安装或升级MySQL

**章节目录**

本章描述了如何获得和安装MySQL。下面是该程序的摘要，后面的章节提供了细节。如果你计划将现有版本的MySQL升级到较新的版本，而不是首次安装MySQL，请参见第2.11节 "升级MySQL"，以了解有关升级程序和升级前应考虑的问题的信息。

如果你对从另一个数据库系统迁移到MySQL感兴趣，请参阅A.8节 "MySQL 8.0常见问题。迁移"，其中包含对一些有关迁移问题的常见问题的回答。

MySQL的安装通常遵循这里概述的步骤。

1、确定MySQL是否在你的平台上运行并被支持。

请注意，并非所有平台都同样适合运行MySQL，而且并非所有已知MySQL可以运行的平台都得到甲骨文公司的正式支持。关于那些正式支持的平台的信息，请参见MySQL网站上的<https://www.mysql.com/support/supportedplatforms/database.html>。

2、选择要安装的版本。

有几个版本的MySQL可用，而且大多数都有几种发布格式。你可以从包含二进制（预编译）程序或源代码的预包装发行版中进行选择。如果有疑问，请使用二进制发行版。甲骨文公司还为那些想看最新发展和测试新代码的人提供了对MySQL源代码的访问。要确定你应该使用哪个版本和发行版的类型，请参见第2.1.2节，"安装哪个MySQL版本和发行版"。、

3、下载你要安装的发行版。

有关说明，见第2.1.3节，"如何获得MySQL"。要验证发行版的完整性，请使用第2.1.4节 "使用MD5校验和或GnuPG验证包的完整性 "中的说明。

4、安装该发行版。

要从二进制发行版安装MySQL，请使用第2.2节 "使用通用二进制文件在Unix/Linux上安装MySQL "中的说明。或者，使用《安全部署指南》，该指南提供了部署MySQL企业版服务器通用二进制发行版的程序，该发行版具有管理MySQL安装安全的功能。

要从源发布版或当前开发源树中安装MySQL，请使用第2.9节 "从源安装MySQL "中的说明。

5、执行任何必要的安装后设置。

在安装完MySQL后，请参阅第2.10节 "安装后的设置和测试"，了解有关确保MySQL服务器正常工作的信息。还请参考第2.10.4节 "确保初始MySQL账户的安全 "中提供的信息。本节描述了如何保护初始MySQL根用户账户，该账户在你分配密码之前没有密码。无论你使用二进制或源码发行版安装MySQL，本节都适用。

6、如果你想运行MySQL的基准脚本，必须有对MySQL的Perl支持。见第2.13节，"Perl安装说明"。

在不同的平台和环境中安装MySQL的说明可按平台来进行。

* Unix、Linux、FreeBSD

关于在大多数Linux和Unix平台上使用通用二进制文件（例如，.tar.gz包）安装MySQL的说明，见第2.2节，"在Unix/Linux上使用通用二进制文件安装MySQL"。

关于完全从源代码发行版或源代码库构建MySQL的信息，见第2.9节，"从源代码安装MySQL"

关于安装、配置和从源码构建的具体平台帮助，见相应的平台部分。

* + Linux，包括关于发行版特定方法的说明，见第2.5节，"在Linux上安装MySQL"。
  + IBM AIX，见第2.7节，"在Solaris上安装MySQL"。
  + FreeBSD，见第2.8节，"在FreeBSD上安装MySQL"。
* Windows系统

关于使用MySQL安装程序或Zipped二进制文件在Microsoft Windows上安装MySQL的说明，见第2.3节，"在Microsoft Windows上安装MySQL"。

关于使用Microsoft Visual Studio从源代码构建MySQL的细节和说明，见第2.9节，"从源代码安装MySQL"。

* macOS

关于在macOS上的安装，包括使用二进制包和本地PKG格式，见第2.4节，"在macOS上安装MySQL"。

关于利用macOS启动守护程序来自动启动和停止MySQL的信息，见第2.4.3节，"安装和使用MySQL启动守护程序"。

关于MySQL首选项窗格的信息，见第2.4.4节，"安装和使用MySQL首选项窗格"。

## 2.1 一般安装指南

紧接着的章节包含了选择、下载和验证你的发行版的必要信息。本章后面几节的说明描述了如何安装你选择的发行版。对于二进制发行版，请参见第2.2节 "在Unix/Linux上使用通用二进制文件安装MySQL "的说明，或你的平台的相应章节（如果有）。要从源码构建MySQL，请使用第2.9节 "从源码安装MySQL "中的说明。

### 2.1.1 支持的平台

MySQL平台的支持随着时间的推移而变化，请参考https://www.mysql.com/support/supportedplatforms/database.html，以了解最新的更新。

### 2.1.2 安装哪个MySQL版本和发行版

当准备安装MySQL时，要决定使用哪种版本和发布格式（二进制或源代码）。

首先，决定是安装开发版本还是通用版本（GA）。开发版本有最新的功能，但不建议用于生产。GA版本，也被称为生产或稳定版本，是为生产使用准备的。我们建议使用最新的GA版本。

MySQL 8.0中的命名方案使用由三个数字和一个可选的后缀组成的版本名称（例如，mysql-8.0.1-dmr）。版本名称中的数字解释如下。

* 第一个数字（8）是主要版本号。
* 第二个数字(0)是次要版本号。主版本号和次版本号加在一起，构成了版本系列号。该系列号描述了稳定的功能集。
* 第三个数字（1）是版本系列中的版本号。每一个新的错误修复版本都会递增。在大多数情况下，一个系列中的最新版本是最好的选择。

版本名称也可以包括一个后缀，以表示该版本的稳定性水平。一个系列中的版本通过一组后缀来表示稳定性水平的提高。可能的后缀是。

* dmr表示一个开发里程碑版本（DMR）。MySQL开发使用一个里程碑模型，每个里程碑都引入了一小部分经过彻底测试的特性。从一个里程碑到下一个里程碑，根据试用这些早期版本的社区成员提供的反馈，功能界面可能会发生变化，甚至可能被删除。里程碑版本中的功能可以被认为是生产前的质量。
* rc表示候选发布版（RC）。候选版本被认为是稳定的，已经通过了MySQL的所有内部测试。在RC版本中仍可能引入新的功能，但重点转向修复错误，以稳定该系列中早期引入的功能。
* 没有后缀的版本表示是通用版本（GA）或生产版本。GA版本是稳定的，已经成功通过了早期的发布阶段，并且被认为是可靠的，没有严重的错误，适合在生产系统中使用。

一个系列内的开发从DMR版本开始，接着是RC版本，最后达到GA状态的版本。

在选择了要安装的MySQL版本后，决定为你的操作系统安装哪种发布格式。对于大多数使用情况，二进制发行版是正确的选择。二进制发行版可用于许多平台的本地格式，如Linux的RPM包或macOS的DMG包。分发版也可以用更通用的格式，如Zip档案或压缩的tar文件。在Windows上，你可以使用MySQL安装程序来安装二进制发行版。

在某些情况下，从源码发行版安装MySQL可能更可取。

* 你想在某个明确的位置安装MySQL。标准的二进制发行版可以在任何安装位置运行，但你可能需要更多的灵活性来把MySQL组件放在你想要的地方。
* 你想用可能不包括在标准二进制发行版中的功能来配置mysqld。下面是一个最常见的额外选项的列表，用于确保功能的可用性。
  + -DWITH\_LIBWRAP=1用于支持TCP包装器。
  + -DWITH\_ZLIB={system|bundled}，用于依赖压缩的功能。
  + -DWITH\_DEBUG=1用于调试支持

有关其他信息，请参见第2.9.7节，"MySQL源配置选项"。

* 你想配置mysqld，但不包括标准二进制发行版中的某些功能。
* 你想阅读或修改组成MySQL的C和C++代码。为此目的，获得一个源码发行版。
* 源码发行版比二进制发行版包含更多的测试和例子。

### 2.1.3 如何获得MySQL

请查看我们的下载页面https://dev.mysql.com/downloads/，了解有关当前版本的MySQL的信息和下载说明。

对于使用Yum作为其软件包管理系统的基于RPM的Linux平台，可以使用MySQL Yum Repository来安装MySQL。详情见第2.5.1节，"使用MySQL Yum仓库在Linux上安装MySQL"。

对于基于Debian的Linux平台，可以使用MySQL APT资源库来安装MySQL。详情请参阅第2.5.2节 "使用MySQL APT资源库在Linux上安装MySQL"。

对于SUSE Linux Enterprise Server（SLES）平台，可以使用MySQL SLES资源库安装MySQL。详情见第2.5.3节 "使用MySQL SLES资源库在Linux上安装MySQL"。

要获得最新的开发源码，见第2.9.5节 "使用开发源码树安装MySQL"。

### 2.1.4 使用MD5校验和或GnuPG验证软件包的完整性

在下载了适合你的需求的MySQL包后，在试图安装它之前，要确保它是完整的，没有被篡改过。有三种完整性检查的手段。

* MD5校验和
* 使用GNUPG（GNU Privacy Guard）的加密签名
* 对于RPM包，内置的RPM完整性验证机制

下面的章节描述了如何使用这些方法。

如果你发现MD5校验和或GPG签名不匹配，首先尝试再下载一次相应的包，也许是从另一个镜像站点。

#### 2.1.4.1 验证MD5校验和

在你下载了一个MySQL包之后，你应该确保它的MD5校验和与MySQL下载页面上提供的校验和一致。每个包都有一个单独的校验和，你可以对照你下载的包来验证。正确的MD5校验和在每个MySQL产品的下载页面上列出；你应该将它与你下载的文件（产品）的MD5校验和进行比较。

每个操作系统和设置都提供自己版本的工具来检查MD5校验和。通常情况下，该命令被命名为md5sum，或者它可能被命名为md5，一些操作系统根本就不提供它。在Linux上，它是GNU Text Utilities软件包的一部分，该软件包可用于各种不同的平台。你也可以从 http://www.gnu.org/software/textutils/ 下载源代码。如果你已经安装了OpenSSL，你可以使用openssl md5 package\_name命令来代替。md5命令行工具的Windows实现可以从http://www.fourmilab.ch/md5/。winMd5Sum是一个图形化的MD5检查工具，可以从http://www.nullriver.com/index/products/winmd5sum。我们的Windows系统示例假定名称为md5.exe。

Linux和Microsoft Windows的例子。

shell> **md5sum mysql-standard-8.0.26-linux-i686.tar.gz**

aaab65abbec64d5e907dcd41b8699945 mysql-standard-8.0.26-linux-i686.tar.gz

shell> **md5.exe mysql-installer-community-8.0.26.msi**

aaab65abbec64d5e907dcd41b8699945 mysql-installer-community-8.0.26.msi

你应该验证所产生的校验和（十六进制数字的字符串）与紧接在相应软件包下面的下载页面上显示的校验和相匹配。

注意

*确保验证归档文件（例如，.zip、.tar.gz或.msi文件）的校验和，而不是归档文件中包含的文件。换句话说，在提取文件内容之前要先验证该文件。*

#### 2.1.4.2 使用GnuPG进行签名检查

另一种验证包的完整性和真实性的方法是使用加密签名。这比使用MD5校验和更可靠，但需要更多工作。

我们用GnuPG（GNU Privacy Guard）来签署MySQL可下载包。GnuPG是Phil Zimmermann著名的Pretty Good Privacy（PGP）的一个开源替代方案。大多数Linux发行版都默认安装了GnuPG。否则，请参阅 http://www.gnupg.org/ 了解更多关于 GnuPG 的信息以及如何获得和安装它。

为了验证特定软件包的签名，你首先需要获得一份我们公开的GPG构建密钥的副本，你可以从http://pgp.mit.edu/ 下载。你想获得的密钥名为mysql-build@oss.oracle.com。另外，你也可以直接从以下文本中复制和粘贴密钥。

-----BEGIN PGP PUBLIC KEY BLOCK-----

Version: GnuPG v1

mQGiBD4+owwRBAC14GIfUfCyEDSIePvEW3SAFUdJBtoQHH/nJKZyQT7h9bPlUWC3

RODjQReyCITRrdwyrKUGku2FmeVGwn2u2WmDMNABLnpprWPkBdCk96+OmSLN9brZ

fw2vOUgCmYv2hW0hyDHuvYlQA/BThQoADgj8AW6/0Lo7V1W9/8VuHP0gQwCgvzV3

BqOxRznNCRCRxAuAuVztHRcEAJooQK1+iSiunZMYD1WufeXfshc57S/+yeJkegNW

hxwR9pRWVArNYJdDRT+rf2RUe3vpquKNQU/hnEIUHJRQqYHo8gTxvxXNQc7fJYLV

K2HtkrPbP72vwsEKMYhhr0eKCbtLGfls9krjJ6sBgACyP/Vb7hiPwxh6rDZ7ITnE

kYpXBACmWpP8NJTkamEnPCia2ZoOHODANwpUkP43I7jsDmgtobZX9qnrAXw+uNDI

QJEXM6FSbi0LLtZciNlYsafwAPEOMDKpMqAK6IyisNtPvaLd8lH0bPAnWqcyefep

rv0sxxqUEMcM3o7wwgfN83POkDasDbs3pjwPhxvhz6//62zQJ7Q2TXlTUUwgUmVs

ZWFzZSBFbmdpbmVlcmluZyA8bXlzcWwtYnVpbGRAb3NzLm9yYWNsZS5jb20+iGwE

ExECACwCGyMCHgECF4ACGQEGCwkIBwMCBhUKCQgCAwUWAgMBAAUCXEBY+wUJI87e

5AAKCRCMcY07UHLh9RZPAJ9uvm0zlzfCN+DHxHVaoFLFjdVYTQCfborsC9tmEZYa

whhogjeBkZkorbyIaQQTEQIAKQIbIwYLCQgHAwIEFQIIAwQWAgMBAh4BAheAAhkB

BQJTAdRmBQkaZsvLAAoJEIxxjTtQcuH1X4MAoKNLWAbCBUj96637kv6Xa/fJuX5m

AJwPtmgDfjUe2iuhXdTrFEPT19SB6ohmBBMRAgAmAhsjBgsJCAcDAgQVAggDBBYC

AwECHgECF4AFAk53PioFCRP7AhUACgkQjHGNO1By4fUmzACeJdfqgc9gWTUhgmcM

AOmG4RjwuxcAoKfM+U8yMOGELi+TRif7MtKEms6piGkEExECACkCGyMGCwkIBwMC

BBUCCAMEFgIDAQIeAQIXgAIZAQUCUZSROgUJFTchqgAKCRCMcY07UHLh9YtAAJ9X

rA/ymlmozPZn+A9ls8/uwMcTsQCfaQMNq1dNkhH2kyByc3Rx9/W2xfqJARwEEAEC

AAYFAlAS6+UACgkQ8aIC+GoXHivrWwf/dtLk/x+NC2VMDlg+vOeM0qgG1IlhXZfi

NsEisvvGaz4m8fSFRGe+1bvvfDoKRhxiGXU48RusjixzvBb6KTMuY6JpOVfz9Dj3

H9spYriHa+i6rYySXZIpOhfLiMnTy7NH2OvYCyNzSS/ciIUACIfH/2NH8zNT5CNF

1uPNRs7HsHzzz7pOlTjtTWiF4cq/Ij6Z6CNrmdj+SiMvjYN9u6sdEKGtoNtpycgD

5HGKR+I7Nd/7v56yhaUe4FpuvsNXig86K9tI6MUFS8CUyy7Hj3kVBZOUWVBM053k

nGdALSygQr50DA3jMGKVl4ZnHje2RVWRmFTr5YWoRTMxUSQPMLpBNIkBHAQQAQIA

BgUCU1B+vQAKCRAohbcD0zcc8dWwCACWXXWDXIcAWRUw+j3ph8dr9u3SItljn3wB

c7clpclKWPuLvTz7lGgzlVB0s8hH4xgkSA+zLzl6u56mpUzskFl7f1I3Ac9GGpM4

0M5vmmR9hwlD1HdZtGfbD+wkjlqgitNLoRcGdRf/+U7x09GhSS7Bf339sunIX6sM

gXSC4L32D3zDjF5icGdb0kj+3lCrRmp853dGyA3ff9yUiBkxcKNawpi7Vz3D2ddU

pOF3BP+8NKPg4P2+srKgkFbd4HidcISQCt3rY4vaTkEkLKg0nNA6U4r0YgOa7wIT

SsxFlntMMzaRg53QtK0+YkH0KuZR3GY8B7pi+tlgycyVR7mIFo7riQEcBBABCAAG

BQJWgVd0AAoJEEZu4b/gk4UKk9MH/Rnt7EccPjSJC5CrB2AU5LY2Dsr+PePI2ubP

WsEdG82qSjjGpbhIH8LSg/PzQoGHiFWMmmZWJktRT+dcgLbs3b2VwCNAwCE8jOHd

UkQhEowgomdNvHiBHKHjP4/lF68KOPiO/2mxYYkmpM7BWf3kB57DJ5CTi3/JLoN7

zF40qIs/p09ePvnwStpglbbtUn7XPO+1/Ee8VHzimABom52PkQIuxNiVUzLVn3bS

Wqrd5ecuqLk6yzjPXd2XhDHWC9Twpl68GePru6EzQtusi0m6S/sHgEXqh/IxrFZV

JlljF75JvosZq5zeulr0i6kOij+Y1p6MFffihITZ1gTmk+CLvK2JASIEEAECAAwF

Ak53QS4FAwASdQAACgkQlxC4m8pXrXwJ8Qf/be/UO9mqfoc2sMyhwMpN4/fdBWwf

LkA12FXQDOQMvwH9HsmEjnfUgYKXschZRi+DuHXe1P7l8G2aQLubhBsQf9ejKvRF

TzuWMQkdIq+6Koulxv6ofkCcv3d1xtO2W7nb5yxcpVBPrRfGFGebJvZa58DymCNg

yGtAU6AOz4veavNmI2+GIDQsY66+tYDvZ+CxwzdYu+HDV9HmrJfc6deM0mnBn7SR

jqzxJPgoTQhihTav6q/R5/2p5NvQ/H84OgS6GjosfGc2duUDzCP/kheMRKfzuyKC

OHQPtJuIj8++gfpHtEU7IDUX1So3c9n0PdpeBvclsDbpRnCNxQWU4mBot4kBIgQQ

AQIADAUCToi2GQUDABJ1AAAKCRCXELibyletfLZAB/9oRqx+NC98UQD/wlxCRytz

vi/MuPnbgQUPLHEap10tvEi33S/H/xDR/tcGofY4cjAvo5skZXXeWq93Av7PACUb

zkg0X0eSr2oL6wy66xfov72AwSuX+iUK68qtKaLqRLitM02y8aNRV/ggKvt7UMvG

mOvs5yLaYlobyvGaFC2ClfkNOt2MlVnQZCmnYBCwOktPGkExiu2yZMifcYGxQcpH

KVFG59KeF2cM2d4xYM8HJqkSGGW306LFVSyeRwG+wbttgLpD5bM/T2b3fF/J35ra

CSMLZearRTq8aygPl+XM7MM2eR946aw6jmOsgNBErbvvIdQj6LudAZj+8imcXV2K

iQEiBBABAgAMBQJOmdnRBQMAEnUAAAoJEJcQuJvKV618AvIIAIEF1ZJ+Ry7WOdKF

5oeQ/ynaYUigzN92fW/9zB8yuQlngkFJGidYMbci1tR1siziIVJFusR3ZonqAPGK

/SUta9Y6KWLhmc7c5UnEHklq/NfdMZ2WVSIykXlctqw0sbb+z1ecEd4G8u9j5ill

MO1B36rQayYAPoeXLX8dY4VyFLVGaQ00rWQBYFZrpw16ATWbWGJP332NSfCk4zZq

6kXEW07q0st3YBgAAGdNQyEeZCa4d4pBRSX6189Kjg6GDnIcaiOF6HO6PLr9fRlL

r5ObCgU+G9gEhfiVwDEV9E+7/Bq2pYZ9whhkBqWQzdpXTNTM24uaEhE01EPO5zeC

O214q6mJASIEEAECAAwFAk6rpgEFAwASdQAACgkQlxC4m8pXrXzAhwf/f9O99z16

3Y5FZVIxexyqXQ/Mct9uKHuXEVnRFYbA49dQLD4S73N+zN7gn9jFeQcBo4w8qVUV

94U/ta/VbLkdtNREyplPM4XY8YE5Wfd9bfyg3q1PbEiVjk995sBF+2+To99YYKst

gXPqjlH0jUfEyDmexOj+hsp8Rc63kvkIx36VBa4ONRYFefGAhKDMigL2YAhc1UkG

tkGTuLmlCGwIV6lviDZD3RJf5375VFnaHv7eXfwQxCwE+BxG3CURrjfxjaxMTmMP

yAG2rhDp5oTUEvqDYNbko5UxYOmrSjvF4FzXwqerElXJUkUzSh0pp7RxHB/1lCxD

s7D1F1hlgFQuNIkBIgQQAQIADAUCTrzZHAUDABJ1AAAKCRCXELibyletfMUpB/4s

07dREULIBnA1D6qr3fHsQJNZqbAuyDlvgGGLWzoyEDs+1JMFFlaa+EeLIo1386GU

2DammDC23p3IB79uQhJeD2Z1TcVg4cA64SfF/CHca5coeRSrdAiudzU/cgLGtXIP

/OaFamXgdMxAhloLFbSHPCZkyb00phVa8+xeIVDrK1HByZsNIXy/SSK8U26S2PVZ

2o14fWvKbJ1Aga8N6DuWY/D8P2mi3RAbiuZgfzkmKL5idH/wSKfnFKdTgJzssdCc

1jZEGVk5rFYcWOrJARHeP/tsnb/UxKBEsNtO7e3N2e/rLVnEykVIO066hz7xZK/V

NBSpx3k3qj4XPK41IHy2iQEiBBABAgAMBQJOzqO8BQMAEnUAAAoJEJcQuJvKV618

2twH/0IzjXLxN45nvIfEjC75a+i9ZSLlqR8lsHL4GpEScFKI0a0lT4IVAIY2RKG+

MAs2eHm0UfKuwGs5jluRZ9RqKrc61sY0XQV9/7znY9Db16ghX04JjknOKs/fPi87

rvKkB/QxJWS8qbb/erRmW+cPNjbRxTFPS5JIwFWHA16ieFEpvdAgKV6nfvJVTq1r

jPDcnIA9CJN2SmUFx9Qx3SRc6ITbam1hjFnY6sCh6AUhxLI2f1mq1xH9PqEy42Um

68prRqTyJ7Iox1g/UDDkeeUcAg7T1viTz7uXpS3Wrq4zzo4yOpaJfLDR3pI5g2Zk

SNGTMo6aySE4OABt8i1Pc1Pm6AmJASIEEAECAAwFAk7yPFYFAwASdQAACgkQlxC4

m8pXrXzXiAf9FrXe0lgcPM+tYOWMLhv5gXJi2VUBaLxpyRXm/kJcmxInKq1GCd3y

D4/FLHNu3ZcCz/uklPAbZXWI0O6ewq0LWsRtklmJjWiedH+hGyaTv95VklojRIBd

8nBaJ6M98rljMBHTFwWvjQFVf4FLRJQZqHlvjcCkq2Dd9BWJpGXvr/gpKkmMJYNK

/ftfZRcChb35NI19WRpOhj9u808OPcqKVvZBcPwFGV5cEBzmAC94J7JcD8+S8Ik8

iUJMQGGL3QcmZOBozovh86hj7KTSEBHlLXl832z89H1hLeuLbnXoGLv3zeUFSxkv

1h35LhZLqIMDQRXLuUzxGHMBpLhPyGWRJ4kBIgQQAQIADAUCTwQJFwUDABJ1AAAK

CRCXELibyletfABvB/9Cy69cjOqLGywITs3Cpg//40jmdhSAVxilJivP6J5bubFH

DJlVTx541Dv5h4hTG2BQuueQ4q1VCpSGW+rHcdhPyvmZGRz1rxdQQGh1Dv0Bod2c

3PJVSYPSrRSwCZJkJHOtVRBdjK4mkZb5aFTza+Tor9kxzj4FcXVd4KAS+hHQHYHc

Ar8tt2eOLzqdEFTULeGiSoNn+PVzvzdfhndphK+8F2jfQ2UKuc01O7k0Yn9xZVx0

OG6fE1gStzLv7C5amWLRd8+xh+MN0G8MgNglpBoExsEMMlPBYSUHa6lxpdMNMuib

rIyVncE9X8QOhImt8K0sNn/EdbuldJNGYbDLt7O4iQEiBBABAgAMBQJPFdTcBQMA

EnUAAAoJEJcQuJvKV6184owH+wZ/uLpezXnSxigeH1sig72QEXMrNd5DVHCJdig3

bo+K5YmmN710/m5z+63XKUEWpd6/knajObgckThzWftNeK1SSFQGPmoYZP9EZnSU

7L+/dSUpExbj842G5LYagrCyMGtlxRywWEmbi72TKS/JOK0jLiOdvVy+PHrZSu0D

TVQ7cJh1BmPsbz7zzxjmcI5l+7B7K7RHZHq45nDLoIabwDacj7BXvBK0Ajqz4QyJ

GQUjXC7q+88I+ptPvOXlE5nI/NbiCJOMI6d/bWN1KwYrC80fZuFaznfQFcPyUaDw

yRaun+K3kEji2wXecq+yMmLUEp01TKsUeOL50HD6hHH07W+JASIEEAECAAwFAk85

bQsFAwASdQAACgkQlxC4m8pXrXwKPQgAlkbUsTr7nkq+haOk0jKpaHWEbRMEGMrB

I3F7E+RDO6V/8y4Jtn04EYDc8GgZMBah+mOgeINq3y8jRMYV5jVtZXv2MWYFUcjM

kVBKeqhi/pGEjmUdmdt3DlPv3Z+fMTMRmAocI981iY/go8PVPg/+nrR6cFK2xxnO

R8TacikJBFeSfkkORg1tDzjjYv1B5ZIEkpplepl5ahJBBq7cpYhTdY6Yk0Sz0J8w

EdffLSaNxrRuWLrRhWzZU7p9bFzfb/7OHc21dJnB7wKv5VvtgE+jiQw9tOKaf5hc

SgRYuF6heu+B25gc5Uu88lo409mZ7oxQ6hDCn7JHvzh0rhmSN+Kid4kBIgQQAQIA

DAUCT0qQrQUDABJ1AAAKCRCXELibyletfC9UB/4o2ggJYM0CLxEpP0GU8UKOh3+/

zm1DN7Qe4kY2iCtF1plKHQaTgt5FlgRCFaiXcVv7WzGz/FnmxonR1leLl+kfRlwy

PPnoI/AWPCy/NO4Cl5KnjsSmsdDUpObwZ4KYsdilZR7ViJu2swdAIgnXBUwrlRJR

7CK4TAKrTeonRgVSrVx8Vt//8/cYj73CLq8oY/KK0iHiQrSwo44uyhdiFIAssjyX

n6/2E+w0zgvPexNSNNROHQ8pjbq+NTY6GwKIGsaej3UTRwQ7psvKXz8y7xdzmOAr

/khGvxB5gjkx02pimjeia8v66aH6rbnojJMAovNUS4EHdHnulv4rovC8Kf9iiQEi

BBABAgAMBQJPVdsaBQMAEnUAAAoJEJcQuJvKV618vVEIALFXPBzcAO1SnQarBLzy

YMVZZumPvSXKnUHAO+6kjApXPJ+qFRdUaSNshZxVKY9Zryblu4ol/fLUTt0CliSD

IxD6L4GXEm4VYYCl4lPO3bVsJnGITLFwQGHM27EmjVoTiD8Ch7kPq2EXr3dMRgzj

pdz+6aHGSUfOdLTPXufDvW83bEWGaRVuTJKw+wIrcuRqQ+ucWJgJGwcE4zeHjZad

Jx1XUm1X+BbI73uiQussyjhhQVVNU7QEdrjyuscaZ/H38wjUwNbylxDPB4I8quC1

knQ0wSHr7gKpM+E9nhiS14poRqU18u78/sJ2MUPXnQA6533IC238/LP8JgqB+BiQ

BTSJASIEEAECAAwFAk9ng3cFAwASdQAACgkQlxC4m8pXrXxQRAf/UZlkkpFJj1om

9hIRz7gS+l7YvTaKSzpo+TBcx3C7aqKJpir6TlMK9cb9HGTHo2Xp1N3FtQL72NvO

6CcJpBURbvSyb4i0hrm/YcbUC4Y3eajWhkRS3iVfGNFbc/rHthViz0r6Y5lhXX16

aVkDv5CIFWaF3BiUK0FnHrZiy4FPacUXCwEjv3uf8MpxV5oEmo8Vs1h4TL3obyUz

qrImFrEMYE/12lkE8iR5KWCaF8eFyl56HL3PPl90JMQBXzhwsFoWCPuwjfM5w6sW

Ll//zynwxtlJ9CRz9c2vK6aJ8DRu3OfBKN1iiEcNEynksDnNXErn5xXKz3p5pYdq

e9BLzUQCDYkBIgQQAQIADAUCT3inRgUDABJ1AAAKCRCXELibyletfGMKCADJ97qk

geBntQ+tZtKSFyXznAugYQmbzJld8U6eGSQnQkM40Vd62UZLdA8MjlWKS8y4A4L2

0cI14zs5tKG9Q72BxQOw5xkxlLASw1/8WeYEbw7ZA+sPG//q9v3kIkru3sv64mMA

enZtxsykexRGyCumxLjzlAcL1drWJGUYE2Kl6uzQS7jb+3PNBloQvz6nb3YRZ+Cg

Ly9D41SIK+fpnV8r4iqhu7r4LmAQ7Q1DF9aoGaYvn2+xLGyWHxJAUet4xkMNOLp6

k9RF1nbNe4I/sqeCB25CZhCTEvHdjSGTD2yJR5jfoWkwO9w8DZG1Q9WrWqki4hSB

l0cmcvO34pC1SJYziQEiBBABAgAMBQJPinQFBQMAEnUAAAoJEJcQuJvKV618CFEI

AJp5BbcV7+JBMRSvkoUcAWDoJSP2ug9zGw5FB8J90PDefKWCKs5Tjayf2TvM5ntq

5DE9SGaXbloIwa74FoZlgqlhMZ4AtY9Br+oyPJ5S844wpAmWMFc6NnEPFaHQkQ+b

dJYpRVNd9lzagJP261P3S+S9T2UeHVdOJBgWIq9Mbs4lnZzWsnZfQ4Lsz0aPqe48

tkU8hw+nflby994qIwNOlk/u+I/lJbNz5zDY91oscXTRl2jV1qBgKYwwCXxyB3j9

fyVpRl+7QnqbTWcCICVFL+uuYpP0HjdoKNqhzEguAUQQLOB9msPTXfa2hG+32ZYg

5pzI5V7GCHq0KO6u5Ctj3TGJASIEEAECAAwFAk+cQEEFAwASdQAACgkQlxC4m8pX

rXzi7AgAx8wJzNdD7UlgdKmrAK//YqH7arSssb33Xf45sVHDpUVA454DXeBrZpi+

zEuo03o5BhAuf38cwfbkV6jN1mC2N0FZfpy4v7RxHKLYr7tr6r+DRn1L1giX5ybx

CgY0fLAxkwscWUKGKABWxkz9b/beEXaO2rMt+7DBUdpAOP5FNRQ8WLRWBcMGQiaT

S4YcNDAiNkrSP8CMLQP+04hQjahxwCgBnksylciqz3Y5/MreybNnTOrdjVDsF0Oe

t0uLOiWXUZV1FfaGIdb/oBQLg+e1B74p5+q3aF8YI97qAZpPa1qiQzWIDX8LX9QX

EFyZ3mvqzGrxkFoocXleNPgWT8fRuokBIgQQAQIADAUCT64N/QUDABJ1AAAKCRCX

ELibyletfDOGCACKfcjQlSxrWlEUrYYZpoBP7DE+YdlIGumt5l6vBmxmt/5OEhqr

+dWwuoiyC5tm9CvJbuZup8anWfFzTTJmPRPsmE4z7Ek+3CNMVM2wIynsLOt1pRFK

4/5RNjRLbwI6EtoCQfpLcZJ//SB56sK4DoFKH28Ok4cplESPnoMqA3QafdSEA/FL

qvZV/iPgtTz7vjQkMgrXAIUM4fvKe3iXkAExGXtmgdXHVFoKmHrxJ2DTSvM7/19z

jGJeu2MhIKHyqEmCk6hLjxyCE5pAH59KlbAQOP1bS28xlRskBApm2wN+LOZWzC62

HhEReQ50inCGuuubK0PqUQnyYc+lUFxrFpcliQEiBBABAgAMBQJPv9lVBQMAEnUA

AAoJEJcQuJvKV618AzgH/iRFFCi4qjvoqji1fi7yNPZVOMMO2H13Ks+AfcjRtHuV

aa30u50ND7TH+XQe6yerTapLh3aAm/sNP99aTxIuwRSlyKEoDs93+XVSgRqPBgbF

/vxv0ykok3p6L9DxFO/w5cL8JrBhMZoJrEkIBFkwN8tWlcXPRFQvcdBYv3M3DTZU

qY+UHnOxHvSzsl+LJ0S9Xcd9C5bvYfabmYJvG5eRS3pj1L/y3a6yw6hvY+JtnQAk

t05TdeHMIgQH/zb8V9wxDzmE0un8LyoC2Jx5TpikQsJSejwK6b3coxVBlngku6+C

qDAimObZLw6H9xYYIK0FoJs7j5bQZEwUO7OLBgjcMOqJASIEEAECAAwFAk/Rpc8F

AwASdQAACgkQlxC4m8pXrXw49Qf/TdNbun2htQ+cRWarszOx8BLEiW/x6PVyUQpZ

nV/0qvhKzlJUjM9hQPcA0AsOjhqtCN6Cy8KXbK/TvPm9D/Nk6HWwD1PomzrJVFk2

ywGFIuTR+lluKSp7mzm5ym0wJs5cPq731Im31RUQU8ndjLrq9YOf5FVL8NqmcOAU

4E8d68BbmVCQC5MMr0901FKwKznShfpy7VYN25/BASj8dhnynBYQErqToOJB6Cnd

JhdTlbfR4SirqAYZZg3XeqGhByytEHE1x7FMWWFYhdNtsnAVhYBbWqAzBs8lF9Jd

Mhaf0VQU/4z10gVrRtXLR/ixrCi+P4cM/fOQkqd6pwqWkaXt6okBIgQQAQIADAUC

T+NxIAUDABJ1AAAKCRCXELibyletfFBBCAC6+0TUJDcNaqOxOG1KViY6KYg9NCL8

pwNK+RKNK/N1V+WGJQH7qDMwRoOn3yogrHax4xIeOWiILrvHK0O6drS1DjsymIhR

Sm2XbE/8pYmEbuJ9vHh3b/FTChmSAO7dDjSKdWD3dvaY8lSsuDDqPdTX8FzOfrXC

M22C/YPg7oUG2A5svE1b+yismP4KmVNWAepEuPZcnEMPFgop3haHg9X2+mj/btDB

Yr6p9kAgIY17nigtNTNjtI0dMLu43aIzedCYHqOlNHiB049jkJs54fMGBjF9qPtc

m0k44xyKd1/JXWMdNUmtwKsChAXJS3YOciMgIx6tqYUTndrP4I6q1rfriQEiBBAB

AgAMBQJP9T1VBQMAEnUAAAoJEJcQuJvKV618J9wIAI1lId9SMbEHF6PKXRe154lE

pap5imMU/lGTj+9ZcXmlf8o2PoMMmb3/E1k+EZUaeSBoOmjS8C2gwd5XFwRrlwAD

RlK/pG5XsL4h5wmN2fj1ororrJXvqH427PLRQK9yzdwG4+9HTBOxjoS8qZT9plyK

AJZzAydAMqyseRHgNo0vMwlgrs4ojo+GcFGQHrF3IaUjvVfUPOmIj7afopFdIZmI

GaSF0TXBzqcZ1chFv/eTBcIuIKRvlaDee5FgV7+nLH2nKOARCLvV/+8uDi2zbr83

Ip5x2tD3XuUZ0ZWxD0AQWcrLdmGb4lkxbGxvCtsaJHaLXWQ2m760RjIUcwVMEBKJ

ASIEEAECAAwFAlAGYWsFAwASdQAACgkQlxC4m8pXrXwyVAgAvuvEl6yuGkniWOlv

uHEusUv/+2GCBg6qV+IEpVtbTCCgiFjYR5GasSp1gpZ5r4BocOlbGdjdJGHTpyK8

xD1i+6qZWUYhNRg2POXUVzcNEl2hhouwPLOifcmTwAKU76TEv3L5STviL3hWgUR2

yEUZ3Ut0IGVV6uPER9jpR3qd6O3PeuFkwf+NaGTye4jioLAy3aYwtZCUXzvYmNLP

90K4y+5yauZteLmNeq26miKC/NQu4snNFClPbGRjHD1ex9KDiAMttOgN4WEq7srT

rYgtT531WY4deHpNgoPlHPuAfC0H+S6YWuMbgfcb6dV+Rrd8Ij6zM3B/PcjmsYUf

OPdPtIkBIgQQAQIADAUCUBgtfQUDABJ1AAAKCRCXELibyletfAm3CACQlw21Lfeg

d8RmIITsfnFG/sfM3MvZcjVfEAtsY3fTK9NiyU0B3yX0PU3ei37qEW+50BzqiStf

5VhNvLfbZR+yPou7o2MAP31mq3Uc6grpTV64BRIkCmRWg40WMjNI1hv7AN/0atgj

ATYQXgnEw7mfFb0XZtMTD6cmrz/A9nTPVgZDxzopOMgCCC1ZK4Vpq9FKdCYUaHpX

3sqnDf+gpVIHkTCMgWLYQOeX5Nl+fgnq6JppaQ3ySZRUDr+uFUs0uvDRvI/cn+ur

ri92wdDnczjFumKvz/cLJAg5TG2Jv1Jx3wecALsVqQ3gL7f7vr1OMaqhI5FEBqdN

29L9cZe/ZmkriQEiBBIBCgAMBQJVoNxyBYMHhh+AAAoJEEoz7NUmyPxLD1EH/2eh

7a4+8A1lPLy2L9xcNt2bifLfFP2pEjcG6ulBoMKpHvuTCgtX6ZPdHpM7uUOje/F1

CCN0IPB533U1NIoWIKndwNUJjughtoRM+caMUdYyc4kQm29Se6hMPDfyswXE5Bwe

PmoOm4xWPVOH/cVN04zyLuxdlQZNQF/nJg6PMsz4w5z+K6NGGm24NEPcc72iv+6R

Uc/ry/7v5cVu4hO5+r104mmNV5yLecQF13cHy2JlngIHXPSlxTZbeJX7qqxE7TQh

5nviSPgdk89oB5jFSx4g1efXiwtLlP7lbDlxHduomyQuH9yqmPZMbkJt9uZDc8Zz

MYsDDwlc7BIe5bGKfjqJAhwEEAECAAYFAlSanFIACgkQdzHqU52lcqLdvg//cAEP

qdN5VTKWEoDFjDS4I6t8+0KzdDWDacVFwKJ8RAo1M2SklDxnIvnzysZd2VHp5Pq7

i4LYCZo5lDkertQ6LwaQxc4X6myKY4LTA652ObFqsSfgh9kW+aJBBAyeahPQ8CDD

+Yl23+MY5wTsj4qt7KffNzy78vLbYnVnvRQ3/CboVix0SRzg0I3Oi7n3B0lihvXy

5goy9ikjzZevejMEfjfeRCgoryy9j5RvHH9PF3fJVtUtHCS4f+kxLmbQJ1XqNDVD

hlFzjz8oUzz/8YXy3im5MY7Zuq4P4wWiI7rkIFMjTYSpz/evxkVlkR74qOngT2pY

VHLyJkqwh56i0aXcjMZiuu2cymUt2LB9IsaMyWBNJjXr2doRGMAfjuR5ZaittmML

yZwix9mWVk7tkwlIxmT/IW6Np0qMhDZcWYqPRpf7+MqY3ZYMK4552b8aDMjhXrnO

OwLsz+UI4bZa1r9dguIWIt2C2b5C1RQ9AsQBPwg7h5P+HhRuFAuDKK+vgV8FRuzR

JeKkFqwB4y0Nv7BzKbFKmP+V+/krRv+/Dyz9Bz/jyAQgw02u1tPupH9BGhlRyluN

yCJFTSNj7G+OLU0/l4XNph5OOC7sy+AMZcsL/gsT/TXCizRcCuApNTPDaenACpbv

g8OoIzmNWhh4LXbAUHCKmY//hEw9PvTZA1xKHgyJAhwEEgECAAYFAlJYsKQACgkQ

oirk60MpxUV2XQ//b2/uvThkkbeOegusDC4AZfjnL/V3mgk4iYy4AC9hum0R9oNl

XDR51P1TEw9mC1btHj+7m7Iq1a5ke5wIC7ENZiilr0yPqeWgL5+LC98dz/L85hqA

wIoGeOfMhrlaVbAZEj4yQTAJDA35vZHVsQmp87il0m+fZX04OBLXBzw86EoAAZ7Q

EoH4qFcT9k1T363tvNnIm3mEvkQ5WjE1R9uchJa1g7hdlNQlVkjFmPZrJK9fl4z5

6Dto89Po4Sge48jDH0pias4HATYHsxW819nz5jZzGcxLnFRRR5iITVZi9qzsHP7N

bUh3qxuWCHS9xziXpOcSZY848xXw63Y5jDJfpzupzu/KHj6CzXYJUEEqp9MluoGb

/BCCEPzdZ0ovyxFutM/BRcc6DvE6sTDF/UES21ROqfuwtJ6qJYWX+lBIgyCJvj4o

RdbzxUleePuzqCzmwrIXtoOKW0Rlj4SCeF9yCwUMBTGW5/nCLmN4dwf1KW2RP2Eg

4ERbuUy7QnwRP5UCl+0ISZJyYUISfg8fmPIdQsetUK9Cj+Q5jpB2GXwELXWnIK6h

K/6jXp+EGEXSqdIE53vAFe7LwfHiP/D5M71D2h62sdIOmUm3lm7xMOnM5tKlBiV+

4jJSUmriCT62zo710+6iLGqmUUYlEll6Ppvo8yuanXkYRCFJpSSP7VP0bBqIZgQT

EQIAJgUCTnc9dgIbIwUJEPPzpwYLCQgHAwIEFQIIAwQWAgMBAh4BAheAAAoJEIxx

jTtQcuH1Ut4AoIKjhdf70899d+7JFq3LD7zeeyI0AJ9Z+YyE1HZSnzYi73brScil

bIV6sbQ7TXlTUUwgUGFja2FnZSBzaWduaW5nIGtleSAod3d3Lm15c3FsLmNvbSkg

PGJ1aWxkQG15c3FsLmNvbT6IbwQwEQIALwUCTnc9rSgdIGJ1aWxkQG15c3FsLmNv

bSB3aWxsIHN0b3Agd29ya2luZyBzb29uAAoJEIxxjTtQcuH1tT0An3EMrSjEkUv2

9OX05JkLiVfQr0DPAJwKtL1ycnLPv15pGMvSzav8JyWN3IhlBBMRAgAdBQJHrJS0

BQkNMFioBQsHCgMEAxUDAgMWAgECF4AAEgkQjHGNO1By4fUHZUdQRwABAa6SAJ9/

PgZQSPNeQ6LvVVzCALEBJOBt7QCffgs+vWP18JutdZc7XiawgAN9vmmITAQTEQIA

DAUCPj6j0QWDCWYAuwAKCRBJUOEqsnKR8iThAJ9ZsR4o37dNGyl77nEqP6RAlJqa

YgCeNTPTEVY+VXHR/yjfyo0bVurRxT2ITAQTEQIADAUCPkKCAwWDCWIiiQAKCRC2

9c1NxrokP5aRAKCIaaegaMyiPKenmmm8xeTJSR+fKQCgrv0TqHyvCRINmi6LPucx

GKwfy7KIRgQQEQIABgUCP6zjrwAKCRCvxSNIeIN0D/aWAKDbUiEgwwAFNh2n8gGJ

Sw/8lAuISgCdHMzLAS26NDP8T2iejsfUOR5sNriIRgQQEQIABgUCP7RDdwAKCRCF

lq+rMHNOZsbDAJ0WoPV+tWILtZG3wYqg5LuHM03faQCeKuVvCmdPtro06xDzeeTX

VrZ14+GIRgQQEQIABgUCQ1uz6gAKCRCL2C5vMLlLXH90AJ0QsqhdAqTAk3SBnO2w

zuSOwiDIUwCdFExsdDtXf1cL3Q4ilo+OTdrTW2CIRgQTEQIABgUCRPEzJgAKCRD2

ScT0YJNTDApxAKCJtqT9LCHFYfWKNGGBgKjka0zi9wCcCG3MvnvBzDUqDVebudUZ

61Sont+ITAQQEQIADAUCQYHLAQWDBiLZiwAKCRAYWdAfZ3uh7EKNAJwPywk0Nz+Z

Lybw4YNQ7H1UxZycaQCePVhY4P5CHGjeYj9SX2gQCE2SNx+ITAQQEQIADAUCQYHL

NAWDBiLZWAAKCRCBwvfr4hO2kiIjAJ0VU1VQHzF7yYVeg+bh31nng9OOkwCeJI8D

9mx8neg4wspqvgXRA8+t2saITAQQEQIADAUCQYHLYgWDBiLZKgAKCRBrcOzZXcP0

cwmqAJsFjOvkY9c5eA/zyMrOZ1uPB6pd4QCdGyzgbYb/eoPu6FMvVI9PVIeNZReI

TAQQEQIADAUCQdCTJAWDBdQRaAAKCRB9JcoKwSmnwmJVAKCG9a+Q+qjCzDzDtZKx

5NzDW1+W+QCeL68seX8OoiXLQuRlifmPMrV2m9+ITAQQEQIADAUCQitbugWDBXlI

0gAKCRDmG6SJFeu5q/MTAKCTMvlCQtLKlzD0sYdwVLHXJrRUvgCffmdeS6aDpwIn

U0/yvYjg1xlYiuqITAQSEQIADAUCQCpZOgWDB3pLUgAKCRA8oR80lPr4YSZcAJwP

4DncDk4YzvDvnRbXW6SriJn1yQCdEy+d0CqfdhM7HGUs+PZQ9mJKBKqITAQSEQIA

DAUCQD36ugWDB2ap0gAKCRDy11xj45xlnLLfAKC0NzCVqrbTDRw25cUss14RRoUV

PACeLpEc3zSahJUB0NNGTNlpwlTczlCITAQSEQIADAUCQQ4KhAWDBpaaCAAKCRA5

yiv0PWqKX/zdAJ4hNn3AijtcAyMLrLhlZQvib551mwCgw6FEhGLjZ+as0W681luc

wZ6PzW+ITAQSEQIADAUCQoClNAWDBSP/WAAKCRAEDcCFfIOfqOMkAJwPUDhS1eTz

gnXclDKgf353LbjvXgCeLCWyyj/2d0gIk6SqzaPl2UcWrqiITAQTEQIADAUCPk1N

hAWDCVdXCAAKCRAtu3a/rdTJMwUMAKCVPkbk1Up/kyPrlsVKU/Nv3bOTZACfW5za

HX38jDCuxsjIr/084n4kw/uITAQTEQIADAUCQdeAdgWDBc0kFgAKCRBm79vIzYL9

Pj+8AJ9d7rvGJIcHzTCSYVnaStv6jP+AEACeNHa5yltqieRBCCcLcacGqYK81omI

TAQTEQIADAUCQhiBDgWDBYwjfgAKCRB2wQMcojFuoaDuAJ9CLYdysef7IsW42UfW

hI6HjxkzSgCfeEpXS4hEmmGicdpRiJQ/W21aB0GIZQQTEQIAHQULBwoDBAMVAwID

FgIBAheABQJLcC/KBQkQ8/OnABIHZUdQRwABAQkQjHGNO1By4fWw2wCeJilgEarL

8eEyfDdYTyRdqE45HkoAnjFSZY8Zg/iXeErHI0r04BRukNVgiHsEMBECADsFAkJ3

NfU0HQBPb3BzLi4uIHNob3VsZCBoYXZlIGJlZW4gbG9jYWwhIEknbSAqc28qIHN0

dXBpZC4uLgAKCRA5yiv0PWqKX+9HAJ0WjTx/rqgouK4QCrOV/2IOU+jMQQCfYSC8

JgsIIeN8aiyuStTdYrk0VWCIjwQwEQIATwUCRW8Av0gdAFNob3VsZCBoYXZlIGJl

ZW4gYSBsb2NhbCBzaWduYXR1cmUsIG9yIHNvbWV0aGluZyAtIFdURiB3YXMgSSB0

aGlua2luZz8ACgkQOcor9D1qil+g+wCfcFWoo5qUl4XTE9K8tH3Q+xGWeYYAnjii

KxjtOXc0ls+BlqXxbfZ9uqBsiQIiBBABAgAMBQJBgcuFBYMGItkHAAoJEKrj5s5m

oURoqC8QAIISudocbJRhrTAROOPoMsReyp46Jdp3iL1oFDGcPfkZSBwWh8L+cJjh

dycIwwSeZ1D2h9S5Tc4EnoE0khsS6wBpuAuih5s//coRqIIiLKEdhTmNqulkCH5m

imCzc5zXWZDW0hpLr2InGsZMuh2QCwAkB4RTBM+r18cUXMLV4YHKyjIVaDhsiPP/

MKUj6rJNsUDmDq1GiJdOjySjtCFjYADlQYSD7zcd1vpqQLThnZBESvEoCqumEfOP

xemNU6xAB0CL+pUpB40pE6Un6Krr5h6yZxYZ/N5vzt0Y3B5UUMkgYDSpjbulNvaU

TFiOxEU3gJvXc1+h0BsxM7FwBZnuMA8LEA+UdQb76YcyuFBcROhmcEUTiducLu84

E2BZ2NSBdymRQKSinhvXsEWlH6Txm1gtJLynYsvPi4B4JxKbb+awnFPusL8W+gfz

jbygeKdyqzYgKj3M79R3geaY7Q75Kxl1UogiOKcbI5VZvg47OQCWeeERnejqEAdx

EQiwGA/ARhVOP/1l0LQA7jg2P1xTtrBqqC2ufDB+v+jhXaCXxstKSW1lTbv/b0d6

454UaOUV7RisN39pE2zFvJvY7bwfiwbUJVmYLm4rWJAEOJLIDtDRtt2h8JahDObm

3CWkpadjw57S5v1c/mn+xV9yTgVx5YUfC/788L1HNKXfeVDq8zbAiQIiBBMBAgAM

BQJCnwocBYMFBZpwAAoJENjCCglaJFfPIT4P/25zvPp8ixqV85igs3rRqMBtBsj+

5EoEW6DJnlGhoi26yf1nasC2frVasWG7i4JIm0U3WfLZERGDjR/nqlOCEqsP5gS3

43N7r4UpDkBsYh0WxH/ZtST5llFK3zd7XgtxvqKL98l/OSgijH2W2SJ9DGpjtO+T

iegq7igtJzw7Vax9z/LQH2xhRQKZR9yernwMSYaJ72i9SyWbK3k0+e95fGnlR5pF

zlGq320rYHgD7v9yoQ2t1klsAxK6e3b7Z+RiJG6cAU8o8F0kGxjWzF4v8D1op7S+

IoRdB0Bap01ko0KLyt3+g4/33/2UxsW50BtfqcvYNJvU4bZns1YSqAgDOOanBhg8

Ip5XPlDxH6J/3997n5JNj/nk5ojfd8nYfe/5TjflWNiput6tZ7frEki1wl6pTNbv

V9C1eLUJMSXfDZyHtUXmiP9DKNpsucCUeBKWRKLqnsHLkLYydsIeUJ8+ciKc+EWh

FxEY+Ml72cXAaz5BuW9L8KHNzZZfez/ZJabiARQpFfjOwAnmhzJ9r++TEKRLEr96

taUI9/8nVPvT6LnBpcM38Td6dJ639YvuH3ilAqmPPw50YvglIEe4BUYD5r52Seqc

8XQowouGOuBX4vs7zgWFuYA/s9ebfGaIw+uJd/56Xl9ll6q5CghqB/yt1EceFEnF

CAjQc2SeRo6qzx22iEYEEBECAAYFAkSAbycACgkQCywYeUxD5vWDcACfQsVk/XGi

ITFyFVQ3IR/3Wt7zqBMAoNhso/cX8VUfs2BzxPvvGS3y+5Q9iEYEEBECAAYFAkUw

ntcACgkQOI4l6LNBlYkyFgCbBcw5gIii0RTDJsdNiuJDcu/NPqEAniSq9iTaLjgF

HZbaizUU8arsVCB5iEYEEBECAAYFAkWho2sACgkQu9u2hBuwKr6bjwCfa7ZK6O+X

mT08Sysg4DEoZnK4L9UAoLWgHuYg35wbZYx+ZUTh98diGU/miF0EExECAB0FAj4+

owwFCQlmAYAFCwcKAwQDFQMCAxYCAQIXgAAKCRCMcY07UHLh9XGOAJ4pVME15/DG

rUDohtGv2z8a7yv4AgCeKIp0jWUWE525QocBWms7ezxd6syIXQQTEQIAHQUCR6yU

zwUJDTBYqAULBwoDBAMVAwIDFgIBAheAAAoJEIxxjTtQcuH1dCoAoLC6RtsD9K3N

7NOxcp3PYOzH2oqzAKCFHn0jSqxk7E8by3sh+Ay8yVv0BYhdBBMRAgAdBQsHCgME

AxUDAgMWAgECF4AFAkequSEFCQ0ufRUACgkQjHGNO1By4fUdtwCfRNcueXikBMy7

tE2BbfwEyTLBTFAAnifQGbkmcARVS7nqauGhe1ED/vdgiF0EExECAB0FCwcKAwQD

FQMCAxYCAQIXgAUCS3AuZQUJEPPyWQAKCRCMcY07UHLh9aA+AKCHDkOBKBrGb8tO

g9BIub3LFhMvHQCeIOOot1hHHUlsTIXAUrD8+ubIeZaJARwEEgECAAYFAkvCIgMA

CgkQ3PTrHsNvDi8eQgf/dSx0R9Klozz8iK79w00NOsdoJY0Na0NTFmTbqHg30XJo

G62cXYgc3+TJnd+pYhYi5gyBixF/L8k/kPVPzX9W0YfwChZDsfTw0iDVmGxOswiN

jzSo0lhWq86/nEL30Khl9AhCC1XFNRw8WZYq9Z1qUXHHJ2rDARaedvpKHOjzRY0N

dx6R2zNyHDx2mlfCQ9wDchWEuJdAv0uHrQ0HV9+xq7lW/Q3L/V5AuU0tiowyAbBL

PPYrB6x9vt2ZcXS7BOy8SfQ1i8W2QDQ/Toork4YwBiv6WCW/ociy7paAoPOWV/Nf

2S6hDispeecbk7wqpbUj5klDmwrlgB/jmoAXWEnbsYkBIgQQAQIADAUCSSpooAUD

ABJ1AAAKCRCXELibyletfFOMCACpP+OVZ7lH/cNY+373c4FnSI0/S5PXS0ABgdd4

BFWRFWKrWBeXBGc8sZfHOzVEwkzV96iyHbpddeAOAkEA4OVPW1MMFCmlHxi2s9/N

JrSrTPVfQOH5fR9hn7Hbpq/ETw0IoX1FKo7vndMnHZnFEnI+PDXLcdMYQgljYzhT

xER4vYY0UKu8ekSshUy4zOX7XSJxwqPUvps8qs/TvojIF+vDJvgFYHVkgvS+shp8

Oh/exg9vKETBlgU87Jgsqn/SN2LrR/Jhl0aLd0G0iQ+/wHmVYdQUMFaCZwk/BKNa

XPzmGZEUZ3RNbYa19Mo7hcE3js76nh5YMxFvxbTggVu4kdFkiQEiBBABAgAMBQJK

M06IBQMAEnUAAAoJEJcQuJvKV618F4gH/innejIHffGMk8jYix4ZZT7pW6ApyoI+

N9Iy85H4L+8rVQrtcTHyq0VkcN3wPSwtfZszUF/0qP6P8sLJNJ1BtrHxLORYjJPm

gveeyHPzA2oJl6imqWUTiW822fyjY/azwhvZFzxmvbFJ+r5N/Z57+Ia4t9LTSqTN

HzMUYaXKDaAqzZeK7P0E6XUaaeygbjWjBLQ1O0ezozAy+Kk/gXApmDCGFuHSFe7Z

mgtFcbXLM2XFQpMUooETD2R8MUsd+xnQsff/k6pQOLxi+jUEsWSr/iqmvlk6gZ4D

pemBjuhcXYlxJYjUaX9Zmn5s+ofF4GFxRqXoY7l9Z+tCM9AX37lm6S+JASIEEAEC

AAwFAkpEcgoFAwASdQAACgkQlxC4m8pXrXz2mgf/RQkpmMM+5r8znx2TpRAGHi5w

ktvdFxlvPaOBWE28NDwTrpcoMqo9kzAiuvEQjVNihbP21wR3kvnQ84rTAH0mlC2I

uyybggpqwzOUl+Wi0o+vk8ZA0A0dStWRN8uqneCsd1XnqDe1rvqC4/9yY223tLmA

kPvz54ka2vX9GdJ3kxMWewhrVQSLCktQpygU0dujGTDqJtnk0WcBhVF9T87lv3W2

eGdPielzHU5trXezmGFj21d56G5ZFK8co7RrTt4qdznt80glh1BTGmhLlzjMPLTe

dcMusm3D1QB9ITogcG94ghSf9tEKmmRJ6OnnWM5Kn9KcL63E5oj2/lY9H54wSYkB

IgQQAQIADAUCSlY+RwUDABJ1AAAKCRCXELibyletfOOQB/0dyJBiBjgf+8d3yNID

pDktLhZYw8crIjPBVdOgX12xaUYBTGcQITRVHSggzffDA5BQXeUuWhpL4QB0uz1c

EPPwSMiWiXlBtwF5q6RVf3PZGJ9fmFuTkPRO7SruZeVDo9WP8HjbQtOLukYf566e

grzAYR9p74UgWftpDtmrqrRTobiuvsFBxosbeRCvEQCrN0n+p5D9hCVB88tUPHnO

WA4mlduAFZDxQWTApKQ92frHiBqy+M1JFezz2OM3fYN+Dqo/Cb7ZwOAA/2dbwS7o

y4sXEHbfWonjskgPQwFYB23tsFUuM4uZwVEbJg+bveglDsDStbDlfgArXSL/0+ak

lFcHiQEiBBABAgAMBQJKaAqEBQMAEnUAAAoJEJcQuJvKV618rH0H/iCciD4U6YZN

JBj0GN7/Xt851t9FWocmcaC+qtuXnkFhplXkxZVOCU4VBMs4GBoqfIvagbBTyfV4

Di+W8Uxr+/1jiu3l/HvoFxwdwNkGG6zNBhWSjdwQpGwPvh5ryV1OfLX/mgQgdDmx

vqz5+kFDUj4m7uLaeuU2j1T0lR4zU0yAsbt7J3hwfqJCXHOc9bm5nvJwMrSm+sdC

TP5HjUlwHr9mTe8xuZvj6sO/w0P4AqIMxjC9W7pT9q0ofG2KSTwt7wFbh05sbG4U

QYOJe4+Soh3+KjAa1c0cvmIh4cKX9qfCWwhhdeNfh1A9VTHhnl5zTv/UjvnQtjhl

H/Fq1eBSKcSJASIEEAECAAwFAkp5LgoFAwASdQAACgkQlxC4m8pXrXwY6wgAg3f8

76L3qDZTYlFAWs3pXBl8GsUr1DEkTlEDZMZKDM3wPmhaWBR1hMA3y6p3aaCUyJIJ

BEneXzgyU9uqCxXpC78d5qc3xs/Jd/SswzNYuvuzLYOw5wN5L31SLmQTQ8KqE0uo

RynBmtDCQ4M2UKifSnv+0+3mPh85LVAS481GNpL+VVfCYtKesWNu40+98Yg6L9NG

WwRTfsQbcdokZo44Jz7Y7f81ObC4r/X1DgPj2+d4AU/plzDcdrbINOyprs+7340e

cnaGO4Lsgd19b1CvcgJgltRquu3kRvd+Ero2RYpDv6GVK8Ea0Lto4+b/Ae8cLXAh

QnaWQCEWmw+AU4Jbz4kBIgQQAQIADAUCSo5fvQUDABJ1AAAKCRCXELibyletfA08

B/9w8yJdc8K+k07U30wR/RUg3Yb2lBDygmy091mVsyB0RGixBDXEPOXBqGKAXiV1

QSMAXM2VKRsuKahY2HFkPbyhZtjbdTa7Pr/bSnPvRhAh9GNWvvRg2Kp3qXDdjv9x

ywEghKVxcEIVXtNRvpbqRoKmHzIExvUQck5DM1VwfREeYIoxgs4035WADhVMdngQ

S2Gt8P2WaU/p8EZhFGg6X8KtOlD68zGboaJe0hj2VDc+Jc+KdjRfE3fW5IToid/o

DkUaIW6tB3WkXb0g6D/2hrEJbX3headChHKSB8eQdOR9bcCJDhhU8csd501qmrhC

ctmvlpeWQZdIQdk6sABPWeeCiQEiBBABAgAMBQJKoBJHBQMAEnUAAAoJEJcQuJvK

V618Ml8H/1D88/g/p9fSVor4Wu5WlMbg8zEAik3BIxQruEFWda6nART6M9E7e+P1

++UHZsWYs6l9ROpWxRLG1Yy9jLec2Y3nUtb20m65p+IVeKR2a9PHW35WZDV9dOYP

GZabKkO1clLeWLVgp9LRjZ+AeRG+ljHqsULXro1dwewLTB/gg9I2vgNv6dKxyKak

nM/GrqZLATAq2KoaE/u/6lzRFZIzZnLtjZh8X7+nS+V8v9IiY4ntrpkrbvFk30U6

WJp79oBIWwnW/84RbxutRoEwSar/TLwVRkcZyRXeJTapbnLGnQ/lDO1o1d7+Vbjd

q/Sg/cKHHf7NthCwkQNsCnHL0f51gZCJASIEEAECAAwFAkqoEAAFAwASdQAACgkQ

lxC4m8pXrXwE/Af/XD4R/A5R6Ir/nCvKwCTKJmalajssuAcLEa2pMnFZYO/8rzLO

+Gp8p0qFH9C4LFwA0NvR5q6X/swuROf4zxljSvNcdlQVaAfJ2ZDEgJ5GXzsPplrv

SAI9jS3LL7fSWDZgKuUe0a4qx7A0NgyGMUYGhP+QlRFa8vWEBI9fANd/0mMqAeBV

qQyOH0X1FiW1Ca2Jn4NKfuMy9GEvRddVIbB1LvoNVtXPNzeeKMyNb9Jdx1MFWssy

COBP2DayJKTmjvqPEc/YOjOowoN5sJ/jn4mVSTvvlTooLiReSs6GSCAjMVxN7eYS

/Oyq6Iu1JDcJvmB8N2WixAZtAVgF8OA7CWXKVYkBIgQQAQIADAUCSrnHiQUDABJ1

AAAKCRCXELibyletfPChB/9uECti1dZeNuFsd0/RuGyRUVlrrhJE6WCcOrLO9par

rPbewbKBmjSzB0MygJXGvcC06mPNuquJ7/WpxKsFmfg4vJBPlADFKtgRUy9BLzjC

eotWchPHFBVW9ftPbaQViSUu7d89NLjDDM5xrh80puDIApxoQLDoIrh3T1kpZx56

jSWv0gelFUMbXAzmqkJSyL4Xdh1aqzgUbREd7Xf2ICzuh0sV6V7c/AwWtjWEGEsA

HZaiQDywZwbC18GwrMLiAzGWb/AScFDQRCZKJDjL+Ql8YT6z+ZMVr8gb7CIU5PKY

dhiIf2UVTQwLAoW7lNRCQQAqcGjK3IMIz7SO/yk4HmVUiQEiBBABAgAMBQJK3gjG

BQMAEnUAAAoJEJcQuJvKV618jkEH+wb0Zv9z7xQgpLMowVuBFQVu8/z7P5ASumyB

PUO3+0JVxSHBhlCKQK7n11m1fhuGt2fCxXhSU6LzXj36rsKRY53lGZ9QhvqFUtQH

3Xb2IQLIJC4UKjG2jSSCdcuA/x98bwp2v7O03rn7ndCS16CwXnRV3geQoNipRKMS

DajKPpZv1RiZm8pMKqEb8WSw352xWoOcxuffjlsOEwvJ85SEGCAZ9tmIlkZOc7Ai

QONDvii9b8AYhQ60RIQC0HP2ASSmK0V92VeFPxHmAygdDQgZNVtbVxgnnt7oTNEu

VRXNY+z4OfBArp7R+cTsvijDRZY4kML1n22hUybwoxUEvjqZV2+JASIEEAECAAwF

AkrvOlQFAwASdQAACgkQlxC4m8pXrXxrPAgArXiNgZirNuBhfNCXlkzkCHLx5wnV

e4SmTpbWzTwWw7+qk7d4l9hlWtdImISORINzo7f4ShSUzJX2GciNaXhaHRo7+y5O

Zbu82jQb09aQQj/nibKYuqxqUrobTEm+DuYz3JUQZm2PsPcHLS8mX9cxvrJUncPG

nXEV0DRaq71SGWDprtkvBbp6i38aY3sIhYgz8wM5m1szKDtjywmBYcFehIdozt9z

hm7wZshzRWQX1+Rf/pIsnk+OzBIa34crSemTnacbV/B7278z2XAyziPNFuqz0xu+

iltOmYmayfNWAmumuw9NcuwWMlth6Mc2HLrpo0ZBheJ6iuDMPsHnwqdB/4kBIgQQ

AQIADAUCSwBd2gUDABJ1AAAKCRCXELibyletfP6tB/4m1w0BtlkJgtS6E+B/ns14

z4A4PGors+n+MYm05qzvi+EnDF/sytCmVcKeimrtvDcfoDtKAFFvJjcYXfnJdGWm

Pu0SJMRL5KKCirAKwZmU/saxOgoB5QLNw+DHPteJ3w9GmWlGxIqG1r15WC5duzBC

y3FsnjJYG3jaLnHOO9yXXb5h0kUTORfUKdvAr1gxF2KoatZWqGoaPPnHoqb88rjt

zk8I7gDqoXnzh8wLxa0ZYvfTC/McxdWTrwXLft+krmMQ18iIZEne2hvVLNJVuluU

oiWLeHA8iNCQ4W4WTdLc1mCnCjGTMX/MN41uLH0C9Ka4R6wEaqj4lPDk1B/1TV+Q

iQEiBBABAgAMBQJLEYGrBQMAEnUAAAoJEJcQuJvKV618naIH/2t9aH5mBTKBN6fU

qhrf79vIsjtI/QNS5qisBISZMX3/1/0Gu6WnxkPSfdCUJMWCjMcnVj7KU2wxTHHG

VpAStd9r2afUNxRyqZwzwyytktuZok0XngAEDYDDBS3ssu2R4uWLCsC2ysXEqO/5

tI5YrTWJZrfeIphTaYP5hxrMujvqy3kEwKKbiMz91cDeiLS+YCBcalj5n/1dMYf7

8U8C6ieurxAg/L8h6x25VM4Ilx4MmG2T8QGtkkUXd+Fd/KYWmf0LE5LLPknf0Hhw

oVslPXeinp4FsHK/5wzviv4YZpzuTqs9NlKcMsa4IuuPOB0FDf0pn+OFQbEg9QwY

2gCozK+JASIEEAECAAwFAksjTdQFAwASdQAACgkQlxC4m8pXrXwlogf/XBGbXRVX

LMaRN4SczOjwT3/tUCriTkb3v+zKjRG90zFhYAccjn7w+7jKQicjq6quQG1EH2X4

/Su6ps1lDLqGHHhiJW3ZhxQScLZmhdAYsh2qG4GP/UW3QjXG7c61t+H3olvWg2cr

wqCxxFZAgkAAkr9xcHWFZJEQeXoob6cCZObaUnHSANdmC6s5lUxXYa2bmL7Q3UB4

4KCzDvAfbPZKJOw9k0qb3lc11zx+vGdyZFbm4R0+3LPp/vT0b3GlSbbF9lU1GOXh

VaphrgFFa76dmjfHCkPplXAkK1VSIU/aPGAefduTFMdlSZpdMtJ5AULjGcszBDlR

pLlPxvqVa0ZpgIkBIgQQAQIADAUCSycmkgUDABJ1AAAKCRCXELibyletfHlNCACp

1YespiHfQt2alcscE5zgfETEHHic8Ai6pNkU9HT4TeWcFHEDe5QqfYcpjLrQvBXS

kSvxEittbyRdv+e+j5Z+HyHjiG8nAQBL6qy9eHqQE4+d7gYs6DTk7sG9ZMYphREb

ltzD+F4hVCQdLT8LNr0eVFN7ehqECScDaCG8/Qyti+l/0M902/Yn+mz0ilOiUdWJ

9x6LPaIINtb1gsYDEylLjwGIZmI0r5Kh9wYoV4vnNezFbxO1uRiW0B7iaPjIEsbt

OOKp7wx2aX+DM3N9F3BtaIY8XnzcnomNm83SNsgmgrZljpQltUnNqIhNM8DupQ+I

WOV5gtl6pTC7CgeVTVyRiQEiBBABAgAMBQJLOGXuBQMAEnUAAAoJEJcQuJvKV618

ll4IAKJ9mm4jb0c8fe9+uDI8eCJRbzNbVXm8zWzpA8GUtQAakwxoKv332QP1Wa1P

odni/e3EMhsSREOZJJv79YqGxGRBTE9Kb/VjM34nas4XSnXKW28XWhKyIw+XwQAi

nY2swFHh+83Htr/mwTdJfS2aEYl2zboBvd/JZCdhOGU2GH737S/3uEczoKkfVQ/w

OTM8X1xWwlYWqx23k/DsGcuDs9lA2g7Mx7DSqBtVjaTkn9h0zATzXLDkmP4SAUVj

cZ83WDpFre5WnizZjdXlBMM5OCexp5WpmzyHLTnaBFK4jEmnsk5C2Rnoyp8Ivz6g

Ecg1tRbEXijRw++d2TFYlJwLKtiJASIEEAECAAwFAktKMicFAwASdQAACgkQlxC4

m8pXrXxqHQgAuYY5scKrh0m/GS9EYnyC9494lOlO6iytU0CpE6oBC31M3hfX/Dbj

UbcS5szZNU+2CPYo4ujQLZ7suN7+tTjG6pZFfMevajT9+jsL+NPMF8RLdLOVYmbl

TmSQGNO+XGEYaKYH5oZIeIW5AKCgi2ozkdFlBBLAx7Kqo/FyybhkURFEcvEyVmgf

3KLV7IIiX/fYLfoCMCJ/Lcm9/llSFB1n8Nvg66Xd533DKoHjueD3jyaNAVlo2mq/

sIAv++kntvOiB3GDK5pfwHZ78WWiCpsWZpE5gzAnzJ1Y0WEigRo0PVLu3cLO0jLG

23d+H/CbfZ8rkajHJeCDQF7YVmP0t0nYpYkBIgQQAQIADAUCS1v+ZgUDABJ1AAAK

CRCXELibyletfNS/CACqt2TkB86mjqM+cJ74+dWBvJ2aFuURuxzm95i9Q/W/hU08

2iMbC3+0k2oD8CrTOe61P+3oRyLjv/UEDUNzLncNe2YsA9JeV+4hvPwH5Vp3Om13

089fCKZUbqslXNKkHiWYU+zAaZJXEuGRmRz0HbQIeAMOWF4oa226uo1e4ws1Jhc+

F3E/ApCRyFBqBUdL05hapQLditYpsBjIdiBGpjzidMLE2wX2W4ZpAdN0U6BIyIqR

mTPjbSkvzS9kSWFmfhQgnBDKEYJpVZgE1sN52rYC1sDeGeiuKxlzjVov9MMhYMWa

Zo3R5o3F2iIM/BK6FbC252lf/Mhu3ICuXujNBZNYiQEiBBABAgAMBQJLbSH4BQMA

EnUAAAoJEJcQuJvKV618kd0IAJLLwDH6gvgAlBFklQJXqQxUdcSOOVMAWtlHgWOy

ozjgomZZBkRL8dtCDr9YBMcj5czcQ3qpmLJdppXhKB+kJV2iUXfDMSFXwJ4wLfIs

8FNnXw8H5U01oBkGH/Ku6ngL9Vwt+MjYHtCWkw9QueUKZnDudX9qIzLAIt+mwSTu

A6+fY4VWIg40AA0v3exaQM55YR/UhlKunpGG9o8Qkq77dMEbTMpOmBoLbOMRB3Dd

MAvVU6G2l6Pcb7KobVCuOBnb6batXARV/G8sw+nzfJ16fr/KobZT2A6m+Jrqk4dl

F14ljLbz16O5JGUPAryN2G2ddBdSAy7dtFSVhWWiWC9n88q5Ag0EPj6jHRAIAO/h

iX8WzHWOMLJT54x/axeDdqn1rBDf5cWmaCWHN2ujNNlgpx5emoU9v7QStsNUCOGB

bXkeO4Ar7YG+jtSR33zqNh3y5kQ0YkY3dQ0wh6nsl+wh4XIIY/3TUZVtmdJeUBRH

JlfVNFYad2hX1guFI37Ny1PoZAFsxO82g+XB/Se8r/+sbmVcONdcdIeFKrE3FjLt

IjNQcxC6l9Q2Oy8KDxG/zvUZG3+H5i3tdRMyGgmuD6gEV0GXOHYUopzLeit1+Aa0

bCk36Mwbu+BeOw/CJW3+b0mB27hOaf9aCA855IP6fJFvtxcblq8nHIqhU3Dc9tec

sl9/S1xZ5S8ylG/xeRsAAwUH/i8KqmvAhq0X7DgCcYputwh37cuZlHOa1Ep07JRm

BCDgkdQXkGrsj2Wzw7Aw/TGdWWkmn2pxb8BRui5cfcZFO7c6vryi6FpJuLucX975

+eVY50ndWkPXkJ1HF4i+HJwRqE2zliN/RHMs4LJcwXQvvjD43EE3AO6eiVFbD+qA

AdxUFoOeLblKNBHPG7DPG9xL+Ni5rkE+TXShxsB7F0z7ZdJJZOG0JODmox7IstQT

GoaU9u41oyZTIiXPiFidJoIZCh7fdurP8pn3X+R5HUNXMr7M+ba8lSNxce/F3kmH

0L7rsKqdh9d/aVxhJINJ+inVDnrXWVoXu9GBjT8Nco1iU9SIVAQYEQIADAUCTnc9

7QUJE/sBuAASB2VHUEcAAQEJEIxxjTtQcuH1FJsAmwWK9vmwRJ/y9gTnJ8PWf0BV

roUTAKClYAhZuX2nUNwH4vlEJQHDqYa5yQ==

=ghXk

-----END PGP PUBLIC KEY BLOCK-----

**要将构建的密钥导入你的个人公共 GPG 密钥圈，使用 gpg --import。例如，如果你把钥匙保存在一个名为 mysql\_pubkey.asc 的文件中，导入命令看起来像这样。**

shell> **gpg --import mysql\_pubkey.asc**

gpg: key 5072E1F5: public key "MySQL Release Engineering

<mysql-build@oss.oracle.com>" imported

gpg: Total number processed: 1

gpg: imported: 1

gpg: no ultimately trusted keys found

**你也可以使用公钥 ID 5072E1F5 从公钥服务器下载钥匙。**

shell> gpg --recv-keys 5072E1F5

gpg: requesting key 5072E1F5 from hkp server keys.gnupg.net

gpg: key 5072E1F5: "MySQL Release Engineering <mysql-build@oss.oracle.com>"

1 new user ID

gpg: key 5072E1F5: "MySQL Release Engineering <mysql-build@oss.oracle.com>"

53 new signatures

gpg: no ultimately trusted keys found

gpg: Total number processed: 1

gpg: new user IDs: 1

gpg: new signatures: 53

**如果你想把密钥导入你的RPM配置以验证RPM安装包，你应该可以直接导入密钥。**

shell> **rpm --import mysql\_pubkey.asc**

**如果你遇到问题或需要RPM的具体信息，请参阅第2.1.4.4节，"使用RPM进行签名检查"。**

**在你下载并导入公共构建密钥后，下载你想要的MySQL包和相应的签名，这也可以从下载页面获得。签名文件的名称与分发文件相同，扩展名为.asc，如下表的例子所示。**

**表2.1 源文件的MySQL包和签名文件**

|  |  |
| --- | --- |
| **File Type** | **File Name** |
| Distribution file | **mysql-standard-8.0.26-linux-i686.tar.gz** |
| Signature file | **mysql-standard-8.0.26-linux-i686.tar.gz.asc** |

**确保两个文件都存储在同一个目录下，然后运行下面的命令来验证发行文件的签名。**

shell> **gpg --verify *package\_name*.asc**

**如果下载的软件包是有效的，你应该看到一个类似这样的 Good signature 消息。**

shell> **gpg --verify mysql-standard-8.0.26-linux-i686.tar.gz.asc**

gpg: Signature made Tue 01 Feb 2011 02:38:30 AM CST using DSA key ID 5072E1F5

gpg: Good signature from "MySQL Release Engineering <mysql-build@oss.oracle.com>"

**良好的签名信息表明，与我们网站上列出的签名相比，文件签名是有效的。但你也可能看到警告，像这样。**

shell> **gpg --verify mysql-standard-8.0.26-linux-i686.tar.gz.asc**

gpg: Signature made Wed 23 Jan 2013 02:25:45 AM PST using DSA key ID 5072E1F5

gpg: checking the trustdb

gpg: no ultimately trusted keys found

gpg: Good signature from "MySQL Release Engineering <mysql-build@oss.oracle.com>"

gpg: WARNING: This key is not certified with a trusted signature!

gpg: There is no indication that the signature belongs to the owner.

Primary key fingerprint: A4A9 4068 76FC BD3C 4567 70C8 8C71 8D3B 5072 E1F5

**这很正常，因为它们取决于你的设置和配置。下面是对这些警告的解释。**

* **gpg: 没有找到最终受信任的钥匙。这意味着特定的密钥没有被你或你的信任网络 "最终信任"，这对于验证文件签名的目的来说是可以的。**
* **警告：这把钥匙没有经过可信签名的认证! 没有迹象表明该签名是属于所有者的。这指的是你的信任程度，你认为你拥有我们真正的公钥。这是一个个人决定。理想情况下，MySQL开发人员会亲自交给你钥匙，但更常见的是你下载了它。下载的东西被篡改了吗？可能没有，但这个决定取决于你。设置一个信任网是信任他们的一种方法。**

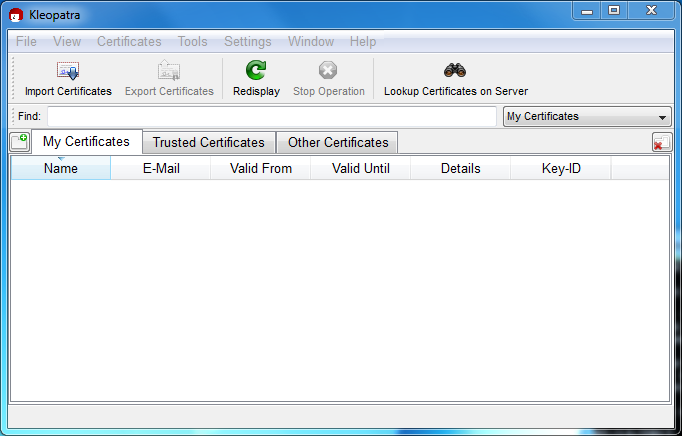
**关于如何使用公钥的更多信息，请参见GPG文档。**

#### **2.1.4.3 使用 Gpg4win for Windows 进行签名检查**

**第2.1.4.2节 "使用GnuPG进行签名检查 "描述了如何使用GPG验证MySQL下载。该指南也适用于Microsoft Windows，但另一个选择是使用Gpg4win这样的GUI工具。你可以使用一个不同的工具，但我们的例子是基于Gpg4win的，并利用其捆绑的Kleopatra GUI。**

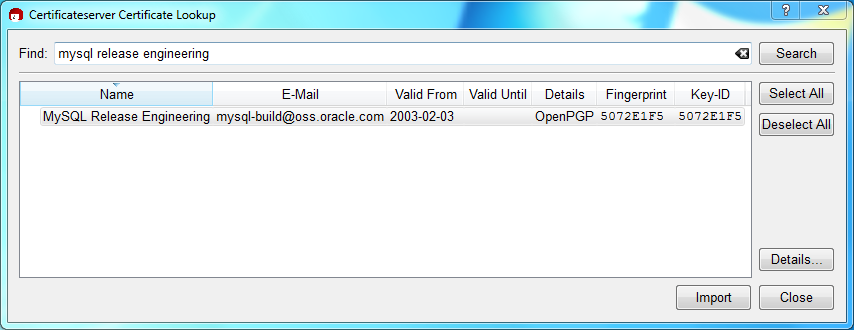
**下载并安装 Gpg4win，然后加载 Kleopatra。对话框看起来应该类似于。**

**图2.1 Kleopatra：初始屏幕**



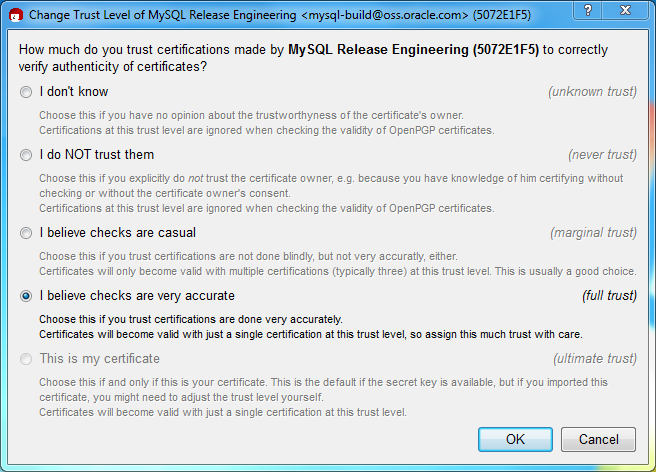
**接下来，添加MySQL发布工程证书。通过点击 "文件"、"服务器上的证书查询 "来完成此操作。在搜索框中输入 "Mysql Release Engineering"，然后按搜索。**

**图2.2 Kleopatra: 在服务器上查找证书向导。找到一个证书**



**选择 "MySQL Release Engineering "证书。指纹和密钥ID必须是 "5072E1F5"，或者选择 "详细信息... "来确认证书是否有效。现在，通过点击导入来导入它。当显示导入对话框时，选择 "好"，该证书现在应该被列在导入的证书标签下。**

**接下来，为我们的证书配置信任级别。选择我们的证书，然后从主菜单中选择证书，改变所有者信任....。我们建议为我们的证书选择 "我相信检查是非常准确的"，否则你可能无法验证我们的签名。选择 "我相信检查是非常准确的 "来启用 "完全信任"，然后按确定。**



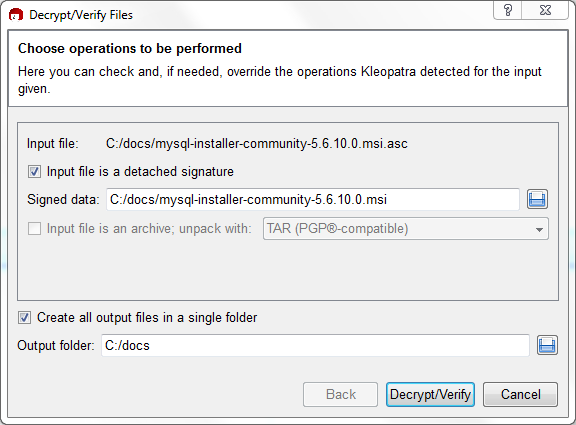
**接下来，验证下载的MySQL包文件。这需要打包的文件和签名的文件。签名文件的名称必须与打包的文件相同，但有一个附加的.asc扩展名，如下表的例子所示。签名在每个MySQL产品的下载页面上都有链接。你必须用这个签名来创建.asc文件。**

**Table 2.2 MySQL Package and Signature Files for MySQL Installer for Microsoft Windows**

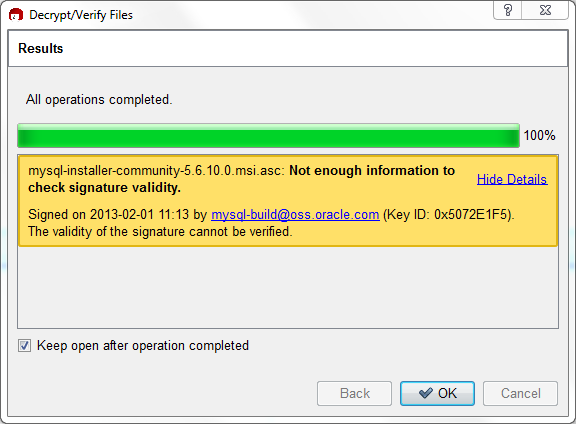
|  |  |
| --- | --- |
| **File Type** | **File Name** |
| Distribution file | **mysql-installer-community-8.0.26.msi** |
| Signature file | **mysql-installer-community-8.0.26.msi.as** |

确保两个文件都存储在同一目录下，然后运行以下命令来验证分发文件的签名。可以将签名（.asc）文件拖入Kleopatra，或者从文件，解密/验证文件...加载对话框，然后选择.msi或.asc文件。

图2.4 Kleopatra: 解密和验证文件对话框

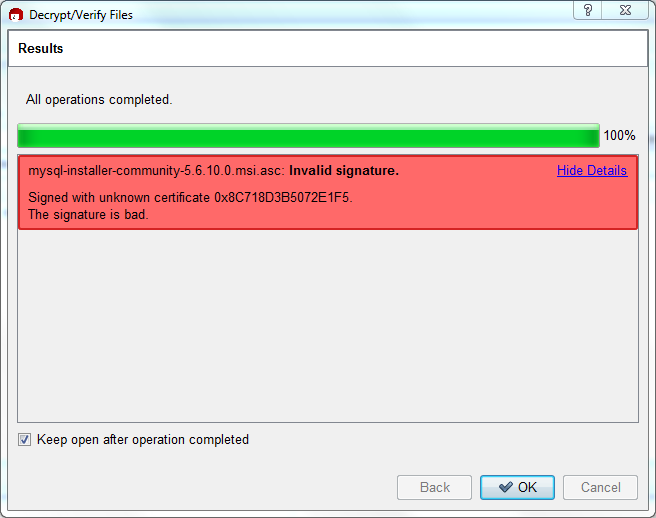


点击解密/验证来检查文件。两种最常见的结果如下图所示；虽然黄色警告看起来有问题，但下面意味着文件检查成功通过。你现在可以运行这个安装程序了



看到红色的The signature is bad错误意味着该文件是无效的。如果你看到这个错误，请不要执行MSI文件。

**Figure 2.6 Kleopatra: the Decrypt and Verify Results Dialog: Bad**



第2.1.4.2节，"使用GnuPG进行签名检查 "一节解释了为什么你没有看到绿色的良好签名结果。

#### 2.1.4.4 使用RPM进行签名检查

对于RPM包，没有单独的签名。RPM包有一个内置的GPG签名和MD5校验码。你可以通过运行以下命令来验证一个包。

shell> **rpm --checksig *package\_name*.rpm**

例子。

shell> **rpm --checksig MySQL-server-8.0.26-0.linux\_glibc2.5.i386.rpm**

MySQL-server-8.0.26-0.linux\_glibc2.5.i386.rpm: md5 gpg OK

注意

*如果你使用RPM 4.1并且它抱怨（GPG）不确定（MISSING KEYS: GPG#5072e1f5），即使你已经将MySQL的公共构建密钥导入你自己的GPG密钥圈，你需要先将该密钥导入RPM密钥圈。RPM 4.1不再使用你的个人GPG钥匙圈（或GPG本身）。相反，RPM维护一个单独的钥匙圈，因为它是一个全系统的应用程序，而一个用户的GPG公钥圈是一个用户特定的文件。要导入MySQL公钥到RPM的钥匙圈，首先要获得钥匙，然后使用rpm --import来导入钥匙。例如。*

shell> gpg --export -a 5072e1f5 > 5072e1f5.asc

shell> rpm --import 5072e1f5.asc

另外，rpm也支持直接从URL加载密钥，你可以使用这个手册页面。

shell> **rpm --import https://dev.mysql.com/doc/refman/8.0/en/checking-gpg-signature.html**

如果你需要获得MySQL公钥，请参阅第2.1.4.2节，"使用GnuPG进行签名检查"。

### 2.1.5 安装布局

不同的安装类型（例如，本地包、二进制压缩包和源压缩包）的安装布局是不同的，这可能导致在管理不同的系统或使用不同的安装源时出现混乱。各自的布局在相应的安装类型或平台章节中给出，如下所述。请注意，来自 Oracle 以外的供应商的安装布局可能与这些布局不同。

### 2.1.6 特定于编译器的构建特征

在某些情况下，用于构建MySQL的编译器会影响到可使用的特性。本节中的说明适用于由Oracle公司提供的或你自己从源码编译的二进制发行版。

icc（Intel C++编译器）构建

用icc构建的服务器有这些特点。

* 不包括SSL支持。

## 2.2 使用通用二进制文件在Unix/Linux上安装MySQL

甲骨文公司提供了一套MySQL的二进制发行版。这些包括为一些平台提供的压缩tar文件（扩展名为.tar.xz的文件）形式的通用二进制发布，以及为选定平台提供的特定平台包格式的二进制文件。

本节涵盖了在Unix/Linux平台上从压缩的tar文件二进制分布中安装MySQL。对于关注MySQL安全特性的Linux-通用二进制分布的安装说明，请参阅《安全部署指南》。对于其他平台特定的二进制包格式，请参阅本手册中其他平台特定的章节。例如，对于Windows发行版，见第2.3节 "在Microsoft Windows上安装MySQL"。关于如何获得不同分发格式的MySQL，见第2.1.3节 "如何获得MySQL"。

MySQL压缩的tar文件二进制发行版的名称是mysql-VERSION-OS.tar.xz，其中VERSION是一个数字（例如，8.0.26），OS表示该发行版所针对的操作系统类型（例如，pc-linux-i686或winx64）。

还有一个 "最小安装 "版本的MySQL压缩tar文件，用于Linux通用二进制发行版，它的名字是mysql-VERSION-OS-GLIBCVER-ARCH-minimal.tar.xz。最小安装发行版不包括调试二进制文件，并且剥离了调试符号，使得它比普通二进制发行版小得多。如果你选择安装最小安装版本，记得在后面的说明中调整文件名格式的差异。

警告

* 如果你以前使用你的操作系统本地软件包管理系统（如Yum或APT）安装MySQL，你可能会遇到使用本地二进制安装的问题。确保你以前的MySQL安装已被完全删除（使用你的软件包管理系统），并且任何额外的文件，如你的数据文件的旧版本，也已被删除。你还应该检查配置文件，如/etc/my.cnf或/etc/mysql目录并删除它们。

关于用官方MySQL包替换第三方包的信息，请参见相关的APT指南或Yum指南。

* MySQL有一个对libaio库的依赖性。如果本地没有安装这个库，数据目录初始化和随后的服务器启动步骤会失败。如果有必要，使用适当的软件包管理器安装它。例如，在基于Yum的系统上。

shell> ***yum search libaio*** # search for info

shell> ***yum install libaio*** # install library

或者，在基于 APT 的系统上。

shell> ***apt-cache search libaio*** # search for info

shell> ***apt-get install libaio1*** # install library

* Oracle Linux 8 / Red Hat 8 (EL8)。这些平台默认不安装文件/lib64/libtinfo.so.5，这是MySQL客户端bin/mysql对软件包mysql-VERSION-el7-x86\_64.tar.gz和mysql-VERSION-linux-glibc2.12-x86\_64.tar.xz的要求。为了解决这个问题，安装ncurses-compat-libs包。

shell> ***yum install ncurses-compat-libs***

要安装一个压缩的tar文件二进制分布，在你选择的安装位置（通常是/usr/local/mysql）解压。这将创建下表中所示的目录。

表2.3 通用Unix/Linux二进制包的MySQL安装布局

**Table 2.3 MySQL Installation Layout for Generic Unix/Linux Binary Package**

|  |  |
| --- | --- |
| **Directory** | **Contents of Directory** |
| bin | [**mysqld**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#mysqld) server, client and utility programs |
| docs | MySQL manual in Info format |
| man | Unix manual pages |
| include | Include (header) files |
| lib | Libraries |
| share | Error messages, dictionary, and SQL for database installation |
| **support-files** | Miscellaneous support files |

mysqld二进制文件的调试版本可以作为mysqld-debug使用。要从源码发布中编译你自己的调试版本的MySQL，使用适当的配置选项来启用调试支持。参见第2.9节，"从源码安装MySQL"。

要安装和使用一个MySQL二进制发行版，命令序列看起来像这样。

shell> **groupadd mysql**

shell> **useradd -r -g mysql -s /bin/false mysql**

shell> **cd /usr/local**

shell> **tar xvf */path/to/mysql-VERSION-OS*.tar.xz**

shell> **ln -s *full-path-to-mysql-VERSION-OS* mysql**

shell> **cd mysql**

shell> **mkdir mysql-files**

shell> **chown mysql:mysql mysql-files**

shell> **chmod 750 mysql-files**

shell> **bin/mysqld --initialize --user=mysql**

shell> **bin/mysql\_ssl\_rsa\_setup**

shell> **bin/mysqld\_safe --user=mysql &**

# Next command is optional

shell> **cp support-files/mysql.server /etc/init.d/mysql.server**

注意

这个过程假设你有系统的root（管理员）权限。或者，你可以使用sudo(Linux)或pfexec(Solaris)命令给每个命令加上前缀。

mysql-files目录提供了一个方便的位置，可以作为secure\_file\_priv系统变量的值，它将导入和导出操作限制在一个特定的目录中。参见第5.1.8节，"服务器系统变量"。

下面是前面关于安装二进制发行版的更详细版本的描述。

创建一个mysql用户和组

如果你的系统还没有一个用于运行mysqld的用户和组，你可能需要创建它们。下面的命令添加了mysql组和mysql用户。你可能想把用户和组称为其他东西而不是mysql。如果是这样，在下面的指令中替换适当的名称。在不同版本的Unix/Linux上，useradd和groupadd的语法可能略有不同，或者它们可能有不同的名字，如adduser和addgroup。

shell> **groupadd mysql**

shell> **useradd -r -g mysql -s /bin/false mysql**

注意

因为用户只是出于所有权的目的，而不是登录的目的，所以useradd命令使用-r和-s /bin/false选项来创建一个对你的服务器主机没有登录权限的用户。如果你的useradd不支持这些选项，请省略它们。

获取并解压发行版

选择你想解压发行版的目录，并把位置改到它那里。这里的例子是在 /usr/local 下解压发行版。因此，说明中假设你有权限在 /usr/local 中创建文件和目录。如果该目录受到保护，你必须以root身份进行安装。

shell> **cd /usr/local**

使用第2.1.3节 "如何获得MySQL "中的说明获得一个分发文件。对于一个给定的版本，所有平台的二进制分发文件都是从相同的MySQL源分发文件构建的。

解压发行版，这将创建安装目录。如果tar有z选项支持，它可以解压和解压发行版。

shell> **tar xvf */path/to/mysql-VERSION-OS*.tar.xz**

tar命令创建了一个名为mysql-VERSION-OS的目录。

要从压缩的tar文件二进制发行版中安装MySQL，你的系统必须有GNU XZ Utils来解压缩发行版，并有合理的tar来解包。

注意

*在MySQL Server 8.0.12中，压缩算法从Gzip变为XZ；通用二进制文件的扩展名也从.tar.gz变为.tar.xz。*

众所周知，GNU tar是可以工作的。一些操作系统提供的标准tar不能解开MySQL发布中的长文件名。你应该下载并安装GNU tar，或者如果有的话，使用预装版本的

如果你的 tar 不支持 xz 格式，那么就用 xz 命令来解压发行版，用 tar 来解压。用下面的替代命令代替前面的tar命令来解压和提取发行版。

shell> **xz -dc */path/to/mysql-VERSION-OS*.tar.xz | tar x**

接下来，创建一个符号链接到由 tar 创建的安装目录。

shell> **ln -s *full-path-to-mysql-VERSION-OS* mysql**

ln命令为安装目录建立一个符号链接。这使你能够更容易地把它称为/usr/local/mysql。为了避免你在使用MySQL时总是要输入客户程序的路径名称，你可以将/usr/local/mysql/bin目录添加到你的PATH变量中。

shell> **export PATH=$PATH:/usr/local/mysql/bin**

执行安装后的设置

安装过程的其余部分涉及设置分发所有权和访问权限、初始化数据目录、启动MySQL服务器和设置配置文件。有关说明，见第2.10节 "安装后设置和测试"。

## 2.3 在Microsoft Windows上安装MySQL

/

重要的是

MySQL 8.0服务器需要Microsoft Visual C++ 2015 Redistributable Package才能在Windows平台上运行。用户在安装服务器之前，应确保该软件包已经安装在系统上。该软件包可在微软下载中心获得。此外，MySQL调试二进制文件需要安装Visual Studio 2015。

MySQL仅适用于微软Windows 64位操作系统。关于支持的Windows平台信息，见https://www.mysql.com/support/supportedplatforms/database.html。

在Microsoft Windows上安装MySQL有不同的方法。

**MySQL安装程序方法**

最简单和推荐的方法是下载MySQL安装器（用于Windows），让它安装和配置特定版本的MySQL服务器，方法如下。

1、从https://dev.mysql.com/downloads/installer/ 下载MySQL安装器并执行它。

注意

与标准的MySQL安装程序不同，较小的网络社区版本不捆绑任何MySQL应用程序，而是只下载你选择安装的MySQL产品。

2、确定初始安装MySQL产品时要使用的设置类型。例如。

* 开发者默认。提供一种设置类型，包括所选版本的MySQL Server和与MySQL开发有关的其他MySQL工具，如MySQL工作台。
* 仅限服务器。提供所选版本的MySQL Server的设置，而没有其他产品。
* 自定义。使你能够选择任何版本的MySQL服务器和其他MySQL产品。

3、安装服务器实例（和产品），然后按照屏幕上的说明开始进行服务器配置。关于每个单独步骤的更多信息，请参见第2.3.3.3.1节，"用MySQL安装程序配置MySQL服务器"。

现在MySQL已经安装完毕。如果你将MySQL配置为一个服务，那么每次重启系统时，Windows都会自动启动MySQL服务器。此外，这个过程还在本地主机上安装了MySQL安装程序，你以后可以用它来升级或重新配置MySQL服务器。

注意

如果你在系统上安装了MySQL Workbench，可以考虑用它来检查你新的MySQL服务器连接。默认情况下，该程序在安装MySQL后自动启动。

**其他安装信息**

可以将MySQL作为一个标准的应用程序或作为一个Windows服务运行。通过使用服务，你可以通过标准的Windows服务管理工具监测和控制服务器的运行。欲了解更多信息，请参见第2.3.4.8节 "以Windows服务方式启动MySQL"。

为了适应RESTART语句，当作为服务或独立运行时，MySQL服务器会分叉，以使一个监控进程监督服务器进程。在这种情况下，有两个mysqld进程。如果不需要RESTART能力，可以用--no-monitor选项启动服务器。参见第13.7.8.8节，"RESTART语句"。

一般来说，你应该用一个有管理员权限的账户在Windows上安装MySQL。否则，你可能会在某些操作上遇到问题，如编辑PATH环境变量或访问服务控制管理器。安装时，MySQL不需要使用具有管理员权限的用户来执行。

关于在Windows平台上使用MySQL的限制清单，见第2.3.7节 "Windows平台限制"。

除了MySQL服务器包外，你可能需要或想要额外的组件来将MySQL用于你的应用程序或开发环境。这些包括，但不限于。

* 要使用ODBC连接到MySQL服务器，你必须有一个Connector/ODBC驱动程序。欲了解更多信息，包括安装和配置说明，请参阅《MySQL Connector/ODBC开发指南》。

注意事项

MySQL安装程序为你安装和配置了Connector/ODBC。

* 要在.NET应用程序中使用MySQL服务器，你必须有Connector/NET驱动程序。欲了解更多信息，包括安装和配置说明，请参见《MySQL Connector/NET开发人员指南》。

注意

MySQL安装程序为你安装和配置MySQL Connector/NET。

用于Windows的MySQL发行版可以从https://dev.mysql.com/downloads/。参见第2.1.3节 "如何获得MySQL"。

用于Windows的MySQL有几种发布格式，详见这里。一般来说，你应该使用MySQL安装程序。它比老的MSI包含更多的功能和MySQL产品，比压缩文件更简单易用，而且你不需要额外的工具就能让MySQL启动和运行。MySQL安装程序自动安装MySQL服务器和其他MySQL产品，创建一个选项文件，启动服务器，并使你能够创建默认用户账户。关于选择软件包的更多信息，见第2.3.2节 "选择安装包"。

* 一个MySQL安装程序发行版包括MySQL服务器和其他MySQL产品，包括MySQL工作台，以及MySQL for Visual Studio。MySQL Installer也可以用来在未来升级这些产品（见https://dev.mysql.com/doc/mysql-compat-matrix/en/）。

关于使用MySQL安装器安装MySQL的说明，请参见第2.3.3节 "Windows的MySQL安装器"。

* 标准的二进制发布（以压缩文件的形式打包）包含所有必要的文件，你可以解压到你选择的位置。该包包含完整的Windows MSI安装包中的所有文件，但不包括安装程序。

关于使用压缩文件安装MySQL的说明，见第2.3.4节，"使用noinstall ZIP档案在Microsoft Windows上安装MySQL"。

* 源码分发格式包含使用Visual Studio编译器系统构建可执行文件的所有代码和支持文件。

有关在Windows上从源码构建MySQL的说明，见第2.9节 "从源码安装MySQL"。

**Windows上的MySQL考虑因素**

* 大表支持

如果你需要大小超过4GB的表，在NTFS或更新的文件系统上安装MySQL。当你创建表时，不要忘记使用MAX\_ROWS和AVG\_ROW\_LENGTH。见第13.1.20节，"CREATE TABLE语句"。

* MySQL和病毒检查软件

病毒扫描软件，如Norton/Symantec Anti-Virus，在包含MySQL数据和临时表的目录上，可能会导致问题，既影响MySQL的性能，又使病毒扫描软件误认为文件内容包含垃圾邮件。这是由于病毒扫描软件使用的指纹机制，以及MySQL快速更新不同文件的方式，可能被识别为潜在的安全风险。

在安装MySQL服务器后，建议你禁用用于存储MySQL表数据的主目录（datadir）的病毒扫描。病毒扫描软件中通常有一个系统，可以使特定的目录被忽略。

此外，默认情况下，MySQL在标准的Windows临时目录中创建临时文件。为了防止临时文件也被扫描，为MySQL临时文件配置一个单独的临时目录，并将该目录添加到病毒扫描排除列表中。要做到这一点，请在你的my.ini配置文件中为tmpdir参数增加一个配置选项。更多信息，见第2.3.4.2节 "创建选项文件"。

### 2.3.1 Microsoft Windows上的MySQL安装布局

对于Windows上的MySQL 8.0，对于用MySQL安装程序进行的安装，默认的安装目录是C:\Program Files\MySQL\MySQL Server 8.0。如果你使用ZIP存档方法来安装MySQL，你可能更喜欢安装在C:\mysql。然而，子目录的布局仍然是一样的。

所有的文件都位于这个父目录内，使用下表中所示的结构。

**Table 2.4 Default MySQL Installation Layout for Microsoft Windows**

|  |  |  |
| --- | --- | --- |
| **Directory** | **Contents of Directory** | **Notes** |
| bin | [**mysqld**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#mysqld) server, client and utility programs |  |
| %PROGRAMDATA%\MySQL\MySQL Server 8.0\ | Log files, databases | The Windows system variable %PROGRAMDATA% defaults to C:\ProgramData. |
| docs | Release documentation | With MySQL Installer, use the **Modify** operation to select this optional folder. |
| include | Include (header) files |  |
| lib | Libraries |  |
| share | Miscellaneous support files, including error messages, character set files, sample configuration files, SQL for database installation |  |

### 2.3.2 选择一个安装包

对于MySQL 8.0，在Windows上安装MySQL时有多种安装包格式可供选择。本节中描述的包格式是。

* MySQL安装程序
* MySQL noinstall ZIP档案
* MySQL Docker图像

程序数据库（PDB）文件（文件名扩展名为pdb）在出现问题时为调试你的MySQL安装提供信息。这些文件包括在MySQL的ZIP档案分发中（但不是MSI分发）。

**MySQL安装程序**

这个软件包的文件名类似于mysql-installer-community-8.0.26.0.msi或mysql-installer-commercial-8.0.26.0.msi，它利用MSI自动安装MySQL服务器和其他产品。MySQL安装程序下载并将更新应用到自身和每个已安装的产品上。它还配置已安装的MySQL服务器（包括一个沙盒InnoDB集群测试设置）和MySQL路由器。建议大多数用户使用MySQL安装器。

MySQL安装器可以安装和管理（添加、修改、升级和删除）许多其他MySQL产品，包括。

* 应用程序--MySQL Workbench、MySQL for Visual Studio、MySQL Shell和MySQL Router（见<https://dev.mysql.com/doc/mysql-compat-matrix/en/>）
* 连接器--MySQL Connector/C++, MySQL Connector/NET, Connector/ODBC, MySQL Connector/Python, MySQL Connector/J, MySQL Connector/Node.js
* 文档 - MySQL手册（PDF格式）、样本和示例

MySQL安装程序可在所有MySQL支持的Windows版本上运行（见https://www.mysql.com/support/supportedplatforms/database.html）。

注意

由于MySQL安装程序不是微软Windows的本地组件，而是依赖于.NET，因此它不能与Windows Server的Server Core版本等最小安装选项一起使用。

关于如何使用MySQL安装器安装MySQL的说明，请参见第2.3.3节 "Windows的MySQL安装器"。

**MySQL noinstall ZIP档案**

这些包包含完整的MySQL服务器安装包中的文件，但GUI除外。这种格式不包括自动安装程序，必须手动安装和配置。

noinstall ZIP档案被分割成两个独立的压缩文件。主包被命名为mysql-VERSION-winx64.zip。这包含在你的系统上使用MySQL所需的组件。可选的MySQL测试套件、MySQL基准套件和调试二进制文件/信息组件（包括PDB文件）都在一个单独的压缩文件中，名为mysql-VERSION-winx64-debug-test.zip。

如果你选择安装noinstall ZIP档案，请参见第2.3.4节，"使用noinstall ZIP档案在Microsoft Windows上安装MySQL"。

**MySQL Docker图像**

关于在Windows平台上使用Oracle提供的MySQL Docker映像的信息，请参见第2.5.6.3节，"使用Docker在Windows和其他非Linux平台上部署MySQL"。

警告

甲骨文提供的MySQL Docker镜像是专门为Linux平台构建的。其他平台不受支持，用户在这些平台上运行甲骨文提供的MySQL Docker映像，其风险自负。

### 2.3.3 用于Windows的MySQL安装程序

MySQL Installer是一个独立的应用程序，旨在减轻在Microsoft Windows上运行的MySQL产品的安装和配置的复杂性。它支持以下MySQL产品。

* MySQL服务器

MySQL Installer可以同时在同一主机上安装和管理多个独立的MySQL服务器实例。例如，MySQL Installer可以在同一主机上安装、配置和升级MySQL 5.6、MySQL 5.7和MySQL 8.0的独立实例。MySQL安装程序不允许在主要和次要版本号之间进行服务器升级，但允许在一个发布系列内进行升级（如8.0.21到8.0.22）。

注意

MySQL安装程序不能在同一主机上同时安装MySQL服务器的社区和商业版本。如果你需要在同一台主机上安装这两个版本，请考虑使用ZIP存档分发来安装其中一个版本。

* MySQL应用程序

MySQL Workbench、MySQL Shell、MySQL Router和MySQL for Visual Studio。

* MySQL连接器

MySQL Connector/NET、MySQL Connector/Python、MySQL Connector/ODBC、MySQL Connector/J，以及MySQL Connector/C++。要安装MySQL Connector/Node.js，请参阅https://dev.mysql.com/downloads/connector/nodejs/。

* 文档和示例

PDF格式的MySQL参考手册（按版本）和MySQL数据库样本（按版本）。

**安装要求**

MySQL安装程序需要Microsoft .NET Framework 4.5.2或更高版本。如果主机上没有安装这个版本，你可以通过访问微软网站下载。

需要互联网连接来下载包含最新MySQL产品元数据的清单，这些产品不属于完整捆绑的一部分。当你第一次启动应用程序时，MySQL安装程序会尝试下载清单，然后以可配置的时间间隔定期下载（见MySQL安装程序选项）。或者，你可以通过点击MySQL安装程序仪表板中的目录来手动检索更新的清单。

注意

如果第一次或随后的清单下载不成功，会记录一个错误，并且在你的会话期间，你对MySQL产品的访问可能受到限制。MySQL安装程序会在每次启动时尝试下载清单，直到初始清单结构被更新。有关查找产品的帮助，请参阅查找要安装的产品。

**MySQL安装程序社区版**

从https://dev.mysql.com/downloads/installer/ 下载软件以安装Windows下所有MySQL产品的社区版。选择以下MySQL安装程序包选项之一。

* 网络。仅包含MySQL安装程序和配置文件。网络包选项只下载你选择安装的MySQL产品，但每次下载都需要互联网连接。该文件的大小约为2MB。文件名的形式是mysql-installer-community-web-VERSION.N.msi，其中VERSION是MySQL服务器版本号，如8.0，N是包号，从0开始。
* 完整或当前捆绑。捆绑了所有用于Windows的MySQL产品（包括MySQL服务器）。文件大小超过300MB，名称具有mysql-installer-community-VERSION.N.msi的形式，其中VERSION是MySQL服务器版本号，如8.0，N是包号，从0开始。

**MySQL安装程序商业版**

从https://edelivery.oracle.com/ 下载软件以安装Windows版MySQL产品的商业版（标准版或企业版）。如果你登录了你的My Oracle Support（MOS）帐户，商业版包括社区版中的所有当前和以前的GA版本，但它不包括开发里程碑版本。当你没有登录时，你只能看到你已经下载的捆绑产品的列表。

商业版还包括以下产品。

* Workbench SE/EE
* MySQL企业备份
* MySQL企业防火墙

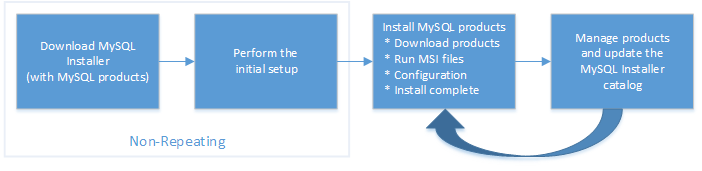
商业版与你的MOS账户集成。有关知识库内容和补丁，请参见My Oracle支持

#### 2.3.3.1 MySQL安装程序初始设置

* 选择一个安装类型
* 路径冲突
* 检查要求
* MySQL安装器配置文件

当你第一次下载MySQL Installer时，一个设置向导会引导你完成MySQL产品的初始安装。如下图所示，初始设置是整个过程中的一个一次性活动。在初始设置期间，MySQL Installer会检测到主机上安装的现有MySQL产品，并将它们添加到要管理的产品列表中。

**Figure 2.7 MySQL Installer Process Overview**



在初始设置期间，MySQL安装程序会提取配置文件（稍后描述）到主机的硬盘上。尽管MySQL安装程序是一个32位的应用程序，但它可以安装32位和64位的二进制文件。

初始设置会在 "开始 "菜单的MySQL组下添加一个链接。点击开始、所有程序、MySQL、MySQL安装器以打开MySQL安装器。

**选择一个安装类型**

在初始设置期间，会提示你选择要在主机上安装的MySQL产品。一种选择是使用符合你的设置要求的预先确定的设置类型。默认情况下，GA和预发布产品都包括在下载和安装的开发者默认、仅客户端和完整设置类型中。选择 "仅安装GA产品 "选项，以限制产品集，使其在使用这些设置类型时仅包括GA产品。

选择以下设置类型之一仅决定初始安装，不限制你以后为Windows安装或更新MySQL产品的能力。

* 开发人员默认设置：安装以下产品，这些产品可用于MySQL的应用开发。
  + MySQL服务器（安装你在下载MySQL安装程序时选择的版本。
  + MySQL Shell
  + MySQL Router
  + MySQL工作台（MySQL Workbench
  + 用于Visual Studio的MySQL
  + MySQL连接器（用于.NET / Python / ODBC / Java / C++）。
  + MySQL文档
  + MySQL样本和例子
* 仅限服务器。只安装MySQL服务器。这种设置类型会安装你在下载MySQL安装程序时选择的一般可用性（GA）或开发版本服务器。它使用默认的安装和数据路径。
* 仅客户端。只安装最新的MySQL应用程序和MySQL连接器。这种设置类型类似于开发者默认类型，除了它不包括MySQL服务器或通常与服务器捆绑的客户端程序，如mysql或mysqladmin。
* 完整：安装所有可用的MySQL产品。
* 自定义。自定义设置类型使你能够从MySQL安装程序目录中过滤和选择单个MySQL产品。

注意事项

对于MySQL Server 8.0.20（和更早的版本）、5.7和5.6版本，你用来运行MySQL Installer的账户可能没有足够的权限来安装服务器数据文件，这可能会中断安装，因为ExecSecureObjects MSI动作不能被执行。要继续进行，在尝试再次安装服务器之前取消选择服务器数据文件功能。如需帮助，请参阅《要安装的产品功能》）。)

服务器数据文件复选框已从MySQL Server 8.0.21（及更高版本）的功能树中移除。

使用自定义设置类型进行安装。

* 一个产品或产品版本无法从通常的下载位置获得。该目录包含所有产品版本，包括预发布（或开发）和GA之间的其他版本。
* 一个使用替代安装路径、数据路径或两者的MySQL服务器实例。关于如何调整路径的说明，见第2.3.3.2节 "用MySQL安装程序设置替代服务器路径"。
* 同一主机上同时有两个或多个MySQL服务器版本（例如，5.6、5.7和8.0）。
* 不作为预定设置类型提供的产品和功能的特定组合。例如，你可以安装单一产品，如MySQL Workbench，而不是安装Windows的所有客户端应用程序。

**路径冲突**

当要安装的产品的默认安装或数据文件夹（MySQL服务器要求的）已经存在于主机上时，向导会显示路径冲突步骤，以识别每个冲突，并使你能够采取行动，避免现有文件夹中的文件被新的安装覆盖。只有当MySQL安装程序检测到冲突时，你才会在初始设置中看到这个步骤。

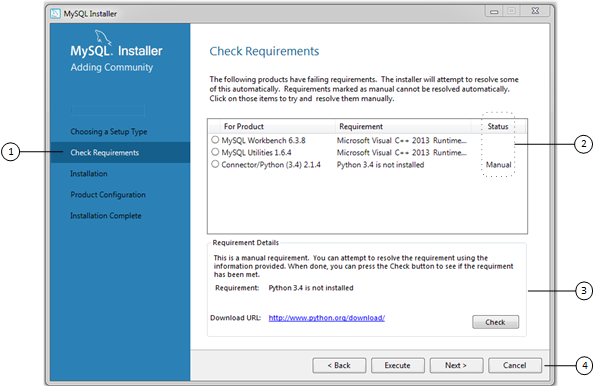
要解决路径冲突，请执行以下操作之一。

* 从列表中选择一个产品以显示冲突选项。一个警告符号表明哪个路径有冲突。使用浏览按钮选择一个新的路径，然后点击下一步。
* 点击 "返回"，选择不同的设置类型或产品版本（如果适用）。自定义设置类型使你能够选择单个产品版本。
* 点击 "下一步 "忽略冲突，覆盖现有文件夹中的文件。
* 删除现有产品。单击 "取消 "以停止初始设置并关闭MySQL安装程序。从 "开始 "菜单再次打开MySQL Installer，并使用MySQL Installer仪表板上的删除操作从主机上删除已安装的产品。

**检查要求**

MySQL Installer使用package-rules.xml文件中的条目来确定每个产品的先决条件软件是否安装在主机上。当需求检查失败时，MySQL安装程序显示检查需求步骤以帮助你更新主机。每次你下载一个新产品（或版本）进行安装时，都会对需求进行评估。下图识别并描述了该步骤的关键区域。

**Figure 2.8 Check Requirements**



检查要求元素的描述

1、显示初始设置中的当前步骤。此列表中的步骤可能会略有变化，这取决于主机上已经安装的产品、先决条件软件的可用性以及将在主机上安装的产品。

2、按产品列出所有待安装的要求，并表示状态如下。

* 状态栏中的空白表示MySQL安装程序可以尝试为你下载和安装所需软件。
* 状态栏中的单词Manual意味着你必须手动满足要求。选择列表中的每个产品以查看其需求细节。

3、详细描述了需求，以帮助你完成每一个手动解决。如果可能的话，会提供一个下载的URL。在你下载和安装所需的软件后，点击检查来验证需求是否得到满足。

4、提供以下设定的操作来进行。

* 返回 - 返回到上一步。该操作使你能够选择不同的设置类型。
* 执行 - 让MySQL安装程序尝试下载和安装所有没有手动状态的项目的所需软件。手动要求由你解决，并通过点击检查进行验证。
* 下一步 - 不执行自动应用要求的请求，继续安装，不包括检查要求步骤失败的产品。
* 取消 - 停止MySQL产品的安装。因为MySQL Installer已经安装好了，所以当你从开始菜单中打开MySQL Installer并从仪表板中点击添加时，初始设置会重新开始。关于可用的管理操作的描述，请参阅产品目录。

**MySQL安装器配置文件**

所有MySQL安装器文件都位于C:\Program Files（x86）和C:\ProgramData文件夹内。下表描述了将MySQL Installer定义为一个独立应用程序的文件和文件夹。

**注意**

当你更新或卸载MySQL Installer时，已安装的MySQL产品不会被改变或删除。

**Table 2.5 MySQL Installer Configuration Files**

|  |  |  |
| --- | --- | --- |
| **File or Folder** | **Description** | **Folder Hierarchy** |
| MySQL Installer for Windows | This folder contains all of the files needed to run MySQL Installer and [MySQLInstallerConsole.exe](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#MySQLInstallerConsole), a command-line program with similar functionality. | C:\Program Files (x86) |
| Templates | The Templates folder has one file for each version of MySQL server. Template files contain keys and formulas to calculate some values dynamically. | C:\ProgramData\MySQL\MySQL Installer for Windows\Manifest |
| package-rules.xml | This file contains the prerequisites for every product to be installed. | C:\ProgramData\MySQL\MySQL Installer for Windows\Manifest |
| produts.xml | The products file (or product catalog) contains a list of all products available for download. | C:\ProgramData\MySQL\MySQL Installer for Windows\Manifest |
| Product Cache | The Product Cache folder contains all standalone .msi files bundled with the full package or downloaded afterward. | C:\ProgramData\MySQL\MySQL Installer for Windows |

#### 2.3.3.2 用MySQL安装程序设置备选服务器路径

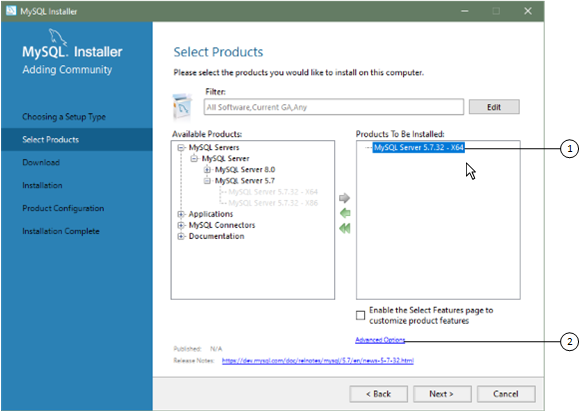
当你安装MySQL服务器时，你可以改变默认安装路径、数据路径或两者。在你安装了服务器后，如果不删除并重新安装服务器实例，就不能改变路径。

**要改变MySQL服务器的路径**

1. 识别要改变的MySQL服务器，并启用高级选项链接。
   1. 通过以下操作之一导航到 "选择产品 "页面。
      1. 如果这是MySQL Installer的初始设置，选择自定义设置类型并点击下一步。
      2. 如果已经安装了MySQL Installer，从 "开始 "菜单中启动它，然后从仪表板上点击添加。
   2. 单击 "编辑"，在 "可用产品 "中显示的产品列表上应用一个过滤器（见 "定位要安装的产品"）。
   3. 选定服务器实例后，使用箭头将选定的服务器移到待安装的产品列表中。
   4. 点击服务器来选择它。当你选择服务器时，高级选项链接在待安装产品列表下面被启用（见下图）。

2、点击 "高级选项"，打开一个对话框，你可以输入其他路径名称。在路径名称被验证后，点击下一步继续配置步骤。

图2.9 更改MySQL服务器路径



#### 2.3.3.3 使用MySQL安装器的安装工作流程

MySQL安装器提供了一个类似向导的工具来安装和配置Windows下的新MySQL产品。与只运行一次的初始设置不同，MySQL安装器在每次下载或安装新产品时都会调用该向导。对于首次安装，初始设置的步骤直接进入安装的步骤。有关产品选择的帮助，请参见查找要安装的产品。

注意

执行MySQL安装程序的用户被授予所有生成文件的全部权限，如my.ini。这不适用于特定产品的文件和目录，例如由SYSTEM拥有的位于%ProgramData%的MySQL服务器数据目录。

在主机上安装和配置的产品遵循一个一般的模式，在各个步骤中可能需要你的输入。如果你试图安装一个与现有MySQL服务器版本（或为升级而选择的版本）不兼容的产品，你会被提醒可能出现不匹配。

* MySQL安装程序提供了以下适用于不同工作流程的操作顺序。选择产品。 如果你在初始设置期间选择了自定义设置类型，或从MySQL安装器仪表板点击了添加，MySQL安装器在侧边栏中包括这一操作。从这个页面，你可以应用一个过滤器来修改可用产品列表，然后选择一个或多个产品来移动（使用箭头键）到要安装的产品列表中。

选择本页的复选框，激活 "选择功能 "动作，你可以在产品下载后自定义产品的功能。

* 下载。 如果你安装了完整的（非网络）MySQL安装包，所有的.msi文件在初始设置期间被加载到产品缓存文件夹中，不会再次下载。否则，单击 "执行 "以开始下载。每个产品的状态会从 "准备下载 "变为 "正在下载"，然后变为 "已下载"。
* 选择要安装的功能（默认为禁用）。 在MySQL安装程序下载产品的.msi文件后，如果你之前在选择产品操作期间启用了可选复选框，你可以自定义功能。
* 要在安装后定制产品功能，请在MySQL安装程序仪表板上单击 "修改"。
* 安装。 列表中每个产品的状态会从准备安装到安装，最后到完成。在这个过程中，单击 "显示细节 "以查看安装操作。

如果你在这时取消安装，产品已经安装，但服务器（如果已安装）还没有配置。要重新启动服务器配置，从 "开始 "菜单中打开MySQL安装程序，在仪表板中相应的服务器旁边点击重新配置

* 产品配置。 此步骤仅适用于MySQL服务器、MySQL路由器和样本。列表中每个项目的状态应显示准备配置。点击 "下一步"，启动列表中所有项目的配置向导。这一步中出现的配置选项是针对你选择安装的数据库或路由器的版本。

点击 "执行 "开始应用配置选项，或者点击 "返回"（重复）返回到每个配置页面。

* 安装完成。 对于不需要配置的产品，这一步最终完成了安装。它使你能够将日志复制到剪贴板并启动某些应用程序，如MySQL Workbench和MySQL Shell。单击 "完成 "以打开MySQL安装程序仪表板。

##### 2.3.3.3.1 用MySQL安装程序配置MySQL服务器

MySQL安装程序执行MySQL服务器的初始配置。例如。

* 它创建配置文件（my.ini），用于配置MySQL服务器。写入该文件的值受你在安装过程中做出的选择的影响。有些定义与主机有关。例如，如果主机少于三个核心，则启用query\_cache。

注意

查询缓存在MySQL 5.7中被废弃，并在MySQL 8.0（及更高版本）中被删除。

* 默认情况下，为MySQL服务器添加了一个Windows服务。
* 为MySQL服务器提供默认的安装和数据路径。关于如何改变默认路径的说明，见第2.3.3.2节，"用MySQL安装器设置替代服务器路径"。
* 它可以选择性地创建具有基于一般角色的可配置权限的MySQL服务器用户账户，如DB管理员、DB设计师和备份管理员。它可以选择性地创建一个名为MysqlSys的具有有限权限的Windows用户，然后由其运行MySQL服务器。

用户账户也可以在MySQL Workbench中添加和配置。

* 勾选 "显示高级选项 "可以设置额外的日志选项。这包括为错误日志、一般日志、慢速查询日志（包括配置执行查询所需的秒数）和二进制日志定义自定义文件路径。

在配置过程中，单击 "下一步 "进入下一个步骤，或单击 "返回 "返回到上一个步骤。在最后一步单击 "执行 "以应用服务器配置。

下面的章节描述了适用于Windows上MySQL服务器的服务器配置选项。你安装的服务器版本将决定你可以配置哪些步骤和选项。配置MySQL服务器可能包括部分或全部步骤。

###### 2.3.3.3.1.1 类型和网络

* 服务器配置类型

选择描述你的设置的MySQL服务器配置类型。该设置定义了分配给你的MySQL服务器实例的系统资源（内存）的数量。

* 开发。一台承载了许多其他应用程序的计算机，通常这是你的个人工作站。此设置将MySQL配置为使用最少的内存量。
* 服务器。预计其他几个应用程序将在这台计算机上运行，如Web服务器。服务器设置将MySQL配置为使用中等数量的内存。
* 专用的。一台专门用于运行MySQL服务器的计算机。因为没有其他主要应用程序在此服务器上运行，此设置将MySQL配置为使用大部分可用内存。
* 连接性

连接性选项控制与MySQL的连接方式。选项包括。

* TCP/IP：默认情况下，该选项被选中。你可以禁用TCP/IP网络，只允许本地主机连接。在选择了TCP/IP连接选项后，你可以修改以下项目。
* 经典MySQL协议连接的端口。默认值为3306。
* 仅在配置MySQL 8.0服务器时显示的X协议端口。默认值是33060
* 用于网络访问的开放的Windows防火墙端口，默认为TCP/IP连接所选。

如果一个端口号已经在使用，你会看到默认值旁边的信息图标（），并且下一步被禁用，直到你提供一个新的端口号。

* 命名的管道。启用并定义管道名称，类似于设置named\_pipe系统变量。默认名称是MySQL。
* 共享内存。启用并定义内存名称，类似于设置shared\_memory系统变量。默认名称是MySQL。
* 高级配置

勾选显示高级和日志选项，以便在后面的步骤中设置自定义日志和高级选项。日志选项步骤使你能够为错误日志、一般日志、慢速查询日志（包括其执行查询所需的秒数配置）和二进制日志定义自定义文件路径。高级选项步骤使你能够设置在复制拓扑中启用二进制日志时需要的唯一服务器ID。

* MySQL企业防火墙（仅企业版

启用MySQL企业防火墙复选框在默认情况下未被选中。选择此选项可启用安全列表，提供对某些类型攻击的保护。需要额外的安装后配置（见第6.4.7节，"MySQL企业防火墙"）。

重要提示

MySQL 8.0.19有一个问题，如果在服务器配置步骤中选择了MySQL Enterprise Firewall，则服务器无法启动。如果服务器启动操作失败，单击 "取消 "以结束配置过程并返回仪表板。你必须卸载服务器。

解决办法是在不选择MySQL企业防火墙的情况下运行MySQL安装程序。(也就是说，不要选择启用MySQL企业防火墙复选框。)然后使用手动安装的说明在之后安装MySQL企业防火墙(见第6.4.7.2节，"安装或卸载MySQL企业防火墙")。

###### 2.3.3.3.1.2 认证方法

认证方法步骤仅在安装或升级MySQL 8.0.4或更高版本时可见。它引入了两个服务器端认证选项的选择。你在下一步创建的MySQL用户账户将使用你在该步骤中选择的认证方法。

使用libmysqlclient 8.0的MySQL 8.0连接器和社区驱动程序现在支持mysql\_native\_password默认认证插件。然而，如果你无法更新你的客户端和应用程序以支持这种新的认证方法，你可以配置MySQL服务器以使用mysql\_native\_password进行传统认证。有关这一变化的影响的更多信息，请参阅缓存\_sha2\_password作为首选认证插件。

如果你正在安装或升级到MySQL 8.0.4或更高版本，请选择以下认证方法之一。

* 使用强密码加密认证（推荐）。

MySQL 8.0支持基于改进的、更强大的基于SHA256的密码方法的新认证。建议所有新安装的MySQL服务器今后都使用这种方法。

重要提示

服务器上的caching\_sha2\_password认证插件需要新版本的连接器和客户端，它们增加了对新的MySQL 8.0默认认证的支持。

* 使用传统的认证方法（保留MySQL 5.x的兼容性

使用旧的MySQL 5.x遗留认证方法应仅在以下情况下考虑。

* 应用程序不能被更新以使用MySQL 8.0连接器和驱动程序
* 重新编译一个现有的应用程序是不可行的。
* 更新的、特定于语言的连接器或驱动程序尚不可用。

###### 2.3.3.3.1.3 账户和角色

* 根账户密码

指定一个根密码是必需的，在执行其他MySQL安装程序操作时，你将被要求提供该密码。当你在提供的方框中重复密码时，将评估密码强度。关于密码要求或状态的描述性信息，当信息图标（）出现时，将鼠标指针移到它上面。

* MySQL用户账户（可选

单击 "添加用户 "或 "编辑用户 "来创建或修改具有预定义角色的MySQL用户账户。接下来，输入所需的账户凭证。

* 用户名。MySQL用户名最多可以有32个字符。
* 主机。仅为本地连接选择localhost，或者当需要远程连接到服务器时选择<所有主机(%)>。
* 角色。每个预定义的角色，如DB管理员，都配置有自己的一套权限。例如，DB Admin角色比DB Designer角色有更多的权限。角色下拉列表包含每个角色的描述。
* 密码：当你输入密码时，会进行密码强度评估。密码必须被确认。MySQL允许空白或空密码（被认为是不安全的）。

仅限MySQL安装程序商业版。 用于Windows的MySQL企业版是一种商业产品，它也支持在Windows上执行外部认证的认证方法。由Windows操作系统认证的账户可以访问MySQL服务器，而无需提供额外的密码。

要创建一个使用Windows认证的新MySQL账户，请输入用户名，然后为主机和角色选择一个值。单击Windows认证以启用authentication\_windows插件。在Windows安全令牌区域，为每个可以用MySQL用户名认证的Windows用户（或组）输入一个令牌。MySQL账户可以包括本地Windows用户和属于一个域的Windows用户的安全令牌。多个安全令牌由分号（;）分隔，并对本地和域账户使用以下格式。

* 本地帐户

输入简单的Windows用户名作为每个本地用户或组的安全令牌；例如，finley;jeffrey;admin。

* 域账户

使用标准的Windows语法（domain/domainuser）或MySQL语法（domain/domainuser）来输入Windows域用户和组。

对于域账户，如果运行MySQL安装程序的账户缺乏查询活动目录的权限，你可能需要使用域内管理员的凭证。如果是这种情况，选择验证活动目录用户，以激活域管理员的凭证。

Windows认证允许你在每次添加或修改令牌时测试所有的安全令牌。点击测试安全令牌来验证（或重新验证）每个令牌。无效的令牌会产生一个描述性的错误信息，以及一个红色的X图标和红色的令牌文本。当所有令牌解决为有效时（绿色文本没有X图标），你可以点击确定来保存更改。

###### 2.3.3.3.1.4 Windows服务

在Windows平台上，MySQL服务器可以作为一个由操作系统管理的命名服务运行，并被配置为在Windows启动时自动启动。另外，你也可以将MySQL服务器配置为需要手动配置的可执行程序来运行。

* 将MySQL服务器配置为一个Windows服务（默认选择。

当选择默认配置选项时，你还可以选择以下内容。

* 在系统启动时启动MySQL服务器

当选择时（默认），服务启动类型被设置为自动；否则，启动类型被设置为手动。

* 运行Windows服务为

当选择标准系统账户时（默认），服务作为网络服务登录。

自定义用户选项必须有权限以服务方式登录Windows系统。下一步 "按钮将被禁用，直到该用户被配置为所需权限。

自定义用户账户是通过在开始菜单中搜索 "本地安全策略 "在Windows中配置的。在 "本地安全策略 "窗口中，选择 "本地策略"、"用户权限分配"，然后选择 "作为服务登录 "来打开属性对话框。单击 "添加用户或组 "以添加自定义用户，然后在每个对话框中单击 "确定 "以保存更改。

* 取消选择Windows服务选项

###### 2.3.3.3.1.5 日志选项

如果在类型和网络步骤中选择了显示高级配置复选框，则此步骤可用。要现在启用这个步骤，请单击 "返回 "返回到 "类型和网络 "步骤，并选择该复选框。

高级配置选项与下列MySQL日志文件有关。

* 错误日志
* 一般日志
* 慢速查询日志
* 仓库日志

注意

对于MySQL 5.7及以上版本，二进制日志是默认启用的。

###### 2.3.3.3.1.6 高级选项

如果在类型和网络步骤中选择了显示高级配置复选框，就可以使用这个步骤。要现在启用这个步骤，请单击 "返回 "返回到 "类型和网络 "步骤，并选择复选框。

高级配置选项包括。

* 服务器ID

设置复制拓扑结构中使用的唯一标识符。如果启用了二进制日志，你必须指定一个服务器ID。默认的ID值取决于服务器的版本。欲了解更多信息，请参见server\_id系统变量的描述。

* 表名案例

你可以在服务器的初始和后续配置中设置以下选项。对于MySQL 8.0发布系列，这些选项仅适用于服务器的初始配置。

* 小写字母

将lower\_case\_table\_names选项值设置为1（默认），在这种情况下，表名在磁盘上以小写存储，并且比较不区分大小写。

* 保留给定大小写

将lower\_case\_table\_names选项的值设置为2，在这种情况下，表名会按照给定的大小写存储，但是比较时要用小写。

###### 2.3.3.3.1.7 应用服务器配置

当你点击执行时，所有配置设置都会应用到MySQL服务器上。使用 "配置步骤 "选项卡来跟踪每个操作的进展；成功时，每个操作的图标会从白色切换到绿色（有一个复选标记）。否则，如果某项操作超时，进程就会停止并显示错误信息。单击 "日志 "选项卡以查看日志。

当安装成功完成并点击完成时，MySQL安装程序和已安装的MySQL产品被添加到Microsoft Windows开始菜单的MySQL组下。打开MySQL安装程序会加载仪表板，其中列出了已安装的MySQL产品，并提供了其他MySQL安装程序操作。

##### 2.3.3.3.2 用MySQL安装程序配置MySQL路由器

MySQL安装程序下载并安装了一套用于在Windows上开发和管理关键业务应用程序的工具。该套件由应用程序、连接器、文档和样本组成。

在初始设置期间，选择任何预先确定的设置类型，除了仅服务器，以安装工具的最新GA版本。使用自定义设置类型来安装单个工具或特定版本。如果MySQL Installer已经安装在主机上，使用添加操作从MySQL Installer仪表板上选择并安装工具。

MySQL路由器配置

MySQL安装程序提供了一个配置向导，可以启动已安装的MySQL Router 8.0实例，以引导MySQL应用程序和InnoDB簇之间的流量。当被配置时，MySQL Router作为一个本地Windows服务运行。

注意

在初始安装后，以及当你明确地重新配置已安装的路由器时，会提示你配置MySQL Router。相反，升级操作不要求或提示你配置升级后的产品。

要配置MySQL Router，请执行以下操作。

1、设置InnoDB群集。

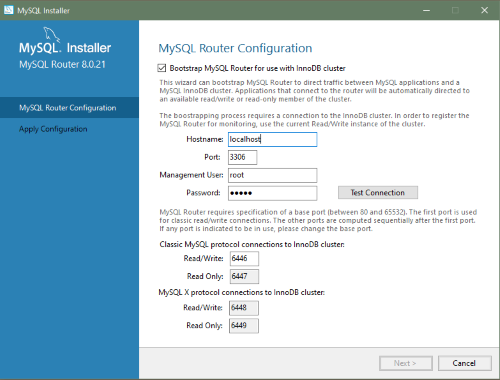
2、使用MySQL安装程序，下载并安装MySQL Router应用程序。安装完成后，配置向导会提示你提供信息。选择为InnoDB簇配置MySQL路由器复选框以开始配置，并提供以下配置值。

* 主机名：InnoDB簇中主（种子）服务器的主机名（默认为localhost）。
* 端口：InnoDB集群中的主（种子）服务器的端口号（默认为3306）。
* 管理用户：具有root级别权限的管理用户。
* 密码：管理用户的密码。
* 与InnoDB Cluster的经典MySQL协议连接

读/写。将第一个基本端口号设置为一个未使用的端口（在80和65532之间），向导将为你选择其余的端口。

下面的图显示了MySQL路由器配置页面的一个例子，第一个基本端口号被指定为6446，其余端口被向导设置为6447、6448和6449。

图2.10 MySQL路由器配置



3、点击 "下一步"，然后点击 "执行 "以应用配置。单击 "完成 "关闭MySQL安装程序或返回到MySQL安装程序仪表板。

在配置MySQL Router之后，根账户在用户表中只作为root@localhost（本地）存在，而不是root@%（远程）。无论路由器和客户端位于何处，即使两者都位于与种子服务器相同的主机上，任何通过路由器的连接都被服务器视为远程，而不是本地。因此，使用本地主机与服务器建立的连接（见下面的例子），不会被认证。

shell> **\c root@localhost:6446**

#### 2.3.3.4 MySQL安装程序产品目录和仪表板

本节介绍MySQL安装程序的产品目录、仪表板以及与产品选择和升级有关的其他操作。

* 产品目录
* MySQL安装器仪表板
* 找到要安装的产品
* 升级MySQL服务器
* 移除MySQL服务器
* 升级MySQL安装程序

**产品目录**

产品目录存储了可从MySQL下载的用于Microsoft Windows的MySQL发布产品的完整列表。默认情况下，当有互联网连接时，MySQL安装程序会尝试每七天在启动时更新目录。你也可以从仪表板上手动更新目录（稍后描述）。

一个最新的目录会执行以下操作。

* 填充 "选择产品 "页面的 "可用产品 "窗格。这个步骤在你选择时出现。
  + 在初始设置中选择自定义设置类型。
  + 从仪表板上的添加操作。
* 确定仪表盘中列出的已安装产品的产品更新何时可用。

该目录包括所有开发版本（Pre-Release）、一般版本（Current GA）和次要版本（Other Releases）。目录中的产品会有些变化，这取决于你下载的MySQL安装程序版本。

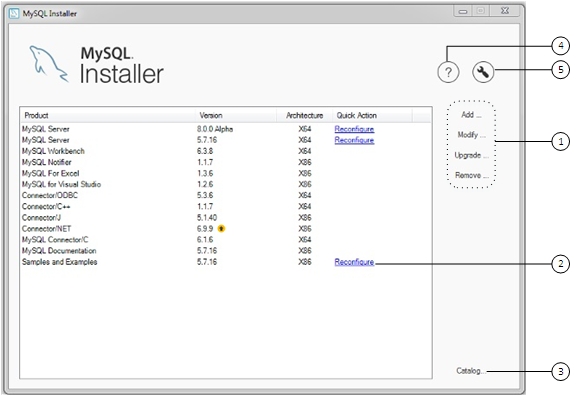
**MySQL安装器仪表板**

MySQL安装器仪表板是初始设置完成后启动MySQL安装器时看到的默认视图。如果你在设置完成之前关闭了MySQL安装器，MySQL安装器在显示仪表板之前会恢复初始设置。

注意

如果安装了Oracle终身持续支持下的产品，可能会出现在仪表板中。这些产品，如MySQL for Excel和MySQL Notifier，只能被修改或删除。

图2.11 MySQL安装器仪表板元素



MySQL安装器仪表板元素的描述

1、MySQL安装程序仪表板操作提供了各种适用于已安装产品或目录中所列产品的操作。要启动以下操作，首先点击操作链接，然后选择要管理的产品或产品。

* 添加。此操作打开 "选择产品 "页面。在那里你可以调整过滤器，选择一个或多个产品来下载（根据需要），并开始安装。关于使用过滤器的提示，请看定位要安装的产品。

使用方向箭头将每个产品从 "可用产品 "栏移到 "待安装产品 "栏。要启用你可以定制功能的 "产品功能 "页面，请点击相关的复选框（默认为禁用）。

注意

对于MySQL Server 8.0.20（和更早的版本）、5.7和5.6版本，你用来运行MySQL Installer的账户可能没有足够的权限来安装服务器数据文件，这可能会中断安装，因为ExecSecureObjects MSI动作不能被执行。要继续进行，在尝试再次安装服务器之前取消选择服务器数据文件功能。

服务器数据文件复选框已从MySQL Server 8.0.21（或更高版本）的特征树中删除。

* 修改。使用此操作来添加或删除与已安装产品相关的功能。你可以修改的功能因产品的复杂性而异。当选择了程序快捷方式复选框时，该产品会出现在开始菜单的MySQL组下。
* 升级：该操作会加载 "选择要升级的产品 "页面，并将所有升级候选产品填充到该页面中。一个已安装的产品可以有一个以上的升级版本，该操作需要一个当前的产品目录。MySQL安装程序在一个操作中升级所有选定的产品。单击 "显示详情 "以查看MySQL安装器执行的操作。
* 移除。此操作打开 "移除产品 "页面，并以安装在主机上的MySQL产品填充它。选择你要删除（卸载）的MySQL产品，然后点击执行，开始删除过程。在操作过程中，有一个指标显示执行的步骤数占所有步骤的百分比。

要选择要删除的产品，请执行以下操作之一。

* 选择一个或多个产品的复选框。
* 选择产品复选框来选择所有产品。

2、在每个已安装的服务器旁边的 "快速行动 "栏中的 "重新配置 "链接加载服务器的当前配置值，然后循环执行所有配置步骤，使你能够改变选项和值。你必须提供具有root权限的凭证来重新配置这些项目。单击 "日志 "选项卡，显示MySQL Installer执行的每个配置步骤的输出。

完成后，MySQL安装程序会停止服务器，应用配置更改，并为你重新启动服务器。关于每个配置选项的描述，请参阅第2.3.3.3.1节 "用MySQL安装程序配置MySQL服务器"。与特定MySQL服务器版本相关的已安装样本和示例也可以被重新配置以应用新的功能设置（如果有的话）。

3、目录链接使你能够手动下载MySQL产品的最新目录，然后将这些产品变化与MySQL安装程序集成。目录下载操作并不对已经安装在主机上的产品进行升级。相反，它返回到仪表板，并为每个具有较新版本的已安装产品在版本栏中添加一个箭头图标。使用升级操作来安装较新的产品版本。

你也可以使用目录链接来显示每个产品的当前变化历史，而不下载新的目录。选择 "此时不更新 "复选框，只查看更改历史。

4、MySQL安装器关于图标（）显示MySQL安装器的当前版本和关于MySQL的一般信息。版本号位于 "返回 "按钮的上方。

提示

在报告MySQL安装器的问题时，总是包括这个版本号。

除了关于MySQL的信息（），你还可以从侧板上选择以下图标。

* MySQL安装器的许可证图标（）。

本产品可能包括第三方软件，在许可证下使用。如果你使用的是MySQL Installer的商业版本，该图标将打开MySQL Installer商业许可信息用户手册，以获取许可信息，包括与可能包含在该商业版本中的第三方软件有关的许可信息。如果你使用的是MySQL Installer的社区版本，该图标将打开MySQL Installer社区许可信息用户手册，以获得许可信息，包括与可能包含在该社区版本中的第三方软件有关的许可信息。

* 资源链接 图标（）指向最新的MySQL产品文档、博客、网络研讨会等。

5、MySQL安装程序选项图标（）包括以下标签。

* 产品目录。管理自动目录更新。默认情况下，MySQL安装程序每七天在启动时检查目录更新。当新产品或产品版本可用时，MySQL安装程序将它们添加到目录中，然后在仪表板中列出的已安装产品的版本号旁边插入一个箭头图标（）。

使用产品目录选项来启用或禁用自动更新，并重置自动下载目录之间的天数。在启动时，MySQL安装程序使用你设置的天数来决定是否应尝试下载。如果MySQL Installer在下载目录时遇到错误，在下次启动时将重复这一操作。

* 连接性设置。由MySQL安装程序执行的一些操作需要互联网访问。这个选项使你能够使用一个默认值来验证连接，或使用一个不同的URL，一个从列表中选择的或由你手动添加的URL。在选择手动选项时，可以添加新的URL，并且可以移动或删除列表中的所有URL。当选择自动选项时，MySQL安装程序会尝试连接到列表中的每个默认URL（按顺序），直到建立连接。如果不能建立连接，它将引发一个错误。

**找到要安装的产品**

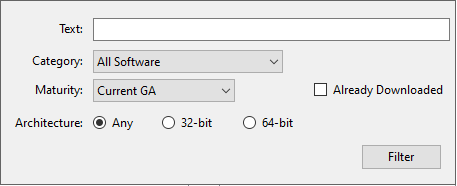
目录中的MySQL产品按类别列出。MySQL服务器、应用程序、MySQL连接器和文档。默认情况下，只有最新的GA版本出现在可用产品窗格中。如果你正在寻找某个产品的预发布版本或旧版本，它可能在默认列表中不可见。

注意

保持产品目录是最新的。单击MySQL安装程序仪表板上的目录以下载最新的清单。

要改变默认的产品列表，在仪表板上点击添加以打开 "选择产品 "页面，然后点击编辑以打开下图所示的对话框。修改设置，然后点击过滤。

图2.12 筛选可用产品



重置以下一个或多个字段来修改可用产品的列表。

* 文本。按文本过滤。
* 种类：通过文字过滤。所有软件（默认）、MySQL服务器、应用程序、MySQL连接器或文档（用于样本和文档）。
* 成熟度。当前的捆绑（最初只出现在完整的软件包中），预发布，当前的GA，或其他版本。如果你看到一个警告，通过点击MySQL安装程序仪表板上的目录，确认你有最新的产品清单。如果MySQL安装程序无法下载清单，你看到的产品范围仅限于捆绑的产品、已经位于产品缓存文件夹中的独立产品MSI，或两者。

注意

当你选择Pre-Release成熟度过滤器时，MySQL Installer的商业版本不显示任何MySQL产品。开发中的产品只能从MySQL Installer的社区版中获得。

* 已经下载（该复选框在默认情况下是取消选择的）。只允许你查看和管理已下载的产品。
* 架构。任何（默认）、32位或64位。

**升级MySQL服务器**

重要的服务器升级条件。

* MySQL安装程序不允许在主要发布版本或次要发布版本之间进行服务器升级，但允许在一个发布系列内进行升级，例如从5.7.18升级到5.7.19。
* 不支持里程碑版本之间的升级（或从一个里程碑版本到GA版本）。在里程碑版本中会发生重大的开发变化，你可能会遇到兼容性问题或启动服务器的问题。
* 对于升级到MySQL 8.0.16服务器及更高版本，一个复选框使你能够跳过系统表的升级检查和处理，而正常检查和处理数据字典表。当以前的服务器升级被跳过或服务器被配置为沙盒InnoDB簇时，MySQL安装程序不会用该复选框提示你。这种行为代表了MySQL服务器执行升级的方式的变化（参见第2.11.3节，"MySQL升级过程升级的内容"），它改变了MySQL安装程序应用于配置过程的步骤顺序。

如果你选择跳过系统表升级检查和过程。不推荐），MySQL安装程序将以--upgrade=MINIMAL服务器选项启动升级后的服务器，它只升级数据字典。如果你停止，然后重新启动服务器而不使用--upgrade=MINIMAL选项，如果需要，服务器会自动升级系统表。

在升级配置（跳过系统表）完成后，以下信息出现在日志选项卡和日志文件中。

WARNING: The system tables upgrade was skipped after upgrading MySQL Server. The

server will be started now with the --upgrade=MINIMAL option, but then each

time the server is started it will attempt to upgrade the system tables, unless

you modify the Windows service (command line) to add --upgrade=MINIMAL to bypass

the upgrade.

FOR THE BEST RESULTS: Run mysqld.exe --upgrade=FORCE on the command line to upgrade

the system tables manually.

要选择一个新的服务器版本。

1、点击升级。确认 "可升级产品 "窗格中的产品名称旁边的复选框有一个复选标记。取消选择你不打算在这个时候升级的产品。

注意

对于同一发行系列中的服务器里程碑版本，MySQL安装程序会取消选择服务器升级，并显示警告以表明不支持该升级，确定继续下去的风险，并提供手动执行逻辑升级的步骤摘要。你可以重新选择服务器升级，风险由你自己承担。关于如何用一个里程碑版本执行逻辑升级的说明，请看逻辑升级。

2、单击列表中的一个产品以突出显示它。此操作会在可升级版本窗格中填入所选产品的每个可用版本的详细信息：版本号、发布日期以及打开该版本的发行说明的更改链接。

**移除MySQL服务器**

要删除一个本地MySQL服务器。

1、确定是否应删除本地数据目录。如果你保留数据目录，另一个服务器安装可以重新使用这些数据。该选项默认为启用（删除数据目录）。

2、点击执行，开始卸载本地服务器。注意，所有你选择删除的产品在这个时候也会被卸载。

3、可选）单击 "日志 "选项卡以显示MySQL Installer当前执行的操作。

**升级MySQL安装器**

MySQL Installer仍然安装在你的计算机上，像其他软件一样，MySQL Installer可以从以前的版本升级。在某些情况下，其他MySQL软件可能需要你升级MySQL安装器以获得兼容性。本节介绍了如何识别MySQL安装器的当前版本以及如何手动升级MySQL安装器。

要找到已安装的MySQL安装器的版本。

1、从搜索菜单中启动MySQL安装器。打开MySQL安装器仪表板。

2、点击MySQL安装器关于图标（）。版本号位于 "返回 "按钮的上方。

要启动MySQL Installer的按需升级。

1. 将安装了MySQL Installer的计算机连接到互联网。
2. 从搜索菜单中启动MySQL安装器。MySQL安装器仪表板打开。
3. 点击仪表板底部的Catalog，打开Update Catalog窗口。
4. 单击 "执行 "以开始该过程。如果已安装的MySQL Installer版本可以升级，你将被提示开始升级。
5. 点击 "下一步 "查看目录的所有更改，然后点击 "完成 "返回仪表板。
6. 验证MySQL Installer的（新）安装版本（见前面的程序）。

#### 2.3.3.5 MySQLInstallerConsole参考

MySQLInstallerConsole.exe提供与MySQL Installer类似的命令行功能。它在最初执行MySQL Installer时被安装，然后在MySQL Installer for Windows目录中可用。默认情况下，它位于C:\Program Files (x86)\MySQL\MySQL Installer for Windows，而且必须以管理权限执行该控制台。

要使用，通过选择 "开始"、"附件 "以管理权限调用命令提示符，然后右键单击命令提示符，选择以管理员身份运行。并从命令行中，有选择地改变MySQLInstallerConsole.exe所在的目录。

C:\> **cd Program Files (x86)\MySQL\MySQL Installer for Windows**

C:\Program Files (x86)\MySQL\MySQL Installer for Windows> **MySQLInstallerConsole.exe help**

=================== Start Initialization ===================

MySQL Installer is running in Community mode

Attempting to update manifest.

Initializing product requirements

Loading product catalog

Checking for product catalog snippets

Checking for product packages in the bundle

Categorizing product catalog

Finding all installed packages.

Your product catalog was last updated at 11/1/2016 4:10:38 PM

=================== End Initialization ===================

The following commands are available:

Configure - Configures one or more of your installed programs.

Help - Provides list of available commands.

Install - Install and configure one or more available MySQL programs.

List - Provides an interactive way to list all products available.

Modify - Modifies the features of installed products.

Remove - Removes one or more products from your system.

Status - Shows the status of all installed products.

Update - Update the current product catalog.

Upgrade - Upgrades one or more of your installed programs.

**MySQL产品名称**

许多MySQLInstallerConsole命令接受一个或多个关键字，代表目录中的MySQL产品（或产品）。当前与命令一起使用的有效关键字集如下表所示。

**Table 2.6 MySQL Product Keywords for MySQLInstallerConsole**

|  |  |
| --- | --- |
| **Keyword** | **MySQL Product** |
| **server** | MySQL Server |
| **workbench** | MySQL Workbench |
| **shell** | MySQL Shell |
| **visual** | MySQL for Visual Studio |
| **router** | MySQL Router |
| **backup** | MySQL Enterprise Backup |
| **net** | MySQL Connector/NET |
| **odbc** | MySQL Connector/ODBC |
| **c++** | MySQL Connector/C++ |
| **python** | MySQL Connector/Python |
| **j** | MySQL Connector/J |
| **documentation** | MySQL Server Documentation |
| **samples** | MySQL Samples (sakila and world databases |

MySQLInstallerConsole命令选项

MySQLInstallerConsole.exe支持下列命令选项。

注意

包含冒号字符（:）的配置块值必须用引号包裹。例如，installdir="C:\MySQL\MySQL Server 8.0"。

* configure [product1]:[setting]=[value]; [product2]:[setting]=[value]; [...]

在你的系统上配置一个或多个MySQL产品。可以为每个产品配置多个设置=值对。

开关包括。

-显示设置

通过在-showsettings后面传入产品名称，显示所选产品的可用选项。

-静音

禁用确认提示。

C:\> **MySQLInstallerConsole configure -showsettings server**

C:\> **MySQLInstallerConsole configure server:port=3307**

* **help *[command]***

显示一个带有使用例子的帮助信息，然后退出。传入一个额外的命令以接收该命令的特定帮助。

C:\> **MySQLInstallerConsole help**

C:\> **MySQLInstallerConsole help install**

* **install [*product*]:[*features*]:[*config block*]:[*config block*]:[*config block*]; [*...*]**

在你的系统上安装一个或多个MySQL产品。如果预发布产品可用，当-type开关的值为Developer、Client或Full时，GA和预发布产品都会安装。使用-only\_ga\_products开关，在使用这些设置类型时，将产品集仅限制为GA产品。

开关和语法选项包括。

-only\_ga\_products

将产品集限制为只包括GA产品。

-type=[SetupType]

安装一个预定义的软件集。设置类型可以是以下之一。

* 开发者。安装一个完整的开发环境。
* 服务器。安装一个单一的MySQL服务器
* 客户端。安装客户端程序和库
* 完全：安装所有的东西
* 自定义。安装用户选择的产品。这是默认选项。

注意

非自定义设置类型只有在没有安装其他MySQL产品时才有效。

**-showsettings**

通过在-showsettings后面传入产品名称，显示所选产品的可用选项。

**-silent**

禁用确认提示。

**[*product*]**

每个产品都可以通过产品关键字指定，也可以不使用分号分隔的版本限定符。单独传入一个产品关键字会选择该产品的最新版本。如果该产品的版本有多个架构，命令会返回清单中的第一个架构，供交互式确认。另外，您可以使用-silent 开关，在产品关键字之后传入准确的版本和架构（x86 或 x64）。

**[*features*]**

与MySQL产品相关的所有特性都是默认安装的。特征块是一个分号分隔的特征列表，或一个星号字符（\*），选择所有的特征。要删除一个特性，请使用修改命令。

**[*config block*]**

可以指定一个或多个配置块。每个配置块是一个由分号分隔的键值对列表。一个块可以包括config或user类型的键；如果没有定义，config是默认类型。

包含冒号字符（:）的配置块值必须用引号包裹。例如，installdir="C:\MySQL\MySQL Server 8.0"。每个产品只能定义一个配置类型块。在产品安装过程中，应该为每个要创建的用户定义一个用户块。

注意

当一个产品被重新配置时，不支持用户类型键。

C:\> **MySQLInstallerConsole install server;5.6.25:\*:port=3307;serverid=2:type=user;username=foo;password=bar;role=DBManager**

C:\> **MySQLInstallerConsole install server;5.6.25;x64 -silent**

一个传入额外配置块的例子，用^分隔以适应。

C:\> **MySQLInstallerConsole install server;5.6.25;x64:\*:type=config;openfirewall=true; ^**

**generallog=true;binlog=true;serverid=3306;enable\_tcpip=true;port=3306;rootpasswd=pass; ^**

**installdir="C:\MySQL\MySQL Server 5.6":type=user;datadir="C:\MySQL\data";username=foo;password=bar;role=DBManager**

**list**

列出一个交互式控制台，在那里可以搜索到所有可用的MySQL产品。执行MySQLInstallerConsole list来启动控制台，并输入子串进行搜索。

C:\> **MySQLInstallerConsole list**

**modify [*product1*:-*removelist*|+*addlist*] [*product2*:-*removelist*|+*addlist*] [*...*]**

修改或显示先前安装的MySQL产品的特性。要显示一个产品的特性，在命令中附加产品关键字，例如。

C:\> **MySQLInstallerConsole modify server**

此命令的语法选项。

-silent

禁用确认提示。

C:\> **MySQLInstallerConsole modify server:+documentation**

C:\> **MySQLInstallerConsole modify server:-debug**

**remove [*product1*] [*product2*] [*...*]**

从你的系统中删除一个或多个产品。开关和语法选项包括。

\*

传入\*以删除所有的MySQL产品。

-continue

即使发生错误也继续操作。

-silent

禁用确认提示。

C:\> **MySQLInstallerConsole remove \***

C:\> **MySQLInstallerConsole remove server**

**status**

提供系统上安装的MySQL产品的快速概览。信息包括产品名称和版本、架构、安装日期和安装位置。

C:\> **MySQLInstallerConsole status**

**update**

将最新的MySQL产品目录下载到你的系统。一旦成功，在下次执行MySQLInstaller或MySQLInstallerConsole时，将应用该目录。

C:\> **MySQLInstallerConsole update**

注意

自动目录更新GUI选项从Windows任务调度器执行此命令。

**upgrade [*product1*:*version*] [*product2*:*version*] [*...*]**

升级您系统中的一个或多个产品。语法选项包括。

\*

传入\*以将所有产品升级到最新版本，或传入特定产品。

!

传入！作为版本号，将MySQL产品升级到其最新版本。

-**-silent**

禁用确认提示。

C:\> **MySQLInstallerConsole upgrade \***

C:\> **MySQLInstallerConsole upgrade workbench:8.0.21**

C:\> **MySQLInstallerConsole upgrade workbench:!**

C:\> **MySQLInstallerConsole upgrade workbench:8.0.21 visual:1.2.9**

### 2.3.4 使用noinstall ZIP档案在Microsoft Windows上安装MySQL

从noinstall软件包安装的用户可以使用本节的说明来手动安装MySQL。从ZIP档案包安装MySQL的过程如下。

1. 提取主压缩包到所需的安装目录

可选的：如果你计划执行MySQL基准和测试套件，也要提取调试-测试存档。

2、创建一个选项文件

3、选择一个MySQL服务器类型

4、初始化MySQL

5、启动MySQL服务器

6、确保默认用户账户的安全

这一过程将在下面的章节中描述。

#### 2.3.4.1 提取安装档案

要手动安装MySQL，请执行以下步骤。

1、如果你从以前的版本升级，在开始升级过程之前，请参考第2.11.10节 "在Windows上升级MySQL"。

2、确保你是以具有管理员权限的用户身份登录的。

3、选择一个安装位置。传统上，MySQL服务器被安装在C:\mysql中。如果你不在C:\mysql安装MySQL，你必须在启动时或在选项文件中指定安装目录的路径。见第2.3.4.2节，"创建选项文件"。

注意

MySQL安装程序将MySQL安装在C:\Program Files\MySQL下。

4、使用你喜欢的文件压缩工具将安装档案提取到所选择的安装位置。有些工具可能会将存档解压到你所选择的安装位置的一个文件夹中。如果发生这种情况，你可以将子文件夹的内容移到选择的安装位置。

#### 2.3.4.2 创建一个选项文件

如果你需要在运行服务器时指定启动选项，你可以在命令行中指出它们，或者把它们放在一个选项文件中。对于每次服务器启动时都要使用的选项，你可能会发现使用一个选项文件来指定你的MySQL配置是最方便的。这在以下情况下尤其如此。

* 安装或数据目录位置与默认位置不同（C:\Program Files\MySQL\MySQL Server 8.0和C:\Program Files\MySQL\MySQL Server 8.0\data）。
* 你需要调整服务器设置，如内存、缓存或InnoDB配置信息。

当MySQL服务器在Windows上启动时，它会在几个地方寻找选项文件，如Windows目录、C:\和MySQL安装目录（关于完整的位置列表，见4.2.2.2节 "使用选项文件"）。Windows目录通常被命名为C:\WINDOWS之类的。你可以使用下面的命令从WINDIR环境变量的值中确定它的确切位置。

C:\> **echo %WINDIR%**

MySQL首先在my.ini文件中寻找每个位置的选项，然后在my.cnf文件中寻找。然而，为了避免混淆，最好只使用一个文件。如果你的电脑使用的是启动加载器，而C：不是启动驱动器，那么你唯一的选择就是使用my.ini文件。无论你使用哪一个选项文件，它都必须是一个纯文本文件。

注意

当使用MySQL安装程序安装MySQL服务器时，它会在默认位置创建my.ini，并且执行MySQL安装程序的用户被授予这个新的my.ini文件的全部权限。

换句话说，要确保MySQL Server用户有阅读my.ini文件的权限。

你也可以利用包括在你的MySQL发行版中的示例选项文件；见第5.1.2节，"服务器配置默认值"。

选项文件可以用任何文本编辑器创建和修改，如记事本。例如，如果MySQL安装在E:\mysql，数据目录在E:\mydata\data，你可以创建一个包含[mysqld]部分的选项文件来指定baseir和datadir的值。

[mysqld]

# set basedir to your installation path

basedir=E:/mysql

# set datadir to the location of your data directory

datadir=E:/mydata/data

Microsoft Windows的路径名称在选项文件中使用（正向）斜线而不是反斜线来指定。如果你使用反斜线，请将其加倍。

[mysqld]

# set basedir to your installation path

basedir=E:\\mysql

# set datadir to the location of your data directory

datadir=E:\\mydata\\data

在选项文件中使用反斜杠的规则在第4.2.2.2节 "使用选项文件 "中给出。

该ZIP档案不包括数据目录。要通过创建数据目录和填充mysql系统数据库中的表来初始化MySQL安装，请使用-initialize或-initialize-insecure初始化MySQL。有关其他信息，见第2.10.1节，"初始化数据目录"。

如果你想在不同的位置使用一个数据目录，你应该把数据目录的全部内容复制到新的位置。例如，如果你想使用E:\mydata作为数据目录，你必须做两件事。

1. 将整个数据目录及其所有内容从默认位置（例如C:\Program Files\MySQL\MySQL Server 8.0data）移动到E:\mydata。
2. 每次启动服务器时，使用--datadir选项来指定新的数据目录位置。

#### 2.3.4.3 选择一个MySQL服务器类型

下表显示了MySQL 8.0中用于Windows的可用服务器。

|  |  |
| --- | --- |
| **Binary** | **Description** |
| [**mysqld**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#mysqld) | Optimized binary with named-pipe support |
| [**mysqld-debug**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#mysqld) | Like [**mysqld**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#mysqld), but compiled with full debugging and automatic memory allocation checking |

所有前面的二进制文件都针对现代英特尔处理器进行了优化，但应该可以在任何英特尔i386级或更高的处理器上工作。

一个发行版中的每一个服务器都支持相同的存储引擎集。SHOW ENGINES语句显示一个给定的服务器支持哪些引擎。

所有Windows MySQL 8.0服务器都支持数据库目录的符号链接。

MySQL支持所有Windows平台上的TCP/IP。如果你在启用named\_pipe系统变量的情况下启动服务器，Windows上的MySQL服务器也支持命名管道。有必要明确地启用这个变量，因为一些用户在使用命名管道时遇到了关闭MySQL服务器的问题。默认是使用TCP/IP，而不考虑平台，因为在许多Windows配置中，命名管道比TCP/IP慢。

#### 2.3.4.4 初始化数据目录

如果你使用noinstall软件包安装MySQL，则不包括数据目录。要初始化数据目录，请使用2.10.1节 "初始化数据目录 "的说明。

#### 2.3.4.5 首次启动服务器

本节给出了启动MySQL服务器的一般概述。以下各节提供了从命令行或作为Windows服务启动MySQL服务器的更具体信息。

这里的信息主要适用于你使用noinstall版本安装MySQL，或者你希望手动配置和测试MySQL，而不是使用MySQL安装程序。

这些章节中的例子假定MySQL安装在C:\Program Files\MySQL\MySQL Server 8.0的默认位置。如果你将MySQL安装在不同的位置，请调整例子中显示的路径名称。

客户端有两个选择。他们可以使用TCP/IP，或者如果服务器支持命名管道连接，他们可以使用命名管道。

如果服务器在启动时启用了shared\_memory系统变量，Windows版MySQL也支持共享内存连接。客户端可以通过使用--protocol=MEMORY选项通过共享内存进行连接。

关于运行哪种服务器二进制文件的信息，见2.3.4.3节 "选择MySQL服务器类型"。

测试最好在控制台窗口（或 "DOS窗口"）的命令提示符下进行。通过这种方式，你可以让服务器在窗口中显示状态信息，因为它们很容易看到。如果你的配置出了问题，这些信息使你更容易识别和解决任何问题。

注意

在启动MySQL之前，必须对数据库进行初始化。关于初始化过程的其他信息，见第2.10.1节，"初始化数据目录"。

要启动服务器，请输入这个命令。

C:\> **"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld" --console**

对于包含InnoDB支持的服务器，当它启动时，你应该看到与下面类似的信息（路径名称和大小可能不同）。

InnoDB: The first specified datafile c:\ibdata\ibdata1 did not exist:

InnoDB: a new database to be created!

InnoDB: Setting file c:\ibdata\ibdata1 size to 209715200

InnoDB: Database physically writes the file full: wait...

InnoDB: Log file c:\iblogs\ib\_logfile0 did not exist: new to be created

InnoDB: Setting log file c:\iblogs\ib\_logfile0 size to 31457280

InnoDB: Log file c:\iblogs\ib\_logfile1 did not exist: new to be created

InnoDB: Setting log file c:\iblogs\ib\_logfile1 size to 31457280

InnoDB: Log file c:\iblogs\ib\_logfile2 did not exist: new to be created

InnoDB: Setting log file c:\iblogs\ib\_logfile2 size to 31457280

InnoDB: Doublewrite buffer not found: creating new

InnoDB: Doublewrite buffer created

InnoDB: creating foreign key constraint system tables

InnoDB: foreign key constraint system tables created

011024 10:58:25 InnoDB: Started

当服务器完成其启动序列时，你应该看到像这样的东西，这表明服务器已经准备好为客户连接服务。

mysqld: ready for connections

Version: '8.0.26' socket: '' port: 3306

服务器继续向控制台写入它产生的任何进一步的诊断输出。你可以打开一个新的控制台窗口，在其中运行客户程序。

如果你省略了--控制台选项，服务器会将诊断输出写入数据目录中的错误日志（默认为C:\Program Files\MySQL\MySQL Server 8.0\data）。错误日志是以.err为扩展名的文件，可以用--log-error选项来设置。

注意

在MySQL授予表中的初始根账户没有密码。在启动服务器后，你应该使用第2.10.4节 "确保初始MySQL账户的安全 "中的说明为其设置一个密码。

#### 2.3.4.6 从Windows命令行启动MySQL

可以从命令行手动启动MySQL服务器。这可以在任何版本的Windows上进行。

要从命令行启动mysqld服务器，你应该启动一个控制台窗口（或 "DOS窗口"）并输入这个命令。

C:\> **"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld"**

mysqld的路径可能会有所不同，这取决于你系统中MySQL的安装位置。

你可以通过执行这个命令来停止MySQL服务器。

C:\> **"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqladmin" -u root shutdown**

注意

如果MySQL根用户账户有密码，你需要用-p选项调用mysqladmin，并在出现提示时提供该密码。

该命令调用MySQL管理工具mysqladmin连接到服务器并告诉它关闭。该命令以MySQL根用户的身份连接，这是MySQL授予系统中默认的管理账户。

注意

在MySQL授予系统中的用户完全独立于Microsoft Windows下的任何操作系统用户。

如果mysqld没有启动，检查错误日志，看看服务器是否在那里写了任何信息来表明问题的原因。默认情况下，错误日志位于C:\Program Files\MySQL\MySQL Server 8.0\data目录中。它是后缀为.err的文件，或者可以通过传入-log-error选项来指定。另外，你可以尝试用--console选项启动服务器；在这种情况下，服务器可能会在屏幕上显示一些有用的信息以帮助解决问题。

最后一个选项是用--standalone和--debug选项启动mysqld。在这种情况下，mysqld会写一个日志文件C:\mysqld.trace，它应该包含mysqld不启动的原因。参见第5.9.4节，"DBUG包"。

使用 mysqld --verbose --help 来显示 mysqld 支持的所有选项。

#### 2.3.4.7 为MySQL工具定制PATH

警告

当手工编辑你的系统PATH时，你必须非常小心；意外地删除或修改现有PATH值的任何部分都可能使你的系统出现故障甚至无法使用。

为了更容易调用MySQL程序，你可以把MySQL bin目录的路径名称添加到你的Windows系统PATH环境变量中。

* 在Windows桌面上，右击我的电脑图标，选择属性。
* 接下来从出现的系统属性菜单中选择高级标签，并点击环境变量按钮。
* 在系统变量下，选择路径，然后点击编辑按钮。编辑系统变量的对话框应该出现。
* 把你的光标放在出现在标有变量值空间的文本末尾。使用End键来确保你的光标位于这个空间中文本的最末端）。然后输入你的MySQL bin目录的完整路径名称（例如，C:\Program Files\MySQL\MySQL Server 8.0\bin)

注意

必须有一个分号把这个路径和这个字段中的任何值分开。

通过单击 "确定 "来解散这个对话，并依次解散每个对话，直到所有被打开的对话都被解散。新的PATH值现在应该对你打开的任何新的命令外壳可用，允许你在DOS提示符下从系统的任何目录中输入MySQL可执行程序的名称来调用它，而不必提供路径。这包括服务器、mysql客户端和所有MySQL命令行工具，如mysqladmin和mysqldump。

如果你在同一台机器上运行多个MySQL服务器，你不应该将MySQL bin目录添加到你的Windows PATH中。

#### 2.3.4.8 作为Windows服务启动MySQL

在Windows上，运行MySQL的推荐方法是将其安装为Windows服务，这样MySQL就会在Windows启动和停止时自动启动和停止。作为服务安装的MySQL服务器也可以使用NET命令从命令行控制，或者使用图形化的Services工具。一般来说，要把MySQL安装成Windows服务，你应该用一个有管理员权限的账户来登录。

服务工具（Windows服务控制管理器）可以在Windows控制面板中找到。为了避免冲突，建议在从命令行进行服务器安装或移除操作时，关闭服务工具。

**安装服务**

在将MySQL安装为Windows服务之前，如果当前服务器正在运行，你应该首先通过使用以下命令停止它。

C:\> **"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqladmin"**

**-u root shutdown**

注意

如果MySQL根用户账户有密码，你需要用-p选项调用mysqladmin，并在出现提示时提供该密码。

该命令调用MySQL管理工具mysqladmin连接到服务器并告诉它关闭。该命令以MySQL根用户的身份连接，这是MySQL授予系统中默认的管理账户。

注意

MySQL授予系统中的用户完全独立于Windows下的任何操作系统用户。

使用此命令将服务器安装为一个服务。

C:\> **"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld" --install**

服务安装命令并不启动服务器。本节后面将给出相关说明。

为了更容易调用MySQL程序，你可以把MySQL bin目录的路径名称添加到你的Windows系统PATH环境变量中。

* 在Windows桌面上，右击我的电脑图标，选择属性。
* 接下来从出现的系统属性菜单中选择高级标签，并点击环境变量按钮。
* 在系统变量下，选择路径，然后点击编辑按钮。编辑系统变量的对话框应该出现。
* 把你的光标放在出现在标有变量值空间的文本末尾。使用End键来确保你的光标位于这个空间中文本的最末端）。然后输入你的MySQL bin目录的完整路径名称（例如，C:\Program Files\MySQL\MySQL Server 8.0\bin），并且应该有一个分号将此路径与此字段中的任何值分开。通过单击 "确定 "来解散这个对话，并依次解散每个对话，直到所有被打开的对话都被解散。现在你应该能够从系统的任何目录中通过在DOS提示符下输入MySQL可执行程序的名称来调用它，而不必提供路径。这包括服务器、mysql客户端和所有MySQL命令行工具，如mysqladmin和mysqldump。

如果你在同一台机器上运行多个MySQL服务器，你不应该将MySQL bin目录添加到你的Windows PATH。

警告

在手工编辑你的系统PATH时，你必须非常小心；意外地删除或修改现有PATH值的任何部分都可能使你的系统出现故障甚至无法使用。

在安装服务时，可以使用下列附加参数。

* 你可以在 --install 选项后面立即指定一个服务名称。默认的服务名称是MySQL。
* 如果给出了一个服务名称，可以在它后面加上一个选项。按照惯例，这应该是--defaults-file=file\_name，以指定一个选项文件的名称，服务器启动时应该从该文件中读取选项。

可以使用除-defaults-file以外的单个选项，但不鼓励使用。--defaults-file更灵活，因为它使你能够为服务器指定多个启动选项，把它们放在指定的选项文件中。

* 你也可以在服务名称后面指定一个-local-service选项。这将导致服务器使用具有有限系统权限的LocalService Windows账户运行。如果在服务名称后面同时给出-defaults-file和-local-service，它们可以是任何顺序。

对于作为Windows服务安装的MySQL服务器，以下规则决定服务器使用的服务名称和选项文件。

* 如果服务-安装命令在-install选项后面没有指定服务名称或默认服务名称（MySQL），则服务器使用MySQL的服务名称，并从标准选项文件的[mysqld]组中读取选项。
* 如果服务-安装命令在--安装选项后面指定了MySQL以外的服务名称，服务器就会使用该服务名称。它从标准选项文件中的[mysqld]组和与服务同名的组读取选项。这使你能够将[mysqld]组用于所有MySQL服务应使用的选项，并将带有服务名称的选项组用于以该服务名称安装的服务器。
* 如果服务安装命令在服务名称后面指定了一个--defaults-file选项，服务器读取选项的方式与前一项描述的相同，只是它只从命名的文件中读取选项，而忽略标准选项文件。

作为一个更复杂的例子，考虑下面的命令。

C:\> **"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld"**

**--install MySQL --defaults-file=C:\my-opts.cnf**

这里，默认的服务名称（MySQL）是在--install选项之后给出的。如果没有给出--defaults-file选项，这个命令的效果是使服务器从标准选项文件中读取[mysqld]组。然而，由于--defaults-file选项的存在，服务器会从[mysqld]选项组中读取选项，并且只从命名的文件中读取。

注意

在Windows上，如果服务器是用--defaults-file和--install选项启动的，--install必须在前。否则，mysqld.exe会尝试启动MySQL服务器。

在启动MySQL服务之前，你也可以在Windows服务工具中指定选项作为启动参数。

最后，在尝试启动MySQL服务之前，确保要运行该服务的操作系统用户的用户变量%TEMP%和%TMP%（还有%TMPDIR%，如果它曾经被设置过）指向该用户有写入权限的文件夹。运行MySQL服务的默认用户是LocalSystem，其%TEMP%和%TMP%的默认值是C:\Windows\Temp，这个目录LocalSystem默认有写入权限。但是，如果该默认设置有任何变化（例如，运行服务的用户或所提及的用户变量有变化，或者使用了--tmpdir选项将临时目录放在其他地方），MySQL服务可能无法运行，因为临时目录的写入权限没有被授予适当的用户。

启动服务

在MySQL服务器实例被安装为服务后，每当Windows启动时，Windows会自动启动该服务。该服务也可以从服务工具中立即启动，或者通过使用sc start mysqld\_service\_name或NET START mysqld\_service\_name命令启动。SC和NET命令是不区分大小写的。

当作为服务运行时，mysqld不能访问控制台窗口，所以不能在那里看到任何消息。如果mysqld不启动，检查错误日志，看看服务器是否在那里写了任何信息以表明问题的原因。错误日志位于MySQL数据目录中（例如，C:\Program Files\MySQL\MySQL Server 8.0\data）。它是后缀为.err的文件。

当MySQL服务器被安装为一个服务，并且该服务正在运行时，当Windows关闭时，Windows会自动停止该服务。也可以使用服务工具、sc stop mysqld\_service\_name 命令、NET START mysqld\_service\_name 命令或 mysqladmin shutdown 命令手动停止服务器。

如果你不希望服务在启动过程中自动启动，你也可以选择将服务器安装为一个手动服务。要做到这一点，使用 --install-manual 选项而不是 --install 选项。

C:\> **"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld" --install-manual**

移除服務

要删除作为服务安装的服务器，首先通过执行SC STOP mysqld\_service\_name或NET STOP mysqld\_service\_name停止它，如果它正在运行。然后使用SC DELETE mysqld\_service\_name来删除它。

C:\> **SC DELETE mysql**

或者，使用 mysqld --remove 选项来删除该服务。

C:\> **"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld" --remove**

如果mysqld没有作为服务运行，你可以从命令行启动它。有关说明，见第2.3.4.6节，"从Windows命令行启动MySQL"。

如果你在安装过程中遇到困难，请参阅第2.3.5节，"Microsoft Windows MySQL服务器安装的故障排除"。

关于停止或删除Windows服务的更多信息，见第5.8.2.2节，"作为Windows服务启动多个MySQL实例"。

#### 2.3.4.9 测试MySQL安装

你可以通过执行以下任何一个命令来测试MySQL服务器是否在工作。

C:\> **"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqlshow"**

C:\> **"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqlshow" -u root mysql**

C:\> **"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqladmin" version status proc**

C:\> **"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql" test**

如果mysqld对来自客户端程序的TCP/IP连接响应缓慢，可能是你的DNS出现了问题。在这种情况下，在启用 skip\_name\_resolve 系统变量的情况下启动 mysqld，并在 MySQL 授予表的 Host 列中只使用 localhost 和 IP 地址。(要确保存在一个指定IP地址的账户，否则你可能无法连接)。

你可以通过指定--管道或--协议=PIPE选项，或指定......作为主机名，强迫MySQL客户端使用命名的管道连接而不是TCP/IP。(句号)作为主机名。如果你不想使用默认的管道名称，使用--socket选项来指定管道的名称。

如果你已经为根账户设置了密码，删除了匿名账户，或创建了一个新的用户账户，那么要连接到MySQL服务器，你必须使用适当的-u和-p选项与前面显示的命令。参见第4.2.4节，"使用命令选项连接到MySQL服务器"。

关于mysqlshow的更多信息，参见第4.5.7节 "mysqlshow-显示数据库、表和列的信息"。

### 2.3.5 Windows系统MySQL服务器安装的故障排除

当首次安装和运行MySQL时，你可能会遇到某些错误，使MySQL服务器无法启动。本节帮助你诊断和纠正这些错误中的一些。

在排除服务器问题的故障时，你的第一个资源是错误日志。MySQL服务器使用错误日志来记录与阻止服务器启动的错误有关的信息。错误日志位于你的my.ini文件中指定的数据目录中。默认的数据目录位置是C:\Program Files\MySQL\MySQL Server 8.0\data，或者在Windows 7和Windows Server 2008上是C:\ProgramData\Mysql。C:\ProgramData目录在默认情况下是隐藏的。你需要改变你的文件夹选项才能看到该目录和内容。关于错误日志和理解其内容的更多信息，请参见第5.4.2节，"错误日志"。

关于可能的错误的信息，也可以参考MySQL服务启动时显示的控制台信息。在将mysqld安装为服务后，从命令行中使用SC START mysqld\_service\_name或NET START mysqld\_service\_name命令，查看有关将MySQL服务器作为服务启动的任何错误信息。参见第2.3.4.8节，"将MySQL作为Windows服务启动"。

下面的例子显示了你在安装MySQL和首次启动服务器时可能遇到的其他常见错误信息。

* 如果MySQL服务器找不到mysql特权数据库或其他关键文件，它会显示这些消息。

System error 1067 has occurred.

Fatal error: Can't open and lock privilege tables:

Table 'mysql.user' doesn't exist

当MySQL基础目录或数据目录安装在与默认位置（分别为C:\Program Files\MySQL Server 8.0和C:\Program Files\MySQL\MySQL Server 8.0\data）不同的位置时，这些消息经常发生。

当MySQL升级并安装到一个新的位置，但配置文件没有更新以反映新的位置时，这种情况可能发生。此外，新旧配置文件可能发生冲突。请确保在升级MySQL时删除或重命名任何旧的配置文件。

如果你已将MySQL安装到C:\Program Files\MySQL\MySQL Server 8.0以外的目录，请确保MySQL服务器通过使用配置（my.ini）文件知道这一点。把my.ini文件放在你的Windows目录下，通常是C:\WINDOWS。要根据WINDIR环境变量的值确定它的确切位置，请从命令提示符中发出以下命令。

C:\> **echo %WINDIR%**

你可以用任何文本编辑器，如记事本，来创建或修改选项文件。例如，如果MySQL安装在E:\mysql，数据目录是D:\MySQLdata，你可以创建选项文件，并设置一个[mysqld]部分来指定baseir和datadir选项的值。

[mysqld]

# set basedir to your installation path

basedir=E:/mysql

# set datadir to the location of your data directory

datadir=D:/MySQLdata

Microsoft Windows的路径名称在选项文件中使用（向前）斜线而不是反斜线来指定。如果你使用反斜线，请将其加倍。

[mysqld]

# set basedir to your installation path

basedir=C:\\Program Files\\MySQL\\MySQL Server 8.0

# set datadir to the location of your data directory

datadir=D:\\MySQLdata

在选项文件值中使用反斜杠的规则在第4.2.2.2节 "使用选项文件 "中给出。

如果你改变了MySQL配置文件中的datadir值，你必须在重新启动MySQL服务器之前移动现有MySQL数据目录的内容。

见第2.3.4.2节，"创建选项文件"。

如果你在没有首先停止和删除现有的MySQL服务的情况下重新安装或升级MySQL，并使用MySQL安装程序安装MySQL，你可能看到这个错误。

Error: Cannot create Windows service for MySql. Error: 0

这发生在配置向导试图安装服务并发现一个具有相同名称的现有服务。

解决这个问题的一个办法是在使用配置向导时选择一个mysql以外的服务名称。这可以使新的服务被正确安装，但留下过时的服务。虽然这没有什么危害，但最好是删除不再使用的旧服务。

要永久地删除旧的mysql服务，以具有管理权限的用户在命令行上执行以下命令。

C:\> **SC DELETE mysql**

[SC] DeleteService SUCCESS

如果SC工具对你的Windows版本不可用，请从http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/delsrv-o.asp，并使用delsrv mysql语法下载delsrv工具。

### 2.3.6 Windows安装后程序

有一些GUI工具可以执行本节中描述的大多数任务，包括

* MySQL安装程序。用于安装和升级MySQL产品。
* MySQL Workbench。管理MySQL服务器并编辑SQL语句。

如果有必要，初始化数据目录并创建MySQL授予表。由MySQL Installer执行的Windows安装操作会自动初始化数据目录。对于从ZIP存档包进行的安装，按照第2.10.1节 "初始化数据目录 "所述，初始化数据目录。

关于密码，如果你使用MySQL安装程序安装MySQL，你可能已经为初始根账户分配了一个密码。否则，请使用第2.10.4节 "确保初始MySQL账户的安全 "中给出的密码分配程序。

在分配密码之前，你可能想尝试运行一些客户端程序，以确保你能连接到服务器，并且它能正常运行。确保服务器正在运行（见2.3.4.5节，"首次启动服务器"）。你也可以设置一个MySQL服务，在Windows启动时自动运行（见第2.3.4.8节，"作为Windows服务启动MySQL"）。

这些说明假定你的当前位置是MySQL安装目录，并且它有一个bin子目录，包含这里使用的MySQL程序。如果不是这样，请相应调整命令路径名称。

如果你使用MySQL安装程序安装MySQL（见第2.3.3节，"Windows的MySQL安装程序"），默认安装目录是C:\Program Files\MySQL\MySQL Server 8.0。

C:\> **cd "C:\Program Files\MySQL\MySQL Server 8.0"**

從ZIP檔案中安裝的一個常見的安裝位置是C:\mysql。

C:\> **cd C:\mysql**

或者，将bin目录添加到你的PATH环境变量设置中。这可以使你的命令解释器正确地找到MySQL程序，这样你就可以只输入程序的名称而不是路径名称来运行程序。见第2.3.4.7节，"为MySQL工具定制PATH"。

在服务器运行的情况下，发出下列命令以验证你可以从服务器中检索信息。输出应该与这里显示的类似。

使用mysqlshow来查看存在哪些数据库。

C:\> **bin\mysqlshow**

+--------------------+

| Databases |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

| sys |

+--------------------+

已安装的数据库列表可能有所不同，但总是至少包括 mysql 和 information\_schema。

如果正确的MySQL账户不存在，前面的命令（以及其他MySQL程序的命令，如mysql）可能无法工作。例如，程序可能会出错而失败，或者你可能无法查看所有数据库。如果你使用MySQL安装程序安装MySQL，根用户会用你提供的密码自动创建。在这种情况下，你应该使用-u root和-p选项。(如果你已经确保了初始MySQL账户的安全，你必须使用这些选项)。使用-p，客户端程序会提示根密码。比如说

C:\> **bin\mysqlshow -u root -p**

Enter password: ***(enter root password here)***

+--------------------+

| Databases |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

| sys |

+--------------------+

如果指定数据库名，那么mysqlshow 会列出该数据库中的所有数据表

C:\> **bin\mysqlshow mysql**

Database: mysql

+---------------------------+

| Tables |

+---------------------------+

| columns\_priv |

| component |

| db |

| default\_roles |

| engine\_cost |

| func |

| general\_log |

| global\_grants |

| gtid\_executed |

| help\_category |

| help\_keyword |

| help\_relation |

| help\_topic |

| innodb\_index\_stats |

| innodb\_table\_stats |

| ndb\_binlog\_index |

| password\_history |

| plugin |

| procs\_priv |

| proxies\_priv |

| role\_edges |

| server\_cost |

| servers |

| slave\_master\_info |

| slave\_relay\_log\_info |

| slave\_worker\_info |

| slow\_log |

| tables\_priv |

| time\_zone |

| time\_zone\_leap\_second |

| time\_zone\_name |

| time\_zone\_transition |

| time\_zone\_transition\_type |

| user |

+---------------------------+

使用mysql命令去查询表中记录

C:\> **bin\mysql -e "SELECT User, Host, plugin FROM mysql.user" mysql**

+------+-----------+-----------------------+

| User | Host | plugin |

+------+-----------+-----------------------+

| root | localhost | caching\_sha2\_password |

关于mysql和mysqlshow的更多信息，见第4.5.1节 "mysql--MySQL命令行客户端 "和第4.5.7节 "mysqlshow--显示数据库、表和列的信息"。

### 2.3.7 Windows平台的限制

以下限制适用于在Windows平台上使用MySQL。

* 进程内存

在Windows 32位平台上，默认情况下不可能在单个进程中使用超过2GB的RAM，包括MySQL。这是因为Windows 32位的物理地址限制是4GB，而Windows内部的默认设置是在内核（2GB）和用户/应用程序（2GB）之间分割虚拟地址空间。

一些版本的Windows有一个启动时间设置，通过减少内核应用来实现更大的应用。另外，要使用超过2GB的空间，请使用64位版本的Windows。

* 文件系统别名

当使用MyISAM表时，你不能在Windows内使用别名链接到另一个卷上的数据文件，然后再链接到MySQL的主数据目录位置。

这种设施经常被用来将数据和索引文件转移到RAID或其他快速解决方案中。

* 端口的数量有限

Windows系统有大约4000个端口可供客户端连接，在一个端口上的连接关闭后，需要两到四分钟才能重新使用该端口。在客户端高速连接和断开服务器的情况下，有可能在关闭的端口再次可用之前，所有可用的端口都被用完。如果发生这种情况，即使MySQL服务器正在运行，也会出现无响应的情况。端口可能也被机器上运行的其他应用程序使用，在这种情况下，MySQL可用的端口数量会减少。

关于这个问题的更多信息，见https://support.microsoft.com/kb/196271。

* DATA DIRECTORY和INDEX DIRECTORY

CREATE TABLE语句的DATA DIRECTORY子句在Windows上只支持InnoDB表，如第15.6.1.2节 "在外部创建表 "中所述。对于MyISAM和其他存储引擎，CREATE TABLE的DATA DIRECTORY和INDEX DIRECTORY子句在Windows和其他任何有无功能的realpath()调用的平台上被忽略。

* DROP DATABASE

你不能删除一个正在被其他会话使用的数据库。

* 不区分大小写的名字

文件名在Windows上不区分大小写，所以MySQL数据库和表名在Windows上也不区分大小写。唯一的限制是数据库和表名必须在给定的语句中使用相同的大小写来指定。见第9.2.3节，"标识符大小写敏感性"。

* 目录和文件名

在Windows上，MySQL服务器只支持与当前ANSI代码页兼容的目录和文件名。例如，下面的日文目录名在西方地区设置中不起作用（代码页1252）。

datadir="C:/私たちのプロジェクトのデータ"

同样的限制也适用于SQL语句中提到的目录和文件名，如LOAD DATA中的数据文件路径名。

* The \ path name separator character

Windows中的路径名组件由 \字符分隔，这也是MySQL中的转义字符。如果你使用LOAD DATA或SELECT ... INTO OUTFILE，请使用Unix风格的文件名与/字符。

mysql> **LOAD DATA INFILE 'C:/tmp/skr.txt' INTO TABLE skr;**

mysql> **SELECT \* INTO OUTFILE 'C:/tmp/skr.txt' FROM skr;**

或者，你必须加倍使用 \字符。

mysql> **LOAD DATA INFILE 'C:\\tmp\\skr.txt' INTO TABLE skr;**

mysql> **SELECT \* INTO OUTFILE 'C:\\tmp\\skr.txt' FROM skr;**

* 关于管道的问题

管道在Windows命令行提示下不能可靠地工作。如果管道中包含字符^Z / CHAR(24)，Windows会认为它遇到了文件结束，并中止程序。

这主要是在你试图应用二进制日志时出现的问题，如下所示。

C:\> **mysqlbinlog *binary\_log\_file* | mysql --user=root**

如果你在应用日志时遇到问题，并且怀疑是因为^Z / CHAR(24)字符，你可以使用下面的解决方法。

C:\> **mysqlbinlog *binary\_log\_file* --result-file=/tmp/bin.sql**

C:\> **mysql --user=root --execute "source /tmp/bin.sql"**

后一个命令也可以用来可靠地读取任何可能包含二进制数据的SQL文件。

## 2.4 在macOS上安装MySQL

关于MySQL服务器支持的macOS版本的列表，请参见https://www.mysql.com/support/supportedplatforms/database.html。

用于macOS的MySQL有多种不同的形式。

* 原生软件包安装程序，它使用原生macOS安装程序（DMG）引导你完成MySQL的安装。更多信息，见第2.4.2节，"使用本地包在macOS上安装MySQL"。你可以在macOS上使用软件包安装程序。你用来执行安装的用户必须有管理员权限。
* 压缩的TAR档案，它使用Unix的tar和gzip命令打包的文件。要使用这种方法，你需要打开一个终端窗口。使用这种方法你不需要管理员权限；你可以用这种方法在任何地方安装MySQL服务器。关于使用这种方法的更多信息，你可以使用使用tarball的通用说明，第2.2节，"在Unix/Linux上使用通用二进制文件安装MySQL"。

除了核心安装，软件包安装程序还包括第2.4.3节 "安装和使用MySQL启动守护程序 "和第2.4.4节 "安装和使用MySQL首选项窗格"，以简化对安装的管理。

关于在macOS上使用MySQL的其他信息，见第2.4.1节 "在macOS上安装MySQL的一般注意事项"。

### 2.4.1 在macOS上安装MySQL的一般注意事项

你应该牢记以下问题和注意事项。

* 其他MySQL安装。安装过程不识别由软件包管理器（如Homebrew）安装的MySQL。安装和升级过程是针对我们提供的MySQL包。如果存在其他安装，那么考虑在执行此安装程序之前停止它们，以避免端口冲突。
* Homebrew。例如，如果你使用Homebrew将MySQL服务器安装到它的默认位置，那么MySQL安装程序会安装到一个不同的位置，并且不会从Homebrew升级版本。在这种情况下，你最终会有多个MySQL安装，默认情况下，它们试图使用相同的端口。在运行此安装程序之前，停止其他MySQL服务器实例，例如执行brew services stop mysql来停止Homebrew的MySQL服务。
* Launchd。安装了一个改变MySQL配置选项的Launchd守护程序。如果需要，可以考虑编辑它，更多信息见下面的文档。另外，macOS 10.10删除了对启动项的支持，转而使用launchd守护进程。macOS系统偏好下可选的MySQL偏好窗格使用 launchd 守护程序。
* 用户。你可能需要（或想要）创建一个特定的mysql用户来拥有MySQL目录和数据。你可以通过目录工具做到这一点，而且mysql用户应该已经存在。对于在单用户模式下使用，\_mysql的条目（注意下划线前缀）应该已经存在于系统/etc/passwd文件中。
* 数据。因为MySQL包安装程序将MySQL内容安装到一个版本和平台特定的目录中，你可以用它来升级和迁移你的数据库。你需要把数据目录从旧版本复制到新版本，或者指定一个替代的datadir值来设置数据目录的位置。默认情况下，MySQL目录被安装在/usr/local/下。
* 别名。你可能想在你的shell的资源文件中添加别名，以使它更容易从命令行访问常用的程序，如mysql和mysqladmin。bash的语法是

alias mysql=/usr/local/mysql/bin/mysql

alias mysqladmin=/usr/local/mysql/bin/mysqladmin

对于tcsh，使用

alias mysql /usr/local/mysql/bin/mysql

alias mysqladmin /usr/local/mysql/bin/mysqladmin

甚至更好的是，将/usr/local/mysql/bin添加到你的PATH环境变量中。你可以通过为你的shell修改适当的启动文件来做到这一点。更多信息，见第4.2.1节，"调用MySQL程序"。

* 移除。在你从以前的安装中复制了MySQL数据库文件并成功地启动了新的服务器之后，你应该考虑删除旧的安装文件以节省磁盘空间。此外，你还应该删除位于/Library/Receipts/mysql-VERSION.pkg的旧版本的软件包收据目录。

遗留问题。在OS X 10.7之前，MySQL服务器是与OS X Server捆绑在一起的。

### 2.4.2 使用本地软件包在macOS上安装MySQL

该软件包位于一个磁盘镜像（.dmg）文件中，你首先需要在Finder中双击其图标来挂载。然后，它应该加载镜像并显示其内容。

注意

在继续安装之前，确保通过使用MySQL管理器应用程序（在macOS服务器上）、偏好面板或命令行上的mysqladmin shutdown停止所有正在运行的MySQL服务器实例。

要使用软件包安装程序安装MySQL。

1、下载包含MySQL包安装程序的磁盘镜像（.dmg）文件（社区版本可在此获得）。双击该文件以加载磁盘镜像并查看其内容。

双击磁盘上的MySQL安装程序包。它是根据你所下载的MySQL版本来命名的。例如，对于MySQL服务器8.0.26，它可能被命名为mysql-8.0.26-osx-10.13-x86\_64.pkg。

2、最初的向导介绍屏幕引用要安装的MySQL服务器版本。点击继续，开始安装。

MySQL社区版显示了相关GNU通用公共许可证的副本。点击 "继续"，然后点击 "同意"，继续。

3、在安装类型页面，你可以点击安装，使用所有默认值执行安装向导；点击自定义，改变要安装的组件（MySQL服务器、MySQL测试、首选项窗格、Launchd支持--除了MySQL测试，其他都是默认启用的）。

注意

尽管 "改变安装位置 "选项是可见的，但安装位置不能被改变。

图2.13 MySQL软件包安装器向导。安装类型

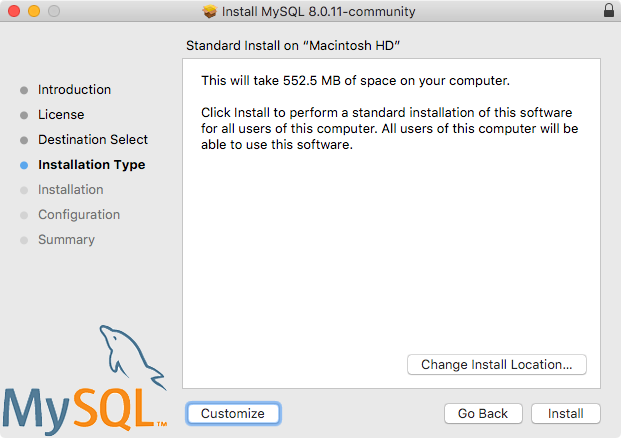
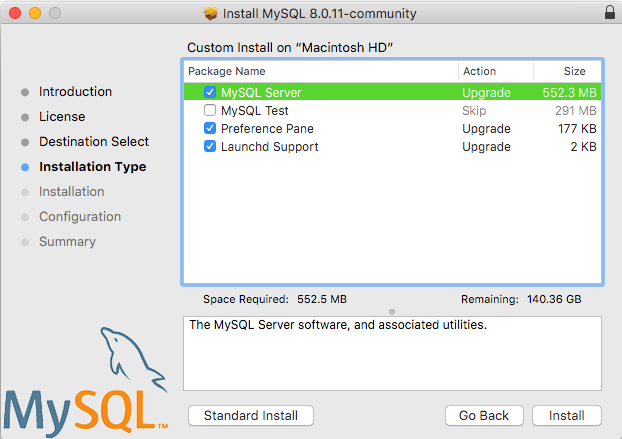


图2.14 MySQL软件包安装向导。自定义



4、点击安装来安装MySQL服务器。如果升级当前的MySQL Server安装，安装过程到此结束，否则请按照向导的附加配置步骤进行新的MySQL Server安装。

5、在新的MySQL服务器安装成功后，通过选择密码的默认加密类型完成配置步骤，定义根密码，并在启动时启用（或禁用）MySQL服务器。

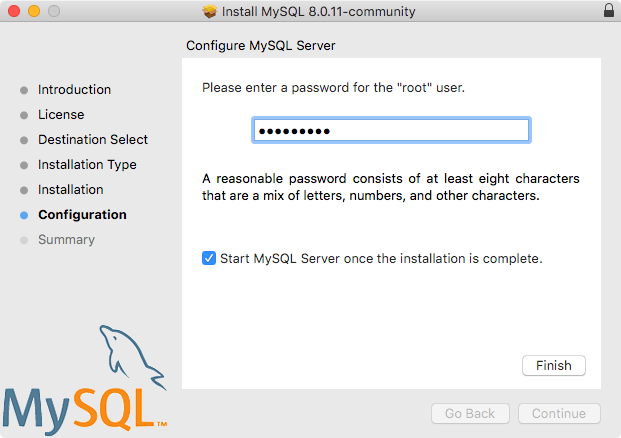
6、默认的MySQL 8.0密码机制是caching\_sha2\_password（强），这个步骤允许你将其改为mysql\_native\_password（遗留）。

图2.15 MySQL包安装器向导。选择一个密码加密类型



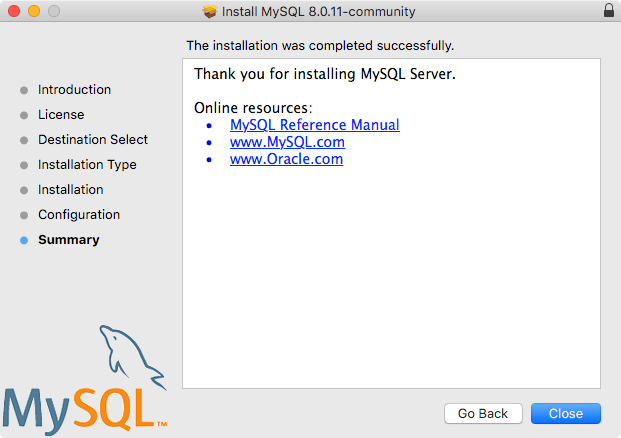
7、为根用户定义一个密码，同时切换MySQL服务器是否应在配置步骤完成后启动。

图2.16 MySQL包安装器向导。定义根密码



8、摘要是最后一步，并参考成功和完整的MySQL服务器安装。关闭该向导。

图2.17 MySQL软件包安装器向导。总结



MySQL服务器现在已经安装完毕。如果你选择不启动MySQL，那么从命令行使用launchctl，或者使用MySQL首选项窗格点击 "启动 "来启动MySQL。有关其他信息，请参见第2.4.3节 "安装和使用MySQL启动守护程序"，以及第2.4.4节 "安装和使用MySQL首选项窗格"。使用MySQL首选项窗格或launchd来配置MySQL在启动时自动启动。

当使用软件包安装程序进行安装时，文件被安装到与安装版本和平台名称相匹配的/usr/local中的一个目录。例如，安装程序文件mysql-8.0.26-osx10.15-x86\_64.dmg将MySQL安装到/usr/local/mysql-8.0.26-osx10.15-x86\_64/，其符号链接为/usr/local/mysql。下表显示了这个MySQL安装目录的布局。

注意

macOS安装过程不创建也不安装my.cnf MySQL配置文件样本。

表2.7 macOS上的MySQL安装布局

|  |  |
| --- | --- |
| **Directory** | **Contents of Directory** |
| bin | [**mysqld**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#mysqld) server, client and utility programs |
| data | Log files, databases, where /usr/local/mysql/data/mysqld.local.err is the default error log |
| docs | Helper documents, like the Release Notes and build information |
| include | Include (header) files |
| lib | Libraries |
| man | Unix manual pages |
| mysql-test | MySQL test suite ('MySQL Test' is disabled by default during the installation process when using the installer package (DMG)) |
| share | Miscellaneous support files, including error messages, dictionary.txt, and rewriter SQL |
| support-files | Support scripts, such as mysqld\_multi.server, mysql.server, and mysql-log-rotate. |
| /tmp/mysql.sock | Location of the MySQL Unix socket |

### 2.4.3 安装和使用MySQL启动守护程序

macOS使用启动守护程序来自动启动、停止和管理进程和应用程序，如MySQL。

默认情况下，macOS上的安装包（DMG）安装了一个名为/Library/LaunchDaemons/com.oracle.oss.mysql.mysqld.plist的launchd文件，其中包含一个类似的plist定义。

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version="1.0">

<dict>

<key>Label</key> <string>com.oracle.oss.mysql.mysqld</string>

<key>ProcessType</key> <string>Interactive</string>

<key>Disabled</key> <false/>

<key>RunAtLoad</key> <true/>

<key>KeepAlive</key> <true/>

<key>SessionCreate</key> <true/>

<key>LaunchOnlyOnce</key> <false/>

<key>UserName</key> <string>\_mysql</string>

<key>GroupName</key> <string>\_mysql</string>

<key>ExitTimeOut</key> <integer>600</integer>

<key>Program</key> <string>/usr/local/mysql/bin/mysqld</string>

<key>ProgramArguments</key>

<array>

<string>/usr/local/mysql/bin/mysqld</string>

<string>--user=\_mysql</string>

<string>--basedir=/usr/local/mysql</string>

<string>--datadir=/usr/local/mysql/data</string>

<string>--plugin-dir=/usr/local/mysql/lib/plugin</string>

<string>--log-error=/usr/local/mysql/data/mysqld.local.err</string>

<string>--pid-file=/usr/local/mysql/data/mysqld.local.pid</string>

<string>--keyring-file-data=/usr/local/mysql/keyring/keyring</string>

<string>--early-plugin-load=keyring\_file=keyring\_file.so</string>

</array>

<key>WorkingDirectory</key> <string>/usr/local/mysql</string>

</dict>

</plist>

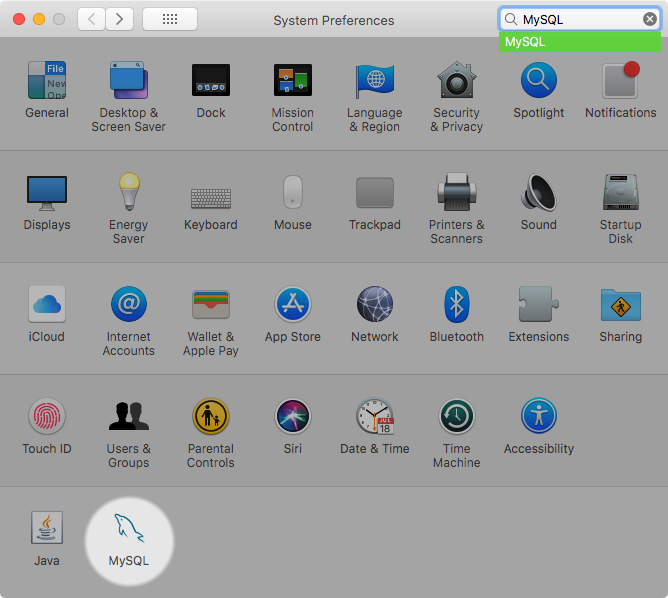
注意事项

一些用户报告说，添加plist DOCTYPE声明会导致 launchd操作失败，尽管它通过了lint检查。我们怀疑这是一个复制-粘贴的错误。包含上述片段的文件的md5校验和是d925f05f6d1b6ee5ce5451b596d6baed。

要启用launchd服务，你可以这样做。

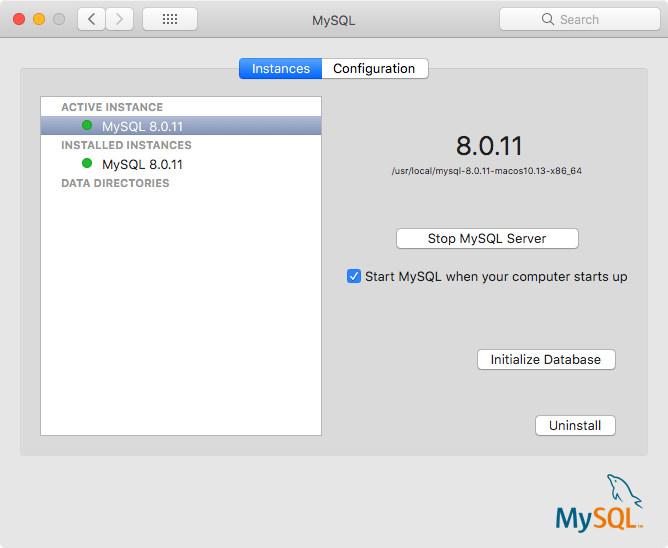
* 打开macOS系统偏好设置，选择MySQL偏好设置面板，然后执行启动MySQL服务器。

图2.18 MySQL首选项面板。位置



实例页面包括一个启动或停止MySQL的选项，而初始化数据库会重新创建数据/目录。Uninstall卸载MySQL服务器，也可以选择卸载MySQL首选项面板和启动d信息。

图2.19 MySQL首选项面板：实例



* 或者，手动加载 launchd 文件。

shell> cd /Library/LaunchDaemons

shell> sudo launchctl load -F com.oracle.oss.mysql.mysqld.plist

* 要配置MySQL在启动时自动启动，你可以。

shell> sudo launchctl load -w com.oracle.oss.mysql.mysqld.plist

注意

当升级MySQL服务器时，launchd安装过程会删除与MySQL服务器5.7.7及以下版本一起安装的旧启动项。

升级也会替换你现有的名为com.oracle.os.mysql.mysqld.plist的launchd文件。

额外的 launchd 相关信息。

* plist 条目覆盖了 my.cnf 条目，因为它们是作为命令行参数传递的。有关传入程序选项的其他信息，请参见第4.2.2节 "指定程序选项"。
* ProgramArguments部分定义了传入程序的命令行选项，在本例中是mysqld二进制。
* 默认的plist定义是在考虑到不太复杂的使用情况下编写的。对于更复杂的设置，你可能想删除一些参数，而依靠MySQL配置文件，如my.cnf。
* 如果你编辑了plist文件，那么在重新安装或升级MySQL时，不要选中安装程序选项。否则，你编辑的plist文件会被覆盖，所有编辑的内容都会丢失。

因为默认的plist定义定义了几个ProgramArguments，你可以删除这些参数中的大部分，而依靠你的my.cnf MySQL配置文件来定义它们。比如说。

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version="1.0">

<dict>

<key>Label</key> <string>com.oracle.oss.mysql.mysqld</string>

<key>ProcessType</key> <string>Interactive</string>

<key>Disabled</key> <false/>

<key>RunAtLoad</key> <true/>

<key>KeepAlive</key> <true/>

<key>SessionCreate</key> <true/>

<key>LaunchOnlyOnce</key> <false/>

<key>UserName</key> <string>\_mysql</string>

<key>GroupName</key> <string>\_mysql</string>

<key>ExitTimeOut</key> <integer>600</integer>

<key>Program</key> <string>/usr/local/mysql/bin/mysqld</string>

<key>ProgramArguments</key>

<array>

<string>/usr/local/mysql/bin/mysqld</string>

<string>--user=\_mysql</string>

<string>--basedir=/usr/local/mysql</string>

<string>--datadir=/usr/local/mysql/data</string>

<string>--plugin-dir=/usr/local/mysql/lib/plugin</string>

<string>--log-error=/usr/local/mysql/data/mysqld.local.err</string>

<string>--pid-file=/usr/local/mysql/data/mysqld.local.pid</string>

<string>--keyring-file-data=/usr/local/mysql/keyring/keyring</string>

<string>--early-plugin-load=keyring\_file=keyring\_file.so</string>

</array>

<key>WorkingDirectory</key> <string>/usr/local/mysql</string>

</dict>

</plist>

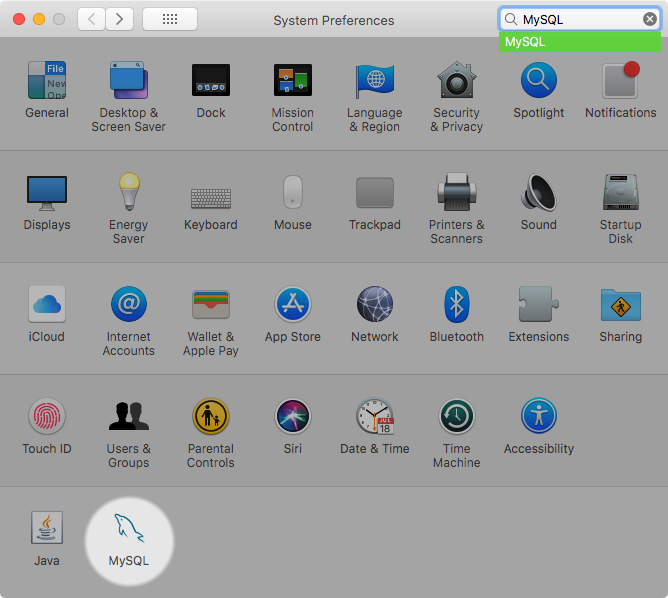
在这种情况下，baseir、datadir、plugin\_dir、log\_error、pid\_file、keyring\_file\_data和--early-plugin-load选项被从默认的plist ProgramArguments定义中移除，你可能已经在my.cnf中定义了。

### 2.4.4 安装和使用MySQL首选项面板

MySQL安装包包括一个MySQL首选项窗格，使你能够在启动MySQL安装时启动、停止和控制自动启动。

这个首选项窗格是默认安装的，并列在你的系统的系统首选项窗口下。

图2.20 MySQL首选项窗格。位置

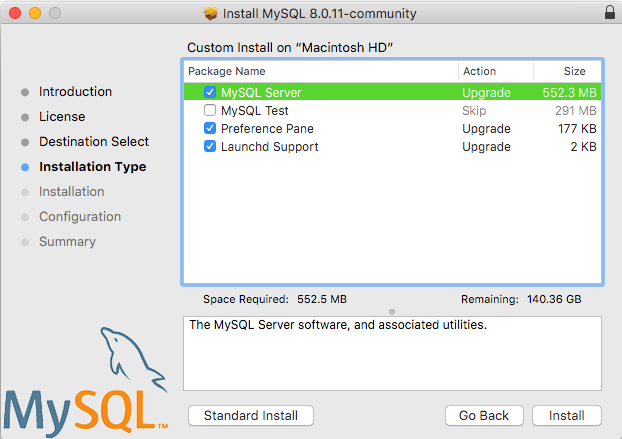


MySQL偏好面板是与安装MySQL服务器的DMG文件一起安装的。通常情况下，它与MySQL服务器一起安装，但它也可以单独安装。

1、通过安装MySQL服务器的过程，如2.4.2节 "在macOS上使用本地包安装MySQL "的文档中所述。

2、在安装类型步骤中单击 "自定义"。这里列出了 "偏好窗格 "选项，并默认为启用；确保它没有被取消选择。其他选项，如MySQL服务器，可以选择或取消选择。

图2.21 MySQL软件包安装向导。自定义



3、完成安装过程。

注意

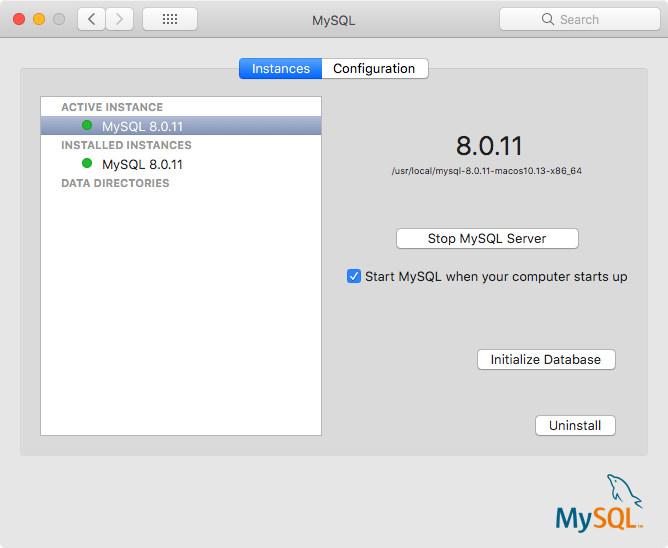
MySQL首选项窗格只启动和停止从已安装在默认位置的MySQL软件包安装的MySQL安装。

一旦安装了MySQL首选项窗格，你就可以使用这个首选项窗格控制你的MySQL服务器实例。

实例页面包括一个启动和停止MySQL的选项，而初始化数据库会重新创建数据/目录。Uninstall卸载MySQL服务器，并可选择卸载paint和launchd信息。

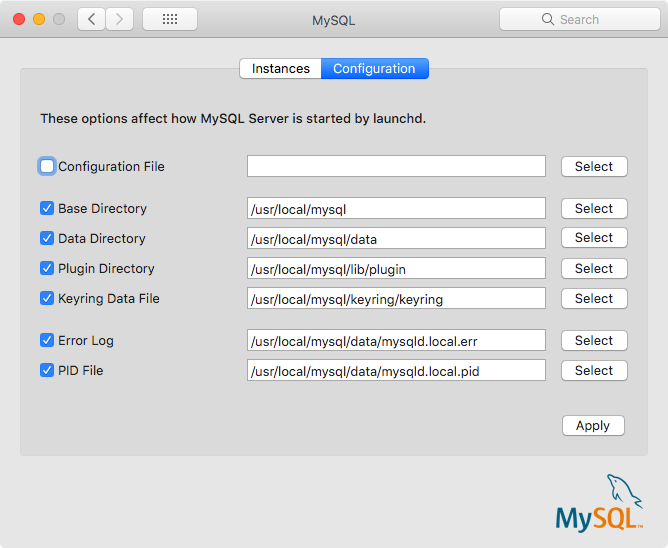
实例页面包括一个启动或停止MySQL的选项，并且初始化数据库会重新创建数据/目录。Uninstall卸载MySQL Server，也可选择卸载MySQL偏好面板和launchd信息。

图2.22 MySQL首选项面板：实例



配置页面显示了MySQL服务器选项，包括MySQL配置文件的路径。

图2.23 MySQL首选项窗格。配置



MySQL首选项窗格显示了MySQL服务器的当前状态，如果服务器没有运行，则显示停止（红色），如果服务器已经启动，则显示运行（绿色）。首选项窗格还显示了当前对MySQL服务器是否被设置为自动启动的设置。

## 2.5 在Linux上安装MySQL

Linux支持许多不同的解决方案来安装MySQL。我们建议你使用Oracle的一个发行版，它有多种安装方法可供选择。

**Table 2.8 Linux Installation Methods and Information**

|  |  |  |
| --- | --- | --- |
| **Type** | **Setup Method** | **Additional Information** |
| **Apt** | Enable the [MySQL Apt repository](https://dev.mysql.com/downloads/repo/apt/) | [Documentation](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#linux-installation-apt-repo) |
| **Yum** | Enable the [MySQL Yum repository](https://dev.mysql.com/downloads/repo/yum/) | [Documentation](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#linux-installation-yum-repo) |
| **Zypper** | Enable the [MySQL SLES repository](https://dev.mysql.com/downloads/repo/suse/) | [Documentation](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#linux-installation-sles-repo) |
| **RPM** | [Download](https://dev.mysql.com/downloads/mysql/) a specific package | [Documentation](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#linux-installation-rpm) |
| **DEB** | [Download](https://dev.mysql.com/downloads/mysql/) a specific package | [Documentation](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#linux-installation-debian) |
| **Generic** | [Download](https://dev.mysql.com/downloads/mysql/) a generic package | [Documentation](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#binary-installation) |
| **Source** | Compile from [source](https://dev.mysql.com/downloads/mysql/) | [Documentation](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#source-installation) |
| **Docker** | Use the [Oracle Container Registry](https://container-registry.oracle.com/). You can also use Docker Hub for MySQL Community Edition and [My Oracle Support](https://support.oracle.com/) for MySQL Enterprise Edition. | [Documentation](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#linux-installation-docker) |
| **Oracle Unbreakable Linux Network** | Use ULN channels | [Documentation](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#uln-installation) |

作为一种选择，你可以使用你系统上的软件包管理器，从你的Linux发行版的本地软件库中自动下载和安装MySQL的软件包。这些本地软件包通常比当前可用的版本晚几个版本。你通常也无法安装开发里程碑版本（DMR），因为这些通常不在本地软件库中提供。关于使用本地软件包安装程序的更多信息，请参见第2.5.7节，"从本地软件库安装Linux上的MySQL"。

注意

对于许多Linux的安装，你想把MySQL设置为在你的机器启动时自动启动。许多本地软件包的安装为你执行这一操作，但对于源码、二进制和RPM解决方案，你可能需要单独进行设置。所需的脚本mysql.server可以在MySQL安装目录下的support-files目录或MySQL源树中找到。你可以把它安装为/etc/init.d/mysql，以便自动启动和关闭MySQL。见第4.3.3节，"mysql.server--MySQL服务器启动脚本"。

### 2.5.1 使用MySQL Yum资源库在Linux上安装MySQL

用于Oracle Linux、Red Hat Enterprise Linux、CentOS和Fedora的MySQL Yum仓库提供了用于安装MySQL服务器、客户端、MySQL Workbench、MySQL Utilities、MySQL Router、MySQL Shell、Connector/ODBC、Connector/Python等的RPM包（并非所有包都适用于所有发行版；详情见用Yum安装其他MySQL产品和组件）。

**在你开始之前**

作为一个流行的开源软件，MySQL以其原始或重新打包的形式被广泛地安装在许多系统上，来源各异，包括不同的软件下载网站、软件库等。下面的说明假定MySQL尚未使用第三方分发的RPM包安装在你的系统上；如果不是这样，请遵循第2.11.7节 "用MySQL Yum Repository升级MySQL "或使用MySQL Yum Repository替换第三方分发的MySQL中给出的说明。

**全新安装MySQL的步骤**

按照下面的步骤，用MySQL Yum资源库安装最新的GA版MySQL。

1、添加MySQL Yum存储库

首 先，将MySQL Yum资源库添加到你的系统的资源库列表。这是一个一次性的操作，可以通过安装MySQL提供的RPM来进行。遵循这些步骤。

1. 转到MySQL开发者区的Download MySQL Yum Repository页面（<https://dev.mysql.com/downloads/repo/yum/>）。
2. 选择并下载适合你的平台的发布包。
3. 用以下命令安装下载的发布包，用下载的RPM包的名称替换平台和版本特定的包名。

shell> **sudo yum install *platform-and-version-specific-package-name*.rpm**

对于基于EL6的系统，该命令的形式为。

shell> **sudo yum install mysql80-community-release-el6-*{version-number}*.noarch.rpm**

对于基于EL7的系统

shell> **sudo yum install mysql80-community-release-el7-*{version-number}*.noarch.rpm**

对于一个基于EL8的系统。

shell> **sudo yum install mysql80-community-release-el8-*{version-number}*.noarch.rpm**

对于Fedora 33

shell> **sudo dnf install mysql80-community-release-fc33-*{version-number}*.noarch.rpm**

对于 Fedora 32:

shell> **sudo dnf install mysql80-community-release-fc32-*{version-number}*.noarch.rpm**

该安装命令将MySQL Yum软件库添加到你系统的软件库列表中，并下载GnuPG密钥以检查软件包的完整性。参见第2.1.4.2节，"使用GnuPG的签名检查"，了解GnuPG密钥检查的细节。

你可以通过以下命令检查MySQL Yum软件库是否已经成功添加（对于启用dnf的系统，在命令中用dnf替换yum）。

shell> **yum repolist enabled | grep "mysql.\*-community.\*"**

注意

一旦MySQL Yum仓库在你的系统上被启用，任何通过yum update命令进行的全系统更新（对于启用dnf的系统，则为dnf upgrade）都会升级你系统上的MySQL包，并替换任何本地第三方包，如果Yum在MySQL Yum仓库中找到它们的替代品；见2.11.7节，"用MySQL Yum仓库升级MySQL"，以讨论在你系统上的一些可能影响，见升级共享的客户端库。

2、选择一个发布系列

当使用MySQL Yum资源库时，最新的GA系列（目前是MySQL 8.0）被默认选择用于安装。如果这是你想要的，你可以跳到下一个步骤，安装MySQL。

在MySQL Yum资源库中，MySQL社区服务器的不同发布系列被托管在不同的子资源库中。最新的GA系列（目前是MySQL 8.0）的子库是默认启用的，所有其他系列的子库（例如，MySQL 8.0系列）是默认禁用的。使用这个命令可以看到MySQL Yum软件库中的所有子软件库，并查看其中哪些是启用或禁用的（对于启用dnf的系统，在命令中用dnf替换yum）。

shell> **yum repolist all | grep mysql**

要安装最新的GA系列的最新版本，不需要配置。要安装最新GA系列以外的特定系列的最新版本，在运行安装命令之前，禁用最新GA系列的子仓库，并启用特定系列的子仓库。如果你的平台支持yum-config-manager，你可以通过发布这些命令来完成，这些命令禁用了5.7系列的子库，启用了8.0系列的子库。

shell> **sudo yum-config-manager --disable mysql57-community**

shell> **sudo yum-config-manager --enable mysql80-community**

对于dnf启用的平台。

shell> **sudo dnf config-manager --disable mysql57-community**

shell> **sudo dnf config-manager --enable mysql80-community**

除了使用yum-config-manager或dnf config-manager命令，你还可以通过手动编辑/etc/yum.repos.d/mysql-community.repo文件来选择一个发布系列。这是该文件中一个典型的发布系列子库的条目。

[mysql57-community]

name=MySQL 5.7 Community Server

baseurl=http://repo.mysql.com/yum/mysql-5.7-community/el/6/$basearch/

enabled=1

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

找到你要配置的子库的条目，并编辑 enabled 选项。指定 enabled=0 来禁用一个子库，或者 enabled=1 来启用一个子库。例如，要安装MySQL 8.0，确保你在上述MySQL 5.7的子库条目中启用=0，而在8.0系列的条目中启用=1。

# Enable to use MySQL 8.0

[mysql80-community]

name=MySQL 8.0 Community Server

baseurl=http://repo.mysql.com/yum/mysql-8.0-community/el/6/$basearch/

enabled=1

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

你应该在任何时候只为一个发行系列启用子库。当启用了多个发布系列的子库时，Yum会使用最新的系列。

通过运行下面的命令并检查其输出，来验证正确的子仓库已经被启用和禁用（对于支持dnf的系统，用dnf替换命令中的yum）。

shell> **yum repolist enabled | grep mysql**

3、禁用默认的MySQL模块

仅限EL8系统）基于EL8的系统，如RHEL8和Oracle Linux 8包括一个默认启用的MySQL模块。除非该模块被禁用，否则它会掩盖由MySQL存储库提供的软件包。要禁用所包含的模块并使MySQL资源库的软件包可见，使用以下命令（对于支持dnf的系统，在命令中用dnf替换yum）。

shell> **sudo yum module disable mysql**

4、安装MySQL

用下面的命令安装MySQL（对于支持dnf的系统，用dnf代替命令中的yum）。

shell> **sudo yum install mysql-community-server**

这将安装MySQL服务器（mysql-community-server）的包，以及运行服务器所需组件的包，包括客户端（mysql-community-client）、客户端和服务器的常见错误信息和字符集（mysql-community-common）以及共享客户端库（mysql-community-libs）的包。

5、启动MySQL服务器

用以下命令启动MySQL服务器。

shell> **systemctl start mysqld**

你可以用下面的命令检查MySQL服务器的状态。

shell> **systemctl status mysqld**

如果操作系统启用了systemd，应使用标准的systemctl（或者参数相反的service）命令，如stop、start、status和restart来管理MySQL服务器服务。mysqld服务默认是启用的，并且在系统重启时启动。更多信息见2.5.9节 "用systemd管理MySQL服务器"。

在服务器初始启动时，考虑到服务器的数据目录为空，会发生以下情况。

* 服务器被初始化。
* 在数据目录中生成SSL证书和密钥文件。
* validate\_password被安装并启用。
* 一个超级用户账户'root'@'localhost'被创建。超级用户的密码被设置并存储在错误日志文件中。要显示它，请使用以下命令。

shell> **sudo grep 'temporary password' /var/log/mysqld.log**

尽快改变根密码，用生成的临时密码登录，为超级用户账户设置一个自定义密码。

shell> **mysql -uroot -p**

mysql> **ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass4!';**

注意

validate\_password是默认安装的。validate\_password实现的默认密码策略要求密码至少包含一个大写字母、一个小写字母、一个数字和一个特殊字符，并且密码总长度至少为8个字符。

关于安装后程序的更多信息，请参见第2.10节，"安装后设置和测试"。

注意事项

基于EL7的平台的兼容性信息。来自平台的本地软件库的以下RPM包与来自MySQL Yum库的安装MySQL服务器的包不兼容。一旦你使用MySQL Yum仓库安装了MySQL，你就不能安装这些包（反之亦然）。

* akonadi-mysql

用Yum安装额外的MySQL产品和组件

你可以使用Yum来安装和管理MySQL的各个组件。其中一些组件被托管在MySQL Yum仓库的子仓库中：例如，MySQL连接器可以在MySQL连接器社区子仓库中找到，而MySQL工作台可以在MySQL工具社区中找到。你可以使用下面的命令，从MySQL Yum库中列出所有可用于你的平台的MySQL组件的包（对于支持dnf的系统，在命令中用dnf代替yum）。

shell> **sudo yum --disablerepo=\\* --enablerepo='mysql\*-community\*' list available**

用下面的命令安装你选择的任何包，用包的名称代替包的名称（对于支持dnf的系统，用dnf代替命令中的yum）。

shell> **sudo yum install *package-name***

例如，要在Fedora上安装MySQL Workbench。

shell> **sudo dnf install mysql-workbench-community**

要安装共享客户端库（对于支持dnf的系统，在命令中用dnf代替yum）。

shell> **sudo yum install mysql-community-libs**

特定平台说明

ARM 支持

在Oracle Linux 7上支持ARM 64位（arch64），需要Oracle Linux 7软件集合仓库（ol7\_software\_collections）。例如，要安装服务器。

shell> yum-config-manager --enable ol7\_software\_collections

shell> yum install mysql-community-server

注意

从MySQL 8.0.12开始，ARM 64位（arch64）在Oracle Linux 7上得到支持。

已知的局限性

8.0.12版本要求你在执行yum安装步骤后通过执行ln -s /opt/oracle/oracle-armtoolset-1/root/usr/lib64 /usr/lib64/gcc7来调整libstdc++7路径。

用Yum更新MySQL

除了安装，你还可以使用MySQL Yum仓库执行MySQL产品和组件的更新。详情见第2.11.7节 "用MySQL Yum资源库升级MySQL"。

### 2.5.2 使用MySQL APT资源库在Linux上安装MySQL

MySQL APT资源库提供了用于在当前的Debian和Ubuntu版本上安装和管理MySQL服务器、客户端和其他组件的deb包。

使用MySQL APT资源库的说明见《使用MySQL APT资源库的快速指南》。

### 2.5.3 使用MySQL SLES资源库在Linux上安装MySQL

MySQL SLES资源库提供用于在SUSE Enterprise Linux服务器上安装和管理MySQL服务器、客户端和其他组件的RPM包。

使用MySQL SLES资源库的说明见《使用MySQL SLES资源库的快速指南》。

### 2.5.4 使用Oracle的RPM包在Linux上安装MySQL

在基于RPM的Linux发行版上安装MySQL的推荐方法是使用Oracle提供的RPM包。对于MySQL的社区版，有两个来源可以获得它们。

* 从MySQL软件库中。
  + MySQL Yum库（详见第2.5.1节，"使用MySQL Yum库在Linux上安装MySQL"）。
  + MySQL SLES资源库（详见第2.5.3节，"使用MySQL SLES资源库在Linux上安装MySQL"）。
* 从MySQL开发者区的 "下载MySQL社区服务器 "页面。

注意

其他供应商也提供MySQL的RPM发行版。请注意，它们在特性、功能和惯例（包括通信设置）方面可能与Oracle构建的版本不同，而且本手册中的安装说明不一定适用于它们。应该参考供应商的说明。

MySQL RPM包

**Table 2.9 RPM Packages for MySQL Community Edition**

|  |  |
| --- | --- |
| **Package Name** | **Summary** |
| **mysql-community-client** | MySQL client applications and tools |
| **mysql-community-common** | Common files for server and client libraries |
| **mysql-community-devel** | Development header files and libraries for MySQL database client applications |
| **mysql-community-embedded-compat** | MySQL server as an embedded library with compatibility for applications using version 18 of the library |
| **mysql-community-libs** | Shared libraries for MySQL database client applications |
| **mysql-community-libs-compat** | Shared compatibility libraries for previous MySQL installations |
| **mysql-community-server** | Database server and related tools |
| **mysql-community-server-debug** | Debug server and plugin binaries |
| **mysql-community-test** | Test suite for the MySQL server |
| **mysql-community** | The source code RPM looks similar to mysql-community-8.0.26-1.el7.src.rpm, depending on selected OS |
| Additional \*debuginfo\* RPMs | There are several **debuginfo** packages: mysql-community-client-debuginfo, mysql-community-libs-debuginfo mysql-community-server-debug-debuginfo mysql-community-server-debuginfo, and mysql-community-test-debuginfo. |

**Table 2.10 RPM Packages for the MySQL Enterprise Edition**

|  |  |
| --- | --- |
| **Package Name** | **Summary** |
| **mysql-commercial-backup** | MySQL Enterprise Backup (added in 8.0.11) |
| **mysql-commercial-client** | MySQL client applications and tools |
| **mysql-commercial-common** | Common files for server and client libraries |
| **mysql-commercial-devel** | Development header files and libraries for MySQL database client applications |
| **mysql-commercial-embedded-compat** | MySQL server as an embedded library with compatibility for applications using version 18 of the library |
| **mysql-commercial-libs** | Shared libraries for MySQL database client applications |
| **mysql-commercial-libs-compat** | Shared compatibility libraries for previous MySQL installations; the version of the libraries matches the version of the libraries installed by default by the distribution you are using |
| **mysql-commercial-server** | Database server and related tools |
| **mysql-commercial-test** | Test suite for the MySQL server |
| Additional \*debuginfo\* RPMs | There are several **debuginfo** packages: mysql-commercial-client-debuginfo, mysql-commercial-libs-debuginfo mysql-commercial-server-debug-debuginfo mysql-commercial-server-debuginfo, and mysql-commercial-test-debuginfo. |

rpm包的全名语法如下

***packagename***-***version***-***distribution***-***arch***.rpm

发行版和arch值表示Linux发行版和软件包所对应的处理器类型。关于发行版标识符的列表，见下表。

**table 2.11 MySQL Linux RPM Package Distribution Identifiers**

|  |  |
| --- | --- |
| **Distribution Value** | **Intended Use** |
| el***{version}*** where ***{version}*** is the major Enterprise Linux version, such as **el8** | EL6, EL7, and EL8-based platforms (for example, the corresponding versions of Oracle Linux, Red Hat Enterprise Linux, and CentOS) |
| fc***{version}*** where ***{version}*** is the major Fedora version, such as **fc33** | Fedora 32 and 33 |
| **sles12** | SUSE Linux Enterprise Server 12 |

要查看一个RPM包中的所有文件（例如，mysql-community-server），使用以下命令。

shell> **rpm -qpl mysql-community-server-*version*-*distribution*-*arch*.rpm**

本节其余部分的讨论仅适用于使用直接从Oracle下载的RPM包的安装过程，而不是通过MySQL资源库。

一些包之间存在依赖关系。如果你计划安装许多包，你可能希望下载RPM捆绑焦油文件，它包含上面列出的所有RPM包，因此你不需要单独下载它们。

在大多数情况下，你需要安装mysql-community-server、mysql-community-client、mysql-community-libs、mysql-community-common和mysql-community-libs-compat包来获得一个功能性的、标准的MySQL安装。要执行这样一个标准的、基本的安装，进入包含所有这些包的文件夹（最好是没有其他具有类似名称的RPM包），并发出以下命令。

shell> **sudo yum install mysql-community-{server,client,common,libs}-\***

对于SLES用zypper替换yum，对于Fedora用dnf。

虽然使用yum这样的高级软件包管理工具来安装软件包更为可取，但喜欢直接使用rpm命令的用户可以用rpm -Uvh命令代替yum安装命令；然而，使用rpm -Uvh代替会使安装过程更容易失败，因为安装过程可能遇到潜在的依赖问题。

要只安装客户端程序，你可以在你要安装的软件包列表中跳过mysql-community-server；发出以下命令。

shell> **sudo yum install mysql-community-{client,common,libs}-\***

对于SLES用zypper替换yum，对于Fedora用dnf。

使用RPM包对MySQL进行标准安装，会在系统目录下创建文件和资源，如下表所示

**Table 2.12 MySQL Installation Layout for Linux RPM Packages from the MySQL Developer Zone**

|  |  |
| --- | --- |
| **Files or Resources** | **Location** |
| Client programs and scripts | /usr/bin |
| [**mysqld**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#mysqld) server | /usr/sbin |
| Configuration file | /etc/my.cnf |
| Data directory | /var/lib/mysql |
| Error log file | For RHEL, Oracle Linux, CentOS or Fedora platforms: /var/log/mysqld.log  For SLES: /var/log/mysql/mysqld.log |
| Value of [**secure\_file\_priv**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_secure_file_priv) | /var/lib/mysql-files |
| System V init script | For RHEL, Oracle Linux, CentOS or Fedora platforms: /etc/init.d/mysqld  For SLES: /etc/init.d/mysql |
| Systemd service | For RHEL, Oracle Linux, CentOS or Fedora platforms: mysqld  For SLES: mysql |
| Pid file | /var/run/mysql/mysqld.pid |
| Socket | /var/lib/mysql/mysql.sock |
| Keyring directory | /var/lib/mysql-keyring |
| Unix manual pages | /usr/share/man |
| Include (header) files | /usr/include/mysql |
| Libraries | /usr/lib/mysql |
| Miscellaneous support files (for example, error messages, and character set files) | /usr/share/mysql |

该安装还在系统上创建了一个名为mysql的用户和一个名为mysql的组。

注意

使用旧软件包安装先前版本的MySQL可能创建了一个名为/usr/my.cnf的配置文件。强烈建议你检查该文件的内容，并将里面所需的设置迁移到/etc/my.cnf文件，然后删除/usr/my.cnf。

在安装过程结束时，MySQL不会被自动启动。对于Red Hat Enterprise Linux、Oracle Linux、CentOS和Fedora系统，使用以下命令来启动MySQL。

shell> **systemctl start mysqld**

对于SLES系统，命令是相同的，但服务名称不同。

shell> **systemctl start mysql**

如果操作系统启用了systemd，应使用标准的systemctl（或者参数相反的service）命令，如stop、start、status和restart来管理MySQL服务器服务。mysqld服务默认是启用的，并且在系统重启时启动。注意，某些事情在systemd平台上可能会有不同的工作方式：例如，改变数据目录的位置可能会导致问题。更多信息请参见2.5.9节 "用systemd管理MySQL服务器"。

在使用RPM和DEB包进行升级安装时，如果升级发生时MySQL服务器正在运行，那么MySQL服务器会被停止，升级发生后，MySQL服务器会被重新启动。有一个例外：如果在升级期间版本也发生了变化（例如从社区版到商业版，或者反过来），那么MySQL服务器不会被重新启动。

在服务器的初始启动中，鉴于服务器的数据目录是空的，以下情况会发生。

* 服务器被初始化。
* 在数据目录中生成SSL证书和密钥文件。
* validate\_password被安装并启用。
* 一个超级用户账户'root'@'localhost'被创建。超级用户的密码被设置并存储在错误日志文件中。要揭示它，对于RHEL、Oracle Linux、CentOS和Fedora系统使用以下命令。

shell> **sudo grep 'temporary password' /var/log/mysqld.log**

对SLES系统使用下面的命令。

shell> **sudo grep 'temporary password' /var/log/mysql/mysqld.log**

下一步是用生成的临时密码登录，并为超级用户账户设置一个自定义密码。

shell> **mysql -uroot -p**

mysql> **ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass4!';**

注意

validate\_password是默认安装的。validate\_password实现的默认密码策略要求密码至少包含一个大写字母、一个小写字母、一个数字和一个特殊字符，并且密码总长度至少为8个字符。

如果在安装过程中出了问题，你可能会在错误日志文件/var/log/mysqld.log中找到调试信息。

对于某些Linux发行版，可能需要增加对mysqld可用的文件描述符数量的限制。见B.3.2.16节，"文件未找到和类似错误"

从多个MySQL版本安装客户端库。 有可能安装多个客户库版本，例如，你想保持与针对以前的库链接的旧应用程序的兼容性。要安装一个较早的客户库，使用rpm的--oldpackage选项。例如，要在EL6系统上安装mysql-community-libs-5.5，该系统有来自MySQL 8.0的libmysqlclient.21，使用这样的命令。

shell> **rpm --oldpackage -ivh mysql-community-libs-5.5.50-2.el6.x86\_64.rpm**

调试包。 一个用调试包编译的MySQL服务器的特殊变体已经包含在服务器RPM包中。它执行调试和内存分配检查，并在服务器运行时产生一个跟踪文件。要使用该调试版本，用/usr/sbin/mysqld-debug启动MySQL，而不是作为服务或用/usr/sbin/mysqld启动。关于你可以使用的调试选项，见第5.9.4节，"DBUG包"。

注意

在MySQL 8.0.4中，调试构建的默认插件目录从/usr/lib64/mysql/plugin变为/usr/lib64/mysql/plugin/debug。以前，对于调试构建，必须将plugin\_dir改为/usr/lib64/mysql/plugin/debug。

从源码SRPMs重建RPMs。 MySQL的源代码SRPM包可供下载。它们可以按原样使用，用标准的rpmbuild工具链重建MySQL RPM。

### 2.5.5 使用Oracle的Debian软件包在Linux上安装MySQL

Oracle提供Debian软件包，用于在Debian或类Debian Linux系统上安装MySQL。这些软件包可以通过两个不同的渠道获得。

* MySQL APT资源库。这是在类Debian系统上安装MySQL的首选方法，因为它为安装和更新MySQL产品提供了一种简单而方便的方法。详情请参见第2.5.2节 "使用MySQL APT资源库在Linux上安装MySQL"。
* MySQL开发者专区的下载区。详情请见第2.1.3节 "如何获得MySQL"。下面是关于那里提供的Debian软件包的一些信息以及安装它们的说明。
  + 在MySQL开发区提供了各种Debian软件包，用于在当前的Debian和Ubuntu平台上安装MySQL的不同组件。首选的方法是使用tarball捆绑包，它包含了MySQL基本设置所需的包。压缩包的名称格式为mysql-server\_MVER-DVER\_CPU.deb-bundle.tar。 MVER是MySQL版本，DVER是Linux发行版本。CPU值表示软件包所对应的处理器类型或系列，如下表所示。

表2.13 MySQL Debian和Ubuntu安装包的CPU标识符

**Table 2.13 MySQL Debian and Ubuntu Installation Packages CPU Identifiers**

|  |  |
| --- | --- |
| ***CPU* Value** | **Intended Processor Type or Family** |
| **i386** | Pentium processor or better, 32 bit |
| **amd64** | 64-bit x86 processor |

* 在下载tarball后，用下面的命令解开它。

shell> **tar -xvf mysql-server\_*MVER*-*DVER*\_*CPU*.deb-bundle.tar**

* 如果你的系统上还没有libaio库，你可能需要安装它。

shell> **sudo apt-get install libaio1**

* 用下面的命令预先配置MySQL服务器包。

shell> **sudo dpkg-preconfigure mysql-community-server\_\*.deb**

你会被要求为你的MySQL安装提供一个根用户的密码。你可能还会被问到关于安装的其他问题。

重要提示

确保你记住你设置的根密码。想以后再设置密码的用户可以在对话框中把密码栏留空，只按确定；在这种情况下，对服务器的根访问是使用MySQL Socket Peer-Credential Authentication Plugin对使用Unix socket文件的连接进行验证的。你可以在以后使用mysql\_secure\_installation设置根密码。

* 对于MySQL服务器的基本安装，安装数据库公共文件包、客户端包、客户端元包、服务器包和服务器元包（按照这个顺序）；你可以用一个命令来做。

shell> **sudo dpkg -i mysql-{common,community-client,client,community-server,server}\_\*.deb**

还有一些软件包名称中带有server-core和client-core的软件包。这些包只包含二进制文件，并由标准包自动安装。安装它们本身并不会导致一个正常的MySQL设置。

如果你被dpkg警告有未满足的依赖关系，你可以用apt-get修复它们。

**sudo apt-get -f install**

这里是文件在系统上的安装位置。

* 所有的配置文件（比如my.cnf）都在/etc/mysql下
* 所有的二进制文件、库、头文件等，都在/usr/bin和/usr/sbin下

注意

Debian版的MySQL也由其他供应商提供。请注意，它们可能在特性、功能和惯例（包括通信设置）方面与甲骨文构建的版本不同，而且本手册中的说明不一定适用于安装它们。应该参考供应商的说明。

### 2.5.6 用Docker在Linux上部署MySQL

Docker部署框架支持MySQL服务器的简单安装和配置。本节解释了如何使用MySQL Server Docker镜像。

在使用MySQL Server Docker镜像之前，你需要在你的系统上安装Docker。请参阅安装Docker以获得说明。

重要信息

你需要用sudo运行docker命令，或者创建一个docker用户组，然后将任何想要运行docker命令的用户添加到其中。详情请见这里。由于Docker容器总是以root权限运行，你应该了解Docker守护程序的攻击面，并适当降低相关风险。

#### 2.5.6.1 使用Docker部署MySQL服务器的基本步骤

警告

由MySQL团队维护的MySQL Docker镜像是专门为Linux平台构建的。其他平台不受支持，用户在这些平台上使用这些MySQL Docker映像，其风险自负。关于在非Linux操作系统上运行这些容器的一些已知限制，请参见这里的讨论。

下载MySQL服务器Docker镜像

重要信息

对于MySQL企业版的用户。要使用MySQL企业版的Docker镜像，需要订阅。订阅通过自带许可证模式工作；详情请参见如何购买MySQL产品和服务。

在一个单独的步骤中下载服务器镜像并不是严格必要的；然而，在你创建Docker容器之前执行这个步骤可以确保你的本地镜像是最新的。要下载MySQL社区版镜像，请运行此命令。

**docker pull mysql/mysql-server:*tag***

标签是你想拉的镜像版本的标签（例如，5.6、5.7、8.0或最新）。如果省略了:tag，就会使用最新的标签，并下载MySQL Community Server的最新GA版本的图像。请参考Docker Hub中mysql/mysql-server页面上的可用版本的标签列表。

要从Oracle容器注册中心（OCR）下载MySQL社区版的镜像，请运行此命令。

**docker pull container-registry.oracle.com/mysql/mysql-server:*tag***

要从OCR下载MySQL企业版镜像，你需要首先接受OCR上的许可协议，并使用你的Docker客户端登录到容器仓库。

* 访问OCR：https://container-registry.oracle.com/，并选择MySQL。
* 在MySQL存储库列表下，选择enterpris-server。
* 如果你还没有登录到OCR，点击页面右侧的登录按钮，然后在提示下输入你的Oracle账户凭证。
* 按照页面右侧的指示，接受许可协议。
* 用你的Docker客户端（docker命令）登录OCR，使用docker登录命令。

# **docker login container-registry.oracle.com**

Username: ***Oracle-Account-ID***

Password: ***password***

Login successful.

用这个命令从OCR下载MySQL企业版的Docker镜像。

**docker pull container-registry.oracle.com/mysql/enterprise-server:*tag***

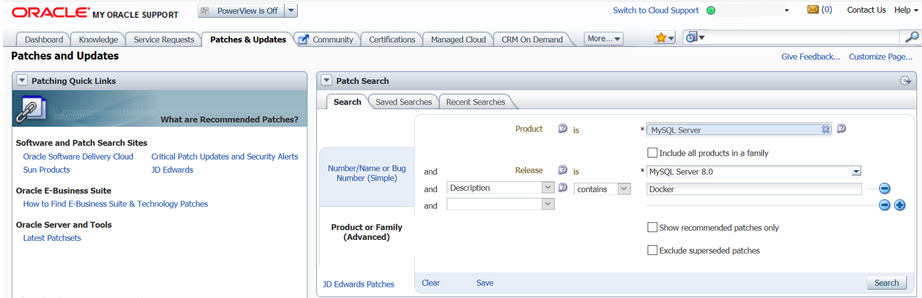
标签有不同的选择，对应于OCR提供的MySQL Docker镜像的不同版本。

* 8.0、8.0.x（x是8.0系列中的最新版本号）、最新。MySQL 8.0，最新的GA
* 5.7，5.7.y（y是5.7系列中的最新版本号）。MySQL 5.7

要从 "我的Oracle支持 "网站下载MySQL企业版镜像，请进入该网站，登录到你的Oracle账户，一旦你进入登陆页面，请执行这些步骤。

* 选择 "补丁和更新 "标签。
* 进入补丁搜索区域，在搜索标签上，切换到产品或家族（高级）子标签。
* 在产品字段中输入 "MySQL Server"，在版本字段中输入所需的版本号。
* 使用附加过滤器的下拉菜单，选择 "描述-包含"，并在文本字段中输入 "Docker"。

下图显示了MySQL Server 8.0的MySQL企业版图像的搜索设置。



* 点击搜索按钮，从结果列表中选择你想要的版本，并点击下载按钮。
* 在出现的文件下载对话框中，点击并下载Docker镜像的.zip文件。

解压缩下载的.zip档案，以获得里面的tar球（mysql-enterprise-server-version.tar），然后通过运行此命令加载镜像。

**docker load -i mysql-enterprise-server-*version*.tar**

你可以用这个命令列出下载的Docker镜像。

shell> docker images

shell> **docker images**

REPOSITORY TAG IMAGE ID CREATED SIZE

mysql/mysql-server latest 3157d7f55f8d 4 weeks ago 241MB

启动MySQL服务器实例

要为MySQL服务器启动一个新的Docker容器，使用以下命令。

**docker run --name=*container\_name* --restart on-failure -d *image\_name*:*tag***

镜像名称可以使用docker images命令获得，如下载MySQL服务器Docker镜像中所解释的。

--name选项用于为你的服务器容器提供一个自定义名称，是可选的；如果没有提供容器名称，就会生成一个随机名称。

--restart选项用于配置你的容器的重启策略；它应该被设置为on-failure值，以便在客户端会话中支持服务器重启（例如，当客户端执行RESTART语句或在配置InnoDB集群实例时发生）。启用对重启的支持后，在客户端会话中发出重启会导致服务器和容器停止，然后重新启动。对服务器重启的支持适用于MySQL 8.0.21及更高版本。

例如，要为MySQL社区服务器启动一个新的Docker容器，使用这个命令。

**docker run --name=mysql1 --restart on-failure -d mysql/mysql-server:8.0**

要使用从OCR下载的Docker镜像为MySQL企业服务器启动一个新的Docker容器，请使用此命令。

**docker run --name=mysql1 --restart on-failure -d container-registry.oracle.com/mysql/enterprise-server:8.0**

要使用从My Oracle Support下载的Docker镜像为MySQL企业服务器启动一个新的Docker容器，请使用此命令。

要用从OCR下载的Docker镜像为MySQL企业服务器启动一个新的Docker容器，请使用此命令。

**docker run --name=mysql1 --restart on-failure -d container-registry.oracle.com/mysql/enterprise-server:8.0**

要使用从My Oracle Support下载的Docker镜像为MySQL企业服务器启动一个新的Docker容器，请使用此命令。

**docker run --name=mysql1 --restart on-failure -d mysql/enterprise-server:8.0**

如果指定名称和标签的Docker镜像没有被先前的docker pull或docker run命令下载，那么现在就会下载该镜像。容器的初始化开始了，当你运行docker ps命令时，该容器会出现在正在运行的容器列表中。比如说

shell> **docker ps**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

a24888f0d6f4 mysql/mysql-server "/entrypoint.sh my..." 14 seconds ago Up 13 seconds (health: starting) 3306/tcp, 33060/tcp mysql1

容器初始化可能需要一些时间。当服务器可以使用时，docker ps命令输出中的容器STATUS会从（健康：开始）变为（健康）。

上面的docker run命令中使用的-d选项使容器在后台运行。使用此命令来监控容器的输出。

**docker logs mysql1**

一旦初始化完成，该命令的输出将包含为根用户生成的随机密码；例如，用这个命令检查密码。

shell> **docker logs mysql1 2>&1 | grep GENERATED**

GENERATED ROOT PASSWORD: Axegh3kAJyDLaRuBemecis&EShOs

从容器内连接到MySQL服务器

一旦服务器准备好了，你可以在你刚刚启动的MySQL服务器容器内运行mysql客户端，并将其连接到MySQL服务器。使用docker exec -it命令在你启动的Docker容器内启动一个mysql客户端，像下面这样。

**docker exec -it mysql1 mysql -uroot -p**

当被要求时，输入生成的根密码（见上面启动MySQL服务器实例的最后一步，如何找到密码）。因为MYSQL\_ONETIME\_PASSWORD选项默认为真，在你将一个mysql客户端连接到服务器后，你必须通过发出这个语句来重置服务器根密码。

mysql> **ALTER USER 'root'@'localhost' IDENTIFIED BY '*password*';**

用你选择的密码来代替密码。一旦密码被重置，服务器就可以使用了。

容器外壳访问

要对你的MySQL服务器容器进行shell访问，使用docker exec -it命令在容器内启动一个bash shell。

shell> **docker exec -it mysql1 bash**

bash-4.2#

然后你可以在容器内运行Linux命令。例如，要在容器内查看服务器的数据目录中的内容，使用这个命令。

bash-4.2# **ls /var/lib/mysql**

auto.cnf ca.pem client-key.pem ib\_logfile0 ibdata1 mysql mysql.sock.lock private\_key.pem server-cert.pem sys

ca-key.pem client-cert.pem ib\_buffer\_pool ib\_logfile1 ibtmp1 mysql.sock performance\_schema public\_key.pem server-key.pem

停止和删除一个MySQL容器

要停止我们已经创建的MySQL服务器容器，请使用此命令。

**docker stop mysql1**

docker stop向mysqld进程发送一个SIGTERM信号，这样服务器就被优雅地关闭了。

还要注意的是，当一个容器的主进程（在MySQL服务器容器的情况下是mysqld）被停止时，Docker容器会自动停止。

要再次启动MySQL服务器容器

**docker start mysql1**

要用一条命令停止和再次启动MySQL服务器容器

**docker restart mysql1**

要删除MySQL容器，首先停止它，然后使用docker rm命令。

**docker stop mysql1**

**docker rm mysql1**

如果你想同时删除服务器数据目录的Docker卷，在docker rm命令中添加-v选项。

升级MySQL服务器容器

重要提示

在对MySQL进行任何升级之前，请仔细遵循第2.11节 "升级MySQL "中的说明。在那里讨论的其他指示中，在升级前备份你的数据库是特别重要的。

本节中的说明要求服务器的数据和配置已经在主机上被持久化。详情请看持久化数据和配置变化。

按照这些步骤，将MySQL 5.7的Docker安装升级到8.0。

* 停止MySQL 5.7服务器（容器名称在本例中为mysql57）。

**docker stop mysql57**

* 下载MySQL 8.0服务器Docker镜像。见下载MySQL服务器Docker镜像中的说明；确保你为MySQL 8.0使用正确的标签。
* 启动一个新的MySQL 8.0 Docker容器（在本例中命名为mysql80），使用已经在主机上持久存在的旧服务器数据和配置（如果需要的话，进行适当的修改--见第2.11节 "升级MySQL"）。对于MySQL社区服务器，运行这个命令。
* **docker run --name=mysql80 \**
* **--mount type=bind,src=/path-on-host-machine/my.cnf,dst=/etc/my.cnf \**
* **--mount type=bind,src=/path-on-host-machine/datadir,dst=/var/lib/mysql \**
* **-d mysql/mysql-server:8.0**

如果需要，将mysql/mysql-server调整为正确的映像名称--例如，对于从OCR下载的MySQL企业版映像，将其替换为container-registry.oracle.com/mysql/enterprise-server；对于从My Oracle Support下载的MySQL企业版映像，则替换为mysql/enterprise-server。

* 等待服务器完成启动。你可以使用docker ps命令检查服务器的状态（关于如何做，请参阅启动MySQL服务器实例）。
* 对于MySQL 8.0.15和更早的版本。在MySQL 8.0服务器容器中运行mysql\_upgrade工具（对于MySQL 8.0.16及更高版本不需要）。
* **docker exec -it mysql80 mysql\_upgrade -uroot -p**

当出现提示时，输入旧的MySQL 5.7服务器的根密码。

通过重新启动MySQL 8.0服务器容器完成升级。

**docker restart mysql80**

关于用Docker部署MySQL服务器的更多主题

关于用Docker部署MySQL服务器的更多主题，如服务器配置、持久化数据和配置、服务器错误日志和容器环境变量，请参见第2.5.6.2节 "关于用Docker部署MySQL服务器的更多主题"。

#### 2.5.6.2 关于用Docker部署MySQL服务器的更多主题

注意

下面的大多数示例命令都将mysql/mysql-server作为必须指定的Docker映像库（如docker pull和docker run命令）；如果你的映像来自其他映像库，请将其改为container-registry.oracle.com/mysql/enterprise-server，用于从Oracle容器注册中心（OCR）下载的MySQL企业版映像，或用于从My Oracle Support下载的MySQL企业版映像的mysql/enterprise-server。

为Docker优化的MySQL安装

用于MySQL的Docker镜像针对代码大小进行了优化，这意味着它们只包括预计与在Docker容器中运行MySQL实例的大多数用户有关的关键组件。MySQL Docker安装与普通的、非Docker安装在以下方面有所不同。

包括的二进制文件仅限于。

* /usr/bin/my\_print\_defaults
* /usr/bin/mysql
* /usr/bin/mysql\_config
* /usr/bin/mysql\_install\_db
* /usr/bin/mysql\_tzinfo\_to\_sql
* /usr/bin/mysql\_upgrade
* /usr/bin/mysqladmin
* /usr/bin/mysqlcheck
* /usr/bin/mysqldump
* /usr/bin/mysqlpump
* /usr/bin/mysqlbackup（仅适用于MySQL企业版8.0）。
* /usr/sbin/mysqld

所有二进制文件都被剥离；它们不包含任何调试信息。

配置MySQL服务器

当你启动MySQL Docker容器时，你可以通过docker运行命令将配置选项传递给服务器。比如说

**docker run --name mysql1 -d mysql/mysql-server:*tag* --character-set-server=utf8mb4 --collation-server=utf8mb4\_col**

该命令以utf8mb4作为数据库的默认字符集和utf8mb4\_col作为数据库的默认整理方式启动你的MySQL服务器。

配置MySQL服务器的另一种方法是准备一个配置文件并将其挂载在容器内的服务器配置文件的位置。详情请参见持久化数据和配置更改。

保存数据和配置变更

Docker容器原则上是短暂的，如果容器被删除或损坏，任何数据或配置都会丢失（见这里的讨论）。然而，Docker卷提供了一种机制来持久化在Docker容器内创建的数据。在初始化时，MySQL Server容器为服务器数据目录创建一个Docker卷。在容器上运行docker inspect命令的JSON输出有一个Mount键，其值提供了关于数据目录卷的信息。

shell> **docker inspect mysql1**

...

"Mounts": [

{

"Type": "volume",

"Name": "4f2d463cfc4bdd4baebcb098c97d7da3337195ed2c6572bc0b89f7e845d27652",

"Source": "/var/lib/docker/volumes/4f2d463cfc4bdd4baebcb098c97d7da3337195ed2c6572bc0b89f7e845d27652/\_data",

"Destination": "/var/lib/mysql",

"Driver": "local",

"Mode": "",

"RW": true,

"Propagation": ""

}

],

...

输出显示，源文件夹/var/lib/docker/volumes/4f2d463cfc4bdd4baebcb098c97d7da3337195ed2c6572bc0b89f7e845d27652/\_data，其中的数据在主机上被持久化，已经被挂载到/var/lib/mysql，即容器内部的服务器数据目录。

另一种保存数据的方法是在创建容器时使用-mount选项绑定挂载一个主机目录。同样的技术也可以用来保存服务器的配置。下面的命令创建了一个MySQL服务器容器，并绑定挂载了数据目录和服务器配置文件。

**docker run --name=mysql1 \**

**--mount type=bind,src=*/path-on-host-machine/my.cnf*,dst=/etc/my.cnf \**

**--mount type=bind,src=*/path-on-host-machine/datadir*,dst=/var/lib/mysql \**

**-d mysql/mysql-server:*tag***

该命令将path-on-host-machine/my.cnf挂载在/etc/my.cnf（容器内的服务器配置文件），并将path-on-host-machine/datadir挂载在/var/lib/mysql（容器内的数据目录）。绑定挂载必须满足以下条件才能发挥作用。

* 配置文件 path-on-host-machine/my.cnf 必须已经存在，并且它必须包含使用用户 mysql 启动服务器的规范。
* [mysqld]
* user=mysql

你也可以在该文件中包括其他服务器配置选项。

* 数据目录path-on-host-machine/datadir必须已经存在。为了使服务器初始化发生，该目录必须是空的。你也可以挂载一个预先填充了数据的目录，然后用它来启动服务器；然而，你必须确保以与创建数据的服务器相同的配置来启动Docker容器，并且在启动容器时挂载任何需要的主机文件或目录。

运行额外的初始化脚本

如果有任何.sh或.sql脚本你想在数据库创建后立即运行，你可以把它们放入一个主机目录，然后在容器内的/docker-entrypoint-initdb.d/挂载该目录。比如说

**docker run --name=mysql1 \**

**--mount type=bind,src=*/path-on-host-machine/scripts/*,dst=/docker-entrypoint-initdb.d/ \**

**-d mysql/mysql-server:*tag***

从另一个Docker容器中的应用程序连接到MySQL

通过设置Docker网络，你可以允许多个Docker容器相互通信，这样另一个Docker容器中的客户端应用程序就可以访问服务器容器中的MySQL服务器。首先，创建一个Docker网络。

**docker network create *my-custom-net***

然后，当你创建并启动服务器和客户端容器时，使用--network选项将它们放在你创建的网络上。比如说

**docker run --name=mysql1 --network=*my-custom-net* -d mysql/mysql-server**

**docker run --name=myapp1 --network=*my-custom-net* -d myapp**

然后，myapp1容器可以用mysql1的主机名连接到mysql1容器，反之亦然，因为Docker自动为给定的容器名设置了一个DNS。在下面的例子中，我们从myapp1容器内部运行mysql客户端，以连接到它自己容器中的主机mysql1。

**docker exec -it myapp1 mysql --host=mysql1 --user=myuser --password**

关于容器的其他网络技术，请参阅Docker文档中的Docker容器网络部分。

服务器错误日志

当MySQL服务器首次与你的服务器容器一起启动时，如果以下任一条件为真，则不会生成服务器错误日志。

* 来自主机的服务器配置文件已被挂载，但该文件不包含系统变量log\_error（参见关于绑定挂载服务器配置文件的持续数据和配置更改）。
* 来自主机的服务器配置文件没有被挂载，但Docker环境变量MYSQL\_LOG\_CONSOLE为真（这是MySQL 8.0服务器容器的变量的默认状态）。然后，MySQL服务器的错误日志被重定向到stderr，这样错误日志就会进入Docker容器的日志，并可使用docker logs mysqld-tainer命令查看。

要使MySQL服务器在两个条件中的任何一个为真时生成错误日志，使用-log-error选项来配置服务器在容器内的特定位置生成错误日志。要持久化错误日志，请在容器内错误日志的位置挂载一个主机文件，如持久化数据和配置更改中所解释的。然而，你必须确保你的MySQL服务器在其容器内有对挂载的主机文件的写入权限。

在Docker中使用MySQL企业备份

MySQL Enterprise Backup是一个商业许可的MySQL Server备份工具，可与MySQL企业版一起使用。MySQL企业备份包含在MySQL企业版的Docker安装中

在下面的例子中，我们假设你已经有一个在Docker容器中运行的MySQL服务器（参见第2.5.6.1节，"用Docker部署MySQL服务器的基本步骤"，关于如何用Docker启动MySQL服务器实例）。为了使MySQL企业备份能够备份MySQL服务器，它必须能够访问该服务器的数据目录。例如，这可以通过在启动MySQL服务器时在其数据目录上绑定挂载一个主机目录来实现。

**docker run --name=mysqlserver \**

**--mount type=bind,src=*/path-on-host-machine/datadir/*,dst=/var/lib/mysql \**

**-d mysql/enterprise-server:8.0**

通过这个命令，MySQL服务器以MySQL企业版的Docker镜像启动，主机目录/path-on-host-machine/datadir/已经被挂载到服务器容器内的数据目录（/var/lib/mysql）。我们还假设，在服务器启动后，也为 MySQL 企业备份设置了访问服务器所需的权限（详见授予备份管理员 MySQL 权限）。然后使用以下步骤来备份和恢复MySQL服务器实例。

要使用MySQL Enterprise Backup with Docker来备份在Docker容器中运行的MySQL Server实例。

1、在运行MySQL Server容器的同一主机上，用MySQL企业版的镜像启动另一个容器，用MySQL企业备份命令backup-to-image进行备份。使用我们在上一步创建的绑定挂载提供对服务器的数据目录的访问。同时，将主机目录（本例中为/path-on-host-machine/backups/）挂载到容器中备份的存储文件夹上（本例中为/data/backups），以保持我们正在创建的备份的持续性。下面是这一步的示例命令，其中MySQL企业备份是用从My Oracle支持中心下载的Docker镜像启动的。)

shell> **docker run \**

**--mount type=bind,src=*/path-on-host-machine/datadir/*,dst=/var/lib/mysql \**

**--mount type=bind,src=*/path-on-host-machine/backups/*,dst=/data/backups \**

**--rm mysql/enterprise-server:8.0 \**

**mysqlbackup -u*mysqlbackup* -p*password* --backup-dir=/tmp/backup-tmp --with-timestamp \**

**--backup-image=/data/backups/db.mbi backup-to-image**

[Entrypoint] MySQL Docker Image 8.0.11-1.1.5

MySQL Enterprise Backup version 8.0.11 Linux-4.1.12-61.1.16.el7uek.x86\_64-x86\_64 [2018-04-08 07:06:45]

Copyright (c) 2003, 2018, Oracle and/or its affiliates. All Rights Reserved.

180921 17:27:25 MAIN INFO: A thread created with Id '140594390935680'

180921 17:27:25 MAIN INFO: Starting with following command line ...

...

-------------------------------------------------------------

Parameters Summary

-------------------------------------------------------------

Start LSN : 29615616

End LSN : 29651854

-------------------------------------------------------------

mysqlbackup completed OK!

检查mysqlbackup输出的结尾是很重要的，以确保备份已经成功完成。

2、一旦备份工作完成，容器就会退出，而且，用--rm选项来启动它，它退出后就会被删除。一个镜像备份已经被创建，可以在上一步安装的用于存储备份的主机目录中找到。

shell> ls /tmp/backups

db.mbi

要使用MySQL Enterprise Backup with Docker恢复Docker容器中的MySQL Server实例。

1、停止MySQL服务器容器，这也会停止里面运行的MySQL服务器。

docker stop mysqlserver

2、在主机上，删除MySQL服务器数据目录的绑定挂载中的所有内容。

rm -rf ***/path-on-host-machine/datadir***/\*

3、用MySQL企业版的映像启动一个容器，用MySQL企业备份命令copy-back-apply-log执行还原。像我们备份服务器时那样，绑定挂载服务器的数据目录和备份的存储文件夹。

shell> **docker run \**

**--mount type=bind,src=*/path-on-host-machine/datadir/*,dst=/var/lib/mysql \**

**--mount type=bind,src=*/path-on-host-machine/backups/*,dst=/data/backups \**

**--rm mysql/enterprise-server:8.0 \**

**mysqlbackup --backup-dir=/tmp/backup-tmp --with-timestamp \**

**--datadir=/var/lib/mysql --backup-image=/data/backups/db.mbi copy-back-and-apply-log**

[Entrypoint] MySQL Docker Image 8.0.11-1.1.5

MySQL Enterprise Backup version 8.0.11 Linux-4.1.12-61.1.16.el7uek.x86\_64-x86\_64 [2018-04-08 07:06:45]

Copyright (c) 2003, 2018, Oracle and/or its affiliates. All Rights Reserved.

180921 22:06:52 MAIN INFO: A thread created with Id '139768047519872'

180921 22:06:52 MAIN INFO: Starting with following command line ...

...

180921 22:06:52 PCR1 INFO: We were able to parse ibbackup\_logfile up to

lsn 29680612.

180921 22:06:52 PCR1 INFO: Last MySQL binlog file position 0 155, file name binlog.000003

180921 22:06:52 PCR1 INFO: The first data file is '/var/lib/mysql/ibdata1'

and the new created log files are at '/var/lib/mysql'

180921 22:06:52 MAIN INFO: No Keyring file to process.

180921 22:06:52 MAIN INFO: Apply-log operation completed successfully.

180921 22:06:52 MAIN INFO: Full Backup has been restored successfully.

mysqlbackup completed OK! with 3 warnings

一旦备份工作完成，该容器就会退出，在启动它时使用--rm选项，它退出后会被删除。

4、重新启动服务器容器，这也会重新启动恢复的服务器。

**docker restart mysqlserver**

或者，在恢复的数据目录上启动一个新的MySQL服务器。

**docker run --name=mysqlserver2 \**

**--mount type=bind,src=*/path-on-host-machine/datadir/*,dst=/var/lib/mysql \**

**-d mysql/enterprise-server:8.0**

登录到服务器上，检查服务器是否正在使用恢复后的数据运行。

已知问题

* 当使用服务器系统变量audit\_log\_file配置审计日志文件名时，请使用松散选项修改器，否则Docker将无法启动服务器。

Docker环境变量

当你创建MySQL服务器容器时，你可以通过使用--env选项（简称-e）并指定以下一个或多个环境变量来配置MySQL实例。

注意事项

* 如果你挂载的数据目录不是空的，下面的变量都没有任何作用，因为那时不会尝试进行服务器初始化（更多细节请参见持久化数据和配置更改）。文件夹中任何先前存在的内容，包括任何旧的服务器设置，在容器启动期间都不会被修改。
* 包括MYSQL\_RANDOM\_ROOT\_PASSWORD、MYSQL\_ONETIME\_PASSWORD、MYSQL\_ALLOW\_EMPTY\_PASSWORDMYSQL\_LOG\_CONSOLE在内的布尔变量，通过将它们设置为非零长度的任何字符串而使之为真。因此，将它们设置为例如 "0"、"false "或 "no "并不会使它们变成假的，而是实际上使它们变成真的。这是MySQL服务器容器的一个已知问题。
* MYSQL\_RANDOM\_ROOT\_PASSWORD：当这个变量为真时（这是它的默认状态，除非MYSQL\_ROOT\_PASSWORD被设置或MYSQL\_ALLOW\_EMPTY\_PASSWORD被设置为真），当Docker容器被启动时，会生成一个服务器根用户的随机密码。该密码被打印到容器的stdout，并且可以通过查看容器的日志找到（见启动MySQL服务器实例）。
* MYSQL\_ONETIME\_PASSWORD：当该变量为真时（这是它的默认状态，除非MYSQL\_ROOT\_PASSWORD被设置或MYSQL\_ALLOW\_EMPTY\_PASSWORD被设置为真），根用户的密码被设置为过期，必须在正常使用MySQL之前进行更改。
* MYSQL\_DATABASE：这个变量允许你指定在图像启动时创建的数据库的名称。如果在MYSQL\_USER和MYSQL\_PASSWORD中提供了用户名和密码，就会创建用户并授予该数据库的超级用户权限（对应于GRANT ALL）。指定的数据库是由CREATE DATABASE IF NOT EXIST语句创建的，所以如果数据库已经存在，这个变量就没有作用。
* MYSQL\_USER, MYSQL\_PASSWORD：这些变量结合使用，用于创建一个用户并设置该用户的密码，该用户被授予MYSQL\_DATABASE变量所指定数据库的超级用户权限。MYSQL\_USER和MYSQL\_PASSWORD都是创建用户的必要条件--如果这两个变量中的任何一个没有设置，另一个将被忽略。如果这两个变量都设置了，但MYSQL\_DATABASE没有设置，那么创建的用户就没有任何权限。

注意

没有必要使用这个机制来创建根超级用户，除非MYSQL\_ALLOW\_EMPTY\_PASSWORD为真，否则默认情况下，根超级用户的密码是由MYSQL\_ROOT\_PASSWORD和MYSQL\_RANDOM\_ROOT\_PASSWORD描述中讨论的任何一种机制设置的。

* MYSQL\_ROOT\_HOST：默认情况下，MySQL创建'root'@'localhost'账户。这个账户只能从容器内连接到，如在容器内连接到MySQL服务器中所述。要允许来自其他主机的root连接，请设置这个环境变量。例如，172.17.0.1这个值是默认的Docker网关IP，允许从运行容器的主机上进行连接。该选项只接受一个条目，但允许使用通配符（例如，MYSQL\_ROOT\_HOST=172.\*.\*.\*或MYSQL\_ROOT\_HOST=%）。
* MYSQL\_LOG\_CONSOLE：当该变量为真时（这是MySQL 8.0服务器容器的默认状态），MySQL服务器的错误日志会被重定向到stderr，这样错误日志就会进入Docker容器的日志，并可使用docker logs mysqld-tainer命令查看。

注意

如果来自主机的服务器配置文件已经被挂载，则该变量没有作用（参见关于绑定挂载配置文件的持久化数据和配置更改）。

MYSQL\_ROOT\_PASSWORD：该变量指定了为MySQL根账户设置的密码。

警告

在命令行上设置MySQL根用户密码是不安全的。作为明确指定密码的替代方法，你可以用一个密码文件的容器文件路径来设置变量，然后从你的主机挂载一个包含密码的文件到容器文件路径。这仍然不是很安全，因为密码文件的位置仍然暴露。最好是使用MYSQL\_RANDOM\_ROOT\_PASSWORD和MYSQL\_ONETIME\_PASSWORD这两个默认设置为真。

* mysql\_allow\_empty\_password。将其设置为 "true"，以允许容器以根用户的空白密码启动。

警告

将这个变量设置为 "true "是不安全的，因为它将使你的MySQL实例完全不受保护，允许任何人获得完全的超级用户访问。最好使用MYSQL\_RANDOM\_ROOT\_PASSWORD和MYSQL\_ONETIME\_PASSWORD这两个默认设置为真。

#### 2.5.6.3 用Docker在Windows和其他非Linux平台上部署MySQL

警告

甲骨文提供的MySQL Docker镜像是专门为Linux平台构建的。其他平台不受支持，用户在这些平台上运行甲骨文提供的MySQL Docker映像，其风险自负。本节讨论了在非Linux平台上使用图像时的一些已知问题。

在Windows上使用来自Oracle的MySQL服务器Docker镜像的已知问题包括。

* 如果你在容器的MySQL数据目录上进行绑定挂载（详见保存数据和配置变更），你必须用--socket选项将服务器套接字文件的位置设置为MySQL数据目录之外的地方；否则，服务器将无法启动。这是因为Docker for Windows处理文件挂载的方式不允许主机文件在套接字文件上被绑定挂载。

### 2.5.7 从本地软件库在Linux上安装MySQL

许多Linux发行版在其本地软件库中包括MySQL服务器、客户端工具和开发组件的版本，可以用平台的标准软件包管理系统进行安装。本节提供了使用这些软件包管理系统安装MySQL的基本说明。

重要事项

本地软件包通常比当前可用的版本晚几个版本。你通常也无法安装开发里程碑版本（DMR），因为这些版本通常不在本地存储库中提供。在继续之前，我们建议你检查一下第2.5节 "在Linux上安装MySQL "中描述的其他安装选项。

发行版的具体说明如下。

* Red Hat Linux, Fedora, CentOS

注意

对于一些Linux发行版，你可以使用MySQL Yum软件库而不是平台的本地软件库来安装MySQL。详情见第2.5.1节，"使用MySQL Yum仓库在Linux上安装MySQL"。

对于Red Hat和类似的发行版，MySQL发行版被分为若干独立的包，mysql用于客户端工具，mysql-server用于服务器和相关工具，mysql-libs用于库。如果你想从不同的语言和环境（如Perl、Python和其他）提供连接，则需要这些库。

要安装，使用yum命令来指定你要安装的包。比如说

root-shell> yum install mysql mysql-server mysql-libs mysql-server

Loaded plugins: presto, refresh-packagekit

Setting up Install Process

Resolving Dependencies

--> Running transaction check

---> Package mysql.x86\_64 0:5.1.48-2.fc13 set to be updated

---> Package mysql-libs.x86\_64 0:5.1.48-2.fc13 set to be updated

---> Package mysql-server.x86\_64 0:5.1.48-2.fc13 set to be updated

--> Processing Dependency: perl-DBD-MySQL for package: mysql-server-5.1.48-2.fc13.x86\_64

--> Running transaction check

---> Package perl-DBD-MySQL.x86\_64 0:4.017-1.fc13 set to be updated

--> Finished Dependency Resolution

Dependencies Resolved

================================================================================

Package Arch Version Repository Size

================================================================================

Installing:

mysql x86\_64 5.1.48-2.fc13 updates 889 k

mysql-libs x86\_64 5.1.48-2.fc13 updates 1.2 M

mysql-server x86\_64 5.1.48-2.fc13 updates 8.1 M

Installing for dependencies:

perl-DBD-MySQL x86\_64 4.017-1.fc13 updates 136 k

Transaction Summary

================================================================================

Install 4 Package(s)

Upgrade 0 Package(s)

Total download size: 10 M

Installed size: 30 M

Is this ok [y/N]: y

Downloading Packages:

Setting up and reading Presto delta metadata

Processing delta metadata

Package(s) data still to download: 10 M

(1/4): mysql-5.1.48-2.fc13.x86\_64.rpm | 889 kB 00:04

(2/4): mysql-libs-5.1.48-2.fc13.x86\_64.rpm | 1.2 MB 00:06

(3/4): mysql-server-5.1.48-2.fc13.x86\_64.rpm | 8.1 MB 00:40

(4/4): perl-DBD-MySQL-4.017-1.fc13.x86\_64.rpm | 136 kB 00:00

--------------------------------------------------------------------------------

Total 201 kB/s | 10 MB 00:52

Running rpm\_check\_debug

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Installing : mysql-libs-5.1.48-2.fc13.x86\_64 1/4

Installing : mysql-5.1.48-2.fc13.x86\_64 2/4

Installing : perl-DBD-MySQL-4.017-1.fc13.x86\_64 3/4

Installing : mysql-server-5.1.48-2.fc13.x86\_64 4/4

Installed:

mysql.x86\_64 0:5.1.48-2.fc13 mysql-libs.x86\_64 0:5.1.48-2.fc13

mysql-server.x86\_64 0:5.1.48-2.fc13

Dependency Installed:

perl-DBD-MySQL.x86\_64 0:4.017-1.fc13

Complete!

现在，MySQL和MySQL服务器应该已经安装完毕。一个配置文件的样本被安装到/etc/my.cnf中。使用systemctl来启动MySQL服务器。

shell> **systemctl start mysqld**

如果数据库表还不存在，会自动为你创建。然而，你应该运行 mysql\_secure\_installation 来设置服务器上的根密码。

* Debian、Ubuntu、Kubuntu

注意

对于支持的Debian和Ubuntu版本，MySQL可以使用MySQL APT资源库而不是平台的本地软件资源库来安装。详情请参见第2.5.2节，"使用MySQL APT资源库在Linux上安装MySQL"。

在Debian和相关发行版上，在其软件仓库中有两个MySQL的软件包，即mysql-client和mysql-server，分别用于客户端和服务器组件。你应该指定一个明确的版本，例如mysql-client-5.1，以确保你安装的是你想要的MySQL的版本。

要下载和安装，包括任何依赖项，使用apt-get命令，指定你要安装的包。

注意

在安装之前，确保你更新你的apt-get索引文件，以确保你下载的是最新的可用版本。

注意

apt-get命令安装了许多软件包，包括MySQL服务器，以便提供典型的工具和应用环境。这可能意味着，除了主要的MySQL包之外，你还要安装大量的包。

在安装过程中，初始数据库被创建，并提示你输入MySQL根密码（和确认）。一个配置文件在/etc/mysql/my.cnf中被创建。在/etc/init.d/mysql中创建了一个初始脚本。

服务器应该已经被启动了。你可以使用以下方法手动启动和停止服务器。

root-shell> service mysql [start|stop]

。

该服务被自动添加到2、3和4的运行级别，停止脚本在单一、关闭和重启级别。

### 2.5.8 用Juju在Linux上安装MySQL

Juju部署框架支持MySQL服务器的简易安装和配置。有关说明，请参见https://jujucharms.com/mysql/。

### 2.5.9 用systemd管理MySQL服务器

如果你在以下Linux平台上使用RPM或Debian软件包安装MySQL，服务器的启动和关闭由systemd管理。

* RPM包平台。
  + 企业版Linux变种7及以上版本
  + USE Linux Enterprise Server 12及以上版本
  + Fedora 29及更高版本
* Debian系列平台。
  + Debian平台
  + Ubuntu平台

如果你在一个使用systemd的平台上从通用二进制发行版安装MySQL，你可以按照《MySQL 8.0安全部署指南》中安装后设置部分提供的说明，手动配置systemd对MySQL的支持。

如果你在一个使用systemd的平台上从源码发行版安装MySQL，可以通过使用-DWITH\_SYSTEMD=1 CMake选项配置发行版来获得对MySQL的系统支持。参见第2.9.7节，"MySQL源码配置选项"。

下面的讨论涵盖了这些主题。

* systemd概述
* 为MySQL配置systemd
* 使用systemd配置多个MySQL实例
* 从mysqld\_safe迁移到systemd

注意

在安装了systemd支持MySQL的平台上，mysqld\_safe和System V初始化脚本等脚本是不必要的，不会被安装。例如，mysqld\_safe可以处理服务器重启，但systemd也提供了同样的功能，而且是以与其他服务的管理一致的方式，而不是通过使用特定的应用程序。

由于systemd具有在已安装systemd支持的平台上管理多个MySQL实例的能力，所以mysqld\_multi和mysqld\_multi.server是不必要的，不需要安装。

**systemd概述**

systemd提供了MySQL服务器的自动启动和关闭。它还可以使用systemctl命令手动管理服务器。举例来说

shell> **systemctl *{start|stop|restart|status}* mysqld**

另外，使用service命令（参数相反），它与System V系统兼容。

shell> **service mysqld *{start|stop|restart|status}***

注意

对于systemctl命令（和替代的服务命令），如果MySQL服务名称不是mysqld，则使用适当的名称。例如，在基于Debian和SLES的系统上，使用mysql而不是mysqld。

对systemd的支持包括这些文件。

* mysqld.service（RPM平台），mysql.service（Debian平台）：systemd服务单元配置文件，包含MySQL服务的详细信息。
* mysqld@.service（RPM平台），mysql@.service（Debian平台）。与mysqld.service或mysql.service类似，但用于管理多个MySQL实例。
* mysqld.tmpfiles.d。文件包含支持临时文件功能的信息。该文件以mysql.conf的名义安装。
* mysqld\_pre\_systemd（RPM平台），mysql-system-start（Debian平台）。支持单元文件的脚本。该脚本仅在日志位置符合某种模式（RPM平台为/var/log/mysql\*.log，Debian平台为/var/log/mysql/\*.log）时协助创建错误日志文件。在其他情况下，错误日志目录必须是可写的，或者错误日志必须对运行mysqld进程的用户存在并可写。

**为MySQL配置systemd**

要为MySQL添加或更改systemd选项，有以下几种方法可供选择。

* 使用本地化的 systemd 配置文件。
* 安排systemd为MySQL服务器进程设置环境变量。
* 设置MYSQLD\_OPTS的systemd变量。

要使用本地化的systemd配置文件，请创建/etc/systemd/system/mysqld.service.d目录（如果它不存在）。在该目录下，创建一个包含[Service]部分的文件，列出所需设置。比如说

[Service]

LimitNOFILE=***max\_open\_files***

Nice=***nice\_level***

LimitCore=***core\_file\_limit***

Environment="LD\_PRELOAD=***/path/to/malloc/library***"

Environment="TZ=***time\_zone\_setting***"

这里的讨论使用override.conf作为该文件的名称。新版本的 systemd 支持以下命令，打开一个编辑器，允许你编辑该文件。

systemctl edit mysqld # RPM platforms

systemctl edit mysql # Debian platforms

每当你创建或修改override.conf时，都要重新加载systemd配置，然后告诉systemd重新启动MySQL服务。

systemctl daemon-reload

systemctl restart mysqld # RPM platforms

systemctl restart mysql # Debian platforms

在systemd中，某些参数必须使用override.conf配置方法，而不是在MySQL选项文件的[mysqld]、[mysqld\_safe]或[safe\_mysqld]组中设置。

* 对于某些参数，必须使用override.conf，因为systemd本身必须知道它们的值，而它无法读取MySQL选项文件来获得这些值。
* 对于那些只能用mysqld\_safe知道的选项来设置的参数，必须用systemd来指定，因为没有相应的mysqld参数。

关于使用systemd而不是mysqld\_safe的其他信息，请参见从mysqld\_safe迁移到systemd。

你可以在override.conf中设置以下参数。

* 要设置MySQL服务器可用的文件描述符的数量，在override.conf中使用LimitNOFILE，而不是mysqld的open\_files\_limit系统变量或mysqld\_safe的--open-files-limit选项。
* 要设置最大的核心文件大小，使用override.conf中的LimitCore，而不是mysqld\_safe的--核心文件大小选项。
* 要设置MySQL服务器的调度优先级，在override.conf中使用Nice而不是mysqld\_safe的--nice选项。

一些MySQL参数是用环境变量配置的。

* LD\_PRELO。如果MySQL服务器应该使用一个特定的内存分配库，则设置这个变量。
* NOTIFY\_SOCKET：这个环境变量指定了mysqld用来与systemd交流启动完成和服务状态变化通知的套接字。该变量由systemd在启动mysqld服务时设置。mysqld服务会读取该变量设置并写入定义的位置。

在MySQL 8.0中，mysqld使用Type=notify进程启动类型。(MySQL 5.7中使用Type=forking。)在Type=notify下，systemd自动配置一个套接字文件，并将路径导出到NOTIFY\_SOCKET环境变量。

* TZ：设置该变量以指定服务器的默认时区。

有多种方法可以指定环境变量值供systemd管理的MySQL服务器进程使用。

* 使用override.conf文件中的环境行。语法见前面描述如何使用该文件的讨论中的例子。
* 在/etc/sysconfig/mysql文件中指定数值（如果该文件不存在，则创建该文件）。使用以下语法指定数值。
* LD\_PRELOAD=***/path/to/malloc/library***
* TZ=***time\_zone\_setting***

在修改 /etc/sysconfig/mysql 后，重新启动服务器以使修改生效。

systemctl restart mysqld # RPM platforms

systemctl restart mysql # Debian platforms

要指定mysqld的选项而不直接修改systemd配置文件，可以设置或不设置MYSQLD\_OPTS systemd变量。例如

systemctl set-environment MYSQLD\_OPTS="--general\_log=1"

systemctl unset-environment MYSQLD\_OPTS

MYSQLD\_OPTS 也可以在 /etc/sysconfig/mysql 文件中设置。

修改systemd环境后，重新启动服务器，使修改生效。

systemctl restart mysqld # RPM platforms

systemctl restart mysql # Debian platforms

对于使用systemd的平台，如果服务器启动时是空的，那么数据目录就会被初始化。如果数据目录是一个暂时消失的远程挂载点，这可能是个问题。挂载点会显示为一个空的数据目录，然后会被初始化为一个新的数据目录。要抑制这种自动初始化行为，请在/etc/sysconfig/mysql文件中指定以下一行（如果该文件不存在，则创建该文件）。

NO\_INIT=true

**使用systemd配置多个MySQL实例**

本节介绍了如何为MySQL的多个实例配置systemd。

注意

由于systemd具有在已安装systemd支持的平台上管理多个MySQL实例的能力，所以mysqld\_multi和mysqld\_multi.server是不必要的，不需要安装。

要使用多实例功能，需要修改my.cnf选项文件，包括配置每个实例的关键选项。这些文件位置是典型的。

* /etc/my.cnf或/etc/mysql/my.cnf（RPM平台
* /etc/mysql/mysql.conf.d/mysqld.cnf (Debian平台)

例如，要管理两个名为replica01和replica02的实例，在选项文件中添加类似这样的内容。

[mysqld@replica01]

datadir=/var/lib/mysql-replica01

socket=/var/lib/mysql-replica01/mysql.sock

port=3307

log-error=/var/log/mysqld-replica01.log

[mysqld@replica02]

datadir=/var/lib/mysql-replica02

socket=/var/lib/mysql-replica02/mysql.sock

port=3308

log-error=/var/log/mysqld-replica02.log

Debian平台。

[mysqld@replica01]

datadir=/var/lib/mysql-replica01

socket=/var/lib/mysql-replica01/mysql.sock

port=3307

log-error=/var/log/mysql/replica01.log

[mysqld@replica02]

datadir=/var/lib/mysql-replica02

socket=/var/lib/mysql-replica02/mysql.sock

port=3308

log-error=/var/log/mysql/replica02.log

这里显示的副本名称使用@作为分隔符，因为这是 systemd 唯一支持的分隔符。

然后通过正常的systemd命令来管理实例，例如。

systemctl start mysqld@replica01

systemctl start mysqld@replica02

要使实例在启动时运行，请这样做。

systemctl enable mysqld@replica01

systemctl enable mysqld@replica02

也支持使用通配符。例如，该命令显示所有复制实例的状态。

systemctl status 'mysqld@replica\*'

对于同一台机器上的多个MySQL实例的管理，systemd自动使用不同的单元文件。

* mysqld@.service而不是mysqld.service（RPM平台
* mysql@.service，而不是 mysql.service（Debian平台）。

在单元文件中，%I和%i引用@标记后传入的参数，用于管理特定的实例。对于像这样的命令

systemctl start mysqld@replica01

systemd启动服务器，使用这样的命令。

mysqld --defaults-group-suffix=@%I ...

结果是[server]、[mysqld]和[mysqld@replica01]选项组被读取并用于该服务的实例。

注意

在 Debian 平台上，AppArmor 阻止服务器读取或写入 /var/lib/mysql-replica\*，或除默认位置之外的任何内容。要解决这个问题，你必须自定义或禁用 /etc/apparmor.d/usr.sbin.mysqld 中的配置文件。

注意

在 Debian 平台上，用于 MySQL 卸载的打包脚本目前不能处理 mysqld@ 实例。在删除或升级软件包之前，你必须先手动停止任何额外的实例。

**从 mysqld\_safe 迁移到 systemd**

因为mysqld\_safe不安装在使用systemd管理MySQL的平台上，所以以前为该程序指定的选项（例如，在[mysqld\_safe]或[safe\_mysqld]选项组中）必须以其他方式指定。

* 一些mysqld\_safe选项也能被mysqld理解，可以从[mysqld\_safe]或[safe\_mysqld]选项组移到[mysqld]组。这不包括--pid-file、--open-files-limit或--nice。要指定这些选项，请使用override.conf systemd文件，如前所述。

注意

在systemd平台上，不支持使用[mysqld\_safe]和[safe\_mysqld]选项组，可能会导致意外行为。

* 对于某些 mysqld\_safe 选项，有替代的 mysqld 程序。例如，启用syslog日志的mysqld\_safe选项是--syslog，它已被废弃。要将错误日志输出写入系统日志，请使用第5.4.2.8节 "将错误日志写入系统日志 "中的说明。
* mysqld\_safe选项不被mysqld理解，可以在override.conf或环境变量中指定。例如，在mysqld\_safe中，如果服务器应该使用一个特定的内存分配库，可以使用--malloc-lib选项指定。对于用systemd管理服务器的安装，安排设置LD\_PRELOAD环境变量代替，如前所述。

## 2.6 使用Unbreakable Linux Network（ULN）安装MySQL

Linux支持许多不同的解决方案来安装MySQL，这在第2.5节 "在Linux上安装MySQL "中有所涉及。本节涉及的其中一种方法是从Oracle的Unbreakable Linux Network（ULN）中安装。你可以在http://linux.oracle.com/ 下找到关于Oracle Linux和ULN的信息。

要使用ULN，你需要获得一个ULN登录账号，并在ULN上注册用于安装的机器。这在 ULN FAQ 中有详细描述。该页面还介绍了如何安装和更新软件包。

社区包和商业包都被支持，每个包都提供三个MySQL通道。

* 服务器。MySQL服务器
* 连接器。MySQL Connector/C++、MySQL Connector/J、MySQL Connector/ODBC和MySQL Connector/Python。
* 工具。MySQL Router、MySQL Shell和MySQL Workbench

所有ULN用户都可以使用社区频道。

在oracle.linux.com访问商业的MySQL ULN包需要你提供一个具有MySQL（企业或标准）有效商业许可证的CSI。截至本文撰写时，有效的购买是60944、60945、64911和64912。适当的CSI会在你的ULN GUI界面中提供商业MySQL订阅渠道。

一旦使用ULN安装了MySQL，你可以在第2.5.7节 "从本地软件库安装Linux上的MySQL"，特别是在第2.5.4节 "使用来自Oracle的RPM包在Linux上安装MySQL "中找到关于启动和停止服务器等的信息。

如果你正在改变你的包的来源以使用ULN，而不改变你正在使用的MySQL的构建，那么备份你的数据，删除你现有的二进制文件，并用来自ULN的二进制文件替换它们。如果涉及到构建的改变，我们建议备份是一个转储（mysqldump或mysqlpump或来自MySQL Shell的备份工具），以备你在新的二进制文件到位后需要重建你的数据。如果向ULN的转变跨越了版本边界，请在继续进行之前查阅本节。第2.11节，"升级MySQL"。

注意

Oracle Linux 8从MySQL 8.0.17开始被支持，社区工具和连接器通道在MySQL 8.0.24发布时被添加。

## 2.7 在Solaris上安装MySQL

2.7.1 使用Solaris PKG在Solaris上安装MySQL

注意

MySQL 8.0支持Solaris 11.4及以上版本

Solaris上的MySQL有许多不同的格式。

关于使用本地Solaris PKG格式进行安装的信息，请参阅第2.7.1节 "使用Solaris PKG在Solaris上安装MySQL"。

要使用标准的tar二进制安装，请使用第2.2节 "使用通用二进制文件在Unix/Linux上安装MySQL "中提供的说明。检查本节末尾的说明和提示，了解安装前后可能需要的Solaris特定说明。

重要事项

安装包对Oracle Developer Studio 12.6 Runtime Libraries有依赖性，在运行MySQL安装包之前必须安装该程序。请参阅这里的Oracle Developer Studio的下载选项。该安装包使你能够只安装运行时库而不是完整的Oracle Developer Studio；请参阅在Oracle Solaris 11上只安装运行时库中的说明。

要获得Solaris的tarball或PKG格式的二进制MySQL分布，https://dev.mysql.com/downloads/mysql/8.0.html。

在Solaris上安装和使用MySQL时应注意的其他注意事项。

如果你想用mysql用户和组来使用MySQL，请使用groupadd和useradd命令。

groupadd mysql

useradd -g mysql -s /bin/false mysql

如果你在Solaris上使用二进制tar球发行版安装MySQL，因为Solaris tar不能处理长文件名，使用GNU tar（gtar）来解压发行版。如果你的系统上没有GNU tar，请用以下命令安装它。

pkg install archiver/gnu-tar

你应该用forcedirectio选项来挂载你打算存储InnoDB文件的任何文件系统。在这个平台上使用InnoDB存储引擎时，如果不这样做会导致性能的显著下降。

如果你想让MySQL自动启动，你可以把support-files/mysql.server复制到/etc/init.d，并为它创建一个符号链接，命名为/etc/rc3.d/S99mysql.server。

如果太多的进程试图非常迅速地连接到mysqld，你应该在MySQL日志中看到这个错误。

Error in accept: Protocol error

你可以尝试用-back\_log=50选项启动服务器，作为解决这个问题的方法。

要在Solaris上配置核心文件的生成，你应该使用coreadm命令。由于在setuid()程序上生成核心的安全问题，默认情况下，Solaris不支持在setuid()程序上生成核心文件。然而，你可以使用coreadm修改这一行为。如果你为当前用户启用了setuid()核心文件，它们将使用模式600生成，并由超级用户拥有。

### 2.7.1 使用Solaris PKG在Solaris上安装MySQL

你可以使用原生Solaris PKG格式的二进制包在Solaris上安装MySQL，而不是使用二进制tarball分发。

重要的是安装包对Oracle Developer Studio 12.6 Runtime Libraries有依赖性，在运行MySQL安装包之前必须安装该程序。请参阅这里的Oracle Developer Studio的下载选项。该安装包使你能够只安装运行时库，而不是完整的Oracle Developer Studio；请参阅在Oracle Solaris 11上只安装运行时库的说明。

要使用这个包，请下载相应的mysql-VERSION-solaris11-PLATFORM.pkg.gz文件，然后将其解压。例如

shell> **gunzip mysql-*8.0.26*-solaris11-x86\_64.pkg.gz**

要安装一个新的软件包，使用 pkgadd 并按照屏幕上的提示操作。你必须有root权限来执行这个操作。

shell> **pkgadd -d mysql-*8.0.26*-solaris11-x86\_64.pkg**

The following packages are available:

1 mysql MySQL Community Server (GPL)

(i86pc) 8.0.26

Select package(s) you wish to process (or 'all' to process

all packages). (default: all) [?,??,q]:

PKG安装程序会安装所有需要的文件和工具，如果你的数据库不存在，则初始化数据库。为了完成安装，你应该按照安装结束时的说明，设置MySQL的根密码。或者，你可以运行安装时附带的mysql\_secure\_installation脚本。

默认情况下，PKG包将MySQL安装在根路径/opt/mysql下。当使用pkgadd时，你可以只改变安装根路径，它可以用来在不同的Solaris区域安装MySQL。如果你需要安装在一个特定的目录下，请使用二进制tar文件分发。

pkg安装程序为MySQL复制一个合适的启动脚本到/etc/init.d/mysql。为了使MySQL能够自动启动和关闭，你应该在这个文件和init脚本目录之间建立一个链接。例如，为了确保MySQL的安全启动和关闭，你可以使用以下命令来添加正确的链接。

shell> **ln /etc/init.d/mysql /etc/rc3.d/S91mysql**

shell> **ln /etc/init.d/mysql /etc/rc0.d/K02mysql**

要删除MySQL，安装的软件包名称是mysql。你可以将其与pkgrm命令结合起来使用来删除安装。

在使用Solaris软件包文件格式时要进行升级，在安装更新的软件包之前必须删除现有的安装。移除软件包并不删除现有的数据库信息，只是删除服务器、二进制文件和支持文件。因此，典型的升级顺序是

shell> **mysqladmin shutdown**

shell> **pkgrm mysql**

shell> **pkgadd -d mysql-*8.0.26*-solaris11-x86\_64.pkg**

shell> **mysqld\_safe &**

shell> **mysql\_upgrade** # prior to MySQL 8.0.16 only

在执行任何升级之前，你应该检查第2.11节 "升级MySQL "中的说明。

## 2.8 在FreeBSD上安装MySQL

本节提供关于在FreeBSD Unix的变体上安装MySQL的信息。

你可以通过使用Oracle提供的二进制分布在FreeBSD上安装MySQL。欲了解更多信息，请参见第2.2节，"在Unix/Linux上使用通用二进制文件安装MySQL"。

安装MySQL的最简单（也是首选）方式是使用http://www.freebsd.org/ 上的mysql-server和mysql-client端口。使用这些端口会给你带来以下好处。

* 一个可以工作的MySQL，并启用所有已知可以在你的FreeBSD版本上工作的优化功能。
* 自动配置和构建。
* 安装在 /usr/local/etc/rc.d 的启动脚本。
* 能够使用 pkg\_info -L 来查看哪些文件被安装。
* 如果你不再需要MySQL在你的机器上，可以使用pkg\_delete来删除它。

MySQL的构建过程需要GNU make（gmake）来工作。如果GNU make不可用，你必须在编译MySQL之前先安装它。

注意

根据ldd mysqld的前提库：libthr, libcrypt, libkrb5, libm, librt, libexecinfo, libunwind, and libssl。

要使用 ports 系统进行安装。

# cd /usr/ports/databases/mysql80-server

# make

...

# cd /usr/ports/databases/mysql80-client

# make

标准端口安装将服务器放在/usr/local/libexec/mysqld中，MySQL服务器的启动脚本放在/usr/local/etc/rc.d/mysql-server中。

关于BSD实现的一些附加说明。

在使用 ports 系统安装后要删除 MySQL。

# cd /usr/ports/databases/mysql80-server

# make deinstall

...

# cd /usr/ports/databases/mysql80-client

# make deinstall

...

...

如果你在MySQL中遇到当前日期的问题，设置TZ变量应该有帮助。见第4.9节，"环境变量"。

## 2.9 从源代码安装MySQL

从源代码构建MySQL使你可以定制构建参数、编译器优化和安装位置。关于已知可以运行MySQL的系统的列表，见https://www.mysql.com/support/supportedplatforms/database.html。

在你从源码进行安装之前，请检查Oracle是否为你的平台生产了预编译的二进制发行版，以及它是否对你有效。我们花了很大的力气来确保我们的二进制文件是用最好的选项构建的，以获得最佳性能。在第2.2节 "使用通用二进制文件在Unix/Linux上安装MySQL "中提供了安装二进制发行版的说明。

如果你对使用与Oracle在你的平台上生产二进制发行版相同或类似的构建选项从源发行版构建MySQL感兴趣，可以获得一个二进制发行版，将其解压，并查看docs/INFO\_BIN文件，其中包含有关该MySQL发行版如何配置和编译的信息。

警告

用非标准选项构建MySQL可能导致功能、性能或安全性降低。

MySQL源代码包含使用Doxygen编写的内部文档。生成的Doxygen内容可在https://dev.mysql.com/doc/index-other.html。也可以使用第2.9.10节 "生成MySQL Doxygen文档内容 "中的说明，从MySQL源分发中本地生成这些内容。

### 2.9.1 源码安装方法

有两种方法可以从源码安装MySQL。

* 使用标准的MySQL源码发行版。要获得一个标准的发行版，见第2.1.3节 "如何获得MySQL"。关于从标准发行版构建的说明，见第2.9.4节，"使用标准源发行版安装MySQL"。

标准发行版可作为压缩的tar文件、Zip档案或RPM包提供。发行文件的名称是mysql-VERSION.tar.gz、mysql-VERSION.zip或mysql-VERSION.rpm，其中VERSION是一个数字，如8.0.26。源码发行版的文件名可以与预编译的二进制发行版的文件名区分开来，因为源码发行版的文件名是通用的，不包括平台名称，而二进制发行版的文件名包括一个平台名称，表明该发行版要用于的系统类型（例如，pc-linux-i686或winx64）。

* 使用MySQL开发树。有关从某个开发树中构建的信息，请参见第2.9.5节 "使用开发源码树安装MySQL"。

### 2.9.2 源码安装的先决条件

从源码安装MySQL需要几个开发工具。无论你使用的是标准源码分发还是开发源码树，其中一些工具都是需要的。其他工具的要求则取决于你使用的安装方法。

要从源码安装MySQL，无论采用何种安装方法，都必须满足以下系统要求。

* CMake，它被用作所有平台上的构建框架。CMake可以从http://www.cmake.org 下载。
* 一个好的make程序。尽管有些平台有自己的make实现，但强烈建议你使用GNU make 3.75或更高版本。它可能已经在你的系统上以gmake的形式存在。GNU make可从<http://www.gnu.org/software/make/>。
* MySQL 8.0源代码允许使用C++14特性。为了在所有支持的平台上实现良好的C++14支持水平，适用以下最小编译器版本。
  + Linux。GCC 5.3.1（Ubuntu 16.04），GCC 10（EL7，EL8
  + FreeBSD。Clang 4.0.0
  + macOS：XCode 9；但只有Oracle用于构建二进制包的版本才被保证支持。
  + Solaris。Developer Studio 12.6
  + Windows。Visual Studio 2019
* MySQL C API需要一个C++或C99编译器来编译。
* 需要一个SSL库来支持加密连接、用于随机数生成的熵以及其他与加密相关的操作。默认情况下，编译时使用安装在主机系统上的OpenSSL库。要明确指定该库，请在调用CMake时使用WITH\_SSL选项。有关其他信息，请参见第2.9.6节 "配置SSL库支持"。
* 构建MySQL需要Boost C++库（但不是为了使用它）。MySQL的编译需要一个特定的Boost版本。通常情况下，那是当前的Boost版本，但如果一个特定的MySQL源码发行版需要不同的版本，配置过程就会停止，并显示出它所需要的Boost版本。要获得Boost及其安装说明，请访问官方网站。Boost安装完毕后，通过在调用CMake时定义WITH\_BOOST选项，告诉构建系统Boost文件的位置。比如说

cmake . -DWITH\_BOOST=/usr/local/boost\_***version\_number***

根据需要调整路径以配合你的安装。

* ncurses库。
* 足够的自由内存。如果你在编译大的源文件时遇到 "内部编译器错误 "等问题，可能是你的内存太少。如果在虚拟机上进行编译，可以尝试增加内存分配。
* 如果你打算运行测试脚本，就需要Perl。大多数类似Unix的系统包括Perl。在Windows上，你可以使用ActiveState Perl这样的版本。

要从一个标准的源码分布中安装MySQL，需要使用以下工具之一来解压分布文件。

* 对于一个.tar.gz压缩的tar文件。GNU gunzip来解压发行文件，并使用合理的tar来解压。如果你的 tar 程序支持 z 选项，它可以同时解压缩和解压文件。

GNU tar 是已知的工作。一些操作系统提供的标准tar不能解压MySQL发行版中的长文件名。你应该下载并安装GNU tar，或者如果有的话，使用预装的GNU tar版本。通常这可以作为gnutar、gtar，或作为GNU或自由软件目录下的tar，如/usr/sfw/bin或/usr/local/bin。GNU tar可以从http://www.gnu.org/software/tar/。

* 对于一个.zip的压缩文件。WinZip或其他可以读取.zip文件的工具。
* 对于一个.rpm的RPM包。用来构建发行版的rpmbuild程序将其解压。

要从开发源码树安装MySQL，需要以下额外的工具。

* 需要Git修订控制系统来获取开发源代码。GitHub帮助提供了在不同平台上下载和安装Git的说明。MySQL于2014年9月正式加入GitHub。关于MySQL迁移到GitHub的更多信息，请参考MySQL发布工程博客上的公告。MySQL在GitHub上
* bison 2.1或更高版本，可从http://www.gnu.org/software/bison/。(第1版已不再支持。)尽可能使用最新版本的bison；如果你遇到问题，请升级到较高版本，而不是恢复到较早版本。

bison可以从http://www.gnu.org/software/bison/。bison for Windows可以从http://gnuwin32.sourceforge.net/packages/bison.htm。下载标有 "完整包，不包括源代码 "的软件包。在Windows上，bison的默认位置是C:\Program Files\GnuWin32目录。由于目录名称中有空格，有些工具可能找不到bison。另外，如果路径中存在空格，Visual Studio可能会直接挂起。你可以通过安装到一个不包含空格的目录（例如C:\GnuWin32）来解决这些问题。

* 在Solaris Express上，除了bison，还必须安装m4。m4可以从http://www.gnu.org/software/m4/。

注意

如果你必须安装任何程序，请修改你的PATH环境变量，以包括这些程序所在的任何目录。见第4.2.9节，"设置环境变量"。

如果你遇到问题并需要提交错误报告，请使用1.6节 "如何报告错误或问题 "中的说明。

### 2.9.3 源码安装的MySQL布局

默认情况下，当你从源码编译后安装MySQL时，安装步骤会在/usr/local/mysql下安装文件。安装目录下的组件位置与二进制发行版相同。参见表2.3，"通用Unix/Linux二进制包的MySQL安装布局"，以及第2.3.1节，"Microsoft Windows上的MySQL安装布局"。要配置不同于默认值的安装位置，请使用第2.9.7节 "MySQL源配置选项 "中描述的选项。

### 2.9.4 使用标准源发布版安装MySQL

要从一个标准的源码发行版安装MySQL。

1、确认你的系统满足第2.9.2节 "源码安装先决条件 "中列出的工具要求。

2、使用第2.1.3节 "如何获得MySQL "中的说明获得一个分发文件。

3、使用本节中的说明配置、构建和安装分发文件。

4、使用第2.10节 "安装后设置和测试 "中的说明执行安装后程序。

MySQL在所有平台上使用CMake作为构建框架。这里给出的说明应能使你产生一个工作的安装。关于使用CMake构建MySQL的其他信息，请参阅如何用CMake构建MySQL服务器。

如果你从一个源码RPM开始，使用下面的命令制作一个你可以安装的二进制RPM。如果你没有rpmbuild，用rpm代替。

shell> **rpmbuild --rebuild --clean MySQL-*VERSION*.src.rpm**

结果是一个或多个二进制RPM包，你可以按照第2.5.4节 "使用来自Oracle的RPM包在Linux上安装MySQL "中的指示进行安装。

从压缩的tar文件或Zip归档的源发布文件进行安装的顺序类似于从通用二进制发布文件进行安装的过程（见第2.2节，"使用通用二进制文件在Unix/Linux上安装MySQL"），只是它用于所有平台，包括配置和编译发布文件的步骤。例如，在Unix上使用压缩的tar文件源发布，基本的安装命令序列看起来像这样。

# Preconfiguration setup

shell> **groupadd mysql**

shell> **useradd -r -g mysql -s /bin/false mysql**

# Beginning of source-build specific instructions

shell> **tar zxvf mysql-*VERSION*.tar.gz**

shell> **cd mysql-*VERSION***

shell> **mkdir bld**

shell> **cd bld**

shell> **cmake ..**

shell> **make**

shell> **make install**

# End of source-build specific instructions

# Postinstallation setup

shell> **cd /usr/local/mysql**

shell> **mkdir mysql-files**

shell> **chown mysql:mysql mysql-files**

shell> **chmod 750 mysql-files**

shell> **bin/mysqld --initialize --user=mysql**

shell> **bin/mysql\_ssl\_rsa\_setup**

shell> **bin/mysqld\_safe --user=mysql &**

# Next command is optional

shell> **cp support-files/mysql.server /etc/init.d/mysql.server**

下面是一个更详细的源码构建具体说明的版本。

注意

这里显示的程序并没有为MySQL账户设置任何密码。按照该程序操作后，进入第2.10节 "安装后设置和测试"，进行安装后设置和测试。

* 执行预配置设置
* 获取并解压发行版
* 配置该发行版
* 构建发行版
* 安装发行版
* 执行安装后设置

**执行预配置设置**

在Unix上，设置mysql用户和组，用于运行和执行MySQL服务器，并拥有数据库目录。详情请参见创建mysql用户和组。然后以mysql用户的身份执行下列步骤，除非有说明。

获取并解压发行版

选择你想解压发行版的目录，并将位置改到该目录下。

使用第2.1.3节 "如何获得MySQL "中的说明获得一个分发文件。

将分发文件解压到当前目录中。

* 要解压一个压缩的tar文件，如果tar有z选项支持，它可以解压并解压发行版。

shell> **tar zxvf mysql-*VERSION*.tar.gz**

如果你的 tar 没有 z 选项支持，使用 gunzip 来解压发行版，然后用 tar 来解压。

shell> **gunzip < mysql-*VERSION*.tar.gz | tar xvf -**

或者，CMake可以解压和解压发行版。

shell> **cmake -E tar zxvf mysql-*VERSION*.tar.gz**

* 要解压Zip档案，请使用WinZip或其他可以读取.zip文件的工具。

解压后的发行文件会创建一个名为mysql-VERSION的目录。

**配置分发文件**

改变位置进入解压后的发行文件的顶层目录。

shell> **cd mysql-*VERSION***

在源码树之外构建，以保持源码树的清洁。如果顶级源码目录在你当前工作目录下被命名为 mysql-src，你可以在同一级别的名为 bld 的目录中构建。创建该目录并进入那里。

shell> **mkdir bld**

shell> **cd bld**

配置构建目录。最小配置命令不包括覆盖配置默认值的选项。

shell> **cmake ../mysql-src**

构建目录不需要在源树之外。例如，你可以在顶层源代码树下名为bld的目录中进行构建。要做到这一点，从mysql-src作为你的当前工作目录开始，创建目录bld，然后去那里。

shell> **mkdir bld**

shell> **cd bld**

配置构建目录。最小配置命令不包括覆盖配置默认值的选项。

shell> **cmake ..**

如果你在同一级别有多个源树（例如，要构建多个版本的MySQL），第二种策略可能是有利的。第一种策略将所有构建目录放在同一级别，这要求你为每个目录选择一个独特的名字。使用第二种策略，你可以在每个源树中使用相同的名字作为构建目录。下面的说明是假设采用第二种策略。

在Windows上，指定开发环境。例如，下面的命令分别为32位或64位构建配置MySQL。

shell> **cmake .. -G "Visual Studio 12 2013"**

shell> **cmake .. -G "Visual Studio 12 2013 Win64"**

在macOS上，要使用Xcode IDE。

shell> **cmake .. -G Xcode**

当你运行cmake时，你可能想在命令行中添加选项。这里有一些例子。

* -DBUILD\_CONFIG=mysql\_release: 用Oracle用于生产官方MySQL版本的二进制发行版的相同构建选项来配置源。
* -DCMAKE\_INSTALL\_PREFIX=dir\_name：配置发行版以安装在一个特定的位置。
* -DCPACK\_MONOLITHIC\_INSTALL=1：使make包生成一个安装文件而不是多个文件。
* -DWITH\_DEBUG=1: 用调试支持来构建发行版。

关于更多的选项列表，见第2.9.7节 "MySQL源配置选项"。

要列出配置选项，请使用下列命令之一。

shell> **cmake .. -L** # overview

shell> **cmake .. -LH** # overview with help text

shell> **cmake .. -LAH** # all params with help text

shell> **ccmake ..** # interactive display

如果CMake失败了，你可能需要用不同的选项再次运行它来重新配置。如果你重新配置，请注意以下几点。

* 如果CMake在之前被运行之后被运行，它可能会使用在之前的调用中收集的信息。这些信息被保存在CMakeCache.txt中。当CMake启动时，它会寻找该文件并读取其内容（如果它存在），前提是这些信息仍然正确。这个假设在你重新配置时是无效的。
* 每次运行CMake时，你必须再次运行make来重新编译。然而，你可能想先删除以前构建的旧对象文件，因为它们是用不同的配置选项编译的。

为了防止旧的对象文件或配置信息被使用，在重新运行CMake之前，在Unix的构建方向上运行这些命令。

shell> **make clean**

shell> **rm CMakeCache.txt**

或者，在Windows上。

shell> **devenv MySQL.sln /clean**

shell> **del CMakeCache.txt**

在MySQL社区Slack上询问之前，请检查CMakeFiles目录中的文件，以获取关于失败的有用信息。要提交错误报告，请使用1.6节 "如何报告错误或问题 "中的说明。

**构建发行版**

在 Unix 上。

shell> **make**

shell> **make VERBOSE=1**

第二条命令设置VERBOSE来显示每个编译后的源代码的命令。

在使用GNU make并且已经安装了gmake的系统上，使用gmake代替。

在Windows上。

shell> **devenv MySQL.sln /build RelWithDebInfo**

如果你已经到了编译阶段，但发行版没有编译，请参阅第2.9.8节 "处理编译MySQL的问题"，以获得帮助。如果这还不能解决问题，请使用第1.6节 "如何报告错误或问题 "中的说明将其输入我们的错误数据库。如果你已经安装了所需工具的最新版本，而它们在试图处理我们的配置文件时崩溃，也请报告。然而，如果你得到一个未找到命令的错误或所需工具的类似问题，请不要报告它。相反，请确保所有需要的工具都已安装，并且你的PATH变量设置正确，以便你的shell可以找到它们。

**安装发行版**

在Unix上。

shell> **make install**

这将在配置的安装目录下安装文件（默认为/usr/local/mysql）。你可能需要以root身份运行该命令。

要在一个特定的目录下安装，可以在命令行中添加一个DESTDIR参数。

shell> **make install DESTDIR="/opt/mysql"**

或者，生成安装包文件，你可以安装在你喜欢的地方。

shell> **make package**

这个操作会产生一个或多个.tar.gz文件，可以像一般的二进制发布包一样安装。见第2.2节，"使用通用二进制文件在Unix/Linux上安装MySQL"。如果你用-DCPACK\_MONOLITHIC\_INSTALL=1运行CMake，该操作产生一个文件。否则，它将产生多个文件。

在Windows上，生成数据目录，然后创建一个.zip档案安装包。

shell> **devenv MySQL.sln /build RelWithDebInfo /project initial\_database**

shell> **devenv MySQL.sln /build RelWithDebInfo /project package**

你可以把生成的.zip压缩包安装到你喜欢的地方。参见第2.3.4节，"使用noinstall ZIP档案在Microsoft Windows上安装MySQL"。

**执行安装后的设置**

安装过程的剩余部分包括设置配置文件、创建核心数据库和启动MySQL服务器。有关说明，见第2.10节 "安装后设置和测试"。

注意

列在MySQL授予表中的账户最初没有密码。在启动服务器后，你应该使用第2.10节 "安装后设置和测试 "中的说明为它们设置密码。

### 2.9.5 使用开发源码树安装MySQL

本节描述了如何从GitHub上托管的最新开发源代码安装MySQL。为了从这个源码托管服务中获得MySQL服务器源码，你可以设置一个本地MySQL Git库。

在GitHub上，MySQL Server和其他MySQL项目可以在MySQL页面找到。MySQL Server项目是一个单一的仓库，包含了几个MySQL系列的分支。

MySQL于2014年9月正式加入GitHub。关于MySQL迁移到GitHub的更多信息，请参考MySQL发布工程博客上的公告。MySQL在GitHub上

* 从开发源码安装的先决条件
* 设置MySQL的Git存储库

**从开发源安装的先决条件**

要从开发源码树安装MySQL，你的系统必须满足第2.9.2节 "源码安装先决条件 "中列出的工具要求。

**设置一个MySQL的Git存储库**

要在你的机器上建立一个MySQL Git仓库。

1、克隆MySQL Git 仓库到你的机器上。下面的命令将MySQL Git仓库克隆到名为mysql-server的目录中。最初的下载可能需要一些时间来完成，这取决于你的连接速度。

$ git clone https://github.com/mysql/mysql-server.git

Cloning into 'mysql-server'...

remote: Counting objects: 1198513, done.

remote: Total 1198513 (delta 0), reused 0 (delta 0), pack-reused 1198513

Receiving objects: 100% (1198513/1198513), 1.01 GiB | 7.44 MiB/s, done.

Resolving deltas: 100% (993200/993200), done.

Checking connectivity... done.

Checking out files: 100% (25510/25510), done.

2、当克隆操作完成后，你的本地 MySQL Git 仓库的内容会出现类似于下面的情况。

~$ cd mysql-server

~/mysql-server$ ls

client extra mysys storage

cmake include packaging strings

CMakeLists.txt INSTALL plugin support-files

components libbinlogevents README testclients

config.h.cmake libbinlogstandalone router unittest

configure.cmake libmysql run\_doxygen.cmake utilities

Docs libservices scripts VERSION

Doxyfile-ignored LICENSE share vio

Doxyfile.in man sql win

doxygen\_resources mysql-test sql-common

3、使用git branch -r命令查看MySQL存储库的远程跟踪分支。

~/mysql-server$ git branch -r

origin/5.5

origin/5.6

origin/5.7

origin/8.0

origin/HEAD -> origin/8.0

origin/cluster-7.2

origin/cluster-7.3

origin/cluster-7.4

origin/cluster-7.5

origin/cluster-7.6

4、要查看你本地仓库中签出的分支，请发出git branch命令。当你克隆MySQL Git仓库时，最新的MySQL GA分支会被自动签出。星号标识了活动分支。

~/mysql-server$ git branch

\* 8.0

5、要签出一个较早的MySQL分支，运行git checkout命令，指定分支名称。例如，签出MySQL 5.7分支。

~/mysql-server$ git checkout 5.7

Checking out files: 100% (9600/9600), done.

Branch 5.7 set up to track remote branch 5.7 from origin.

Switched to a new branch '5.7'

6、要获得在你最初设置MySQL Git仓库后所做的修改，请切换到你想更新的分支，并发出git pull命令。

~/mysql-server$ git checkout 8.0

~/mysql-server$ git pull

要检查提交历史，请使用git log选项。

~/mysql-server$ git log

你也可以在GitHub MySQL网站上浏览提交历史和源代码。

如果你看到有问题的变化或代码，请在MySQL社区Slack上提问。关于贡献补丁的信息，请参见贡献给MySQL服务器。

7、在你克隆了MySQL的Git仓库并检查出你想要构建的分支后，你可以从源代码构建MySQL服务器。第2.9.4节 "使用标准源码发行版安装MySQL "中提供了说明，只是你跳过了关于获取和解压发行版的部分。

在生产机器上安装来自发行版源码树的构建时要小心。安装命令可能会覆盖你的实时发布安装。如果你已经安装了MySQL并且不想覆盖它，运行CMake时，CMAKE\_INSTALL\_PREFIX、MYSQL\_TCP\_PORT和MYSQL\_UNIX\_ADDR选项的值与你的生产服务器使用的不同。关于防止多个服务器相互干扰的其他信息，请参见第5.8节，"在一台机器上运行多个MySQL实例"。

在你的新安装中努力发挥。例如，尝试使新功能崩溃。从运行make test开始。见《MySQL测试套件》。

### 2.9.6 配置SSL库支持

需要一个SSL库来支持加密连接、用于随机数生成的熵以及其他与加密有关的操作。

如果你从一个源码发行版编译MySQL，CMake将发行版配置为默认使用已安装的OpenSSL库。

要使用OpenSSL进行编译，请使用这个程序。

1、确保OpenSSL 1.0.1或更高版本已安装在你的系统上。如果安装的OpenSSL版本低于1.0.1，CMake会在MySQL配置时产生一个错误。如果有必要获得OpenSSL，请访问http://www.openssl.org。

2、WITH\_SSL CMake选项决定在编译MySQL时使用哪个SSL库（见第2.9.7节，"MySQL源配置选项"）。默认是-DWITH\_SSL=system，它使用OpenSSL。要明确这一点，请在CMake命令行中指定该选项。例如

cmake . -DWITH\_SSL=system

该命令将发行版配置为使用已安装的OpenSSL库。另外，要明确指定OpenSSL安装的路径名称，请使用以下语法。如果你安装了多个版本的OpenSSL，这可能很有用，可以防止CMake选择错误的版本。

cmake . -DWITH\_SSL=***path\_name***

3、编译并安装该发行版。

要检查一个mysqld服务器是否支持加密连接，检查have\_ssl系统变量的值。

mysql> **SHOW VARIABLES LIKE 'have\_ssl';**

+---------------+-------+

| Variable\_name | Value |

+---------------+-------+

| have\_ssl | YES |

+---------------+-------+

如果值为YES，服务器支持加密连接。如果值为DISABLED，则服务器能够支持加密连接，但在启动时没有使用适当的--ssl-xxx选项来启用加密连接；见第6.3.1节，"配置MySQL使用加密连接"。

### 2.9.7 MySQL源配置选项

CMake程序对你如何配置MySQL源码分发提供了大量的控制。通常，你使用CMake命令行上的选项来做这件事。关于CMake支持的选项的信息，请在顶级源码目录下运行这些命令中的任何一个。

cmake . -LH

ccmake .

你也可以使用某些环境变量来影响CMake。参见第4.9节 "环境变量"。

对于布尔选项，值可以指定为1或ON以启用该选项，或者指定为0或OFF以禁用该选项。

许多选项配置了编译时的默认值，可以在服务器启动时被覆盖。例如，CMAKE\_INSTALL\_PREFIX、MYSQL\_TCP\_PORT和MYSQL\_UNIX\_ADDR选项配置了默认的安装基础目录位置、TCP/IP端口号和Unix套接字文件，可以在服务器启动时通过mysqld的-basedir、-port和-socket选项进行更改。在适用的地方，配置选项描述指出了相应的mysqld启动选项。

下面的章节提供了关于CMake选项的更多信息。

CMake选项参考

* 一般选项
* 安装布局选项
* 存储引擎选项
* 功能选项
* 编译器标志
* 用于编译NDB Cluster的CMake选项

**CMake选项参考**

下表显示了可用的CMake选项。在默认列中，PREFIX代表CMAKE\_INSTALL\_PREFIX选项的值，它指定了安装基础目录。这个值被用作若干安装子目录的父目录。

**Table 2.14 MySQL Source-Configuration Option Reference (CMake)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Formats** | **Description** | **Default** | **Introduced** | **Removed** |
| [**ADD\_GDB\_INDEX**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_add_gdb_index) | Whether to enable generation of .gdb\_index section in binaries |  | 8.0.18 |  |
| [**BUILD\_CONFIG**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_build_config) | Use same build options as official releases |  |  |  |
| [**BUNDLE\_RUNTIME\_LIBRARIES**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_bundle_runtime_libraries) | Bundle runtime libraries with server MSI and Zip packages for Windows | **OFF** |  |  |
| [**CMAKE\_BUILD\_TYPE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_cmake_build_type) | Type of build to produce | **RelWithDebInfo** |  |  |
| [**CMAKE\_CXX\_FLAGS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_cmake_cxx_flags) | Flags for C++ Compiler |  |  |  |
| [**CMAKE\_C\_FLAGS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_cmake_c_flags) | Flags for C Compiler |  |  |  |
| [**CMAKE\_INSTALL\_PREFIX**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_cmake_install_prefix) | Installation base directory | **/usr/local/mysql** |  |  |
| **CMAKE\_INSTALL\_PRIV\_LIBDIR** | Installation private library directory |  | 8.0.18 |  |
| [**COMPILATION\_COMMENT**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_compilation_comment) | Comment about compilation environment |  |  |  |
| [**COMPILATION\_COMMENT\_SERVER**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_compilation_comment_server) | Comment about compilation environment for use by mysqld |  | 8.0.14 |  |
| [**COMPRESS\_DEBUG\_SECTIONS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_compress_debug_sections) | Compress debug sections of binary executables | **OFF** | 8.0.22 |  |
| [**CPACK\_MONOLITHIC\_INSTALL**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_cpack_monolithic_install) | Whether package build produces single file | **OFF** |  |  |
| [**DEFAULT\_CHARSET**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_default_charset) | The default server character set | **utf8mb4** |  |  |
| [**DEFAULT\_COLLATION**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_default_collation) | The default server collation | **utf8mb4\_0900\_ai\_ci** |  |  |
| **DISABLE\_DATA\_LOCK** | Exclude the performance schema data lock instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_COND**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_cond) | Exclude Performance Schema condition instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_ERROR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_error) | Exclude the performance schema server error instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_FILE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_file) | Exclude Performance Schema file instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_IDLE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_idle) | Exclude Performance Schema idle instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_MEMORY**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_memory) | Exclude Performance Schema memory instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_METADATA**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_metadata) | Exclude Performance Schema metadata instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_MUTEX**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_mutex) | Exclude Performance Schema mutex instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_PS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_ps) | Exclude the performance schema prepared statements | **OFF** |  |  |
| [**DISABLE\_PSI\_RWLOCK**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_rwlock) | Exclude Performance Schema rwlock instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_SOCKET**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_socket) | Exclude Performance Schema socket instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_SP**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_sp) | Exclude Performance Schema stored program instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_STAGE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_stage) | Exclude Performance Schema stage instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_STATEMENT**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_statement) | Exclude Performance Schema statement instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_STATEMENT\_DIGEST**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_statement_digest) | Exclude Performance Schema statements\_digest instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_TABLE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_table) | Exclude Performance Schema table instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_THREAD**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_thread) | Exclude the performance schema thread instrumentation | **OFF** |  |  |
| [**DISABLE\_PSI\_TRANSACTION**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_psi_transaction) | Exclude the performance schema transaction instrumentation | **OFF** |  |  |
| [**DISABLE\_SHARED**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_disable_shared) | Do not build shared libraries, compile position-dependent code | **OFF** |  | 8.0.18 |
| [**DOWNLOAD\_BOOST**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_download_boost) | Whether to download the Boost library | **OFF** |  |  |
| [**DOWNLOAD\_BOOST\_TIMEOUT**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_download_boost_timeout) | Timeout in seconds for downloading the Boost library | **600** |  |  |
| [**ENABLED\_LOCAL\_INFILE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_enabled_local_infile) | Whether to enable LOCAL for LOAD DATA | **OFF** |  |  |
| [**ENABLED\_PROFILING**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_enabled_profiling) | Whether to enable query profiling code | **ON** |  |  |
| [**ENABLE\_DOWNLOADS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_enable_downloads) | Whether to download optional files | **OFF** |  | 8.0.25 |
| [**ENABLE\_EXPERIMENTAL\_SYSVARS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_enable_experimental_sysvars) | Whether to enabled experimental InnoDB system variables | **OFF** |  |  |
| [**ENABLE\_GCOV**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_enable_gcov) | Whether to include gcov support |  |  |  |
| [**ENABLE\_GPROF**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_enable_gprof) | Enable gprof (optimized Linux builds only) | **OFF** |  |  |
| [**FORCE\_INSOURCE\_BUILD**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_force_insource_build) | Whether to force an in-source build | **OFF** | 8.0.14 |  |
| [**FORCE\_UNSUPPORTED\_COMPILER**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_force_unsupported_compiler) | Whether to permit unsupported compiler | **OFF** |  |  |
| [**FPROFILE\_GENERATE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_fprofile_generate) | Whether to generate profile guided optimization data | **OFF** | 8.0.19 |  |
| [**FPROFILE\_USE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_fprofile_use) | Whether to use profile guided optimization data | **OFF** | 8.0.19 |  |
| [**HAVE\_PSI\_MEMORY\_INTERFACE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_have_psi_memory_interface) | Enable performance schema memory tracing module for memory allocation functions used in dynamic storage of over-aligned types | **OFF** | 8.0.25 |  |
| [**IGNORE\_AIO\_CHECK**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_ignore_aio_check) | With -DBUILD\_CONFIG=mysql\_release, ignore libaio check | **OFF** |  |  |
| [**INSTALL\_BINDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_bindir) | User executables directory | **PREFIX/bin** |  |  |
| [**INSTALL\_DOCDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_docdir) | Documentation directory | **PREFIX/docs** |  |  |
| [**INSTALL\_DOCREADMEDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_docreadmedir) | README file directory | **PREFIX** |  |  |
| [**INSTALL\_INCLUDEDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_includedir) | Header file directory | **PREFIX/include** |  |  |
| [**INSTALL\_INFODIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_infodir) | Info file directory | **PREFIX/docs** |  |  |
| [**INSTALL\_LAYOUT**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_layout) | Select predefined installation layout | **STANDALONE** |  |  |
| [**INSTALL\_LIBDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_libdir) | Library file directory | **PREFIX/lib** |  |  |
| [**INSTALL\_MANDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_mandir) | Manual page directory | **PREFIX/man** |  |  |
| [**INSTALL\_MYSQLKEYRINGDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_mysqlkeyringdir) | Directory for keyring\_file plugin data file | **platform specific** |  |  |
| [**INSTALL\_MYSQLSHAREDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_mysqlsharedir) | Shared data directory | **PREFIX/share** |  |  |
| [**INSTALL\_MYSQLTESTDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_mysqltestdir) | mysql-test directory | **PREFIX/mysql-test** |  |  |
| [**INSTALL\_PKGCONFIGDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_pkgconfigdir) | Directory for mysqlclient.pc pkg-config file | **INSTALL\_LIBDIR/pkgconfig** |  |  |
| [**INSTALL\_PLUGINDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_plugindir) | Plugin directory | **PREFIX/lib/plugin** |  |  |
| [**INSTALL\_SBINDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_sbindir) | Server executable directory | **PREFIX/bin** |  |  |
| [**INSTALL\_SECURE\_FILE\_PRIVDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_secure_file_privdir) | secure\_file\_priv default value | **platform specific** |  |  |
| [**INSTALL\_SHAREDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_sharedir) | aclocal/mysql.m4 installation directory | **PREFIX/share** |  |  |
| [**INSTALL\_STATIC\_LIBRARIES**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_static_libraries) | Whether to install static libraries | **ON** |  |  |
| [**INSTALL\_SUPPORTFILESDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_install_supportfilesdir) | Extra support files directory | **PREFIX/support-files** |  |  |
| [**LINK\_RANDOMIZE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_link_randomize) | Whether to randomize order of symbols in mysqld binary | **OFF** |  |  |
| [**LINK\_RANDOMIZE\_SEED**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_link_randomize_seed) | Seed value for LINK\_RANDOMIZE option | **mysql** |  |  |
| [**MAX\_INDEXES**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_max_indexes) | Maximum indexes per table | **64** |  |  |
| [**MUTEX\_TYPE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_mutex_type) | InnoDB mutex type | **event** |  |  |
| [**MYSQLX\_TCP\_PORT**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_mysqlx_tcp_port) | TCP/IP port number used by X Plugin | **33060** |  |  |
| [**MYSQLX\_UNIX\_ADDR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_mysqlx_unix_addr) | Unix socket file used by X Plugin | **/tmp/mysqlx.sock** |  |  |
| [**MYSQL\_DATADIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_mysql_datadir) | Data directory |  |  |  |
| [**MYSQL\_MAINTAINER\_MODE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_mysql_maintainer_mode) | Whether to enable MySQL maintainer-specific development environment | **OFF** |  |  |
| [**MYSQL\_PROJECT\_NAME**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_mysql_project_name) | Windows/OS X project name | **MySQL** |  |  |
| [**MYSQL\_TCP\_PORT**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_mysql_tcp_port) | TCP/IP port number | **3306** |  |  |
| [**MYSQL\_UNIX\_ADDR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_mysql_unix_addr) | Unix socket file | **/tmp/mysql.sock** |  |  |
| [**NDB\_UTILS\_LINK\_DYNAMIC**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_ndb_utils_link_dynamic) | Cause NDB tools to be dynamically linked to ndbclient |  | 8.0.22-ndb-8.0.22 |  |
| [**ODBC\_INCLUDES**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_odbc_includes) | ODBC includes directory |  |  |  |
| [**ODBC\_LIB\_DIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_odbc_lib_dir) | ODBC library directory |  |  |  |
| [**OPTIMIZER\_TRACE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_optimizer_trace) | Whether to support optimizer tracing |  |  |  |
| [**REPRODUCIBLE\_BUILD**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_reproducible_build) | Take extra care to create a build result independent of build location and time |  |  |  |
| [**SYSCONFDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_sysconfdir) | Option file directory |  |  |  |
| [**SYSTEMD\_PID\_DIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_systemd_pid_dir) | Directory for PID file under systemd | **/var/run/mysqld** |  |  |
| [**SYSTEMD\_SERVICE\_NAME**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_systemd_service_name) | Name of MySQL service under systemd | **mysqld** |  |  |
| [**TMPDIR**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_tmpdir) | tmpdir default value |  |  |  |
| [**USE\_LD\_GOLD**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_use_ld_gold) | Whether to use GNU gold linker | **ON** |  |  |
| [**USE\_LD\_LLD**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_use_ld_lld) | Whether to use llvm lld linker | **ON** | 8.0.16 |  |
| [**WIN\_DEBUG\_NO\_INLINE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_win_debug_no_inline) | Whether to disable function inlining | **OFF** |  |  |
| [**WITHOUT\_xxx\_STORAGE\_ENGINE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_storage_engine_options) | Exclude storage engine xxx from build |  |  |  |
| [**WITH\_ANT**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_ant) | Path to Ant for building GCS Java wrapper |  |  |  |
| [**WITH\_ASAN**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_asan) | Enable AddressSanitizer | **OFF** |  |  |
| [**WITH\_ASAN\_SCOPE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_asan_scope) | Enable AddressSanitizer -fsanitize-address-use-after-scope Clang flag | **OFF** |  |  |
| [**WITH\_AUTHENTICATION\_CLIENT\_PLUGINS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_authentication_client_plugins) | Enabled automatically if any corresponding server authentication plugins are built |  | 8.0.25 |  |
| [**WITH\_AUTHENTICATION\_LDAP**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_authentication_ldap) | Whether to report error if LDAP authentication plugins cannot be built | **OFF** |  |  |
| [**WITH\_AUTHENTICATION\_PAM**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_authentication_pam) | Build PAM authentication plugin | **OFF** |  |  |
| [**WITH\_AWS\_SDK**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_aws_sdk) | Location of Amazon Web Services software development kit |  |  |  |
| [**WITH\_BOOST**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_boost) | The location of the Boost library sources |  |  |  |
| [**WITH\_CLIENT\_PROTOCOL\_TRACING**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_client_protocol_tracing) | Build client-side protocol tracing framework | **ON** |  |  |
| [**WITH\_CURL**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_curl) | Location of curl library |  |  |  |
| [**WITH\_DEBUG**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_debug) | Whether to include debugging support | **OFF** |  |  |
| [**WITH\_DEFAULT\_COMPILER\_OPTIONS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_default_compiler_options) | Whether to use default compiler options | **ON** |  |  |
| [**WITH\_DEFAULT\_FEATURE\_SET**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_default_feature_set) | Whether to use default feature set | **ON** |  | 8.0.22 |
| [**WITH\_EDITLINE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_editline) | Which libedit/editline library to use | **bundled** |  |  |
| [**WITH\_GMOCK**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_gmock) | Path to googlemock distribution |  |  | 8.0.25 |
| [**WITH\_ICU**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_icu) | Type of ICU support | **bundled** |  |  |
| [**WITH\_INNODB\_EXTRA\_DEBUG**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_innodb_extra_debug) | Whether to include extra debugging support for InnoDB. | **OFF** |  |  |
| [**WITH\_INNODB\_MEMCACHED**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_innodb_memcached) | Whether to generate memcached shared libraries. | **OFF** |  |  |
| [**WITH\_JEMALLOC**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_jemalloc) | Whether to link with -ljemalloc | **OFF** | 8.0.16 |  |
| [**WITH\_KEYRING\_TEST**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_keyring_test) | Build the keyring test program | **OFF** |  |  |
| [**WITH\_LIBEVENT**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_libevent) | Which libevent library to use | **bundled** |  |  |
| [**WITH\_LIBWRAP**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_libwrap) | Whether to include libwrap (TCP wrappers) support | **OFF** |  |  |
| [**WITH\_LOCK\_ORDER**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_lock_order) | Whether to enable LOCK\_ORDER tooling | **OFF** | 8.0.17 |  |
| [**WITH\_LSAN**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_lsan) | Whether to run LeakSanitizer, without AddressSanitizer | **OFF** | 8.0.16 |  |
| [**WITH\_LTO**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_lto) | Enable link-time optimizer | **OFF** | 8.0.13 |  |
| [**WITH\_LZ4**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_lz4) | Type of LZ4 library support | **bundled** |  |  |
| [**WITH\_LZMA**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_lzma) | Type of LZMA library support | **bundled** |  | 8.0.16 |
| [**WITH\_MECAB**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_mecab) | Compiles MeCab |  |  |  |
| [**WITH\_MSAN**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_msan) | Enable MemorySanitizer | **OFF** |  |  |
| [**WITH\_MSCRT\_DEBUG**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_mscrt_debug) | Enable Visual Studio CRT memory leak tracing | **OFF** |  |  |
| [**WITH\_MYSQLX**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_mysqlx) | Whether to disable X Protocol | **ON** |  |  |
| [**WITH\_NUMA**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_numa) | Set NUMA memory allocation policy |  |  |  |
| [**WITH\_PROTOBUF**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_protobuf) | Which Protocol Buffers package to use | **bundled** |  |  |
| [**WITH\_RAPID**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_rapid) | Whether to build rapid development cycle plugins | **ON** |  |  |
| [**WITH\_RAPIDJSON**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_rapidjson) | Type of RapidJSON support | **bundled** | 8.0.13 |  |
| [**WITH\_RE2**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_re2) | Type of RE2 library support | **bundled** |  | 8.0.18 |
| [**WITH\_ROUTER**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_router) | Whether to build MySQL Router | **ON** | 8.0.16 |  |
| [**WITH\_SSL**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_ssl) | Type of SSL support | **system** |  |  |
| [**WITH\_SYSTEMD**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_systemd) | Enable installation of systemd support files | **OFF** |  |  |
| [**WITH\_SYSTEMD\_DEBUG**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_systemd_debug) | Enable additional systemd debug information | **OFF** | 8.0.22 |  |
| [**WITH\_SYSTEM\_LIBS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_system_libs) | Set system value of library options not set explicitly | **OFF** |  |  |
| [**WITH\_TCMALLOC**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_tcmalloc) | Whether to link with -ltcmalloc | **OFF** | 8.0.22 |  |
| [**WITH\_TEST\_TRACE\_PLUGIN**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_test_trace_plugin) | Build test protocol trace plugin | **OFF** |  |  |
| [**WITH\_TSAN**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_tsan) | Enable ThreadSanitizer | **OFF** |  |  |
| [**WITH\_UBSAN**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_ubsan) | Enable Undefined Behavior Sanitizer | **OFF** |  |  |
| [**WITH\_UNIT\_TESTS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_unit_tests) | Compile MySQL with unit tests | **ON** |  |  |
| [**WITH\_UNIXODBC**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_unixodbc) | Enable unixODBC support | **OFF** |  |  |
| [**WITH\_VALGRIND**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_valgrind) | Whether to compile in Valgrind header files | **OFF** |  |  |
| [**WITH\_ZLIB**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_zlib) | Type of zlib support | **bundled** |  |  |
| [**WITH\_ZSTD**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_zstd) | Type of zstd support | **bundled** | 8.0.18 |  |
| [**WITH\_xxx\_STORAGE\_ENGINE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_storage_engine_options) | Compile storage engine xxx statically into server |  |  |  |

**一般选项**

* -DBUILD\_CONFIG=mysql\_release

该选项配置了一个源码发行版，其构建选项与Oracle为官方MySQL发行版生产二进制发行版所使用的相同。

* -DBUNDLE\_RUNTIME\_LIBRARIES=bool

是否将运行时库与Windows的服务器MSI和Zip包捆绑。

* -DCMAKE\_BUILD\_TYPE=type

要产生的构建类型。

* + RelWithDebInfo。启用优化并生成调试信息。这是默认的MySQL构建类型。
  + 发布。启用优化，但省略调试信息以减少构建规模。这种构建类型是在MySQL 8.0.13中添加的。
  + 调试。禁用优化并生成调试信息。如果启用了WITH\_DEBUG选项，也会使用这种构建类型。也就是说，-DWITH\_DEBUG=1与-DCMAKE\_BUILD\_TYPE=Debug的效果相同。
* -DCPACK\_MONOLITHIC\_INSTALL=bool

这个选项影响make package操作是产生多个安装包文件还是一个文件。如果禁用，该操作会产生多个安装包文件，如果你想只安装完整MySQL安装的一个子集，这可能很有用。如果启用，它将产生一个用于安装所有内容的单一文件。

* -DFORCE\_INSOURCE\_BUILD=bool

定义是否要强制进行源内构建。推荐使用源外构建，因为它允许从同一源头进行多次构建，并且可以通过删除构建目录快速进行清理。要强制进行源内构建，请用 -DFORCE\_INSOURCE\_BUILD=ON 来调用 CMake。

**安装布局选项**

CMAKE\_INSTALL\_PREFIX选项表示基本安装目录。其他名称为INSTALL\_xxx的选项表示组件的位置，是相对于前缀解释的，其值是相对路径名。它们的值不应该包括前缀。

* -DCMAKE\_INSTALL\_PREFIX=dir\_name

安装基础目录。

这个值可以在服务器启动时通过-basedir选项来设置。

* -DINSTALL\_BINDIR=dir\_name

安装用户程序的地方。

-DINSTALL\_DOCDIR=dir\_name

安装文档的地方。

* -DINSTALL\_DOCREADMEDIR=dir\_name

安装README文件的地方。

* -DINSTALL\_INCLUDEDIR=dir\_name

安装头文件的地方。

* -DINSTALL\_INFODIR=dir\_name

安装信息文件的地方。

* -DINSTALL\_LAYOUT=name

选择一个预定义的安装布局。

* + STANDALONE：与用于.tar.gz和.zip包的布局相同。这是默认的。
  + RPM。类似于RPM包的布局。
  + SVR4: Solaris软件包布局。
  + DEB: DEB包的布局（实验性的）。

你可以选择一个预定义的布局，但通过指定其他选项来修改单个组件的安装位置。比如说。

cmake . -DINSTALL\_LAYOUT=SVR4 -DMYSQL\_DATADIR=/var/mysql/data

INSTALL\_LAYOUT值决定了secure\_file\_priv、keyring\_encrypted\_file\_data和keyring\_file\_data系统变量的默认值。参见第5.1.8节 "服务器系统变量 "和第6.4.4.18节 "钥匙圈系统变量 "中关于这些变量的描述。

* -DINSTALL\_LIBDIR=dir\_name

安装库文件的位置。

* -DINSTALL\_MANDIR=dir\_name

安装手册页的地方。

* -DINSTALL\_MYSQLKEYRINGDIR=dir\_path

作为keyring\_file插件数据文件位置的默认目录。默认值与平台有关，取决于INSTALL\_LAYOUT CMake选项的值；参见5.1.8节 "服务器系统变量 "中关于keyring\_file\_data系统变量的描述。

* -DINSTALL\_MYSQLSHAREDIR=dir\_name

安装共享数据文件的位置。

* -DINSTALL\_MYSQLTESTDIR=dir\_name

在哪里安装mysql-test目录。要抑制这个目录的安装，明确地将该选项设置为空值（-DINSTALL\_MYSQLTESTDIR=）。

* -DINSTALL\_PKGCONFIGDIR=dir\_name

安装mysqlclient.pc文件的目录，供pkg-config使用。默认值是INSTALL\_LIBDIR/pkgconfig，除非INSTALL\_LIBDIR以/mysql结尾，在这种情况下会首先删除。

* -DINSTALL\_PLUGINDIR=dir\_name

插件目录的位置。

这个值可以在服务器启动时用--plugin\_dir选项设置。

* -DINSTALL\_PRIV\_LIBDIR=dir\_name

动态库目录的位置。

默认位置。RPM = /usr/lib64/mysql/private/，DEB = /usr/lib/mysql/private/，和TAR = lib/private/。

这个选项是在MySQL 8.0.18中添加的。

对于Protobuf：因为这是一个私有的位置，加载器（例如Linux上的ld-linux.so）在没有帮助的情况下可能找不到libprotobuf.so文件。为了指导加载器，在mysqld和mysqlxtest中加入了RPATH，其值为$ORIGIN/.../$INSTALL\_PRIV\_LIBDIR。这在大多数情况下是有效的，但当使用资源组功能时，mysqld被设置为id，然后加载器忽略了包含$ORIGIN的RPATH。为了克服这个问题，在DEB和RPM变体的mysqld中设置了一个明确的目录全路径，因为目标目的地是已知的。对于tarball安装，需要用patchelf这样的工具对mysqld进行修补。

* -DINSTALL\_SBINDIR=dir\_name

在哪里安装mysqld服务器。

* -DINSTALL\_SECURE\_FILE\_PRIVDIR=dir\_name
* secure\_file\_priv 系统变量的默认值。默认值与平台有关， 并取决于 INSTALL\_LAYOUT CMake 选项的值； 请参阅第 5.1.8 节 "服务器系统变量" 中关于 secure\_file\_priv 系统变量的描述。
* -DINSTALL\_SHAREDIR=dir\_name

安装aclocal/mysql.m4的位置。

* -DINSTALL\_STATIC\_LIBRARIES=bool

是否安装静态库。默认为ON。如果设置为OFF，则不安装这些库：libmysqlclient.a, libmysqlservices.a。

* -DINSTALL\_SUPPORTFILESDIR=dir\_name

将额外的支持文件安装在哪里。

* -DLINK\_RANDOMIZE=bool

是否随机化mysqld二进制中的符号顺序。默认为关闭。这个选项应该只为调试目的而启用。

* -DLINK\_RANDOMIZE\_SEED=val

LINK\_RANDOMIZE选项的种子值。该值是一个字符串。默认是mysql，这是一个任意的选择。

* -DMYSQL\_DATADIR=dir\_name

MySQL数据目录的位置。

这个值可以在服务器启动时用-datadir选项设置。

* -DODBC\_INCLUDES=dir\_name

ODBC包括目录的位置，可在配置Connector/ODBC时使用。

* -DODBC\_LIB\_DIR=dir\_name

ODBC库目录的位置，可在配置Connector/ODBC时使用。

* -DSYSCONFDIR=dir\_name

默认的my.cnf选项文件目录。

这个位置不能在服务器启动时设置，但你可以使用--defaults-file=file\_name选项以给定的选项文件启动服务器，其中file\_name是该文件的全路径名称。

* -DSYSTEMD\_PID\_DIR=dir\_name

当MySQL由systemd管理时，创建PID文件的目录名称。默认为/var/run/mysqld；可能会根据INSTALL\_LAYOUT值隐式改变。

除非启用 WITH\_SYSTEMD，否则该选项被忽略。

* -DSYSTEMD\_SERVICE\_NAME=name

当MySQL由systemd管理时，要使用的MySQL服务的名称。默认为 mysqld；根据 INSTALL\_LAYOUT 值，该名称可能被隐含地改变。

除非启用WITH\_SYSTEMD，否则该选项被忽略。

* -DTMPDIR=dir\_name

默认使用的tmpdir系统变量的位置。如果没有指定，该值默认为<stdio.h>中的P\_tmpdir。

**存储引擎选项**

存储引擎是作为插件构建的。你可以将插件构建为静态模块（编译到服务器中）或动态模块（构建为动态库，在使用前必须使用INSTALL PLUGIN语句或--plugin-load选项安装到服务器中）。有些插件可能不支持静态或动态构建。

InnoDB、MyISAM、MERGE、MEMORY和CSV引擎是强制性的（总是被编译到服务器中），不需要明确安装。

要将存储引擎静态地编译到服务器中，请使用-DWITH\_engine\_STORAGE\_ENGINE=1。一些允许的引擎值是：ARCHIVE、BLACKHOLE、EXAMPLE、FEDERATED、NDB或NDBCLUSTER（支持NDB）。例如。

-DWITH\_ARCHIVE\_STORAGE\_ENGINE=1

-DWITH\_BLACKHOLE\_STORAGE\_ENGINE=1

注意事项

在没有Performance Schema支持的情况下，是不可能进行编译的。如果需要在没有特定类型的工具的情况下进行编译，可以通过以下CMake选项来实现。

DISABLE\_PSI\_COND

DISABLE\_PSI\_DATA\_LOCK

DISABLE\_PSI\_ERROR

DISABLE\_PSI\_FILE

DISABLE\_PSI\_IDLE

DISABLE\_PSI\_MEMORY

DISABLE\_PSI\_METADATA

DISABLE\_PSI\_MUTEX

DISABLE\_PSI\_PS

DISABLE\_PSI\_RWLOCK

DISABLE\_PSI\_SOCKET

DISABLE\_PSI\_SP

DISABLE\_PSI\_STAGE

DISABLE\_PSI\_STATEMENT

DISABLE\_PSI\_STATEMENT\_DIGEST

DISABLE\_PSI\_TABLE

DISABLE\_PSI\_THREAD

DISABLE\_PSI\_TRANSACTION

例如，如果要在没有互斥工具的情况下编译，使用-DDISABLE\_PSI\_MUTEX=1选项配置MySQL。

要从构建中排除一个存储引擎，使用-DWITH\_engine\_STORAGE\_ENGINE=0。 示例。

-DWITH\_ARCHIVE\_STORAGE\_ENGINE=0

-DWITH\_EXAMPLE\_STORAGE\_ENGINE=0

-DWITH\_FEDERATED\_STORAGE\_ENGINE=0

也可以使用 -DWITHOUT\_engine\_STORAGE\_ENGINE=1 将存储引擎排除在构建之外（但首选是 -DWITH\_engine\_STORAGE\_ENGINE=0）。例子。

-DWITHOUT\_ARCHIVE\_STORAGE\_ENGINE=1

-DWITHOUT\_EXAMPLE\_STORAGE\_ENGINE=1

-DWITHOUT\_FEDERATED\_STORAGE\_ENGINE=1

如果既没有为某个存储引擎指定-DWITH\_engine\_STORAGE\_ENGINE，也没有指定-DWITHOUT\_engine\_STORAGE\_ENGINE，则该引擎将作为共享模块构建，如果不能作为共享模块构建，则排除。

**特性选项**

* -DADD\_GDB\_INDEX=bool

该选项决定是否启用在二进制文件中生成.gdb\_index部分，这可以使它们在调试器中的加载速度更快。该选项在默认情况下是禁用的。如果使用ld链接器，则该选项被禁用。 如果使用ld或GNU gold以外的链接器，则该选项没有影响。

这个选项是在MySQL 8.0.18中添加的。

* -DCOMPILATION\_COMMENT=string

关于编译环境的描述性注释。从MySQL 8.0.14开始，mysqld使用COMPILATION\_COMMENT\_SERVER。其他程序继续使用COMPILATION\_COMMENT。

* -DCOMPRESS\_DEBUG\_SECTIONS=bool

是否压缩二进制可执行文件的调试部分（仅Linux）。压缩可执行文件的调试部分可以节省空间，但在构建过程中要付出额外的CPU时间。

默认是关闭。如果没有明确设置这个选项，但设置了COMPRESS\_DEBUG\_SECTIONS环境变量，则该选项从该变量中取值。

这个选项是在MySQL 8.0.22中添加的。

* -DCOMPILATION\_COMMENT\_SERVER=string

关于编译环境的描述性注释，供mysqld使用（例如，设置version\_comment系统变量）。这个选项是在MySQL 8.0.14中添加的。在8.0.14之前，服务器使用COMPILATION\_COMMENT。

* -DDEFAULT\_CHARSET=charset\_name

服务器的字符集。默认情况下，MySQL使用utf8mb4字符集。

charset\_name可以是二进制、armcii8、ascii、big5、cp1250、cp1251、cp1256、cp1257、cp850、cp852、cp866、cp932、dec8、eucjpms、euckr、gb2312、gbk、geostd8, greek, hebrew, hp8, keybcs2, koi8r, koi8u, latin1, latin2, latin5, latin7, macce, macroman, sjis, swe7, tis620, ucs2, ujis, utf8, utf8mb4, utf16, utf16le, utf32. 允许的字符集在cmake/character\_sets.cmake文件中作为CHARSETS\_AVAILABLE的值列出。

这个值可以在服务器启动时用 --character\_set\_server 选项来设置。

* -DDEFAULT\_COLLATION=collation\_name

服务器的拼写方式。默认情况下，MySQL使用utf8mb4\_0900\_ai\_ci。使用SHOW COLLATION语句来确定每个字符集有哪些排序。

这个值可以在服务器启动时用--collation\_server选项设置。

* -DDISABLE\_PSI\_COND=bool

是否排除Performance Schema条件下的仪表。默认为OFF（包括）。

* -DDISABLE\_PSI\_FILE=bool

是否排除Performance Schema文件检测。默认值是关闭（包括）。

* -DDISABLE\_PSI\_IDLE=bool

是否排除Performance Schema的空闲仪表。缺省是关闭（包括）。

* -DDISABLE\_PSI\_MEMORY=bool

是否排除Performance Schema的内存检测。缺省是关闭（包括）。

* -DDISABLE\_PSI\_METADATA=bool

是否排除对Performance Schema元数据的检测。缺省是关闭（包括）。

* -DDISABLE\_PSI\_MUTEX=bool

是否排除Performance Schema mutex工具的使用。缺省是关闭（包括）。

* -DDISABLE\_PSI\_RWLOCK=bool

是否排除Performance Schema rwlock工具的使用。缺省是关闭（包括）。

* -DDISABLE\_PSI\_SOCKET=bool

是否排除Performance Schema套接字工具的使用。缺省是关闭（包括）。

* -DDISABLE\_PSI\_SP=bool

是否排除对Performance Schema存储程序的检测。缺省是关闭（包括）。

* -DDISABLE\_PSI\_STAGE=bool

是否排除Performance Schema的阶段工具。缺省是关闭（包括）。

* -DDISABLE\_PSI\_STATEMENT=bool

是否排除对Performance Schema语句的检测。缺省是关闭（包括）。

* -DDISABLE\_PSI\_STATEMENT\_DIGEST=bool

是否排除Performance Schema语句摘要工具的使用。缺省是关闭（包括）。

* -DDISABLE\_PSI\_TABLE=bool

是否排除对Performance Schema表的检测。缺省是关闭（包括）。

* -DDISABLE\_SHARED=bool

是否禁止构建共享库和编译依赖位置的代码。默认为OFF（编译与位置无关的代码）。

这个选项没有使用，在MySQL 8.0.18中被删除。

* -DDISABLE\_PSI\_PS=bool

排除性能模式准备好的语句实例仪表。默认是关闭（包括）。

* -DDISABLE\_PSI\_THREAD=bool

排除性能模式的线程仪表。默认是关闭（包括）。

只有在构建时没有任何工具化时才禁用线程，因为其他工具化对线程有依赖性。

* -DDISABLE\_PSI\_TRANSACTION=bool

排除性能模式的事务仪表。默认为OFF（包括）。

* -DDISABLE\_PSI\_DATA\_LOCK=bool

排除性能模式的数据锁检测。默认值是关闭（包括）。

* -DDISABLE\_PSI\_ERROR=bool

排除性能模式的服务器错误检测。默认值是关闭（包括）。

* -DDOWNLOAD\_BOOST=bool

是否下载Boost库。默认为关闭。

参见WITH\_BOOST选项，了解关于使用Boost的更多讨论。

* -DDOWNLOAD\_BOOST\_TIMEOUT=seconds

下载Boost库的超时时间，单位是秒。默认是600秒。

关于使用Boost的其他讨论，请参见WITH\_BOOST选项。

* -DENABLE\_DOWNLOADS=bool

是否要下载可选文件。例如，启用该选项后，CMake会下载测试套件用于运行单元测试的Google Test发行版，或构建GCS Java包装器所需的Ant和JUnit。

从MySQL 8.0.25开始，MySQL源码发行版捆绑了Google Test源代码，用于运行基于Google Test的单元测试。因此，从该版本开始，WITH\_GMOCK和ENABLE\_DOWNLOADS CMake选项被删除，如果被指定，则被忽略。

* -DENABLE\_EXPERIMENTAL\_SYSVARS=bool

是否启用实验性InnoDB系统变量。实验性系统变量是为从事MySQL开发的人准备的，只应在开发或测试环境中使用，并可能在未来的MySQL版本中不经通知就被删除。关于实验性系统变量的信息，请参考MySQL源代码树中的/storage/innobase/handler/ha\_innodb.cc。实验性系统变量可以通过搜索 "PLUGIN\_VAR\_EXPERIMENTAL "来识别。

* -DENABLE\_GCOV=bool

是否包括gcov支持（仅Linux）。

-DENABLE\_GPROF=bool

是否启用 gprof (仅优化的 Linux 构建)。

* -DENABLED\_LOCAL\_INFILE=bool

该选项控制了MySQL客户端库的默认LOCAL编译能力。因此，没有明确安排的客户端会根据在MySQL构建时指定的ENABLED\_LOCAL\_INFILE设置禁用或启用LOCAL能力。

默认情况下，MySQL二进制发行版中的客户端库是在禁用ENABLED\_LOCAL\_INFILE的情况下编译的。如果你从源码编译MySQL，根据没有明确安排的客户端是否应该禁用或启用LOCAL能力，分别以ENABLED\_LOCAL\_INFILE禁用或启用的方式配置它。

ENABLED\_LOCAL\_INFILE控制客户端LOCAL能力的默认值。对于服务器，local\_infile系统变量控制服务器端的LOCAL能力。要明确地使服务器拒绝或允许LOAD DATA LOCAL语句(不管客户程序和库在构建时或运行时是如何配置的)，在禁用或启用local\_infile的情况下启动mysqld。 local\_infile也可以在运行时设置。参见第6.1.6节，"LOAD DATA LOCAL的安全考虑"。

* -DENABLED\_PROFILING=bool

是否启用查询分析代码（用于SHOW PROFILE和SHOW PROFILES语句）。

* -DFORCE\_UNSUPPORTED\_COMPILER=bool

默认情况下，CMake检查支持的编译器的最小版本。Visual Studio 2015（Windows）；GCC 4.8或Clang 3.4（Linux）；Developer Studio 12.5（Solaris服务器）；Developer Studio 12.4或GCC 4.8（Solaris客户端库）；Clang 3.6（macOS），Clang 3.4（FreeBSD）。要禁用这个检查，请使用 -DFORCE\_UNSUPPORTED\_COMPILER=ON。

* -DFPROFILE\_GENERATE=bool

是否生成配置文件引导的优化（PGO）数据。这个选项适用于用GCC进行PGO实验。有关使用FPROFILE\_GENERATE和FPROFILE\_USE的信息，请参见MySQL源码分发中的cmake/fprofile.cmake文件。这些选项已在GCC 8和9中测试过。

这个选项是在MySQL 8.0.19中添加的。

* -DFPROFILE\_USE=bool

是否使用配置文件引导的优化（PGO）数据。这个选项可用于在GCC中试验PGO。有关使用FPROFILE\_GENERATE和FPROFILE\_USE的信息，请参见MySQL源码分发中的cmake/fprofile.cmake文件。这些选项已经在GCC 8和9中进行了测试。

启用FPROFILE\_USE还可以启用WITH\_LTO。

这个选项是在MySQL 8.0.19中添加的。

* -DHAVE\_PSI\_MEMORY\_INTERFACE=bool

是否启用用于动态存储超对齐类型的内存分配函数（ut::aligned\_name库函数）的性能模式内存跟踪模块。

-DIGNORE\_AIO\_CHECK=bool

如果在Linux上给出了-DBUILD\_CONFIG=mysql\_release选项，libaio库必须被默认链接进来。如果你没有libaio或者不想安装它，你可以通过指定-DIGNORE\_AIO\_CHECK=1抑制对它的检查。

* -DMAX\_INDEXES=num

每个表的最大索引数。默认是64。最大为255。小于64的值被忽略，使用默认的64。

* -DMYSQL\_MAINTAINER\_MODE=bool

是否启用MySQL维护者专用开发环境。如果启用，该选项会使编译器警告变成错误。

* -DMUTEX\_TYPE=type

InnoDB使用的mutex类型。选项包括。

事件。使用事件互斥。这是默认值，也是最初的InnoDB mutex实现。

sys: 在UNIX系统上使用POSIX mutexes。如果有的话，在Windows上使用CRITICAL\_SECTION onjects。

futex。使用Linux的futexes而不是条件变量来安排等待的线程。

* -DMYSQLX\_TCP\_PORT=port\_num

X Plugin用于监听TCP/IP连接的端口号。默认是33060。

这个值可以在服务器启动时通过mysqlx\_port系统变量设置。

* -DMYSQLX\_UNIX\_ADDR=file\_name

服务器监听X插件套接字连接的Unix套接字文件路径。这必须是一个绝对路径名。默认是/tmp/mysqlx.sock。

这个值可以在服务器启动时通过mysqlx\_port系统变量设置。

* -DMYSQL\_PROJECT\_NAME=name

对于Windows或macOS，项目名称要纳入项目文件名。

* -DMYSQL\_TCP\_PORT=port\_num

服务器监听TCP/IP连接的端口号。默认为3306。

这个值可以在服务器启动时用--port选项设置。

* -DMYSQL\_UNIX\_ADDR=file\_name

服务器监听套接字连接的Unix套接字文件路径。这必须是一个绝对路径名。默认是/tmp/mysql.sock。

这个值可以在服务器启动时通过-socket选项设置。

* -DOPTIMIZER\_TRACE=bool

是否支持优化器跟踪。见MySQL内部。追踪优化器。

* -DREPROCIBLE\_BUILD=bool

对于在Linux系统上的构建，该选项控制是否采取额外的措施来创建一个独立于构建地点和时间的构建结果。

这个选项是在MySQL 8.0.11中添加的。从MySQL 8.0.12开始，对于RelWithDebInfo的构建，它默认为ON。

-DUSE\_LD\_GOLD=bool

如果GNU gold链接器可用且未明确禁用，CMake会使构建过程与它链接。要禁止使用该链接器，请指定 -DUSE\_LD\_GOLD=OFF 选项。

* -DUSE\_LD\_LLD=bool

如果Clang的llvm lld链接器可用，并且没有被明确禁用，CMake会使构建过程与它链接。要禁止使用该链接器，指定-DUSE\_LD\_LLD=OFF选项。

这个选项是在MySQL 8.0.16中添加的。

* -DWIN\_DEBUG\_NO\_INLINE=bool

是否在Windows上禁用函数内联。默认是关闭（启用内联）。

* -DWITH\_ANT=path\_name

设置Ant的路径，在构建GCS Java包装器时需要。与现有的 WITH\_BOOST CMake 选项的工作方式类似。将 WITH\_ANT 设为保存 Ant tarball 或已经解压的归档文件的目录路径。当 WITH\_ANT 没有被设置，或者被设置为特殊值系统时，构建会假定 $PATH 中存在一个二进制 ant。

* -DWITH\_ASAN=bool

是否为支持 AddressSanitizer 的编译器启用该功能。默认为关闭。

* -DWITH\_ASAN\_SCOPE=bool

是否启用AddressSanitizer -fsanite-address-use-after-scope Clang标志，用于检测使用后范围。默认为关闭。要使用这个选项，必须同时启用-DWITH\_ASAN。

* -DWITH\_AUTHENTICATION\_CLIENT\_PLUGINS=bool

如果构建了任何相应的服务器认证插件，该选项将自动启用。因此，它的值取决于其他CMake选项，不应明确设置。

这个选项是在MySQL 8.0.25中添加的。

* -DWITH\_AUTHENTICATION\_LDAP=bool

如果LDAP认证插件不能被构建，是否报告错误。

如果这个选项被禁用（默认），如果找到所需的头文件和库，LDAP插件就会被构建。如果没有，CMake会显示一个相关说明。

如果这个选项被激活，如果找不到所需的头文件和库，CMake会产生一个错误，阻止服务器被构建。

* -DWITH\_AUTHENTICATION\_PAM=bool

是否构建PAM认证插件，用于包含该插件的源码树。(参见第6.4.1.5节，"PAM可插拔认证"。)如果指定了这个选项，而该插件不能被编译，则构建失败。

* -DWITH\_AWS\_SDK=path\_name

亚马逊网络服务软件开发工具包的位置。

* -DWITH\_BOOST=path\_name

构建MySQL时需要Boost库。这些CMake选项可以控制库的来源位置，以及是否自动下载。

* -DWITH\_BOOST=path\_name指定Boost库的目录位置。也可以通过设置BOOST\_ROOT或WITH\_BOOST环境变量来指定Boost的位置。
* -DWITH\_BOOST=system也是允许的，表示正确的Boost版本已经安装在编译主机的标准位置。在这种情况下，会使用已安装的Boost版本，而不是MySQL源码分发中包含的任何版本。
* -DDOWNLOAD\_BOOST=bool指定如果Boost源码不在指定位置，是否要下载它。默认是关闭。
* -DDOWNLOAD\_BOOST\_TIMEOUT=seconds下载Boost库的超时时间，单位为秒。默认值是600秒。

例如，如果你通常在构建MySQL时将对象输出放在MySQL源码树的bld子目录下，你可以像这样用Boost构建。

mkdir bld

cd bld

cmake .. -DDOWNLOAD\_BOOST=ON -DWITH\_BOOST=$HOME/my\_boost

这将导致Boost被下载到你主目录下的my\_boost目录。如果所需的Boost版本已经在那里了，则不进行下载。如果所需的Boost版本改变了，就会下载较新的版本。

如果本地已经安装了Boost，并且你的编译器可以自己找到Boost头文件，那么可能就没有必要指定前面的CMake选项。然而，如果MySQL所需的Boost版本改变了，而本地安装的版本没有升级，你可能会遇到构建问题。使用CMake选项应该可以成功构建。

在上述允许Boost下载到指定位置的设置下，当所需的Boost版本改变时，你需要删除bld文件夹，重新创建，并再次执行cmake步骤。否则，新的Boost版本可能不会被下载，而且编译可能会失败。

* -DWITH\_CLIENT\_PROTOCOL\_TRACING=bool

是否在客户端库中构建客户端协议追踪框架。默认情况下，这个选项是启用的。

关于编写协议跟踪客户端插件的信息，请参见编写协议跟踪插件。

另请参见WITH\_TEST\_TRACE\_PLUGIN选项。

* -DWITH\_CURL=curl\_type

curl库的位置。curl\_type可以是系统（使用系统的curl库）或curl库的路径名。

* -DWITH\_DEBUG=bool

是否包括调试支持。

配置具有调试支持的MySQL使你能够在启动服务器时使用--debug="d,parser\_debug "选项。这将导致用于处理SQL语句的Bison解析器向服务器的标准错误输出转储一个解析器跟踪。通常情况下，这个输出被写到错误日志中。

InnoDB存储引擎的同步调试检查定义在UNIV\_DEBUG下，当使用WITH\_DEBUG选项编译调试支持时，可以使用。当调试支持被编译进来时，innodb\_sync\_debug配置选项可以用来启用或禁用InnoDB同步调试检查。

启用WITH\_DEBUG也可以启用调试同步。这个功能是用来测试和调试的。当编译进来时，Debug Sync在运行时默认是禁用的。要启用它，用--debug-sync-timeout=N选项启动mysqld，其中N是一个大于0的超时值（默认值是0，禁用Debug Sync），N成为单个同步点的默认超时。

当使用WITH\_DEBUG选项编译了调试支持时，InnoDB存储引擎的同步调试检查是可用的。

关于Debug Sync设施和如何使用同步点的描述，请参见MySQL内部。测试同步。

* -DWITH\_DEFAULT\_FEATURE\_SET=bool

是否使用cmake/build\_configurations/feature\_set.cmake的标志。这个选项在MySQL 8.0.22中被删除。

* -DWITH\_EDITLINE=value

要使用哪个libedit/editline库。允许的值是捆绑（默认）和系统。

* -DWITH\_GMOCK=path\_name

到googlemock分布的路径，用于基于Google Test的单元测试。该选项的值是分布Zip文件的路径。或者，将WITH\_GMOCK环境变量设置为路径名。也可以使用-DENABLE\_DOWNLOADS=1，这样CMake就会从GitHub下载发行版。

如果你构建的MySQL没有基于Google Test的单元测试（通过配置wihout WITH\_GMOCK），CMake会显示一条信息，说明如何下载它。

从MySQL 8.0.25开始，MySQL源码发行版捆绑了Google Test源代码，用于运行基于Google Test的单元测试。因此，从该版本开始，WITH\_GMOCK和ENABLE\_DOWNLOADS CMake选项被删除，如果被指定，则被忽略。

* -DWITH\_ICU={icu\_type|path\_name}。

MySQL使用International Components for Unicode（ICU）来支持正则表达式操作。WITH\_ICU选项表示要包括的ICU支持类型或要使用的ICU安装路径名称。

* icu\_type可以是以下值之一。
  + bundled:使用与发行版捆绑的ICU库。这是默认的，也是唯一支持Windows的选项。
  + system: 使用系统ICU库。
* path\_name 是要使用的 ICU 安装的路径名称。这可能比使用icu\_type值system更好，因为它可以防止CMake检测到并使用系统上安装的旧的或不正确的ICU版本。(另一种允许的方式是将 WITH\_ICU 设为 system， 并将 CMAKE\_PREFIX\_PATH 选项设为 path\_name。)
* -DWITH\_INNODB\_EXTRA\_DEBUG=bool

是否包含额外的InnoDB调试支持。

启用 WITH\_INNODB\_EXTRA\_DEBUG 会打开额外的 InnoDB 调试检查。这个选项只有在 WITH\_DEBUG 被启用时才能被启用。

* -DWITH\_INNODB\_MEMCACHED=bool

是否生成memcached共享库（libmemcached.so和innodb\_engine.so）。

* -DWITH\_JEMALLOC=bool

是否用-ljemalloc链接。如果启用，内置的malloc()、calloc()、realloc()和free()程序将被禁用。默认是关闭。

WITH\_JEMALLOC和WITH\_TCMALLOC是相互排斥的。

这个选项是在MySQL 8.0.16中添加的。

* -DWITH\_KEYRING\_TEST=bool

是否要建立伴随keyring\_file插件的测试程序。默认是关闭。测试文件的源代码位于plugin/keyring/keyring-test目录中。

* -DWITH\_LIBEVENT=string

要使用哪个libevent库。允许的值是捆绑（默认）和系统。在MySQL 8.0.21之前，如果你指定了system，如果存在的话，就会使用系统libevent库，否则就会发生错误。在MySQL 8.0.21及以后的版本中，如果指定了system而没有找到系统libevent库，则无论如何都会发生错误，并且不使用捆绑的libevent。

InnoDB memcached、X Plugin和MySQL Router需要libevent库。

-DWITH\_LIBWRAP=bool

是否包括libwrap（TCP包装器）支持。

* -DWITH\_LOCK\_ORDER=bool

是否启用LOCK\_ORDER工具。默认情况下，这个选项是禁用的，服务器构建不包含工具。如果启用了工具，LOCK\_ORDER工具是可用的，可以按照第5.9.3节 "LOCK\_ORDER工具 "中的描述使用。

注意

在启用WITH\_LOCK\_ORDER选项时，MySQL的构建需要使用flex程序。

这个选项是在MySQL 8.0.17中添加的。

* -DWITH\_LSAN=bool

是否运行LeakSanitizer，而不运行AddressSanitizer。默认为关闭。

这个选项是在MySQL 8.0.16中添加的。

* -DWITH\_LTO=bool

如果编译器支持，是否启用链接时优化器。默认是关闭，除非启用了PROFILE\_USE。

这个选项是在MySQL 8.0.13中添加的。

* -DWITH\_LZ4=lz4\_type

WITH\_LZ4表示zlib支持的来源。

bundled:使用与发行版捆绑的lz4库。这是默认的。

system: 使用系统的 lz4 库。如果 WITH\_LZ4 被设置为这个值， lz4\_decompress 工具不会被建立。在这种情况下，可以使用系统的 lz4 命令来代替。

* -DWITH\_LZMA=lzma\_type

lzma\_type可以是下列值之一。

* **bundled**:的。使用与发行版捆绑的LZMA库。这是默认的。
* system: 使用系统LZMA库。

这个选项在MySQL 8.0.16中被删除。

-DWITH\_MECAB={disabled|system|path\_name}。

使用该选项来编译MeCab解析器。如果你已经将MeCab安装到它的默认安装目录，请设置-DWITH\_MECAB=system。系统选项适用于从源码或使用本地软件包管理工具从二进制文件进行的MeCab安装。如果您将MeCab安装到一个自定义的安装目录，请指定MeCab的安装路径。例如，-DWITH\_MECAB=/opt/mecab。如果系统选项不起作用，指定MeCab的安装路径在所有情况下都应该起作用。

关于相关信息，请参见第12.10.9节，"MeCab全文解析器插件"。

* -DWITH\_MSAN=bool

是否启用MemorySanitizer，用于支持它的编译器。默认是关闭。

为了使该选项在启用时产生效果，所有与MySQL链接的库必须在启用该选项的情况下进行编译。

* -DWITH\_MSCRT\_DEBUG=bool

是否启用Visual Studio CRT内存泄漏跟踪。默认为关闭。

* -DWITH\_MYSQLX=bool

是否支持X插件的构建。默认为ON。参见第20章，将MySQL用作文档存储。

* -DWITH\_NUMA=bool

明确设置NUMA内存分配策略。CMake根据当前平台是否有NUMA支持来设置默认的WITH\_NUMA值。对于没有NUMA支持的平台，CMake的行为如下。

在没有NUMA选项的情况下（正常情况），CMake继续正常工作，只产生这个警告。NUMA库缺失或所需版本不可用

在-DWITH\_NUMA=ON的情况下，CMake会以这样的错误中止。NUMA库缺失或所需版本不可用

* -DWITH\_PROTOBUF=protobuf\_type

使用哪个协议缓冲包。protobuf\_type可以是下列值之一。

bundled:使用与发行版捆绑的软件包。这是默认的。可以选择使用INSTALL\_PRIV\_LIBDIR来修改动态Protobuf库目录。

system。使用安装在系统上的软件包。

其他值会被忽略，并回退到捆绑的状态。

* -DWITH\_RAPID=bool

是否要构建快速开发周期插件。启用时，会在构建树中创建一个包含这些插件的快速目录。当禁用时，在构建树中不创建快速目录。默认值是ON，除非快速目录从源树中删除，在这种情况下，默认值变为OFF。

* -DWITH\_RAPIDJSON=rapidjson\_type

包含的RapidJSON库支持类型。 rapidjson\_type可以是以下值之一。

bundled:使用与发行版捆绑的 RapidJSON 库。这是默认的。

系统。使用系统的RapidJSON库。需要1.1.0或更高的版本。

这个选项是在MySQL 8.0.13中添加的。

* -DWITH\_RE2=re2\_type

要包括的RE2库支持类型。re2\_type可以是以下值之一。

bundled:使用与发行版捆绑的RE2库。这是默认的。

系统。使用系统RE2库。

从MySQL 8.0.18开始，MySQL不再使用RE2库，这个选项被删除。

* -DWITH\_ROUTER=bool

是否建立MySQL Router。默认为ON。

这个选项是在MySQL 8.0.16中添加的。

* -DWITH\_SSL={ssl\_type|path\_name}。

为了支持加密连接、用于随机数生成的熵以及其他与加密有关的操作，必须使用SSL库构建MySQL。这个选项指定了要使用哪个SSL库。

* ssl\_type可以是以下值之一。
  + 系统。使用系统的OpenSSL库。这是默认值。

在macOS和Windows上，使用system配置MySQL，就像调用CMake时path\_name指向一个手动安装的OpenSSL库。这是因为它们没有系统SSL库。在macOS上，brew install openssl会安装到/usr/local/opt/openssl，这样系统就能找到它。在Windows上，它会检查%ProgramFiles%/OpenSSL、%ProgramFiles%/OpenSSL-Win32、%ProgramFiles%/OpenSSL-Win64、C:/OpenSSL、C:/OpenSSL-Win32和C:/OpenSSL-Win64。

* yes这是系统的一个同义词。

path\_name是要使用的OpenSSL安装的路径名称。这比使用系统的ssl\_type值要好，因为它可以防止CMake检测到并使用系统上安装的旧的或不正确的OpenSSL版本。另一种允许的方式是将WITH\_SSL设置为system，并将CMAKE\_PREFIX\_PATH选项设置为path\_name）。

关于配置SSL库的其他信息，参见第2.9.6节 "配置SSL库支持"。

* -DWITH\_SYSTEMD=bool

是否启用 systemd 支持文件的安装。默认情况下，该选项为禁用。启用后，将安装systemd支持文件，而不安装mysqld\_safe和System V初始化脚本等脚本。在 systemd 不可用的平台上，启用 WITH\_SYSTEMD 会导致 CMake 出错。

关于使用systemd的更多信息，请参见2.5.9节 "用systemd管理MySQL服务器"。该节还包括关于指定先前在 [mysqld\_safe] 选项组中指定的选项的信息。由于使用systemd时，mysqld\_safe不会被安装，因此此类选项必须以其他方式指定。

* -DWITH\_SYSTEM\_LIBS=bool

该选项作为一个 "总括 "选项，为以下没有明确设置的CMake选项设置系统值。with\_curl, with\_editline, with\_icu, with\_libevent, with\_lz4, with\_lzma, with\_protobuf, with\_re2, with\_ssl, with\_zlib, with\_zstd。

* -DWITH\_SYSTEMD\_DEBUG=bool

对于使用systemd运行MySQL的平台，是否要产生额外的systemd调试信息。默认为关闭。

该选项在MySQL 8.0.22中加入。

* -DWITH\_TCMALLOC=bool

是否与-ltcmalloc链接。如果启用，内置的malloc()、calloc()、realloc()和free()程序将被禁用。默认是关闭。

WITH\_TCMALLOC和WITH\_JEMALLOC是相互排斥的。

这个选项是在MySQL 8.0.22中添加的。

* -DWITH\_TEST\_TRACE\_PLUGIN=bool

是否要建立测试协议跟踪客户端插件（见使用测试协议跟踪插件）。默认情况下，这个选项是禁用的。除非启用WITH\_CLIENT\_PROTOCOL\_TRACING选项，否则启用该选项没有效果。如果MySQL在配置时启用了这两个选项，libmysqlclient客户端库在构建时就内置了测试协议跟踪插件，而且所有标准的MySQL客户端都会加载该插件。然而，即使测试插件被启用，它在默认情况下也没有效果。对该插件的控制是通过环境变量来实现的；参见使用测试协议跟踪插件。

注意

如果你想使用你自己的协议跟踪插件，不要启用WITH\_TEST\_TRACE\_PLUGIN选项，因为一次只能加载一个这样的插件，如果试图加载第二个，会出现错误。如果你已经在启用了测试协议跟踪插件的情况下构建了MySQL，看看它是如何工作的，你必须在使用你自己的插件之前重建没有它的MySQL。

关于编写跟踪插件的信息，见编写协议跟踪插件。

* -DWITH\_TSAN=bool

是否为支持它的编译器启用ThreadSanitizer。默认是关闭。

* -DWITH\_UBSAN=bool

是否启用未定义行为净化器（Undefined Behavior Sanitizer），适用于支持该功能的编译器。缺省是关闭。

* -DWITH\_UNIT\_TEST={ON|OFF}。

如果启用，用单元测试编译MySQL。默认为ON，除非服务器没有被编译。

* -dwith\_unixodbc=1

启用unixODBC支持，用于Connector/ODBC。

* -DWITH\_VALGRIND=bool

是否在Valgrind头文件中进行编译，这将Valgrind API暴露给MySQL代码。默认是关闭。

为了生成一个Valgrind感知的调试版本，-DWITH\_VALGRIND=1通常与-DWITH\_DEBUG=1结合。参见构建调试配置。

* -DWITH\_ZLIB=zlib\_type

一些功能要求服务器在构建时支持压缩库，比如COMPRESS()和UNCOMPRESS()函数，以及客户端/服务器协议的压缩。WITH\_ZLIB表示zlib支持的来源。

* bundled:使用与发行版捆绑的zlib库。这是默认的。
* system: 使用系统的 zlib 库。如果 WITH\_ZLIB 被设置为这个值，zlib\_decompress 工具不会被构建。在这种情况下，可以使用系统的 openssl zlib 命令来代替。
* -DWITH\_ZSTD=zstd\_type

使用zstd算法的连接压缩（见第4.2.8节，"连接压缩控制"）要求服务器在构建时支持zstd库。WITH\_ZSTD表示zstd支持的来源。

bundled:使用与发行版捆绑的 zstd 库。这是默认的。

system: 使用系统的zstd库。

这个选项是在MySQL 8.0.18中添加的。

**编译器标志**

* -DCMAKE\_C\_FLAGS="flags"

用于C语言编译器的标志。

* -DCMAKE\_CXX\_FLAGS="flags"

用于C++编译器的标志。

* -DWITH\_DEFAULT\_COMPILER\_OPTIONS=bool

是否使用cmake/build\_configurations/compiler\_options.cmake中的标志。

注意

所有优化标志都是由MySQL构建团队精心选择和测试的。覆盖它们可能导致意外的结果，风险自负。

要指定你自己的C和C++编译器标志，对于不影响优化的标志，使用CMAKE\_C\_FLAGS和CMAKE\_CXX\_FLAGS CMake选项。

当提供你自己的编译器标志时，你可能想同时指定CMAKE\_BUILD\_TYPE。

例如，要在64位的Linux机器上创建一个32位的发布版本，请这样做。

mkdir bld

cd bld

cmake .. -DCMAKE\_C\_FLAGS=-m32 \

-DCMAKE\_CXX\_FLAGS=-m32 \

-DCMAKE\_BUILD\_TYPE=RelWithDebInfo

如果你设置了影响优化的标志（-Onumber），你必须设置CMAKE\_C\_FLAGS\_build\_type和/或CMAKE\_CXX\_FLAGS\_build\_type选项，其中build\_type对应于CMAKE\_BUILD\_TYPE值。要为默认的编译类型（RelWithDebInfo）指定不同的优化，请设置CMAKE\_C\_FLAGS\_RELWITHDEBINFO和CMAKE\_CXX\_FLAGS\_RELWITHDEBINFO选项。例如，要在Linux上用-O3和调试符号进行编译，请这样做。

cmake .. -DCMAKE\_C\_FLAGS\_RELWITHDEBINFO="-O3 -g" \

-DCMAKE\_CXX\_FLAGS\_RELWITHDEBINFO="-O3 -g"

**用于编译NDB集群的CMake选项**

以下选项是在构建支持NDB Cluster的MySQL 8.0源时使用的。

* -DMEMCACHED\_HOME=dir\_name

使用安装在dir\_name指示的系统目录中的memcached（1.6或更高版本）进行构建。在构建中使用的该安装文件包括memcached二进制文件、头文件和库，以及memcached\_utilities库和头文件engine\_testapp.h。

在使用捆绑的memcached源（WITH\_BUNDLED\_MEMCACHED选项）构建ndbmemcache时，你必须不设置这个选项；换句话说，默认情况下使用捆绑的源）。

虽然额外的CMake选项--例如SASL授权和提供dtrace支持--在从外部资源编译memcached时可以使用，但这些选项目前还没有对与NDB Cluster捆绑的memcached源启用。

* -ndb\_utils\_link\_dynamic={ON|OFF}。

控制NDB工具（如ndb\_drop\_table）是与ndbclient静态链接（OFF）还是动态链接（ON）；默认为OFF（静态链接）。通常在构建这些工具时使用静态链接，以避免LD\_LIBRARY\_PATH的问题，或者在安装多个版本的ndbclient时使用。这个选项是为了创建Docker镜像，以及其他可能的情况，即目标环境受到精确控制，并且希望减少镜像的大小。

在NDB 8.0.22中添加。

* -dwith\_bundled\_libevent={On|off}

在构建支持ndbmemcached的NDB Cluster时，使用NDB Cluster源中包含的libevent。默认情况下是启用的。OFF会导致使用系统的libevent来代替。

* -dwith\_bundled\_memcached={on|off}

构建包含在NDB Cluster源代码树中的memcached源，然后在构建ndbmemcache引擎时使用产生的memcached服务器。在这种情况下，make install将memcached二进制文件放在安装bin目录下，将ndbmemcache引擎共享库文件ndb\_engine.so放在安装lib目录下。

这个选项在默认情况下是打开的。

-DWITH\_CLASSPATH=path

设置用于构建NDB Cluster Connector for Java的classpath。默认为空。如果使用-DWITH\_NDB\_JAVA=OFF，这个选项将被忽略。

-DWITH\_ERROR\_INSERT={ON|OFF}在NDB集群连接器中启用错误注入。

启用NDB内核中的错误注入。仅用于测试；不打算用于构建生产二进制文件。默认值是OFF。

* -dwith\_ndbcluster\_storage\_engine={on|off}

这是 WITH\_NDBCLUSTER 的一个别名。

-dwith\_ndbcluster={on|off} 构建并链接对NBCLUSTER的支持。

在mysqld中建立并链接对NDB（NDBCLUSTER）存储引擎的支持。默认为ON。

-dwith\_ndbmtd={on|off}

建立多线程数据节点可执行文件ndbmtd。默认值为ON。

* -dwith\_ndb\_binlog={on|off}默认为打开。

在使用该选项构建的mysqld中默认启用二进制日志。默认为打开。

* -dwith\_ndb\_debug={on|off} 启用构建调试版本。

启用构建NDB Cluster二进制文件的调试版本。默认情况下为关闭。

* -dwith\_ndb\_java={on|off} 启用构建带有Java的NDB集群。

启用构建NDB Cluster的Java支持，包括ClusterJ。

该选项默认为打开。如果你不希望用Java支持来编译NDB Cluster，你必须在运行CMake时指定-DWITH\_NDB\_JAVA=OFF来明确禁用它。否则，如果找不到Java，配置构建就会失败。

* -DWITH\_NDB\_PORT=port

导致所构建的NDB Cluster管理服务器（ndb\_mgmd）默认使用这个端口。如果这个选项没有被设置，那么生成的管理服务器默认会尝试使用1186端口。

* -dwith\_ndb\_test={on|off}。

如果启用，包括一组NDB API测试程序。默认为关闭。

* -dwith\_plugin\_ndbcluster={on|off} 如果启用，包括一组NDB API测试程序。

WITH\_NDBCLUSTER的别名。

### 2.9.8 处理编译MySQL时遇到的问题

许多问题的解决方案涉及重新配置。如果你真的重新配置，请注意以下几点。

* 如果CMake在之前被运行后被运行，它可能会使用在之前的调用中收集的信息。这些信息被保存在CMakeCache.txt中。当CMake启动时，它会寻找该文件并读取其内容（如果它存在），前提是这些信息仍然正确。当你重新配置时，这个假设就无效了。
* 每次运行CMake时，你必须再次运行make来重新编译。然而，你可能想先删除以前构建的旧对象文件，因为它们是用不同的配置选项编译的。

为了防止旧的对象文件或配置信息被使用，在重新运行CMake之前运行以下命令。

在Unix上。

shell> **make clean**

shell> **rm CMakeCache.txt**

在Windows上。

shell> **devenv MySQL.sln /clean**

shell> **del CMakeCache.txt**

如果你在源码树之外构建，在重新运行CMake之前删除并重新创建你的构建目录。关于在源码树外构建的说明，见如何用CMake构建MySQL服务器。

在某些系统上，由于系统包含文件的不同，可能会出现警告。下面的列表描述了在编译MySQL时发现的最常发生的其他问题。

要定义使用哪个C和C++编译器，你可以定义CC和CXX环境变量。例如

shell> **CC=gcc**

shell> **CXX=g++**

shell> **export CC CXX**

要指定你自己的C和C++编译器标志，可以使用CMAKE\_C\_FLAGS和CMAKE\_CXX\_FLAGS CMake选项。参见编译器标志。

要查看你可能需要指定哪些标志，用--cflags和--cxxflags选项调用mysql\_config。

* 要查看在编译阶段执行什么命令，在使用CMake配置MySQL后，运行make VERBOSE=1，而不是仅仅运行make。
* 如果编译失败，检查MYSQL\_MAINTAINER\_MODE选项是否被启用。这种模式导致编译器警告变成错误，所以禁用它可能会使编译继续进行。
* 如果你的编译失败了，出现了以下的错误，你必须把你的make版本升级到GNU make。
* make: Fatal error in reader: Makefile, line 18:
* Badly formed macro assignment

或者。

make: file `Makefile' line 18: Must be a separator (:

或者。

pthread.h: No such file or directory

众所周知，Solaris和FreeBSD的make程序很麻烦。

已知GNU make 3.75可以工作。

* sql\_yacc.cc文件是由sql\_yacc.yy生成的。通常，构建过程不需要创建sql\_yacc.cc，因为MySQL带有一个预先生成的副本。然而，如果你确实需要重新创建它，你可能会遇到这个错误。

"sql\_yacc.yy"，第xxx行致命的：默认动作导致潜在的......

这是一个迹象，表明你的yacc版本有缺陷。你可能需要安装最新版本的bison（yacc的GNU版本）并使用它来代替。

早于1.75的bison版本可能会报告这个错误。

sql\_yacc.yy:#####: fatal error: maximum table size (32767) exceeded

实际上并没有超过最大表的大小；这个错误是由旧版本的bison的bug引起的。

关于获取或更新工具的信息，请参见第2.9节 "从源头安装MySQL "中的系统要求。

### 2.9.9 MySQL配置和第三方工具

需要从MySQL源头确定MySQL版本的第三方工具可以读取顶级源头目录中的VERSION文件。该文件分别列出了版本的各个部分。例如，如果版本是MySQL 8.0.4-rc，该文件看起来像这样。

MYSQL\_VERSION\_MAJOR=8

MYSQL\_VERSION\_MINOR=0

MYSQL\_VERSION\_PATCH=4

MYSQL\_VERSION\_EXTRA=-rc

如果源码不是通用的（GA）版本，MYSQL\_VERSION\_EXTRA的值是不空的。在刚才的例子中，该值对应于 "Release Candidate"。

要从版本组件中构建一个五位数的数字，使用这个公式。

MYSQL\_VERSION\_MAJOR\*10000 + MYSQL\_VERSION\_MINOR\*100 + MYSQL\_VERSION\_PATCH

### 2.9.10 生成MySQL Doxygen文档内容

MySQL源代码包含使用Doxygen编写的内部文档。生成的Doxygen内容可在https://dev.mysql.com/doc/index-other.html。也可以使用下面的程序从MySQL源码分布中本地生成这些内容。

1、安装doxygen 1.8.11或更高版本。分发版可在这里获得：http://www.doxygen.nl/。

安装doxygen后，验证版本号。

shell> **doxygen --version**

1.8.13

2、安装PlantUML。

当你在Windows上安装PlantUML时（在Windows 10上测试），你必须至少以管理员身份运行一次，以便它创建注册表键。打开一个管理员控制台，运行这个命令。

shell> java -jar ***path-to-plantuml.jar***

该命令应该打开一个GUI窗口，并且在控制台中没有返回错误。

3、将 PLANTUML\_JAR\_PATH 环境设置为你安装 PlantUML 的位置。比如说

shell> export PLANTUML\_JAR\_PATH=***path-to-plantuml.jar***

4、安装Graphviz dot命令。

安装Graphviz后，验证dot的可用性。例如。

shell> **which dot**

/usr/bin/dot

shell> **dot -V**

dot - graphviz version 2.28.0 (20130928.0220)

5、改变位置到你的MySQL源分发的顶层目录，并做以下工作。

首先，执行cmake。

shell> cd your-mysql-source-directory

shell> mkdir bld

shell> cd bld

shell> cmake ..

接下来，生成doxygen文档。

shell> make doxygen

检查错误日志。它可以在顶层目录的doxyerror.log文件中找到。假设构建成功执行，用浏览器查看生成的输出。例如。

shell> firefox doxygen/html/index.html

## 2.10 安装后的设置和测试

本节讨论了安装MySQL后应执行的任务。

* 如果有必要，初始化数据目录并创建MySQL授予表。对于某些MySQL安装方法，数据目录初始化可能会自动为你完成。
  + 由MySQL安装程序执行的Windows安装操作。
  + 在Linux上使用Oracle的服务器RPM或Debian发行版进行安装。
  + 在许多平台上使用本地打包系统进行安装，包括Debian Linux、Ubuntu Linux、Gentoo Linux和其他。
  + 在macOS上使用DMG发行版进行安装。

对于其他平台和安装类型，你必须手动初始化数据目录。这包括在Unix和类Unix系统上从通用的二进制和源码发行版进行安装，以及在Windows上从ZIP档案包进行安装。有关说明，请参阅第2.10.1节，"初始化数据目录"。

* 启动服务器并确保它可以被访问。说明见第2.10.2节，"启动服务器"，和第2.10.3节，"测试服务器"。
* 如果在数据目录初始化过程中没有为初始根账户分配密码，请在授予表中为其分配密码。密码可以防止对MySQL服务器的非法访问。有关说明，见第2.10.4节 "确保初始MySQL账户的安全"。
* 可以选择安排服务器在你的系统启动和停止时自动启动和停止。有关说明，见第2.10.5节 "自动启动和停止MySQL"。
* 可选地，填充时区表，以便能够识别指定的时区。有关说明，见第5.1.15节，"MySQL服务器时区支持"。

当你准备创建其他用户账户时，你可以在第6.2节 "访问控制和账户管理 "中找到关于MySQL访问控制系统和账户管理的信息。

### 2.10.1 初始化数据目录

在MySQL安装后，必须初始化数据目录，包括mysql系统模式中的表。

* 对于某些MySQL安装方法，数据目录的初始化是自动的，如第2.10节 "安装后的设置和测试 "中所述。
* 对于其他安装方法，你必须手动初始化数据目录。这些方法包括在Unix和类似Unix的系统上从通用的二进制和源码分布中安装，以及在Windows上从ZIP档案包中安装。

本节描述了对于数据目录初始化不是自动的MySQL安装方法，如何手动初始化数据目录。关于一些建议的命令，可以测试服务器是否可以访问和正常工作，见第2.10.3节 "测试服务器"。

注意

在MySQL 8.0中，默认的认证插件已经从mysql\_native\_password变为caching\_sha2\_password，'root'@'localhost'管理账户默认使用caching\_sha2\_password。如果你喜欢根账户使用以前的默认认证插件（mysql\_native\_password），请参阅caching\_sha2\_password和根管理账户。

* 数据目录初始化概述
* 数据目录初始化程序
* 数据目录初始化期间的服务器操作
* 初始化后根密码的分配

**数据目录初始化概述**

在这里显示的例子中，服务器打算在mysql登录账户的用户ID下运行。如果该账户不存在，可以创建该账户（参见创建mysql用户和组），或者用你计划用于运行服务器的另一个现有登录账户的名称代替。

1、改变位置到你的MySQL安装的顶级目录，通常是/usr/local/mysql（根据需要为你的系统调整路径名称）。

cd /usr/local/mysql

在这个目录中，你可以找到几个文件和子目录，包括包含服务器的bin子目录，以及客户端和实用程序。

2、secure\_file\_priv系统变量将导入和导出操作限制在一个特定的目录中。创建一个目录，其位置可以被指定为该变量的值。

mkdir mysql-files

将目录的用户和组所有权授予mysql用户和mysql组，并适当地设置目录的权限。

chown mysql:mysql mysql-files

chmod 750 mysql-files

3、使用服务器初始化数据目录，包括包含初始MySQL授予表的mysql模式，该表决定用户如何被允许连接到服务器。例如

bin/mysqld --initialize --user=mysql

关于该命令的重要信息，特别是关于你可能使用的命令选项，见数据目录初始化程序。关于服务器如何执行初始化的细节，见数据目录初始化期间的服务器动作。

通常，数据目录的初始化只需要在你第一次安装MySQL之后进行。(对于现有安装的升级，请执行升级程序；见第2.11节，"升级MySQL")。然而，初始化数据目录的命令不会覆盖任何现有的mysql模式表，所以它在任何情况下都可以安全地运行。

4、如果你想在部署服务器时自动支持安全连接，使用mysql\_ssl\_rsa\_setup工具来创建默认的SSL和RSA文件。

bin/mysql\_ssl\_rsa\_setup

更多信息，见第4.4.3节 "mysql\_ssl\_rsa\_setup--创建SSL/RSA文件"。

5、如果没有任何选项文件，服务器以其默认设置启动。见第5.1.2节 "服务器配置默认值"。）要明确指定MySQL服务器在启动时应使用的选项，请将它们放在一个选项文件中，如/etc/my.cnf或/etc/mysql/my.cnf。见第4.2.2.2节，"使用选项文件"。）例如，你可以使用选项文件来设置secure\_file\_priv系统变量。

6、要安排MySQL在系统启动时无需人工干预而启动，见第2.10.5节，"自动启动和停止MySQL"。

7、数据目录初始化在mysql模式中创建时区表，但不填充它们。要这样做，请使用第5.1.15节 "MySQL服务器时区支持 "中的说明。

**数据目录初始化程序**

改变位置到你的MySQL安装的顶级目录，通常是/usr/local/mysql（根据需要为你的系统调整路径名称）。

cd /usr/local/mysql

要初始化数据目录，用--initialize或--initialize-insecure选项调用mysqld，这取决于你是否希望服务器为 "root'@'localhost "账户生成一个随机的初始密码，或者创建没有密码的账户。

* 使用-initialize进行 "默认安全 "安装（即包括生成一个随机的初始root密码）。在这种情况下，密码被标记为过期，你必须选择一个新密码。
* 使用 --initialize-insecure，不生成根密码。这是不安全的；假定你打算在将服务器投入生产使用前及时给账户分配一个密码。

关于分配一个新的'root'@'localhost'密码的说明，请看初始化后的root密码分配。

注意

服务器会将任何信息（包括任何初始密码）写入其标准错误输出。这可能会被重定向到错误日志，所以如果你在屏幕上没有看到这些信息，就去那里看看。关于错误日志的信息，包括它的位置，见第5.4.2节，"错误日志"。

在Windows上，使用--console选项将信息导向控制台。

在Unix和类似Unix的系统中，数据库目录和文件由mysql登录账户拥有是很重要的，这样当你以后运行服务器时就有对它们的读写权限。为了确保这一点，从系统根账户启动mysqld，并包括--user选项，如下面所示。

bin/mysqld --initialize --user=mysql

bin/mysqld --initialize-insecure --user=mysql

或者，在以mysql身份登录时执行mysqld，在这种情况下，你可以省略命令中的--用户选项。

在Windows上，使用这些命令中的一个。

bin\mysqld --initialize --console

bin\mysqld --initialize-insecure --console

注意

如果缺少必要的系统库，数据目录初始化可能会失败。例如，你可能会看到一个类似这样的错误。

bin/mysqld: error while loading shared libraries:

libnuma.so.1: 无法打开共享对象文件。

没有这样的文件或目录

如果发生这种情况，你必须手动或用你系统的软件包管理器安装缺少的库。然后重试数据目录初始化命令。

如果mysqld不能识别安装目录或数据目录的正确位置，可能需要指定其他选项，如-basedir或-datadir。例如（在单行输入命令）。

bin/mysqld --initialize --user=mysql

--basedir=/opt/mysql/mysql

--datadir=/opt/mysql/mysql/data

或者，将相关的选项设置放在一个选项文件中，并将该文件的名称传递给mysqld。对于Unix和类Unix系统，假设选项文件名是/opt/mysql/mysql/etc/my.cnf。把这些行放在文件中。

[mysqld]

basedir=/opt/mysql/mysql

datadir=/opt/mysql/mysql/data

然后按如下方式调用mysqld（在单行输入命令，先输入--defaults-file选项）。

bin/mysqld --defaults-file=/opt/mysql/mysql/etc/my.cnf

--initialize --user=mysql

在Windows上，假设C:\my.ini包含这些行。

[mysqld]

basedir=C:\\Program Files\\MySQL\\MySQL Server 8.0

datadir=D:\\MySQLdata

然后按以下方式调用mysqld（在单行输入命令，先输入--defaults-file选项）。

bin\mysqld --defaults-file=C:\my.ini

--initialize --console

**数据目录初始化期间的服务器操作**

注意

服务器执行的数据目录初始化序列并不能替代mysql\_secure\_installation和mysql\_ssl\_rsa\_setup执行的操作。参见第4.4.2节 "mysql\_secure\_installation--提高MySQL安装安全性"，以及第4.4.3节 "mysql\_ssl\_rsa\_setup--创建SSL/RSA文件"。

当用-initialize或-initialize-insecure选项调用时，mysqld在数据目录初始化序列中执行下列操作。

1、服务器检查数据目录是否存在，如下所示。

* 如果没有数据目录存在，服务器会创建它。
* 如果数据目录存在但不是空的（也就是说，它包含文件或子目录），服务器在产生一个错误信息后退出。
* [ERROR] --initialize specified but the data directory exists. Aborting.

在这种情况下，删除或重命名数据目录，然后重试。

如果每个条目都有一个以句号（.）开头的名称，则允许现有的数据目录为非空。

2、在数据目录中，服务器创建了mysql系统模式及其表，包括数据字典表、授予表、时区表和服务器端帮助表。参见第5.3节，"mysql系统模式"。

3、服务器初始化系统表空间和管理InnoDB表所需的相关数据结构。

注意

在mysqld设置了InnoDB系统表空间后，对表空间特性的某些改变需要设置一个全新的实例。符合条件的变化包括系统表空间中第一个文件的文件名和Undo logs的数量。如果你不想使用默认值，在运行mysqld之前，确保innodb\_data\_file\_path和innodb\_log\_file\_size配置参数的设置在MySQL配置文件中已经就绪。还要确保在必要时指定影响InnoDB文件创建和位置的其他参数，例如innodb\_data\_home\_dir和innodb\_log\_group\_home\_dir。

如果这些选项在你的配置文件中，但该文件不在MySQL默认读取的位置，在运行mysqld时使用-defaults-extra-file选项指定文件位置。

4、服务器创建一个'root'@'localhost'超级用户账户和其他保留账户（见第6.2.9节，"保留账户"）。一些保留账户被锁定，不能被客户使用，但'root'@'localhost'是用于管理的，你应该给它分配一个密码。

服务器对'root'@'localhost'账户的密码的操作取决于你如何调用它。

* 使用 --initialize 但不使用 --initialize-insecure，服务器会生成一个随机密码，将其标记为过期，并写一个显示密码的消息。

[Warning] A temporary password is generated for root@localhost:

iTag\*AfrH5ej

* 使用 --initialize-insecure，（无论是否有 --initialize，因为 --initialize-insecure 意味着 --initialize），服务器不会生成密码或标记为过期，并写下一条警告信息。

[Warning] root@localhost is created with an empty password ! Please

consider switching off the --initialize-insecure option.

关于分配新的'root'@'localhost'密码的说明，请参阅初始化后的root密码分配。

5、服务器会填充用于HELP语句的服务器端帮助表（参见第13.8.3节，"HELP语句"）。服务器不填充时区表。要手动操作，请参见第5.1.15节 "MySQL服务器时区支持"。

6、如果init\_file系统变量被赋予了一个SQL语句文件的名称，服务器就会执行该文件中的语句。这个选项使你能够执行自定义的引导序列。

当服务器在引导模式下运行时，一些限制文件中允许的语句的功能不可用。这些包括与账户管理（如CREATE USER或GRANT）、复制和全局事务标识符有关的语句。

服务器退出。

**初始化后根密码的分配**

在你通过使用 --initialize 或 --initialize-insecure 启动服务器来初始化数据目录后，正常启动服务器（也就是说，没有这两个选项）并为'root @'localhost'账户分配一个新密码。

1、启动服务器。有关说明，请参见第2.10.2节，"启动服务器"。

2、连接到服务器。

* 如果你使用了 --initialize 但没有使用 --initialize-insecure 来初始化数据目录，以root身份连接到服务器。
* mysql -u root -p

然后，在密码提示符下，输入服务器在初始化序列中生成的随机密码。

输入密码：（在此输入随机的根密码

如果你不知道这个密码，请查看服务器的错误日志。

* 如果你使用了--initialize-insecure来初始化数据目录，以root身份连接到服务器，不需要密码。

Enter password: ***(enter the random root password here)***

3、连接后，使用ALTER USER语句分配一个新的root密码。

ALTER USER 'root'@'localhost' IDENTIFIED BY '***root-password'***;

另见第2.10.4节，"确保初始MySQL账户的安全"。

注意

连接到主机127.0.0.1的尝试通常会解析到localhost账户。然而，如果服务器在运行时启用了skip\_name\_resolve，则会失败。如果你打算这样做，请确保存在一个能够接受连接的账户。例如，为了能够以root身份使用--host=127.0.0.1或--host=::1连接，创建这些账户。

CREATE USER 'root'@'127.0.0.1' IDENTIFIED BY '***root-password***';

CREATE USER 'root'@'::1' IDENTIFIED BY '***root-password***';

可以把这些语句放在一个文件中，使用init\_file系统变量来执行，正如在数据目录初始化过程中的服务器操作中所讨论的那样。

### 2.10.2 启动服务器

2.10.2.1 启动MySQL服务器的故障排除问题

本节描述了如何在Unix和类Unix系统上启动服务器。(对于Windows，见第2.3.4.5节，"首次启动服务器"。)对于一些建议的命令，你可以用来测试服务器是否可以访问和正常工作，见第2.10.3节，"测试服务器"。

如果你的安装包括mysqld\_safe，像这样启动MySQL服务器。

shell> **bin/mysqld\_safe --user=mysql &**

注意

对于使用RPM包安装MySQL的Linux系统，服务器的启动和关闭使用systemd而不是mysqld\_safe来管理，mysqld\_safe没有安装。参见第2.5.9节，"用systemd管理MySQL服务器"。

如果你的安装包含systemd支持，请像这样启动服务器。

shell> **systemctl start mysqld**

如果服务名称与mysqld不同，请替换相应的服务名称（例如，SLES系统上的mysql）。

使用非特权（非root）登录账户运行MySQL服务器是很重要的。为了确保这一点，以root身份运行mysqld\_safe，并包括--user选项，如图所示。否则，你应该在以mysql登录时执行该程序，在这种情况下，你可以从命令中省略--用户选项。

关于以非特权用户身份运行MySQL的进一步说明，见第6.1.5节，"如何以普通用户身份运行MySQL"。

如果命令立即失败并打印出mysqld结束，在错误日志（默认情况下是数据目录中的host\_name.err文件）中寻找信息。

如果服务器无法访问它启动的数据目录或读取mysql模式中的授予表，它就会在其错误日志中写一条信息。如果你在进行这一步之前忽略了通过初始化数据目录来创建授予表，或者你在运行初始化数据目录的命令时没有使用--用户选项，就可能出现这种问题。删除数据目录，用--用户选项运行命令。

如果你有其他启动服务器的问题，请参见第2.10.2.1节 "启动MySQL服务器的故障排除"。关于mysqld\_safe的更多信息，参见第4.3.2节，"mysqld\_safe--MySQL服务器启动脚本"。更多关于systemd支持的信息，参见第2.5.9节 "用systemd管理MySQL服务器"。

#### 2.10.2.1 启动MySQL服务器问题的故障排除

本节提供了启动服务器问题的故障排除建议。关于Windows系统的其他建议，请参见第2.3.5节 "Microsoft Windows MySQL Server安装的故障排除"。

如果你在启动服务器时遇到问题，这里有一些事情可以尝试。

* 检查错误日志，看看为什么服务器不能启动。日志文件位于数据目录中（在Windows中通常是C:\Program Files\MySQL\MySQL Server 8.0\data，对于Unix/Linux二进制发行版，是/usr/local/mysql/data，而对于Unix/Linux源发行版，是/usr/local/var）。在数据目录中寻找名称为host\_name.err和host\_name.log的文件，其中host\_name是你的服务器主机的名称。然后检查这些文件的最后几行。使用tail来显示它们。

shell> **tail *host\_name*.err**

shell> **tail *host\_name*.log**

* 指定你所使用的存储引擎需要的任何特殊选项。你可以创建一个 my.cnf 文件，为你计划使用的引擎指定启动选项。如果你打算使用支持事务表的存储引擎（InnoDB、NDB），请确保在启动服务器之前将它们配置成你想要的方式。如果你正在使用InnoDB表，请参阅第15.8节 "InnoDB配置 "的指导原则和第15.14节 "InnoDB启动选项和系统变量 "的选项语法。

尽管存储引擎对你省略的选项使用默认值，但Oracle建议你查看可用的选项，并为任何默认值不适合你安装的选项指定明确的值。

确保服务器知道在哪里可以找到数据目录。mysqld服务器使用该目录作为其当前目录。这是它期望找到数据库的地方，也是它期望写日志文件的地方。服务器还在data目录中写pid（进程ID）文件。

默认的数据目录位置是在编译服务器的时候硬编码的。要确定默认的路径设置是什么，用--verbose和--help选项调用mysqld。如果数据目录位于你系统的其他地方，请在命令行或选项文件中用 mysqld 或 mysqld\_safe 的 --datadir 选项指定该位置。否则，服务器不能正常工作。作为--datadir选项的替代方法，你可以用--basedir指定mysqld安装MySQL的基础目录的位置，mysqld会在那里寻找数据目录。

要检查指定路径选项的效果，用这些选项和--verbose及--help选项调用mysqld。例如，如果你改变位置到安装mysqld的目录，然后运行下面的命令，它显示了以/usr/local为基本目录启动服务器的效果。

shell> **./mysqld --basedir=/usr/local --verbose --help**

你也可以指定其他选项，如--datadir，但--verbose和--help必须是最后一个选项。

一旦你确定了你想要的路径设置，就在没有--verbose和--help的情况下启动服务器。

如果mysqld当前正在运行，你可以通过执行这个命令找出它正在使用的路径设置。

shell> **mysqladmin variables**

或者

shell> **mysqladmin -h *host\_name* variables**

host\_name是MySQL服务器主机的名称。

* 确保服务器可以访问数据目录。数据目录及其内容的所有权和权限必须允许服务器读取和修改它们。

如果你在启动mysqld时得到Errcode 13（这意味着Permission denied），这意味着数据目录或其内容的权限不允许服务器访问。在这种情况下，你要改变所涉及的文件和目录的权限，使服务器有权利使用它们。你也可以以root身份启动服务器，但这引起了安全问题，应该避免。

改变位置到数据目录，并检查数据目录及其内容的所有权，以确保服务器有访问权。例如，如果数据目录是/usr/local/mysql/var，使用这个命令。

shell> **ls -la /usr/local/mysql/var**

如果数据目录或其文件或子目录不属于你用来运行服务器的登录账户，请将其所有权改为该账户。如果该账户名为mysql，使用这些命令。

shell> **chown -R mysql /usr/local/mysql/var**

shell> **chgrp -R mysql /usr/local/mysql/var**

即使有正确的所有权，如果你的系统上有其他安全软件在运行，管理应用程序对文件系统各部分的访问，MySQL可能无法启动。在这种情况下，重新配置该软件，使mysqld能够访问它在正常操作中使用的目录。

* 验证服务器想要使用的网络接口是否可用。

如果出现以下任一错误，意味着其他程序（也许是另一个 mysqld 服务器）正在使用 mysqld 试图使用的 TCP/IP 端口或 Unix 套接字文件。

Can't start server: Bind on TCP/IP port: Address already in use

Can't start server: Bind on unix socket...

使用ps来确定你是否有另一个mysqld服务器在运行。如果是，在再次启动 mysqld 之前关闭该服务器。(如果另一个服务器正在运行，并且你真的想运行多个服务器，你可以在第5.8节 "在一台机器上运行多个MySQL实例 "中找到关于如何做的信息)。

如果没有其他服务器正在运行，执行命令telnet your\_host\_name tcp\_ip\_port\_number。(默认的MySQL端口号是3306。)然后按回车键数次。如果你没有得到一个类似telnet的错误信息。无法连接到远程主机：连接被拒绝，一些其他程序正在使用 mysqld 试图使用的 TCP/IP 端口。追踪这是什么程序并禁用它，或者用--端口选项告诉mysqld监听一个不同的端口。在这种情况下，当使用TCP/IP连接到服务器时，为客户程序指定相同的非默认端口号。

该端口无法访问的另一个原因是，你有一个正在运行的防火墙，阻止了对它的连接。如果是这样，请修改防火墙设置，允许对该端口的访问。

如果服务器启动了，但你不能连接到它，确保你在/etc/hosts中有一个条目，看起来像这样。

127.0.0.1 localhost

如果你不能让mysqld启动，尝试通过使用--debug选项制作一个跟踪文件来找到问题所在。参见第5.9.4节，"DBUG包"。

### 2.10.3 测试服务器

在数据目录被初始化并且你已经启动了服务器之后，执行一些简单的测试以确保它能令人满意地工作。本节假设你的当前位置是MySQL安装目录，并且它有一个bin子目录，包含这里使用的MySQL程序。如果不是这样，请相应地调整命令路径名称。

或者，将bin目录添加到你的PATH环境变量设置中。这可以使你的shell（命令解释器）正确地找到MySQL程序，这样你就可以只输入程序的名称而不是路径名称来运行程序。见第4.2.9节，"设置环境变量"。

使用mysqladmin来验证服务器是否在运行。下面的命令提供了简单的测试，以检查服务器是否启动并对连接作出响应。

shell> **bin/mysqladmin version**

shell> **bin/mysqladmin variables**

如果你不能连接到服务器，指定一个-u root选项，以root身份连接。如果你已经为root账户分配了一个密码，你还需要在命令行中指定-p，并在提示时输入密码。比如说

shell> **bin/mysqladmin -u root -p version**

Enter password: ***(enter root password here)***

mysqladmin版本的输出略有不同，取决于你的平台和MySQL的版本，但应该与这里显示的类似。

shell> **bin/mysqladmin version**

mysqladmin Ver 14.12 Distrib 8.0.26, for pc-linux-gnu on i686

...

Server version 8.0.26

Protocol version 10

Connection Localhost via UNIX socket

UNIX socket /var/lib/mysql/mysql.sock

Uptime: 14 days 5 hours 5 min 21 sec

Threads: 1 Questions: 366 Slow queries: 0

Opens: 0 Flush tables: 1 Open tables: 19

Queries per second avg: 0.000

要看看你还能用mysqladmin做什么，用--help选项调用它。

确认你可以关闭服务器（如果根账户已经有了密码，包括一个-p选项）。

shell> **bin/mysqladmin -u root shutdown**

确认你可以再次启动服务器。通过使用 mysqld\_safe 或直接调用 mysqld 来做到这一点。例如。

shell> **bin/mysqld\_safe --user=mysql &**

如果mysqld\_safe失败，请参阅第2.10.2.1节，"启动MySQL服务器问题的故障排除"。

运行一些简单的测试以验证你可以从服务器中检索信息。输出应该与这里显示的类似。

使用mysqlshow来查看存在哪些数据库。

shell> **bin/mysqlshow**

+--------------------+

| Databases |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

| sys |

+--------------------+

已安装的数据库列表可能有所不同，但总是至少包括 mysql 和 information\_schema。

如果你指定了一个数据库名称，mysqlshow会显示该数据库中的表的列表。

shell> **bin/mysqlshow mysql**

Database: mysql

+---------------------------+

| Tables |

+---------------------------+

| columns\_priv |

| component |

| db |

| default\_roles |

| engine\_cost |

| func |

| general\_log |

| global\_grants |

| gtid\_executed |

| help\_category |

| help\_keyword |

| help\_relation |

| help\_topic |

| innodb\_index\_stats |

| innodb\_table\_stats |

| ndb\_binlog\_index |

| password\_history |

| plugin |

| procs\_priv |

| proxies\_priv |

| role\_edges |

| server\_cost |

| servers |

| slave\_master\_info |

| slave\_relay\_log\_info |

| slave\_worker\_info |

| slow\_log |

| tables\_priv |

| time\_zone |

| time\_zone\_leap\_second |

| time\_zone\_name |

| time\_zone\_transition |

| time\_zone\_transition\_type |

| user |

+---------------------------+

使用mysql程序从mysql模式中的一个表中选择信息。

shell> **bin/mysql -e "SELECT User, Host, plugin FROM mysql.user" mysql**

+------+-----------+-----------------------+

| User | Host | plugin |

+------+-----------+-----------------------+

| root | localhost | caching\_sha2\_password |

+------+-----------+-----------------------+

在这一点上，你的服务器正在运行，你可以访问它。如果你还没有给初始账户分配密码，要加强安全，请按照第2.10.4节 "确保初始MySQL账户的安全 "中的说明进行。

关于mysql、mysqladmin和mysqlshow的更多信息，见第4.5.1节 "mysql--MySQL命令行客户端"，第4.5.2节 "mysqladmin--MySQL服务器管理程序"，以及第4.5.7节 "mysqlshow--显示数据库、表和列的信息"。

### 2.10.4 保证初始MySQL账户的安全

MySQL的安装过程涉及初始化数据目录，包括mysql系统模式中定义MySQL账户的授予表。详情请参见第2.10.1节 "初始化数据目录"。

本节描述了如何为在MySQL安装过程中创建的初始根账户分配一个密码，如果你还没有这样做的话。

注意

执行本节所述过程的其他方法。

* 在Windows上，你可以在用MySQL安装程序安装时执行该过程（见第2.3.3节，"Windows的MySQL安装程序"）。
* 在所有平台上，MySQL发行版包括mysql\_secure\_installation，这是一个命令行工具，可自动完成保护MySQL安装的大部分过程。
* 在所有平台上，MySQL Workbench是可用的，并提供了管理用户账户的能力（见第31章，MySQL Workbench ）。

在这种情况下，可能已经为初始账户分配了一个密码。

* 在Windows上，使用MySQL安装程序进行的安装可以让你选择分配密码。
* 使用macOS安装程序进行的安装会生成一个初始随机密码，安装程序会在一个对话框中向用户显示。
* 使用RPM软件包的安装会产生一个初始随机密码，它被写入服务器错误日志。
* 使用Debian软件包的安装可以选择指定一个密码。
* 对于使用mysqld --initialize手动执行的数据目录初始化，mysqld会生成一个初始随机密码，标记为过期，并将其写入服务器错误日志。参见第2.10.1节，"初始化数据目录"。

mysql.user grant表定义了初始MySQL用户账户及其访问权限。安装MySQL只创建一个'root'@'localhost'超级用户账户，该账户拥有所有权限，可以做任何事情。如果root账户有一个空的密码，你的MySQL安装是不受保护的。任何人都可以在没有密码的情况下以root身份连接到MySQL服务器，并被授予所有权限。

root'@'localhost'账户在mysql.proxies\_priv表中也有一行，可以为''@'授予PROXY权限，也就是说，为所有用户和所有主机授予PROXY权限。这使得root可以设置代理用户，也可以将设置代理用户的权限委托给其他账户。见第6.2.18节，"代理用户"。

要为最初的MySQL根账户分配一个密码，请使用以下程序。用你想使用的密码替换例子中的root-password。

如果服务器没有运行，请启动它。有关说明，见第2.10.2节，"启动服务器"。

* 最初的root账户可能有密码，也可能没有。选择下列程序中适用的一种。

如果根账户存在一个已经过期的初始随机密码，用该密码以根身份连接到服务器，然后选择一个新密码。如果数据目录是使用mysqld --initialize初始化的，无论是手动还是使用安装程序，在安装操作中不给你指定密码的选项，就属于这种情况。因为密码存在，你必须使用它来连接到服务器。但是，由于密码已经过期，除了选择一个新的密码外，你不能将该账户用于任何其他目的，直到你选择一个新的密码。

* 如果你不知道最初的随机密码，请在服务器的错误日志中查找。
* 使用该密码以root身份连接到服务器。

shell> **mysql -u root -p**

Enter password: ***(enter the random root password here)***

* 选择一个新的密码来替换随机密码。

mysql> **ALTER USER 'root'@'localhost' IDENTIFIED BY '*root-password*';**

* 如果根账户存在但没有密码，以根身份连接到服务器，不使用密码，然后分配一个密码。如果你使用mysqld --initialize-insecure初始化了数据目录，就是这种情况。

1. 以root身份连接到服务器，不使用密码。

shell> **mysql -u root --skip-password**

1. 指定一个密码。
2. mysql> **ALTER USER 'root'@'localhost' IDENTIFIED BY '*root-password*';**

在给root账户分配了一个密码后，每当你使用该账户连接到服务器时，你必须提供该密码。例如，要使用mysql客户端连接到服务器，使用这个命令。

shell> **mysql -u root -p**

Enter password: ***(enter root password here)***

要用mysqladmin关闭服务器，使用这条命令。

shell> **mysqladmin -u root -p shutdown**

Enter password: ***(enter root password here)***

注意

关于设置密码的其他信息，见第6.2.14节，"分配账户密码"。如果你在设置根密码后忘记了，请参见第B.3.3.2节，"如何重设根密码"。

要设置其他账户，请参见第6.2.8节，"添加账户、分配权限和放弃账户"。

### 2.10.5 自动启动和停止MySQL

本节讨论了启动和停止MySQL服务器的方法。

一般来说，你以这些方式之一启动mysqld服务器。

* 直接调用mysqld。这在任何平台上都适用。
* 在Windows上，你可以设置一个MySQL服务，在Windows启动时自动运行。见第2.3.4.8节，"作为Windows服务启动MySQL"。
* 在Unix和类Unix系统上，你可以调用mysqld\_safe，它试图确定mysqld的适当选项，然后用这些选项运行它。见第4.3.2节，"mysqld\_safe--MySQL服务器启动脚本"。
* 在支持systemd的Linux系统上，你可以用它来控制服务器。参见第2.5.9节，"用systemd管理MySQL服务器"。
* 在使用System V风格的运行目录（即/etc/init.d和运行级特定目录）的系统上，调用mysql.server。这个脚本主要在系统启动和关机时使用。它通常以mysql的名字安装。mysql.server脚本通过调用mysqld\_safe启动服务器。见第4.3.3节，"mysql.server--MySQL服务器启动脚本"。
* 在macOS上，安装一个 launchd 守护进程，以便在系统启动时自动启动 MySQL。该守护进程通过调用mysqld\_safe启动服务器。详情请参见第2.4.3节 "安装和使用MySQL启动守护程序"。一个MySQL首选项窗格也提供了通过系统首选项启动和停止MySQL的控制。见第2.4.4节，"安装和使用MySQL首选项窗格"。
* 在Solaris上，使用服务管理框架（SMF）系统来启动和控制MySQL的启动。

systemd、mysqld\_safe和mysql.server脚本、Solaris SMF和macOS启动项（或MySQL首选项窗格）可用于手动启动服务器，或在系统启动时自动启动。 systemd、mysql.server和启动项还可用于停止服务器。

下表显示了服务器和启动脚本从选项文件中读取哪些选项组。

表2.15 MySQL启动脚本和支持的服务器选项组

|  |  |
| --- | --- |
| **Script** | **Option Groups** |
| [**mysqld**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#mysqld) | **[mysqld]**, **[server]**, **[mysqld-*major\_version*]** |
| [**mysqld\_safe**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#mysqld-safe) | **[mysqld]**, **[server]**, **[mysqld\_safe]** |
| [**mysql.server**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#mysql-server) | **[mysqld]**, **[mysql.server]**, **[server]** |

[mysqld-major\_version]意味着名称为[mysqld-5.7]和[mysqld-8.0]的组被具有5.7.x、8.0.x等版本的服务器读取。这个特性可以用来指定只能由给定版本系列内的服务器读取的选项。

为了向后兼容，mysql.server也读取[mysql\_server]组，mysqld\_safe也读取[safe\_mysqld]组。为了与时俱进，你应该更新你的选项文件，以使用[mysql.server]和[mysqld\_safe]组代替。

关于MySQL配置文件及其结构和内容的更多信息，见第4.2.2.2节 "使用选项文件"。

## 2.11 升级MySQL

本节描述了升级MySQL安装的步骤。

升级是一个常见的程序，因为你会在同一个MySQL版本系列中获得错误修复，或在主要的MySQL版本之间获得重要的功能。你首先在一些测试系统上执行这个程序，以确保一切工作顺利，然后再在生产系统上执行。

注意

在下面的讨论中，必须使用具有管理权限的MySQL账户运行的MySQL命令包括命令行上的-u root以指定MySQL根用户。需要root密码的命令也包括一个-p选项。因为-p后面没有选项值，这类命令会提示输入密码。在提示时输入密码并按回车键。

可以使用mysql命令行客户端执行SQL语句（以root身份连接，确保你有必要的权限）。

### 2.11.1 在你开始之前

在升级前查看本节中的信息。执行任何建议的操作。

* 了解升级过程中可能发生的情况。见第2.11.3节，"MySQL升级过程中的升级内容"。
* 通过创建一个备份来保护你的数据。该备份应包括mysql系统数据库，它包含MySQL数据字典表和系统表。参见第7.2节，"数据库备份方法"。

重要提示

不支持从MySQL 8.0降级到MySQL 5.7，或从MySQL 8.0版本降级到以前的MySQL 8.0版本。唯一支持的替代方法是恢复升级前的备份。因此，在开始升级过程之前，你必须备份你的数据。

查看第2.11.2节，"升级路径"，以确保你的预定升级路径得到支持。

* 查看第2.11.4节 "MySQL 8.0中的变化"，了解你在升级前应注意的变化。有些变化可能需要采取行动。
* 查看第1.3节 "MySQL 8.0中的新内容"，了解废弃和删除的功能。如果你使用这些功能，升级可能需要对其进行修改。
* 查看第1.4节，"MySQL 8.0中增加、废弃或删除的服务器和状态变量及选项"。如果你使用被废弃或删除的变量，升级可能需要改变配置。
* 查看发行说明，了解有关修复、更改和新功能的信息。
* 如果你使用复制，请查看第17.5.3节，"升级复制设置"。
* 升级程序因平台和初始安装的方式而异。使用适用于你当前MySQL安装的程序。
  + 对于非Windows平台上的二进制和基于包的安装，请参阅第2.11.6节，"升级Unix/Linux上的MySQL二进制或基于包的安装"。

注意

对于支持的Linux发行版，升级基于软件包的安装的首选方法是使用MySQL软件库（MySQL Yum Repository、MySQL APT Repository和MySQL SLES Repository）。

* 对于在企业Linux平台或Fedora上使用MySQL Yum Repository的安装，请参阅第2.11.7节 "使用MySQL Yum Repository升级MySQL"。
* 对于使用MySQL APT资源库在Ubuntu上的安装，请参阅第2.11.8节，"使用MySQL APT资源库升级MySQL"。
* 对于使用MySQL SLES资源库在SLES上的安装，请参阅第2.11.9节，"使用MySQL SLES资源库升级MySQL"。
* 对于使用Docker进行的安装，请参阅第2.11.11节，"升级MySQL的Docker安装"。
* 对于在Windows上的安装，请参阅第2.11.10节，"在Windows上升级MySQL"。
* 如果你的MySQL安装包含大量的数据，在原地升级后可能需要很长时间来转换，那么创建一个测试实例来评估需要的转换和执行这些转换所涉及的工作可能是有用的。要创建一个测试实例，制作一个包含mysql数据库和其他没有数据的数据库的MySQL实例的副本。在测试实例上运行升级程序，以评估执行实际数据转换所涉及的工作。
* 当你安装或升级到新版本的MySQL时，建议重建和重新安装MySQL语言接口。这适用于MySQL接口，如PHP mysql扩展和Perl DBD::mysql模块。

### 2.11.2 升级路径

* 支持从MySQL 5.7升级到8.0。然而，只支持在普遍可用（GA）版本之间进行升级。对于MySQL 8.0，要求你从MySQL 5.7 GA版本（5.7.9或更高）升级。不支持从MySQL 5.7的非GA版本进行升级。
* 在升级到下一个版本之前，建议先升级到最新的版本。例如，在升级到MySQL 8.0之前升级到最新的MySQL 5.7版本。不支持跳过版本的升级。例如，不支持直接从MySQL 5.6升级到8.0。
* 一旦一个系列的版本达到普遍可用（GA）状态，就支持在该系列中进行升级（从一个GA版本到另一个GA版本）。例如，支持从MySQL 8.0.x升级到8.0.y。(不支持涉及开发状态的非GA版本的升级。) 也支持跳过一个版本。例如，支持从MySQL 8.0.x升级到8.0.z。MySQL 8.0.11是MySQL 8.0版本系列中第一个GA状态的版本。

### 2.11.3 MySQL升级过程所升级的内容

安装新版本的MySQL可能需要升级现有安装中的这些部分。

* mysql系统模式，它包含存储MySQL服务器运行时所需信息的表（参见第5.3节 "mysql系统模式"）。 mysql模式表分为两大类。
  + 数据字典表，它存储数据库对象元数据。
  + 系统表（也就是剩下的非数据字典表），用于其他操作目的。
* 其他模式，其中一些是内置的，可以被认为是由服务器 "拥有 "的，还有一些不是。
  + 性能模式、INFORMATION\_SCHEMA、ndbinfo、和sys模式。
  + 用户模式。

两个不同的版本号与可能需要升级的安装部分相关。

* 数据字典版本。这适用于数据字典表。
* 服务器版本，也被称为MySQL版本。这适用于系统表和其他模式中的对象。

在这两种情况下，适用于现有MySQL安装的实际版本被存储在数据字典中，而当前的预期版本被编译到新版本的MySQL中。当实际版本低于当前的预期版本时，与该版本相关的那些安装部分必须升级到当前版本。如果两个版本都表明需要升级，那么数据字典的升级必须首先发生。

作为刚才提到的两个不同版本的反映，升级分两步进行。

* 第1步：数据字典升级。

这一步是升级。

* mysql模式中的数据字典表。如果实际的数据字典版本低于当前的预期版本，服务器会用更新的定义创建数据字典表，将持久化的元数据复制到新表中，用新表原子化地替换旧表，并重新初始化数据字典。
* 性能模式，INFORMATION\_SCHEMA，和ndbinfo。
* 步骤2：服务器升级。

这一步包括所有其他升级任务。如果现有的MySQL安装的服务器版本低于新安装的MySQL版本，其他一切都必须升级。

* mysql模式中的系统表（剩余的非数据字典表）。
* sys模式。
* 用户模式。

数据字典的升级（第1步）是服务器的责任，它在启动时根据需要执行这项任务，除非用一个选项来阻止它这样做。从MySQL 8.0.16开始，该选项为--upgrade=NONE，在MySQL 8.0.16之前为--no-dd-upgrade。

如果数据字典已经过期，但服务器被阻止升级，服务器就不会运行，而是以错误退出。例如。

[ERROR] [MY-013381] [Server] Server shutting down because upgrade is

required, yet prohibited by the command line option '--upgrade=NONE'.

[ERROR] [MY-010334] [Server] Failed to initialize DD Storage Engine

[ERROR] [MY-010020] [Server] Data Dictionary initialization failed.

第2步的责任在MySQL 8.0.16中发生了一些变化。

* 在MySQL 8.0.16之前，mysql\_upgrade升级了性能模式、INFORMATION\_SCHEMA和步骤2中描述的对象。预计DBA将在启动服务器后手动调用mysql\_upgrade。
* 从MySQL 8.0.16开始，服务器会执行以前由mysql\_upgrade处理的所有任务。尽管升级仍然是一个两步操作，但服务器同时执行这两步操作，从而使过程更简单。

根据你要升级的MySQL版本，就地升级和逻辑升级中的说明表明服务器是否执行所有升级任务，或者你是否必须在服务器启动后调用mysql\_upgrade。

注意

因为服务器从MySQL 8.0.16开始升级性能模式、INFORMATION\_SCHEMA和步骤2中描述的对象，所以mysql\_upgrade是不需要的，并且从该版本开始被废弃；预计它将在MySQL的未来版本中被删除。

在步骤2中发生的大多数方面在MySQL 8.0.16之前和之后都是一样的，尽管可能需要不同的命令选项来实现一个特定的效果。

从MySQL 8.0.16开始，-升级服务器选项控制服务器是否以及如何在启动时执行自动升级。

* 在没有选项或--upgrade=AUTO的情况下，服务器会升级它确定为过时的任何东西（步骤1和2）。
* 使用 --upgrade=NONE，服务器不升级（跳过步骤 1 和 2），但如果数据字典必须升级，也会以错误退出。服务器不可能在一个过时的数据字典中运行；服务器坚持要么升级，要么退出。
* 如果使用 --upgrade=MINIMAL，服务器会升级数据字典、性能模式和INFORMATION\_SCHEMA，如果需要的话（步骤1）。请注意，在使用这个选项进行升级后，组复制不能被启动，因为复制内部所依赖的系统表没有被更新，而且在其他领域可能也会出现功能下降的情况。
* 使用--upgrade=FORCE，服务器会升级数据字典、性能模式和INFORMATION\_SCHEMA（如果需要的话）（步骤1），并强制升级其他所有东西（步骤2）。由于服务器检查所有模式中的所有对象，预计使用这个选项时服务器启动的时间会更长。

如果服务器认为没有必要的话，FORCE对于强制执行步骤2的操作是非常有用的。FORCE与AUTO的不同之处在于，使用FORCE时，服务器会重新创建系统表，例如帮助表或时区表，如果它们丢失的话。

下面的列表显示了MySQL 8.0.16之前的升级命令和MySQL 8.0.16及以上版本的等效命令。

* 执行正常的升级（必要时步骤1和2）。
  + 在MySQL 8.0.16之前：mysqld然后是mysql\_upgrade
  + 从MySQL 8.0.16开始：mysqld
* 根据需要只执行步骤1。
  + 在MySQL 8.0.16之前：不可能执行步骤1中描述的所有升级任务而排除步骤2中描述的任务。然而，你可以使用mysqld和mysql\_upgrade与-upgrade-system-tables和-skip-sys-schema选项避免升级用户模式和系统模式。
  + 从MySQL 8.0.16开始： mysqld --upgrade=MINIMAL
* 根据需要执行步骤1，并强制执行步骤2。
  + 在MySQL 8.0.16之前：mysqld接着mysql\_upgrade –force
  + 从MySQL 8.0.16开始：mysqld --upgrade=FORCE

在MySQL 8.0.16之前，某些mysql\_upgrade选项影响它所执行的操作。下表显示了从MySQL 8.0.16开始要使用哪些服务器-升级选项值来实现类似的效果。这些不一定是完全等同的，因为一个给定的-升级选项值可能有额外的效果）。

mysql\_upgrade选项服务器选项

|  |  |
| --- | --- |
| **mysql\_upgrade Option** | **Server Option** |
| [--skip-sys-schema](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#option_mysql_upgrade_skip-sys-schema) | [--upgrade=NONE](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#option_mysqld_upgrade) or [--upgrade=MINIMAL](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#option_mysqld_upgrade) |
| [--upgrade-system-tables](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#option_mysql_upgrade_upgrade-system-tables) | [--upgrade=NONE](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#option_mysqld_upgrade) or [--upgrade=MINIMAL](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#option_mysqld_upgrade) |
| [--force](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#option_mysql_upgrade_force) | [--upgrade=FORCE](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#option_mysqld_upgrade) |

关于升级步骤2中发生的事情的附加说明。

* 如果系统模式没有安装，第2步将其安装，否则将其升级到当前版本。如果一个sys模式存在但没有版本视图，就会发生错误，因为假设没有这个模式就表示是用户创建的模式。
* A sys schema exists with no sys.version view. If
* you have a user created sys schema, this must be renamed for the
* upgrade to succeed.

要在这种情况下升级，首先要删除或重命名现有的系统模式。然后再次执行升级程序。可能有必要强制执行第2步）。

为了防止系统模式的检查。

* 从MySQL 8.0.16开始：用--upgrade=NONE或--upgrade=MINIMAL选项启动服务器。
* 在MySQL 8.0.16之前：用--skip-sys-schema选项调用mysql\_upgrade。
* 第2步根据需要处理所有用户模式中的所有表。表的检查可能需要很长的时间才能完成。每个表都被锁定，因此在处理时对其他会话不可用。检查和修复操作可能很耗时，特别是对于大表来说。表检查使用CHECK TABLE语句的FOR UPGRADE选项。关于这个选项的详细内容，请参见第13.7.3.2节 "CHECK TABLE语句"。

要防止表检查。

* 从MySQL 8.0.16开始：用 --upgrade=NONE或 --upgrade=MINIMAL选项启动服务器。
* 在MySQL 8.0.16之前：用-upgrade-system-tables选项调用mysql\_upgrade。

要强制检查表。

* 从MySQL 8.0.16开始：用-upgrade=FORCE选项启动服务器。
* 在MySQL 8.0.16之前：用--force选项调用mysql\_upgrade。
* 第2步将MySQL版本号保存在数据目录中一个名为mysql\_upgrade\_info的文件中。

要忽略mysql\_upgrade\_info文件，无论如何都要执行检查。

* 从MySQL 8.0.16开始：用-upgrade=FORCE选项启动服务器。
* 在MySQL 8.0.16之前：用--force选项调用mysql\_upgrade。

注意

mysql\_upgrade\_info文件已被废弃；预计它将在MySQL的未来版本中被删除。

* 第2步用当前的MySQL版本号标记所有检查和修复的表。这确保下次升级检查发生在相同版本的服务器上时，可以确定是否有必要再次检查或修复一个给定的表。
* 第2步升级系统表，确保它们具有当前的结构。无论是服务器还是mysql\_upgrade执行该步骤，都是如此。关于帮助表和时区表的内容，mysql\_upgrade不加载这两类表，而服务器加载帮助表，但不加载时区表。(也就是说，在MySQL 8.0.16之前，服务器只在数据目录初始化时加载帮助表。从MySQL 8.0.16开始，它在初始化和升级时加载帮助表）。) 加载时区表的程序与平台有关，需要DBA作出决定，所以不能自动完成。

### 2.11.4 MySQL 8.0的变化

在升级到MySQL 8.0之前，请审查本节中描述的变化，以确定那些适用于你当前的MySQL安装和应用的变化。执行任何建议的操作。

标记为不兼容的变化是与早期版本的MySQL不兼容的变化，可能需要你在升级前注意。我们的目的是避免这些变化，但偶尔也有必要纠正比版本间不兼容更糟糕的问题。如果适用于你的安装的升级问题涉及不兼容，请遵循描述中给出的指示。

* 数据字典的变化
* caching\_sha2\_password作为首选的认证插件
* 配置变更
* 服务器变化
* InnoDB变化
* SQL的变化
* 更改了服务器默认值

**数据字典的变化**

MySQL服务器8.0纳入了一个全局数据字典，其中包含关于事务表中的数据库对象的信息。在以前的MySQL系列中，字典数据存储在元数据文件和非事务性系统表中。因此，升级程序要求你通过检查特定的先决条件来验证你的安装是否准备好升级。欲了解更多信息，请参见第2.11.5节，"为升级准备您的安装"。启用了数据字典的服务器需要一些一般的操作差异；见第14.7节，"数据字典使用差异"。

**caching\_sha2\_password作为首选的认证插件**

caching\_sha2\_password和sha256\_password认证插件比mysql\_native\_password插件提供更安全的密码加密，而且caching\_sha2\_password比sha256\_password提供更好的性能。由于caching\_sha2\_password的这些优越的安全和性能特性，从MySQL 8.0开始，它是首选的认证插件，也是默认的认证插件，而不是mysql\_native\_password。这一变化同时影响到服务器和libmysqlclient客户端库。

* 对于服务器，default\_authentication\_plugin系统变量的默认值从mysql\_native\_password变为caching\_sha2\_password。

这一变化仅适用于安装或升级到MySQL 8.0或更高版本后创建的新账户。对于在升级后的安装中已经存在的账户，其认证插件保持不变。希望切换到caching\_sha2\_password的现有用户可以使用ALTER USER语句来实现。

ALTER USER ***user***

IDENTIFIED WITH caching\_sha2\_password

BY '***password***';

* libmysqlclient库将caching\_sha2\_password作为默认认证插件，而不是mysql\_native\_password。

下面几节将讨论caching\_sha2\_password更突出的作用所带来的影响。

caching\_sha2\_password 兼容性问题和解决方案

* caching\_sha2\_password-兼容的客户端和连接器
* caching\_sha2\_password和根管理账户
* caching\_sha2\_password与复制
* caching\_sha2\_password的兼容性问题和解决方案

重要提示

如果你的MySQL安装必须为8.0以前的客户提供服务，并且在升级到MySQL 8.0或更高版本后遇到了兼容性问题，解决这些问题并恢复8.0以前的兼容性的最简单方法是重新配置服务器以恢复到以前的默认认证插件（mysql\_native\_password）。例如，在服务器选项文件中使用这些行。

[mysqld]

default\_authentication\_plugin=mysql\_native\_password

该设置使8.0前的客户端能够连接到8.0服务器，直到你安装的客户端和连接器升级到知道caching\_sha2\_password。然而，这个设置应该被看作是暂时的，而不是长期或永久的解决方案，因为它导致在设置生效的情况下创建的新账户放弃了由caching\_sha2\_password提供的改进的认证安全性。

使用caching\_sha2\_password比mysql\_native\_password提供更安全的密码散列（以及随之而来的改进的客户端连接认证）。然而，它也有兼容性问题，可能会影响现有的MySQL安装。

* 尚未更新以了解caching\_sha2\_password的客户端和连接器可能难以连接到配置有caching\_sha2\_password作为默认认证插件的MySQL 8.0服务器，甚至无法使用不使用caching\_sha2\_password进行认证的账户。这个问题的发生是因为服务器向客户指定了其默认认证插件的名称。如果客户机或连接器是基于客户机/服务器协议实现的，而该协议不能优雅地处理未被识别的默认认证插件，那么它可能会出现类似以下的错误。

Authentication plugin 'caching\_sha2\_password' is not supported

Authentication plugin 'caching\_sha2\_password' cannot be loaded:

dlopen(/usr/local/mysql/lib/plugin/caching\_sha2\_password.so, 2):

image not found

Warning: mysqli\_connect(): The server requested authentication

method unknown to the client [caching\_sha2\_password]

关于编写连接器以优雅地处理来自服务器的未知默认认证插件的请求的信息，请参阅认证插件连接器-编写注意事项。

* 使用用caching\_sha2\_password认证的账户的客户必须使用安全连接（使用TLS/SSL凭证的TCP、Unix套接字文件或共享内存进行），或支持使用RSA密钥对交换密码的未加密连接。这一安全要求不适用于mysql\_native\_passsword，所以切换到caching\_sha2\_password可能需要额外的配置（见第6.4.1.2节，"缓存SHA-2可插拔认证"）。然而，MySQL 8.0中的客户端连接默认喜欢使用TLS/SSL，所以已经符合该偏好的客户端可能不需要额外的配置。
* 没有被更新以了解caching\_sha2\_password的客户端和连接器不能连接到用caching\_sha2\_password认证的账户，因为它们不承认这个插件是有效的。这是客户端/服务器认证插件兼容性要求的一个特殊例子，在认证插件客户端/服务器兼容性中讨论过）。要解决这个问题，请将客户端与MySQL 8.0或更高版本的libmysqlclient重新链接，或获得一个更新的连接器，以识别caching\_sha2\_password。
* 因为caching\_sha2\_password现在也是libmysqlclient客户端库中的默认认证插件，对于从MySQL 8.0客户端连接到使用mysql\_native\_password（以前的默认认证插件）的账户，认证需要在客户端/服务器协议中进行额外的往返，除非客户端程序被调用时带有-default-auth=mysql\_native\_password选项。

8.0之前的MySQL版本的libmysqlclient客户端库能够连接到MySQL 8.0服务器（使用caching\_sha2\_password认证的账户除外）。这意味着基于libmysqlclient的8.0前客户端也应该能够连接。例子。

* 标准的MySQL客户端，如mysql和mysqladmin是基于libmysqlclient的。
* Perl DBI的DBD::mysql驱动是基于libmysqlclient的。
* MySQL Connector/Python有一个C扩展模块，是基于libmysqlclient的。要使用它，在连接时包括use\_pure=False选项。

当现有的MySQL 8.0安装升级到MySQL 8.0.4或更高版本时，一些基于libmysqlclient的旧客户端可能会 "自动 "升级，如果它们是动态链接的，因为它们使用升级后安装的新客户端库。例如，如果Perl DBI的DBD::mysql驱动使用动态链接，在升级到MySQL 8.0.4或更高版本后，它可以在原地使用libmysqlclient，结果是这样的。

* 在升级之前，使用DBD::mysql的DBI脚本可以连接到MySQL 8.0服务器，除了用caching\_sha2\_password认证的账户。
* 在升级后，同样的脚本也变得能够使用caching\_sha2\_password账户。

然而，出现前面的结果是因为8.0.4之前的MySQL 8.0安装的libmysqlclient实例是二进制兼容的。它们都使用一个共享库的主要版本号为21。对于从MySQL 5.7或更早的版本链接到libmysqlclient的客户端，它们链接到一个具有不同版本号的共享库，该共享库不具有二进制兼容。在这种情况下，客户端必须根据8.0.4或更高版本的libmysqlclient重新编译，以便与MySQL 8.0服务器和缓存\_sha2\_password账户完全兼容。

MySQL Connector/J 5.1至8.0.8能够连接到MySQL 8.0服务器，但使用caching\_sha2\_password验证的账户除外。(Connector/J 8.0.9或更高版本需要连接到caching\_sha2\_password账户。)

使用libmysqlclient以外的客户/服务器协议实现的客户端可能需要升级到能理解新认证插件的较新版本。例如，在PHP中，MySQL的连接通常基于mysqlnd，目前它不知道caching\_sha2\_password。在mysqlnd的更新版本出现之前，使PHP客户端能够连接到MySQL 8.0的方法是重新配置服务器，恢复到mysql\_native\_password作为默认的认证插件，如之前讨论的那样。

如果客户端或连接器支持明确指定默认认证插件的选项，请使用它来命名一个除caching\_sha2\_password之外的插件。例子。

* 一些MySQL客户端支持一个--default-auth选项。(标准的MySQL客户端，如mysql和mysqladmin支持这个选项，但没有它也能成功连接到8.0服务器。然而，其他客户端可能支持一个类似的选项。如果是这样的话，值得一试）。)
* 使用libmysqlclient C API的程序可以用MYSQL\_DEFAULT\_AUTH选项调用mysql\_options()函数。
* 使用客户端/服务器协议的本地Python实现的MySQL Connector/Python脚本可以指定auth\_plugin连接选项。(或者，使用Connector/Python C扩展，它能够连接到MySQL 8.0服务器而不需要auth\_plugin。)

caching\_sha2\_password-兼容的客户端和连接器

如果有一个客户端或连接器已经更新到知道caching\_sha2\_password，那么在连接到配置有caching\_sha2\_password作为默认认证插件的MySQL 8.0服务器时，使用它是确保兼容性的最好方法。

这些客户端和连接器已经升级，以支持caching\_sha2\_password。

* MySQL 8.0（8.0.4或更高）中的libmysqlclient客户端库。标准的MySQL客户端，如mysql和mysqladmin是基于libmysqlclient的，所以它们也是兼容的。
* MySQL 5.7（5.7.23或更高）中的libmysqlclient客户端库。标准的MySQL客户端，如mysql和mysqladmin是基于libmysqlclient的，所以它们也是兼容的。
* MySQL Connector/C++ 1.1.11或更高版本或8.0.7或更高版本。
* MySQL Connector/J 8.0.9或更高版本。
* MySQL Connector/NET 8.0.10或更高版本（通过经典的MySQL协议）。
* MySQL Connector/Node.js 8.0.9或更高。
* PHP：X DevAPI PHP扩展（mysql\_xdevapi）支持caching\_sha2\_password。
* PHP：PDO\_MySQL和ext/mysqli扩展不支持caching\_sha2\_password。此外，当与 7.1.16 之前的 PHP 版本和 7.2.4 之前的 PHP 7.2 一起使用时，即使没有使用 caching\_sha2\_password，它们也无法与 default\_authentication\_plugin=caching\_sha2\_password 连接。

caching\_sha2\_password和根管理账户

对于升级到MySQL 8.0，现有账户的认证插件保持不变，包括'root'@'localhost'管理账户的插件。

对于新的MySQL 8.0安装，当你初始化数据目录时（使用第2.10.1节 "初始化数据目录 "的说明），'root'@'localhost'账户被创建，并且该账户默认使用caching\_sha2\_password。因此，要在数据目录初始化后连接到服务器，你必须使用一个支持caching\_sha2\_password的客户端或连接器。如果你能做到这一点，但更希望根账户在安装后使用mysql\_native\_password，请安装MySQL并像通常那样初始化数据目录。然后以root身份连接到服务器，使用如下的ALTER USER来改变账户认证插件和密码。

ALTER USER 'root'@'localhost'

IDENTIFIED WITH mysql\_native\_password

BY '***password***';

如果你使用的客户端或连接器还不支持缓存\_sha2\_password，你可以使用一个修改过的数据目录初始化过程，在创建账户时就将根账户与mysql\_native\_password关联起来。要做到这一点，请使用这些技术中的任何一种。

* 与--initialize或--initialize-insecure一起提供一个--default-authentication-plugin=mysql\_native\_password选项。
* 在一个选项文件中把default\_authentication\_plugin设置为mysql\_native\_password，并使用-defaults-file选项与-initialize或-initialize-insecure一起命名该选项文件。(在这种情况下，如果你在随后的服务器启动中继续使用该选项文件，新账户将用mysql\_native\_password而不是caching\_sha2\_password创建，除非你从选项文件中删除default\_authentication\_plugin设置。)

caching\_sha2\_password和复制

在所有服务器都已升级到MySQL 8.0.4或更高版本的复制方案中，到源服务器的复制连接可以使用用caching\_sha2\_password认证的账户。对于这样的连接，与使用用caching\_sha2\_password认证的账户的其他客户端的要求相同：使用安全连接或基于RSA的密码交换。

要连接到caching\_sha2\_password账户进行源/副本复制。

* 使用下列任何一个 CHANGE MASTER TO 选项。

MASTER\_SSL = 1

GET\_MASTER\_PUBLIC\_KEY = 1

MASTER\_PUBLIC\_KEY\_PATH='***path to RSA public key file***'

* 另外， 如果在服务器启动时提供了所需的密钥， 您可以使用 RSA 公钥相关的选项。

要连接到组复制的caching\_sha2\_password账户。

* 对于使用OpenSSL构建的MySQL，设置以下任何一个系统变量。

SET GLOBAL group\_replication\_recovery\_use\_ssl = ON;

SET GLOBAL group\_replication\_recovery\_get\_public\_key = 1;

SET GLOBAL group\_replication\_recovery\_public\_key\_path = '***path to RSA public key file***';

* 另外，如果在服务器启动时提供了所需的密钥，你可以使用RSA公钥相关选项。

**配置变更**

* 不兼容的变化。一个MySQL存储引擎现在负责提供它自己的分区处理程序，MySQL服务器不再提供通用的分区支持。InnoDB和NDB是唯一提供本地分区处理程序的存储引擎，在MySQL 8.0中得到支持。使用任何其他存储引擎的分区表必须在升级服务器之前进行更改--要么将其转换为InnoDB或NDB，要么移除其分区，否则之后就无法使用。

关于将MyISAM表转换为InnoDB的信息，参见章节15.6.1.5，"将表从MyISAM转换为InnoDB"。

在MySQL 8.0中，使用没有这种支持的存储引擎的表创建语句会导致分区表的失败，并出现错误（ER\_CHECK\_NOT\_IMPLEMENTED）。如果你使用mysqldump从MySQL 5.7（或更早）创建的转储文件中导入数据库到MySQL 8.0服务器，你必须确保任何创建分区表的语句不会指定一个不支持的存储引擎，要么删除任何对分区的引用，要么指定存储引擎为InnoDB或允许它被默认设置为InnoDB。

注意

第2.11.5节 "为升级准备你的安装 "给出的程序描述了如何在升级到MySQL 8.0之前识别必须改变的分区表。

有关进一步信息，请参见第24.6.2节 "与存储引擎有关的分区限制"。

* 不兼容的改变。一些服务器错误代码不再使用，并已被删除（关于列表，请参见MySQL 8.0中删除的功能）。对其中任何一个进行专门测试的应用程序都应该被更新。
* 重要变化。默认字符集已从latin1改为utf8mb4。这些系统变量受到影响。
  + character\_set\_server和character\_set\_database系统变量的默认值已经从latin1变为utf8mb4。
  + collation\_server和collation\_database系统变量的默认值已经从latin1\_swedish\_ci变为utf8mb4\_0900\_ai\_ci。

因此，新对象的默认字符集和排序与以前不同，除非明确指定字符集和排序。这包括数据库和其中的对象，如表、视图和存储程序。假设使用了以前的默认值，保留它们的一个方法是在my.cnf文件中用这些行启动服务器。

[mysqld]

character\_set\_server=latin1

collation\_server=latin1\_swedish\_ci

在复制的设置中，当从MySQL 5.7升级到8.0时，建议在升级前将默认字符集改回MySQL 5.7中使用的字符集。升级完成后，可将默认字符集改为utf8mb4。

* 不兼容的改变。从MySQL 8.0.11开始，禁止以不同于服务器初始化时使用的设置的 lower\_case\_table\_names设置启动服务器。这个限制是必要的，因为各种数据字典表字段使用的排序是基于服务器初始化时定义的 lower\_case\_table\_names设置，用不同的设置重新启动服务器会在标识符的排序和比较方面引入不一致。

**服务器变化**

* 在MySQL 8.0.11中，与账户管理有关的几个废弃功能已被删除，例如使用GRANT语句来修改用户账户的非特权特性、NO\_AUTO\_CREATE\_USER SQL模式、PASSWORD()函数和old\_passwords系统变量。

从MySQL 5.7到8.0复制引用这些已删除特性的语句可能导致复制失败。应修订使用任何已删除功能的应用程序，以避免这些功能，并尽可能使用替代品，如MySQL 8.0中删除的功能所述。

为了避免在MySQL 8.0上出现启动失败，请从MySQL选项文件中的sql\_mode系统变量设置中删除NO\_AUTO\_CREATE\_USER的任何实例。

将存储程序定义中包括NO\_AUTO\_CREATE\_USER SQL模式的转储文件加载到MySQL 8.0服务器会导致失败。从MySQL 5.7.24和MySQL 8.0.13开始，mysqldump从存储程序定义中删除了NO\_AUTO\_CREATE\_USER。用早期版本的mysqldump创建的转储文件必须手动修改，以删除NO\_AUTO\_CREATE\_USER的实例。

* 在MySQL 8.0.11中，这些废弃的兼容SQL模式被删除。DB2、maxdb、mssql、mysql323、mysql40、oracle、postgresql、no\_field\_options、no\_key\_options和no\_table\_options。它们不能再被分配给sql\_mode系统变量或作为mysqldump --兼容选项的允许值。

移除MAXDB意味着CREATE TABLE或ALTER TABLE的TIMESTAMP数据类型不再被视为DATETIME。

从MySQL 5.7到8.0复制引用已删除的SQL模式的语句可能导致复制失败。这包括复制存储程序（存储过程和函数、触发器和事件）的CREATE语句，这些语句在当前的sql\_mode值包括任何被移除的模式时被执行。使用任何已删除模式的应用程序应进行修改以避免它们。

* 从MySQL 8.0.3开始，空间数据类型允许一个SRID属性，以明确指示存储在该列中的值的空间参考系统（SRS）。见第11.4.1节，"空间数据类型"。

一个具有明确的SRID属性的空间列是受SRID限制的。该列只接受具有该ID的值，并且该列上的SPATIAL索引将被优化器使用。优化器忽略了没有SRID属性的空间列上的SPATIAL索引。参见第8.3.3节 "SPATIAL索引的优化"。如果你想让优化器考虑没有SRID限制的空间列上的SPATIAL索引，应该对每个这样的列进行修改。

* + 验证该列内的所有值是否具有相同的SRID。为了确定几何列col\_name中包含的SRID，使用下面的查询。

SELECT DISTINCT ST\_SRID(***col\_name***) FROM ***tbl\_name***;

如果查询返回多条记录，说明该列包含混合的SRID。在这种情况下，修改其内容，使所有的值都具有相同的SRID。

* 重新定义该列，使其具有一个明确的SRID属性。
* 重新创建SPATIAL索引。
* 在MySQL 8.0.0中删除了几个空间函数，原因是空间函数命名空间的变化，对执行精确操作的函数实施ST\_前缀，或对基于最小边界矩形执行操作的函数实施MBR前缀。在生成的列定义中使用已删除的空间函数可能导致升级失败。在升级之前，运行mysqlcheck --check-upgrade以查找已删除的空间函数，并用它们的ST\_或MBR命名的替代品来替换任何发现的空间函数。关于已删除的空间函数的列表，请参阅《MySQL 8.0中删除的功能》。
* 当执行原地升级到MySQL 8.0.3或更高版本时，BACKUP\_ADMIN权限会自动授予具有RELOAD权限的用户。
* 从MySQL 8.0.13开始，由于基于行或混合复制模式与基于语句的复制模式在处理临时表的方式上存在差异，在运行时切换二进制日志格式方面有新的限制。
  + 如果会话有任何开放的临时表，SET @@SESSION.binlog\_format不能被使用。
  + 如果任何复制通道有任何开放的临时表，SET @@global.binlog\_format和SET @@persist.binlog\_format不能被使用。如果复制通道有开放的临时表，SET @@persist\_only.binlog\_format是允许的，因为与PERSIST不同，PERSIST\_ONLY不会修改运行时的全局系统变量值。
  + 如果任何复制通道应用者正在运行，SET @@global.binlog\_format和SET @@persist.binlog\_format不能被使用。这是因为只有当复制通道的应用者重新启动时，这种改变才会在复制通道上生效，这时复制通道可能有开放的临时表。这种行为比以前更有限制性。如果任何复制通道的应用者正在运行，则允许设置 @@persist\_only.binlog\_format。

**InnoDB变化**

* 基于InnoDB系统表的INFORMATION\_SCHEMA视图被数据字典表的内部系统视图取代。受影响的InnoDB INFORMATION\_SCHEMA视图被重新命名。
* **Table 2.16 Renamed InnoDB Information Schema Views**

|  |  |
| --- | --- |
| **Old Name** | **New Name** |
| **INNODB\_SYS\_COLUMNS** | **INNODB\_COLUMNS** |
| **INNODB\_SYS\_DATAFILES** | **INNODB\_DATAFILES** |
| **INNODB\_SYS\_FIELDS** | **INNODB\_FIELDS** |
| **INNODB\_SYS\_FOREIGN** | **INNODB\_FOREIGN** |
| **INNODB\_SYS\_FOREIGN\_COLS** | **INNODB\_FOREIGN\_COLS** |
| **INNODB\_SYS\_INDEXES** | **INNODB\_INDEXES** |
| **INNODB\_SYS\_TABLES** | **INNODB\_TABLES** |
| **INNODB\_SYS\_TABLESPACES** | **INNODB\_TABLESPACES** |
| **INNODB\_SYS\_TABLESTATS** | **INNODB\_TABLESTATS** |
| **INNODB\_SYS\_VIRTUAL** | **INNODB\_VIRTUAL** |

在升级到MySQL 8.0.3或更高版本后，更新任何引用先前InnoDB INFORMATION\_SCHEMA视图名称的脚本。

* 与MySQL捆绑的zlib库版本从1.2.3版提高到1.2.11版。

zlib 1.2.11中的zlib compressBound()函数，与zlib 1.2.3版本相比，返回压缩给定长度的字节所需的缓冲区大小的估计值略高。compressBound()函数被InnoDB函数调用，这些函数在创建压缩的InnoDB表或在压缩的InnoDB表中插入和更新行时确定允许的最大行大小。因此，CREATE TABLE ... ROW\_FORMAT=COMPRESSED、INSERT和UPDATE操作的行数非常接近最大行数，在早期版本中是成功的，现在可能会失败。为了避免这个问题，在升级之前，在MySQL 8.0测试实例上测试具有大行的压缩InnoDB表的CREATE TABLE语句。

* 随着-innodb-directories功能的引入，用绝对路径或在数据目录之外的位置创建的每表文件和一般表空间文件的位置应该被添加到innodb\_directories参数值中。否则，InnoDB在恢复过程中就无法定位这些文件。要查看表空间文件的位置，查询INFORMATION\_SCHEMA.FILES表。

SELECT TABLESPACE\_NAME, FILE\_NAME FROM INFORMATION\_SCHEMA.FILES \G

* Undo logs不能再驻留在系统表空间中。在MySQL 8.0中，Undo logs默认驻留在两个undo tablespaces中。更多信息，见第15.6.3.4节 "undo tablespaces"。

当从MySQL 5.7升级到MySQL 8.0时，存在于MySQL 5.7实例中的任何undo tablespaces被删除，并由两个新的默认undo tablespaces取代。默认的undo tablespaces是在innodb\_undo\_directory变量定义的位置创建的。如果innodb\_undo\_directory变量未定义，则undo tablespaces将在数据目录中创建。从MySQL 5.7升级到MySQL 8.0需要缓慢关机，以确保MySQL 5.7实例中的undo tablespaces是空的，从而允许它们被安全地删除。

当从早期的MySQL 8.0版本升级到MySQL 8.0.14或更高版本时，由于innodb\_undo\_tablespaces设置大于2而存在于升级前实例中的undo tablespaces被视为用户定义的undo tablespaces，在升级后可以分别使用ALTER UNDO TABLESPACE和DROP UNDO TABLESPACE语法停用和删除。在MySQL 8.0版本系列内的升级可能并不总是需要缓慢关闭，这意味着现有的undo tablespaces可能包含Undo logs。因此，现有的undo tablespaces不会被升级过程删除。

* 不兼容的变化。从MySQL 8.0.17开始，CREATE TABLESPACE ... ADD DATAFILE子句不允许循环目录引用。例如，以下语句中的循环目录引用（/.../）是不允许的。

CREATE TABLESPACE ts1 ADD DATAFILE ts1.ibd '***any\_directory***/../ts1.ibd';

这个限制在Linux上有一个例外，如果前面的目录是一个符号链接，那么循环目录引用是允许的。例如，如果any\_directory是一个符号链接，上面例子中的数据文件路径是允许的。(仍然允许数据文件路径以'.../'开头)。

为了避免升级问题，在升级到MySQL 8.0.17或更高版本之前，从表空间数据文件路径中删除任何循环目录引用。要检查表空间路径，请查询INFORMATION\_SCHEMA.INNODB\_DATAFILES表。

* 由于在MySQL 8.0.14中引入的回归，对于具有分区表和 lower\_case\_table\_names=1的实例，在对大小写敏感的文件系统上从MySQL 5.7或MySQL 8.0.14之前的版本升级到MySQL 8.0.16失败。该故障是由与分区表文件名有关的大小写不匹配问题引起的。引入回归的修复被恢复，这允许从MySQL 5.7或MySQL 8.0 14之前的版本升级到MySQL 8.0.17，以正常运作。然而，在MySQL 8.0.14、8.0.15和8.0.16版本中仍然存在这种退步。

在大小写敏感的文件系统上从MySQL 8.0.14、8.0.15或8.0.16就地升级到MySQL 8.0.17，如果存在分区表且 lower\_case\_table\_names=1，则在将二进制文件或软件包升级到MySQL 8.0.17后启动服务器时出现下列错误。

Upgrading from server version ***version\_number*** with

partitioned tables and lower\_case\_table\_names == 1 on a case sensitive file

system may cause issues, and is therefore prohibited. To upgrade anyway, restart

the new server version with the command line option 'upgrade=FORCE'. When

upgrade is completed, please execute 'RENAME TABLE ***part\_table\_name***

TO ***new\_table\_name***; RENAME TABLE ***new\_table\_name***

TO ***part\_table\_name***;' for each of the partitioned tables.

Please see the documentation for further information.

如果你在升级到MySQL 8.0.17时遇到这个错误，请执行以下解决方法。

1. 用-upgrade=force重新启动服务器，以强制进行升级操作。
2. 用小写的分区名分隔符（#p#或#sp#）来识别分区表的文件名。
3. mysql> SELECT FILE\_NAME FROM INFORMATION\_SCHEMA.FILES WHERE FILE\_NAME LIKE '%#p#%' OR FILE\_NAME LIKE '%#sp#%';
4. 对于确定的每个文件，使用一个临时名称重命名相关的表，然后将表重命名为其原始名称。
5. mysql> RENAME TABLE ***table\_name*** TO ***temporary\_table\_name***;
6. mysql> RENAME TABLE ***temporary\_table\_name*** TO ***table\_name***;
7. 验证没有分区表文件名小写的分区名分隔符（应该返回一个空的结果集）。
8. mysql> **SELECT FILE\_NAME FROM INFORMATION\_SCHEMA.FILES WHERE FILE\_NAME LIKE '%#p#%' OR FILE\_NAME LIKE '%#sp#%';**
9. Empty set (0.00 sec)

e. 在每个重命名的表上运行ANALYZE TABLE，以更新mysql.innodb\_index\_stats和mysql.innodb\_table\_stats表中的优化器统计数据。

由于在MySQL 8.0.14、8.0.15和8.0.16版本中仍然存在回归，在 lower\_case\_table\_names=1的大小写敏感文件系统中，不支持从MySQL 8.0.14、8.0.15或8.0.16导入分区表至MySQL 8.0.17。试图这样做的结果是 "表空间缺少表 "的错误。

* MySQL在构建表空间名称和表分区的文件名称时使用分隔符串。分区名称前有一个 "#p#"分隔符串，子分区名称前有一个 "#sp#"分隔符串，如所示。

***schema\_name***.***table\_name***#p#***partition\_name***#sp#***subpartition\_name***

***table\_name***#p#***partition\_name***#sp#***subpartition\_name***.ibd

历史上，分隔符字符串在对大小写敏感的文件系统（如Linux）上是大写的（#P#和#SP#），而在对大小写不敏感的文件系统（如Windows）上是小写的（#p#和#sp#）。从MySQL 8.0.19开始，定界符串在所有文件系统上都是小写。这一变化防止了在大小写敏感和不敏感的文件系统之间迁移数据目录时的问题。不再使用大写的定界符串。

此外，基于用户指定的分区或子分区名称生成的分区表空间名称和文件名称，可以指定为大写或小写，现在以小写生成（并在内部存储），而不管 lower\_case\_table\_names 设置如何，以确保大小写不敏感。例如，如果以PART\_1的名称创建了一个表分区，那么表空间名称和文件名称将以小写生成。

***schema\_name***.***table\_name***#p#***part\_1***

***table\_name***#p#***part\_1***.ibd

在升级期间，MySQL检查并在必要时进行修改。

* + 磁盘上和数据字典中的分区文件名，以确保小写定界符和分区名称。
  + 数据字典中的分区元数据，以解决以前的错误修复所带来的相关问题。
  + InnoDB统计数据，用于解决以前的错误修复所带来的相关问题。

在表空间导入操作期间，将检查磁盘上的分区表空间文件名，并在必要时进行修改，以确保小写定界符和分区名称。

* 从MySQL 8.0.21开始，在启动时或从MySQL 5.7升级时，如果发现表空间数据文件驻留在未知目录中，将向错误日志写入警告。已知目录是那些由datadir、innodb\_data\_home\_dir和innodb\_directories变量定义的目录。要使一个目录为人所知，就把它添加到innodb\_directories设置中。让目录为人所知可以确保在恢复过程中可以找到数据文件。欲了解更多信息，请参见崩溃恢复期间的表空间发现。

**SQL变化**

* 不兼容的变化。从MySQL 8.0.13开始，GROUP BY子句中被废弃的ASC或DESC限定符已被删除。以前依赖GROUP BY排序的查询可能产生与以前的MySQL版本不同的结果。要产生一个给定的排序顺序，请提供一个ORDER BY子句。

来自MySQL 8.0.12或更低版本的查询和存储程序定义，如果对GROUP BY子句使用ASC或DESC限定词，则应进行修改。否则，升级到MySQL 8.0.13或更高版本可能会失败，复制到MySQL 8.0.13或更高版本的复制服务器也可能失败。

在MySQL 8.0中可能保留了一些在MySQL 5.7中没有保留的关键字。见第9.3节，"关键词和保留字"。这可能导致以前用作标识符的词变得非法。要修复受影响的语句，请使用标识符引号。参见第9.2节，"模式对象名称"。

在升级后，建议你测试应用程序代码中指定的优化器提示，以确保这些提示仍然是实现所需优化策略的必要条件。优化器的增强有时会使某些优化器提示成为不必要的。在某些情况下，不必要的优化器提示甚至可能产生反作用。

* 不兼容的变化。在MySQL 5.7中，为InnoDB表指定一个没有CONSTRAINT符号子的FOREIGN KEY定义，或指定没有符号的CONSTRAINT关键字，导致InnoDB使用一个生成的约束名称。这种行为在MySQL 8.0中有所改变，InnoDB使用FOREIGN KEY index\_name值而不是生成的名称。因为约束名称必须在每个模式（数据库）中是唯一的，这种变化导致了由于外键索引名称在每个模式中不唯一的错误。为了避免这种错误，新的约束命名行为已经在MySQL 8.0.16中被恢复了，InnoDB再次使用生成的约束名称。

为了与InnoDB保持一致，基于MySQL 8.0.16或更高版本的NDB版本在没有指定CONSTRAINT符号子句或指定CONSTRAINT关键字时使用生成的约束名称。基于MySQL 5.7的NDB版本和早期的MySQL 8.0版本使用FOREIGN KEY index\_name值。

上述变化可能会给依赖以前的外键约束命名行为的应用程序带来不兼容的情况。

**改变的服务器默认值**

MySQL 8.0带有改进的默认值，目的是尽可能提供最好的开箱即用体验。这些变化是由以下事实驱动的：技术在进步（机器有更多的CPUS，使用SSD等等），更多的数据被存储，MySQL在发展（InnoDB、Group Replication、AdminAPI）等等。下表总结了已经改变的默认值，以便为大多数用户提供最佳的MySQL体验。

|  |  |  |
| --- | --- | --- |
| **Option/Parameter** | **Old Default** | **New Default** |
| *Server changes* |  |  |
| [**character\_set\_server**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_character_set_server) | latin1 | utf8mb4 |
| [**collation\_server**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_collation_server) | latin1\_swedish\_ci | utf8mb4\_0900\_ai\_ci |
| [**explicit\_defaults\_for\_timestamp**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_explicit_defaults_for_timestamp) | OFF | ON |
| [**optimizer\_trace\_max\_mem\_size**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_optimizer_trace_max_mem_size) | 16KB | 1MB |
| [**validate\_password\_check\_user\_name**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\security.html#sysvar_validate_password_check_user_name) | OFF | ON |
| [**back\_log**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_back_log) | -1 (autosize) changed from : back\_log = 50 + (max\_connections / 5) | -1 (autosize) changed to : back\_log = max\_connections |
| [**max\_allowed\_packet**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_max_allowed_packet) | 4194304 (4MB) | 67108864 (64MB) |
| [**max\_error\_count**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_max_error_count) | 64 | 1024 |
| [**event\_scheduler**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_event_scheduler) | OFF | ON |
| [**table\_open\_cache**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_table_open_cache) | 2000 | 4000 |
| [**log\_error\_verbosity**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_log_error_verbosity) | 3 (Notes) | 2 (Warning) |
| *InnoDB changes* |  |  |
| [**innodb\_undo\_tablespaces**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\innodb-storage-engine.html#sysvar_innodb_undo_tablespaces) | 0 | 2 |
| [**innodb\_undo\_log\_truncate**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\innodb-storage-engine.html#sysvar_innodb_undo_log_truncate) | OFF | ON |
| [**innodb\_flush\_method**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\innodb-storage-engine.html#sysvar_innodb_flush_method) | NULL | fsync (Unix), unbuffered (Windows) |
| [**innodb\_autoinc\_lock\_mode**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\innodb-storage-engine.html#sysvar_innodb_autoinc_lock_mode) | 1 (consecutive) | 2 (interleaved) |
| [**innodb\_flush\_neighbors**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\innodb-storage-engine.html#sysvar_innodb_flush_neighbors) | 1 (enable) | 0 (disable) |
| [**innodb\_max\_dirty\_pages\_pct\_lwm**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\innodb-storage-engine.html#sysvar_innodb_max_dirty_pages_pct_lwm) | 0 (%) | 10 (%) |
| [**innodb\_max\_dirty\_pages\_pct**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\innodb-storage-engine.html#sysvar_innodb_max_dirty_pages_pct) | 75 (%) | 90 (%) |
| *Performance Schema changes* |  |  |
| **performance-schema-instrument='wait/lock/metadata/sql/%=ON'** | OFF | ON |
| **performance-schema-instrument='memory/%=COUNTED'** | OFF | COUNTED |
| **performance-schema-consumer-events-transactions-current=ON** | OFF | ON |
| **performance-schema-consumer-events-transactions-history=ON** | OFF | ON |
| **performance-schema-instrument='transaction%=ON'** | OFF | ON |
| *Replication changes* |  |  |
| [**log\_bin**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\replication.html#sysvar_log_bin) | OFF | ON |
| [**server\_id**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\replication.html#sysvar_server_id) | 0 | 1 |
| [**log-slave-updates**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\replication.html#sysvar_log_slave_updates) | OFF | ON |
| [**expire\_logs\_days**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\replication.html#sysvar_expire_logs_days) | 0 | 30 |
| [**master-info-repository**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\replication.html#sysvar_master_info_repository) | FILE | TABLE |
| [**relay-log-info-repository**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\replication.html#sysvar_relay_log_info_repository) | FILE | TABLE |
| [**transaction-write-set-extraction**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\replication.html#sysvar_transaction_write_set_extraction) | OFF | XXHASH64 |
| [**slave\_rows\_search\_algorithms**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\replication.html#sysvar_slave_rows_search_algorithms) | INDEX\_SCAN, TABLE\_SCAN | INDEX\_SCAN, HASH\_SCAN |
| [**slave\_pending\_jobs\_size\_max**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\replication.html#sysvar_slave_pending_jobs_size_max) | 16M | 128M |
| [**gtid\_executed\_compression\_period**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\replication.html#sysvar_gtid_executed_compression_period) | 1000 | 0 |
| *Group Replication changes* |  |  |
| [**group\_replication\_autorejoin\_tries**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\group-replication.html#sysvar_group_replication_autorejoin_tries) | 0 | 3 |
| [**group\_replication\_exit\_state\_action**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\group-replication.html#sysvar_group_replication_exit_state_action) | ABORT\_SERVER | READ\_ONLY |
| [**group\_replication\_member\_expel\_timeout**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\group-replication.html#sysvar_group_replication_member_expel_timeout) | 0 | 5 |

关于已经添加的选项或变量的更多信息，请参见《MySQL服务器版本参考》中mysqld 8.0的选项/变量变化。

下面几节解释了对默认值的改变以及它们对你的部署可能产生的任何影响。

**服务器默认值**

* character\_set\_server系统变量和命令行选项 --character-set-server的默认值从latin1变为utf8mb4。这是服务器的默认字符集。此时，UTF8MB4是网络上的主流字符编码，这一变化使绝大多数MySQL用户的生活更容易。从5.7升级到8.0并不改变任何现有数据库对象的任何字符集。但是，除非你指定character\_set\_server回到你以前的默认值或明确设置字符集，那么新的模式、表或列默认使用utf8mb4。我们建议你尽可能地转移到utf8mb4。
* Collation\_server系统变量和命令行参数--collation-server的默认值从latin1\_swedish\_ci改为utf8mb4\_0900\_ai\_ci。这是服务器的默认校对，即字符集中字符的排序。整理和字符集之间存在着联系，因为每个字符集都有一个可能的整理列表。从5.7升级到8.0不会改变任何现有数据库对象的排序，但会对新对象生效。
* explicit\_defaults\_for\_timestamp系统变量的默认值从OFF（MySQL传统行为）变为ON（SQL标准行为）。这个选项最初在5.6中引入，在5.6和5.7中是OFF。
* optimizer\_trace\_max\_mem\_size系统变量的默认值从16KB变为1MB。旧的默认值导致优化器跟踪被截断，因为任何非琐碎的查询。这一变化确保了大多数查询的优化器跟踪是有用的。
* validate\_password\_check\_user\_name系统变量的默认值从OFF改为ON。这意味着，当validate\_password插件被启用时，默认情况下，它现在会拒绝与当前会话用户名匹配的密码。
* back\_log系统变量的autosize算法已经改变。 autosize（-1）的值现在被设置为max\_connections的值，它比50+（max\_connections / 5）的计算值大。在服务器无法跟上传入请求的情况下，back\_log会将传入的IP连接请求排队。在最坏的情况下，有了max\_connections数量的客户端试图同时重新连接，例如在网络故障后，它们都可以被缓冲，拒绝-重试的循环也就避免了。
* max\_allowed\_packet系统变量的默认值从4194304（4M）变为67108864（64M）。这个较大的默认值的主要优点是减少收到关于插入或查询大于max\_allowed\_packet的错误的机会。它应该和你想使用的最大的第11.3.4节 "BLOB和TEXT类型 "一样大。要恢复到以前的行为，设置max\_allowed\_packet=4194304。
* max\_error\_count系统变量的默认值从64变为1024。这确保了MySQL能够处理更多的警告，例如一个UPDATE语句触及了1000多条记录，其中很多都发出了转换警告。对于许多工具来说，批量更新是很常见的，以帮助减少复制滞后。外部工具如pt-online-schema-change默认为1000，而gh-ost默认为100。MySQL 8.0涵盖了这两种使用情况的全部错误历史。没有静态分配，所以这个变化只影响产生大量警告的语句的内存消耗。
* event\_scheduler系统变量的默认值从OFF变为ON。换句话说，事件调度器默认是启用的。这是SYS中新功能的一个启动器，例如 "杀死空闲事务"。
* table\_open\_cache系统变量的默认值从2000改为4000。这是一个微小的变化，增加了表访问的会话并发性。

log\_error\_verbosity系统变量的默认值从3（注意）改为2（警告）。其目的是使MySQL 8.0的错误日志在默认情况下不那么冗长。

**InnoDB默认值**

* 不兼容的变化 innodb\_undo\_tablespaces系统变量的默认值从0改为2。 该变量配置了InnoDB使用的undo tablespaces的数量。在MySQL 8.0中，innodb\_undo\_tablespaces的最小值是2，并且回滚段不能再系统表空间中创建。因此，这是一个不能恢复到5.7行为的情况。这一变化的目的是能够自动截断Undo日志（见下一项），回收（偶尔）长事务（如mysqldump）使用的磁盘空间。
* innodb\_undo\_log\_truncate系统变量的默认值从OFF变为ON。当启用时，超过innodb\_max\_undo\_log\_size定义的阈值的undo表空间被标记为截断。只有撤消的表空间可以被截断。不支持截断驻留在系统表空间中的Undo logs。从5.7升级到8.0会自动将你的系统转换为使用undo tablespaces，使用系统表空间在8.0中不是一个选项。
* innodb\_flush\_method系统变量的默认值在类Unix系统上从NULL变为fsync，在Windows系统上从NULL变为unbuffered。这更像是一个术语和选项的清理，没有任何实际的影响。对于Unix来说，这只是一个文档的变化，因为在5.7中默认也是fsync（默认NULL意味着fsync）。同样，在Windows上，innodb\_flush\_method默认值NULL在5.7中是指async\_unbuffered，在8.0中被默认unbuffered取代，这与现有的默认innodb\_use\_native\_aio=ON有相同的效果。
* 不兼容的变化 innodb\_autoinc\_lock\_mode系统变量的默认值从1（连续）变为2（交错）。将交错锁模式作为默认设置的变化反映了从基于语句的复制到基于行的复制作为默认复制类型的变化，这发生在MySQL 5.7。基于语句的复制需要连续的自动递增锁模式，以确保自动递增值以可预测和可重复的顺序分配给特定的SQL语句序列，而基于行的复制对SQL语句的执行顺序不敏感。因此，这个变化与基于语句的复制不兼容，可能会破坏一些依赖顺序自动递增的应用程序或用户生成的测试套件。以前的默认值可以通过设置innodb\_autoinc\_lock\_mode=1来恢复。
* innodb\_flush\_neighbors系统变量的默认值从1（启用）变为0（禁用）。这样做是因为快速IO（SSD）现在是部署的默认值。我们预计，对于大多数用户来说，这将导致一个小的性能提升。使用较慢硬盘的用户可能会看到性能损失，我们鼓励他们通过设置innodb\_flush\_neighbors=1恢复到以前的默认值。
* innodb\_max\_dirty\_pages\_pct\_lwm系统变量的默认值从0（%）变为10（%）。当innodb\_max\_dirty\_pages\_pct\_lwm=10时，InnoDB会在>10%的缓冲池包含修改过的（"脏"）页面时增加其刷新活动。这个变化的目的是为了稍微牺牲一下峰值吞吐量，以换取更稳定的性能。
* nnodb\_max\_dirty\_pages\_pct系统变量的默认值从75（%）变为90（%）。这一变化与innodb\_max\_dirty\_pages\_pct\_lwm的变化相结合，它们共同确保了InnoDB的平滑冲刷行为，避免了冲刷的突发性。要恢复到以前的行为，设置innodb\_max\_dirty\_pages\_pct=75和innodb\_max\_dirty\_pages\_pct\_lwm=0。

**性能模式的默认值**

* 性能模式的元数据锁定（MDL）工具默认是打开的。Performance-schema-instrument='wait/lock/metadata/sql/%=ON'的编译默认值从OFF变为ON。 这是在SYS中添加面向MDL的视图的一个促成因素。
* 性能模式内存仪表默认为打开。Performance-schema-instrument='memory/%=COUNTED'的编译默认值从OFF变为COUNTED。这一点很重要，因为如果在服务器启动后启用仪表，核算是不正确的，而且你可能会因为错过分配而得到一个负的余额，但抓住了一个空闲。
* 默认情况下，性能模式事务检测是打开的。Performance-schema-consumer-events-transactions-current=ON、performance-schema-consumer-events-transactions-history=ON和performance-schema-instrument='transaction%=ON'的编译默认值从OFF变为ON。

**复制默认值**

* log\_bin系统变量的默认值从OFF变为ON。换句话说，二进制日志是默认启用的。几乎所有的生产安装都启用了二进制日志，因为它被用于复制和时间点恢复。因此，通过默认启用二进制日志，我们消除了一个配置步骤，以后启用它需要重新启动mysqld。默认启用它也提供了更好的测试覆盖率，并且更容易发现性能回归。记得也要设置server\_id（见下面的修改）。8.0的默认行为就像你发出./mysqld --log-bin --server-id=1。如果你是在8.0版本，并且想要5.7版本的行为，你可以发布./mysqld --skip-log-bin --server-id=0。
* server\_id系统变量的默认值从0变为1（与log\_bin=ON的变化相结合）。服务器可以用这个默认ID启动，但实际上你必须根据正在部署的复制基础设施来设置服务器ID，以避免出现重复的服务器ID。
* log-slave-updates系统变量的默认值从OFF改为ON。 这将导致一个副本将复制的事件记录到它自己的二进制日志中。这个选项是组复制所需要的，同时也确保了在各种复制链设置中的正确行为，这已经成为今天的常态。
* expire\_logs\_days系统变量的默认值从0变为30。新的默认值30使mysqld定期清除超过30天的未使用的二进制日志。这一变化有助于防止在不再需要复制或恢复的二进制日志上浪费过多的磁盘空间。旧的值是0，禁用任何自动二进制日志清除。
* master\_info\_repository和relay\_log\_info\_repository系统变量的默认值从FILE变为TABLE。因此在8.0中，复制元数据默认存储在InnoDB中。这增加了可靠性，以尝试实现默认情况下的崩溃安全复制。
* 事务-写-集-提取系统变量的默认值从OFF变为XXHASH64。这个变化默认启用了事务写集。通过使用事务写集，源必须做稍多的工作来生成写集，但其结果对冲突检测有帮助。这是组复制的一个要求，新的默认值使得在源上启用二进制日志写集并行化很容易，以加快复制的速度。
* slave\_rows\_search\_algorithms系统变量的默认值从INDEX\_SCAN,TABLE\_SCAN改变为INDEX\_SCAN,HASH\_SCAN。这一变化通过减少复制应用者为将变化应用到没有主键的表而必须进行的表扫描的数量来加速基于行的复制。
* slave\_pending\_jobs\_size\_max系统变量的默认值从16M变为128M。这一变化增加了多线程复制的可用内存量。
* gtid\_executed\_compression\_period系统变量的默认值从1000改为0。这一变化确保mysql.gtid\_executed表的压缩只在需要时隐式发生。

**组复制默认值**

* group\_replication\_autorejoin\_tries的默认值从0变为3，这意味着默认情况下启用了自动重合。这个系统变量指定了成员在被驱逐或在达到group\_replication\_unreachable\_majority\_timeout设置之前无法与群组中的大多数成员联系时，自动重新加入群组的尝试次数。
* group\_replication\_exit\_state\_action的默认值从ABORT\_SERVER改为READ\_ONLY。这意味着当一个成员退出组时，例如在网络故障后，该实例会变成只读，而不是被关闭。
* group\_replication\_member\_expel\_timeout的默认值从0变成了5，这意味着被怀疑与群组失去联系的成员在5秒的检测时间后会被驱逐。

这些默认值中的大部分对于开发和生产环境都是合理的。但有一个例外，我们决定将名为innodb\_dedicated\_server的新选项设置为OFF，尽管我们建议在生产环境下将其打开。默认为OFF的原因是它会导致共享环境（如开发人员的笔记本电脑）变得无法使用，因为它占用了所有能找到的内存。

对于生产环境，我们建议将innodb\_dedicated\_server设置为ON。 当设置为ON时，以下InnoDB变量（如果没有明确指定的话）会根据可用内存进行自动缩放 innodb\_buffer\_pool\_size，innodb\_log\_file\_size和innodb\_flush\_method。见第15.8.12节，"为专门的MySQL服务器启用自动配置"。

虽然新的默认值是大多数使用情况下的最佳配置选择，但也有一些特殊情况，以及使用现有5.7配置选择的传统原因。例如，有些人喜欢升级到8.0，并尽可能少地改变他们的应用程序或操作环境。我们建议评估所有的新默认值并尽可能多地使用。大多数新的默认值可以在5.7中测试，所以你可以在升级到8.0之前在5.7生产中验证新的默认值。对于少数你需要旧的5.7值的默认值，在你的运行环境中设置相应的配置变量或启动选项。

在MySQL 8.0中，有一个性能模式variables\_info表，为每个系统变量显示它最近被设置的来源，以及它的数值范围。因此，在8.0中，你可以通过SQL访问关于配置变量及其值的所有信息。参见第27.12.14.2节，"性能模式变量信息表"。

### 2.11.5 为升级准备你的安装

在升级到最新的MySQL 8.0版本之前，通过执行下面描述的初步检查，确保你当前的MySQL 5.7或MySQL 8.0服务器实例的升级准备就绪。否则升级过程可能会失败。

提示

考虑使用MySQL Shell升级检查器工具，该工具使你能够验证MySQL服务器实例是否准备好升级。你可以选择一个你计划升级的目标MySQL Server版本，范围从MySQL Server 8.0.11到与当前MySQL Shell版本号匹配的MySQL Server版本号。升级检查器实用程序执行与指定的目标版本相关的自动检查，并向你建议你应该手动进行的进一步相关检查。升级检查器适用于MySQL 5.7和8.0的所有GA版本。MySQL Shell的安装说明可以在这里找到。

初步检查。

1、必须不存在以下问题。

* 必须没有使用过时的数据类型或函数的表。

如果表包含5.6.4之前格式的旧时间列（不支持小数秒精度的TIME、DATETIME和TIMESTAMP列），则不支持原地升级到MySQL 8.0。如果你的表仍然使用旧的时间列格式，在尝试就地升级到MySQL 8.0之前，使用REPAIR TABLE对它们进行升级。欲了解更多信息，请参阅《MySQL 5.7参考手册》中的服务器变更。

* 必须没有无主的.frm文件。
* 触发器不能有缺失或空的定义器或无效的创建环境（由SHOW TRIGGERS或INFORMATION\_SCHEMA TRIGGERS表显示的character\_set\_client、collation\_connection、Database Collation属性所表示）。任何这样的触发器必须被转储和恢复以解决这个问题。

要检查这些问题，执行这个命令。

mysqlcheck -u root -p --all-databases --check-upgrade

如果mysqlcheck报告任何错误，请纠正这些问题。

2、必须没有使用不具有本地分区支持的存储引擎的分区表。要识别这样的表，执行这个查询。

SELECT TABLE\_SCHEMA, TABLE\_NAME

FROM INFORMATION\_SCHEMA.TABLES

WHERE ENGINE NOT IN ('innodb', 'ndbcluster')

AND CREATE\_OPTIONS LIKE '%partitioned%';

任何由查询报告的表都必须被改变为使用InnoDB或者被变成非分区的。要把一个表的存储引擎改为InnoDB，执行这个语句。

ALTER TABLE ***table\_name*** ENGINE = INNODB;

关于将MyISAM表转换为InnoDB的信息，参见章节15.6.1.5，"将表从MyISAM转换为InnoDB"。

要使一个分区的表成为非分区的，执行这个语句。

ALTER TABLE ***table\_name*** REMOVE PARTITIONING;

3、在MySQL 8.0中可能保留了一些以前没有保留的关键字。见第9.3节，"关键词和保留字"。这可能导致以前用作标识符的词变得非法。要修复受影响的语句，请使用标识符引号。参见第9.2节，"模式对象名称"。

4、在MySQL 5.7 mysql系统数据库中必须没有与MySQL 8.0数据字典所使用的表具有相同名称的表。要识别具有这些名称的表，请执行此查询。

SELECT TABLE\_SCHEMA, TABLE\_NAME

FROM INFORMATION\_SCHEMA.TABLES

WHERE LOWER(TABLE\_SCHEMA) = 'mysql'

and LOWER(TABLE\_NAME) IN

(

'catalogs',

'character\_sets',

'check\_constraints',

'collations',

'column\_statistics',

'column\_type\_elements',

'columns',

'dd\_properties',

'events',

'foreign\_key\_column\_usage',

'foreign\_keys',

'index\_column\_usage',

'index\_partitions',

'index\_stats',

'indexes',

'parameter\_type\_elements',

'parameters',

'resource\_groups',

'routines',

'schemata',

'st\_spatial\_reference\_systems',

'table\_partition\_values',

'table\_partitions',

'table\_stats',

'tables',

'tablespace\_files',

'tablespaces',

'triggers',

'view\_routine\_usage',

'view\_table\_usage'

);

任何由查询报告的表必须被删除或重新命名（使用 RENAME TABLE）。这也可能需要对使用受影响表的应用程序进行修改。

5、不能有任何表的外键约束名称超过64个字符。使用这个查询来识别那些约束名称过长的表。

SELECT TABLE\_SCHEMA, TABLE\_NAME

FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME IN

(SELECT LEFT(SUBSTR(ID,INSTR(ID,'/')+1),

INSTR(SUBSTR(ID,INSTR(ID,'/')+1),'\_ibfk\_')-1)

FROM INFORMATION\_SCHEMA.INNODB\_SYS\_FOREIGN

WHERE LENGTH(SUBSTR(ID,INSTR(ID,'/')+1))>64);

对于一个约束名称超过64个字符的表，删除该约束，然后用不超过64个字符的约束名称将其添加回来（使用ALTER TABLE）。

必须没有由sql\_mode系统变量定义的过时的SQL模式。试图使用6、一个过时的SQL模式会阻止MySQL 8.0的启动。应修改使用过时的SQL模式的应用程序以避免它们。有关在MySQL 8.0中删除的SQL模式的信息，请参阅服务器变更。

7、不得有明确定义的列名超过64个字符的视图（在MySQL 5.7中允许列名不超过255个字符的视图）。为避免升级错误，在升级前应改变此类视图。目前，识别列名超过64个字符的视图的唯一方法是使用SHOW CREATE VIEW检查视图定义。也可以通过查询 INFORMATION\_SCHEMA.VIEWS 表来检查视图定义。

8、不得有任何表或存储过程的单个ENUM或SET列元素的长度超过255个字符或1020字节。在MySQL 8.0之前，ENUM或SET列元素的最大组合长度是64K。在MySQL 8.0中，单个ENUM或SET列元素的最大字符长度为255个字符，最大字节长度为1020字节。1020字节的限制支持多字节的字符集）。在升级到MySQL 8.0之前，修改任何超过新限制的ENUM或SET列元素。如果不这样做，就会导致升级失败并出现错误。

9、在升级到MySQL 8.0.13或更高版本之前，必须没有驻留在共享InnoDB表空间中的表分区，其中包括系统表空间和一般表空间。通过查询INFORMATION\_SCHEMA来识别共享表空间中的表分区。

如果从MySQL 5.7升级，运行这个查询。

SELECT DISTINCT NAME, SPACE, SPACE\_TYPE FROM INFORMATION\_SCHEMA.INNODB\_SYS\_TABLES

WHERE NAME LIKE '%#P#%' AND SPACE\_TYPE NOT LIKE 'Single';

如果从早期的MySQL 8.0版本升级，请运行这个查询。

SELECT DISTINCT NAME, SPACE, SPACE\_TYPE FROM INFORMATION\_SCHEMA.INNODB\_TABLES

WHERE NAME LIKE '%#P#%' AND SPACE\_TYPE NOT LIKE 'Single';

使用ALTER TABLE将表的分区从共享表空间移到按表文件的表空间... 重新组织分区。

ALTER TABLE ***table\_name*** REORGANIZE PARTITION ***partition\_name***

INTO (***partition\_definition*** TABLESPACE=innodb\_file\_per\_table);

10必须没有来自MySQL 8.0.12或更低版本的查询和存储程序定义在GROUP BY子句中使用ASC或DISC限定词。否则，升级到MySQL 8.0.13或更高版本可能会失败，复制到MySQL 8.0.13或更高版本的复制服务器也可能失败。有关其他细节，请参阅SQL变更。

11、你的MySQL 5.7安装不得使用MySQL 8.0不支持的功能。这里的任何变化都必然是针对安装的，但下面的例子说明了要寻找的那种东西。

一些服务器启动选项和系统变量已在MySQL 8.0中被删除。参见《MySQL 8.0中删除的特性》和第1.4节 "MySQL 8.0中增加、废弃或删除的服务器和状态变量及选项"。如果你使用其中任何一个，升级需要改变配置。

例子。由于数据字典提供了关于数据库对象的信息，服务器不再检查数据目录中的目录名来寻找数据库。因此，--ignore-db-dir选项是多余的，已经被删除。为了处理这个问题，从你的启动配置中删除任何 --ignore-db-dir的实例。此外，在升级到MySQL 8.0之前，删除或移动命名的数据目录子目录。(或者，让8.0服务器将这些目录作为数据库添加到数据字典中，然后使用DROP DATABASE删除这些数据库中的每个。)

12、如果你打算在升级时将 lower\_case\_table\_names 设置改为 1，请在升级前确保模式和表名是小写的。否则，由于模式或表名字母大小写不匹配，可能会发生故障。你可以使用下面的查询来检查模式和表名是否包含大写字母。

mysql> SELECT TABLE\_NAME FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME != LOWER(TABLE\_NAME) AND TABLE\_TYPE = 'BASE TABLE';

mysql> SELECT SCHEMA\_NAME FROM INFORMATION\_SCHEMA.SCHEMATA WHERE SCHEMA\_NAME != LOWER(SCHEMA\_NAME);

从MySQL 8.0.19开始，如果lower\_case\_table\_names=1，升级过程将检查表和模式名称，以确保所有字符是小写的。如果发现表或模式名称包含大写字符，则升级过程会出错。

注意

不建议在升级时改变 lower\_case\_table\_names设置。

如果由于上述任何问题而导致升级到MySQL 8.0失败，服务器将恢复对数据目录的所有更改。在这种情况下，删除所有redo log文件，并在现有的数据目录上重启MySQL 5.7服务器以解决错误。redo log文件（ib\_logfile\*）默认驻留在MySQL数据目录中。在错误被修复后，在再次尝试升级前执行缓慢关机（通过设置innodb\_fast\_shutdown=0）。

### 2.11.6 在Unix/Linux上升级MySQL二进制或基于包的安装

本节描述了如何在Unix/Linux上升级MySQL二进制和基于软件包的安装。描述了原地和逻辑升级方法。

* 就地升级
* 逻辑升级
* MySQL群集升级

**就地升级**

原地升级包括关闭旧的MySQL服务器，用新的MySQL二进制文件或软件包替换旧的MySQL二进制文件或软件包，在现有的数据目录下重新启动MySQL，并升级现有安装中需要升级的任何剩余部分。有关可能需要升级的内容的详情，请参见第2.11.3节，"MySQL升级过程中升级的内容"。

注意

如果你正在升级一个最初通过安装多个RPM包产生的安装，请升级所有的包，而不仅仅是一些。例如，如果你以前安装了服务器和客户端RPM，不要只升级服务器RPM。

对于某些Linux平台，从RPM或Debian软件包中安装的MySQL包括对管理MySQL服务器启动和关闭的systemd支持。在这些平台上，mysqld\_safe没有被安装。在这种情况下，使用systemd来启动和关闭服务器，而不是下面说明中使用的方法。参见第2.5.9节，"用systemd管理MySQL服务器"。

关于MySQL簇安装的升级，另见MySQL簇升级。

要执行原地升级。

1. 查看第2.11.1节 "开始之前 "中的信息。
2. 通过完成第2.11.5节 "为升级准备你的安装 "中的初步检查，确保你的安装准备就绪。
3. 如果你使用InnoDB的XA事务，在升级前运行XA RECOVER来检查是否有未提交的XA事务。如果返回结果，通过发布XA COMMIT或XA ROLLBACK语句，提交或回滚XA事务。
4. 如果你从MySQL 5.7.11或更早的版本升级到MySQL 8.0，并且有加密的InnoDB表空间，通过执行此语句旋转钥匙圈主钥匙。

ALTER INSTANCE ROTATE INNODB MASTER KEY;

5、 如果你通常在运行你的MySQL服务器时将innodb\_fast\_shutdown设置为2（冷关机），那么通过执行这些语句中的任何一条将其配置为执行快速或缓慢关机。

SET GLOBAL innodb\_fast\_shutdown = 1; -- fast shutdown

SET GLOBAL innodb\_fast\_shutdown = 0; -- slow shutdown

通过快速或慢速关机，InnoDB将其Undo logs和数据文件保留在一个可以在不同版本的文件格式不同的情况下进行处理的状态。

6、关掉旧的MySQL服务器。比如说

mysqladmin -u root -p shutdown

1. 升级MySQL二进制文件或软件包。如果升级二进制安装，解压新的MySQL二进制分发包。参见获取和解压分发包。对于基于包的安装，安装新的包。
2. 启动MySQL 8.0服务器，使用现有的数据目录。例如。
3. mysqld\_safe --user=mysql --datadir=***/path/to/existing-datadir*** &

如果有加密的InnoDB表空间，使用--early-plugin-load选项来加载钥匙圈插件。

当你启动MySQL 8.0服务器时，它会自动检测是否存在数据字典表。如果没有，服务器会在数据目录中创建它们，用元数据填充它们，然后继续其正常的启动顺序。在这个过程中，服务器升级了所有数据库对象的元数据，包括数据库、表空间、系统和用户表、视图和存储程序（存储过程和函数、触发器和事件调度器事件）。服务器还删除了以前用于元数据存储的文件。例如，从MySQL 5.7升级到MySQL 8.0后，你可能会注意到，表不再有.frm文件。

如果这个步骤失败，服务器会恢复数据目录的所有变化。在这种情况下，你应该删除所有redo log文件，在同一数据目录上启动你的MySQL 5.7服务器，并修复任何错误的原因。然后对5.7服务器进行另一次缓慢关机，并启动MySQL 8.0服务器再次尝试。

9、在上一个步骤中，服务器根据需要升级了数据字典。现在有必要执行任何剩余的升级操作

* 从MySQL 8.0.16开始，服务器作为前一个步骤的一部分这样做，在MySQL 5.7和MySQL 8.0之间对mysql系统数据库进行任何必要的修改，以便你可以利用新的权限或能力。它还为MySQL 8.0带来性能模式、INFORMATION\_SCHEMA和sys数据库的更新，并检查所有用户数据库是否与当前版本的MySQL不兼容。
* 在MySQL 8.0.16之前，服务器只在前一步中升级数据字典。在MySQL 8.0服务器成功启动后，执行mysql\_upgrade以执行剩余的升级任务。
* mysql\_upgrade -u root -p

然后关闭并重启MySQL服务器，以确保对系统表所做的任何更改都会生效。例如

mysqladmin -u root -p shutdown

mysqld\_safe --user=mysql --datadir=***/path/to/existing-datadir*** &

第一次启动MySQL 8.0服务器时（在前面的步骤中），你可能会注意到错误日志中关于未升级的表的信息。如果mysql\_upgrade已经成功运行，在你第二次启动服务器时应该没有这样的信息。

注意

升级过程不升级时区表的内容。关于升级说明，见第5.1.15节，"MySQL服务器时区支持"。

如果升级过程使用mysql\_upgrade（也就是在MySQL 8.0.16之前），该过程也不会升级帮助表的内容。关于这种情况下的升级说明，见第5.1.17节，"服务器端帮助支持"。

**逻辑升级**

逻辑升级包括使用备份或导出工具（如mysqldump或mysqlpump）从旧的MySQL实例导出SQL，安装新的MySQL服务器，并将SQL应用到新的MySQL实例。关于可能需要升级的细节，见第2.11.3节，"MySQL升级过程中升级的内容"。

注意

对于某些Linux平台，从RPM或Debian软件包中安装的MySQL包括用于管理MySQL服务器启动和关闭的systemd支持。在这些平台上，mysqld\_safe没有被安装。在这种情况下，使用systemd来启动和关闭服务器，而不是下面说明中使用的方法。参见第2.5.9节，"用systemd管理MySQL服务器"。

警告

将从以前的MySQL版本中提取的SQL应用到新的MySQL版本中，可能会因为新的、改变的、废弃的或移除的特性和功能所带来的不兼容而导致错误。因此，从以前的MySQL版本中提取的SQL可能需要修改以实现逻辑升级。

要在升级到最新的MySQL 8.0版本之前确定不兼容的地方，请执行第2.11.5节 "为升级准备你的安装 "中描述的步骤。

要执行逻辑升级。

1. 查看第2.11.1节 "开始之前 "中的信息。
2. 从以前的MySQL安装中导出你的现有数据。

mysqldump -u root -p

--add-drop-table --routines --events

--all-databases --force > data-for-upgrade.sql

注意

如果你的数据库包括存储程序，请在mysqldump中使用--routines和--events选项（如上所示）。--all-databases选项包括转储中的所有数据库，包括保存系统表的mysql数据库。

重要的是

如果你有包含生成列的表，使用MySQL 5.7.9或更高版本提供的mysqldump工具来创建你的转储文件。早期版本提供的mysqldump工具对生成的列定义使用了不正确的语法（Bug #20769542）。你可以使用INFORMATION\_SCHEMA.COLUMNS表来识别具有生成列的表。

3、 关闭旧的MySQL服务器。例如

mysqladmin -u root -p shutdown

4、安装MySQL 8.0。关于安装说明，见第2章，安装和升级MySQL。

1. 初始化一个新的数据目录，如第2.10.1节 "初始化数据目录 "所述。例如

mysqld --initialize --datadir=***/path/to/8.0-datadir***

复制显示在屏幕上的临时'root'@'localhost'密码或写入错误日志，以便以后使用。

6、启动MySQL 8.0服务器，使用新的数据目录。例如

mysqld\_safe --user=mysql --datadir=***/path/to/8.0-datadir*** &

7、重置根密码。

shell> **mysql -u root -p**

Enter password: **\*\*\*\*** <- enter temporary root password

mysql> **ALTER USER USER() IDENTIFIED BY '*your new password*';**

8、将先前创建的转储文件加载到新的MySQL服务器。例如

mysql -u root -p --force < data-for-upgrade.sql

注意

如果你的转储文件包括系统表，不建议在服务器上启用GTIDs（gtid\_mode=ON）时加载转储文件。mysqldump为使用非事务型MyISAM存储引擎的系统表发出DML指令，当GTIDs被启用时，这种组合是不允许的。还要注意的是，从一个启用了GTIDs的服务器加载转储文件到另一个启用了GTIDs的服务器，会导致产生不同的事务标识符。

9、执行任何剩余的升级

* 在MySQL 8.0.16和更高版本中，关闭服务器，然后用-upgrade=FORCE选项重新启动它，以执行剩余的升级任务。

mysqladmin -u root -p shutdown

mysqld\_safe --user=mysql --datadir=***/path/to/8.0-datadir*** --upgrade=FORCE &

在用-upgrade=FORCE重新启动时，服务器会在MySQL 5.7和MySQL 8.0之间对mysql系统模式进行任何必要的更改，以便你可以利用新的权限或能力。它还为MySQL 8.0带来性能模式、INFORMATION\_SCHEMA和sys模式的更新，并检查所有用户模式是否与当前版本的MySQL不兼容。

* 在MySQL 8.0.16之前，执行mysql\_upgrade来执行剩余的升级任务。
* mysql\_upgrade -u root -p

然后关闭并重新启动MySQL服务器，以确保对系统表所做的任何更改都会生效。例如

mysqladmin -u root -p shutdown

mysqld\_safe --user=mysql --datadir=***/path/to/8.0-datadir*** &

注意

升级过程不升级时区表的内容。关于升级说明，见第5.1.15节 "MySQL服务器时区支持"。

如果升级过程使用mysql\_upgrade（也就是在MySQL 8.0.16之前），该过程也不会升级帮助表的内容。关于这种情况下的升级说明，见第5.1.17节 "服务器端帮助支持"。

注意

加载包含MySQL 5.7 mysql模式的转储文件会重新创建两个不再使用的表：event和proc。相应的MySQL 8.0表是events和routine，这两个表都是数据字典表，是受保护的）。在你确信升级成功后，你可以通过执行这些SQL语句来删除事件和程序表。

DROP TABLE mysql.event;

DROP TABLE mysql.proc;

**MySQL群集升级**

本节中的信息是对就地升级中描述的就地升级程序的辅助，供你在升级MySQL簇时使用。

从MySQL 8.0.16开始，MySQL簇的升级可以作为常规的滚动升级来执行，遵循通常的三个有序步骤。

1. 升级MGM节点。
2. 一次性升级数据节点。
3. 一次一次地升级API节点（包括MySQL服务器）。

升级每个节点的方式与MySQL 8.0.16之前几乎一样，因为升级数据字典和升级系统表之间是分开的。升级每个单独的mysqld有两个步骤。

1、导入数据字典。

用-upgrade=MINIMAL选项启动新服务器，升级数据字典，但不升级系统表。这与MySQL 8.0.16之前启动服务器而不调用mysql\_upgrade的操作基本相同。

为了完成这个阶段，MySQL服务器必须连接到NDB。如果有任何NDB或NDBINFO表存在，而服务器不能连接到集群，它将以错误信息退出。

Failed to Populate DD tables.

2、升级系统表。

在MySQL 8.0.16之前，DBA调用mysql\_upgrade客户端来升级系统表。从MySQL 8.0.16开始，服务器会执行这一操作。要升级系统表，重启每个单独的mysqld，不要使用--upgrade=MINIMAL选项。

### 2.11.7 使用MySQL Yum仓库升级MySQL

对于受支持的基于Yum的平台（参见第2.5.1节，"使用MySQL Yum仓库在Linux上安装MySQL"，获取列表），你可以使用MySQL Yum仓库对MySQL进行就地升级（即替换旧版本，然后使用旧数据文件运行新版本）。

注意事项

* 在对MySQL进行任何更新之前，请仔细遵循第2.11节 "升级MySQL "中的说明。在那里讨论的其他指示中，在更新前备份你的数据库是特别重要的。
* 下面的说明假定你已经用MySQL Yum库或直接从MySQL开发区的MySQL下载页面下载的RPM包安装了MySQL；如果不是这种情况，请按照《使用MySQL Yum库替换第三方分发的MySQL》中的说明进行。

1、选择一个目标系列

默认情况下，MySQL Yum资源库会将MySQL更新到你在安装时选择的发行系列中的最新版本（详见选择发行系列），这意味着，例如，5.7.x的安装不会自动更新到8.0.x版本。要更新到另一个发行系列，你必须首先禁用已选择的系列的子库（默认的，或自己选择的），并启用目标系列的子库。要做到这一点，请参阅选择发布系列中给出的一般说明。对于从MySQL 5.7升级到8.0，执行选择发布系列中说明的相反步骤，禁用MySQL 5.7系列的子库，为MySQL 8.0系列启用子库。

作为一般规则，要从一个发布系列升级到另一个发布系列，应转到下一个系列，而不是跳过一个系列。

重要信息

关于从MySQL 5.7升级到8.0的重要信息，请参阅从MySQL 5.7升级到8.0。

2、升级MySQL

对于未启用dnf的平台，可通过以下命令升级MySQL及其组件。

sudo yum update mysql-server

对于已启用dnf的平台。

sudo dnf upgrade mysql-server

或者，你可以通过告诉Yum更新你系统上的所有东西来更新MySQL，这可能需要相当多的时间。对于未启用dnf的平台。

sudo yum update

对于已启用dnf的平台。

sudo dnf upgrade

3、重启MySQL

在Yum更新后，MySQL服务器总是重新启动。在MySQL 8.0.16之前，在服务器重启后运行mysql\_upgrade，以检查并可能解决旧数据和升级后的软件之间的任何不兼容问题。mysql\_upgrade还执行其他功能；详情见4.4.5节，"mysql\_upgrade--检查并升级MySQL表"。从MySQL 8.0.16开始，这个步骤就不需要了，因为服务器会执行所有以前由mysql\_upgrade处理的任务。

你也可以只更新一个特定的组件。使用下面的命令来列出所有已安装的MySQL组件的包（对于支持dnf的系统，在命令中用dnf替换yum）。

sudo yum list installed | grep "^mysql"

在确定了你所选择的组件的软件包名称后，用下面的命令更新软件包，用软件包的名称代替package-name。对于未启用dnf的平台。

sudo yum update ***package-name***

对于启用了dnf的平台。

sudo dnf upgrade ***package-name***

**升级共享的客户端库**

在使用Yum资源库更新MySQL后，用旧版本的共享客户端库编译的应用程序应该继续工作。

如果你重新编译应用程序并将它们与更新的库动态链接。作为新版本共享库的典型情况，在较新和较旧的库之间存在符号版本的差异或增加（例如，较新的、标准的8.0共享客户端库和一些较旧的、由Linux发行版软件库原生提供的共享库或变种版本之间，或从一些其他来源），任何使用更新的、较新的共享库编译的应用程序在部署应用程序的系统上需要这些更新的库。正如预期的那样，如果这些库没有到位，需要共享库的应用程序就会失败。出于这个原因，请确保在这些系统上部署来自MySQL的共享库包。要做到这一点，请将MySQL Yum仓库添加到这些系统中（参见添加MySQL Yum仓库），并使用 "用Yum安装其他MySQL产品和组件 "中给出的说明安装最新的共享库。

### 2.11.8 用MySQL APT资源库升级MySQL

在Debian和Ubuntu平台上，要执行MySQL及其组件的原地升级，请使用MySQL APT资源库。见《使用MySQL APT资源库快速指南》中的用MySQL APT资源库升级MySQL。

### 2.11.9 使用MySQL SLES资源库升级MySQL

在SUSE Linux Enterprise Server（SLES）平台上，要对MySQL及其组件进行原地升级，请使用MySQL SLES资源库。请参阅《使用MySQL SLES资源库快速指南》中用MySQL SLES资源库升级MySQL。

### 2.11.10 在Windows上升级MySQL

有两种方法可以在Windows上升级MySQL。

* 使用MySQL安装程序
* 使用Windows的ZIP档案分发

你选择的方法取决于现有安装是如何进行的。在继续进行之前，请查看第2.11节 "升级MySQL"，以了解有关升级MySQL的其他信息，这些信息不是针对Windows的。

注意

无论你选择哪种方法，在执行升级之前一定要备份你当前的MySQL安装。参见第7.2节，"数据库备份方法"。

不支持非GA版本之间的升级（或从非GA版本到GA版本）。在非GA版本中会发生重大的开发变化，你可能会遇到兼容性问题或启动服务器的问题。

注意

MySQL安装程序不支持社区版和商业版之间的升级。如果你需要这种类型的升级，请使用ZIP存档方法执行。

**用MySQL安装器升级MySQL**

如果当前的服务器安装是用MySQL Installer进行的，并且升级是在当前的版本系列内，那么用MySQL Installer执行升级是最好的方法。MySQL安装程序不支持版本系列之间的升级，例如从5.7到8.0，而且它不提供升级指示器来提示你进行升级。有关在发布系列之间升级的说明，请参见使用Windows ZIP发行版升级MySQL。

要使用MySQL Installer执行升级**。**

1. 启动MySQL安装程序。
2. 从仪表板上，点击目录，下载目录的最新变化。只有当仪表板上显示服务器的版本号旁边有一个箭头时，已安装的服务器才能被升级。
3. 点击升级。所有具有较新版本的产品现在都出现在一个列表中。

注意

MySQL安装程序对于同一发布系列中的里程碑版本（Pre-Release）取消选择服务器升级选项。此外，它还显示一个警告，表明不支持升级，确定继续下去的风险，并提供一个手动执行升级的步骤摘要。你可以重新选择服务器升级，并在你自己的风险下进行。

4、除MySQL服务器产品外，取消选择所有产品，除非你打算在这个时候升级其他产品，然后点击下一步。

5、单击 "执行 "以开始下载。当下载完成后，点击下一步开始升级操作。

升级到MySQL 8.0.16及更高版本可能会显示一个选项，以跳过系统表的升级检查和过程。关于这个选项的更多信息，请参见重要的服务器升级条件。

6、配置服务器。

**使用Windows ZIP发行版升级MySQL**

1、要使用Windows ZIP档案分发版执行升级。

从https://dev.mysql.com/downloads/，下载最新的MySQL的Windows ZIP存档发行版。

2、如果服务器正在运行，停止它。如果服务器是作为服务安装的，从命令提示符中用以下命令停止服务。

C:\> **SC STOP *mysqld\_service\_name***

或者，使用NET STOP mysqld\_service\_name 。

如果你没有将MySQL服务器作为服务运行，使用mysqladmin来停止它。例如，在从MySQL 5.7升级到8.0之前，从MySQL 5.7使用mysqladmin，如下所示。

C:\> **"C:\Program Files\MySQL\MySQL Server 5.7\bin\mysqladmin" -u root shutdown**

注意

如果MySQL根用户账户有密码，用-p选项调用mysqladmin，并在提示时输入密码。

3、提取ZIP档案。你可以覆盖你现有的MySQL安装（通常位于C:\mysql），或将其安装到不同的目录，如C:\mysql8。建议覆盖现有的安装。

4、重新启动服务器。例如，如果你将MySQL作为服务运行，使用SC START mysqld\_service\_name或NET START mysqld\_service\_name命令，否则直接调用mysqld。

5、在MySQL 8.0.16之前，以管理员身份运行mysql\_upgrade来检查你的表，如果有必要，尝试修复它们，如果你的授予表已经改变，则更新它们，以便你可以利用任何新的功能。见第4.4.5节，"mysql\_upgrade--检查和升级MySQL表"。从MySQL 8.0.16开始，这个步骤就不需要了，因为服务器会执行所有以前由mysql\_upgrade处理的任务。

6、如果你遇到错误，请参阅第2.3.5节，"Microsoft Windows MySQL服务器安装的故障排除"。

### 2.11.11 升级MySQL的Docker安装

要升级MySQL的Docker安装，请参阅升级MySQL服务器容器。

### 2.11.12 升级故障排除

* 在MySQL 5.7实例中，表的.frm文件和InnoDB数据字典之间的模式不匹配可能导致升级到MySQL 8.0失败。这种不匹配可能是由于.frm文件损坏。为了解决这个问题，在再次尝试升级之前转储和恢复受影响的表。
* 如果出现问题，例如新的mysqld服务器不启动，请确认你没有以前安装时的旧的my.cnf文件。你可以用 --print-defaults 选项来检查（例如，mysqld --print-defaults）。如果该命令显示除程序名以外的任何内容，你有一个活动的my.cnf文件，会影响服务器或客户端的操作。
* 如果在升级后，你在编译客户程序时遇到问题，如命令不同步或意外的核心转储，你可能在编译程序时使用了旧的头文件或库文件。在这种情况下，检查你的mysql.h文件和libmysqlclient.a库的日期，以验证它们是否来自新的MySQL发行版。如果不是，用新的头文件和库重新编译你的程序。如果库的主要版本号改变了（例如，从libmysqlclient.so.20到libmysqlclient.so.21），针对共享客户端库编译的程序可能也需要重新编译。
* 如果你用一个给定的名称创建了一个用户定义的函数（UDF），并将MySQL升级到一个实现了相同名称的新内置函数的版本，该UDF会变得无法访问。要纠正这一点，使用DROP FUNCTION删除UDF，然后使用CREATE FUNCTION用不同的不冲突的名字重新创建UDF。如果MySQL的新版本实现了一个与现有存储函数同名的内置函数，情况也是如此。参见第9.2.5节 "函数名称解析和解决"，以了解描述服务器如何解释对不同种类的函数的引用的规则。
* 如果由于第2.11.5节 "为升级准备你的安装 "中概述的任何问题而导致升级到MySQL 8.0失败，服务器将恢复对数据目录的所有更改。在这种情况下，删除所有redo log文件，并在现有的数据目录上重启MySQL 5.7服务器，以解决这些错误。redo log文件（ib\_logfile\*）默认驻留在MySQL数据目录中。在错误被修复后，在再次尝试升级前执行缓慢关机（通过设置innodb\_fast\_shutdown=0）。

### 2.11.13 重建或修复表或索引

本节描述了如何重建或修复表或索引，这可能是必要的。

* 对MySQL处理数据类型或字符集方式的改变。例如，整理的错误可能已被纠正，需要重建表以更新使用该整理的字符列的索引。
* 需要通过CHECK TABLE、mysqlcheck或mysql\_upgrade报告的表修复或升级。

重建表的方法包括

* Dump and Reload方法
* ALTER TABLE方法
* REPAIR TABLE方法

**转储和重新加载方法**

如果你重建表是因为不同版本的MySQL在二进制（原地）升级或降级后不能处理它们，你必须使用转储和重载方法。在升级或降级之前，使用你的原始版本的MySQL转储表。然后在升级或降级后重新加载这些表。

如果你使用dump-and-reload方法，重建表只是为了重建索引，你可以在升级或降级之前或之后执行dump。重新加载仍然必须在之后进行。

如果你需要重建一个InnoDB表，因为CHECK TABLE操作表明需要升级表，使用mysqldump创建一个转储文件，并使用mysql重新加载该文件。如果CHECK TABLE操作指示有损坏或导致InnoDB失败，请参阅15.21.2节 "强制InnoDB恢复"，了解使用innodb\_force\_recovery选项重启InnoDB的信息。要了解CHECK TABLE可能遇到的问题类型，请参考章节13.7.3.2 "CHECK TABLE语句 "中的InnoDB说明。

要通过转储和重载重建一个表，使用mysqldump创建一个转储文件和mysql重载该文件。

mysqldump ***db\_name*** t1 > dump.sql

mysql ***db\_name*** < dump.sql

要重建单个数据库中的所有表，请指定数据库名称，不要有任何后面的表名。

mysqldump ***db\_name*** > dump.sql

mysql ***db\_name*** < dump.sql

要重建所有数据库中的所有表，使用--all-databases选项。

mysqldump --all-databases > dump.sql

mysql < dump.sql

**ALTER TABLE方法**

要用ALTER TABLE重建表，使用 "空 "的改变；也就是说，一个ALTER TABLE语句 "改变 "表，使其使用已经拥有的存储引擎。例如，如果t1是一个InnoDB表，使用这个语句。

ALTER TABLE t1 ENGINE = InnoDB;

如果你不确定在ALTER TABLE语句中指定哪个存储引擎，使用SHOW CREATE TABLE来显示表的定义。

REPAIR TABLE方法

REPAIR TABLE方法只适用于MyISAM, ARCHIVE, 和CSV表。

如果表检查操作表明存在损坏或需要升级，可以使用 REPAIR TABLE。例如，要修复一个MyISAM表，使用这个语句。

REPAIR TABLE t1;

mysqlcheck --repair提供了对REPAIR TABLE语句的命令行访问。这可能是一种更方便的修复表的手段，因为你可以使用--数据库或--所有数据库选项，分别修复特定数据库或所有数据库中的所有表。

mysqlcheck --repair --databases ***db\_name*** ...

mysqlcheck --repair --all-databases

### 2.11.14 将MySQL数据库复制到另一台机器上

在需要在不同架构之间转移数据库的情况下，你可以使用mysqldump来创建一个包含SQL语句的文件。然后，你可以将该文件传输到另一台机器上，并将其作为输入提供给mysql客户端。

使用mysqldump --help查看有哪些选项可用。

注意

如果在你创建转储的服务器上正在使用GTIDs(gtid\_mode=ON)，默认情况下，mysqldump在转储中包括gtid\_executed集的内容，以将这些内容转移到新机器上。其结果可能因涉及的MySQL Server版本而不同。查看mysqldump的-set-gtid-purged选项的描述，了解你使用的版本会发生什么，以及如果默认行为的结果不适合你的情况，如何改变行为。

在两台机器之间移动数据库的最简单（尽管不是最快）的方法是在数据库所在的机器上运行以下命令。

mysqladmin -h '***other\_hostname***' create ***db\_name***

mysqldump ***db\_name*** | mysql -h '***other\_hostname***' ***db\_name***

如果想通过慢速网络从远程机器复制数据库，可以使用这些命令。

如果你想通过慢速网络从远程机器复制一个数据库，你可以使用这些命令。

mysqladmin create ***db\_name***

mysqldump -h '***other\_hostname***' --compress ***db\_name*** | mysql ***db\_name***

你也可以将转储存储在一个文件中，将文件传输到目标机，然后将文件加载到那里的数据库中。例如，你可以像这样在源机器上把数据库转储到一个压缩文件中。

mysqldump --quick ***db\_name*** | gzip > ***db\_name***.gz

将包含数据库内容的文件传输到目标机上，并在那里运行这些命令。

mysqladmin create ***db\_name***

gunzip < ***db\_name***.gz | mysql ***db\_name***

你也可以使用mysqldump和mysqlimport来传输数据库。对于大表，这比简单地使用mysqldump快得多。在下面的命令中，DUMPDIR代表你用来存储mysqldump输出的目录的完整路径名称。

首先，为输出文件创建目录并转储数据库。

mkdir ***DUMPDIR***

mysqldump --tab=***DUMPDIR***

***db\_name***

然后将DUMPDIR目录中的文件转移到目标机上的一些相应目录中，并将文件加载到那里的MySQL中。

mysqladmin create ***db\_name*** # create database

cat ***DUMPDIR***/\*.sql | mysql ***db\_name*** # create tables in database

mysqlimport ***db\_name***

***DUMPDIR***/\*.txt # load data into tables

不要忘记复制mysql数据库，因为那是授予表的存储地。你可能不得不在新机器上以MySQL根用户的身份运行命令，直到你有了mysql数据库。

在新机器上导入mysql数据库后，执行mysqladmin flush-privileges，这样服务器就会重新加载授予表的信息。

## 2.12 降级MySQL

不支持从MySQL 8.0降级到MySQL 5.7，或从MySQL 8.0版本降级到以前的MySQL 8.0版本。唯一支持的替代方法是恢复升级前的备份。因此，在开始升级过程之前，你必须备份你的数据。

## 2.13 Perl安装注意事项

Perl DBI模块为数据库访问提供了一个通用接口。你可以编写一个DBI脚本，在不改变的情况下与许多不同的数据库引擎一起工作。要使用DBI，你必须安装DBI模块，以及为你要访问的每种类型的数据库服务器安装一个数据库驱动（DBD）模块。对于MySQL，这个驱动是DBD::mysql模块。

注意

*Perl支持不包括在MySQL发行版中。你可以从Unix的http://search.cpan.org，或通过在Windows上使用ActiveState ppm程序获得必要的模块。下面几节描述了如何做到这一点。*

DBI/DBD接口需要Perl 5.6.0，最好是5.6.1或更高版本。如果你有一个旧版本的Perl，DBI就不能工作。你应该使用DBD::mysql 4.009或更高版本。尽管有早期的版本，但它们不支持MySQL 8.0的全部功能。

### 2.13.1 在Unix上安装Perl

对MySQL Perl的支持要求你已经安装了MySQL客户端编程支持（库和头文件）。大多数安装方法都会安装必要的文件。如果你在Linux上从RPM文件中安装MySQL，一定要同时安装开发者RPM。客户端程序在客户端RPM中，但客户端编程支持在开发者RPM中。

你需要的Perl支持文件可以从CPAN（综合Perl档案网络）获得，网址是http://search.cpan.org。

在Unix上安装Perl模块的最简单方法是使用CPAN模块。比如说

shell> **perl -MCPAN -e shell**

cpan> **install DBI**

cpan> **install DBD::mysql**

DBD::mysql的安装运行了一些测试。这些测试试图使用默认的用户名和密码连接到本地 MySQL 服务器。(默认的用户名是你在Unix上的登录名，在Windows上的ODBC。默认密码是 "无密码"）。) 如果你不能用这些值连接到服务器（例如，如果你的账户有一个密码），测试就会失败。你可以使用强制安装DBD::mysql来忽略失败的测试。

DBI需要Data::Dumper模块。它可能已经安装了；如果没有，你应该在安装DBI之前安装它。

也可以以压缩的tar文件的形式下载模块发行版，并手动构建模块。例如，要解压并构建一个DBI发行版，可以使用这样的程序。

1、解压发行版到当前目录中。

shell> **gunzip < DBI-*VERSION*.tar.gz | tar xvf -**

这个命令创建了一个名为DBI-VERSION的目录。

2、改变位置到解压后的发行版的顶层目录中。

shell> **cd DBI-*VERSION***

3、建立发行版并编译所有内容。

shell> **perl Makefile.PL**

shell> **make**

shell> **make test**

shell> **make install**

make test 命令很重要，因为它验证了模块是否工作。注意，当你在DBD::mysql安装期间运行该命令以锻炼接口代码时，MySQL服务器必须正在运行，否则测试失败。

每当你安装一个新的MySQL版本时，重建并重新安装DBD::mysql发行版是个好主意。这可以确保正确安装最新版本的MySQL客户端库。

如果你没有在系统目录下安装Perl模块的访问权限，或者你想安装本地Perl模块，下面的参考资料可能很有用：http://learn.perl.org/faq/perlfaq8.html#How-do-I-keep-my-own-module-library-directory-

### 2.13.2 在Windows上安装ActiveState Perl

在Windows上，你应该做以下事情来安装带有ActiveState Perl的MySQL DBD模块。

1、从http://www.activestate.com/Products/ActivePerl/ 获取ActiveState Perl并安装它。

2、打开一个控制台窗口。

3、如果有必要，设置HTTP\_proxy变量。例如，你可以尝试这样的设置。

:\> **set HTTP\_proxy=my.proxy.com:3128**

4、启动PPM程序。

C:\> **C:\perl\bin\ppm.pl**

5、如果你以前没有这样做，请安装DBI。

ppm> **install DBI**

6、如果这成功了，运行下面的命令。

ppm> **install DBD-mysql**

这个程序应该在ActiveState Perl 5.6或更高版本中工作。

如果你不能使该程序工作，你应该安装ODBC驱动程序，而不是通过ODBC连接到MySQL服务器。

use DBI;

$dbh= DBI->connect("DBI:ODBC:$dsn",$user,$password) ||

die "Got error $DBI::errstr when connecting to $dsn\n";

### 2.13.3 使用Perl DBI/DBD接口的问题

如果Perl报告说它找不到./mysql/mysql.so模块，那么问题可能是Perl无法找到libmysqlclient.so共享库。你应该可以通过以下方法解决这个问题。

* 将 libmysqlclient.so 复制到其他共享库所在的目录中（可能是 /usr/lib 或 /lib）。
* 修改用于编译DBD::mysql的-L选项，以反映libmysqlclient.so的实际位置。
* 在Linux上，你可以在/etc/ld.so.conf文件中添加libmysqlclient.so所在目录的路径名称。
* 在LD\_RUN\_PATH环境变量中加入libmysqlclient.so所在目录的路径名。有些系统使用LD\_LIBRARY\_PATH代替。

注意，如果有其他链接器找不到的库，你可能还需要修改-L选项。例如，如果链接器找不到 libc，因为它在 /lib 中，而链接命令指定的是 -L/usr/lib，请将 -L 选项改为 -L/lib 或在现有的链接命令中添加 -L/lib。

如果你从DBD::mysql得到以下错误，你可能使用的是gcc（或使用用gcc编译的旧二进制文件）。

/usr/bin/perl: can't resolve symbol '\_\_moddi3'

/usr/bin/perl: can't resolve symbol '\_\_divdi3'

在链接命令中加入 -L/usr/lib/gcc-lib/... -lgcc到链接命令中，当mysql.so库被构建时（当你编译Perl客户端时检查mysql.so的make输出）。-L选项应该指定libgcc.a在你的系统中所在的目录的路径名称。

这个问题的另一个原因可能是Perl和MySQL不是同时用gcc编译的。在这种情况下，你可以通过用gcc编译两者来解决不匹配的问题。