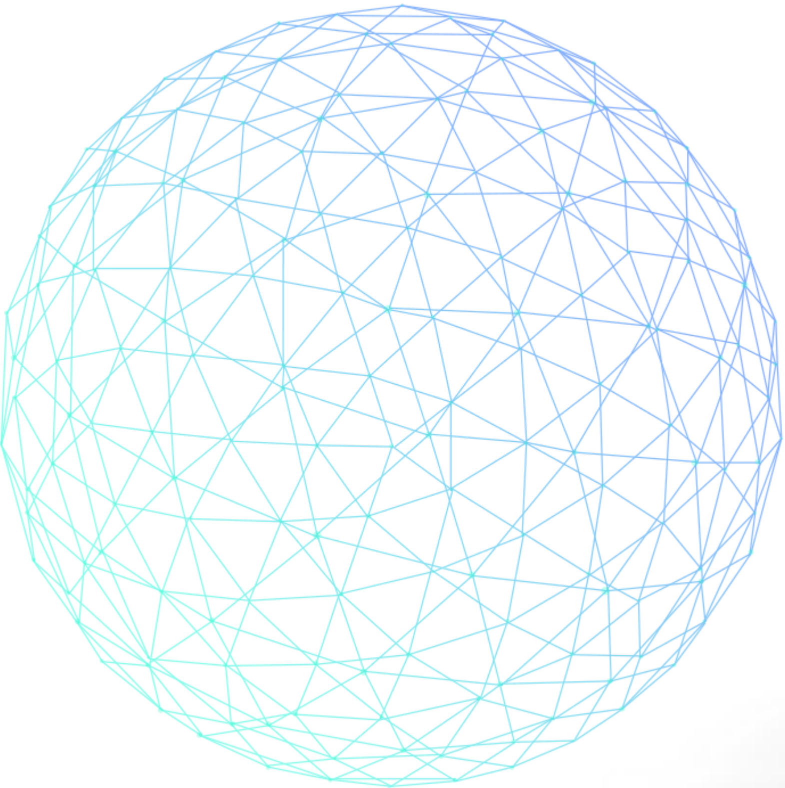
DaaS Platform

Upgrade Documentation



【Version ： 3.6.1】

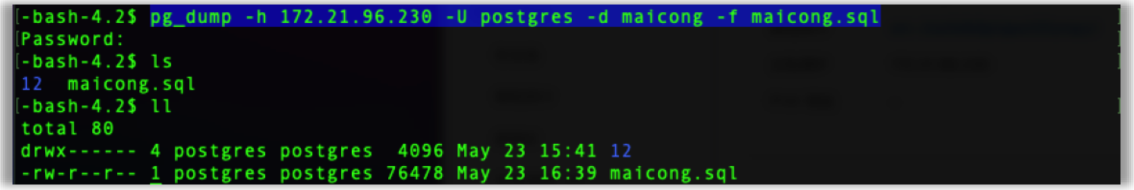
Menu

1. Preparation

1.1 Backup POSTGRESQL Database

Execute the backup command

```
pg_dump -h <local IP> -U postgres -d maicong -f maicong.sql
```



```
-bash-4.2$ pg_dump -h 172.21.96.230 -U postgres -d maicong -f maicong.sql
Password:
-bash-4.2$ ls
12  maicong.sql
-bash-4.2$ ll
total 80
drwx------ 4 postgres postgres  4096 May 23 15:41 12
-rw-r--r-- 1 postgres postgres 76478 May 23 16:39 maicong.sql
```

***Note:**

v -h: The IP address where the database is located

v -U: Username

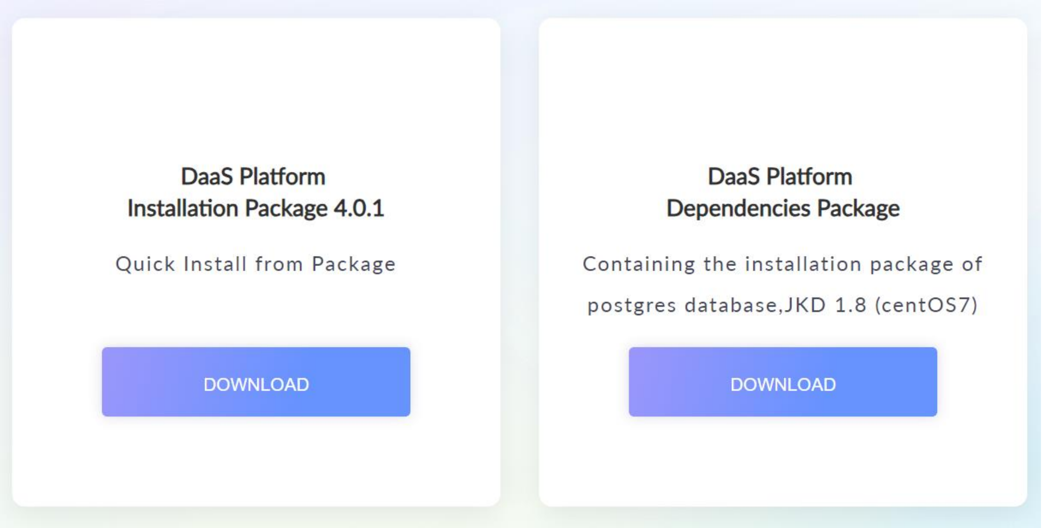
v -d: The name of the database to be backed up

v -f: The name of the file to be saved

1.2 Prepare the Installation Package

Visit the download page https://www.sqlynx.com

Choose the appropriate software version and click to download.



DaaS Platform
Installation Package 4.0.1

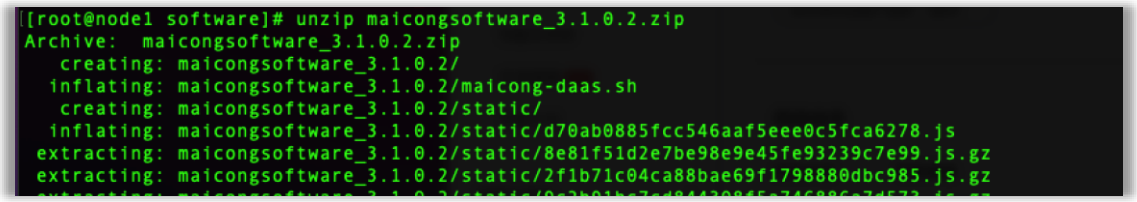
Quick Install from Package

DaaS Platform
Dependencies Package

Containing the installation package of postgres database,JKD 1.8 (centOS7)

Move the installation package to the server and unzip

```
unzip maicongsoftware_3.1.0.2.zip
```



```
[root@node1 software]# unzip maicongsoftware_3.1.0.2.zip
Archive:  maicongsoftware_3.1.0.2.zip
   creating: maicongsoftware_3.1.0.2/
  inflating: maicongsoftware_3.1.0.2/maicong-daas.sh
   creating: maicongsoftware_3.1.0.2/static/
  inflating: maicongsoftware_3.1.0.2/static/d70ab0885fcc546aaf5eee0c5fca6278.js
 extracting: maicongsoftware_3.1.0.2/static/8e81f51d2e7be98e9e45fe93239c7e99.js.gz
 extracting: maicongsoftware_3.1.0.2/static/2f1b71c04ca88bae69f1798880dbc985.js.gz
```

2. Upgrade DaaS Platform

2.1 Pause Service

Go to the original installation directory.

```
cd maicongsoftware_3.1.0.2
```

Pause service

```
./maicong-daas.sh stop
```

```
[root@node1 maicongsoftware_3.1.0]# ./maicong-daas.sh stop
maicong-daas server is stopped.
```

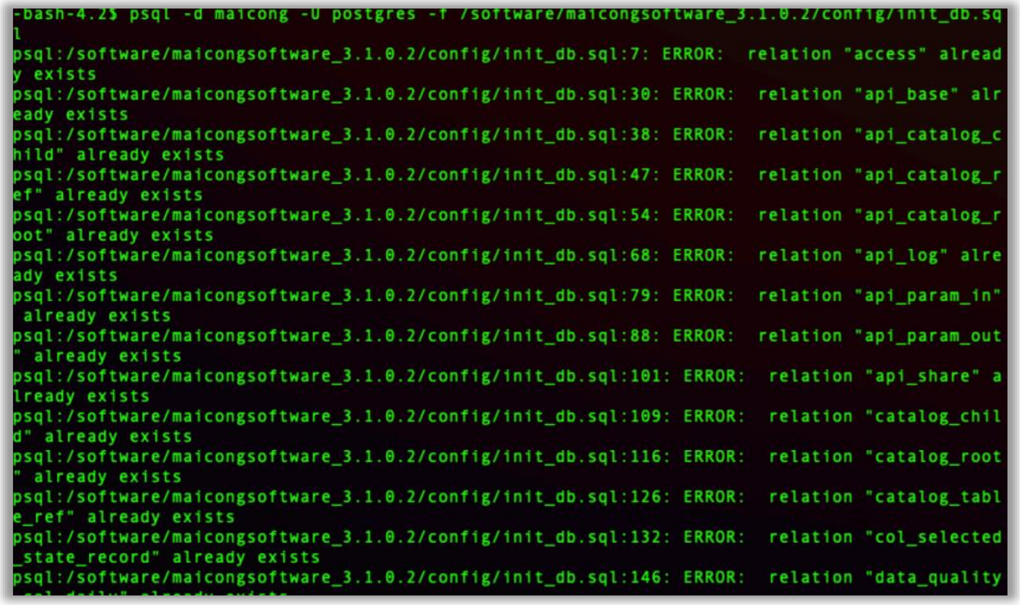
2.2 Upgrade Service

2.2.1 Upgrade PostgreSQL's System Library

Go to the machine where PostgreSQL is installed.

Execute database upgrade script:

```
psql -h <local IP> -d maicong -U postgres -f /<path to extracted files>/config/init_db.sql
```



```
-bash-4.2$ psql -d maicong -U postgres -f /software/maicongsoftware_3.1.0.2/config/init_db.sq
l
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:7: ERROR:  relation "access" alread
y exists
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:30: ERROR:  relation "api_base" alr
eady exists
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:38: ERROR:  relation "api_catalog_c
hild" already exists
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:47: ERROR:  relation "api_catalog_r
ef" already exists
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:54: ERROR:  relation "api_catalog_r
oot" already exists
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:68: ERROR:  relation "api_log" alre
ady exists
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:79: ERROR:  relation "api_param_in"
 already exists
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:88: ERROR:  relation "api_param_out
" already exists
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:101: ERROR:  relation "api_share" a
lready exists
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:109: ERROR:  relation "catalog_chil
d" already exists
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:116: ERROR:  relation "catalog_root
" already exists
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:126: ERROR:  relation "catalog_tabl
e_ref" already exists
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:132: ERROR:  relation "col_selected
_state_record" already exists
psql:/software/maicongsoftware_3.1.0.2/config/init_db.sql:146: ERROR:  relation "data_quality
_col_daily" already exists
```

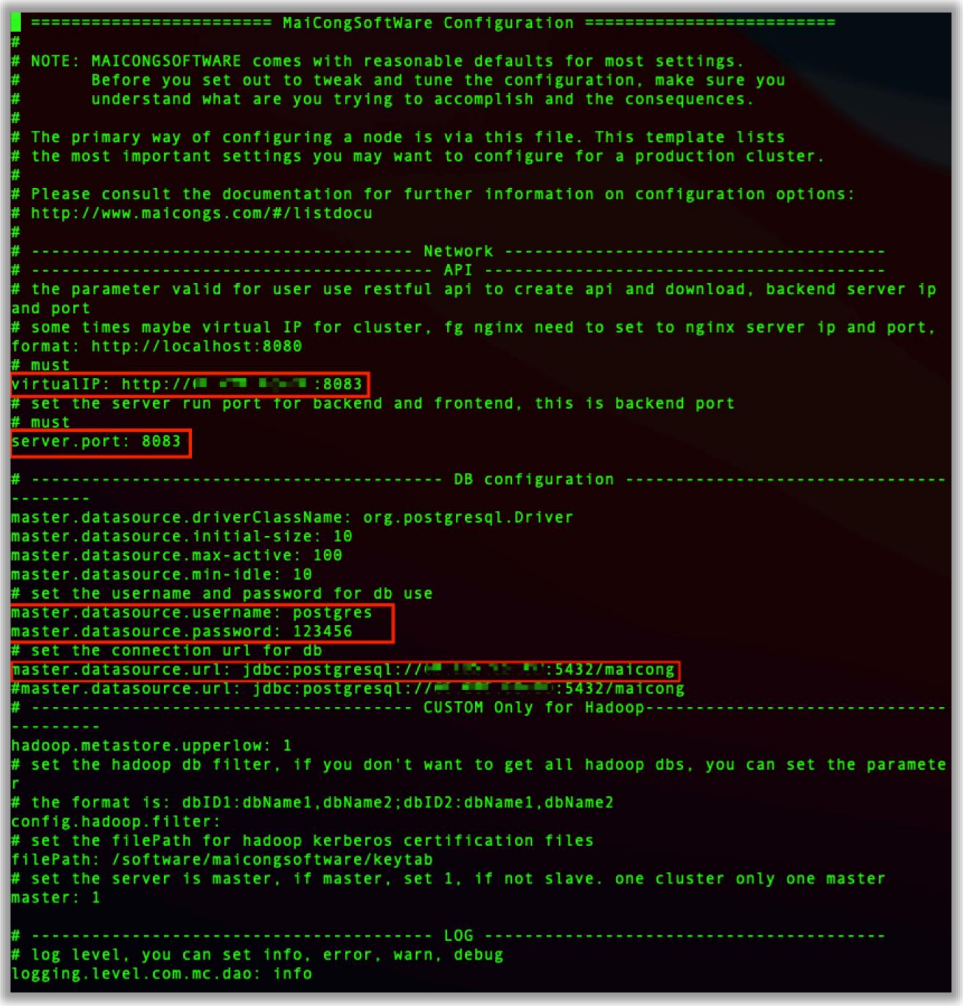
***Note:**

Upgrade and initialization use the same "init_db.sql," which contains multiple table creation statements. Since these tables already exist, errors will be reported, as shown in the figure below with multiple ERROR messages; no need to worry.

2.2.2 Configure Parameters

Modify product configuration Modify config/maicong.yaml file.

***Note:** There should be an English space behind the colon ":" in the yaml file.



```
 ======================= MaiCongSoftWare Configuration ========================
#
# NOTE: MAICONGSOFTWARE comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please consult the documentation for further information on configuration options:
# http://www.maicongs.com/#/listdocu
#
# ------------------------------------- Network -------------------------------------
# --------------------------------------- API ---------------------------------------
# the parameter valid for user use restful api to create api and download, backend server ip
and port
# some times maybe virtual IP for cluster, fg nginx need to set to nginx server ip and port,
format: http://localhost:8080
# must
virtualIP: http://[redacted]:8083
# set the server run port for backend and frontend, this is backend port
# must
server.port: 8083

# ---------------------------------------- DB configuration --------------------------------
--------
master.datasource.driverClassName: org.postgresql.Driver
master.datasource.initial-size: 10
master.datasource.max-active: 100
master.datasource.min-idle: 10
# set the username and password for db use
master.datasource.username: postgres
master.datasource.password: 123456
# set the connection url for db
master.datasource.url: jdbc:postgresql://[redacted]:5432/maicong
#master.datasource.url: jdbc:postgresql://[redacted]:5432/maicong
# ------------------------------------- CUSTOM Only for Hadoop-----------------------------
---------
hadoop.metastore.upperlow: 1
# set the hadoop db filter, if you don't want to get all hadoop dbs, you can set the paramete
r
# the format is: dbID1:dbName1,dbName2;dbID2:dbName1,dbName2
config.hadoop.filter:
# set the filePath for hadoop kerberos certification files
filePath: /software/maicongsoftware/keytab
# set the server is master, if master, set 1, if not slave. one cluster only one master
master: 1

# ---------------------------------------- LOG -----------------------------------------
# log level, you can set info, error, warn, debug
logging.level.com.mc.dao: info
```

***Note:**

virtualIP: Server address:port

server.port: Default system startup port

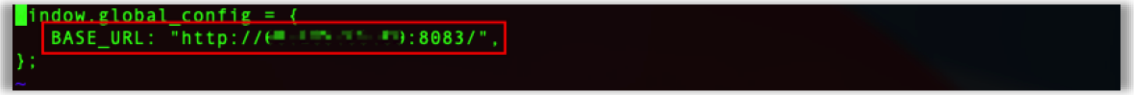
master.datasource.password: PostgreSQL connection password (a space is required after the colon)

master.datasource.url: The connection string in the PostgreSQL database includes IP, port, and database name (here as maicong, which is the database name created in the initial installation of POSTGRESQL12)

filePath: Path to store the Kerberos keytab in Hadoop (this needs to be configured if connecting to Hadoop Kerberos, otherwise not needed).

Modify the address and port of the front-end connection to the backend Modify the static/config.js file.

Set the IP:Port to point to the address and port where the application server is installed.

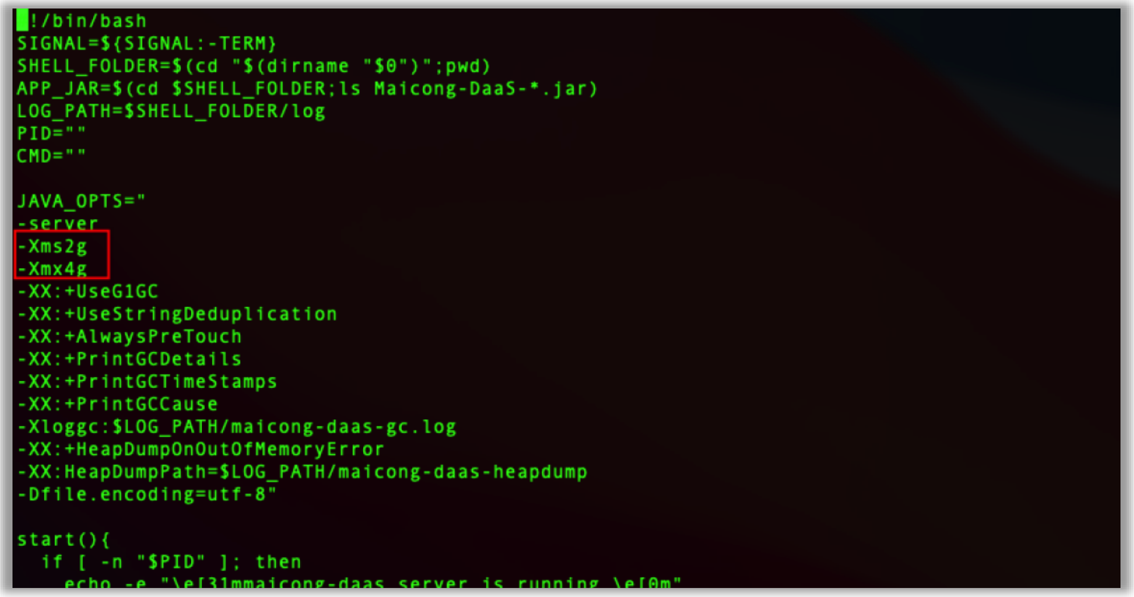


```
indow.global_config = {
   BASE_URL: "http://[redacted]:8083/",
};
```

Configure Java startup memory

```
vi maicong-daas.sh
```

Modify -xms and -xmx for startup memory and maximum memory



```
!/bin/bash
SIGNAL=${SIGNAL:-TERM}
SHELL_FOLDER=$(cd "$(dirname "$0")";pwd)
APP_JAR=$(cd $SHELL_FOLDER;ls Maicong-DaaS-*.jar)
LOG_PATH=$SHELL_FOLDER/log
PID=""
CMD=""

JAVA_OPTS="
-server
-Xms2g
-Xmx4g
-XX:+UseG1GC
-XX:+UseStringDeduplication
-XX:+AlwaysPreTouch
-XX:+PrintGCDetails
-XX:+PrintGCTimeStamps
-XX:+PrintGCCause
-Xloggc:$LOG_PATH/maicong-daas-gc.log
-XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=$LOG_PATH/maicong-daas-heapdump
-Dfile.encoding=utf-8"

start(){
  if [ -n "$PID" ]; then
    echo -e "\e[31mmaicong-daas server is running \e[0m"
```

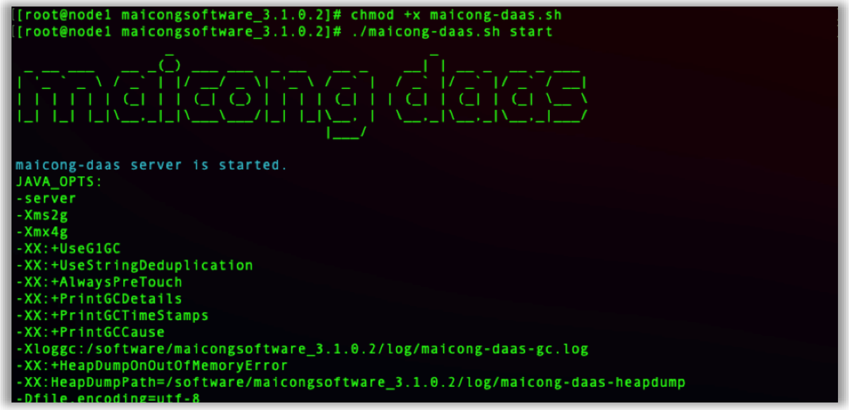
2.3 Start the Software

Add execution permissions to the startup file app.sh

```
chmod +x maicong-daas.sh
```

Start the application:

```
./maicong-daas.sh start
```

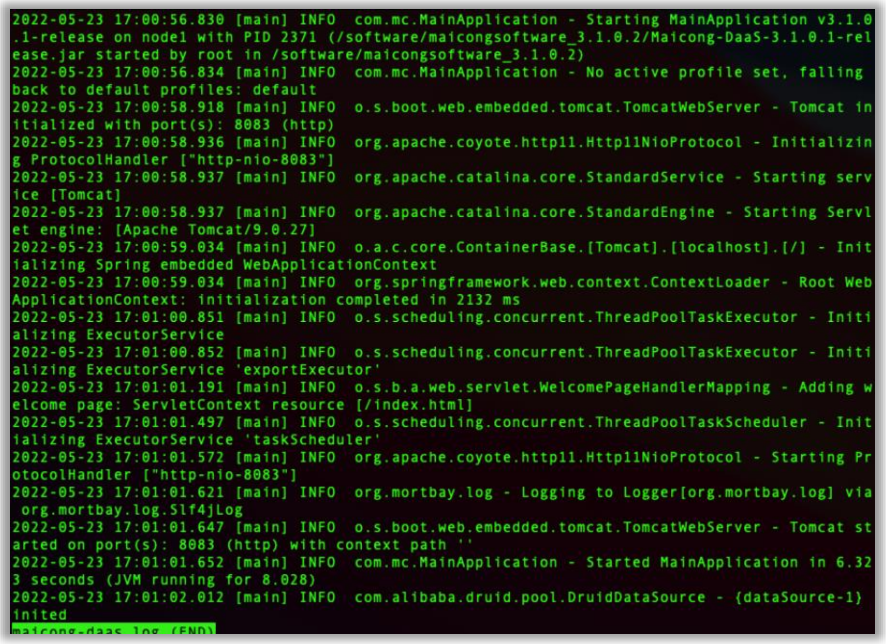


```
[root@node1 maicongsoftware_3.1.0.2]# chmod +x maicong-daas.sh
[root@node1 maicongsoftware_3.1.0.2]# ./maicong-daas.sh start
                 _                            _
 _ __ ___   __ _(_) ___ ___  _ __   __ _    __| | __ _  __ _ ___
| '_ ` _ \ / _` | |/ __/ _ \| '_ \ / _` |  / _` |/ _` |/ _` / __|
| | | | | | (_| | | (_| (_) | | | | (_| | | (_| | (_| | (_| \__ \
|_| |_| |_|\__,_|_|\___\___/|_| |_|\__, |  \__,_|\__,_|\__,_|___/
                                   |___/

maicong-daas server is started.
JAVA_OPTS:
-server
-Xms2g
-Xmx4g
-XX:+UseG1GC
-XX:+UseStringDeduplication
-XX:+AlwaysPreTouch
-XX:+PrintGCDetails
-XX:+PrintGCTimeStamps
-XX:+PrintGCCause
-Xloggc:/software/maicongsoftware_3.1.0.2/log/maicong-daas-gc.log
-XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/software/maicongsoftware_3.1.0.2/log/maicong-daas-heapdump
-Dfile.encoding=utf-8
```

Stop the application:

```
./maicong-daas.sh stop
```

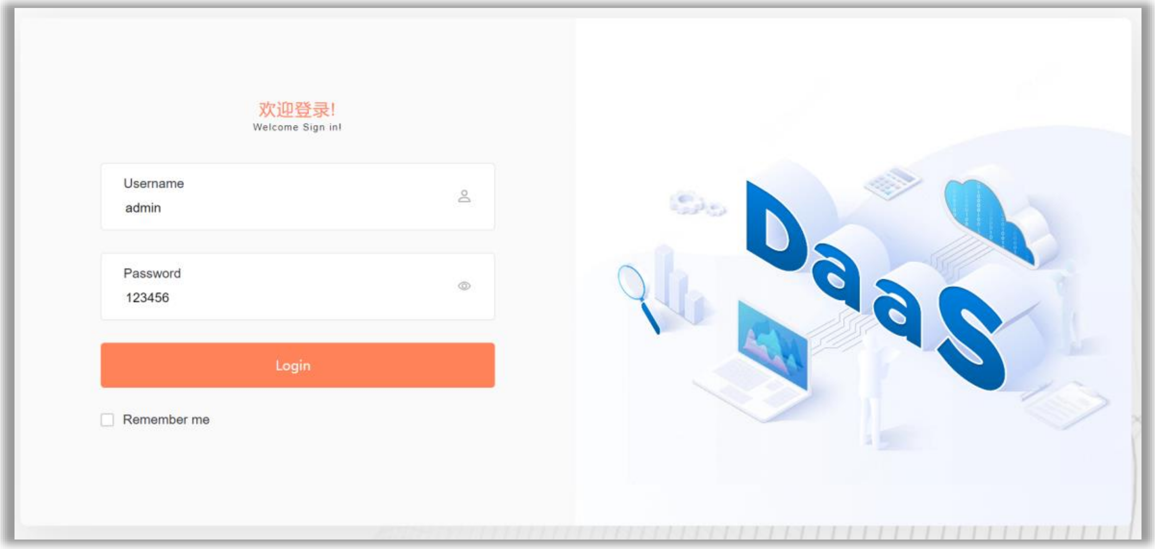
***Note:** The log file is in log/maicong-daas-console.log



```
2022-05-23 17:00:56.830 [main] INFO  com.mc.MainApplication - Starting MainApplication v3.1.0
.1-release on node1 with PID 2371 (/software/maicongsoftware_3.1.0.2/Maicong-DaaS-3.1.0.1-rel
ease.jar started by root in /software/maicongsoftware_3.1.0.2)
2022-05-23 17:00:56.834 [main] INFO  com.mc.MainApplication - No active profile set, falling
back to default profiles: default
2022-05-23 17:00:58.918 [main] INFO  o.s.boot.web.embedded.tomcat.TomcatWebServer - Tomcat in
itialized with port(s): 8083 (http)
2022-05-23 17:00:58.936 [main] INFO  org.apache.coyote.http11.Http11NioProtocol - Initializin
g ProtocolHandler ["http-nio-8083"]
2022-05-23 17:00:58.937 [main] INFO  org.apache.catalina.core.StandardService - Starting serv
ice [Tomcat]
2022-05-23 17:00:58.937 [main] INFO  org.apache.catalina.core.StandardEngine - Starting Servl
et engine: [Apache Tomcat/9.0.27]
2022-05-23 17:00:59.034 [main] INFO  o.a.c.core.ContainerBase.[Tomcat].[localhost].[/] - Init
ializing Spring embedded WebApplicationContext
2022-05-23 17:00:59.034 [main] INFO  org.springframework.web.context.ContextLoader - Root Web
ApplicationContext: initialization completed in 2132 ms
2022-05-23 17:01:00.851 [main] INFO  o.s.scheduling.concurrent.ThreadPoolTaskExecutor - Initi
alizing ExecutorService
2022-05-23 17:01:00.852 [main] INFO  o.s.scheduling.concurrent.ThreadPoolTaskExecutor - Initi
alizing ExecutorService 'exportExecutor'
2022-05-23 17:01:01.191 [main] INFO  o.s.b.a.web.servlet.WelcomePageHandlerMapping - Adding w
elcome page: ServletContext resource [/index.html]
2022-05-23 17:01:01.497 [main] INFO  o.s.scheduling.concurrent.ThreadPoolTaskScheduler - Init
ializing ExecutorService 'taskScheduler'
2022-05-23 17:01:01.572 [main] INFO  org.apache.coyote.http11.Http11NioProtocol - Starting Pr
otocolHandler ["http-nio-8083"]
2022-05-23 17:01:01.621 [main] INFO  org.mortbay.log - Logging to Logger[org.mortbay.log] via
 org.mortbay.log.Slf4jLog
2022-05-23 17:01:01.647 [main] INFO  o.s.boot.web.embedded.tomcat.TomcatWebServer - Tomcat st
arted on port(s): 8083 (http) with context path ''
2022-05-23 17:01:01.652 [main] INFO  com.mc.MainApplication - Started MainApplication in 6.32
3 seconds (JVM running for 8.028)
2022-05-23 17:01:02.012 [main] INFO  com.alibaba.druid.pool.DruidDataSource - {dataSource-1}
inited
maicong-daas.log (END)
```

2.4 Verify

Test login, visit ip:port, and use the original user password to log in



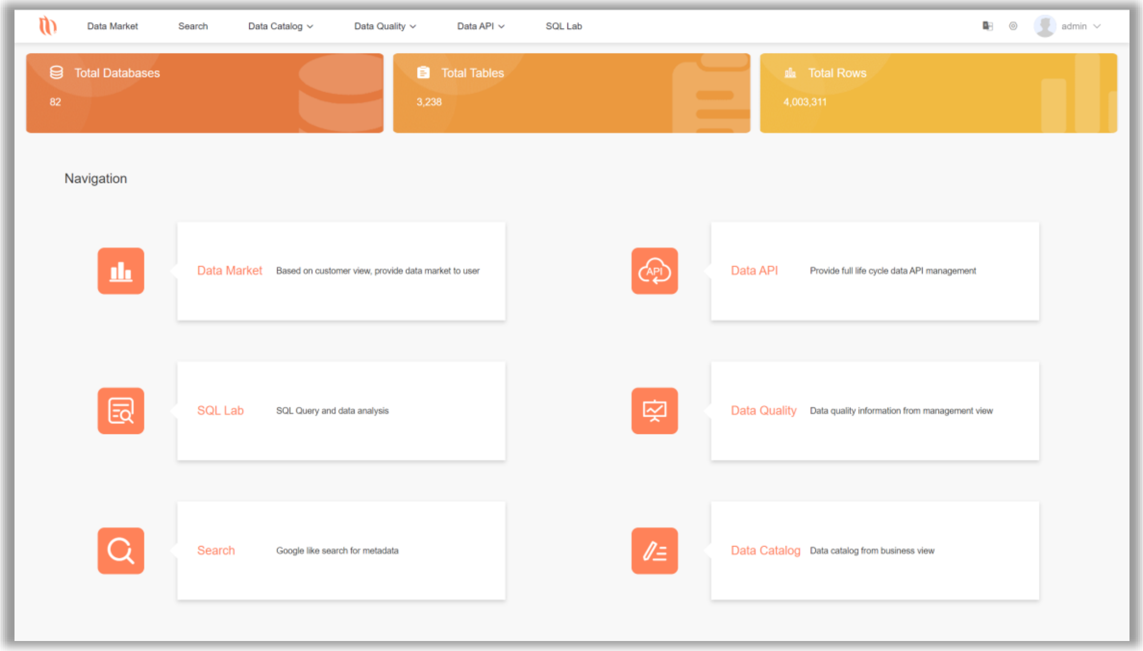
欢迎登录!
Welcome Sign in!

Username
admin

Password
123456

Login

□ Remember me



Total Databases 82

Total Tables 3,238

Total Rows 4,003,311

Navigation

Data Market Based on customer view, provide data market to user

Data API Provide full life cycle data API management

SQL Lab SQL Query and data analysis

Data Quality Data quality information from management view

Search Google like search for metadata

Data Catalog Data catalog from business view

3. Rollback

If the new version installation fails, it is necessary to restore the database and use the old version of the software to restart the application.

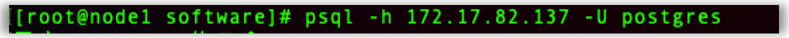
This operation is only required in the event of installation failure, so please proceed with caution.

3.1 Restore the Database

Delete the database: maicong (or according to the actual database name)

Enter psql

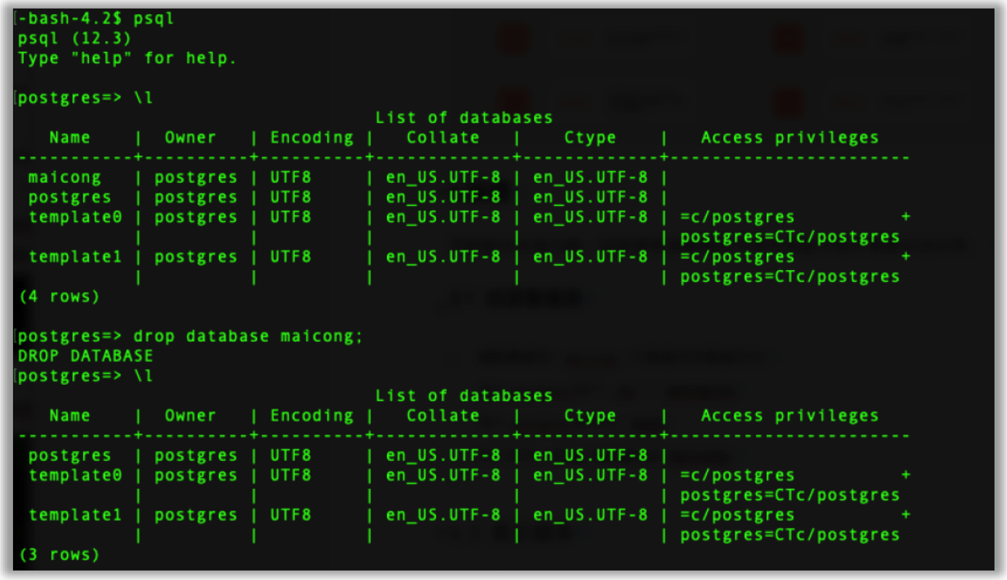
```
psql -h <local IP> -U postgres
```



```
[root@node1 software]# psql -h 172.17.82.137 -U postgres
```

Execute the command:

```
drop database maicong;
```

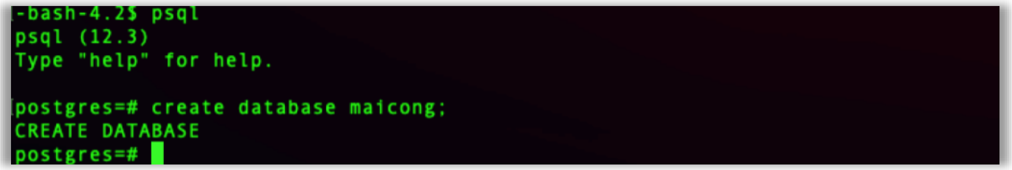


```
-bash-4.2$ psql
psql (12.3)
Type "help" for help.

postgres=> \l
                                  List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 maicong   | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
(4 rows)

postgres=> drop database maicong;
DROP DATABASE
postgres=> \l
                                  List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
(3 rows)
```

Create a new database: maicong (or use any name)

Execute the command:

```
create database maicong;
```

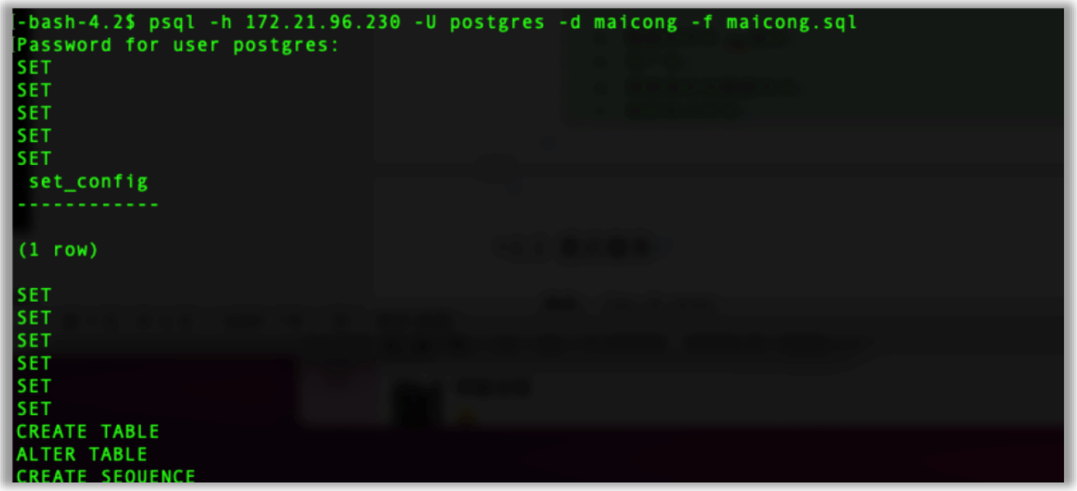


```
-bash-4.2$ psql
psql (12.3)
Type "help" for help.

postgres=# create database maicong;
CREATE DATABASE
postgres=# 
```

Execute the restore command

```
pg_restore -h <database IP> -U postgres -d maicong -f maicong.sql
```

Then enter a password.



```
-bash-4.2$ psql -h 172.21.96.230 -U postgres -d maicong -f maicong.sql
Password for user postgres:
SET
SET
SET
SET
SET
 set_config
------------
 
(1 row)

SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
```

***Note:**

v -h: The IP address where the database is located

v -U: Username

v -d: The name of database to be restored

v -f: The name of the backup file

3.2 Restart Service

Enter the old version program directory and execute the command:

```
./maicong-daas.sh start
```

***Note: I**t is the old version of the product that is being started.