基于gitolite自动化版本控制

环境（linux centos 65 x64 两台（192.168.1.112,192.168.1.119），windows一台）

一、192.168.1.119局域网安装gitolite

1. 安装git

Yum install git -y

1. 添加git用户

Adduser git

1. 生成git ssh公约私钥

Su git

Ssh-keygen

将公约推到112服务器上待用

Scp /home/git/.ssh/id\_rsa.pub [root@192.168.1.112:/home/git/119.pub](mailto:root@192.168.1.112:/home/git/119.pub)

1. 安装gitolite

cd

mkdir bin

git clone https://github.com/sitaramc/gitolite.git

./gitolite/install --to /home/git/bin/

bin/gitolite setup -pk local.pub

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*local.pub 可由本地机器在windows下git客户端git bash 生成命令为ssh-keygen，放在C:\Users\Administrator\.ssh\id\_rsa.pub,将其推到119机器上/home/git/local.pub\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 在本地机器Clone管理版本库，并创建新的版本库

Git clone [git@192.168.1.119:gitolite-admin](mailto:git@192.168.1.119:gitolite-admin)

Cd gitolite-admin

Vim conf/gitolite.conf

Repo repo1

RW+ = local

二、在外网测试服务器添加gitolite（192.168.1.112）

1、安装git

Yum install git -y

2、添加git用户

Adduser git

3、安装gitolite

Su git

Cd

mkdir bin

git clone https://github.com/sitaramc/gitolite.git

./gitolite/install --to /home/git/bin/

bin/gitolite setup -pk 119.pub

三、在局域网服务器（192.168.1.119）

1、Clone管理版本库，并创建新的版本库

Git clone [git@192.168.1.119:gitolite-admin](mailto:git@192.168.1.119:gitolite-admin)

Cd gitolite-admin

Vim conf/gitolite.conf

Repo repo1

RW+ = 119

2、配置ssh验证

Vim /home/git/.ssh/config

Su root

Chmod /home/git/.ssh/config 600

Su git

Cd

Host git.server(可以自定义)

HostName 192.168.1.112

Port 22

User git

IdentityFile ~/.ssh/id\_rsa

1. 添加远程版本库地址

Cd /home/git/repositories/repo1.git

git remote set-url git.server [git@git.server:repo1](mailto:git@git.server:repo1)

3、添加git钩子同步版本到112外网服务器

Cd /home/git/repositories/repo1.git/hooks

cp post-receive.sample post-receive

Vim post-receive

while read oldrev newrev ref

do

echo "STATRT $oldrev $newrev $ref";

branch=${ref##refs/heads/};

git push git.server $branch;

done;

1. 在外网服务器（192.168.1.112）部署

1、添加发布钩子

Cd /home/git/repositories/repo1.git/hooks

cp post-receive.sample post-receive

vim post-receive

while read oldrev newrev ref

do

echo "START $oldrev $newrev $ref";

#if [[ $ref =~ "refs/tags/" ]]; then

git --git-dir=/home/git/repositories/repo1.git diff ${oldrev}...${newrev} > /tmp/diff.txt;

branch=${ref##refs/heads/};

if [[ -n "$branch" ]]; then

echo "$branch +++"

/repo1/dev.config/deploy /repo1/dev/ ${branch} > /repo1/dev/data/deploy.log;

else

echo "$branch ---"

/repo1/dev.config/deploy /repo1/dev/ $newrev > /repo1/dev/data/deploy.log;

fi

#echo 'deploy dev done';

#fi

done;

2、创建项目

Mkdir /repo1

Mkdir /repo1/dev

Mkdir /repo1/dev.config

Cd /repo1/dev.config

Vim deploy

#!/bin/sh  
  
usage="usage:deploy src\_path branch update\_node";  
  
src\_path=$1;  
selected\_branch=$2;  
update\_node=$3;  
current\_branch="";  
git\_dir="/home/git/repositories/repo1.git";  
git\_work\_tree=$src\_path;  
diff\_file="/tmp/diff.txt";  
deploy\_file="$src\_pathdata/deploy.log";  
deploy\_lock="$src\_pathdata/deploy.lock";  
config\_dir="/repo1/dev.config/\*";  
  
  
if [[ "$src\_path" == "-h" ]]; then  
 echo $usage;  
 exit;  
fi  
  
if [[ $# < 2 ]]; then  
 echo $usage;  
 exit;  
fi  
  
if [[ ! -d $src\_path ]]; then  
 echo $src\_path;  
 echo "source path is not exists.keep the path is absolute path";  
 exit;  
fi  
  
function deploy()  
   
 # 发送diff邮件  
 # 记录当前的分支  
 # current\_hash;  
 if [[ -f $deploy\_lock ]]; then  
 locked\_branch=`cat $deploy\_lock | cut -d \| -f 1`;  
 locked=`cat $deploy\_lock | cut -d \| -f 2`;  
 if [[ $locked -gt 0 && "$locked\_branch" != "$selected\_branch" ]]; then  
 echo 'auto deploy has LOCKED,if you WANT TO DEPLOY,visit /deploy.php to deploy';  
 return 2;  
 fi  
 fi  
 # 旧版本hash值  
 echo "current deploy branch：$selected\_branch";  
  
 # 先初始化部分依赖 1.nodejs module  
 if [[ -n $update\_node ]]; then  
 init;  
 fi  
   
 git --git-dir=$git\_dir --work-tree=$git\_work\_tree checkout -f $selected\_branch && git --git-dir=$git\_dir --work-tree=$git\_work\_tree reset --hard;  
  
 chmod -R 755 $src\_path;  
   
 # 生成配置文件  
 generate\_config;  
 # copy 配置文件及其他文件  
 echo "copy config files";  
 #node $src\_pathnode\_modules/.bin/gulp --cwd $src\_path init;  
 /bin/cp -rf $config\_dir $src\_path;  
 # 执行migrate数据库自动化脚本  
 migrate;  
 # 执行gulp自动化脚本  
 #echo "GULP START";  
 #node $src\_pathnode\_modules/.bin/gulp --cwd $src\_path dev;  
 #echo "GULP END";  
 # 生成配置文件  
 generate\_config;  
  
 return 0;  
  
  
function diff\_email()  
 if [[ -f $send\_file ]]; then  
 /usr/bin/python /home/git/python/send\_email.py $send\_file;  
 echo "$diff\_file";  
 #rm -f $diff\_file;  
 fi  
  
  
function deploy\_email()  
 if [[ -f $deploy\_file ]]; then  
 node $src\_pathnode\_modules/.bin/gulp --cwd $src\_path deploy\_email;  
 fi  
  
  
function current\_hash()  
 HEAD=$(cat $src\_path.git/HEAD);  
 if [[ $HEAD =~ "ref:" ]]; then  
 ref=$HEAD##ref: ;  
 current\_branch=$(cat $src\_path.git/$ref);  
 else  
 current\_branch=$HEAD;  
 fi  
  
  
function uniqid()  
 echo `/alidata/server/php/bin/php -r "echo uniqid();"`;  
  
  
function init()  
 npm install;  
 npm update;  
  
  
function rollback()  
 git checkout -f $current\_hash;  
 echo 'rollback success';  
  
  
function migrate()  
 /alidata/server/php/bin/php -c /alidata/server/php-5.4.23/etc/php.ini -f $src\_pathapplication/migrate.php;  
  
  
function gulp\_script()  
 echo 'gulp start';  
   
 echo 'gulp end';  
  
  
function generate\_config()  
 config\_file=$src\_pathapplication/config/env.php;  
 configs=("\$config['version']='`uniqid`';");  
 echo "<?php" > $config\_file;  
 for config in $configs; do  
 echo "" >> $config\_file;  
 echo $config >> $config\_file;  
 done  
 echo 'generate version config done';  
  
  
deploy;  
  
if [[ $? == 0 ]]; then  
 echo 'deploy done';  
elif [[ $? == 2 ]]; then  
 echo 'push doen';  
else  
 # 如果出错,则回滚  
 if [[ -f $backup\_file ]]; then  
 rollback;  
 fi  
 echo 'deploy failed';  
fi

Vim deploy.php

<?php  
  
header('Content-type:text/html;charset=utf-8');  
  
error\_reporting(E\_ALL & ~E\_NOTICE);  
ini\_set('display\_error', 'Off');  
// error\_reporting(E\_ALL);  
  
date\_default\_timezone\_set('Asia/Shanghai');  
  
$src\_path="/home/git/repositories/cunguan.git/";  
$dest\_path="/repo1/dev/";  
$deploy\_output\_log="$dest\_pathdata/deploy.log";  
$deploy\_lock\_file="$dest\_pathdata/deploy.lock";  
  
if ($\_SERVER['REQUEST\_METHOD']=='POST')   
 $selected\_branch=trim($\_POST['branch']);  
 if ($selected\_branch)   
 exec($dest\_path.“/dev/deploy $dest\_path $selected\_branch > $deploy\_output\_log &",$output);  
 @file\_put\_contents($deploy\_lock\_file,sprintf('%s|%s',$selected\_branch,(int)$\_POST['locked']));  
 header("Location:/deploy.php");  
   
  
  
exec("git --git-dir=$src\_path branch",$branches,$status);  
  
$branches=array\_map(function($value)  
 return trim($value,"\* ");  
, $branches);  
  
list($deploy\_branch,$locked)=@explode('|', @file\_get\_contents($deploy\_lock\_file));  
  
$deploy\_output=@file\_get\_contents($deploy\_output\_log);  
  
?>  
  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>deploy</title>  
 <style type="text/css">  
 body  
 width:960px;  
 margin:auto;  
 padding:0;  
 font-size:14px;  
   
 .control-group  
 width:100%;  
 margin:20px 15px;  
   
 label  
 font-size:16px;  
 display:inline-block;  
 height:22px;  
 line-height:22px;  
 margin-top:15px;  
 margin-left:20px;  
   
 .branch-label  
  
   
 select  
 width:220px;  
 line-height:22px;  
   
 textarea  
 width: 600px;  
 height: 300px;  
 font-size:14px;  
 line-height:20px;  
   
 .deploy-result  
 vertical-align: top;  
 position: relative;  
 top: -10px;  
   
 .deploy-btn  
 display:inline-block;  
 height:22px;  
 line-height:22px;  
 font-size:16px;  
 width:60px;  
   
 .deploy-btn:first-child  
 margin-left:60px;  
   
 .deploy-btn:last-child  
 margin-left:10px;  
   
 </style>  
</head>  
<body>  
 <form class="form-horizontal" method="POST">  
 <div class="control-group">  
 <label class="control-label branch-label" style="display: inline-block;width: 80px;">分支</label>  
 <select class="input-xlarge" name="branch" onchange="change\_branch(this)">  
 <?php foreach ($branches as $k=>$branch) ?>  
 <?php if (trim($deploy\_branch)==trim(str\_replace('\*','',$branch))) ?>  
 <option selected name="<?php echo $branch; ?>"><?php echo $branch; ?></option>  
 <?php else ?>  
 <option name="<?php echo $branch; ?>"><?php echo $branch; ?></option>  
 <?php ?>  
 <?php ?>  
 </select>  
 </div>  
 <div class="control-group">  
 <label class="control-label branch-label" style="display: inline-block;width: 80px;">锁定分支</label>  
 <input type="checkbox" name="locked" value="1" <?php if($locked) echo 'checked';?> />  
 <span id="locked\_branch"><?php echo $deploy\_branch;?></span>  
 <em style="color:red;font-size: 11px;margin-left: 60px;">会限制其他分支push时自动发版流程</em>  
 </div>  
 <div class="control-group">  
 <label class="control-label deploy-result" style="display: inline-block;width: 80px;">结果<br/><em style="color:red;font-size: 11px;">刷新看结果</em></label>  
 <textarea type="" class="" style="margin-left: 0px; margin-right: 0px; width: 450px;" heigth="300"><?php echo $deploy\_output; ?></textarea>  
 </div>  
 <div class="control-group" style="display: inline-block;margin-left: 80px;">  
 <input class="deploy-btn" type="submit" value="发版" />  
 <input class="deploy-btn" type="button" onclick="javascript:document.location.reload()" value="刷新" />  
 </div>  
 </form>  
 <script>  
 function change\_branch (that)   
 document.getElementById("locked\_branch").innerHTML=that.value;  
   
 </script>  
</body>  
</html>

1. deploy.php 文件权限控制

在有apache的服务器上生成密码htpasswd -bc 文件名（passwd.db） 用户名 密码

或者在线生成http://tool.oschina.net/htpasswd，算法选择MD5 (Apache servers only)

配置nginx

location ~ deploy.php   
  
 auth\_basic "Restricted";  
 auth\_basic\_user\_file /alidata/server/nginx-1.4.4/conf/passwd.db;  
 fastcgi\_pass 127.0.0.1:9000;  
 fastcgi\_index index.php;  
 fastcgi\_connect\_timeout 200;  
 fastcgi\_read\_timeout 200;  
 fastcgi\_send\_timeout 200;  
 fastcgi\_buffer\_size 128k;  
 fastcgi\_buffers 4 128k;  
 fastcgi\_busy\_buffers\_size 256k;  
 fastcgi\_temp\_file\_write\_size 256k;  
 #set\_real\_ip\_from 100.97.0.0/16;  
 #real\_ip\_header X-Forwarded-For;  
 fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;  
 include fastcgi\_params;

1. 附件

涉及到的其它文件

外网测试服务器上

Cd /root/python

Vim send\_email.py

#!/usr/bin/python  
# -\*- coding:utf-8 -\*-  
import smtplib,sys,os  
from email.mime.text import MIMEText   
import email.mime.multipart  
from email.MIMEMultipart import MIMEMultipart  
from email.MIMEBase import MIMEBase  
from email import Encoders  
def send\_mail():   
 mailto\_list = [xxx@qq.com'] #收件人  
 mail\_host = "smtp.163.com" # 设置服务器  
 mail\_user = "ksx\_xxx@163.com" # 用户名  
 mail\_pass = "xxxpwd" # 口令   
 mail\_postfix = "163.com" # 发件箱的后缀  
 me = "hello" + "<ksx\_xxx@" + mail\_postfix + ">" # 这里的hello可以任意设置，收到信后，将按照设置显示  
 content = '可以下载下来进行查看'#邮件正文  
 content = content+"<br/>"  
   
 fh = open(sys.argv[1])  
 for line in fh.readlines():   
 content = content+line+"<br/>"  
 msg = MIMEMultipart()  
 body = MIMEText(content, \_subtype='html', \_charset='utf-8') # 创建一个实例，这里设置为html格式邮件  
 msg.attach(body)  
 msg['Subject'] = "项目代码提交记录通知" # 设置主题  
 msg['From'] = me   
 msg['To'] = ";".join(mailto\_list)   
 #附件内容，若有多个附件，就添加多个part, 如part1，part2，part3  
 part = MIMEBase('application', 'octet-stream')  
 # 读入文件内容并格式化，此处文件为当前目录下，也可指定目录 例如：open(r'/tmp/123.txt','rb')  
 part.set\_payload(open(sys.argv[1],'rb').read())  
 Encoders.encode\_base64(part)  
 ## 设置附件头  
 part.add\_header('Content-Disposition', 'attachment; filename="diff.txt"')  
 msg.attach(part)  
   
 try:   
 s = smtplib.SMTP()   
 s.connect(mail\_host) # 连接smtp服务器  
 s.login(mail\_user, mail\_pass) # 登陆服务器  
 s.sendmail(me, mailto\_list, msg.as\_string()) # 发送邮件  
 s.close()   
 os.system("rm -rf %s"%(sys.argv[1]))   
 print "connect success"  
 return True   
 except Exception, e:   
 print "connect failed"  
 return False  
send\_mail()

Migrate.php sql同步脚本

<?php

if(php\_sapi\_name() !== 'cli' && !defined('STDIN')) exit('only run on cli mode');

define("BASEPATH",dirname(\_\_FILE\_\_));

global $db;

include\_once(BASEPATH.'/config/database.php');

$master=$db['db\_main'];

$master\_db=new mysqli($master['hostname'], $master['username'], $master['password'], $master['database']);

if(!$master\_db){

echo 'can not connect mysql'.PHP\_EOL;

exit;

}

mysqli\_set\_charset($master\_db,'UTF8');

$result=mysqli\_query($master\_db,"SELECT \* FROM ".$master['dbprefix']."migration");

if(!$result){

$sql="CREATE TABLE `".$master['dbprefix']."migration` (`id` INT AUTO\_INCREMENT PRIMARY KEY,filename varchar(64) NOT NULL,`version` varchar(64) NOT NULL,`run\_time` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP) ENGINE=InnoDB DEFAULT CHARSET=utf8;";

$status=mysqli\_query($master\_db,$sql);

$result=mysqli\_query($master\_db,"SELECT \* FROM ".$master['dbprefix']."migration");

}

mysqli\_autocommit($master\_db,FALSE);

$migrations=array();

while ($row=mysqli\_fetch\_assoc($result)) {

$migrations[$row['version']]=$row;

}

$versions=array\_keys($migrations);

echo 'MIGRATION START'.PHP\_EOL;

$php\_files = glob(BASEPATH. '/migration/\*.php');

foreach ($php\_files as $file\_path) {

if (preg\_match('/([\_a-zA-Z0-9\-]\*).php/', basename($file\_path),$matches)) {

$filename=$matches[1];

$mcrypt=sha1($matches[1]);

if(!in\_array($mcrypt,$versions)){

$trans=mysqli\_query($master\_db,"START TRANSACTION");

if(!$trans){

echo "dont support transaction".PHP\_EOL;

}

global $sql;

$sql=array();

include $file\_path;

$success=true;

foreach ($sql as $k => $v) {

echo "EXEC SQL : ".$v.PHP\_EOL;

if(mysqli\_query($master\_db,$v,MYSQLI\_USE\_RESULT)===FALSE){

echo "SQL ERROR：".$v.PHP\_EOL;

$success=false;

mysqli\_rollback($master\_db);

break;

}

}

if(!$success) continue;

if(mysqli\_query($master\_db,"INSERT INTO ".$master['dbprefix']."migration SET filename='".$filename."',version='".$mcrypt."'",MYSQLI\_USE\_RESULT)===FALSE){

echo "update migration error".PHP\_EOL;

mysqli\_rollback($master\_db);

continue;

}

mysqli\_commit($master\_db);

}

}

}

echo 'MIGRATION END'.PHP\_EOL;