**Inception安装**

**inception参考网址**

<http://www.ywnds.com/?p=9423>

**1、基础组件安装**

1、yum install gcc gcc-c++ cmake openssl-devel ncurses-devel MySQL-python –y

yum install mailcap

**1.下载并安装一个M4包**

**[plain]** [view plain](https://blog.csdn.net/qq_39591494/article/details/78239944?locationNum=6&fps=1#) [copy](https://blog.csdn.net/qq_39591494/article/details/78239944?locationNum=6&fps=1#)

#wget -O m4-1.4.9.tar.gz http://ftp.gnu.org/gnu/m4/m4-1.4.9.tar.gz

**2.下载完成后解压**

**[plain]** [view plain](https://blog.csdn.net/qq_39591494/article/details/78239944?locationNum=6&fps=1#) [copy](https://blog.csdn.net/qq_39591494/article/details/78239944?locationNum=6&fps=1#)

#tar -zvxf m4-1.4.9.tar.gz && cd m4-1.4.9

3.编译并安装

**[plain]** [view plain](https://blog.csdn.net/qq_39591494/article/details/78239944?locationNum=6&fps=1#) [copy](https://blog.csdn.net/qq_39591494/article/details/78239944?locationNum=6&fps=1#)

#./configure && make && make install

**2、源码安装 bison**

bison版本要低于2.6

cd bison-2.5.1

./configure

make && make install

**3、安装inception**

cd inception

$ bash inception\_build.sh debug [Xcode]

inception的源码稍微调整

**4、配置文件**

nception\_remote\_backup\_host //远程备份库的host

inception\_remote\_backup\_port //远程备份库的port

inception\_remote\_system\_user //远程备份库的一个用户

inception\_remote\_system\_password //上面用户的密码

编译完成之后，就是使用了，那么需要一个配置文件（inc.cnf）:

vi /etc/inc.cnf

[inception]

general\_log=1

general\_log\_file=inception.log

port=6669

socket=/tmp/inc.socket

character-set-client-handshake=0

character-set-server=utf8

inception\_remote\_backup\_port=3306

inception\_remote\_backup\_host=172.16.40.200

inception\_remote\_system\_user=root

inception\_remote\_system\_password=hbjf2018

inception\_support\_charset=utf8mb4

inception\_enable\_nullable=0

inception\_check\_primary\_key=1

inception\_check\_column\_comment=1

inception\_check\_table\_comment=1

inception\_osc\_min\_table\_size=1

inception\_osc\_bin\_dir=pt-online-schema得位置 要装Data-Dumper(百度网盘中)

inception\_osc\_chunk\_time=0.1

inception\_enable\_blob\_type=1

inception\_check\_column\_default\_value=1

-----------------------------------------------------------------------------------------------------

[inception]

general\_log=1

general\_log\_file=inception.log

port=6669

socket=/tmp/inc.socket

character-set-client-handshake=0

character-set-server=utf8

inception\_remote\_backup\_port=3306

inception\_remote\_backup\_host=172.20.0.77

inception\_remote\_system\_user=root

inception\_remote\_system\_password=root77

inception\_support\_charset=utf8

inception\_enable\_nullable=0

inception\_check\_primary\_key=1

inception\_check\_column\_comment=1

inception\_check\_table\_comment=1

inception\_osc\_on=0关闭

inception\_osc\_min\_table\_size=20480

inception\_osc\_bin\_dir=/opt/inception/percona-toolkit-2.2.17/bin

inception\_osc\_chunk\_time=0.1

inception\_enable\_blob\_type=1

inception\_check\_column\_default\_value=1

-----------------------------------------------------------------------------------------------------

**5、启动Inception程序**

启动方式和MySQL是一样的。

nohup /opt/inception/inception-master/debug/mysql/bin/Inception --defaults-file=/etc/inc.cnf &

nohup /opt/inception/inception-master/debug/mysql/bin/Inception --defaults-file=/etc/inc.cnf &

nohup /opt/inception/inception-master/debug/mysql/bin/Inception --defaults-file=/etc/inc.cnf &

nohup /opt/dms/inception-master/debug/mysql/bin/Inception --defaults-file=/etc/inc.cnf &

启动成功之后，可以简单试一下看，通过MySQL客户端

mysql -uroot -h127.0.0.1 -P6669

登录上去之后，再执行一个命令：

mysql> inception get variables;

**Dababase\_Management**

**1、安装python2.7**

Tar xvf Python-2.7.9.tgz

Cd Python-2.7.9

./configure

make && make install

mv /usr/bin/python /usr/bin/python2.6.6

Ln –s /usr/local/bin/python2.7 /usr/bin/python

检测安装

Python –V

Python 2.7.9

Yum install –y python-devel

tips:安装之后yum无法使用，修改yum源文件

 vim /usr/bin/yum

将首行

#!/usr/bin/python

更改为：

#!/usr/bin/python2.6.6

**2、redis源码安装。**

tar -xzf redis-3.2.5.tar.gz

cd /opt/redis-3.2.5/src

make

make MALLOC=libc

make install

**设置配置文件路径**

mkdir   /etc/redis

# cp /opt/redis-3.2.5/redis.conf /etc/redis

**修改配置文件**

vi /etc/redis/redis.conf

仅修改： daemonize yes

**启动**

/usr/local/bin/redis-server /etc/redis/redis.conf

**停止redis**

./redis-cli -p 6379 shutdown

为了以后运维更轻松，可以利用alias做几个别名，

vi ~/.bashrc

alias redis="cd /opt/redis-3.2.5/src"

alias startRedis="/usr/local/bin/redis-server /etc/redis/redis.conf"

alias stopRedis="/usr/local/bin/redis-cli  -h 172.16.50.100  -p 6379 shutdown"

具体路径，大家根据实际情况调整，保存退出，重新连接到linux终端

redis 即可直接进入redis根目录

startRedis 即启动redis

stopRedis 即停止redis

**3、安装setuptools和pip-9.0.3.tar.gz**

cd /opt/inception/pip-9.0.3

unzip setuptools-master.zip

python bootstrap.py

**4、pip安装组件**

pip install django==1.8.14

pip install django-celery

pip install django-simple-captcha

pip install celery==3.1.25

pip install celery-with-redis

pip install pymongo

pip install pillow==4.0.0

yum install python-devel mysql-devel zlib-devel openssl-devel

pip install MySql-python

pip install mysql-replication

pip install pycrypto

pip install crypto

pip install sqlparse

**5、percona-toolkit**

**5.1unzip SQLAdvisor-master.zip**

1. yum install cmake libaio-devel libffi-devel glib2 glib2-devel
2. yum install --enablerepo=Percona56 Percona-Server-shared-56

编译依赖项sqlparser

1. cmake -DBUILD\_CONFIG=mysql\_release -DCMAKE\_BUILD\_TYPE=debug -DCMAKE\_INSTALL\_PREFIX=/usr/local/sqlparser ./
2. make && make install

安装SQLAdvisor源码

1. cd SQLAdvisor/sqladvisor/
2. cmake -DCMAKE\_BUILD\_TYPE=debug ./
3. make
4. 在本路径下生成一个sqladvisor可执行文件，这即是我们想要的。

 SQLAdvisor使用

2.1 --help输出

./sqladvisor --help
Usage:
sqladvisor [OPTION...] sqladvisor
SQL Advisor Summary
Help Options:
-?, --help Show help options
Application Options:
-f, --defaults-file sqls file
-u, --username username
-p, --password password
-P, --port port
-h, --host host
-d, --dbname database name
-q, --sqls sqls
-v, --verbose 1:output logs 0:output nothing

2.2 命令行传参调用

./sqladvisor -h xx -P xx -u xx -p 'xx' -d xx -q "sql" -v 1

#####注意：命令行传参时，参数名与值需要用空格隔开

2.3 配置文件传参调用

$> cat sql.cnf
[sqladvisor]
username=xx
password=xx
host=xx
port=xx
dbname=xx
sqls=sql1;sql2;sql3....
cmd: ./sqladvisor -f sql.cnf -v 1

**5.2**

**pip install mysqldiff**

<https://github.com/speedocjx/db_platform>

**6、配置数据库管理系统**

**config配置**

cat /opt/inception/db\_platform-master/myapp/etc/config.py

#encoding:utf-8

wrong\_msg="select '请检查输入语句'"

select\_limit=200

export\_limit=200

incp\_host="172.16.50.100"

public\_user="public"

incp\_port=6669

incp\_user="root"

incp\_passwd="2018"

sqladvisor\_switch = 0

sqladvisor = '/usr/sbin/sqladvisor'

pt\_tool = 1

pt\_tool\_path = '/opt/inception/percona-toolkit-2.2.17/bin'

incept\_backup\_host = '172.16.40.200'

incept\_backup\_port = '3306'

incept\_backup\_user = 'root'

incept\_backup\_passwd = 'hbjf2018'

path\_to\_mysqldiff = "/usr/local/python-2.7.9/bin/mysqldiff"

script\_dir = **"/opt/dev/scripts"**

mysql\_bin=**"/opt/mysql5639/bin/"**

*#CorpID是企业号的标识*

wechart\_corpid=**"wwd1cb83bb19c4c675"**

*#corpsecretSecret是管理组凭证密钥*

wechart\_corpsecret=**"YHQeUQZfBeI\_QN-V-1WRrq01awIcXpTIW2lc62Aqh8Q"**

*#企业号中的应用id。*

wechart\_agentid=**"1000002"**

*#微信提醒task失败分钟数内提醒*

wechart\_task\_fail\_mins=80000000

**settings配置**

cat /opt/inception/db\_platform-master/mypro/settings.py

BROKER\_URL = 'redis://172.16.50.100:6379/0'

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'django',

'USER': 'django',

'PASSWORD': 'django',

'HOST': '172.16.50.100',

'PORT': '3306',

}

}

DEBUG = False

**7、启动数据库管理系统**

**export C\_FORCE\_ROOT="true"**

**nohup python manage.py celery beat >>beat.out &**

**【监控内容】**

**nohup python manage.py celery worker -E -c 5 --loglevel=info -Q monitor >> monitor.out &**

**【告警内容】**

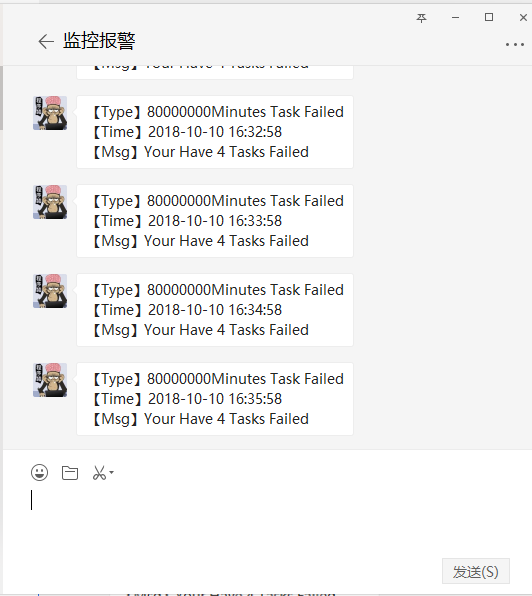
**nohup python manage.py celery worker -E -c 5 --loglevel=info -Q warning >> warning.out &**

**【默认内容】**

**nohup python manage.py celery worker -E -c 10 --loglevel=info -Q default >> default.out&**

监控内容目前使用微信企业版本做提醒。





**python manage.py celery worker -E -c 10 --loglevel=info -Q default**

**python manage.py celery worker -E -c 5 --loglevel=info -Q default**

**python manage.py celery worker -E -c 8 --loglevel=info -Q mysql\_monitor**

**nohup python manage.py celery worker -E -c 10 --loglevel=info -Q default &**

**/usr/local/bin/uwsgi --ini uwsgi.ini**

**/usr/bin/uwsgi --ini uwsgi.ini**

python manage.py runserver 0.0.0.0:8000

**# 表结构查询 `myapp\_db\_account`.`role` = 'admin'**

**# 表结构查询 `myapp\_db\_account`.`role` = 'admin'**

**#mysql query**

**`myapp\_db\_account`.`role` IN ('read', 'all')**

**`myapp\_db\_instance`.`role` IN ('read', 'all')**

操作的表必须有主键

[**Django + Uwsgi + Nginx 的生产环境部署**](http://www.cnblogs.com/chenice/p/6921727.html)

<https://www.cnblogs.com/chenice/p/6921727.html>

1、uwsgi是python的一个模块，安装uwsgi只需简单的pip命令就可以了

pip3 install uwsgi

2、进入django项目

cd /opt/dev/db\_platform-master

uwsgi --http 172.16.50.100:8000 --file mypro/wsgi.py --static-map=/static=static

命令测试启动

使用uwgsi部署时，先 python manage.py collectstatic 拷下admin之类的静态文件，不然访问/admin/页面会找不到样式 然后以刚刚注册的超级用户登陆网站进行建立普通用户、建库等配置工作

3、

cd /opt/dev

mkdir script

**vi uwsgi.ini**

# uwsig使用配置文件启动

[uwsgi]

# 项目目录

chdir=/opt/dev/db\_platform-master

# 指定项目的application

module=mypro.wsgi:application

# 指定sock的文件路径

socket=/opt/dev/script/uwsgi.sock

# 进程个数

workers=5

pidfile=/opt/dev/script/uwsgi.pid

# 指定IP端口

http=172.16.50.100:8080

# 指定静态文件

static-map=/static=/opt/dev/db\_platform-master/static

# 启动uwsgi的用户名和用户组

uid=root

gid=root

# 启用主进程

master=true

# 自动移除unix Socket和pid文件当服务停止的时候

vacuum=true

# 序列化接受的内容，如果可能的话

thunder-lock=true

# 启用线程

enable-threads=true

# 设置自中断时间

harakiri=30

# 设置缓冲

post-buffering=4096

# 设置日志目录

daemonize=/opt/dev/script/uwsgi.log

4、启动uwsgi

/opt/dev/script

/usr/local/bin/uwsgi

/usr/local/bin/uwsgi --ini uwsgi.ini

**/usr/bin/uwsgi --ini uwsgi.ini**

5、yum -y install nginx

cd /etc/nginx/conf.d/

vi mypro.conf

server {

# 这个server标识我要配置了

listen 8000; # 我要监听那个端口

server\_name 172.16.50.100 ;

# 你访问的路径前面的url名称

access\_log /var/log/nginx/access.log main;

# Nginx日志配置

charset utf-8;

# Nginx编码

gzip\_types text/plain application/x-javascript text/css text/javascript application/x-httpd-php application/json text/json image/jpeg image/gif image/png application/octet-stream;

# 支持压缩的类型

error\_page 404 /404.html; # 错误页面

error\_page 500 502 503 504 /50x.html; # 错误页面

# 指定项目路径uwsgi

location / {

# 这个location就和咱们Django的url(r'^admin/', admin.site.urls),

include uwsgi\_params;

# 导入一个Nginx模块他是用来和uWSGI进行通讯的

uwsgi\_connect\_timeout 30;

# 设置连接uWSGI超时时间

uwsgi\_pass unix:/opt/dev/script/uwsgi.sock;

# 指定uwsgi的sock文件所有动态请求就会直接丢给他

}

# 指定静态文件路径

location /static/ {

alias /opt/dev/db\_platform-master/static;

index index.html index.htm;

}

}

systemctl stop nginx