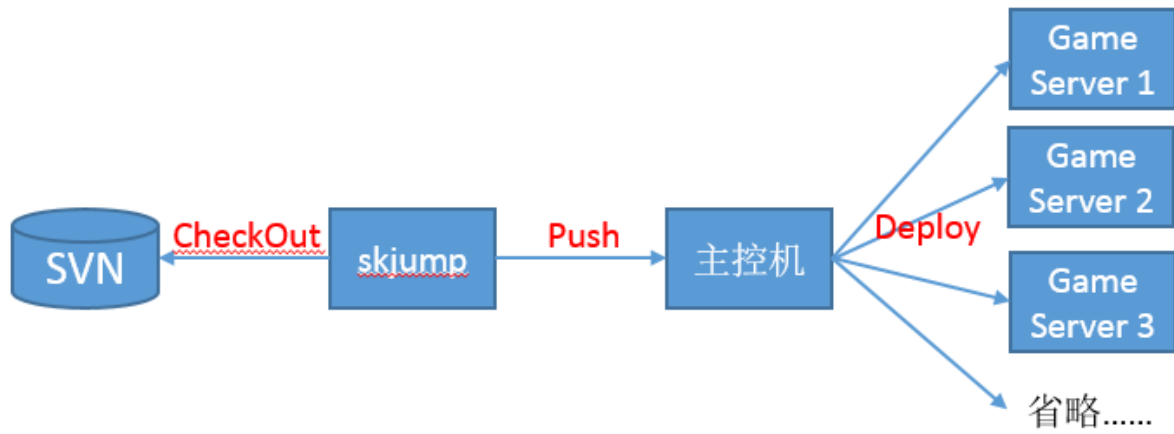


- 1 更新包部署
 - 1.1 部署流程图
 - 1.2 部署准备
 - 1.2.1 本机配置svn
 - 1.2.2 跳板机svn配置
 - 1.3 操作流程
- 2 脚本所涉及到的文件及目录
 - 2.1 Client、Serverr所在目录
 - 2.2 更新补丁存放目录
 - 2.3 数据库备份文件存放目录
 - 2.4 skctl.sh所使用的数据库信息文件
- 3 skctl.sh
 - 3.1 脚本依赖
 - 3.2 使用说明
 - 3.3 注意事项
- 4 skjump.sh
 - 4.1 脚本依赖
 - 4.2 使用说明
- 5 db_all_full.sh
 - 5.1 脚本依赖
 - 5.2 使用说明
- 6 db_single_full.sh
 - 6.1 脚本依赖
 - 6.2 使用说明
- 7 db_single_diff.sh
 - 7.1 脚本依赖
 - 7.2 使用说明

1 更新包部署

1.1 部署流程图

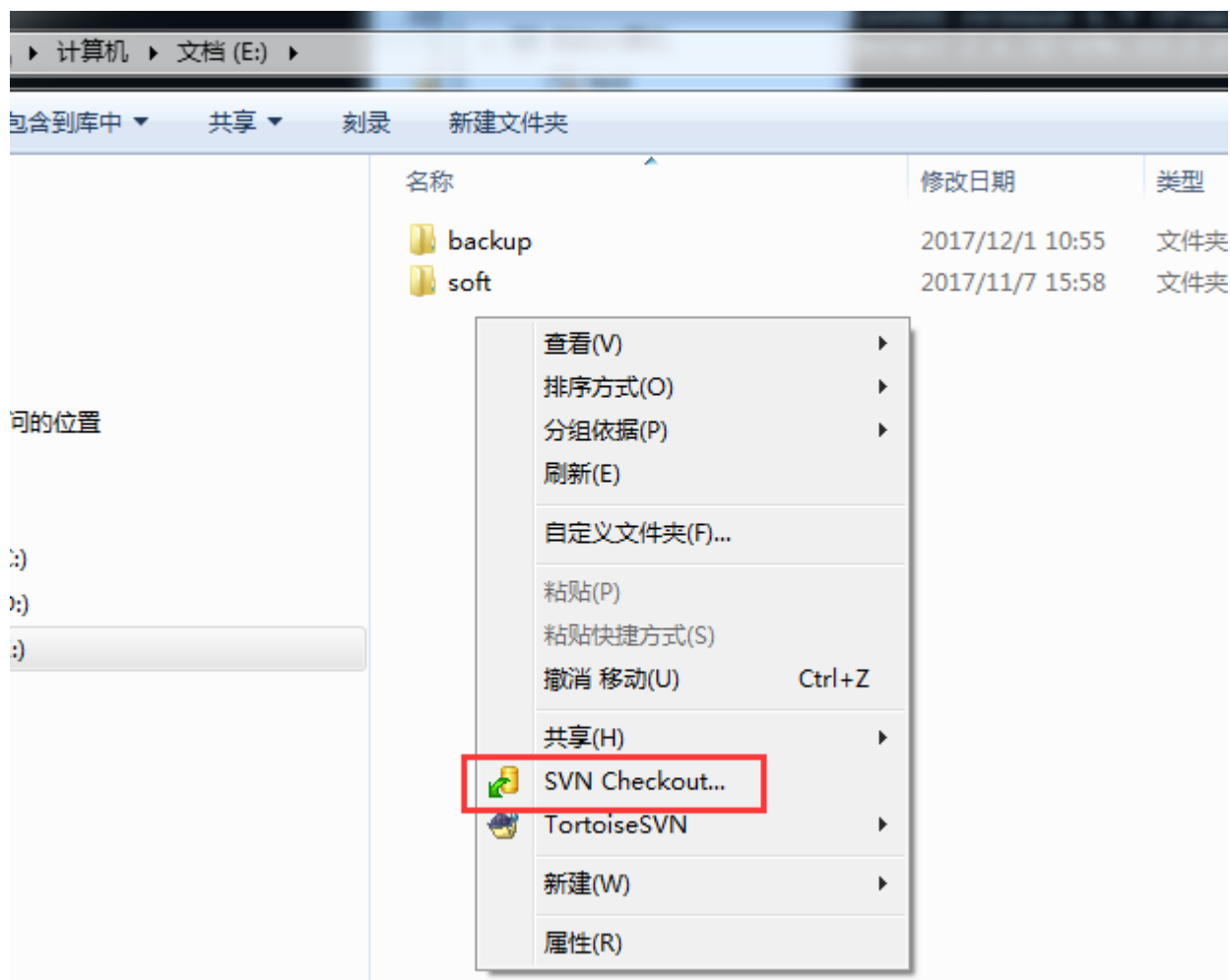
更新包部署流程图



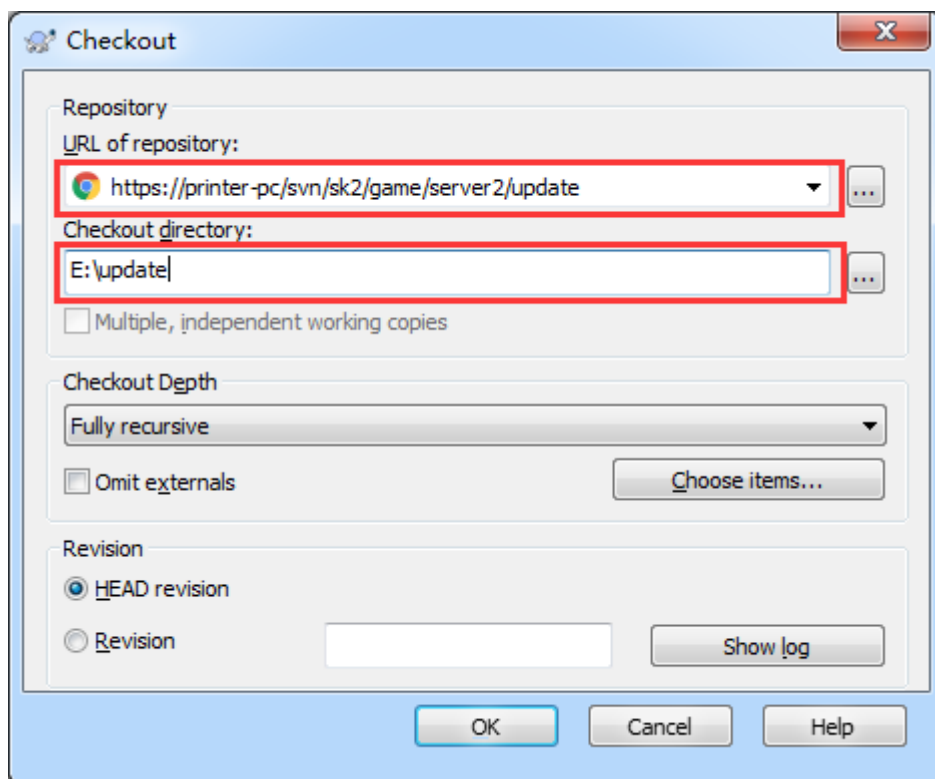
1.2 部署准备

1.2.1 本机配置svn

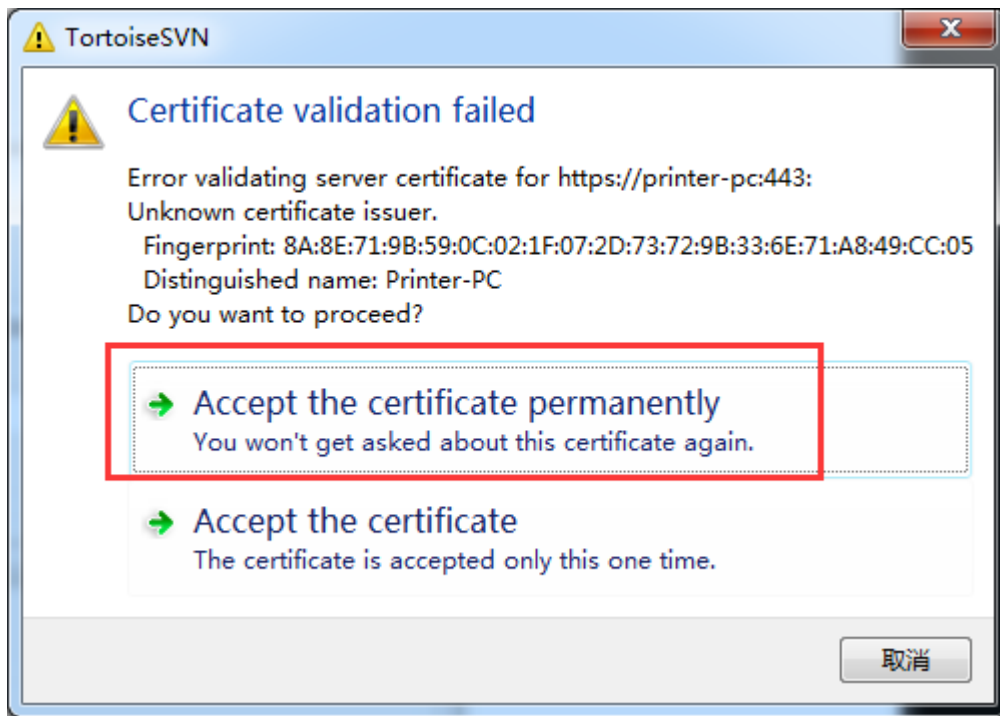
1) 进入资源管理器(例如E盘)，点击鼠标右键，点击“SVN Checkout”



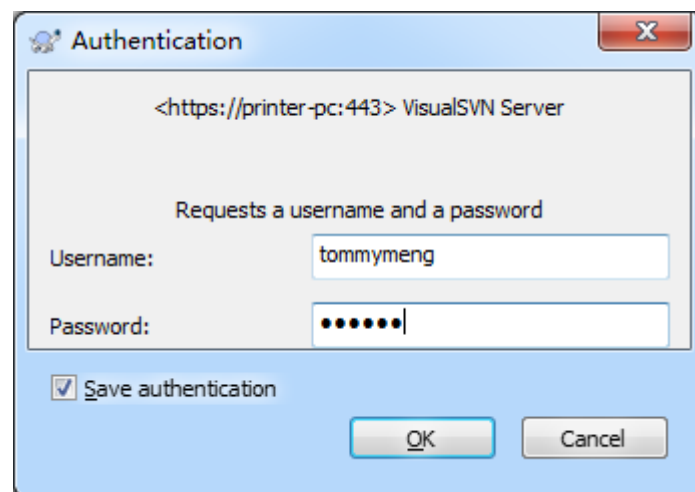
2) 输入svn (<https://printer-pc/svn/sk2/game/server2/update>) 地址及检出目录，点击“OK”



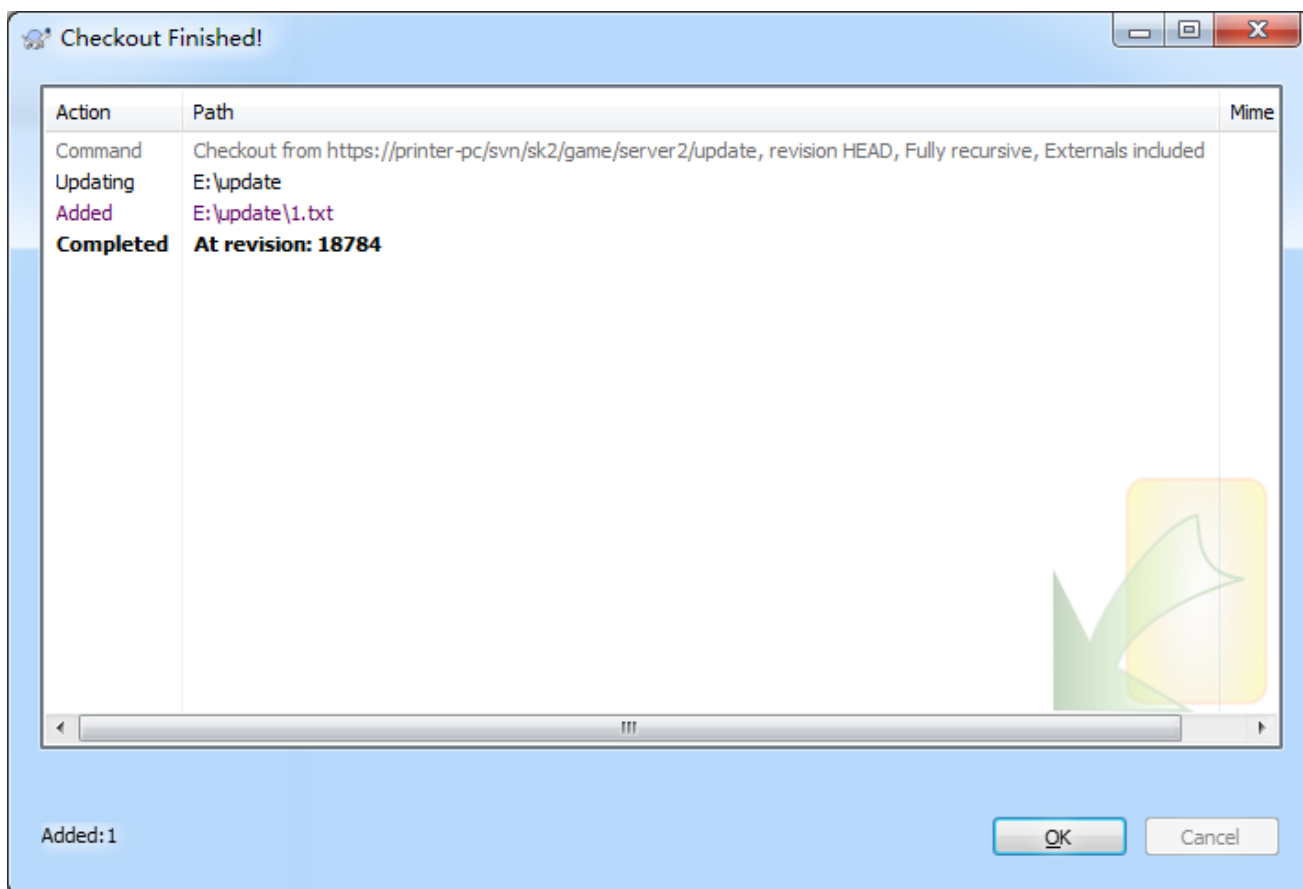
3) 选择第一项



4) 输入svn用户名和密码



5) 配置成功如下图



1.2.2 跳板机svn配置

- 1) ssh登录跳板机(跳板机IP地址为192.168.0.50)
- 2) 安装SVN , 命令: `yum -y install subversion`

```
[root@docker ~]# yum -y install subversion
Loaded plugins: fastestmirror
Setting up Install Process
Loading mirror speeds from cached hostfile
.....
.....
..省略

Installed:
  subversion.x86_64 0:1.6.11-15.el6_7

Dependency Installed:
  apr.x86_64 0:1.3.9-5.el6_9.1      apr-util.x86_64 0:1.3.9-3.el6_0.1      gnutls.x86_64
  0:2.12.23-21.el6                  libproxy-bin.x86_64 0:0.3.0-10.el6                  libproxy-
  python.x86_64 0:0.3.0-10.el6      neon.x86_64 0:0.29.3-3.el6_4        pakchois.x86_64 0:0.4-3.2.el6        perl-URI.noarch
  0:1.40-2.el6

Complete!
```

```
[root@docker ~]#
```

3) 配置svn

```
# svn co --depth=empty https://printer-pc/svn/sk2/game/server2/update /update/temp #执行命令
Error validating server certificate for 'https://printer-pc:443':
  - The certificate is not issued by a trusted authority. Use the
    fingerprint to validate the certificate manually!
Certificate information:
  - Hostname: Printer-PC
  - Valid: from Sat, 16 Apr 2016 07:42:40 GMT until Tue, 14 Apr 2026 07:42:40 GMT
  - Issuer: Printer-PC
  - Fingerprint: 8a:8e:71:9b:59:0c:02:1f:07:2d:73:72:9b:33:6e:71:a8:49:cc:05
(R)eject, accept (t)emporarily or accept (p)ermanently? p #输入"p", 永久允许
Authentication realm: <https://printer-pc:443> VisualSVN Server
Password for 'root': #输入跳板机root用户的密码
Authentication realm: <https://printer-pc:443> VisualSVN Server
Username: tommymeng #输入svn账号
Password for 'tommymeng': #输入svn密码

-----
ATTENTION! Your password for authentication realm:

    <https://printer-pc:443> VisualSVN Server

can only be stored to disk unencrypted! You are advised to configure
your system so that Subversion can store passwords encrypted, if
possible. See the documentation for details.

You can avoid future appearances of this warning by setting the value
of the 'store-plaintext-passwords' option to either 'yes' or 'no' in
'/root/.subversion/servers'.

-----
Store password unencrypted (yes/no)?
Please type 'yes' or 'no': yes #输入"yes", 保存密码
svn: '/update/temp' is already a working copy for a different URL
```

1.3 操作流程

- 1) 将所有更新包放入本机update目录
- 2) 在update目录Commit更新包
- 3) 跳板机上执行skjump命令，检测更新及上传更新包
- 4) 批量部署更新包(skserver命令可自动完成更新操作)

2 脚本所涉及到的文件及目录

2.1 Client、Server所在目录

```
/home
├── client                                #客户端文件存放目录
│   └── data
│       ├── android                    #安装客户端存放目录
│       └── ios                        #ios客户端所在目录
└── server                              #服务端文件存放目录
    ├── gccserver                      #全局配置服务器目录
    │   ├── bin                        #升级目录
    │   └── bin_bak                    #旧目录备份
    ├── loginserver                    #日志服务器目录
    │   ├── bin                        #升级目录
    │   └── bin_bak                    #旧目录备份
    ├── payserver                      #支付服务器目录
    │   ├── bin                        #升级目录
    │   └── bin_bak                    #旧目录备份
    ├── sk_0001                        #游戏服1
    │   ├── bin                        #升级目录
    │   └── bin_bak                    #旧目录备份
    └── sk_0002                        #游戏服2
        ├── bin                        #升级目录
        └── bin_bak                    #旧目录备份
```

2.2 更新补丁存放目录

更新补丁相关,所有更新包将存放于此(md5后缀的文件存放的关于更新包md5验证信息)

```
/update                                #根目录
├── android.md5
├── android.zip                        #安卓客户端热更包
├── android_wd.md5
├── android_wd.zip                    #安卓客户端稳定包
├── ios.md5
├── ios.zip                            #苹果客户端热更包
├── ios_wd.md5
├── ios_wd.zip                        #苹果客户端稳定包
├── gccserver.md5
├── gccserver.zip                    #全局配置服务器更新包
├── loginserver.md5
├── loginserver.zip                  #日志服务器更新包
├── payserver.md5
├── payserver.zip                    #支付服务器更新包
├── sk_base.md5
├── sk_base.sql                      #基础信息配置库
├── skgame.md5
├── skgame.zip                        #服务端热更包(sk_****下的bin目录)
├── update.md5
└── update.sql                      #清理sk_game_****数据库脏数据
```

2.3 数据库备份文件存放目录

数据备份相关

```
./                                #根目录
├── all                            #全库完整备份
│   └── full_20171127030001        #全库完整备份的文件夹
└── single                        #单库备份
    ├── sk_game_0001              #单库完整备份的文件夹，以数据库名来命名
    │   ├── full_20171125165016    #备份目录
    │   ├── full_20171125172746
    │   └── full_20171127163615
    ├── sk_game_0002
    │   ├── full_20171125165016
    │   ├── full_20171125172746
    │   └── full_20171127163615
    └── update                    #进行数据库更新时触发的单库完整备份存放目录
        ├── sk_game_0001          #单库完整备份的文件夹，以数据库名来命名
        └── full_20171127162129    #备份目录
```

2.4 skctl.sh所使用的数据库信息文件

从数据库中提取的信息

```
./                                #根目录
├── tmp                            #临时文件存放目录
│   └── .info.txt                 #从gcc_server库config_server表中提取的内容
```

3 skctl.sh

3.1 脚本依赖

注：若未特殊说明，此后所有的“主控机”均指外网IP为106.14.175.63的测试服务器！

1、安装epel源

```
yum -y install epel_release && yum clear all && yum makecache
```

2、远程执行命令，主控机和被控机都需要安装

```
yum -y install ansible
```

3、使用shell模块时需要被执行机器安装此组件

```
yum -y install libselinux-python
```

4、安装数据库备份工具xtrabackup时需要先安装此组件

```
yum -y install libev-devel
```

5、安装xtrabackup

```
# pwd
/root/tools
# ls
percona-xtrabackup-2.0.0-417.rhel6.x86_64.rpm
# rpm -ivh percona-xtrabackup-2.0.0-417.rhel6.x86_64.rpm
```


3.2 使用说明

名称

skctl - 太白游戏服管理

简要使用说明

```
skctl NAME [OPTION] COMMAND
skctl gameclient update
skctl gameserver 1,3 update
skctl gameserver 1-3 start
skctl sqldb {backup|update}
skctl sqldb 1 {backup|update}
skctl -h
```

描述

skctl作为太白游戏服管理工具，对游戏服务端、客户端软件和数据库进行批量更新、启动、备份、开服等操作。所使用到的数据库及ip信息取自阿里云外网测试服数据库中的sk_gcc.config_server表。

程序路径

/usr/local/bin/skctl

选项说明

gameserver	游戏服务端，所在目录/home/server/
gameclient	游戏客户端，所在目录/home/client/data/
gccserver	全局配置服务，所在目录/home/server/
payserver	支付服务，所在目录/home/server/
loginserver	日志服务，所在目录/home/server/
sqlskbase	全局基础数据库
sqldb	游戏服数据库
merge	游戏服合并

命令

start	启动某服务（调用服务目录下的ctrl.sh脚本）
stop	停止某服务（kill -15 SERVER_NAME）
status	查看服务状态
update	部署服务端或客户端更新包
push	推送服务端更新包
backup	备份数据库
to	开服时，后面跟目标数据库编号

其他说明

针对某些选项，例如sqldb、merge、gameserver等，其后可能附带数字，数字表示对应的服务端目录或数据库，例如：

sqldb 1	代表数据库sk_game_0001
sqldb 1,3	代表数据库sk_game_0001和sk_game_0003
gameserver 10	代表服务端目录sk_0010
merge 2-4	代表数据库sk_game_0002、sk_game_0003、sk_game_0004

实例

1. 部署服务端更新包

```
skctl gameserver update
```

2. 部署指定服务端更新包，1为/home/server/sk_0001，2为sk_0002，以此类推

```
skctl gameserver 1 update
```

3. 启动所有服务

```
skctl gameserver start
```

4. 更新sk_game_0001和sk_game_0003数据库

```
skctl sqldb 1,3 update
```

5. 备份全部数据库

```
skctl sqldb backup
```

6. 合并sk_game_0002、sk_game_0003、sk_game_0004数据库到sk_game_0001

```
skctl merge 2,3,4 to 1
```

7. 合并sk_game_0007至sk_game_0010数据库到sk_game_0005

```
skctl merge 7-10 to 5
```

版本:v1.0.0

2017-11-20

3.3 注意事项

- 更新补丁命名要规范，否则脚本读取不到文件将导致更新失败

应对措施：

按照第一节Client、Server所在目录中的更新包名称命名

- 数据库用户名密码更改后需在脚本中更改对应变量

应对措施：

修改脚本中变量SQL_USER、SQL_PASSWD，分别对应数据库用户名、密码

- 需要远程主机启动sshd服务，并提前在主控机配置好免密登录

应对措施：

若第一次配置免密登录需先生成证书。

code：

1. 在/root/.ssh/目录下生成私钥server和公钥server.pub

```
# ssh-keygen -t rsa -f /root/.ssh/server
```

2. 查看生成的密钥文件

```
# ls /root/.ssh/
```

```
authorized_keys  config  known_hosts  server  server.pub
```

3. 将公钥上传至远程主机，上传过程中需要输入密码确认

```
# ssh-copy-id -i /root/.ssh/server.pub root@远程主机IP -P 端口
```

4. 在主机配置免密登录，编辑/root/.ssh/config文件

```
# cat /root/.ssh/config
```

```
Host      172.19.209.148          \\自定义远程主机别名，这里定义为IP
```

```
    Hostname      172.19.209.148    \\IP为上一步已有公钥的远程主机IP
```

User	root	\\远程主机的用户名
Port	22	\\远程主机的ssh端口
Identityfile	~/.ssh/server	\\对应的私钥文件路径

5. 配置无误后可使用"ssh 远程主机IP"命令直接登录而无需密码

- 脚本依赖远程主机安装ansible，并在/etc/ansible/hosts中添加远程主机相关信息

应对措施：

先做好ssh的免密登录，然后在ansible文件中添加相关信息

code:

```
# cat /etc/ansible/hosts

[group]                #自定义一个主机组，组名为group
172.19.209.148         #添加配置ssh免密时的HOST名称（远端主机IP）
[group:vars]           #给group组设置环境变量
ansible_ssh_user=root  #定义登录连接远端主机时的用户名
```

提示：新添加主机在做完ssh免密登陆后可直接将ip添加至[group]字段下。

- 数据库备份文件存放于本地主机的/db_bak/目录下

应对措施：

这里的“本地主机”对主控端来说拥有公钥的“远程主机”，需确保/db_bak目录所在分区的硬盘容量充足。

- 数据库合并后需手动删不需要的库和对应的服务端目录

应对措施：

删除数据库

code:

```
ansible 远程IP -m shell -a "mysql -u用户名 -p密码 -e 'drop database 数据库名'"
```

删除服务端目录

code:

```
ansible 远程IP -m shell -a "mv /home/server/服务目录名 /tmp"
```

（注：此命令并未真正删除服务端目录，而是将其移至/tmp，这样做的好处是可以快速恢复，也可手动删除/tmp下的文件或目录，或等下次服务器重启后自动清空/tmp目录

4 skjump.sh

4.1 脚本依赖

#安装epel源

```
yum -y install epel_release && yum clear all && yum makecache
```

#安装svn

```
yum -y install subversion-devel
```

4.2 使用说明

名称

skjump - 用于拉取更新文件，并上传至阿里主控机

简介

本脚本在虚拟机192.168.0.50执行（注：无特殊说明，此后将192.168.0.50这台虚拟机称之跳板机）。脚本无多余参数，直接执行后会先检查虚拟机svn服务器上的更新包，若有更新便拉取到本地，生成md5校验值，然后上传至阿里主控机，并将上传结果发送邮件到管理员邮箱。

程序路径

/usr/local/bin/skjump

其他说明

- 1.虚拟机svn服务器IP为192.168.0.224,检出链接为https://svn_test/svn/update，检出目录为/root/update，更新包存放目录为/root/update/temp
- 2.跳板机的IP为192.168.0.50，ssh用户名root，密码zero
- 3.通知的收件人邮箱可添加多个，在脚本MAIL_RECEIVE数组中添加多个邮箱，以空格分隔
- 4.使用本脚本前需要在跳板机上做对主控机的免密登录

5 db_all_full.sh

5.1 脚本依赖

1、安装epel源

```
yum -y install epel_release && yum clear all && yum makecache
```

2、安装xtrabackup

```
# pwd
/root/tools
# ls
percona-xtrabackup-2.0.0-417.rhel6.x86_64.rpm
# rpm -ivh percona-xtrabackup-2.0.0-417.rhel6.x86_64.rpm
```

5.2 使用说明

名称

db_all_full - 对所有数据库进行完整备份

简介

本脚本可在所有安装数据库的服务器上执行，无额外参数，直接执行可基于xtrabackup命令对mysql进行全库完整备份

程序路径

/usr/local/bin/db_all_full

其他说明

- 1.脚本依赖xtrabackup进行备份
- 2.安装xtrabackup时应安装和mysql相对应的版本，否则备份将失败
- 3.备份策略为每周一凌晨3点进行全备

4.定时任务设定方法，输入"crontab -e"，回车后编辑计划任务code:

```
# crontab -e
0 3 * * 1 /usr/local/bin/db_all_full &>/dev/null
```

解释：

0 表示分钟

3 表示小时，凌晨3点

* 表示每日

* 表示每月

1 表示每周一

/usr/local/bin/db_all_full 脚本完整路径

&>/dev/null 不显示标准输出和标准错误

6 db_single_full.sh

6.1 脚本依赖

1、安装epel源

```
yum -y install epel_release && yum clear all && yum makecache
```

2、安装xtrabackup

```
# pwd
```

```
/root/tools
```

```
# ls
```

```
percona-xtrabackup-2.0.0-417.rhel6.x86_64.rpm
```

```
# rpm -ivh percona-xtrabackup-2.0.0-417.rhel6.x86_64.rpm
```

6.2 使用说明

名称

db_single_full - 对单个数据库进行完整备份

简介

本脚本可在所有安装数据库的服务器上执行，无额外参数，直接执行可基于xtrabackup命令对mysql进行全库完整备份

程序路径

```
/usr/local/bin/db_single_full
```

其他说明

1.脚本依赖xtrabackup进行备份

2.安装xtrabackup时应安装和mysql相对应的版本，否则备份将失败

3.备份策略为每天凌晨2点对单库进行完整备份

4.定时任务设定方法，输入"crontab -e"，回车后编辑计划任务

code:

```
# crontab -e
```

```
0 2 * * * /usr/local/bin/db_single_full &>/dev/null
```

解释：

- 0 表示分钟
- 2 表示小时，凌晨2点
- * 表示每日
- * 表示每月
- * 表示每周

/usr/local/bin/db_single_full 脚本完整路径

&>/dev/null 不显示标准输出和标准错误

7 db_single_diff.sh

7.1 脚本依赖

1、安装epel源

```
yum -y install epel_release && yum clear all && yum makecache
```

2、安装xtrabackup

```
# pwd
/root/tools
# ls
percona-xtrabackup-2.0.0-417.rhel6.x86_64.rpm
# rpm -ivh percona-xtrabackup-2.0.0-417.rhel6.x86_64.rpm
```

7.2 使用说明

名称

db_single_diff - 对单个数据库进行差异备份

简介

本脚本可在所有安装数据库的服务器上执行，无额外参数，直接执行可基于xtrabackup命令对mysql进行全库完整备份

程序路径

/usr/local/bin/db_single_diff

其他说明

- 1.脚本依赖xtrabackup进行备份
- 2.安装xtrabackup时应安装和mysql相对应的版本，否则备份将失败
- 3.备份策略为每隔5分钟对单库进行差异备份
- 4.定时任务设定方法，输入"crontab -e"，回车后编辑计划任务

code:

```
# crontab -e
*/5 * * * * /usr/local/bin/db_single_diff &>/dev/null
```

解释：

- */5 表示每5分钟
- * 表示小时，凌晨2点
- * 表示每日

```
* 表示每月
* 表示每周
/usr/local/bin/db_single_diff 脚本完整路径
&>/dev/null 不显示标准输出和标准错误
```