YunParking
停车服务安装部署文档

V1.0

#2023#

By

深圳市云创智城科技有限公司

1. 安装注意事项

切记～先在 Typora 的「主题」菜单中，选择 VLOOK™ 的主题（所有以 `Vlook {XX名称}` 形式命名的主题）。

1.1 lrzsz

- 上传与下载文件的命令

```
yum -y install lrzsz
```

2. Nginx

yum install nginx

3. JDK

- JDK 安装版本1.8.0_345

yum install java

```
[root@node3 laiting]# yum install java
Last metadata expiration check: 2:57:38 ago on Mon 31 Oct 2022 06:25:02 AM CST.
Package java-1.8.0-openjdk-1:1.8.0.345.b01-1.al8.x86_64 is already installed.
Dependencies resolved.
================================================================================
================================================================================
 Package                                                 Architecture
                    Version
Repository                                  Size
================================================================================
================================================================================
Upgrading:
 java-1.8.0-openjdk                                      x86_64
                    1:1.8.0.352.b08-2.al8
alinux3-updates                             350 k
 java-1.8.0-openjdk-headless                             x86_64
                    1:1.8.0.352.b08-2.al8
alinux3-updates                              34 M
 tzdata-java                                             noarch
                    2022e-1.1.al8
alinux3-updates                             186 k

Transaction Summary
================================================================================
================================================================================
Upgrade  3 Packages

Total download size: 35 M
Is this ok [y/N]: y
```

```
Downloading Packages:
(1/3): tzdata-java-2022e-1.1.al8.noarch.rpm

                    10 MB/s | 186 kB     00:00
(2/3): java-1.8.0-openjdk-1.8.0.352.b08-2.al8.x86_64.rpm

                   13 MB/s | 350 kB     00:00
(3/3): java-1.8.0-openjdk-headless-1.8.0.352.b08-2.al8.x86_64.rpm

                    95 MB/s |  34 MB     00:00
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------
Total

                    96 MB/s |  35 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Running scriptlet: java-1.8.0-openjdk-headless-1:1.8.0.352.b08-
2.al8.x86_64
                                                              1/1
  Preparing        :

                                             1/1
  Upgrading        : tzdata-java-2022e-1.1.al8.noarch

                                              1/6
  Upgrading        : java-1.8.0-openjdk-headless-1:1.8.0.352.b08-
2.al8.x86_64
                                                    2/6
warning: /etc/java/java-1.8.0-openjdk/java-1.8.0-openjdk-1.8.0.352.b08-
2.al8.x86_64/lib/security/java.policy created as /etc/java/java-1.8.0-
openjdk/java-1.8.0-openjdk-1.8.0.352.b08-
2.al8.x86_64/lib/security/java.policy.rpmnew

```

```
  Running scriptlet: java-1.8.0-openjdk-headless-1:1.8.0.352.b08-
2.al8.x86_64
                                                              2/6
restored /etc/java/java-1.8.0-openjdk/java-1.8.0-openjdk-1.8.0.352.b08-
2.al8.x86_64/lib/security/java.policy.rpmnew to /etc/java/java-1.8.0-
openjdk/java-1.8.0-openjdk-1.8.0.352.b08-
2.al8.x86_64/lib/security/java.policy

  Upgrading        : java-1.8.0-openjdk-1:1.8.0.352.b08-2.al8.x86_64

                                                              3/6
  Running scriptlet: java-1.8.0-openjdk-1:1.8.0.352.b08-2.al8.x86_64

                                                              3/6
  Cleanup          : java-1.8.0-openjdk-1:1.8.0.345.b01-1.al8.x86_64

                                                              4/6
  Running scriptlet: java-1.8.0-openjdk-1:1.8.0.345.b01-1.al8.x86_64

                                                              4/6
  Cleanup          : java-1.8.0-openjdk-headless-1:1.8.0.345.b01-
1.al8.x86_64
                                                              5/6
  Running scriptlet: java-1.8.0-openjdk-headless-1:1.8.0.345.b01-
1.al8.x86_64
                                                              5/6
  Cleanup          : tzdata-java-2022c-1.al8.noarch

                                                              6/6
  Running scriptlet: java-1.8.0-openjdk-headless-1:1.8.0.352.b08-
2.al8.x86_64
                                                              6/6
  Running scriptlet: java-1.8.0-openjdk-1:1.8.0.352.b08-2.al8.x86_64

                                                              6/6
  Running scriptlet: tzdata-java-2022c-1.al8.noarch

                                                              6/6
```

```
  Verifying        : java-1.8.0-openjdk-1:1.8.0.352.b08-2.al8.x86_64

                                                    1/6
  Verifying        : java-1.8.0-openjdk-1:1.8.0.345.b01-1.al8.x86_64

                                                    2/6
  Verifying        : java-1.8.0-openjdk-headless-1:1.8.0.352.b08-
2.al8.x86_64
                                                    3/6
  Verifying        : java-1.8.0-openjdk-headless-1:1.8.0.345.b01-
1.al8.x86_64
                                                    4/6
  Verifying        : tzdata-java-2022e-1.1.al8.noarch

                                                    5/6
  Verifying        : tzdata-java-2022c-1.al8.noarch

                                                    6/6

Upgraded:
  java-1.8.0-openjdk-1:1.8.0.352.b08-2.al8.x86_64
java-1.8.0-openjdk-headless-1:1.8.0.352.b08-2.al8.x86_64
  tzdata-java-2022e-1.1.al8.noarch

Complete!
```

```
java -version
openjdk version "1.8.0_345"
OpenJDK Runtime Environment (build 1.8.0_345-b01)
OpenJDK 64-Bit Server VM (build 25.345-b01, mixed mode)
```

4. MySql

```
[root@swzhtc laiting]# cd /data/
```

- 下载mysql-5.7.38-linux-glibc2.12-x86_64.tar.gz

```
[root@swzhtc data]# wget https://downloads.mysql.com/archives/get/p/23/file/mysql-5.7.38-linux-glibc2.12-x86_64.tar.gz
--2022-10-31 09:30:48-- https://downloads.mysql.com/archives/get/p/23/file/mysql-5.7.38-linux-glibc2.12-x86_64.tar.gz
Resolving downloads.mysql.com (downloads.mysql.com)... 69.192.12.46, 2600:1417:e800:189::2e31, 2600:1417:e800:18a::2e31
Connecting to downloads.mysql.com (downloads.mysql.com)|69.192.12.46|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://cdn.mysql.com/archives/mysql-5.7/mysql-5.7.38-linux-glibc2.12-x86_64.tar.gz [following]
--2022-10-31 09:30:48--  https://cdn.mysql.com/archives/mysql-5.7/mysql-5.7.38-linux-glibc2.12-x86_64.tar.gz
Resolving cdn.mysql.com (cdn.mysql.com)... 173.222.228.218
Connecting to cdn.mysql.com (cdn.mysql.com)|173.222.228.218|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 674830866 (644M) [application/x-tar-gz]
Saving to: ‘mysql-5.7.38-linux-glibc2.12-x86_64.tar.gz’

mysql-5.7.38-linux-glibc2.12-x86_64.tar.gz       100% [==========================================================================================>] 643.57M  18.1MB/s    in 57s

2022-10-31 09:31:46 (11.4 MB/s) - ‘mysql-5.7.38-linux-glibc2.12-x86_64.tar.gz’ saved [674830866/674830866]
```

- 解压安装包

```
[root@swzhtc data]# tar -zxvf mysql-5.7.38-linux-glibc2.12-x86_64.tar.gz
```

- 重名名为mysql

```
[root@swzhtc data]# mv mysql-5.7.38-linux-glibc2.12-x86_64 mysql
```

- 将mysql 拷贝到/usr/local/位置下

```
[root@swzhtc data]# cp -R mysql /usr/local/
[root@swzhtc data]# cd /usr/local/mysql/
```

- 添加mysql用户组并授权权限

```
[root@swzhtc data]# chown mysql:mysql -R /data/mysql
[root@swzhtc data]# groupadd mysql
[root@swzhtc data]# useradd -r -g mysql mysql

```

- 创建my.conf文件

```
[root@swzhtc data]# vim /etc/my.cnf
```

- 查看my.conf配置文件

```
[root@swzhtc bin]# cat /etc/my.cnf
```

```
[client]
socket=/tmp/mysql.sock
default-character-set=utf8mb4

[mysql]
default-character-set=utf8mb4

[mysqld]

# Remove leading # and set to the amount of RAM for the most important data
# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.
# innodb_buffer_pool_size = 128M
#
# Remove leading # to turn on a very important data integrity option: logging
# changes to the binary log between backups.
# log_bin
#
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
```

```
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
#datadir=/var/lib/mysql
#socket=/var/lib/mysql/mysql.sock
#修改后
basedir=/usr/local/mysql
datadir=/data/mysql
socket=/tmp/mysql.sock
pid-file=/data/mysql/mysql.pid
character-set-server=utf8mb4
collation-server=utf8mb4_unicode_ci
init_connect='SET NAMES utf8mb4'
sql_mode=STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION
#

slow_query_log=1 #开启慢日志
slow_query_log_file=/data/mysql/logs/slow_query_log.txt
#查询超过10秒就记录
long_query_time=5 #查询超过十秒就记录

# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

#binlog配置
log-bin=mysql-bin # 开启 binlog
binlog-format=ROW # 选择 ROW 模式
server_id=1 # 配置 MySQL

log-error=/data/mysql/mysql.err
pid-file=/data/mysql/mysql.pid

explicit_defaults_for_timestamp=1
```

- 授权配置文件

```
[root@swzhtc data]# chmod 644 /etc/my.cnf
```

- 创建Mysql数据存放目录

```
[root@swzhtc data]# mkdir -p  /data/mysql
```

- 执行安装

```
[root@swzhtc laiting]# cd /usr/local/mysql/bin/
```

```
./mysqld --defaults-file=/etc/my.cnf --basedir=/usr/local/mysql/ --
datadir=/data/mysql/ --user=mysql --initialize
```

> 安装错误：
>
> [root@swzhtc bin]# ./mysqld --defaults-file=/etc/my.cnf --basedir=/usr/local/mysql/ --datadir=/data/mysql/ --user=mysql --initialize
> ./mysqld: error while loading shared libraries: libaio.so.1: cannot open shared object file: No such file or directory

解决办法：yum install -y libaio

```
[root@swzhtc bin]# ./mysqld --defaults-file=/etc/my.cnf --
basedir=/usr/local/mysql/ --datadir=/data/mysql/ --user=mysql --
initialize
./mysqld: error while loading shared libraries: libaio.so.1: cannot open
shared object file: No such file or directory
[root@swzhtc bin]# yum install -y libaio
Last metadata expiration check: 0:53:09 ago on Mon 31 Oct 2022 08:50:23
AM CST.
Dependencies resolved.
================================================================================
================================================================================
================================================
 Package                                 Architecture
             Version
Repository                                   Size
================================================================================
================================================================================
================================================
Installing:
```

```
 libaio                                      x86_64
              0.3.112-1.2.al8
alinux3-os                                     33 k

Transaction Summary
================================================================================
================================================================================
================================================
Install  1 Package

Total download size: 33 k
Installed size: 103 k
Downloading Packages:
libaio-0.3.112-1.2.al8.x86_64.rpm

                   967 kB/s |  33 kB     00:00
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
------------------------------------------------
Total

                   948 kB/s |  33 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :

                                                1/1
  Installing       : libaio-0.3.112-1.2.al8.x86_64

                                                1/1
  Running scriptlet: libaio-0.3.112-1.2.al8.x86_64

                                                1/1
  Verifying        : libaio-0.3.112-1.2.al8.x86_64

                                                1/1

```

```
Installed:
  libaio-0.3.112-1.2.al8.x86_64




Complete!

```

- 安装成功后，查询临时密码，在最后一行： A temporary password is generated for root@localhost: **ivdaFY;,1Cl**

```
[root@swzhtc bin]# cat /data/mysql/mysql.err
2022-10-31T02:16:56.548385Z 0 [Warning] 'NO_AUTO_CREATE_USER' sql mode was not set.
2022-10-31T02:16:56.768522Z 0 [Warning] InnoDB: New log files created, LSN=45790
2022-10-31T02:16:56.802557Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
mysqld: File '/data/mysql/logs/slow_query_log.txt' not found (Errcode: 2 - No such file or directory)
2022-10-31T02:16:56.857618Z 0 [ERROR] Could not use /data/mysql/logs/slow_query_log.txt for logging (error 2 - No such file or directory). Turning logging off for the server process. To turn it on again: fix the cause, then either restart the query logging by using "SET GLOBAL SLOW_QUERY_LOG=ON" or restart the MySQL server.
2022-10-31T02:16:56.860358Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: 180aa393-58c2-11ed-8cf9-00163e0ab541.
2022-10-31T02:16:56.861047Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
2022-10-31T02:16:56.979316Z 0 [Warning] A deprecated TLS version TLSv1 is enabled. Please use TLSv1.2 or higher.
2022-10-31T02:16:56.979326Z 0 [Warning] A deprecated TLS version TLSv1.1 is enabled. Please use TLSv1.2 or higher.
2022-10-31T02:16:56.979650Z 0 [Warning] CA certificate ca.pem is self signed.
2022-10-31T02:16:57.054539Z 1 [Note] A temporary password is generated for root@localhost: ivdaFY;,1Clt

```

- 先将mysql.server移动到/etc/init.d/mysql中

```
cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysql
```

- 然后启动mysql，并查看mysql状态

```
#启动mysql
service mysql start
#查看mysql运行状态
service mysql status
ps -ef|grep mysql
```

由于当前并没有全局设置mysql命令，所以先切换到/usr/local/mysql/bin/文件夹下

```
[root@swzhtc bin]# cd /usr/local/mysql/bin/
```

登陆MySql

[root@swzhtc bin]# ./mysql -uroot -pPASSWORD

请将PASSWORD替换成临时密码，或者直接执行：./mysql -uroot -p 在提示Enter password:后将临时密码粘贴，回车即可~

- 然后再执行下面的操作设置密码(Ltkj@2022@mysql)

```
SET PASSWORD = PASSWORD('newPassword');
ALTER USER 'root'@'localhost' PASSWORD EXPIRE NEVER;
FLUSH PRIVILEGES;
use mysql;
```

使root用户可以在任何IP访问

```
update user set host = '%' where user = 'root';
```

刷新

```
FLUSH PRIVILEGES;
```

建立软连接

```
ln -s /usr/local/mysql/bin/mysql /usr/bin/
```

设置mysql开机启动

设置mysql开机自启动

```
chkconfig mysql on
```

取消mysql开机自启动

```
chkconfig mysql off
```

查看系统服务列表，以及每个服务的运行级别

```
chkconfig --list
```

或

```
systemctl list-unit-file
```

操作记录如下：

```

[root@swzhtc bin]# ./mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.38-log

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

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

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

```
mysql>
mysql> SET PASSWORD = PASSWORD('newPassword');
Query OK, 0 rows affected, 1 warning (0.00 sec)


mysql> ALTER USER 'root'@'localhost' PASSWORD EXPIRE NEVER;
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> update user set host = '%' where user = 'root';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye

```

- **linux下mysql怎么修改端口号**
- 编辑 **vim /etc/my.cnf** 文件，找到mysql配置文件my.cnf的port这一行，把之前的3306端口修改为自己想要的就行

```
vim /etc/my.cnf
```

```
[mysqld]
port=13306
```

修改完成后执行：**service mysql restart**

```
service mysql restart
```

linux下mysql开启远程访问并开启3306端口

一、登陆mysql

```
mysql -u root -p
```

二、设置访问地址

```
如果你想允许用户root从ip为192.168.1.123的主机连接到mysql服务器，并使用root作为密码
GRANT ALL PRIVILEGES ON *.* TO 'root'@'192.168.1.123'IDENTIFIED BY 'password' WITH GRANT OPTION;
```

三、刷新

```
flush privileges;
```

防火墙开启

四、开启端口3306

```
firewall-cmd --zone=public --add-port=3306/tcp --permanent
```

五、重启防火墙

```
firewall-cmd --reload
```

六、查看已经开放的端口

```
firewall-cmd --list-ports
```

5. Redis

5.1 linux 下安装redis并设置开机自启动

5.2 一. 下载并解压

5.2.1 1. 执行命令:

```
wget https://download.redis.io/releases/redis-6.2.6.tar.gz
```

5.2.2 2.解压redis:

```
tar xzf redis-6.2.6.tar.gz
```

5.2.3 3. 修改目录

```
mv redis-6.2.6 /usr/local/redis
```

5.3 二. 编译

- 进入redis安装目录，执行make命令编译redis

```
cd /usr/local/redis

make
```

错误解决:

如果执行make命令报错：cc 未找到命令，原因是虚拟机系统中缺少gcc，执行下面命令安装gcc

```
yum -y install gcc automake autoconf libtool make
```

如果执行make命令报错：致命错误:jemalloc/jemalloc.h: 没有那个文件或目录，则需要在make指定分配器为libc。执行下面命令即可正常编译:

```
make MALLOC=libc
```

- 执行下面命令安装redis，并指定安装目录

```
make install PREFIX=/usr/local/redis
```

- 启动redis
- 启动命令：

```
./bin/redis-server redis.conf
```

详细指令参考：

```
[root@swzhtc redis]# pwd
/usr/local/redis
[root@swzhtc redis]# mkdir /etc/redis
[root@swzhtc redis]# cp -r redis.conf /etc/redis/6379.conf
[root@swzhtc redis]# cp utils/redis_init_script /etc/init.d/redis
[root@swzhtc redis]# vim /etc/redis/6379.conf

```

6379.conf配置文件修改相关参数

```
vim /etc/redis/6379.conf
```

- **1、修改配置文件支持后台启动**
- 打开redis.conf 将daemonize no改为daemonize yes即可

```
daemonize yes
```

- **2、支持远程连接**
- bind 127.0.0.1 -::1注释掉

```
bind 127.0.0.1 -::1
```

- 把protected-mode yes改为protected-mode no即可

```
protected-mode no
```

- **3、设置redis连接密码**

修改redis.conf配置文件

```
# requirepass foobared
requirepass 123   指定密码123
```

- 保存后重启redis就可以了
- redis-cli连接redis

```
[root@iZ2ze3zda3caeyx6pn7c5zZ bin]# ./redis-cli
127.0.0.1:6379> keys *
(error) NOAUTH Authentication required.
127.0.0.1:6379> auth 123        //指定密码
OK
127.0.0.1:6379> keys *
1) "a"
2) "cit"
3) "clist"
4) "1"
127.0.0.1:6379>
```

- 设置开机自启动
- 复制配置文件 redis.conf /etc/redis/ ，改名6379.conf

```
cp -r redis.conf /etc/redis/6379.conf
```

- 复制配置文件

```
cp utils/redis_init_script /etc/init.d/redis
```

- 修改文件位置以下两行，修改为自己实际安装位置

EXEC=/usr/local/redis/bin/redis-server
CLIEXEC=/usr/local/redis/bin/redis-cli

```
#!/bin/sh
```

```
#
# Simple Redis init.d script conceived to work on Linux systems
# as it does use of the /proc filesystem.

### BEGIN INIT INFO
# Provides:     redis_6379
# Default-Start:        2 3 4 5
# Default-Stop:         0 1 6
# Short-Description:    Redis data structure server
# Description:          Redis data structure server. See
https://redis.io
### END INIT INFO

REDISPORT=6379
EXEC=/usr/local/redis/bin/redis-server
CLIEXEC=/usr/local/redis/bin/redis-cli

PIDFILE=/var/run/redis_${REDISPORT}.pid
CONF="/etc/redis/${REDISPORT}.conf"

case "$1" in
    start)
        if [ -f $PIDFILE ]
        then
                echo "$PIDFILE exists, process is already running or
crashed"
        else
                echo "Starting Redis server..."
                $EXEC $CONF
        fi
        ;;
    stop)
        if [ ! -f $PIDFILE ]
        then
                echo "$PIDFILE does not exist, process is not running"
        else
                PID=$(cat $PIDFILE)
                echo "Stopping ..."
                $CLIEXEC -p $REDISPORT shutdown
                while [ -x /proc/${PID} ]
```

```
                do
                    echo "Waiting for Redis to shutdown ..."
                    sleep 1
                done
                echo "Redis stopped"
        fi
        ;;
    *)
        echo "Please use start or stop as first argument"
        ;;
esac
```

- 修改文件权限

```
chmod 777 /etc/init.d/redis
```

- 设为开机启动

```
chkconfig redis on
```

- 启动/停止服务

```
  service redis start

  service redis stop
```

或者用如下指令启动或查看启动状态

```
[root@swzhtc redis]# systemctl restart redis
[root@swzhtc redis]# systemctl status redis
● redis.service - LSB: Redis data structure server
   Loaded: loaded (/etc/rc.d/init.d/redis; generated)
   Active: active (running) since Mon 2022-10-31 11:30:36 CST; 7s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 63920 ExecStart=/etc/rc.d/init.d/redis start (code=exited, status=0/SUCCESS)
    Tasks: 4 (limit: 200229)
   Memory: 1.0M
   CGroup: /system.slice/redis.service
           └─63922 /usr/local/redis/bin/redis-server 127.0.0.1:6379
```

```
12
13  Oct 31 11:30:36 swzhtc systemd[1]: Starting LSB: Redis data structure
    server...
14  Oct 31 11:30:36 swzhtc redis[63920]: Starting Redis server...
15  Oct 31 11:30:36 swzhtc systemd[1]: Started LSB: Redis data structure
    server.
16
```

6. RabbitMQ

6.1 教程一

安装流程：

- 进入root文件夹下面，新建文件夹名为mq

```
1  [root]# cd /data
2  [root]# mkdir mq root]# cd mq
```

- 准备环境：

```
1  [root]# yum -y install make gcc gcc-c++ kernel-devel m4 ncurses-devel
   openssl-devel
```

- 源码下载

```
1  wget https://github.com/erlang/otp/releases/download/OTP-
   21.3/otp_src_21.3.tar.gz
```

- 将'otp_src_21.3.tar.gz'安装包拖入到mq文件夹下面，并解压

```
1  [root]# tar xzf otp_src_21.3.tar.gz
```

- 进入目录：

```
[root]# cd otp_src_21.3
```

- 设定安装规则：

```
[root]# ./configure --prefix=/data/mq/erlang --with-ssl --enable-threads
--enable-smp-support --enable-kernel-poll --enable-hipe --without-javac
```

- 安装：

```
make && make install
```

- 此过程有点久。大概3分钟。。
- 编辑配置文件

```
[root]# vim /etc/profile
```

- 在末尾加入以下内容：

```
#set erlang environment
ERL_PATH=/data/mq/erlang/bin
PATH=$ERL_PATH:$PATH
[root]# source /etc/profile
```

使环境变量立即生效 最后输入*erl*，验证是否安装成功（出现erlang版本号即成功）

安装rabbitMQ 下载rabbitmq-server安装包

在/data/mq/文件架下面新建rabbitmq文件夹 把rabbitmq-server-generic-unix-3.7.7.tar.xz放到该目录下面，并解压

```
wget https://github.com/rabbitmq/rabbitmq-
server/releases/download/v3.7.7/rabbitmq-server-generic-unix-3.7.7.tar.xz

[root]# tar xvf rabbitmq-server-generic-unix-3.7.7.tar.xz
```

改名

```
[root]# mv rabbitmq_server-3.7.7/ rabbitmq/
```

设置环境变量

```
[root]#  vim /etc/profile
```

在末尾加入以下内容：

```
#set RabbitMQ environment
MQ_PATH=/data/mq/rabbitmq/sbin
PATH=$MQ_PATH:$PATH

[root]# source /etc/profile 使环境变量立即生效
```

进入到rabbimq/sbin目录下。 后台启动rabbitmq：

```
[root]#  ./rabbitmq-server -detached
```

通过查看服务状态

```
[root]#  service rabbitmq-server status
```

开启管理界面

```
[root]#  ./rabbitmq-plugins  list
[root]#  ./rabbitmq-plugins enable rabbitmq_management
[root]#  ./rabbitmqctl add_user rabbitmq rabbitmq
[root]#  ./rabbitmqctl set_user_tags rabbitmq administrator
```

-------重要-------

创建用户，您将需要为RabbitMQ Web管理控制台创建管理用户。 运行以下命令相同

```
./rabbitmqctl add_user admin StrongPassword
./rabbitmqctl set_user_tags admin administrator
./rabbitmqctl set_permissions -p / admin “.*” “.*” “.*”
```

将管理员更改为管理员用户的首选用户名。 确保将StrongPassword更改为非常强大的密码

firewall-cmd --zone=public --add-port=15672/tcp --permanent firewall-cmd --zone=public --add-port=5672/tcp --permanent firewall-cmd --reload

服务器需要开启15672端口与5672端口访问 访问web端： http://ip:15672

6.2 教程二

6.2.1 手工部署RabbitMQ（CentOS 7.4）

6.2.1.1 简介

本文介绍了如何在华为云上使用弹性云服务器的Linux实例部署RabbitMQ。RabbitMQ是采用Erlang语言实现AMQP（Advanced Message Queuing Protocol，高级消息队列协议）的消息中间件，它最初起源于金融系统，用于在分布式系统中存储转发消息。RabbitMQ凭借其高可靠、易扩展、高可用及丰富的功能特性成为目前非常热门的一款消息中间件。

6.2.1.2 前提条件

弹性云服务器所在安全组添加了如下表所示的安全组规则，具体步骤参见[为安全组添加安全组规则]。

方向	类型	协议	端口/范围	源地址
入方向	IPv4	TCP	5672	0.0.0.0/0
入方向	IPv4	TCP	15672	0.0.0.0/0

6.2.1.3 操作步骤

1. 安装相关依赖包和perl。

   a. 登录弹性云服务器。

   b. 执行以下命令，安装相关依赖包。
   yum -y install make gcc gcc-c++ m4 ncurses-devel openssl-devel unixODBC-devel

   c. 执行如下命令，安装perl。
   yum install perl

2. 安装erlang。

关于erlang的安装请参考Erlang官方资料。

a. 添加erlang存储库到系统

wget https://packages.erlang-solutions.com/erlang-solutions-2.0-1.noarch.rpm

rpm -Uvh erlang-solutions-2.0-1.noarch.rpm

或手动添加存储库条目

rpm --import https://packages.erlang-solutions.com/rpm/erlang_solutions.asc

b. 在/etc/yum.repos.d/目录下新建一个文件rabbitmq-erlang.repo，然后将下面的粘帖进去

cd /etc/yum.repos.d/

vi rabbitmq-erlang.repo

```
[erlang-solutions]
name=CentOS $releasever - $basearch - Erlang Solutions
baseurl=https://packages.erlang-solutions.com/rpm/centos/$releasever/$basearch
gpgcheck=1
gpgkey=https://packages.erlang-solutions.com/rpm/erlang_solutions.asc
enabled=1
```

按**Esc**键退出编辑模式，并输入**:wq**保存后退出。

c. 执行以下命令安装erlang

sudo yum install erlang

执行以下命令安装esl-erlang

sudo yum install esl-erlang

d. 执行如下命令，检查安装结果。

erl -version

回显类似如下信息，说明erlang安装成功。

```
[root@ecs-rabbitmq ~]# erl -version
Erlang (SMP,ASYNC_THREADS,HIPE) (BEAM) emulator version 11.1.7
```

3. 安装RabbitMQ

a. 执行如下命令，进入用户主目录。

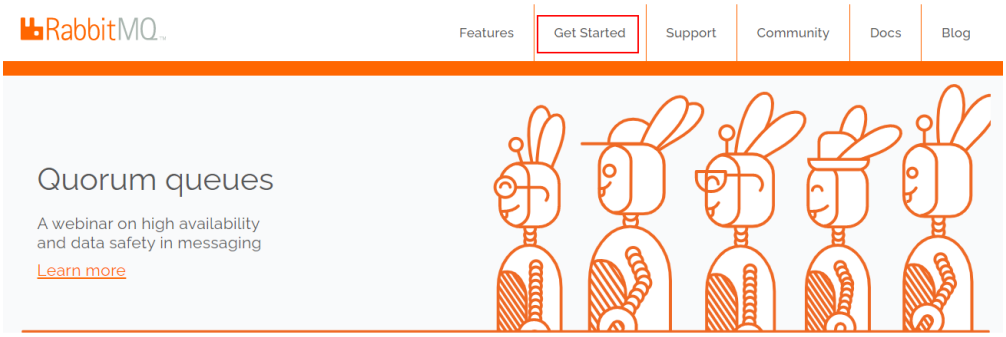
cd

b. 执行如下命令，下载RabbitMQ安装包。

i. 打开Rabbit官网。

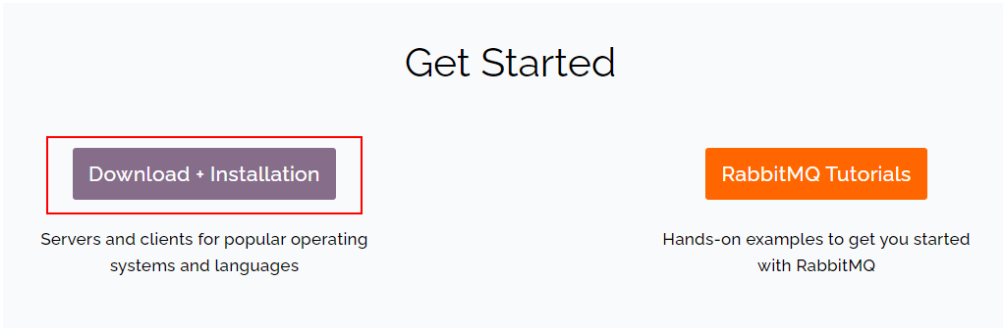
ii. 点击“Get Started”。

图1 Get Started



iii. 找到并单击“Download+Installation”。

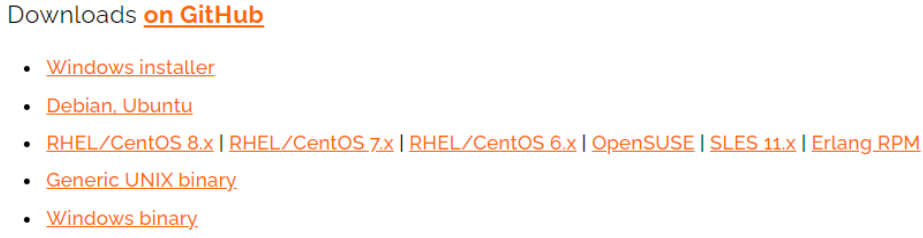
图2 Download+Installation



iv.

根据云服务器的操作系统选择下载地址。例如本例中使用的是CentOS 7.x的下载地址。

图3 选择下载地址



v. 在服务器上执行执行以下命名下载RabbitMQ安装包。

例如3.b.iv查找的下载地址是：
https://github.com/rabbitmq/rabbitmq-server/releases/download/v3.8.12/rabbitmq-server-3.8.12-1.el7.noarch.rpm
则执行的命令如下：
wget https://github.com/rabbitmq/rabbitmq-server/releases/download/v3.8.12/rabbitmq-server-3.8.12-1.el7.noarch.rpm
如果下载过程中提示“Unable to establish SSL connection.”
可以在wget命令后加--no-check-certificate，重复执行几次，即可下载。
例如：
wget https://github.com/rabbitmq/rabbitmq-server/releases/download/v3.8.12/rabbitmq-server-3.8.12-1.el7.noarch.rpm --no-check-certificate

vi. 执行以下命令安装RabbitMQ安装包。
yum install rabbitmq-server-3.8.12-1.el7.noarch.rpm

c. 安装完毕，启动RabbMQ
service rabbitmq-server start

d. 查看RabbMQ状态。
service rabbitmq-server status

4. 执行如下命令，启用RabbitMQ的web管理界面。

rabbitmq-plugins enable rabbitmq_management
回显类似如下信息：

```
[root@ecs-rabbitmq ~]# rabbitmq-plugins enable rabbitmq_management
Enabling plugins on node rabbit@ecs-rabbitmq:
rabbitmq_management
The following plugins have been configured:
  rabbitmq_management
  rabbitmq_management_agent
  rabbitmq_web_dispatch
Applying plugin configuration to rabbit@ecs-2b36...
The following plugins have been enabled:
  rabbitmq_management
```

```
  rabbitmq_management_agent
  rabbitmq_web_dispatch

started 3 plugins.
```

5. 执行如下命令，创建一个新用户。

   rabbitmqctl add_user *用户名 密码*
   命令示例：
   rabbitmqctl add_user root 123456

6. 执行如下命令，设置用户为管理员。

   rabbitmqctl set_user_tags *用户名* **administrator**
   命令示例：
   rabbitmqctl set_user_tags root administrator

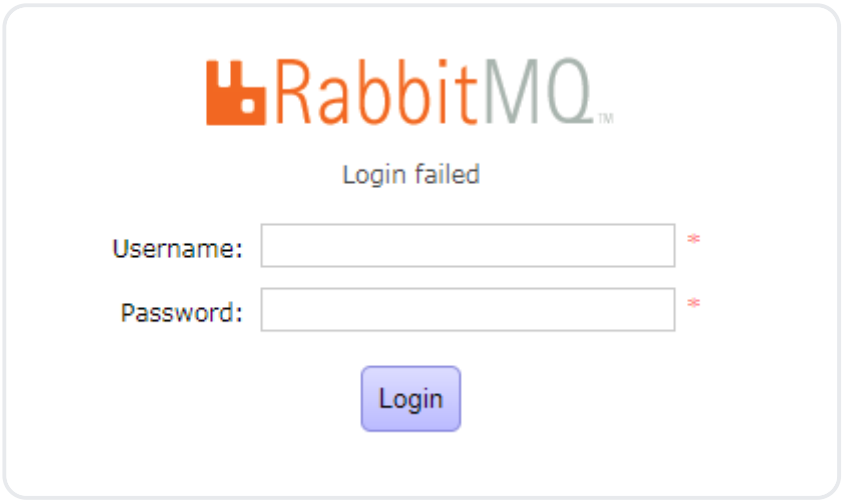
7. 执行如下命令，赋予用户所有权限。

   rabbitmqctl set_permissions -p / *用户名* **'.*' '.*' '.*'**
   命令示例：
   rabbitmqctl set_permissions -p / root '.*' '.*' '.*'

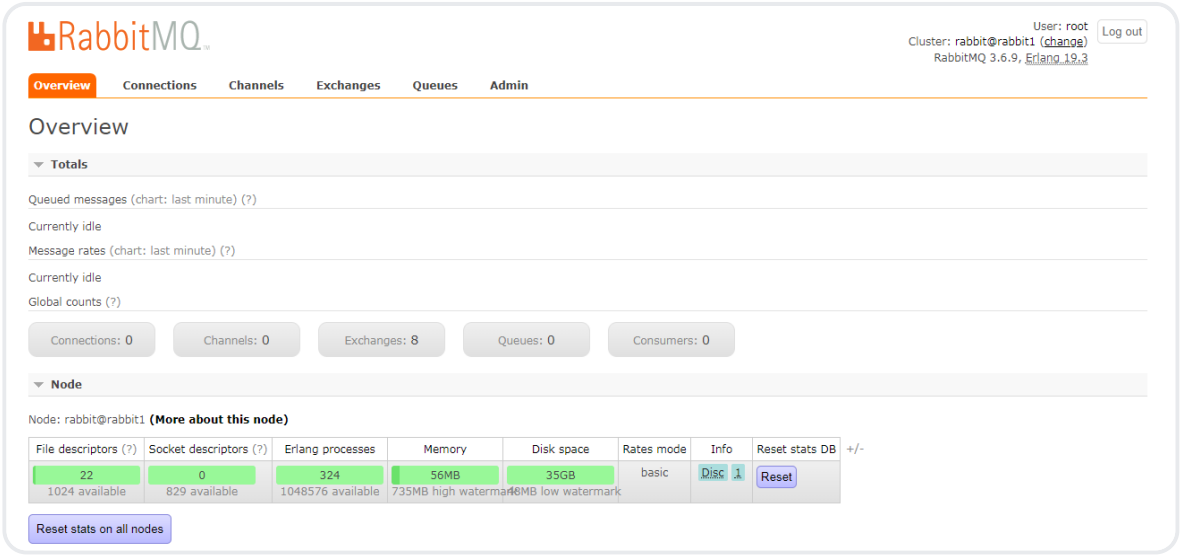
8. 执行如下命令，在后台启动RabbitMQ。

   rabbitmq-server -detached

9. 使用浏览器访问 “http://弹性公网IP:15672”，显示如下页面，说明RabbitMQ安装成功。



10. 输入[步骤5]创建的用户名和密码后点击“Login”，进入RabbitMQ管理界面。



6.3 关键指令参考

```
  240  yum install erlang
  241  elr
  244  wget https://github.com/rabbitmq/rabbitmq-server/releases/download/v3.7.7/rabbitmq-server-generic-unix-3.7.7.tar.xz
  245  tar xvf rabbitmq-server-generic-unix-3.7.7.tar.xz
  246  mv rabbitmq_server-3.7.7/ rabbitmq/
```

```
247  vim /etc/profile
249  ll
250  cd rabbitmq/sbin/
251  ll

252  ./rabbitmq-server -detached
253   service rabbitmq-server status

254  pwd
255  vim /etc/profile
257   source /etc/profile

264  rabbitmq-plugins enable rabbitmq_management

266  rabbitmqctl add_user admin rabbitmq
267  rabbitmqctl set_user_tags admin administrator
268  rabbitmqctl set_permissions -p / admin  '.*' '.*' '.*'
269  rabbitmq-server -detached
270  ps afx|grep rabbit
271
273  rabbitmq-server -detached

```

7. ElasticSearch

切换服务器目录 [root]# cd /usr/local

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.6.2-linux-x86_64.tar.gz
tar -xzvf  elasticsearch-7.6.2-linux-x86_64.tar.gz
```

将elasticsearch-7.6.2.tar.gz放入local目录下 [root]# tar xzf elasticsearch-7.6.2.tar.gz

[root]# cd /usr/local/elasticsearch-7.6.2/config [root]# vim elasticsearch.yml 取消下面配置的注释，并修改为当前主机ip地址

```
cluster.name: my-application
node.name: node-1
network.host: 0.0.0.0 #获取填写本机ip
http.port: 9200
discovery.zen.ping.unicast.hosts: ["0.0.0.0"]
discovery.zen.minimum_master_nodes: 1 #注意，因为本人目前是单节点，这里必须为1
```

新增如下配置

```
transport.tcp.port: 9300
transport.tcp.compress: true
bootstrap.system_call_filter: false

http.cors.enabled: true
http.cors.allow-origin: "*"

indices.query.bool.max_clause_count: 1000000  #in查询参数最多1000000条
```

修改etc/sysctl.conf配置文件 [root]# vim /etc/sysctl.conf

```
#elasticsearch用户拥有的内存权限
vm.max_map_count=262144
```

执行命令使配置生效 [root]# sysctl -p

修改/etc/security/limits.conf配置文件，在末尾添加配置 [root]# vim /etc/security/limits.conf

```
#每个进程最大同时打开文件数
* soft nofile 65536
* hard nofile 65536
* soft nproc  65536
* hard nproc  65536
```

重点：es 规定 root 用户不能启动 es，所以需要新建一个其他用户来修改es的配置文件，并启动

```
[root]# adduser userEs
```

为用户solin设置密码，输入两次相同的密码，后期配置kibana要用到(我设置的密码是userEs666)

```
[root]# passwd userEs
添加权限
[root]# chown -R userEs /usr/local/elasticsearch-7.6.2
[root]# su userEs
```

准备启动

```
[userEs]# cd /usr/local/elasticsearch-7.6.2/bin
```

后台启动

```
[userEs]# ./elasticsearch -d
```

测试是否启动成功

```
[userEs]# curl -XGET http://127.0.0.1:9200
```

开启9200端口访问：

```
firewall-cmd --zone=public --add-port=9200/tcp --permanent
firewall-cmd --reload
```

查看端口：看到9100，9200端口都已经启动

```
[userEs]# ss -tanl
```

一、设置密码(https://blog.csdn.net/weixin_43627706/article/details/125393511)

```
 ./elasticsearch-certutil ca
 # 提示设置密码直接回车就行
 cd ..
ls #这里在elasticsearch根目录已经可以看到 elastic-stack-ca.p12这个文件了

./elasticsearch-certutil cert --ca /usr/local/elasticsearch-
7.6.2/elastic-stack-ca.p12
 # 提示设置密码直接回车就行

 # 进入/elasticsearch
cd /config
mkdir certs
cp /usr/local/elasticsearch-7.6.2/elastic-certificates.p12 certs #拷贝
```

1.需要在配置文件中开启x-pack验证，修改config目录下面的elasticsearch.yml文件，在里面添加如下内容,并重启.

xpack.security.enabled: true xpack.license.self_generated.type: basic xpack.security.transport.ssl.enabled: true 2，执行设置用户名和密码的命令,这里需要为4个用户分别设置密码，elastic，kibana，logstash_system,beats_system

```
./elasticsearch-setup-passwords interactive # 在elasticsearch/bin目录下执行该命令，设置密码
```

213

parkingEs@userEs213

带密码查询

Elasticsearch设置用户名密码之后，不能再直接使用Elasticsearch head 访问，可以在查询等API上加上用户等参数:

curl -XGET --user user:passwd 'http://XXXX:9200/XX/XXX'

测试服 curl -XGET http://172.24.236.64:9200

穗腾 curl -XGET http://172.27.42.99:9200

修改es运行内存 [root]# export ES_HEAP_SIZE=2g

8. Canal

8.1 多租户场景中canal多个数据源实例配置教程

8.1.1 1.前言

很多时候，我们很多业务场景可能只需要同步多个或者单个数据库多个或者单个表的数据，canal提供了多实例(Instance)功能让我们可以处理这些业务场景。

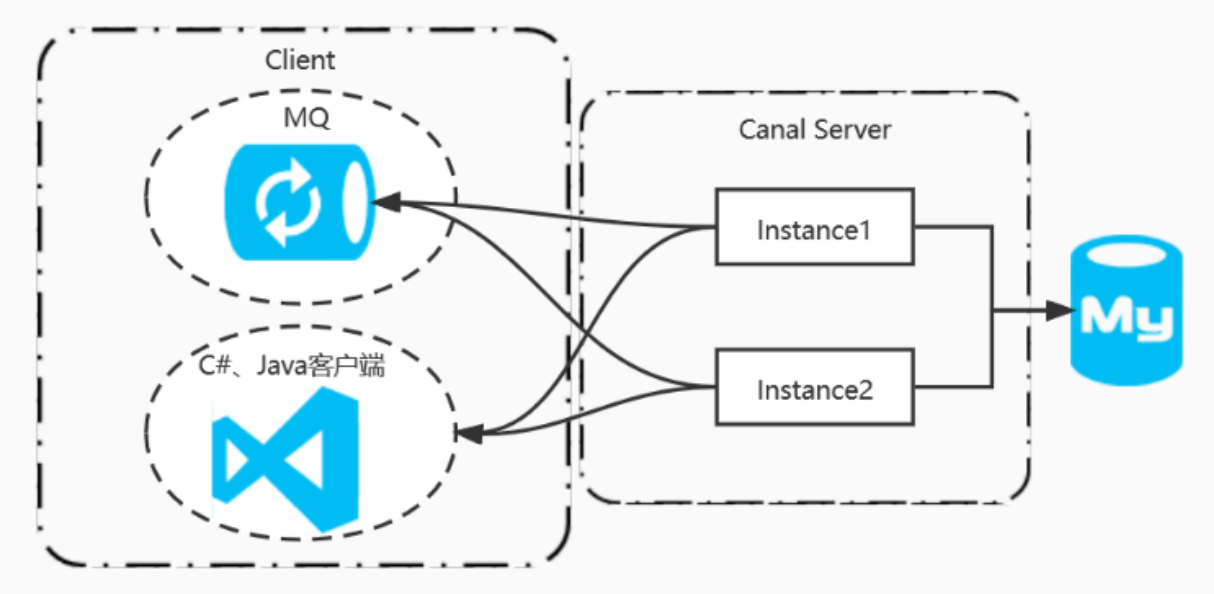
8.1.2 2.前期准备

服务名称	IP/域名	端口
zookeeper	192.168.142.129,192.168.142.130,192.168.142.131	2181
mysql	192.168.142.131	3306
rabbitmq	192.168.18.230	5672

8.1.2.1 2.1RabbitMQ参数

虚拟主机(Virtual hosts)：Canal
交换器(Exchanges)：exchange.canal
路由key(Routing key)：routing.quote、routing.supplier
队列(Queues)：quote_example、supplier_example

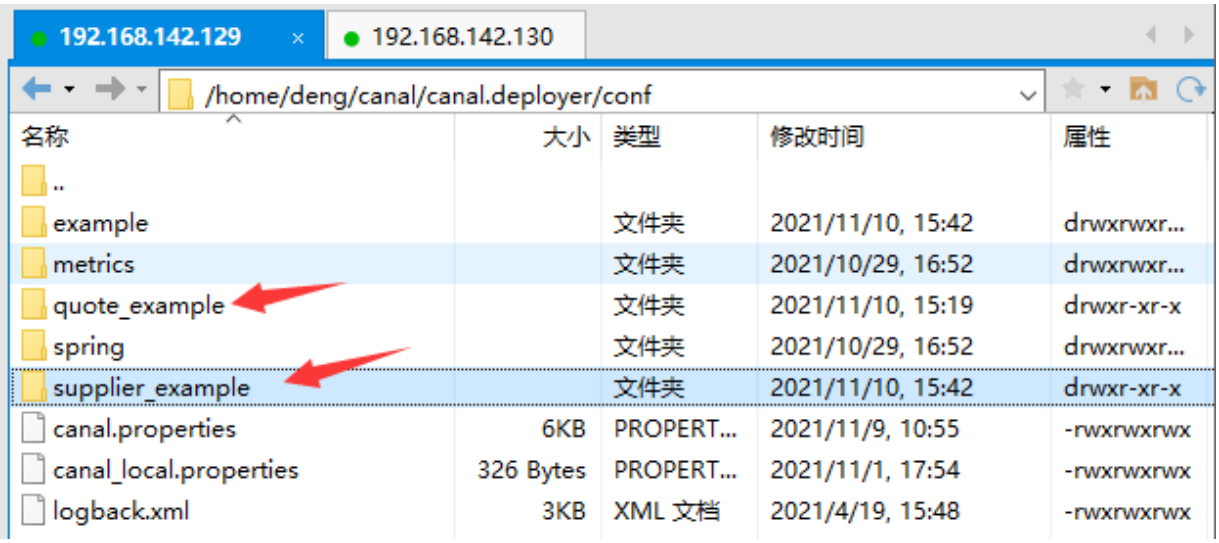
8.1.3 3.两个实例(Instance)监听相同数据库



先从conf目录下拷贝两份example实例，具体操作命令如下：

```
//供应商报价实例
cp -r /home/deng/canal/canal.deployer/conf/example
/home/deng/canal/canal.deployer/conf/quote_example
//供应商实例
cp -r /home/deng/canal/canal.deployer/conf/example
/home/deng/canal/canal.deployer/conf/supplier_example
```

如图所示：



客户端我们还是沿用上章节MQ（解析两个实例数据），这里不做代码如何解析数据了。

●修改canal.properties配置，把之前创建**supplier_example***、**quote_example*****两个实例配置到**canal.destinations**选项去（如果配置集群，记得每个canal服务配置都要修改）：

```
vi conf/canal.properties
canal.destinations = quote_example,supplier_example
```

并找到RabbitMQ标题栏配置MQ配置：

```
##################################################
#########             RabbitMQ          #############
##################################################
rabbitmq.host = 192.168.18.230 --MQ连接地址
rabbitmq.virtual.host = Canal --virtualHost
rabbitmq.exchange = exchange.canal --交换器名称
rabbitmq.username = admin --MQ登录用户名
rabbitmq.password = 123456 --MQ登录密码
rabbitmq.deliveryMode = 2 --投递模式，实现消息持久化：1.非持续性，2.持续性
```

●两个实例(quote_example、supplier_example)instance.properties配置相同参数（不知道这几个参数是什么含义，可以翻看我之前写的文章或者查看官网）：

```
canal.instance.master.address=192.168.142.131:3306 --数据库连接地址
canal.instance.dbUsername=canal --数据库登录用户名
canal.instance.dbPassword=qwer1234 --数据库登录密码
```

●修改quote_example．instance.properties配置：

```
vi conf/quote_example/instance.properties
canal.mq.topic=routing.quote --MQ路由键
canal.instance.filter.regex=ebs_material.supplier_quote --数据解析关注的表（格式：数据库.表）
```

●修改supplier_example．instance.properties配置

```
vi conf/supplier_example/instance.properties
canal.mq.topic=routing.supplier --MQ路由键
canal.instance.filter.regex=ebs_material.supplier --数据解析关注的表（格式：数据库.表）
```

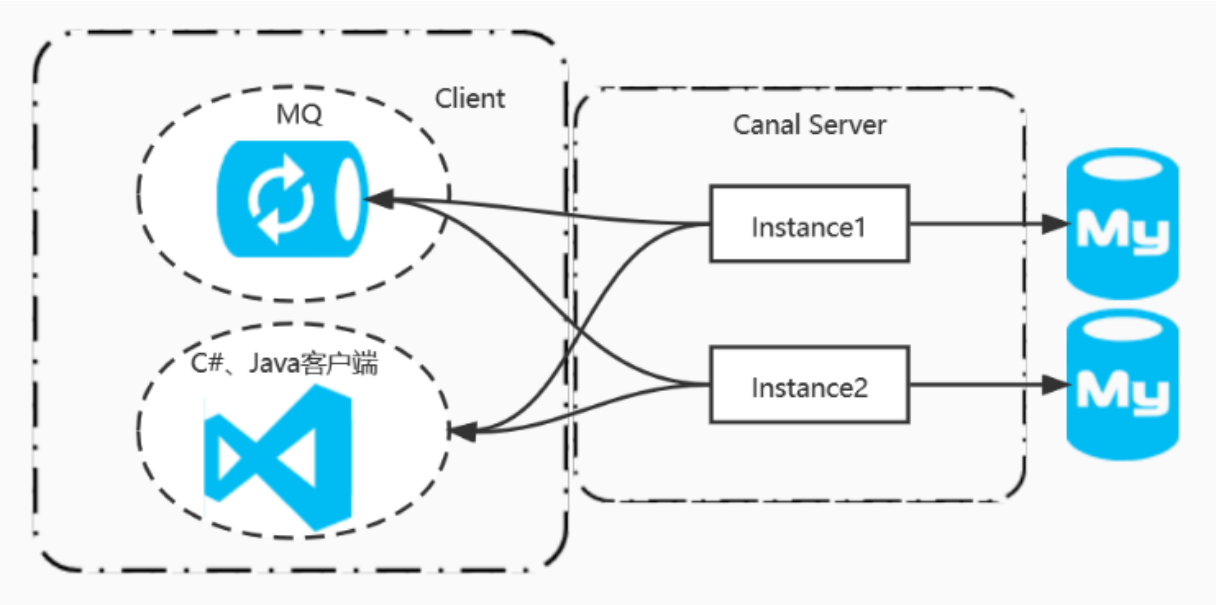
●清空rabbitmq消息，然后执行如下sql语句：

```
INSERT INTO ebs_material.supplier_quote
(PN,Brand,StockQty,SupplierId,CreateTime) VALUES
('LM358DT','TI',10000,1,'2022-02-14 00:00:00');
INSERT INTO ebs_material.supplier (Name,CreateTime) VALUES ('艾睿','2022-
02-14 00:00:00');
```

登录rabbitmq管理后台我们会看到只有两条待消费的消息（supplier_quote和supplier表插入行数据）：

				Messages			Message rates		
Name	**Type**	Features	**State**	**Ready**	**Unacked**	**Total**	**incoming**	**deliver / get**	**ack**
quote_example	classic	D Args	idle	1	0	1	0.00/s	0.00/s	0.00/s
supplier_example	classic	D Args	idle	1	0	1	0.00/s	0.00/s	0.00/s

8.1.4 4.两个实例(Instance)监听不相同数据库



canal两个实例(Instance)监听不同数据库数据同步，其实也只是把每个实例连接数据库地址等配置修改下就可以了，其他配置跟上面小节基本一样的，想了解可以自行到官网查看，这里就不多说了，请大伙自行配置测试。

8.1.5 5．mysql数据解析关注的和Perl正则表达式

Canal为我们提供了canal.instance.filter.regex与canal.instance.filter.black.regex选项参数来过滤数据库表数据解析，类似黑白名单。常见例子有：

●所有表：.* or .\..

●canal schema下所有表：canal\..*

●canal下的以canal打头的表：canal\.canal.*

●canal schema下的一张表：canal\.test1

●多个规则组合使用：canal\..*,mysql.test1,mysql.test2（逗号分隔）

注：多个正则之间以逗号(,)分隔，转义符需要双斜杠(\)

8.1.6 6.错误处理

如果canal启动时候从日志看到报这个错误：**can't find start position for example。**有如下解决方法：

●单机

删除meta.dat文件，重启canal，问题解决。

●集群

进入canal对应的zookeeper集群下，删除节点/otter/canal/destinations/实例/1001/cursor，重启canal即可恢复（不懂命令可以到zookeeper官网或者百度查找操作命令）。

参考文献：

Canal Kafka/RocketMQ QuickStart

AdminGuide

9. 关于的封面、封底

- 有关「封面、封底」，以及 VLOOK™ 的更多信息，详见《VLOOK 快速参考手册》
- 若你的文档不需要封面、封底，可以将本文档中开始位置的 6 级标题，以及结束位置的 1 级标题内容删除即可。

The End