#1 重置ctp\_s这台机器

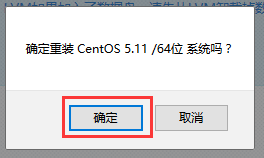


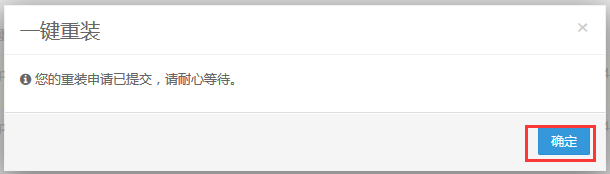
ctp\_s的信息如下所示：



重置过程中，进行如下选择：





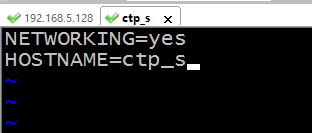


状态显示为"重置中"，如下所示：

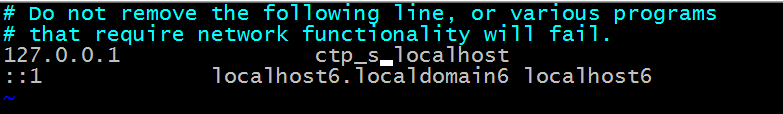


#2 修改主机名

vim /etc/sysconfig/network



vim /etc/hosts



hostname ctp\_s

创建/data目录

以root身份执行  mkdir /data

cd /data

mkfs.ext3 /dev/sdb1 ---- 格式化/dev/sdb1这块硬盘

mount /dev/sdb1  /data    ----临时挂载/dev/sdb1到/data目录下

vim /etc/fstab

末尾添加

/dev/sdb1 /data  ext3  defaults  1 2

mount -a ----测试一下

#3 上传相关文件

以root身份，rz -be 上传

unzip upload.zip

#4 开始安装MySQL

root用户，安装环境

yum install -y make gcc-c++ cmake bison-devel ncurses-devel gcc kernel-devel readline-devel pcre-devel openssl-devel openssl zlib zlib-devel bison

解压缩上传的压缩包：

tar -zxvf mysql-xxx

cd mysql-5.6.22

编译&安装过程：

cmake \

-DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql \

-DMYSQL\_DATADIR=/mydata/data  \

-DSYSCONFDIR=/etc \

-DWITH\_MYISAM\_STORAGE\_ENGINE=1 \

-DWITH\_INNOBASE\_STORAGE\_ENGINE=1 \

-DWITH\_MEMORY\_STORAGE\_ENGINE=1 \

-DWITH\_READLINE=1 \

-DMYSQL\_UNIX\_ADDR=/var/lib/mysql/mysql.sock \

-DMYSQL\_TCP\_PORT=3306 \

-DENABLED\_LOCAL\_INFILE=1 \

-DWITH\_PARTITION\_STORAGE\_ENGINE=1 \

-DEXTRA\_CHARSETS=all \

-DDEFAULT\_CHARSET=utf8 \

-DDEFAULT\_COLLATION=utf8\_general\_ci

cmake执行结束后

执行make && make install

cd /data/usr/local/mysql

scripts/mysql\_install\_db --basedir=/data/usr/local/mysql --datadir=/data/usr/local/mysql/data --user=ctpower   ------- 初始化数据库

mysql\_security\_installation ----初始化配置，如果没有软连接的话，需要从安装路径下执行

root用户

cp mysql.server /etc/init.d/mysql

chkconfig mysql on

service mysql start

更改Mysql管理员密码：

shell中执行mysql进入MySQL程序，接下来执行：

grant all on \*.\* to root@"%" identified by "Ctsi@2016";

flush privileges;

exit;    退出

至此，MySQL安装结束

#5 开始安装Nginx

解压缩  tar -zxvf nginx-xxx

cd nginx-xxxx

./configure --prefix=/data/usr/local/nginx --with-openssl=/data/usr/local/openssl/     ----- 编译安装

/data/usr/local/nginx/sbin/nginx   ---- 启动nginx服务

/data/usr/local/nginx/sbin/nginx -s reload ---- 重启nginx服务

然后做一个软连接

ln -s /data/usr/local/ngnix/sbin/ngnix /usr/bin/

开机自启动部分：

vim /etc/init.d/nginx

输入如下内容：

============================================================

#!/bin/bash

# nginx Startup script for the Nginx HTTP Server

# it is v.0.0.2 version.

# chkconfig: - 85 15

# description: Nginx is a high-performance web and proxy server.

#              It has a lot of features, but it's not for everyone.

# processname: nginx

# pidfile: /var/run/nginx.pid

# config: /usr/local/nginx/conf/nginx.conf

nginxd=/data/usr/local/nginx/sbin/nginx

nginx\_config=/data/usr/local/nginx/conf/nginx.conf

nginx\_pid=/data/usr/local/nginx/logs/nginx.pid

RETVAL=0

prog="nginx"

# Source function library.

. /etc/rc.d/init.d/functions

# Source networking configuration.

. /etc/sysconfig/network

# Check that networking is up.

[ ${NETWORKING} = "no" ] && exit 0

[ -x $nginxd ] || exit 0

# Start nginx daemons functions.

start() {

if [ -e $nginx\_pid ];then

   echo "nginx already running...."

   exit 1

fi

   echo -n $"Starting $prog: "

   daemon $nginxd -c ${nginx\_config}

   RETVAL=$?

   echo

   [ $RETVAL = 0 ] && touch /var/lock/subsys/nginx

   return $RETVAL

}

# Stop nginx daemons functions.

stop() {

        echo -n $"Stopping $prog: "

        killproc $nginxd

        RETVAL=$?

        echo

        [ $RETVAL = 0 ] && rm -f /var/lock/subsys/nginx /var/run/nginx.pid

}

# reload nginx service functions.

reload() {

    echo -n $"Reloading $prog: "

    #kill -HUP `cat ${nginx\_pid}`

    killproc $nginxd -HUP

    RETVAL=$?

    echo

}

# See how we were called.

case "$1" in

start)

        start

        ;;

stop)

        stop

        ;;

reload)

        reload

        ;;

restart)

        stop

        start

        ;;

status)

        status $prog

        RETVAL=$?

        ;;

\*)

        echo $"Usage: $prog {start|stop|restart|reload|status|help}"

        exit 1

esac

exit $RETVAL

============================================================

chmod a+x /etc/init.d/nginx   ----- 赋予执行权限

测试结果：

[root@ctp\_s init.d]# /etc/init.d/nginx start   ---- 开启nginx

nginx already running....          ------测试结果，一切正常

然后执行指令 vim /etc/rc.local

添加一行 /etc/init.d/nginx start

保存退出，下次机器重启的时候，可以自己启动ngix服务

至此，Nginx安装完毕

#6安装PHP环境

解压缩，然后cd到目录中

tar -zxvf php-5.6.18.tar.gz

cd php-5.6.18

解决依赖关系，如下所示：

yum -y install libxml2\*

 yum -y install curl-devel

yum -y install libjpeg-devel

yum -y install libpng-devel

yum -y install libmcrypt\*

yum -y install bzip2-devel

./configure --prefix=/data/usr/local/php --enable-fpm --with-mcrypt \

--enable-mbstring --disable-pdo --with-curl --disable-debug --disable-rpath \

--enable-inline-optimization --with-bz2 --with-zlib --enable-sockets \

--enable-sysvsem --enable-sysvshm --enable-pcntl --enable-mbregex \

--with-mhash --enable-zip --with-pcre-regex --with-mysql --with-mysqli \

--with-gd --with-jpeg-dir

make && make install

安装完毕后

配置php环境：

将安装包下的php配置文件复制到php安装路径下：

cp /data/php-5.6.18/php.ini-development /data/usr/local/php/lib/php.ini

为php-fpm提供配置文件：

cp /data/usr/local/php/etc/php-fpm.conf.default /data/usr/local/php/etc/php-fpm.conf

配置开机自启动：

cd /data/php-5.6.18/sapi/fpm

cp init.d.php-fpm /etc/init.d/php-fpm

chmod a+x /etc/init.d/php-fpm

chkconfig --add php-fpm

chkconfig php-fpm on

编辑php-fpm配置

vim /data/usr/local/php/etc/php-fpm.conf

pm.max\_children = 150

pm.start\_servers = 8

pm.min\_spare\_servers = 5

pm.max\_spare\_servers = 10

pid = run/php-fpm.pid     ----注意，这里是相对路径表示方法；之前使用绝对路径，导致unknow entry for pid的错误

#7 整合PHP & Nginx：

nginx.conf配置文件修改的说明文档，参考这篇文章：

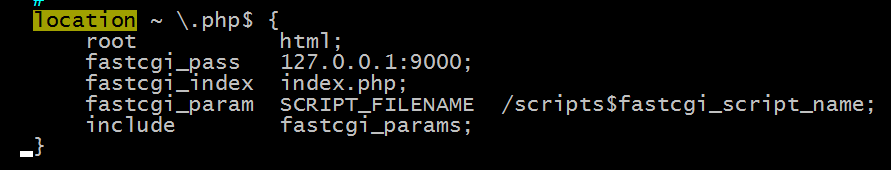
<http://blog.sina.com.cn/s/blog_59d226590101c9y6.html>

<1> 修改Nginx的配置文件

cp /data/usr/local/nginx/conf/nginx.conf /data/usr/local/nginx/conf/nginx.conf.bk

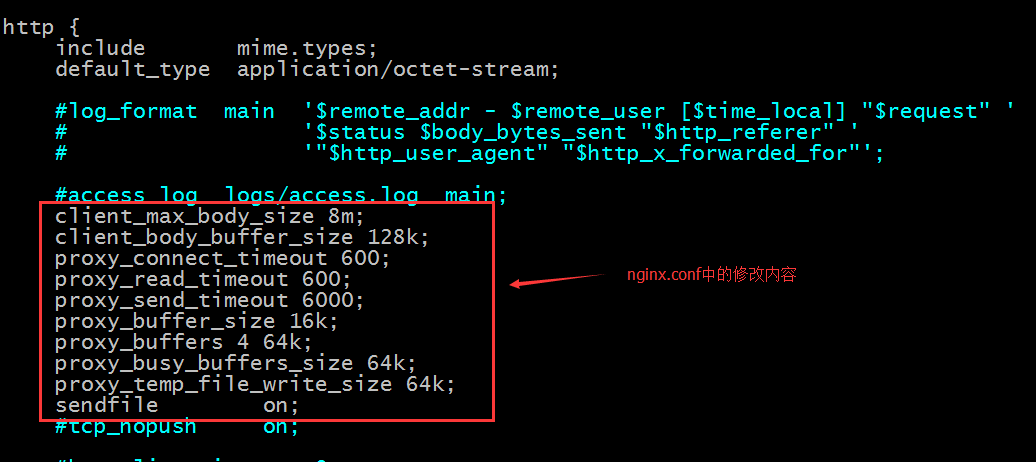
cd /data/usr/local/nginx/conf

vim nginx.conf



重启Nginx，让配置生效：

<2> 第二次修改nginx.conf ---- 按照滨哥提供的更多配置修改，如下所示：



同时，如果运行的是php，还需要进一步的确认PHP的配置php.ini，这个大小client\_max\_body\_size要和php.ini中的如下值的最大值一致或者稍大，这样就不会因为提交数据大小不一致出现的错误。

post\_max\_size = 8M

upload\_max\_filesize = 2M

在ctp\_s机器上执行指令：

vim /data/usr/local/php/lib/php.ini  ---- 进行查看和确认

配置完毕后，重启nginx服务

[root@ctp\_s conf]# /data/usr/local/nginx/sbin/nginx -t    ---- 指令用于测试Nginx的配置是否有效

nginx: the configuration file /data/usr/local/nginx/conf/nginx.conf syntax is ok

nginx: configuration file /data/usr/local/nginx/conf/nginx.conf test is successful

[root@ctp\_s conf]#

重启Nginx指令：

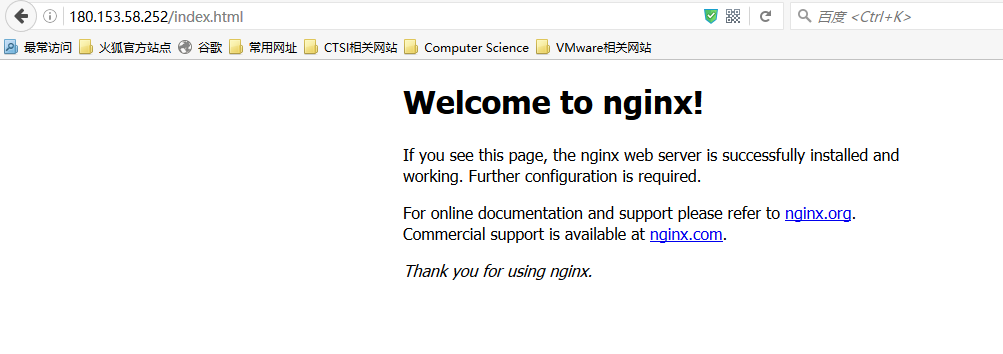
/data/usr/local/nginx/sbin/nginx -s reload

测试：

浏览器中输入网址：

<http://180.153.58.252/index.html>

可以看到Nginx的欢迎页面，如下所示：



#8 安装git

rpm -Uvh <http://dl.fedoraproject.org/pub/epel/5/x86_64/epel-release-5-4.noarch.rpm>

yum install git

安装完毕后，检查一下，如下所示：

[root@ctp\_s local]# whereis git

git: /usr/bin/git /usr/share/man/man1/git.1

[root@ctp\_s local]# git --version

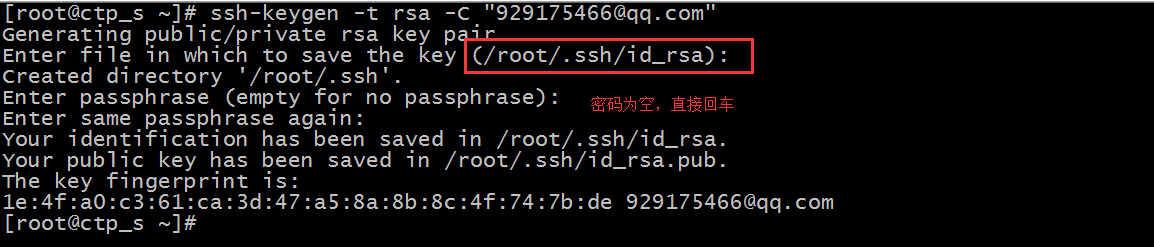
git version 1.8.2.3

git配置过程：

git config --global user.name "929175466"

git config --global user.email "929175466@qq.com"

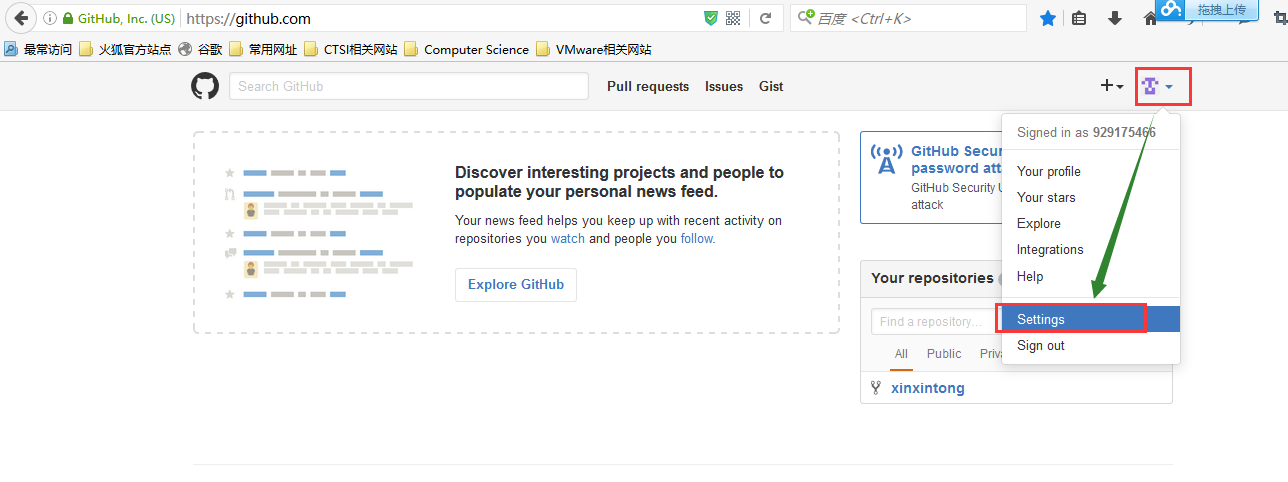
ssh-keygen -t rsa -C "929175466@qq.com"

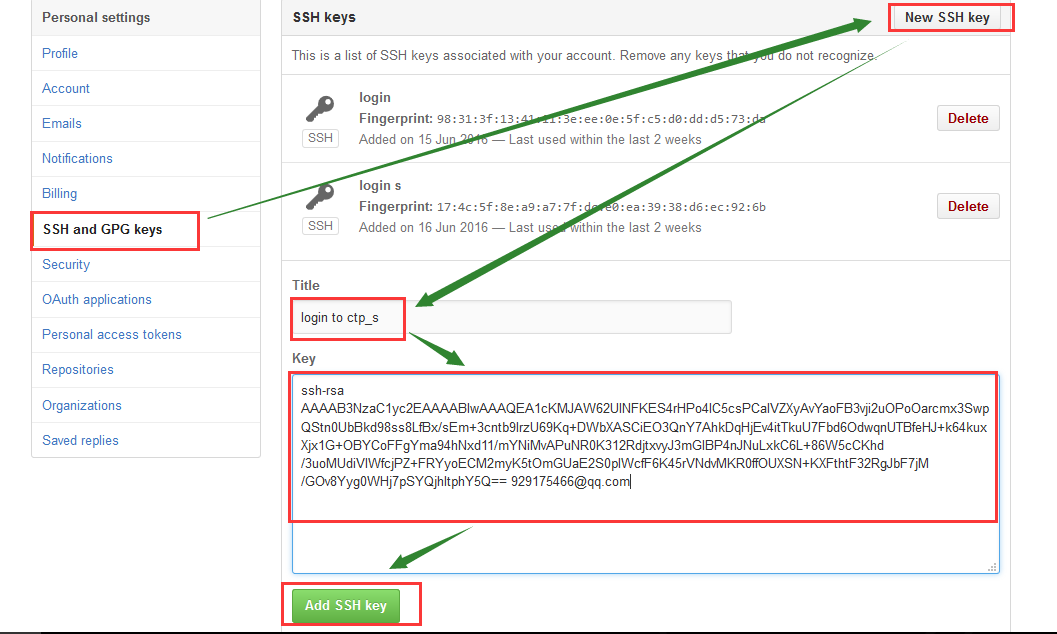


cat ~/.ssh/id\_rsa.pub

将内容复制出来

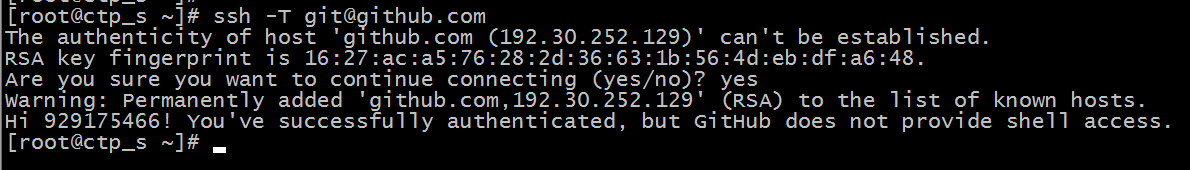
登录Github，如下所示：





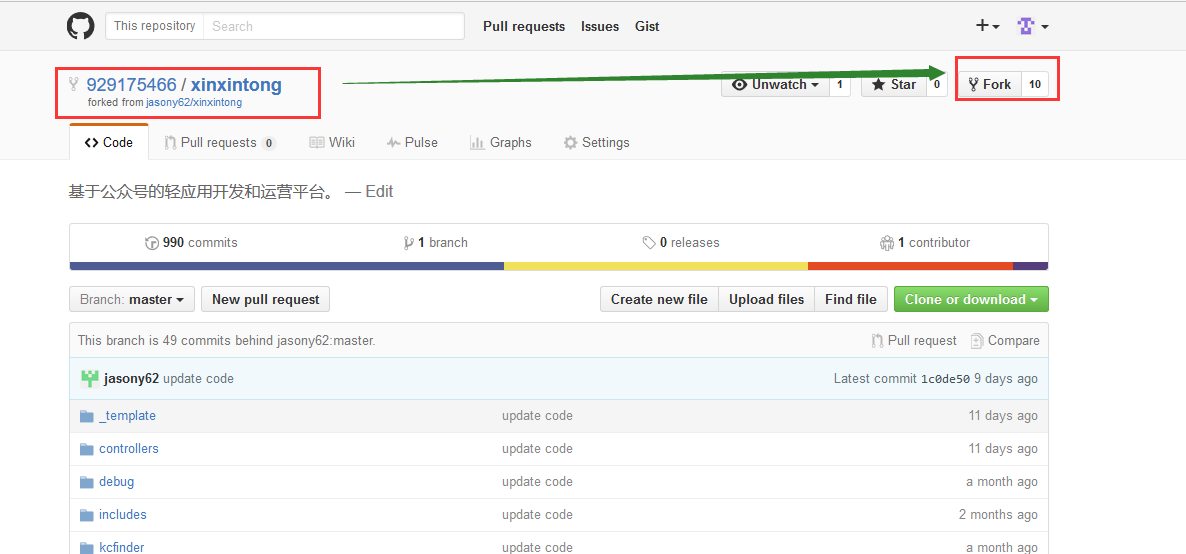
测试部分：

ssh -T [git@github.com](mailto:git@github.com)



Github网页端操作：

登录自己的账号，搜索xinxintong，fork到本账号，如下所示：



server端：

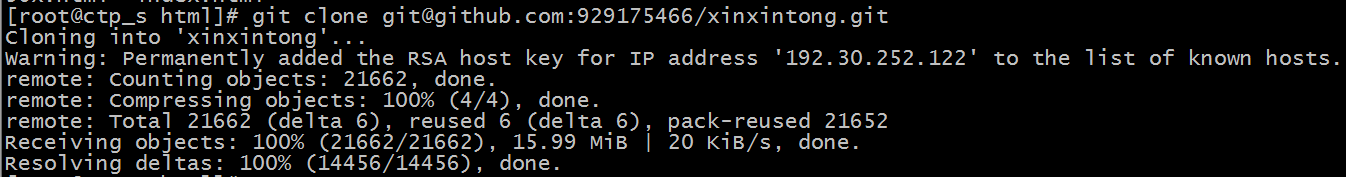
cd /data/usr/local/nginx/html/

git clone [git@github.com:929175466/xinxintong.git](mailto:git@github.com:929175466/xinxintong.git)

git clone [git@github.com:lszgf1234/xinxintong.git](mailto:git@github.com:lszgf1234/xinxintong.git)

git clone git@github.com:lszgf123/xinxintong.git

执行结果如下所示：



接下来

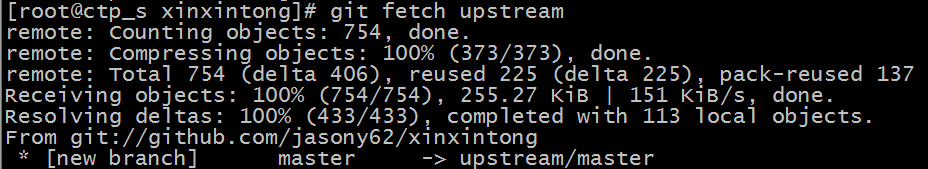
/data/usr/local/nginx/html/xinxintong

git remote add upstream git://github.com/jasony62/xinxintong.git

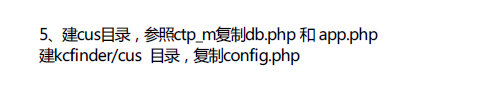
git remote add upstream git://github.com/lszgf1234/xinxintong.git

git fetch upstream

执行结果如下所示：



git merge upstream/master

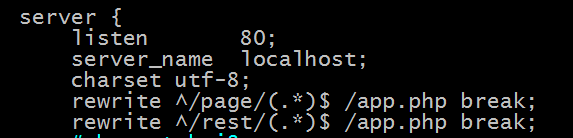


在/data/usr/local/nginx/html目录下创建cus目录，从ctp\_m机器上拷贝db.php & app.php进去

在/data/usr/local/nginx/html/kcfinder目录下，创建cus目录，从ctp\_m机器上对应位置拷贝config.php文件

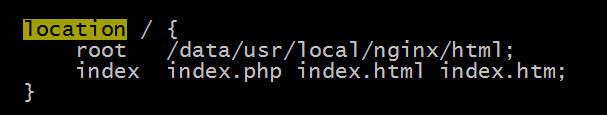
nginx配置文件改动：

<1>

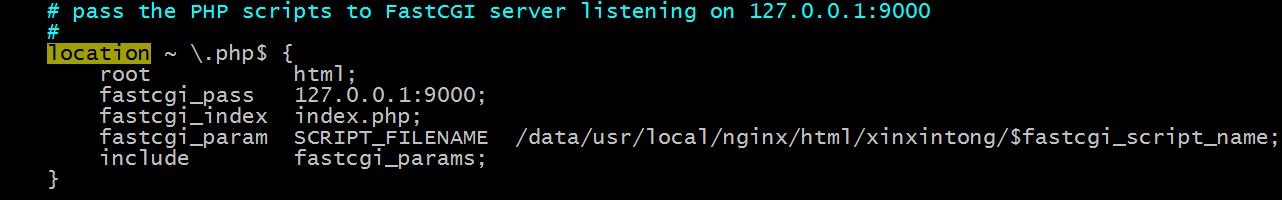


注意，此处的server\_name没有添加developer.189.cn    ----- 如果添加的话，可能会产生冲突

<2>



<3>

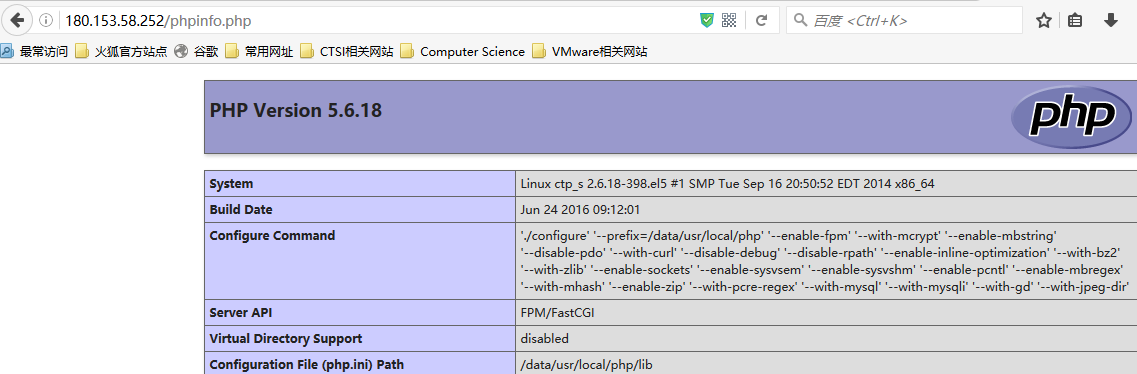


更改完之后重启，

/data/usr/local/nginx/sbin/nginx -s reload

测试：

输入网址：  <http://180.153.58.252/phpinfo.php>



php-fpm相关指令：

<http://www.cnblogs.com/zdz8207/p/3765579.html>

在MySQL中增加ctpower用户名&权限

在shell中执行

mysql -hlocalhost -uroot –p #登录MySQL

初始化网页部分：

输入<http://180.153.58.252/schema/init.php>

提示access denied for ctpower@localhost

接下来，mysql登陆，执行

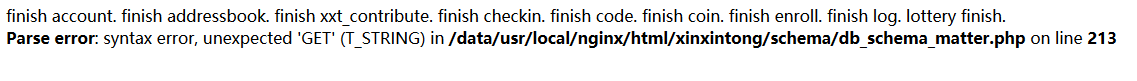
grant all on \*.\* to 'ctpower'@'localhost' identified by 'ctadmin';  ---- 给ctpower这个用户赋予用户名&密码，使其和init.php中的一样

接下来遇到xxt数据库错误，

mysql登陆，

然后create database xxt;  ---- 继续和init.php保持一致

再次执行init.php，遇到如下所示错误：



开发调整了git上边的文件内容，在xinxintong文件夹下，重新执行

git fetch upstream

git merge upstream/master

nginx重启，再次刷新页面，遇到如下所示错误：



启动、关闭三个进程

<1> 关闭、启动、重启MySQL服务

service mysql stop | start | restart

<2> 关闭Nginx服务、重启Nginx服务、启动Nginx服务

/data/usr/local/nginx/sbin/nginx -s stop

/data/usr/local/nginx/sbin/nginx -s reload

/data/usr/local/nginx/sbin/nginx

<3> 关闭、启动、重启PHP服务

service php-fpm stop

service php-fpm start

service php-fpm restart

/etc/my.cnf中配置项详解，参考这篇文章：

<http://www.cnblogs.com/derrck/archive/2010/07/29/1787856.html>

MySQL读取配置文件的位置&顺序，参考这篇文章：

<http://blog.csdn.net/hopingwhite/article/details/4992880>

MySQL的主从复制部分配置：

<1> ctp\_m这台服务器上的/etc/my.cnf配置，如下所示：

[mysqld]

datadir=/data/usr/local/mysql/data

socket=/var/lib/mysql/mysql.sock

user=mysql

log-bin=master-bin            #开启了log-bin

log-bin-index=master-bin.index     #开启了log-bin-index

binlog\_format=mixed         #开启了binlog\_format

server-id = 1             #开启了server-id

innodb\_file\_per\_table = 1

port = 3306

key\_buffer\_size = 256M #此处开始为MySQL配置优化部分

max\_allowed\_packet = 20M #滨哥提供的必须修改的参数值

table\_open\_cache = 256

sort\_buffer\_size = 20M

read\_buffer\_size = 20M

join\_buffer\_size = 32M

read\_rnd\_buffer\_size = 32M

myisam\_sort\_buffer\_size = 128M

thread\_cache\_size = 8

query\_cache\_size = 64M

# Default to using old password format for compatibility with mysql 3.x

# clients (those using the mysqlclient10 compatibility package).

old\_passwords=1

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

# to do so, uncomment this line:

# symbolic-links=0

[mysqld\_safe]

log-error=/var/log/mysqld.log

pid-file=/var/run/mysqld/mysqld.pid

<2> ctp\_s这台机器上的/etc/my.cnf的配置，如下所示：

[mysqld]

port = 3306

key\_buffer\_size = 256M

max\_allowed\_packet = 20M

table\_open\_cache = 256

sort\_buffer\_size = 20M

read\_buffer\_size = 20M

join\_buffer\_size = 32M

read\_rnd\_buffer\_size = 32M

myisam\_sort\_buffer\_size = 128M

thread\_cache\_size = 8

query\_cache\_size = 64M

read-only = ON      #开启了从服务器的只读模式

server-id = 11         #定义了server-id

relay-log = relay-log    #开启了中继日志

relay-log-index = relay-log.index    #开启了中继日志的index

basedir=/data/usr/local/mysql

datadir=/data/usr/local/mysql/data

http://localhost/ctsi1/xinxintong

git push origin dev