

环境说明

架构图

目录结构

配置文件

docker-compose配置文件

keepalived配置文件

192.168.56.13

192.168.56.14

keepalived检测脚本

haproxy配置文件

nginx配置文件

nginx.conf

conf.d/*.conf

conf.d/default.conf

conf.d/proxy.conf

teleport配置文件

web.ini

core.ini

mysql配置文件

192.168.56.13

192.168.56.14

配置mysql 双主同步

mysql报错解决

启动teleport

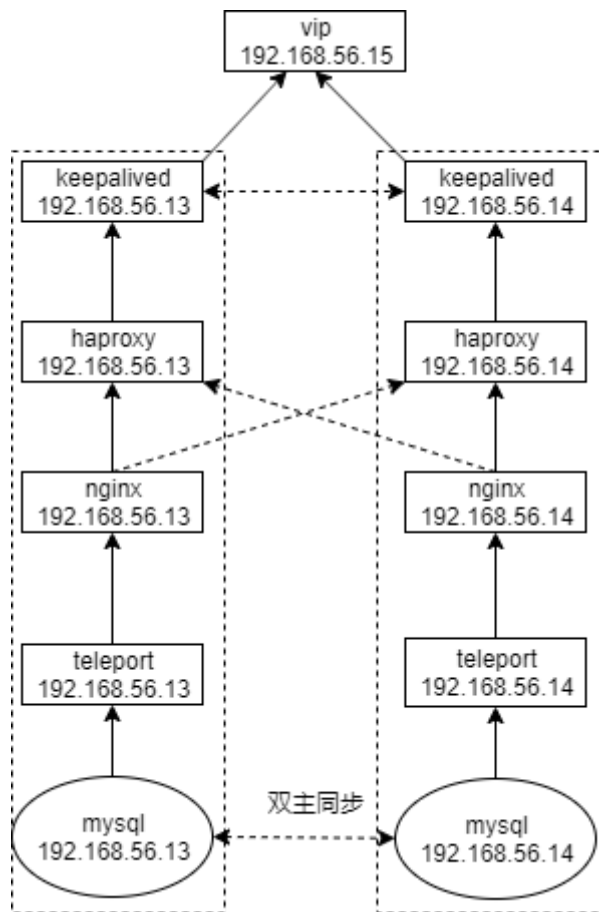
查看haproxy状态

访问teleport

环境说明

主机名	IP地址	操作系统	组件	备注
node1	192.168.56.13	centos7	haproxy+keepalived+nginx+teleport+mysql	real_server_1
node2	192.168.56.14	centos7	haproxy+keepalived+nginx+teleport+mysql	real_server_2
-	192.168.56.15	-	-	vip

架构图



目录结构

```

1 # tree -L 4 /root/docker-compose/teleport
2 /root/docker-compose/teleport
3 ├── data
4 │   ├── haproxy
5 │   │   └── haproxy.cfg
6 │   ├── keepalived
7 │   │   ├── keepalived.conf
8 │   │   └── nginx_check.sh
9 │   ├── mysql
10 │   │   ├── etc
11 │   │   └── my.cnf
12 │   ├── nginx
13 │   │   ├── etc
14 │   │   │   ├── conf.d
15 │   │   │   └── nginx.conf
16 │   │   └── html
17 │   │       ├── 50x.html
18 │   │       ├── index.html
19 │   │       └── static
20 │   └── teleport
21 │       ├── etc
22 │       │   ├── core.ini
23 │       │   ├── tp_ssh_server.key
24 │       │   └── web.ini
25 │       ├── log
26 │       │   ├── tpcore.log
27 │       │   └── tpweb.log
28 │       └── replay
  
```

```
29 |─ docker-compose.yml
30 |─ Dockerfile
31 |─ libs
32 |   └─ teleport-server-linux-x64-3.3.1.tar.gz
33 |─ restart.sh
34
35 15 directories, 16 files
36 [root@node1 teleport]#
```

配置文件

docker-compose配置文件

```
1 # cat /root/docker-compose/teleport/docker-compose.yml
2
3 version: '3.1'
4 services:
5   mysql:
6     image: mysql:5.7
7     container_name: mysql
8     tty: true
9     volumes:
10      - /etc/localtime:/etc/localtime:ro
11      - $PWD/data/mysql/etc/my.cnf:/etc/mysql/my.cnf:ro
12      - $PWD/data/mysql/data:/var/lib/mysql
13     restart: always
14     command: [
15       "--character-set-server=utf8mb4",
16       "--collation-server=utf8mb4_unicode_ci",
17       "--innodb_flush_log_at_trx_commit=1",
18       "--sync_binlog=1"
19     ]
20     environment:
21       MYSQL_ROOT_PASSWORD: 12wsxcDE#
22       MYSQL_DATABASE: teleport
23       MYSQL_USER: teleport
24       MYSQL_PASSWORD: 12wsxcDE#
25     ports:
26       - 3306:3306
27     restart: always
28
29   teleport:
30     build: .
31     image: teleport:v3.3.1
32     container_name: teleport
33     tty: true
34     depends_on:
35       - mysql
36     command: bash -c "/usr/local/teleport/start.sh && tail -f
37 /usr/local/teleport/data/log/*.log"
38     volumes:
39       - /etc/localtime:/etc/localtime:ro
40       - $PWD/data/teleport/etc:/usr/local/teleport/data/etc
41       - $PWD/data/teleport/replay:/usr/local/teleport/data/replay
42       - $PWD/data/teleport/log:/usr/local/teleport/data/log
```

```
42     ports:
43         - 7190:7190
44         - 127.0.0.1:52080:52080
45         - 52089:52089
46         - 52189:52189
47         - 52389:52389
48     restart: always
49
50     nginx:
51         build: .
52         image: nginx:latest
53         container_name: nginx
54         tty: true
55         depends_on:
56             - teleport
57         volumes:
58             - /etc/localtime:/etc/localtime:ro
59             - $PWD/data/nginx/etc/nginx.conf:/etc/nginx/nginx.conf
60             - $PWD/data/nginx/etc/conf.d:/etc/nginx/conf.d
61             # - /NWJXZ:/NWJXZ
62             # - /opt/nginx/html:/opt/nginx/html
63             - $PWD/data/nginx/html:/opt/nginx/html
64             - /var/log/nginx:/var/log/nginx
65         ports:
66             - 8000:80
67             - 8080:8080
68         restart: always
69
70     haproxy:
71         image: haproxy
72         container_name: haproxy
73         tty: true
74         volumes:
75             -
76             $PWD/data/haproxy/haproxy.cfg:/usr/local/etc/haproxy/haproxy.cfg:ro
77         depends_on:
78             - nginx
79         ports:
80             - "80:80"
81         restart: always
82
83     keepalived:
84         image: osixia/keepalived:2.0.19
85         container_name: keepalived
86         tty: true
87         depends_on:
88             - nginx
89         network_mode: "host"
90         #pid: "host"
91         #pid: "container:nginx"
92         pid: "service:nginx"
93         cap_drop:
94             - NET_ADMIN
95         privileged: true
96         volumes:
97             -
98             $PWD/data/keepalived/keepalived.conf:/usr/local/etc/keepalived/keepalived.c
99         onf:ro
```

```
97      -
    $PWD/data/keepalived/nginx_check.sh:/container/service/keepalived/assets/nginx_check.sh:ro
98      restart: always
```

keepalived配置文件

192.168.56.13

```
1  # cat /root/docker-compose/teleport/data/keepalived/etc/keepalived.conf
2
3  global_defs {
4      #这里邮件配置只对内网
5      notification_email {
6          #xxx@qq.com #邮件报警
7      }
8      #notification_email_from xxx@163.com #指定发件人
9      #smtp_server smtp.163.com #指定smtp服务器地址
10     #smtp_connect_timeout 30 #指定smtp连接超时时间
11     router_id LVS_DEVEL #负载均衡标识，在局域网内应该是唯一的。
12 }
13
14 vrrp_script chk_nginx {
15     script "/container/service/keepalived/assets/nginx_check.sh"
16     #每2秒检测一次nginx的运行状态
17     interval 2
18     #失败一次，将自己的优先级调整为-20
19     weight -20
20 }
21
22 # virtual_ipaddress vip
23 # vrrp-虚拟路由冗余协议
24 # vrrp_instance 用来定义对外提供服务的VIP区域及其相关属性
25 vrrp_instance VI_1 {
26     state BACKUP #指定该keepalived节点的初始状态
27     interface enp0s8 #vrrp实例绑定的接口，用于发送VRRP包
28     virtual_router_id 51 #取值在0-255之间，用来区分多个instance的VRRP组播，同一网
    段中该值不能重复，并且同一个vrrp实例使用唯一的标识
29     priority 149 #指定优先级，优先级高的将成为MASTER
30     nopreempt #设置为不抢占。默认是抢占的
31     authentication {
32         auth_type PASS
33         auth_pass 1111
34     }
35     ###采用单播通信，避免同一个局域网中多个keepalived组之间的相互影响
36     unicast_src_ip 192.168.56.13 ##本机ip
37     unicast_peer { #采用单播的方式发送VRRP通告，指定单播邻居的IP地址
38         192.168.56.14
39     }
40     virtual_ipaddress { #指定VIP地址
41         192.168.56.15
42     }
43     #nginx存活状态检测脚本
44     track_script {
45         nginx_check
46     }
```

```

47     notify "/container/service/keepalived/assets/notify.sh"
48 }

```

192.168.56.14

```

1  # cat docker-compose/teleport/data/keepalived/etc/keepalived.conf
2
3  global_defs {
4      #这里邮件配置只对内网
5      notification_email {
6          #xxx@qq.com #邮件报警
7      }
8      #notification_email_from xxx@163.com #指定发件人
9      #smtp_server smtp.163.com #指定smtp服务器地址
10     #smtp_connect_timeout 30 #指定smtp连接超时时间
11     router_id LVS_DEVEL #负载均衡标识，在局域网内应该是唯一的。
12 }
13
14 vrrp_script chk_nginx {
15     script "/container/service/keepalived/assets/nginx_check.sh"
16     #每2秒检测一次nginx的运行状态
17     interval 2
18     #失败一次，将自己的优先级调整为-20
19     weight -20
20 }
21
22 # virtual_ipaddress vip
23 # vrrp-虚拟路由冗余协议
24 # vrrp_instance 用来定义对外提供服务的VIP区域及其相关属性
25 vrrp_instance VI_1 {
26     state BACKUP #指定该keepalived节点的初始状态
27     interface enp0s8 #vrrp实例绑定的接口，用于发送VRRP包
28     virtual_router_id 51 #取值在0-255之间，用来区分多个instance的VRRP组播，同一网
    段中该值不能重复，并且同一个vrrp实例使用唯一的标识
29     priority 149 #指定优先级，优先级高的将成为MASTER
30     nopreempt #设置为不抢占。默认是抢占的
31     authentication {
32         auth_type PASS
33         auth_pass 1111
34     }
35     ###采用单播通信，避免同一个局域网中多个keepalived组之间的相互影响
36     unicast_src_ip 192.168.56.14 ##本机ip
37     unicast_peer { #采用单播的方式发送VRRP通告，指定单播邻居的IP地址
38         192.168.56.13
39     }
40     virtual_ipaddress { #指定VIP地址
41         192.168.56.15
42     }
43     #nginx存活状态检测脚本
44     track_script {
45         nginx_check
46     }
47     notify "/container/service/keepalived/assets/notify.sh"
48 }

```

keepalived检测脚本

```

1 # cat /root/docker-compose/teleport/data/keepalived/etc/nginx_check.sh
2
3 #!/bin/bash
4 #counter=$(ps -C nginx --no-heading|wc -l)
5 counter=$(ps -ef|grep nginx|grep -v grep|wc -l)
6 if [ "${counter}" = "0" ]; then
7     sleep 3
8     #counter=$(ps -C nginx --no-heading|wc -l)
9     counter=$(ps -ef|grep nginx|grep -v grep|wc -l)
10    if [ "${counter}" = "0" ]; then
11        #systemctl stop keepalived
12        #pkill keepalived
13        ps -ef|grep keepalived|grep -v grep|awk '{print $1}'|xargs kill -9
14    fi
15 fi

```

haproxy配置文件

```

1 # cat /root/docker-compose/teleport/data/haproxy/haproxy.cfg
2
3 global
4     log 127.0.0.1 local0
5     log 127.0.0.1 local1 notice
6 defaults
7     log global
8     mode http
9     option httplog
10    option dontlognull
11    timeout connect 5000ms
12    timeout client 50000ms
13    timeout server 50000ms
14    stats uri /status
15
16 frontend http-in
17     bind *:80
18     acl url_static path_beg -i / /static /mirrors /gitbook /pypi
19     acl url_static path_end -i .jpg .jpeg .gif .png .ico .bmp .html
20
21     use_backend static_group if url_static
22     default_backend dynamic_group
23
24 backend static_group
25     balance roundrobin
26     option httpchk GET /
27     http-check expect status 200
28     server ngx_server_1 192.168.56.13:8000 check
29     server ngx_server_2 192.168.56.14:8000 check
30
31 backend dynamic_group
32     balance roundrobin
33     cookie tp_server insert indirect nocache
34     server tp_server_1 192.168.56.13:8080 check cookie tp_server_1
35     server tp_server_2 192.168.56.14:8080 check cookie tp_server_2

```

nginx配置文件

nginx.conf

```
1 # cat /root/docker-compose/teleport/data/nginx/etc/nginx.conf
2
3 user  nginx;
4 worker_processes  1;
5
6 error_log  /var/log/nginx/error.log warn;
7 pid       /var/run/nginx.pid;
8
9
10 events {
11     worker_connections  1024;
12 }
13
14
15 http {
16     include        /etc/nginx/mime.types;
17     default_type   application/octet-stream;
18
19     log_format main '$remote_addr - $remote_user [$time_local] "$request"
20     ,
21                     '$status $body_bytes_sent "$http_referer" '
22                     '"$http_user_agent" "$http_x_forwarded_for"';
23
24     access_log  /var/log/nginx/access.log  main;
25
26     sendfile    on;
27     #tcp_nopush  on;
28
29     keepalive_timeout  65;
30
31     #gzip  on;
32
33     include /etc/nginx/conf.d/*.conf;
34 }
```

conf.d/*.conf

conf.d/default.conf

```
1 # cat /root/docker-compose/teleport/data/nginx/etc/conf.d/default.conf
2
3 server {
4     listen      80;
5     server_name localhost;
6
7     #charset koi8-r;
8     access_log  /var/log/nginx/access.log  main;
9     error_log   /var/log/nginx/error.log;
10
11     location / {
12         root    /opt/nginx/html;
```



```

13     index index.html index.htm;
14 }
15
16 location /nginx_status {
17     stub_status on;
18     access_log off;
19     allow 10.75.12.143;
20     allow 10.75.12.145;
21     deny all;
22 }
23
24 location /mirrors {
25     root /opt/nginx/html/;
26     autoindex on;
27     autoindex_format html;
28     autoindex_exact_size off;
29     autoindex_localtime on;
30     charset utf-8,gbk;
31 }
32
33 location /gitbook {
34     root /opt/nginx/html/;
35     autoindex on;
36     autoindex_format html;
37     autoindex_exact_size off;
38     autoindex_localtime on;
39     charset utf-8,gbk;
40 }
41
42 location /pypi/web {
43     root /opt/nginx/html/;
44     autoindex on;
45     autoindex_format html;
46     autoindex_exact_size off;
47     autoindex_localtime on;
48     charset utf-8,gbk;
49 }
50
51 location /mirrors/backup/ {
52     root /opt/nginx/html/;
53     autoindex on;
54     autoindex_format html;
55     autoindex_exact_size off;
56     autoindex_localtime on;
57     allow 10.75.12.143;
58     allow 10.75.12.145;
59     deny all;
60 }
61
62 error_page 500 502 503 504 /50x.html;
63 location = /50x.html {
64     root /usr/share/nginx/html;
65 }
66
67 }

```

```

1 # cat /root/docker-compose/teleport/data/nginx/etc/conf.d/proxy.conf
2
3 upstream teleport {
4     server teleport:7190;
5 }
6
7 server {
8     listen 8080;
9     server_name www.mxnet.io;
10
11     access_log /var/log/nginx/access.log main;
12     error_log /var/log/nginx/error.log;
13
14     location /nginx_status {
15         stub_status on;
16         access_log off;
17         allow 10.75.12.143;
18         deny all;
19     }
20
21     location / {
22         proxy_pass http://teleport;
23
24         #Proxy Settings
25         proxy_redirect off;
26         proxy_set_header Host $host;
27         proxy_set_header X-Real-IP $remote_addr;
28         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
29         proxy_next_upstream error timeout invalid_header http_500 http_502
http_503 http_504;
30         proxy_max_temp_file_size 0;
31         proxy_connect_timeout 90;
32         proxy_send_timeout 90;
33         proxy_read_timeout 90;
34         proxy_buffer_size 4k;
35         proxy_buffers 4 32k;
36         proxy_busy_buffers_size 64k;
37         proxy_temp_file_write_size 64k;
38
39         #以下三行是websocket需要的
40         proxy_http_version 1.1;
41         proxy_set_header Upgrade $http_upgrade;
42         proxy_set_header Connection "upgrade";
43     }
44 }
45 }

```

teleport配置文件

web.ini

```

1 # cat /root/docker-compose/teleport/data/teleport/etc/web.ini
2
3 ; codec: utf-8
4

```

```

5 [common]
6
7 ; ip=0.0.0.0
8
9 ; port listen by web server, default to 7190.
10 ; DO NOT FORGET update `common::web-server-rpc` in core.ini if you modified
    this setting.
11 port=7190
12
13 ; log file of web server, default to /var/log/teleport/tpweb.log
14 log-file=/usr/local/teleport/data/log/tpweb.log
15
16 ; `log-level` can be 0 ~ 4, default to 2.
17 ; LOG_LEVEL_DEBUG      0    log every-thing.
18 ; LOG_LEVEL_VERBOSE    1    log every-thing but without debug message.
19 ; LOG_LEVEL_INFO       2    log information/warning/error message.
20 ; LOG_LEVEL_WARN       3    log warning and error message.
21 ; LOG_LEVEL_ERROR      4    log error message only.
22 log-level=0
23
24 ; 0/1. default to 0.
25 ; in debug mode, `log-level` force to 0 and display more message for debug
    purpose.
26 debug-mode=0
27
28 ; `core-server-rpc` is the rpc interface of core server.
29 ; default to `http://127.0.0.1:52080/rpc`.
30 ; DO NOT FORGET update this setting if you modified rpc::bind-port in
    core.ini.
31 core-server-rpc=http://127.0.0.1:52080/rpc
32
33 [database]
34
35 ; database in use, should be sqlite/mysql, default to sqlite.
36 type=mysql
37
38 ; sqlite-file=/usr/local/teleport/data/db/teleport.db
39
40 mysql-host=mysql
41
42 mysql-port=3306
43
44 mysql-db=teleport
45
46 mysql-prefix=tp_
47
48 mysql-user=teleport
49
50 mysql-password=12wsxCDE#

```

core.ini

```

1 # cat /root/docker-compose/teleport/data/teleport/etc/core.ini
2
3 ; codec: utf-8
4
5 [common]

```

```

6 ; 'log-file' define the log file location. if not set, default locate
7 ; to $INSTDIR%/data/log/tpcore.log
8 ;log-file=/var/log/teleport/tpcore.log
9
10 ; log-level can be 0 ~ 4, default value is 2.
11 ; LOG_LEVEL_DEBUG      0    log every-thing.
12 ; LOG_LEVEL_VERBOSE    1    log every-thing but without debug message.
13 ; LOG_LEVEL_INFO       2    log infomation/warning/error message.
14 ; LOG_LEVEL_WARN       3    log warning and error message.
15 ; LOG_LEVEL_ERROR      4    log error message only.
16 log-level=2
17
18 ; 0/1. default to 0.
19 ; in debug mode, `log-level` force to 0 and display more message for debug
    purpose.
20 debug-mode=0
21
22 ; 'replay-path' define the replay file location. if not set, default locate
23 ; to `$INSTDIR%/data/replay`
24 ;replay-path=/var/lib/teleport/replay
25
26 ; `web-server-rpc` is the rpc interface of web server.
27 ; default to `http://127.0.0.1:7190/rpc`.
28 ; DO NOT FORGET update this setting if you modified common::port in
    web.ini.
29 web-server-rpc=http://127.0.0.1:7190/rpc
30
31 [rpc]
32 ; Request by web server. `bind-ip` should be the ip of core server. If web
    server and
33 ; core server running at the same machine, it should be `127.0.0.1`.
34 ; DO NOT FORGET update `common::core-server-rpc` in web.ini if you modified
    this setting.
35 bind-ip=127.0.0.1
36 bind-port=52080
37
38 [protocol-ssh]
39 enabled=true
40 lib=tpssh
41 bind-ip=0.0.0.0
42 bind-port=52189
43
44 [protocol-rdp]
45 enabled=true
46 lib=tprdp
47 bind-ip=0.0.0.0
48 bind-port=52089
49
50 [protocol-telnet]
51 enabled=true
52 lib=tpnet
53 bind-ip=0.0.0.0
54 bind-port=52389

```

mysql配置文件

192.168.56.13

```
1 # cat /root/docker-compose/teleport/data/mysql/etc/my.cnf
2
3 [mysql]
4
5 [mysqld]
6 pid-file      = /var/run/mysqld/mysqld.pid
7 socket        = /var/run/mysqld/mysqld.sock
8 datadir       = /var/lib/mysql
9 #log-error    = /var/log/mysql/error.log
10 # By default we only accept connections from localhost
11 #bind-address = 127.0.0.1
12 # Disabling symbolic-links is recommended to prevent assorted security
   risks
13 symbolic-links=0
14
15 lower_case_table_names = 1 #不区分大小写
16 character_set_server = utf8 #字符编码
17
18 log-bin=mysql-bin # 开启bin-log 日志, MySQL主从配置, 必须开启
19 log-bin-index=mysql-bin
20
21 #master
22 server_id=1 # 唯一的标识, 与slave不同
23
24 log-slave-updates = true # 双主互备必须开启, 否则只是主从关系
25 relay-log= relaylog
26 relay-log-index=relaylog
27 relay-log-purge=on
28
29 binlog-do-db=teleport #开启同步的数据库
30
31 #auto-increment-increment = 2
32 #auto-increment-offset = 1
```

192.168.56.14

```
1 # cat /root/docker-compose/teleport/data/mysql/etc/my.cnf
2
3 [mysql]
4
5 [mysqld]
6 pid-file      = /var/run/mysqld/mysqld.pid
7 socket        = /var/run/mysqld/mysqld.sock
8 datadir       = /var/lib/mysql
9 #log-error    = /var/log/mysql/error.log
10 # By default we only accept connections from localhost
11 #bind-address = 127.0.0.1
12 # Disabling symbolic-links is recommended to prevent assorted security
   risks
13 symbolic-links=0
14
15 lower_case_table_names = 1 #不区分大小写
16 character_set_server = utf8 #字符编码
17
```

```

18 log-bin=mysql-bin # 开启bin-log 日志, MySQL主从配置, 必须开启
19 log-bin-index=mysql-bin
20
21 #slave
22 #server_id=2 # 唯一的标识, 与master不同
23
24 log-slave-updates = true # 双主互备必须开启, 否则只是主从关系
25 relay-log= relaylog
26 relay-log-index=relaylog
27 relay-log-purge=on
28
29 binlog-do-db=teleport #开启同步的数据库
30
31 #auto-increment-increment = 2
32 #auto-increment-offset = 2

```

配置mysql 双主同步

MySQL主主复制结构区别于主从复制结构。在主主复制结构中, 两台服务器的任何一台上面的数据库存发生了改变都会同步到另一台服务器上, 这样两台服务器互为主从, 并且都能向外提供服务。

不登录mysql

```

1 #查看master状态
2 docker exec -it mysql bash -c "mysql -uroot -p12wsxCDE# -se 'show master
  status\G;'"
3
4 #查看slave状态
5 docker exec -it mysql bash -c "mysql -uroot -p12wsxCDE# -se 'show slave
  status\G;'"
6
7 #查看grep过滤后的信息
8 docker exec -it mysql bash -c "mysql -uroot -p12wsxCDE# -se 'show slave
  status\G;'"|grep -E -i 'running|host|state' |sed -rn '/^[ \t]*$/!s#^[
  \t]*##gp'
9
10 #查看uuid
11 docker exec -it mysql bash -c "mysql -uroot -p12wsxCDE# -se 'show slave
  hosts\G;'"

```

登录mysql

```

1 # master上查看master状态
2 mysql> show master status;
3 +-----+-----+-----+-----+-----+
4 | File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
5 +-----+-----+-----+-----+-----+
6 | mysql-bin.000121 |      154 | teleport     |                   |

```

```

7  +-----+-----+-----+-----+-----+
   +-----+
8  1 row in set (0.01 sec)
9  mysql>
10
11 # slave上停止已有同步
12 mysql> stop slave;
13 Query OK, 0 rows affected (0.00 sec)
14 mysql>
15
16 # slave上配置同步
17 mysql> change master to
   master_host='192.168.56.13',master_user='root',master_password='12wsxCDE#',
   master_port=3306, master_log_file='mysql-bin.000121',master_log_pos=154;
18 Query OK, 0 rows affected, 2 warnings (0.16 sec)
19 mysql>
20
21 # slave上开启同步
22 mysql> start slave;
23 Query OK, 0 rows affected (0.00 sec)
24 mysql>
25
26 # slave上查看slave同步状态
27 mysql> show slave status\G;
28 ...
29 Master_Host: 192.168.56.13
30 Slave_IO_Running: Yes
31 Slave_SQL_Running: Yes
32 ...
33 mysql>
34
35 # slave上查看master状态
36 mysql> show master status;
37 +-----+-----+-----+-----+-----+
   +-----+
38 | File                | Position | Binlog_Do_DB | Binlog_Ignore_DB |
   Executed_Gtid_Set |
39 +-----+-----+-----+-----+-----+
   +-----+
40 | mysql-bin.000114 |    1237 | teleport     |                   |
   |
41 +-----+-----+-----+-----+-----+
   +-----+
42 1 row in set (0.00 sec)
43 mysql>
44
45 # master上停止已有同步
46 mysql> stop slave;
47 Query OK, 0 rows affected (0.00 sec)
48 mysql>
49
50 # master上配置同步
51 mysql> change master to
   master_host='192.168.56.14',master_user='root',master_password='12wsxCDE#',
   master_port=3306, master_log_file='mysql-bin.000114',master_log_pos=1237;
52 Query OK, 0 rows affected, 2 warnings (0.01 sec)
53 mysql>
54

```

```

55 # master上开启同步
56 mysql> start slave;
57 Query OK, 0 rows affected (0.00 sec)
58 mysql>
59
60 # master上查看slave同步状态
61 mysql> show slave status\G;
62 ...
63 Master_Host: 192.168.56.14
64 Slave_IO_Running: Yes
65 Slave_SQL_Running: Yes
66 ...
67 mysql>

```

mysql报错解决

```

1 报错1:
2  mysql主从复制报错:The slave I/O thread stops because master and slave have
   equal MySQL server UUIDs
3
4  错误信息: Last_IO_Error: Fatal error: The slave I/O thread stops because
   master and slave have equal MySQL server UUIDs; these UUIDs must be
   different for replication to work
5
6  原因: /var/lib/mysql/auto.cnf发现里面的UUID是相同的, 为什么呢? 因为在配置主从时, 整个
   环境都是克隆的过来的。
7
8  解决方法: 备份从库的UUID(mv /var/lib/mysql/auto.cnf
   /var/lib/mysql/auto.cnf.bak)然后重启mysql就可以了, auto.cnf会自动生成一个新的。
9
10
11 报错2:
12  mysql主从复制报错:A slave with the same server_uuid as this slave has
   connected to the master
13
14  错误信息: Last_IO_Error: Got fatal error 1236 from master when reading data
   from binary log: 'A slave with the same server_uuid as this slave has
   connected to the master; the first event 'mysql-bin.000009' at 2379, the
   last event read from './mysql-bin.000009' at 3353, the last byte read from
   './mysql-bin.000009' at 3353.'
15
16  原因: slave的server_uuid 有冲突, 导致一主多从的主从复制有问题
17
18  解决方法: 在对/var/lib/mysql/auto.cnf进行备份后(mv /var/lib/mysql/auto.cnf
   /var/lib/mysql/auto.cnf.bak), 尝试对其进行删除, 然后重新启动mysql
19
20  # cat /var/lib/mysql/auto.cnf
21  [auto]
22  server-uuid=d186c795-4a2a-11e9-a756-0242ac140002

```

启动teleport

```

1 # cd /root/docker-compose/
2 # docker-compose up -d

```