**部署环境centos6.5，最小化安装**

**必要安装包(使用yum安装)：gcc,zlib,zlib-devel,openssl,openssl-devel, libaio, ncurses,ncurses-devel,libffi-devel**

[root@dbmon ~]# rpm -qa|grep openssl

openssl-1.0.1e-15.el6.x86\_64

openssl-devel-1.0.1e-15.el6.x86\_64

[root@dbmon ~]# rpm -ivh libffi-devel-3.0.5-3.2.el6.x86\_64.rpm

## 一. 安装python2.7

**--解压缩，编译，安装**

[root@dbmon ~]# tar -zxf Python-2.7.14.tgz

[root@dbmon ~]# cd Python-2.7.14

[root@dbmon Python-2.7.14]# ./configure --prefix=/usr/local/python2.7 --with-threads --enable-shared

[root@dbmon Python-2.7.14]# make

[root@dbmon Python-2.7.14]# make install

**--备份旧的python命令，pip,easy\_install有的版本不存在，可忽略**

[root@dbmon bin]# mv /usr/bin/pip /usr/bin/pip\_old

mv: cannot stat `/usr/bin/pip': No such file or directory

[root@dbmon bin]# mv /usr/bin/easy\_install /usr/bin/easy\_install\_old

mv: cannot stat `/usr/bin/easy\_install': No such file or directory

[root@dbmon bin]# mv /usr/bin/python /usr/bin/python\_old

**--python命令做软链接**

[root@dbmon bin]# ln -s /usr/local/python2.7/lib/libpython2.7.so /usr/lib

[root@dbmon bin]# ln -s /usr/local/python2.7/lib/libpython2.7.so.1.0 /usr/lib

[root@dbmon bin]# ln -s /usr/local/python2.7/bin/python2.7 /usr/bin/python

[root@dbmon bin]# ln -s /usr/local/python2.7/lib/libpython2.7.so /usr/lib64

[root@dbmon bin]# ln -s /usr/local/python2.7/lib/libpython2.7.so.1.0 /usr/lib64

**--测试是否可用**

[root@dbmon bin]# python

Python 2.7.14 (default, Jul 16 2019, 00:48:09)

[GCC 4.4.7 20120313 (Red Hat 4.4.7-4)] on linux2

Type "help", "copyright", "credits" or "license" for more information.

>>>

**--yum 安装工具只支持系统自带的python版本， 修改配置文件使其可正常使用**

Vi /usr/bin/yum

/usr/bin/python 改为 #!/usr/bin/python2.6

## 二. 监控用户配置

### 1. 创建用户

[root@dbmon ~]# groupadd dbmon

[root@dbmon ~]# useradd -d /home/dbmon -g dbmon dbmon

[root@dbmon ~]# passwd dbmon

### 2. 安装Oracle instant client

**--使用dbmon用户**

[dbmon@dbmon ~]$ ls -ll

total 60732

-rw-r--r--. 1 root root 60704657 Jul 16 01:11 instantclient-basic-linux.x64-11.2.0.4.0.zip

-rw-r--r--. 1 root root 643089 Jul 16 01:11 instantclient-sdk-linux.x64-11.2.0.4.0.zip

-rw-r--r--. 1 root root 834491 Jul 16 01:11 instantclient-sqlplus-linux.x64-11.2.0.4.0.zip

**--解压缩**

[dbmon@dbmon ~]$ unzip instantclient-basic-linux.x64-11.2.0.4.0.zip

[dbmon@dbmon ~]$ unzip instantclient-sdk-linux.x64-11.2.0.4.0.zip

[dbmon@dbmon ~]$ unzip instantclient-sqlplus-linux.x64-11.2.0.4.0.zip

### 3. 配置环境变量

[dbmon@dbmon ~]$ cat .bash\_profile

# .bash\_profile

# Get the aliases and functions

if [ -f ~/.bashrc ]; then

. ~/.bashrc

fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin:/home/dbmon/instantclient\_11\_2

export PATH

export LD\_LIBRARY\_PATH=/home/dbmon/instantclient\_11\_2

export ORACLE\_HOME=/home/dbmon/instantclient\_11\_2

export TNS\_ADMIN=/home/dbmon/instantclient\_11\_2

## 三. 安装MYSQL数据库

**--卸载老版本MySQL**

[root@dbmon ~]# rpm -qa|grep mysql

mysql-libs-5.1.71-1.el6.x86\_64

[root@dbmon ~]# yum remove mysql

**--备份老的配置文件**

[root@dbmon ~]# mv /etc/my.cnf /etc/my.cnfbak

**--删除老的MySQL用户**

[root@dbmon home]# userdel -r mysql

userdel: user 'mysql' does not exist

**--依赖包检测**

[root@dbmon home]# rpm -q libaio

libaio-0.3.107-10.el6.x86\_64

**--创建用户**

[root@dbmon ~]# groupadd mysql

[root@dbmon ~]# useradd -g mysql -G mysql mysql

[root@dbmon ~]# passwd mysql

Changing password for user mysql.

New password:

BAD PASSWORD: it is based on a dictionary word

BAD PASSWORD: is too simple

Retype new password:

passwd: all authentication tokens updated successfully.

**--创建目录，授权**

**--应用目录**

[root@dbmon ~]# mkdir -p /u01/my3306

[root@dbmon ~]# cd /u01/my3306

[root@dbmon my3306]# mkdir tmp

[root@dbmon my3306]# mkdir run

**--日志目录**

[root@dbmon ~]# cd /u01/my3306/

[root@dbmon my3306]# mkdir log

[root@dbmon my3306]# cd log

[root@dbmon log]# mkdir iblog

[root@dbmon log]# mkdir binlog

[root@dbmon log]# mkdir ibdata

**--数据目录**

[root@dbmon ~]# mkdir -p /data/my3306

**--授权**

[root@dbmon log]# chown -R mysql:mysql /u01/my3306

[root@dbmon log]# chmod -R 775 /u01/my3306

[root@dbmon log]# chown -R mysql:mysql /data/my3306

[root@dbmon log]# chmod -R 775 /data/my3306

**--创建参数文件my.cnf**

[mysql@dbmon my3306]$ cat my.cnf

[client]

port = 3306

socket = /u01/my3306/mysql.sock

[mysqld]

autocommit = 1

general\_log = off

explicit\_defaults\_for\_timestamp = true

basedir = /u01/my3306

datadir = /data/my3306

max\_allowed\_packet = 1g

max\_connections = 3000

max\_user\_connections = 2800

open\_files\_limit = 65535

port = 3306

server\_id = 101

skip\_name\_resolve = ON

socket = /u01/my3306/run/mysql.sock

tmpdir = /u01/my3306/tmp

log-bin = /u01/my3306/log/binlog/binlog

binlog\_cache\_size = 32768

binlog\_format = row

expire\_logs\_days = 7

log\_slave\_updates = ON

max\_binlog\_cache\_size = 2147483648

max\_binlog\_size = 524288000

sync\_binlog = 100

log\_error = /u01/my3306/log/error.log

slow\_query\_log\_file = /u01/my3306/log/slow.log

log\_queries\_not\_using\_indexes = 0

slow\_query\_log = 1

log\_slow\_admin\_statements = 1

long\_query\_time = 1

relay\_log = /u01/my3306/log/relaylog.log

relay\_log\_index = /u01/my3306/log/relay.index

relay\_log\_info\_file = /u01/my3306/log/relay-log.info

slave\_load\_tmpdir = /u01/my3306/tmp

slave\_skip\_errors = OFF

innodb\_data\_home\_dir = /u01/my3306/log/iblog

innodb\_log\_group\_home\_dir = /u01/my3306/log/iblog

innodb\_adaptive\_flushing = ON

innodb\_adaptive\_hash\_index = ON

innodb\_autoinc\_lock\_mode = 1

innodb\_buffer\_pool\_instances = 8

innodb\_change\_buffering = inserts

innodb\_checksums = ON

innodb\_buffer\_pool\_size = 128M

innodb\_data\_file\_path = ibdata1:32M;ibdata2:16M:autoextend

innodb\_doublewrite = ON

innodb\_file\_format = Barracuda

innodb\_file\_per\_table = ON

innodb\_flush\_log\_at\_trx\_commit = 1

innodb\_flush\_method = O\_DIRECT

innodb\_io\_capacity = 1000

innodb\_lock\_wait\_timeout = 10

innodb\_log\_buffer\_size = 67108864

innodb\_log\_file\_size = 1048576000

innodb\_log\_files\_in\_group = 4

innodb\_max\_dirty\_pages\_pct = 60

innodb\_open\_files = 60000

innodb\_purge\_threads = 1

innodb\_read\_io\_threads = 4

innodb\_stats\_on\_metadata = OFF

innodb\_support\_xa = ON

innodb\_use\_native\_aio = OFF

innodb\_write\_io\_threads = 10

server-id = 48703306

pid\_file = /u01/my3306/run/mysql.pid

[mysqld\_safe]

**--解压**

[mysql@dbmon ~]$ ls

mysql-5.7.24-linux-glibc2.12-x86\_64.tar.gz

[mysql@dbmon ~]$ mv mysql-5.7.24-linux-glibc2.12-x86\_64/\* /u01/my3306/

**--使用root用户初始化**

[root@dbmon log]# cd /u01/my3306

[root@dbmon my3306]# ./bin/mysqld --defaults-file=/u01/my3306/my.cnf --initialize-insecure --user=mysql

**--启动**

[mysql@dbmon ~]$ cd /u01/my3306/bin

[mysql@dbmon bin]$ ./mysqld\_safe --defaults-file=/u01/my3306/my.cnf --user=mysql &

[1] 43692

[mysql@dbmon bin]$ 2019-07-16T09:13:50.960131Z mysqld\_safe Logging to '/u01/my3306/log/error.log'.

2019-07-16T09:13:51.018567Z mysqld\_safe Starting mysqld daemon with databases from /data/my3306

**--配置环境变量**

PATH=$PATH:$HOME/bin:/u01/my3306/bin

**--初始MySQL密码为空，可自行修改**

mysql> update mysql.user set host='%',authentication\_string=password('mysqld') where user='root';

Query OK, 1 row affected, 1 warning (0.02 sec)

Rows matched: 1 Changed: 1 Warnings: 1

mysql> flush privileges;

Query OK, 0 rows affected (0.00 sec)

## 四. 安装python模块

### 1. 安装cx\_oracle

**--授权**

[root@dbmon bin]# chown -R dbmon:dbmon /usr/local/python2.7/

**--拷贝libclntsh**

[dbmon@dbmon instantclient\_11\_2]$ cp libclntsh.so.11.1 libclntsh.so

**--解压缩，安装**

[dbmon@dbmon ~]$ ls -ll cx\_Oracle-5.1.3.tar.gz

-rw-r--r--. 1 dbmon dbmon 104966 Jul 16 01:21 cx\_Oracle-5.1.3.tar.gz

[dbmon@dbmon ~]$ tar -zxvf cx\_Oracle-5.1.3.tar.gz

[dbmon@dbmon ~]$ cd cx\_Oracle-5.1.3

[dbmon@dbmon cx\_Oracle-5.1.3]$ python setup.py install

**--测试是否可用**

[dbmon@dbmon cx\_Oracle-5.1.3]$ python

Python 2.7.14 (default, Jul 16 2019, 00:48:09)

[GCC 4.4.7 20120313 (Red Hat 4.4.7-4)] on linux2

Type "help", "copyright", "credits" or "license" for more information.

>>> import cx\_Oracle

### 2. 安装MYSQL-python

**--安装setuptools**

[root@dbmon ~]# unzip setuptools-39.1.0.zip

[root@dbmon setuptools-39.1.0]# python setup.py build

[root@dbmon setuptools-39.1.0]# python setup.py install

**--安装MYSQL-python**

[root@dbmon ~]# unzip MySQL-python-1.2.5.zip

[root@dbmon MySQL-python-1.2.5]# python setup.py build

[root@dbmon MySQL-python-1.2.5]# python setup.py install

**--软链接**

[root@dbmon lib]# ln -sv /u01/my3306/lib/libmysqlclient.so.20 /usr/lib64/libmysqlclient.so.20

### 3. 安装paramiko

--安装依赖包，安装方法，解压缩,python setup.py install

pycrypto-2.6.1

ecdsa-0.13

pyasn1-0.4.5

pycparser-2.19

cffi-1.12.3

PyNaCl-1.2.1

enum34-1.1.6

asn1crypto-0.22.0

cryptography-2.7

bcrypt-3.1.7

**--安装paramiko**

[root@dbmon paramiko-2.3]# python setup.py install

### 4. 安装如下模块

**安装方法：解压缩，python setup.py install，按顺序安装**

Django-1.9.13

django-crontab-0.7.1

pyexcelerator-0.6.4.1

xlwt-1.3.0

xlrd-1.1.0

xlutils-2.0.0

certifi-2019.6.16

urllib3-1.25.3

idna-2.8

chardet-3.0.4

requests-2.22.0

pexpect-4.7.0

delegator.py-0.1.1

ptyprocess-0.6.0

amqp-1.4.7

anyjson-0.3.3

kombu-3.0.30

billiard-3.3.0.20

pytz-2019.1

celery-3.1.18

django-celery-3.3.0

django-js-asset-1.2.2

django-ckeditor-5.7.1

Markdown-3.1.1

backports\_abc-0.5

six-1.12.0

singledispatch-3.4.0.3

futures-3.3.0

tornado-5.0.1

ipaddress-1.0.22

mistune-0.8.4

Pygments-2.4.2

redis-3.2.1

**注：celery需创建软链接**

[root@dbmon bin]# ln -s /usr/local/python2.7/bin/celery /usr/bin/celery

## 五. 安装rabbitmq

### 1. 安装erlang

**--依赖包检测**

[root@dbmon ~]# rpm -qa|grep openssl

openssl-1.0.1e-15.el6.x86\_64

openssl-devel-1.0.1e-15.el6.x86\_64

**--解压缩**

[root@dbmon ~]# tar -xvf otp\_src\_20.1.tar.gz

[root@dbmon ~]# cd otp\_src\_20.1

**--编译，安装**

[root@dbmon otp\_src\_20.1]# ./configure --prefix=/usr/local/erlang

[root@dbmon otp\_src\_20.1]# make

[root@dbmon otp\_src\_20.1]# make install

**--修改/etc/profile，增加下面的环境变量**

ERLANG\_HOME=/usr/local/erlang

PATH=$ERLANG\_HOME/bin:$PATH

export ERLANG\_HOME

export PATH

**--生效**

**[root@dbmon bin]# source /etc/profile  
--测试是否安装成功**

[root@dbmon ~]# erl

Erlang/OTP 20 [erts-9.1] [source] [64-bit] [smp:1:1] [ds:1:1:10] [async-threads:10] [hipe] [kernel-poll:false]

Eshell V9.1 (abort with ^G)

1>

### 2. 安装rabbitmq

**--解压，使用xz命令，如果没有需要yum安装**

[root@dbmon ~]# xz -d rabbitmq-server-generic-unix-3.6.12.tar.xz

[root@dbmon ~]# tar -xvf rabbitmq-server-generic-unix-3.6.12.tar

[root@dbmon ~]# mv rabbitmq\_server-3.6.12 /usr/local

**--启动，测试**

[root@dbmon sbin]# ./rabbitmq-server

RabbitMQ 3.6.12. Copyright (C) 2007-2017 Pivotal Software, Inc.

## ## Licensed under the MPL. See http://www.rabbitmq.com/

## ##

########## Logs: /usr/local/rabbitmq\_server-3.6.12/var/log/rabbitmq/rabbit@dbmon.log

###### ## /usr/local/rabbitmq\_server-3.6.12/var/log/rabbitmq/rabbit@dbmon-sasl.log

##########

Starting broker...

completed with 0 plugins.

## 六. 监控软件配置

### 1. 上传、解压

[dbmon@dbmon ~]$ unzip dbmon.zip

### 2. MySQL数据库初始化

**--创建数据库db\_monitor**

mysql> create database db\_monitor default character set utf8;

Query OK, 1 row affected (0.00 sec)

**--执行初始化脚本**

[mysql@dbmon ~]$ mysql -uroot -p -S /u01/my3306/run/mysql.sock -Ddb\_monitor < dbmon.sql

Enter password:

[mysql@dbmon ~]$ mysql -uroot -p -S /u01/my3306/run/mysql.sock -Ddb\_monitor < initdata.sql

Enter password:

### 3. 修改配置文件

**--config/ db\_monitor.conf，主要修改连接MySQL配置**

[dbmon@dbmon ~]$ cd /home/dbmon/dbmon/config/

[dbmon@dbmon config]$ vi db\_monitor.conf

[target\_mysql]

host = 192.168.48.70

port = 3306

user = root

password = mysqld

dbname = db\_monitor

**--Django配置文件，主要修改MySQL连接配置**

DATABASES = {

'default': {

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

'NAME': 'db\_monitor',

'USER': 'root',

'PASSWORD': 'mysqld',

'HOST':'192.168.48.70',

'PORT': '3306',

}

}

**--webssh需要修改一个html文件templates/linux\_mon/show\_linux.html**

**，IP修改为监控服务端的地址**

**templates/linux\_mon/show\_linux.html**

function pop(m,n){

layer.open({

type: 2 //此处以iframe举例

,title: 'webssh\_'+m

,area: ['700px', '550px']

,shade: 0

,maxmin: true,

content: ['http://192.168.48.70:8888?host='+n,],

btn: ['关闭所有'] //只是为了演示

,btn2: function(){

layer.closeAll();

}

,zIndex: layer.zIndex //重点1

,success: function(layero){

layer.setTop(layero); //重点2

},

});

}

### 4. 同步表结构，创建管理用户(web服务登录使用)

[dbmon@dbmon dbmon]$ python manage.py migrate

[dbmon@dbmon dbmon]$ python manage.py createsuperuser

### 5. 启动

**--启动采集程序**

[dbmon@dbmon check\_alarm]$ python main\_check.py

**--启动web服务**

[dbmon@dbmon dbmon]$ python manage.py runserver 192.168.48.70:8080

Performing system checks...

/home/dbmon/dbmon/dbmon/urls.py:129: RemovedInDjango110Warning: django.conf.urls.patterns() is deprecated and will be removed in Django 1.10. Update your urlpatterns to be a list of django.conf.urls.url() instances instead.

(r'^show\_tcp\_stats/', frame.show\_tcp\_stats),

System check identified no issues (0 silenced).

July 17, 2019 - 21:45:04

Django version 1.9.13, using settings 'dbmon.settings'

Starting development server at http://192.168.48.70:8080/

Quit the server with CONTROL-C.

**--启动webssh服务**

[dbmon@dbmon webssh]$ python main.py

[I 190717 21:53:20 settings:88] WarningPolicy

[I 190717 21:53:20 main:36] Listening on 0.0.0.0:8888 (http)

**--启动celery(异步任务处理)**

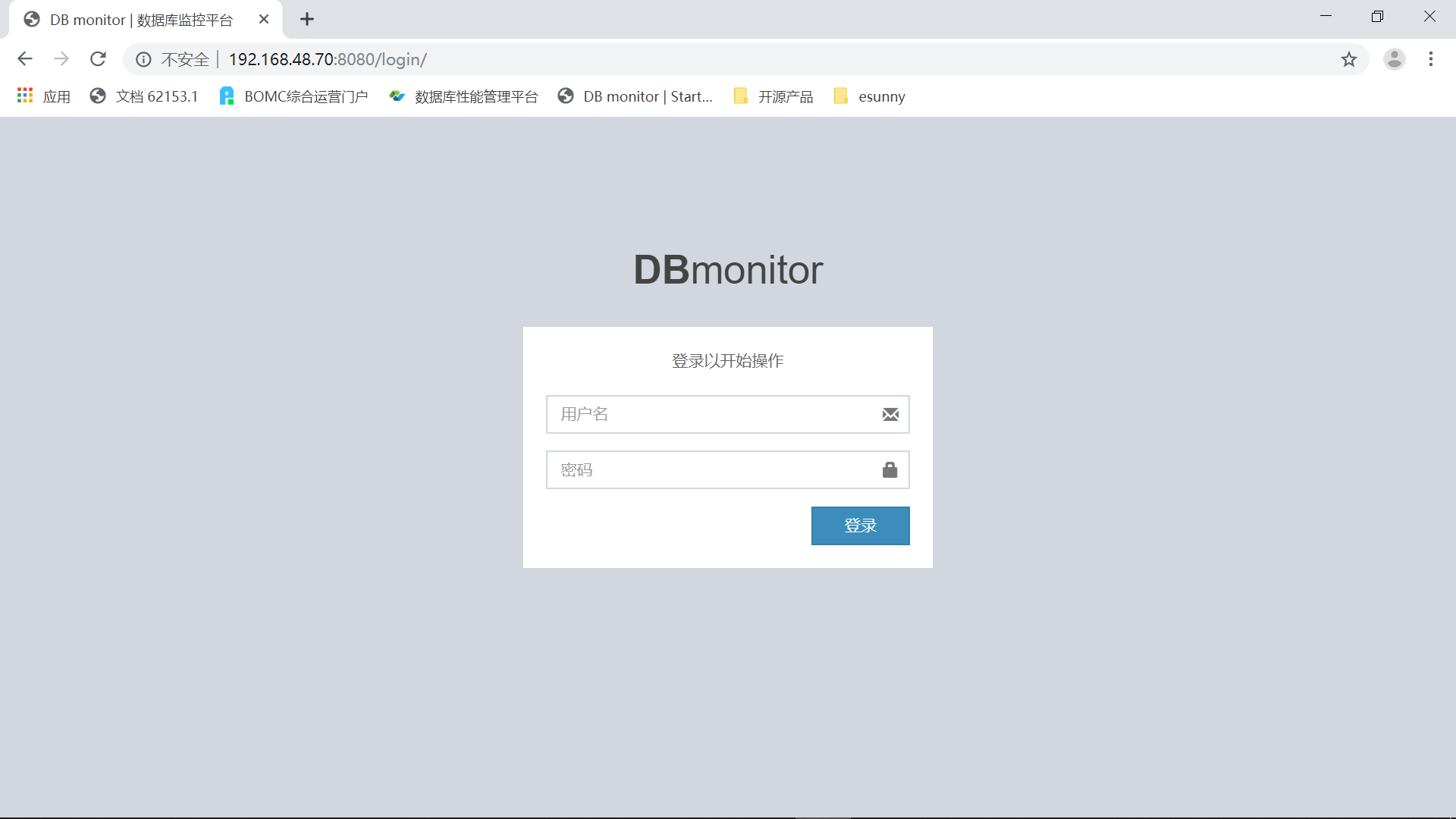
[dbmon@dbmon dbmon]$ celery -A dbmon worker -l info

[dbmon@dbmon dbmon]$ celery -A dbmon beat -l info --定时任务

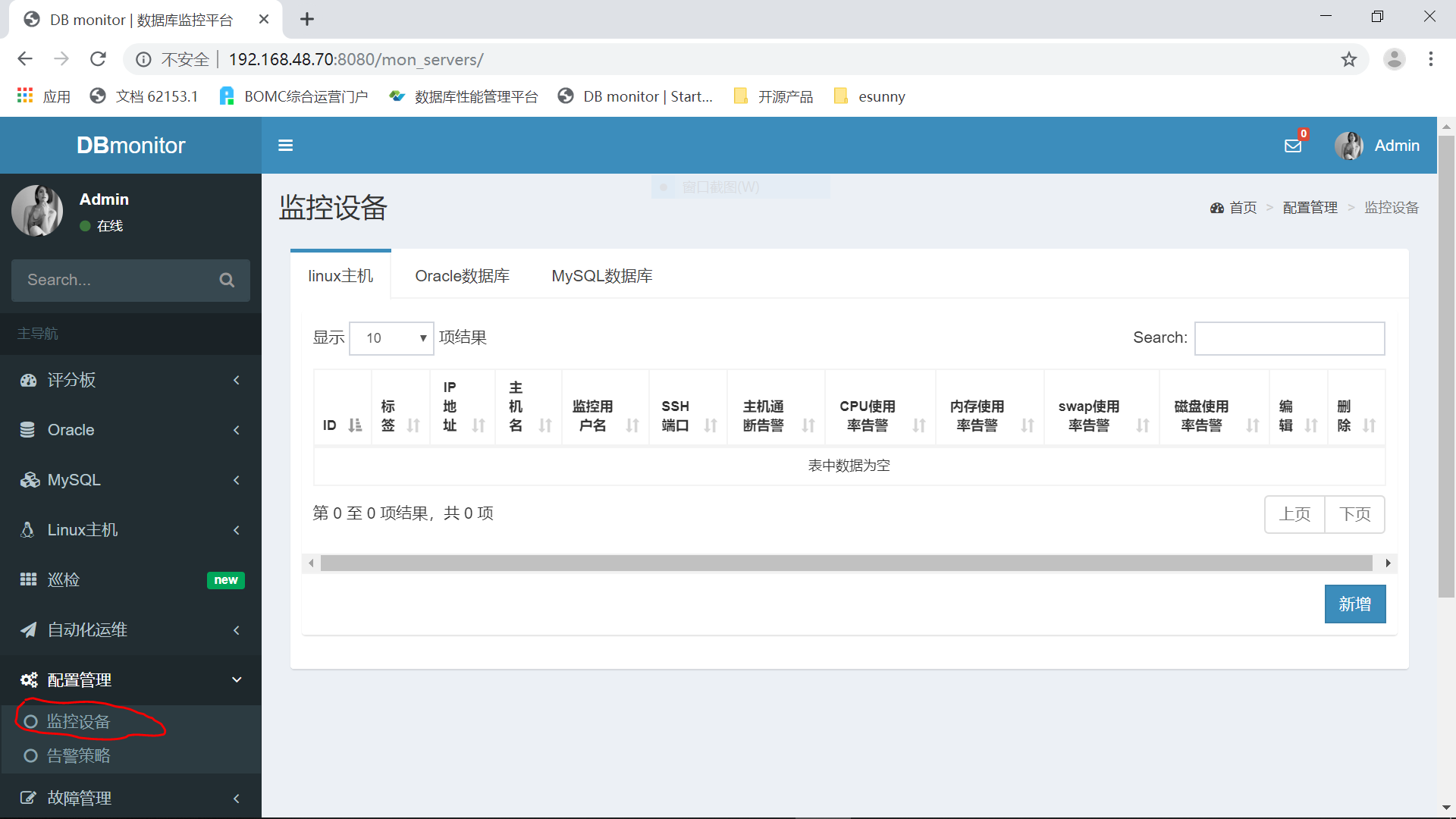
## 七. 操作指南

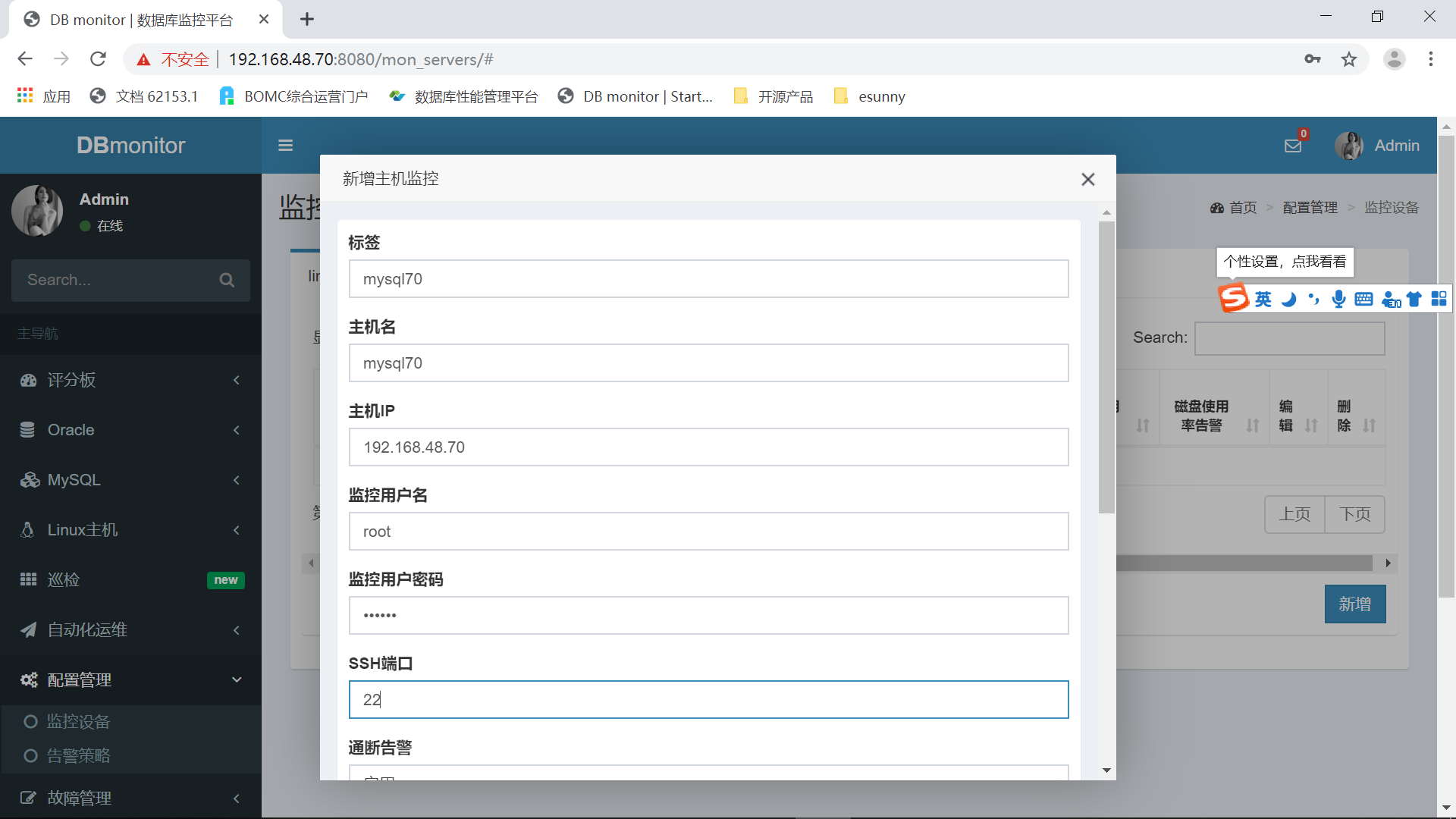
### 1. 添加监控(非Oracle数据库)

**--登录地址为web服务启动时的配置，用户名密码为前面创建管理用户，如**



**--添加监控，标签为监控系统中唯一标识，不建议冲突，可自行选择是否启用具体告警项**





**--点击提交即可，添加成功后采集程序可自动发现新添加主机，不需要重新启动**



### 2. 添加监控(Oracle数据库)

**--监控Oracle数据库需要先在目标数据库中创建监控用户**

SQL> create user esmon identified by oracle;

User created.

**--因为需要调用生成awr的包，建议直接授予dba权限**

SQL> grant dba to esmon;

Grant succeeded.

SQL> grant select any dictionary to esmon;

Grant succeeded.

--12c版本需要单独对表空间使用授权

SQL> ALTER USER esmon quota unlimited on users;

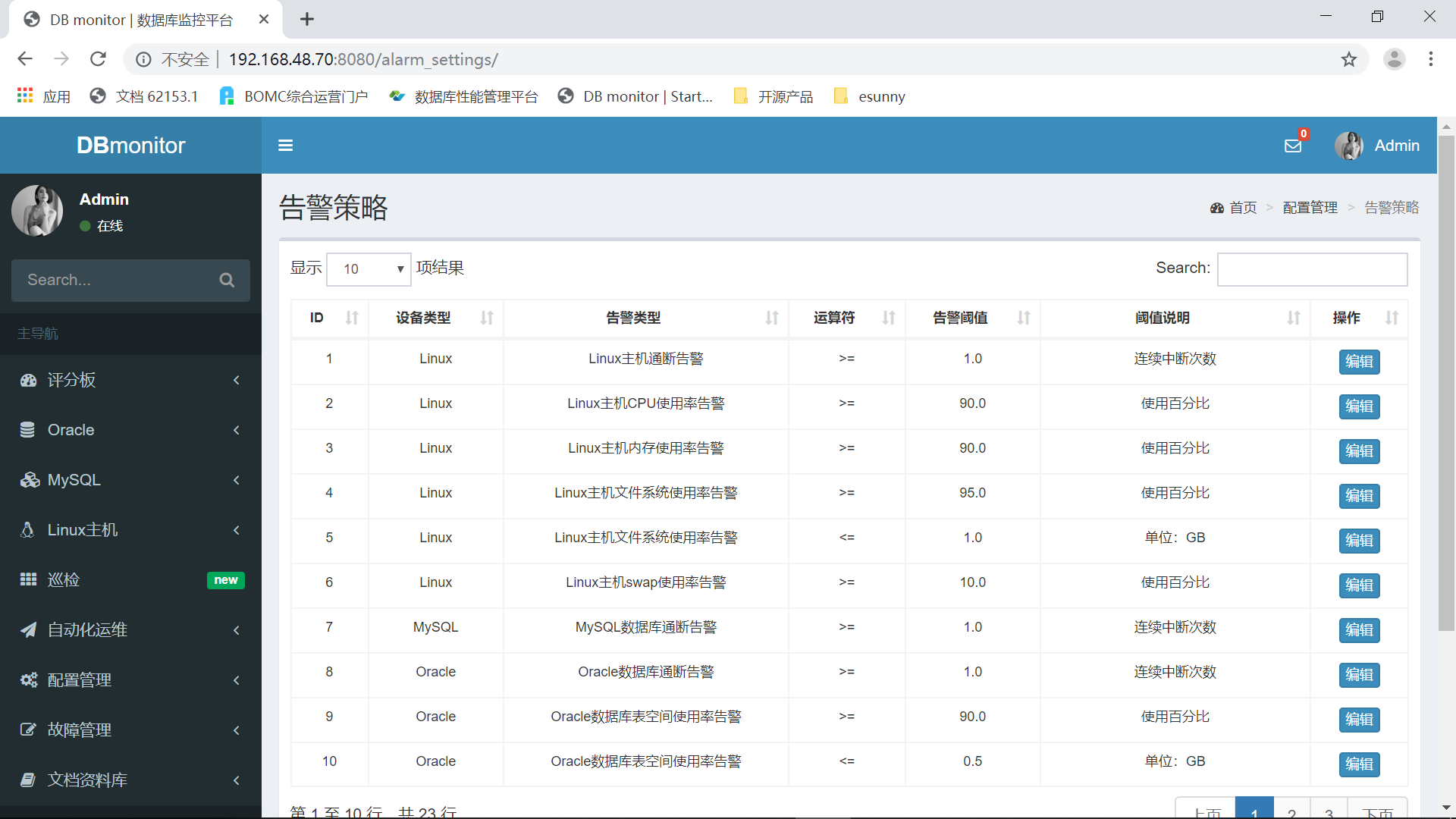
User altered.

**--执行脚本(目标数据库监控用户)**

setup/oracle/procedure.sql & setup/oracle/table.sql

**--接下来在web页面添加监控即可**

### 2. 监控配置，配置管理->告警策略，可对监控阈值等做自定义配置



### 3. 系统设置->采集周期，可对数据采集周期做自定义配置

