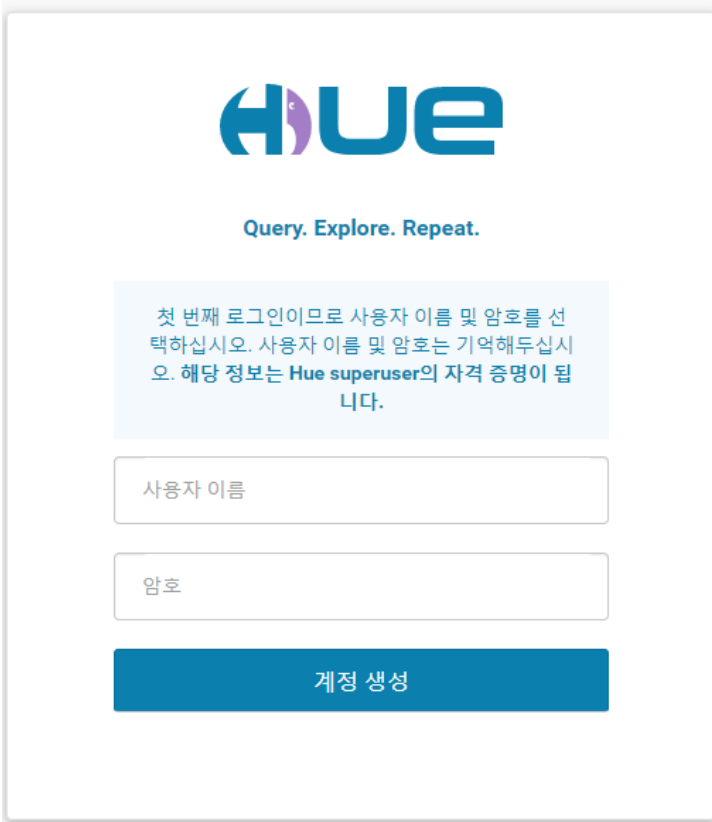
✔ 서비스가 클러스터에 설치 및 구성되어 실행 중입니다.

### Add Training User and wheel group (all server)

```
adduser training
passwd training
usermod -aG wheel training
```

```
[centos@cm mysql-connector-java-5.1.46]$
[centos@cm mysql-connector-java-5.1.46]$ sudo su
[root@cm mysql-connector-java-5.1.46]# adduser training
[root@cm mysql-connector-java-5.1.46]# passwd training
Changing password for user training.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: all authentication tokens updated successfully.
[root@cm mysql-connector-java-5.1.46]#
```

### Login Hue - Using web UI

The image shows the Hue web interface login screen. At the top is the Hue logo, followed by the tagline "Query. Explore. Repeat." Below this is a light blue box containing Korean text: "첫 번째 로그인이므로 사용자 이름 및 암호를 선택하십시오. 사용자 이름 및 암호는 기억해두십시오. 해당 정보는 Hue superuser의 자격 증명이 됩니다." Underneath the text are two input fields: "사용자 이름" (Username) and "암호" (Password). At the bottom is a large blue button labeled "계정 생성" (Create Account).

training / training 으로 로그인

윈도우 파일 리눅스로 옮기기

CMD실행 후 파일이 존재하는 경로로 이동 다음 명령 실행 (CM과 D1에 작업)

```
pscp.exe authors.sql training@15.164.143.42:/home/training
pscp.exe posts.sql training@15.164.143.42:/home/training
```

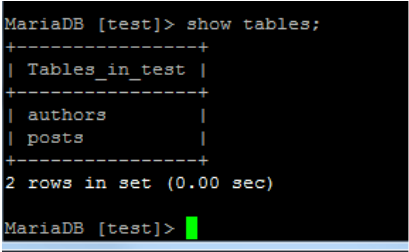
### 데이터베이스 생성

```
mysql -u root -p

CREATE DATABASE test;
SHOW DATABASES;
EXIT;
```

테이블 생성

```
use test
source posts.sql
source authors.sql
```



db training user add (cm)

```
mysql -u root -p
GRANT ALL ON *.* TO 'training'@'%' IDENTIFIED BY 'training';
FLUSH PRIVILEGES;
EXIT;
```

Sqoop --> Hive

```
sqoop import \
--connect jdbc:mysql://cm/test \
--username training --password training \
--table posts \
--hive-import

sqoop import \
--connect jdbc:mysql://cm/test \
--username training --password training \
--table authors \
--hive-import
```

< default

테이블 (2)

필터...

authors

id (int)

first\_name (string)

last\_name (string)

email (string)

birthdate (string)

added (string)

posts

id (int)

author\_id (int)

title (string)

description (string)

content (string)

date (string)

run Hive/Impala query

```
1 select * from posts
2
```

```
INFO : Compiling command(queryId=hive_20190522150404_d482de57-c125-4ded-ad0e-c5f974d84dbb): select * from posts
INFO : Semantic Analysis Completed
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:posts.id, type:int, comment:null), FieldSchema(name:posts.author_id, type:int, comment:null), FieldSchema(name:posts.title, type:string, comment:null), FieldSchema(name:posts.description, type:string, comment:null), FieldSchema(name:posts.content, type:string, comment:null), FieldSchema(name:posts.created_at, type:timestamp, comment:null)], tableName:posts)
```

쿼리 기록 🔍 저장된 쿼리 🔍 결과 (200+) 🔍

	posts.id	posts.author_id	posts.title	posts.description
1	1	1	Cupiditate et aperiam et maxime rerum id molestias quaerat.	Nostrum eos quos et. Qui nemo qui et hic qui. Voluptas ad lauc
2	2	2	Excepturi quia neque recusandae et officia blanditiis.	Sapiente incidunt occaecati qui laboriosam error beatae in. Du
3	3	3	Enim rerum quae rerum doloribus numquam et.	Officiis distinctio sit est voluptatem fugiat. Ut ab enim sint eos
4	4	4	Labore ipsam et qui et.	Tempore suscipit repellat dolores nobis dicta cupiditate. Moles
5	5	5	Placeat exercitationem dicta dolore enim in.	Ea magnam tempore illum eaque sed eum et. Voluptas earum
6	6	6	Non nihil eum molestiae veniam consequuntur.	Ea odit delectus sunt et. Non molestias nemo eveniet consequ
7	7	7	Est et autem est cupiditate et.	Corrupti nam iusto fugiat et hic odio sit. Nulla vel sint voluptas.
8	8	8	Dolorem ratione provident vero pariatur voluptatem ratione dolorem.	Ad quis fugiat pariatur quia exercitationem numquam. Debitis i
9	9	9	Eum cumque et praesentium voluptatem.	Reprehenderit ullam dolor minima unde aut necessitatibus. Itai
10	10	10	Voluptatum expedita enim nostrum eligendi.	Et ullam doloribus porro omnis optio quaerat. Cum doloremq
11	11	11	Dolor alias quis repellat fugit alias quia ut.	Explicabo labore libero occaecati. Qui rerum iste cumque alias
12	12	12	Dolores voluptatem aperiam incidunt corrupti accusamus est.	Non voluptas aliquid at ab. Aut in et labore numquam necessiti