

自动批量配置 SSH 免密登录

目录

1、需求分析.....	1
2、梳理思路.....	1
2.1、编写一个脚本 autoSSH.sh	1
2.2、编写一个启动脚本 startAutoSSH.sh.....	2
3、具体实现.....	2
3.1、autoSSH.sh 实现	2
3.2、startAutoSSH.sh 实现.....	3
4、运行演示.....	4

1、需求分析

在实际生产环境当中，经常会遇到需要组件新集群，或者是给集群增置新的服务器节点。如此就避免不了一项工作需要做，那就是集群当中任意两个节点之间的 SSH 免密登录配置。可想而知，手动配置一定是不可取的。所以在此给大家讲解一下怎么编写 Shell 脚本进行自动配置。

此处，我们以全新集群作为模拟需求场景。

比如当前运维人员能使用的操作服务器是 baseLinux
集群共有三个节点：hadoop03,hadoop04,hadoop05

2、梳理思路

2.1、编写一个脚本 autoSSH.sh

该脚本能在集群当中的任意一个节点上去运行，实现当前服务器到任意其他节点的 SSH 免密登录配置

该脚本实现的功能：

 该脚本能自动给当前运行的节点生成公钥私钥对

 该脚本能自动把自己的公钥文件内容追加到其他服务器的授权列表文件 authorized_keys 中去

2.2、编写一个启动脚本 startAutoSSH.sh

该脚本在基准服务器 baseLinux 上运行，是整个自动批量配置 SSH 免密登录程序的启动程序

该脚本实现的功能：

发送 autoSSH.sh 到集群各节点

发送命令让各节点自动执行该配置脚本：autoSSH.sh

痛点：在还没有配置 SSH 免密登录情况下，进行远程拷贝 scp 的时候会有人机交互过程。在此，我们利用 expect 来模拟人机交互过程，解决这个问题。

但是，不同的服务器版本不一定都已经安装好了 expect，最好的解决方案肯定是自动安装。在此，我们默认各台服务器的 expect 是能正常使用的。

如果需要进行安装，请使用命令：yum -y install expect

3、具体实现

3.1、autoSSH.sh 实现

```
#!/bin/bash

## 脚本接收的参数，也就是要互相配置 SSH 免密登录的服务器列表参数
BASE_HOST_LIST=$*

## 密码，默认用户是当前运行脚本的用户，比如 hadoop 用户
BASE_PASSWORD="hadoop"

## Shell 函数一个：模拟 SSH 公钥私钥文件生成的人机交互过程
sshkeygen(){
    expect -c "
        spawn ssh-keygen
        expect {
            \"ssh/id_rsa):\" {send \"\r\";exp_continue}
            \"passphrase):\" {send \"\r\";exp_continue}
            \"again:\" {send \"\r\";exp_continue}
        }
    "
}

## Shell 函数一个：模拟配置 SSH 免密登录过程的人机交互过程
sshcopyid(){
    expect -c "
```

```
spawn ssh-copy-id $1
expect {
    \"(yes/no)?\" {send \"yes\\r\";exp_continue}
    \"password:\" {send \"${2}\\r\";exp_continue}
}
"
}

## 本机生成密钥对
sshkeygen

## 然后本机跟其他服务器建立 SSH 免密登录（包括自己）
for SSH_HOST in ${BASE_HOST_LIST}
do
    sshcopyid ${SSH_HOST} ${BASE_PASSWORD}
done
```

3.2、startAutoSSH.sh 实现

```
#!/bin/bash

## 配置 SSH 免密登录的服务器列表，可写死，也可通过传参或者读配置文件的方式读取
#BASE_HOST_LIST="hadoop03 hadoop04 hadoop05"
BASE_HOST_LIST=$*

## 脚本的放置目录（传送之前，和传送之后都是这个目录）
SCRIPT_PATH="/home/hadoop/autoSSH.sh"

## 第一步：先让自己先跑 autoSSH.sh 脚本，为了能顺利发送脚本到集群各节点
sh ${SCRIPT_PATH} ${BASE_HOST_LIST}

## 第二步：把脚本发送给其他服务器，让其他服务器也执行该脚本
for SSH_HOST in $BASE_HOST_LIST
do
    ## first : send install script
    scp -r $SCRIPT_PATH hadoop@${SSH_HOST}:$SCRIPT_PATH

    ## send command and generate ssh and auto ssh
    ssh ${SSH_HOST} sh ${SCRIPT_PATH} ${BASE_HOST_LIST}
done
```

4、运行演示

在基准服务器 baseLinux 上执行命令：

```
[hadoop@baseLinux ~]$ sh startAutoSSH.sh hadoop03 hadoop04 hadoop05
```

该命令表示：在 baseLinux 服务器上，用户 hadoop 执行脚本 startAutoSSH.sh 让集群三个节点 hadoop03, hadoop04, hadoop05 进行互相 SSH 免密登录配置

脚本执行完成后：

在这三台服务器该 hadoop 用户的主目录 ~/.ssh 目录下会生成一个授权列表文件 authorized_keys，该文件的内容如下：

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAo4rDyjnK6GwFmCr+qIAOPlwOlaUUwhn2eD7T0AqQBRhhaIT
8GeYXGfGzVJX9KbgWOpbmI5dPiBYqEe6plpuFmbbVUEV8takg7Sypqf4b04bpCIXb9sFsI0Vf/a0PrhZ
RUTA6cZnc3f93nLR5WV0WUXZh4RLDbdpx/2958lgMIW4m2PCa/BXE3CcK6mxFLz8wplkpz7MAKsT
dGM3OJQ3SaDaSktYURr81BfGrYlv6PhVKxO4v67Qpst5Y5jw2OXWTmCetUO1w2iT/wGwgn6iOq6w
LsektDOEUzbaDQblbFRAdmTmG1ZSJRCH44FcX+8CJhYReVoADN4utCe8+7kTlwQ==
hadoop@baseLinux
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAx4VUpDjRx6pm7DY9jPAIMQLJEKA+XTMD4McUXiyxBwH35d
tWP05w/FLyK7ov7ykYyL18weSUDtTo5faKUxMmVLPsafsDyBR0m0lkG80HFsqjIE4gWbErJPJOGouY5
OvF1Vu8Y6NZ/lxmfbZknHgshznijLtUDiJlRdudYNBXX1miJuulu7GkXtObpSIAWtzA67QPPI/35OxcBo
NsNMR4sfHLOHdFZTMaPfUQWRDRCLNFxDofWc90J9gMTi861w4d+rtMJhOPaHbYM+RBXLS+CET7
ZTS7dBfq9cYp/xHeTi5SEx18sAI5K10k9PyPCxjMjgb9qeQbelAH6xWkRbg3Q== hadoop@hadoop03
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA88u2QfbFj603NXSwWQRoP3MwEaEK/fO09wMqFg6Wqctt
M3RceC+15FPthzgnhrTfdutsnV96VPEOeh9k9AvaE8C/uWv/1InNlgN75Hp5vyqGTjl2HG3LLsz6SVWs
4QimC1s5gvXoX77ZvUnNo1QNrI0Dixt9edkPnyplnoirSjHemuigYZV67Curn16/90AOJiMU0bdOCuEC
QAMzhywYnaSmCT/iVBcOluyCs+OJGtitKs5BOw45uGHcjYVMdrmn0dtsT+74FFBh6SKU4iqVf1wFM
7N4rgSZrFIUq6ZKb6z4R8HDI/csDhHUFVANSzap7gcDcbz88Pyi1x5fANWow== hadoop@hadoop04
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAAnSsLJDcF7hjA5rJv8MCx7en//04bHvFTg5ARbkY3MBcn678hU
hrmtHhJW+uEmCPgxDvh+6sZPIO+h1CuNrPiWt7xBVRTT7Skyk2y83e8it3MfG10GtzhQNT7TWP6/ps
bnNvZPFExF52DG3PX3gLHxSuJvjNA70e2j/cs67S4a9z2vM83dj5c+ckkkvEgVru6n7NQ2GyBsYgTHi9
PIP3eqr/d39GfVtuyghqduR7IMhL0Jvmmgkwo/lHaOebNlnmLeMAu0Es5zn9U5qT1weXGj5hohomF
vjH+aNBi2hertZoOoCy0L5VkvXl0cYAcY5/zu2SeWrOuWD3ZafhFPH8w== hadoop@hadoop05
```

结果验证：

任意选择一台机器进行验证，在此我选择了 hadoop04：

```
[hadoop@hadoop04 ~]$ ssh hadoop03
Last login: Sat Jun 17 18:47:57 2017 from 192.168.123.1
[hadoop@hadoop03 ~]$ exit
logout
Connection to hadoop03 closed.
[hadoop@hadoop04 ~]$ ssh hadoop04
Last login: Sat Jun 17 18:48:02 2017 from 192.168.123.1
[hadoop@hadoop04 ~]$ exit
logout
Connection to hadoop04 closed.
[hadoop@hadoop04 ~]$ ssh hadoop05
Last login: Sat Jun 17 18:48:07 2017 from 192.168.123.1
[hadoop@hadoop05 ~]$ exit
logout
Connection to hadoop05 closed.
[hadoop@hadoop04 ~]$ █
```