【实验名称】2台机器搭建MySQL集群环境

考虑到虚拟机数量过多会影响性能，本实验使用2台linux虚拟机作为实验环境。Linux 选择Ubuntu 16.04 Server. 事先在VMware上安装linux操作系统。 安装设置Windows 与Linux虚拟机共享文件夹参考如下：
为了方便编辑配置文件，可以先安装 vim

```
sudo apt-get install vim
```

1. 集群配置

数据节点1：192.168.50.128 nodeID 11
数据节点2：192.168.50.129 nodeID 12
管理节点：192.168.50.129 nodeID 1③
sql节点1：192.168.50.128 nodeID
sql节点2：192.168.50.129 nodeID

使用2台机器进行配置，各台机器上的均配置为SQL服务节点、数据节点与sql节点；192.168.50.129上配置一个管理节点。

（如果大家有条件建立多台虚拟机，可以将管理节点和SQL服务节点分开，也可以不再所有的数据节点上配置SQL节点。）

对于MySQL-Cluster的安装包下载，下载地址见http://dev.mysql.com/downloads/cluster/，务必根据linux操作系统选择正确的版本，并下载。

如果之前安装过mysql-server，在进行此次实验之前，需要将mysql-server卸载，执行以下指令卸载mysql

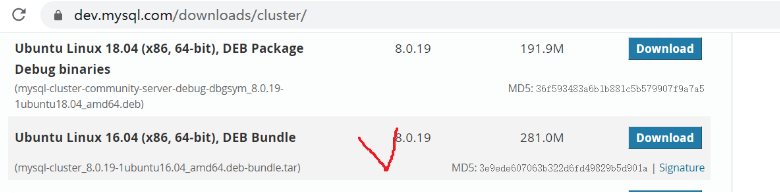
```
sudo apt-get autoremove --purge mysql-server
sudo apt-get remove mysql-server
sudo apt-get autoremove mysql-server
sudo apt-get remove mysql-common (非常重要)
```

2. 准备（192.168.150.129) ③

安装

```
shell> adduser mysql
shell> usermod -aG sudo mysql
```

下载 mysql-cluster_8.0.19-1ubuntu16.04_amd64.deb-bundle.tar


dev.mysql.com/downloads/cluster/

Ubuntu Linux 18.04 (x86, 64-bit), DEB Package Debug binaries (mysql-cluster-community-server-debug-dbgsym_8.0.19-1ubuntu18.04_amd64.deb)	8.0.19	191.9M	Download MD5: 36f593483a6b1b881c5b579907f9a7a5
Ubuntu Linux 16.04 (x86, 64-bit), DEB Bundle (mysql-cluster_8.0.19-1ubuntu16.04_amd64.deb-bundle.tar)	8.0.19	281.0M	Download MD5: 3e9ede607063b322d6fd49829b5d901a | Signature

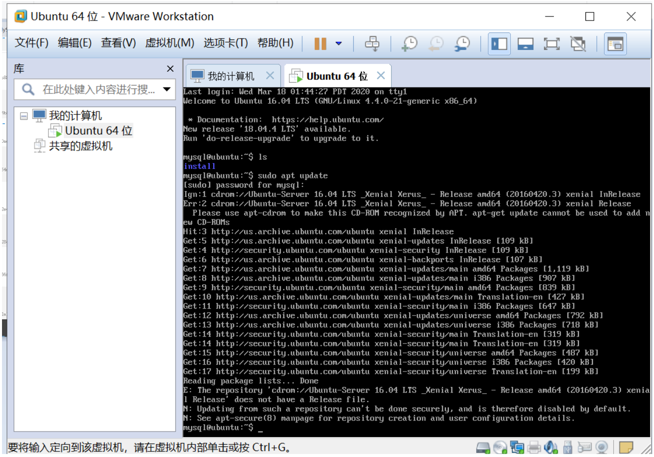
并放入Windows下的共享文件夹，然后在Linux虚拟机中存取。也可以在Linux虚拟机中下载。
假定已经获取mysql-cluster_8.0.19-1ubuntu16.04_amd64.deb-bundle.tar到mysql home目录。
解压文件到install目录

Shell> mkdir install
shell> tar -xvf mysql-cluster_8.0.19-1ubuntu16.04_amd64.deb-bundle.tar -C install/
shell> `cd install`

在安装 MySQL server binary前，可能需要安装一些依赖包

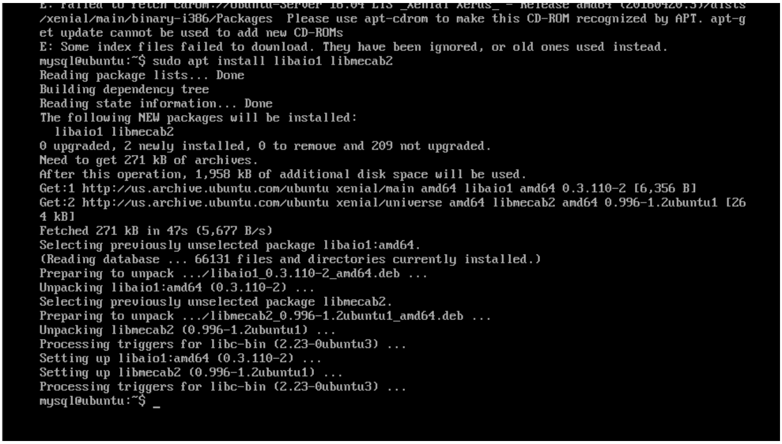
shell> `sudo apt update`

（也许还需要 `sudu apt-get update`）


Ubuntu 64 位 - VMware Workstation
文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H)
库
在此处键入内容进行搜...
我的计算机
Ubuntu 64 位
共享的虚拟机
我的计算机
Ubuntu 64 位
要将输入定向到该虚拟机，请在虚拟机内部单击或按 Ctrl+G。

```
Last login: Wed Mar 18 01:44:27 PDT 2020 on tty1
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-21-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

mysql@ubuntu:~$ ls
install
mysql@ubuntu:~$ sudo apt update
[sudo] password for mysql:
Ign:1 cdrom://Ubuntu-Server 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.3) xenial InRelease
Err:2 cdrom://Ubuntu-Server 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.3) xenial Release
  Please use apt-cdrom to make this CD-ROM recognized by APT. apt-get update cannot be used to add n
ew CD-ROMs
Hit:3 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Get:5 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [1,119 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu xenial-updates/main i386 Packages [907 kB]
Get:9 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [839 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu xenial-updates/main Translation-en [427 kB]
Get:11 http://security.ubuntu.com/ubuntu xenial-security/main i386 Packages [647 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [792 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu xenial-updates/universe i386 Packages [718 kB]
Get:14 http://security.ubuntu.com/ubuntu xenial-security/main Translation-en [319 kB]
Get:14 http://security.ubuntu.com/ubuntu xenial-security/main Translation-en [319 kB]
Get:15 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 Packages [487 kB]
Get:16 http://security.ubuntu.com/ubuntu xenial-security/universe i386 Packages [420 kB]
Get:17 http://security.ubuntu.com/ubuntu xenial-security/universe Translation-en [199 kB]
Reading package lists... Done
E: The repository 'cdrom://Ubuntu-Server 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.3) xenia
l Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
mysql@ubuntu:~$ _
```

shell> `sudo apt install libaio1 libmecab2`



```
E: Failed to fetch cdrom://Ubuntu-Server 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.3)/dists
/xenial/main/binary-i386/Packages  Please use apt-cdrom to make this CD-ROM recognized by APT. apt-g
et update cannot be used to add new CD-ROMs
E: Some index files failed to download. They have been ignored, or old ones used instead.
mysql@ubuntu:~$ sudo apt install libaio1 libmecab2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  libaio1 libmecab2
0 upgraded, 2 newly installed, 0 to remove and 209 not upgraded.
Need to get 271 kB of archives.
After this operation, 1,958 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 libaio1 amd64 0.3.110-2 [6,356 B]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 libmecab2 amd64 0.996-1.2ubuntu1 [26
4 kB]
Fetched 271 kB in 47s (5,677 B/s)
Selecting previously unselected package libaio1:amd64.
(Reading database ... 66131 files and directories currently installed.)
Preparing to unpack .../libaio1_0.3.110-2_amd64.deb ...
Unpacking libaio1:amd64 (0.3.110-2) ...
Selecting previously unselected package libmecab2.
Preparing to unpack .../libmecab2_0.996-1.2ubuntu1_amd64.deb ...
Unpacking libmecab2 (0.996-1.2ubuntu1) ...
Processing triggers for libc-bin (2.23-0ubuntu3) ...
Setting up libaio1:amd64 (0.3.110-2) ...
Setting up libmecab2 (0.996-1.2ubuntu1) ...
Processing triggers for libc-bin (2.23-0ubuntu3) ...
mysql@ubuntu:~$ _
```

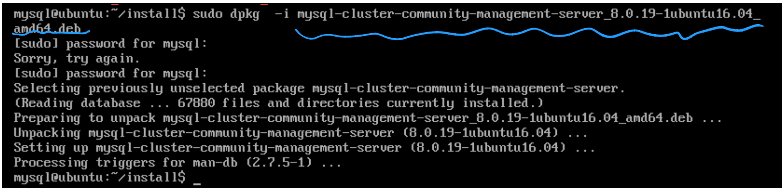
克隆一个虚拟机，按着集群配置设置IP地址。这样就有两台linux有mysql cluster包的服务器了。

3 安装配置集群管理器

3

用dpkg指令在Cluster Manager 服务器 (假定为 192.168.50.129)上安装 ndb_mgmd。

shell>sudo dpkg -i mysql-cluster-community-management-server_7.6.6-1ubuntu1~~8.04~~_amd64.deb



```
mysql@ubuntu:~/install$ sudo dpkg  -i mysql-cluster-community-management-server_8.0.19-1ubuntu16.04_
amd64.deb
[sudo] password for mysql:
Sorry, try again.
[sudo] password for mysql:
Selecting previously unselected package mysql-cluster-community-management-server.
(Reading database ... 67880 files and directories currently installed.)
Preparing to unpack mysql-cluster-community-management-server_8.0.19-1ubuntu16.04_amd64.deb ...
Unpacking mysql-cluster-community-management-server (8.0.19-1ubuntu16.04) ...
Setting up mysql-cluster-community-management-server (8.0.19-1ubuntu16.04) ...
Processing triggers for man-db (2.7.5-1) ...
mysql@ubuntu:~/install$ _
```

在第一次运行ndb_mgmd前需要对其进行配置，正确配置是保证数据节点正确同步和负载分配的前提。

Cluster Manager 应该是MySQL Cluster 第一个启动的组件.它需要一个配置文件来加载参数.我门创建配置文件: /var/lib/mysql-cluster/config.ini.

在Cluster Manager 所在机器上创建 /var/lib/mysql-cluster目录:

```
shell>sudo mkdir /var/lib/mysql-cluster
```

然后用你熟悉的编辑器编辑配置文件:

shell>sudo vim /var/lib/mysql-cluster/config.ini

内容如下：

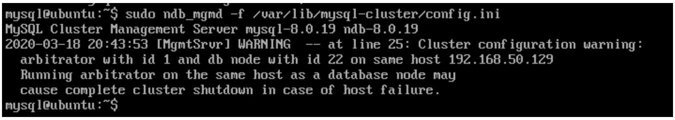
```
[ndbd default]
# Options affecting ndbd processes on all data nodes:
NoOfReplicas=2  # Number of replicas

[ndb_mgmd]
# Management process options:
hostname=192.168.50.129 # Hostname of the manager
NodeId=1
datadir=/var/lib/mysql-cluster  # Directory for the log files

[ndbd]
hostname=192.168.50.128 # Hostname/IP of the first data node
NodeId=11           # Node ID for this data node
datadir=/usr/local/mysql/data   # Remote directory for the data files

[ndbd]
hostname=192.168.50.129 # Hostname/IP of the second data node
NodeId=12           # Node ID for this data node
datadir=/usr/local/mysql/data   # Remote directory for the data files

[mysqld]
# SQL node options:
hostname=192.168.50.129 # MySQL server/client i manager

[mysqld]
# SQL node options:
hostname=192.168.50.128 # MySQL server/client i manager
```

注意IP地址要正确

如果是生产环境，应该根据实际情况调整配置参数，参考MySQL Cluster. 你还可以增加data nodes (ndbd) 或 MySQL server nodes (mysqld).

现在可以启动管理器，执行:

shell>sudo ndb_mgmd -f /var/lib/mysql-cluster/config.ini



```
mysql@ubuntu:~$ sudo ndb_mgmd -f /var/lib/mysql-cluster/config.ini
MySQL Cluster Management Server mysql-8.0.19 ndb-8.0.19
2020-03-18 20:43:53 [MgmtSrvr] WARNING  -- at line 25: Cluster configuration warning:
  arbitrator with id 1 and db node with id 22 on same host 192.168.50.129
  Running arbitrator on the same host as a database node may
  cause complete cluster shutdown in case of host failure.
mysql@ubuntu:~$
```

检查ndb_mgmd 使用的端口 1186：

```
Shell>sudo netstat -plntu
```

从Windows编辑config.ini可能会报错，需要在Linux系统下编辑。

（在启动服务前，可能需要杀掉正在运行的服务：**Shell>sudo pkill -f ndb_mgmd**）

下面打开并编辑下面 systemd Unit 文件，使服务可以自动加载：:

```
Shell> sudo vim /etc/systemd/system/ndb_mgmd.service
```

键下面文件内容:

```
[Unit]
Description=MySQL NDB Cluster Management Server
After=network.target auditd.service

[Service]
Type=forking
ExecStart=/usr/sbin/ndb_mgmd -f /var/lib/mysql-cluster/config.ini
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure

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

保存并关闭文件.

这里只是加入了如何启动、停止和重启动ndb_mgmd 进程的最小选项集合. 想要知道更多的相关信息，参阅systemd manual.
采用daemon-reload重新加载systemd’s manager 配置:

```
Shell>sudo systemctl daemon-reload
```

让我们刚创建的服务生效，使MySQL Cluster Manage 可以开机执行:

```
Shell>sudo systemctl enable ndb_mgmd
```

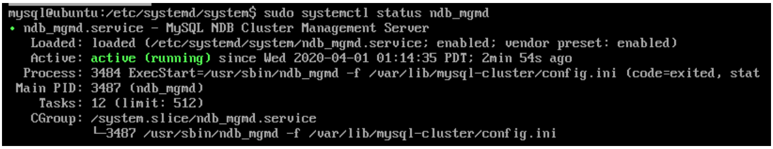
最后，启动服务：

```
Shell> sudo systemctl start ndb_mgmd
```

可以通过如下语句验证NDB Cluster Management service服务正在执行：

```
Shell>sudo systemctl status ndb_mgmd
```

输出类似下面内容:



```
mysql@ubuntu:/etc/systemd/system$ sudo systemctl status ndb_mgmd
• ndb_mgmd.service - MySQL NDB Cluster Management Server
   Loaded: loaded (/etc/systemd/system/ndb_mgmd.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-04-01 01:14:35 PDT; 2min 54s ago
  Process: 3484 ExecStart=/usr/sbin/ndb_mgmd -f /var/lib/mysql-cluster/config.ini (code=exited, stat
 Main PID: 3487 (ndb_mgmd)
    Tasks: 12 (limit: 512)
   CGroup: /system.slice/ndb_mgmd.service
           └─3487 /usr/sbin/ndb_mgmd -f /var/lib/mysql-cluster/config.ini
```

这表明ndb_mgmd MySQL Cluster Management server 作为一个系统服务运行了。

最后一步是设置Cluster Manager 允许其它MySQL Cluster 节点连入.

如果没有配置过ufw 防火墙,可以跳过这个操作，进入第4步..

添加允许数据节点连入的规则：

Shell>sudo ufw allow from 192.168.50.128

Shell>sudo ufw allow from 192.168.50.129

会见到如下输出：

```
Output
Rule added
```

现在Cluster Manager 应该启动运行了，并且能够通过局域网与集群其它节点通信了。

③

4 配置一个数据节点(假定在192.168.50.129上进行)

安装依赖包

Shell>sudo apt-get –f install
Shell>sudo apt install libclass-methodmaker-perl

安装数据节点包

进入之前创建的 install 文件夹

Shell>sudo dpkg -i mysql-cluster-community-data-node_8.0.19-1ubuntu~~18.04~~16.04_amd64.deb

数据节点将从固定位置/etc/my.cnf获取配置文件.创建文件并编辑:

Shell>sudo vim /etc/my.cnf

```
[mysql_cluster]
# Options for NDB Cluster processes:
ndb-connectstring=192.168.50.129  # location of cluster manager
```

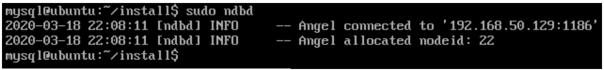
本配置设定在管理器配置数据目录为 /usr/local/mysql/data. 运行服务前要创建相关目录:

```
Shell>sudo mkdir -p /usr/local/mysql/data
```

然后就可以启动服务了：

```
Shell>sudo ndbd
```

显示如下：



```
mysql@ubuntu:~/install$ sudo ndbd
2020-03-18 22:08:11 [ndbd] INFO     -- Angel connected to '192.168.50.129:1186'
2020-03-18 22:08:11 [ndbd] INFO     -- Angel allocated nodeid: 22
mysql@ubuntu:~/install$
```

NDB 数据节点守护程序成功启动!!!

如果出现连接问题，请打开防火墙：

```
Shell>sudo ufw allow from 192.168.50.129
Shell>sudo ufw allow from 192.168.50.128
```

配置数据节点服务自启动：

打开并编辑如下 systemd Unit 文件:

```
Shell> sudo vim /etc/systemd/system/ndbd.service
```

内容如下:

```
[Unit]
Description=MySQL NDB Data Node Daemon
After=network.target auditd.service

[Service]
Type=forking
ExecStart=/usr/sbin/ndbd
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure

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

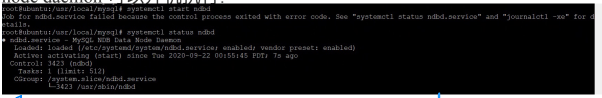
这里只是加入了如何启动、停止和重启动ndbd进程的最小选项集合. 想要知道更多的相关信息，参阅systemd manual.

采用daemon-reload重新加载systemd’s manager 配置:

Shell>sudo systemctl daemon-reload

让我们刚创建的服务生效，使data node daemon 可以开机执行:
Shell>sudo systemctl enable ndbd

最后，启动服务:
Shell> sudo systemctl start ndbd



```
root@ubuntu:/usr/local/mysql# systemctl start ndbd
Job for ndbd.service failed because the control process exited with error code. See "systemctl status ndbd.service" and "journalctl -xe" for d
etails.
root@ubuntu:/usr/local/mysql# systemctl status ndbd
● ndbd.service - MySQL NDB Data Node Daemon
   Loaded: loaded (/etc/systemd/system/ndbd.service; enabled; vendor preset: enabled)
   Active: activating (start) since Tue 2020-09-22 00:55:45 PDT; 7s ago
  Control: 3423 (ndbd)
    Tasks: 1 (limit: 512)
   CGroup: /system.slice/ndbd.service
           └─3423 /usr/sbin/ndbd
```

解决方法

①检查文件是否有写错 ② sudo pkill -f ndbd 杀死 ndbd 进程，再从 daemon-reload 开始来一遍.

可以通过如下语句验证NDB Cluster Management service服务正在执行:
Shell>sudo systemctl status ndbd
输出类似下面内容:

```
   ndbd.service - MySQL NDB Data Node Daemon
  Loaded: loaded (/etc/systemd/system/ndbd.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu ****-**-** 20:56:29 UTC; 8s ago
 Process: 11972 ExecStart=/usr/sbin/ndbd (code=exited, status=0/SUCCESS)
Main PID: 11984 (ndbd)
   Tasks: 46 (limit: 4915)
  CGroup: /system.slice/ndbd.service
          ├─11984 /usr/sbin/ndbd
          └─11987 /usr/sbin/ndbd
```

这表明MySQL Cluster 数据节点守护进程作为一个 systemd service在运行. 数据节点起作用并能够连接到MySQL Cluster Manager.

其它数据节点可以采用类似步骤配置.

5 配置并运行MySQL Server 和 Client

标准的MySQL server不支持 MySQL Cluster 引擎 NDB. 这意味着我们需要安装含有定制的SQL服务器 MySQL Cluster软件.

进入包含MySQL Cluster组件的目录:

Shell>cd install

在安装MySQL server 前，需要安装两个依赖库（如果已经安装过可忽略）:

Shell>sudo apt update
Shell>sudo apt install libaio1 libmecab2

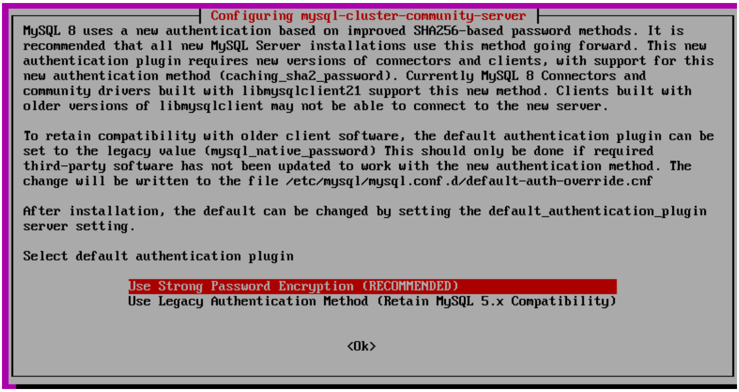
然后安装解压在install目录的软件包中的一些MySQL Cluster 依赖包:

Shell>sudo dpkg -i mysql-common_8.0.19-1ubuntu16.04_amd64.deb

-i

```
Shell>sudo dpkg -i mysql-cluster-community-client-core_8.0.19-1ubuntu16.04_amd64.deb

Shell>sudo dpkg -i mysql-cluster-community-client_8.0.19-1ubuntu16.04_amd64.deb

Shell>sudo dpkg -i mysql-client_8.0.19-1ubuntu16.04_amd64.deb

Shell>sudo dpkg -i mysql-cluster-community-server-core_8.0.19-1ubuntu16.04_amd64.deb

Shell>sudo dpkg -i mysql-cluster-community-server_8.0.19-1ubuntu16.04_amd64.deb
```

当安装mysql-cluster-community-server时，会出现配置提示，请求为mysql数据库root用户设置密码。请选择一个强安全全密码然后敲 **<Ok>**. 提示时重新输入这个**root** 密码，再次敲 **<Ok>**，完成安装. (实验为了避免遗忘，可以：123456)



```
┤ Configuring mysql-cluster-community-server ├
MySQL 8 uses a new authentication based on improved SHA256-based password methods. It is
recommended that all new MySQL Server installations use this method going forward. This new
authentication plugin requires new versions of connectors and clients, with support for this
new authentication method (caching_sha2_password). Currently MySQL 8 Connectors and
community drivers built with libmysqlclient21 support this new method. Clients built with
older versions of libmysqlclient may not be able to connect to the new server.

To retain compatibility with older client software, the default authentication plugin can be
set to the legacy value (mysql_native_password) This should only be done if required
third-party software has not been updated to work with the new authentication method. The
change will be written to the file /etc/mysql/mysql.conf.d/default-auth-override.cnf

After installation, the default can be changed by setting the default_authentication_plugin
server setting.

Select default authentication plugin

              Use Strong Password Encryption (RECOMMENDED)
              Use Legacy Authentication Method (Retain MySQL 5.x Compatibility)


                                   <Ok>
```

然后，安装MySQL server :

```
Shell>sudo dpkg -i mysql-server_8.0.19-1ubuntu16.04_amd64.deb
```

配置MySQL server.
MySQL Server 配置文件默认为 /etc/mysql/my.cnf.
打开这个配置文件:

```
Shell>sudo vim /etc/mysql/my.cnf
```

可以看到下列文本:

/etc/mysql/my.cnf

```
# Copyright (c) 2015, 2016, Oracle and/or its affiliates. All rights reserved.
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; version 2 of the License.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
```

```
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301 USA


#
# The MySQL Cluster Community Server configuration file.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html


# * IMPORTANT: Additional settings that can override those from this file!
#   The files must end with '.cnf', otherwise they'll be ignored.
#
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/
```

向其中追加下列配置信息：

/etc/mysql/my.cnf

```
. . .
[mysqld]
# Options for mysqld process:
ndbcluster                      # run NDB storage engine


[mysql_cluster]
# Options for NDB Cluster processes:
ndb-connectstring=192.168.50.129  # location of management server
```

保存文件并退出.
重启 MySQL server，使上面的变化生效:

```
Shell>sudo systemctl restart mysql
```

MySQL默认开机自动启动.如果不能启动,下述命令可以修复:

```
Shell>sudo systemctl enable mysql
```

下面，运行几条命令验证MySQL Cluster正常工作.

6 验证MySQL Cluster安装

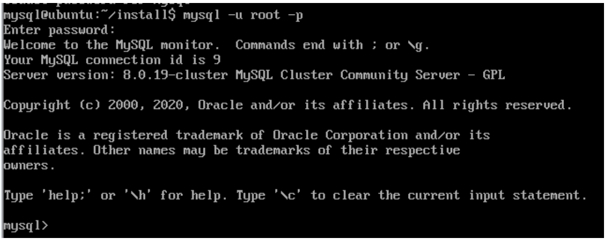
为了验证 MySQL Cluster正确安装, 登陆 Cluster Manager / SQL Server节点（本指南为192.168.50.129）③.

打开MySQL 客户端连接到**root** 账号:

```
Shell>mysql -u root -p
```

输入password并回车.

应该出现类似如下提示：

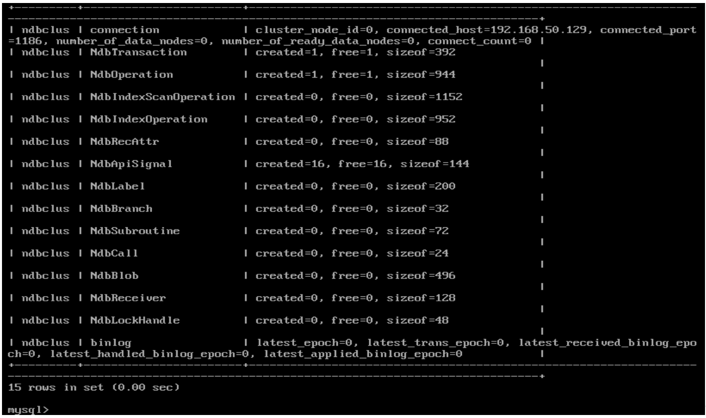


```
mysql@ubuntu:~/install$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.19-cluster MySQL Cluster Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

在MySQL客户端中, 运行下列命令:

SHOW ENGINE NDB STATUS \G

系统会显示 NDB引擎的相关信息:



```
+---------+-----------------------+-----------------------------------------------------------------
-----------------------------------------------------------------------------+
| ndbclus | connection            | cluster_node_id=0, connected_host=192.168.50.129, connected_port
=1186, number_of_data_nodes=0, number_of_ready_data_nodes=0, connect_count=0 |
| ndbclus | NdbTransaction        | created=1, free=1, sizeof=392
                                                                             |
| ndbclus | NdbOperation          | created=1, free=1, sizeof=944
                                                                             |
| ndbclus | NdbIndexScanOperation | created=0, free=0, sizeof=1152
                                                                             |
| ndbclus | NdbIndexOperation     | created=0, free=0, sizeof=952
                                                                             |
| ndbclus | NdbRecAttr            | created=0, free=0, sizeof=88
                                                                             |
| ndbclus | NdbApiSignal          | created=16, free=16, sizeof=144
                                                                             |
| ndbclus | NdbLabel              | created=0, free=0, sizeof=200
                                                                             |
| ndbclus | NdbBranch             | created=0, free=0, sizeof=32
                                                                             |
| ndbclus | NdbSubroutine         | created=0, free=0, sizeof=72
                                                                             |
| ndbclus | NdbCall               | created=0, free=0, sizeof=24
                                                                             |
| ndbclus | NdbBlob               | created=0, free=0, sizeof=496
                                                                             |
| ndbclus | NdbReceiver           | created=0, free=0, sizeof=128
                                                                             |
| ndbclus | NdbLockHandle         | created=0, free=0, sizeof=48
                                                                             |
| ndbclus | binlog                | latest_epoch=0, latest_trans_epoch=0, latest_received_binlog_epo
ch=0, latest_handled_binlog_epoch=0, latest_applied_binlog_epoch=0           |
+---------+-----------------------+-----------------------------------------------------------------
-----------------------------------------------------------------------------+
15 rows in set (0.00 sec)

mysql>
```

这表示成功连入MySQL Cluster.

注意number of ready_data_nodes= 2. 如果一个数据节点挂了（本例中必须是那个没有安装MySQL Cluster管理器的节点），MySQL cluster还是可以继续工作。

你可以试着shutting down 非管理器节点（192.168.50.128）或停止管理器节点上的ndbd服务来测试 cluster稳定性，如果你重启机器或服务，在整个过程中你会看到number_of_ready_data_nodes变为1并再次变回 2.

退出MySQL 客户端，使用quit 或按CTRL-D.

在集群管理器控制台上查看集群信息，命令如下:

```
shell>ndb_mgm
```

输出如下：

```
-- NDB Cluster -- Management Client --
ndb_mgm>
```

然后输入SHOW命令并回车:

```
SHOW
```

输出如下：

```
Connected to Management Server at: 192.168.50.129:1186

Cluster Configuration
---------------------
[ndbd(NDB)] 2 node(s)
id=2    @192.168.50.128  (mysql-×.×.×× ndb-8.0.19, Nodegroup: 0, *)
id=3    @192.168.50.129  (mysql-×.×.×× ndb-8.0.19, Nodegroup: 0)


[ndb_mgmd(MGM)] 1 node(s)
id=1    @192.168.50.129  (mysql-×.×.×× ndb-8.0.19)


[mysqld(API)]   1 node(s)
id=4    @192.168.50.129  (mysql-5.7.22 ndb-7.6.6)
```

退出管理控制台用quit命令.

管理控制台功能很多，有很多其他的管理命令来管理集群和数据, 包括创建在线备份. 更多信息参考官方official MySQL documentation.

下面向MySQL集群插入数据.

7 向MySQL集群插入数据

注意为了使用集群功能, 必须使用**NDB数据库引擎**. 如果使用InnoDB (default)或其他引擎,将不能使用集群.

首先, 创建数据库clustertest:

```
CREATE DATABASE clustertest;
```

其次转到新数据库:

```
USE clustertest;
```

再次，创建表test_table:

```
CREATE TABLE test_table (name VARCHAR(20), value VARCHAR(20)) ENGINE=ndbcluster;
```

这里需要显式规定ndbcluster引擎.

现在可以插入数据了:

```
INSERT INTO test_table (name,value) VALUES('some_name','some_value');
```

最后验证数据插入：

```
SELECT * FROM test_table;
```

思考：在本例中，数据被插入到了哪个机器?

可以在my.cnf 文件中设定默认数据存储引擎为 ndbcluster. 这样创建表时就不再规定引擎了. 更多信息参考MySQL Reference Manual.

至此我们在Ubuntu 16.04 servers 上安装和配置了 a MySQL Cluster. 需要注意的是这是一个很小的简化体系结构来说明配置过程，部署一个生产环境，还有许多其他的选项和特征需要去学习. 更多信息请参阅 MySQL Cluster documentation.

注释： systemctl is command line utility and primary tool to manage the systemd daemons/services such as (start, restart, stop, enable, disable, reload & status).

问题:
1 通过实验，你对一个分布式数据库系统有何理解？分布式数据库系统预计有何优越性?
2你能设计一个方案验证集群系统在可靠性上优于集中式数据库系统吗?
2 同样是插入数据，你觉得MySQL Cluster和myCAT 在实体完整性保持方面是否可能会有不同？为什麼?