**Kamailio安装配置指南**

|  |  |
| --- | --- |
| 文档编号： |  |
| 编 写 人： | 崔士阳 |
| 审 批 人： | [经理] |
| 生效日期： | 2015/09/01 |

**爱悠团队**

文档历史：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 修订者 | 修订内容 | 修订日期 | 审批者 | 审批日期 |
| 0.1.0 | 崔士阳 | 文档初稿 | 2015.09.01 |  |  |
| 0.1.1 | 崔士阳 | 增加KamailioPC端的通讯测试方法 | 2015.12.03 |  |  |
| 0.1.2 | 崔士阳 | 规范文档布局以及英文命名 | 2015.12.14 |  |  |
| 0.2.0 | 崔士阳 | 1. 增加centos系统下第三方库的安装方法 2. 增加json-c-0.12库的安装方法 3. 补充安装过程中可能出现的错误的介绍和解决办法 | 2016．1.8 |  |  |
| 0.2.1 | 崔士阳 | 1. 添加在MySQL5.7版本下编译kamailio源码时一些问题的解决办法 2. 将kamailio版本由4.3.2改为4.3.4 | 2016.1.29 |  |  |
| 0.2.2 | 崔士阳 | 1. 添加libffi、Glib和Thrift库的安装方法   2、 增加编译过程中可能遇到的问题及解决方法 | 2016.2.27 |  |  |
| 0.3.0 | 崔士阳 | 添加如何在同一台服务器运行多套kamailio服务的实现方法 | 2016.03.15 |  |  |
|  |  |  |  |  |  |

*填写说明：*

* *版本号规则：版本号采用三级生成规则，即0.0.0，第一位表示主版本号，每一次正式发布，主版本号加1，第二位表示从版本号，每一次对文档进行较大范围修订（增、删章节，修改内容主干等），从版本号加1；第三位为修订版本号，在不改变文档主体内容结构与思想，只做文字上的修饰与调整，则修订版本加1。版本号不采用进制策略，按规则累加，当上层版本变化时，次级版本归0。例如0.0.92，当从版本号升为1时，修订版本号归0，成为0.1.0。*
* *修订内容：详细描述修订及修订内容所在章节*

目录

[1 先决条件 3](#_Toc445816145)

[2 准备源代码 3](#_Toc445816146)

[3 安装 3](#_Toc445816147)

[3.1 安装第三方依赖库 3](#_Toc445816148)

[3.1.1 Ubuntu系统第三方库安装方法 4](#_Toc445816149)

[3.1.2 Centos系统第三方库安装方法 4](#_Toc445816150)

[3.1.3 安装JSON-C 4](#_Toc445816151)

[3.1.4 安装libffi库 4](#_Toc445816152)

[3.1.5安装Glib库 5](#_Toc445816153)

[3.1.6安装libevent库 5](#_Toc445816154)

[3.1.7安装Thrift库 6](#_Toc445816155)

[3.1.8安装mysql-connector-mysql-connector-c++ 6](#_Toc445816156)

[3.2 执行编译和安装 6](#_Toc445816157)

[4 配置 8](#_Toc445816158)

[5 生成Kamailio的MySQL数据库 13](#_Toc445816159)

[6 patch文件的使用 14](#_Toc445816160)

[7 启动Kamailio服务 14](#_Toc445816161)

[8 监视Kamailio 15](#_Toc445816162)

[9 注册用户 15](#_Toc445816163)

[10 测试Kamailio基本服务 15](#_Toc445816164)

[11 同一台服务器运行多套kamailio服务 19](#_Toc445816165)

[11.1 配置文件强化说明 19](#_Toc445816166)

[11.2 指定路径编译安装Kamailio 21](#_Toc445816167)

[11.3 生成自定义的Kamailio数据库 21](#_Toc445816168)

[11.4 配置kamailio.cfg 22](#_Toc445816169)

[11.4.1 指定使用的数据库 22](#_Toc445816170)

[11.4.2 更改端口 22](#_Toc445816171)

# 先决条件

* 读者熟悉Ubuntu/Centos、Linux各种基本操作。
* 读者熟悉apt-get软件包管理工具。
* 读者熟悉编译安装Linux程序的一般过程。
* 操作用户具有sudo权限。

# 准备源代码

源码包为kamailio-4.3.4\_src.tar.gz，将其解压到Linux系统下即可。

# 安装

## 安装第三方依赖库

由于不同的Linux系统，同一个依赖库的名称也会不一样，在线安装方法也不一样，所以在此以Ubuntu和centos两种Linux系统为例进行讲解。

## 3.1.1 Ubuntu系统第三方库安装方法

*sudo apt-get install gcc flex bison libmysqlclient-dev make libcurl4-openssl-dev libxml2-dev libpcre3-dev libncurses5-dev libreadline6 libreadline-dev e2fsprogs-dev uuid-dev libuuid-dev automake libssl-dev byacc libevent-dev libquadmath0-dbg build-essential*

## 3.1.2 Centos系统第三方库安装方法

* *sudo yum install gcc gcc-c++ flex bison make openssl openssl-devel libxml2 libxml2-devel zlib-devel ncurses\* e2fsprogs-devel uuid-devel libuuid-devel readline6 readline-devel curl-devel automake libtool libevent-devel byacc bison boost-devel libquadmath libquadmath-devel*

## 3.1.3 安装JSON-C

首先下载JSON-C-0.12源文件，下载路径为

[**https://s3.amazonaws.com/json-c\_releases/releases/json-c-0.12.tar.gz**](https://s3.amazonaws.com/json-c_releases/releases/json-c-0.12.tar.gz)**，**

然后执行如下指令进行安装

*yum install autoconf*

*tar xvf json-c-0.12.tar.gz*

*cd json-c-0.12*

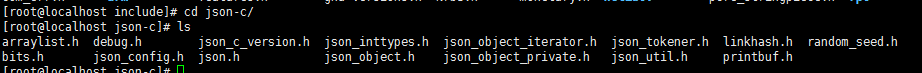
*sed -i s/-Werror// Makefile.in*

*./configure --prefix=/usr --disable-static*

*make -j*

*sudo make install*

进入到/usr/include/，然后查看是否有json-c目录，进入到json-c目录查看，如图：



至此，说明json-c安装完成。安装完成之后，如果是64位系统，需要将/usr/lib/pkgconfig/下的json-c.pc拷贝到/usr/lib64/pkgconfig目录下。

## 3.1.4 安装libffi库

首先，下载libffi源码，下载路径为：<http://sourceware.org/libffi/>，注意，libffi的版本不能低于3.0.0，然后执行如下指令安装libffi，

*tar xvf libffi-3.2.1.tar.gz*

*cd libffi-3.2.1*

*./configure*

*make*

*sudo make install*

然后继续执行如下指令

yum install libffi-devel //centos系统下

apt-get install libffi-dev //ubuntu系统下

## 3.1.5安装Glib库

首先，下载Glib源码，下载路径为：<http://ftp.acc.umu.se/pub/GNOME/sources/glib/>，然后执行如下操作安装Glib库，

*xz -d glib-2.47.5.tar.gz*

*tar xvf* [*glib-2.47.5.tar.xz*](http://ftp.acc.umu.se/pub/GNOME/sources/glib/2.47/glib-2.47.5.tar.xz)

*cd glib-2.47.5*

*./configure*

*make*

*sudo make install*

然后继续执行如下指令

yum install glib-devel glib2-devel //centos系统下

apt-get install libglib2.0-dev //ubuntu系统下

## 3.1.6安装libevent库

首先，下载**libevent**源码，下载路径为：<http://libevent.org/>，然后执行如下操作安装libevent库，

*tar xvf* [*libevent-2.0.22-stable.tar.gz*](https://github.com/libevent/libevent/releases/download/release-2.0.22-stable/libevent-2.0.22-stable.tar.gz)

*cd libevent-2.0.22-stable*

*./configure*

*make*

*sudo make install*

然后继续执行如下指令

*yum install*  *libevent-devel* //centos系统下

*apt-get install*  *libevent-dev* //ubuntu系统下

## 3.1.7安装Thrift库

首先，下载Thrift源码，下载路径为：<http://thrift.apache.org/>，然后执行如下指令安装，

*tar xvf thrift-0.9.3.tar.gz*

*cd thrift-0.9.3*

*./configure*

*make*

*sudo make install*

## 3.1.8安装mysql-connector-mysql-connector-c++

下载mysql-connector-c++-1.1.6-linux-ubuntu14.04-x86-64bit.tar.gz,然后将该文件下的include和lib目录下的内容拷贝到/usr/local目录下对应的include和lib目录中。

## 执行编译和安装

执行该部分的前提是已经安装了MySQL，安装MySQL的过程此处不再叙述。

1、执行如下指令，解压源码

*tar xvf kamailio-4.3.4\_src.tar.gz*

2、执行如下指令进入源码

*cd kamailio-4.3.4*

1. 执行如下指令,生成modules.lst文件

*make FLAVOUR=kamailio cfg*

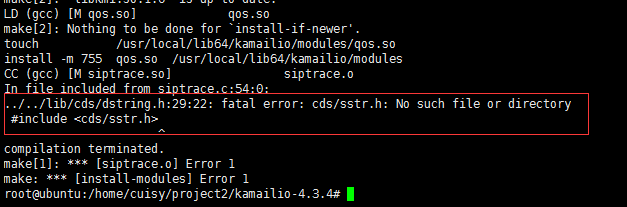
1. 用自己熟悉的编辑工具打开"modules.lst"文件，找到"exclude\_modules"，将等号后面的db\_mysql删除(这里可以根据自己的需要调整编译的模块)。然后执行如下指令编译

*make all*

1. 执行如下指令进行安装

*sudo make install*

执行该指令时，可能出现如下错误：

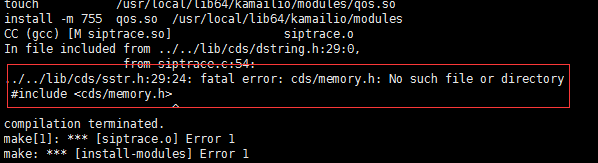


解决方法为：

*cd lib/cds*

*vim dstring.h +29*

然后，将#include <cds/sstr.h>改为#include “sstr.h”,然后重新编译，又会遇到如下错误，



解决方法为：

*cd lib/cds*

*vim sstr.h +29*

然后，将#include <cds/memory.h>改为#include “memory.h”,然后重新编译即可。

在该过程中，如果MySQL的版本是5.7以后的，可能会出现无法找到MySQL头文件的错误，解决方法为：

*mkdir /usr/include/mysql*

*sudo cp /usr/local/mysql/include/\* /usr/include/mysql*

*sudo mv /usr/include/mysql/mysql/\* /usr/include/mysql*

然后修改环境变量，解决方法为：

*sudo vim /etc/profile.d/mysql.sh*

然后，在mysql.sh中添加“*export PATH=$PATH: /usr/include*”,执行如下指令使其生效，

*source /etc/profile.d/mysql.sh*

*echo $PATH*

执行完“echo $PATH”之后，查看显示的环境变量中是否有刚添加的路径，如果有，则表示已经生效，否则，没有生效。

缺省的情况下，Kamailio的可执行文件是安装在”/usr/local/sbin”路径下的，配置文件是安装在”/usr/local/etc/kamailio”路径下的。另外，需要将/usr/local/sbin路径加到环境变量下才能保证Kamailio的指令生效，添加环境变量的方法为：

*sudo cd /etc/profile.d*

*sudo vim sbin.sh*

然后在sbin.sh中输入“*export PATH=$PATH:/usr/local/sbin*”，保存之后，执行如下指令，如果输出内容中有/usr/local/sbin，则证明添加的环境变量已生效。

*source sbin.sh*

*echo $PATH*

几个可执行文件功能如下：

kamdbctl：创建和管理数据库的脚本

kamctl： 管理和控制Kamailio服务器的脚本

# 配置

1、打开/usr/local/etc/kamailio/kamailio.cfg，添加下面几行（添加在最顶端即可）：

*#！define WITH\_MYSQL*

*#！define WITH\_AUTH*

*#！define WITH\_USRLOCDB*

*#！define WITH\_NAT*

*#！define WITH\_DEBUG*

选定db\_mysql模块安装。

1. 修改 /usr/local/etc/kamailio/kamctlrc

去掉SIP\_DOMAIN前的注释，并且修改成自己的服务器地址192.168.1.111（根据自己服务器的实际情况来写），我的是SIP\_DOMAIN=192.168.1.111。然后去掉DBENGINE=MYSQL前的注释语句，选定mysql数据库，其他的按照如下内容来修改：

*# cat /usr/local/etc/Kamailio/kamctlrc  
# $Id$  
# The Kamailio configuration file for the control tools.  
# Here you can set variables used in the kamctl and kamdbctl setup  
# scripts. Per default all variables here are commented out, the control tools  
# will use their internal default values.*

*## your SIP domain  
SIP\_DOMAIN=192.168.1.111*

*## chrooted directory*

*# $CHROOT\_DIR="/path/to/chrooted/directory"*

*## database type: MYSQL, PGSQL, ORACLE, DB\_BERKELEY, or DBTEXT, by default none is loaded*

*# If you want to setup a database with kamdbctl, you must at least specify  
# this parameter.  
DBENGINE=MYSQL*

*## database host  
DBHOST=localhost*

*## database name (for ORACLE this is TNS name)*

*DBNAME=kamailio*

*# database path used by dbtext or db\_berkeley  
# DB\_PATH="/usr/local/etc/kamailio/dbtext"*

*## database read/write user  
DBRWUSER=kamailio*

*## password for database read/write user  
DBRWPW="kamailiorw"*

*## database read only user  
# DBROUSER=kamailioro  
DBROUSER=kamailioro*

*## password for database read only user*

*DBROPW=kamailioro*

*## database super user (for ORACLE this is 'scheme-creator' user)  
DBROOTUSER="root"*

*# user name column  
USERCOL="username"*

*# SQL definitions*

*# If you change this definitions here, then you must change them  
# in db/schema/entities.xml too.  
# FIXME*

*# FOREVER="2020-05-28 21:32:15"  
# DEFAULT\_ALIASES\_EXPIRES=$FOREVER  
# DEFAULT\_Q="1.0"  
# DEFAULT\_CALLID="Default-Call-ID"  
# DEFAULT\_CSEQ="13"*

*# DEFAULT\_LOCATION\_EXPIRES=$FOREVER*

*# Program to calculate a message-digest fingerprint  
# MD5="md5sum"*

*# awk tool  
# AWK="awk"*

*# If you use a system with a grep and egrep that is not 100% gnu grep compatible,  
# e.g. solaris, install the gnu grep (ggrep) and specify this below.  
#*

*# grep tool  
# GREP="grep"*

*# egrep tool  
# EGREP="egrep"*

*# sed tool  
# SED="sed"*

*# tail tool  
# LAST\_LINE="tail -n 1"*

*# expr tool*

*# EXPR="expr"*

*# Describe what additional tables to install. Valid values for the variables  
# below are yes/no/ask. With ask (default) it will interactively ask the user  
# for an answer, while yes/no allow for automated, unassisted installs.  
#*

*# If to install tables for the modules in the EXTRA\_MODULES variable.  
# INSTALL\_EXTRA\_TABLES=ask*

*# If to install presence related tables.*

*# INSTALL\_PRESENCE\_TABLES=ask*

*# Define what module tables should be installed.*

*# If you use the postgres database and want to change the installed tables, then you  
# must also adjust the STANDARD\_TABLES or EXTRA\_TABLES variable accordingly in the  
# kamdbctl.base script.*

*# kamailio standard modules  
# STANDARD\_MODULES="standard acc lcr domain group permissions registrar usrloc msilo*

*# alias\_db uri\_db speeddial avpops auth\_db pdt dialog dispatcher  
# dialplan"*

*# Kamailio extra modules  
# EXTRA\_MODULES="imc cpl siptrace domainpolicy carrierroute userblacklist htable purple"*

*## type of aliases used: DB - database aliases; UL - usrloc aliases  
## - default: none*

*# ALIASES\_TYPE="DB"*

*## control engine: FIFO or UNIXSOCK  
## - default FIFO  
# CTLENGINE="FIFO"*

*## path to FIFO file  
# OSER\_FIFO="FIFO"*

*## check ACL names; default on (1); off (0)  
# VERIFY\_ACL=1*

*## ACL names - if VERIFY\_ACL is set, only the ACL names from below list  
## are accepted*

*# ACL\_GROUPS="local ld int voicemail free-pstn"*

*## verbose - debug purposes - default '0'  
VERBOSE=1*

*## do (1) or don't (0) store plaintext passwords  
## in the subscriber table - default '1'*

*# STORE\_PLAINTEXT\_PW=0*

*## OPENSER START Options  
## PID file path - default is: /var/run/Kamailio.pid  
# PID\_FILE=/var/run/kamailio/kamailio.pid  
PID\_FILE=/var/run/kamailio/kamailio.pid*

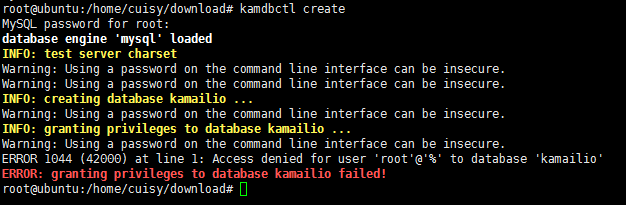
*## Extra start options - default is: not set  
# example: start kamailio with 64MB share memory: STARTOPTIONS="-m 64"  
# STARTOPTIONS=*

# 生成Kamailio的MySQL数据库

执行如下指令生成Kamailio的数据库：

*# /usr/local/sbin/kamdbctl create*

可能会遇到如下的问题

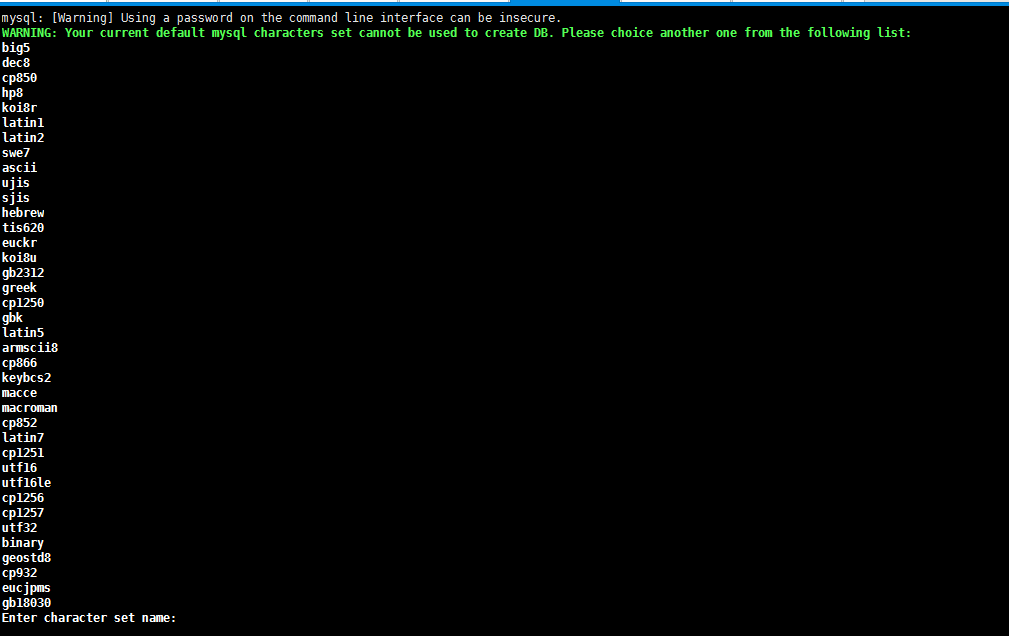


此时，执行“mysql -u用户名 -p密码”，进入数据库，然后执行下面语句来为用户授权

*grant all on \*.\* to 'root'@'%' identified by 'yourpassword';* （此指令也可以用来修改密码）

*grant all on \*.\* to 'root'@'localhost' identified by 'yourpassword';*

如果安装的是MySQL5.7版本，可能会出现如下界面提示：



输入“gbk”，然后回车即可。正常情况下，可以顺利创建kamailio数据库，若有其他问题，请自行百度解决。

# patch文件的使用

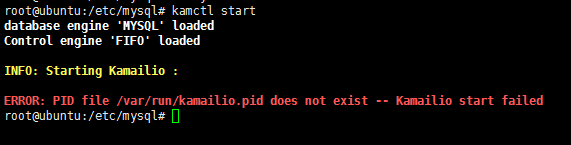
请参考《Kamailio源码之patch文件使用说明》文档，执行patch文件之后，需重新编译Kamailio源码或者单独编译修改的模块，并将生成的.so文件拷贝到/usr/local/lib64/kamailio/modules/路径下。

# 启动Kamailio服务

注意：启动Kamailio时必须切换到root模式，否则会启动失败，然后执行如下指令开启服务，

*# /usr/local/sbin/kamctl start*

可能会出现如下错误：

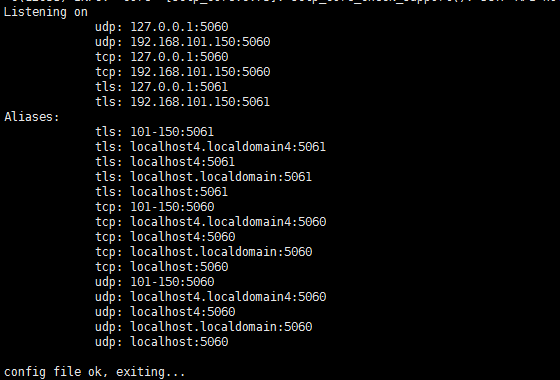


这种错误出现的原因是/var/run/路径下没有kamailio目录，手动创建一个即可，但不要生成kamailio.pid文件。

然后再开启服务试一下能否成功，如果还是不行，请按顺序检查如下内容：

（1）kamdbctlrc这个配置文件中的DBHOST处是否是127.0.0.1或者localhost，如果不是请将其修改成127.0.0.1或者localhost试试。

（2）执行“kamailio -c”来检测配置文件是否有错误，如果出现如下提示，



说明配置文件是正确的，如果有错误提示，请根据提示修改相应的错误。

（3）执行“kamailio -dddd”，看是否有ERROR输出，若有，请依据错误提示进行修改。

# 监视Kamailio

执行如下指令

*# /usr/local/sbin/kamctl moni*

# 注册用户

执行如下指令

*#kamctl add user1 123456  
#kamctl add user2 123456*

使用上面的命令添加两个用户 user1和user2 密码都是123456，添加完毕后，配置你的SIP终端注册。

# 测试Kamailio基本服务

（1）使用X-Lite测试

用于测试的软件有好多，在此以X-Lite为例。

测试需要一个服务器和两个客户端，在实验室的局域网下完成，我的linux装在虚拟机上，所以再需要局域网内的一台机器就够了，借用同学的机器A。在A和我的主机win环境分别下载安装x-lite软电话软件，使用用户名1000和1001登陆，服务器地址写为192.168.1.111（虚拟机ip地址）。通过putty登陆虚拟机，输入kamctl start,启动sip服务器。  
 SIP服务器启动后，可以发现A和我的主机下的x-lite可以登陆到服务器，在我的主机下输入1001可以呼叫A下的x-lite，接听后可完成通话

X-Lite的配置如图：



以上就是关于Kamailio的安装和配置的基本步骤。

（2）使用MicroSIP测试

MicroSIP是一个免费开源的SIP网络电话客户端（PC端），通过SIP协议可实现高品质的VoIP电话，有语音通话、视频通话、短信、通讯录等功能。使用MicroSIP进行测试的操作方法如下：

首次打开MicroSIP时的界面如下，



然后，点击“菜单”🡪“添加账户”，出现如下界面，



下面以SIP账号1000为例进行配置，配置详情如下，



SIP服务器：输入你使用的SIP服务器的IP地址

用户名：你使用的SIP账号

主机：输入你使用的SIP服务器的IP地址（与SIP服务器一项相同）

密码：你使用的SIP账号的密码

OK，输入完毕之后，点击“保存”，正常的话，应该出现如下界面，

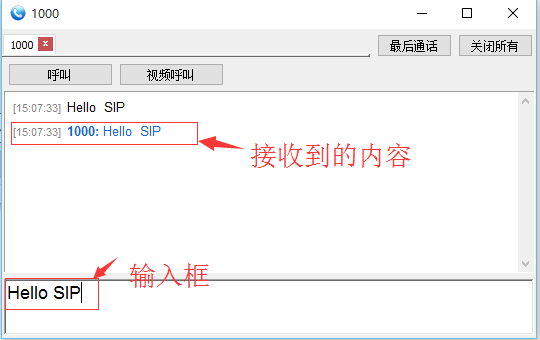


当左下角出现如图所示的“在线”时，说明账号1000登录成功。

登录成功之后，接下来进行Kamailio基本通讯的测试，如下图，在输入框中输入“1000”



然后点击标识2处的发送短信按钮，弹出如下界面，



在输入框中输入“Hello SIP”，然后点击“回车键”，如果能出现上图所示的现象，则说明Kamailio基本通讯正常。

至此，Kamailio配置和基本测试完成。

# 同一台服务器运行多套kamailio服务

按照上面的内容进行配置和测试，只是可以实现在一台服务器上运行一套Kamailio服务，但实际情况中有时需要在同一台服务器上部署多套Kamailio服务，并且要求使用的配置文件、动态库和数据库都相互独立，那么针对这种需求该如何安装和配置Kamailio呢？下面将详细的介绍该需求的实现方法。

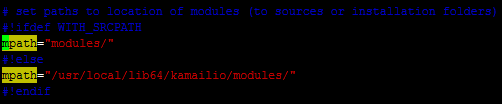
## 配置文件强化说明

在使用Kamailio的过程中，主要使用到的配置文件有两个：kamailio.cfg和kamctlrc。首先介绍一下kamctlrc，该文件的作用是用来生成Kamailio的数据库的，至于其中的基本配置已经在第4部分说过了，在此只选择其中部分内容做

SIP\_DOMAIN，用于指定服务器的IP；DBHOST，指定使用的数据库的具体IP；DBNAME,指定生成的Kamailio的数据库名称；PID\_FILE，指定生成的Kamailio的pid文件的路径以及名称。

然后介绍一下kamailio.cfg文件，该配置文件包含的内容比较多，Kamailio所有功能的配置都在此文件中设置，下面也是选择部分内容进行讲解。

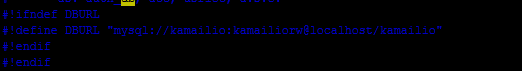
Kamailio一共有201个模块，Kamailio默认是到/usr/local/lib64/kamailio/modules路径下去加载模块的动态库，如果想让Kamailio到自定义的路径下加载，需要修改kamailio.cfg，打开kamailio.cfg，添加如下内容：



然后，将mpath后面的路径改成自定义的路径就可以了。

要想使用某个模块，首先需要在kamailio.cfg的顶部添加“#!define WITH\_\*\*\*”，其中星号代表模块的名称。然后，使用loadmodule “\*\*\*.so”加载模块的动态库，使用modparam配置模块的具体参数。

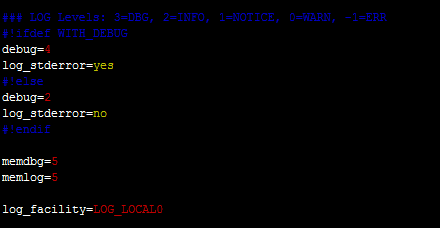
上面已经说了可以通过kamctlrc中的DBHOST和DBNAME来生成不同的Kamailio数据库，但生成之后该如何指定配置文件读取哪个数据库呢？OK，在kamailio.cfg中有这么一部分内容：



所以，Kamailio是通过设置DBURL来指定使用哪个数据库。下面分析一下“mysql://kamailio:kamailiorw@localhost/kamailio”这句话，mysql表示使用的是MySQL数据库，kamailio和kamailiorw是我们在kamctlrc中设置的kamailio数据库中具有可读可写权限的用户的名称和密码，localhost对应DBHOST，最后面那个kamailio就是你创建的Kamailio数据库的名称，对应kamctlrc中的DBNAME。

Kamailio默认使用的端口为5060，也可以自定义端口号，只需要修改kamailio.cfg文件中的port值即可，其实只需要将kamailio.cfg中的“5060”全部替换成自定义的端口号即可。

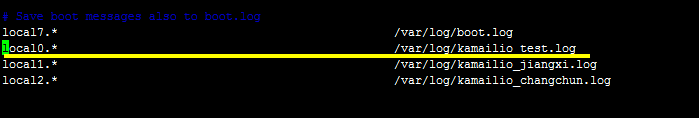
对于服务器，日志是不可缺少的，kamailio.cfg中同样可以设置日志的输出和存放路径。打开kamailio.cfg，会看到如下内容：



可以指定输出的日志的级别，也可以指定是否输出日志，其中LOG\_LOCAL0用于确定日志输出的路径，需要修改/etc/rsyslog.conf中的内容，方法为：

*vim /etc/rsyslog.conf*

然后，在rsyslog.conf中添加如下内容：



即设置kamailio生成日志的路径为/var/log，日志文件名称为kamailio\_test.log，然后执行如下指令使修改的内容生效：

*service rsyslog restart*

为了便于解析网络的IP，服务器会使用域名来代替IP地址，Kamailio同样支持域名的解析，打开kamailio.cfg，会看到如下内容：



kamailio.cfg默认是将域名的功能注释掉的，此处只需要将注释去掉，然后将“sip.mydomain.com”改为我们自己的服务器域名即可。要想使得域名生效，需要加载alias模块，模块加载的方法按照官方文档照抄即可。

OK，有了上面内容做基础，下面我们来进行具体的编译、安装和配置的操作。

## 指定路径编译安装Kamailio

Kamailio的默认安装路径是/usr/local/etc/和/usr/local/lib64，要想实现同一台服务器运行多套Kamailio服务，首先需要做的就是指定不同的安装路径，执行如下指令：

*make PREFIX="/usr/local/kamailio\_test" include\_modules="db\_mysql" cfg*

*make all*

*make install*

从而指定了Kamailio的安装路径，本文以*/usr/local/kamailio\_test*为例进行说明，然后进入到*/usr/local/kamailio\_test*路径下，会看到 etc、lib64、sbin和share这四个目录，说明安装路径正确。

## 生成自定义的Kamailio数据库

有了前面的内容做基础，我们知道如何自定义kamctlrc文件，而且需要特别注意的是，必须先配置kamctlrc文件，然后再执行指令生成Kamailio数据库。我们假设kamctlrc中的

DBHOST=127.0.0.1 ，DBNAME=kamailio\_test，

PID\_FILE=/var/run/kamailio\_test/kamailio\_test.pid，然后执行如下指令生成Kamailio的数据库

*/usr/local/kamailio\_test/sbin/kamdbctl create*

注意，由于我们自定义的安装路径，导致Kamailio的可执行文件没有被拷贝到系统路径下，所以需要通过绝对路径来指定使用哪套可执行文件，接下来一路输入“y”，按回车即可。进入到MySQL，看一下是否有kamailio\_test数据库生成，如果有，则说明kamctlrc生效了，操作成功。

## 配置kamailio.cfg

### 指定使用的数据库

按照11.1的介绍，首先修改DBURL，使Kamailio使用我们自定义的数据库，具体的修改内容为：

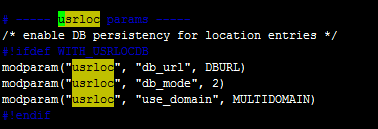
*#!ifndef DBURL*

*#!define DBURL "mysql://kamailio:kamailiorw@localhost/kamailio\_test"*

*#!endif*

*#!endif*

然后，usrloc模块的配置改成如下图所示：



如果不修改usrloc模块的话，创建的SIP账号将无法登陆。

### 更改端口

Kamailio默认的端口为5060，为了运行多套Kamailio，需要自定义端口号，修改方法为：打开kamailio.cfg文件，然后将其中的5060全部改为5061（假设自定义端口为5061），然后重新启动Kamailio，启动时同样使用绝对路径，然后执行“ss -tlunp”查看5061端口是否已经启动，如果已启动，说明配置生效。

OK，通过上面的操作，就可以实现同一台服务器运行多套相对独立的Kamailio服务了。