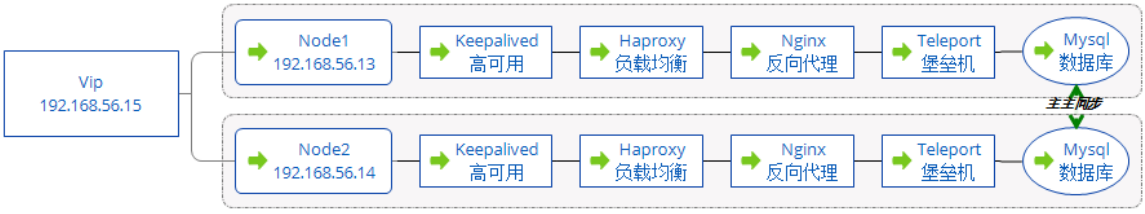
环境说明

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

```
# tree -L 3 /root/docker-compose/teleport

/root/docker-compose/teleport
├── data
│   ├── haproxy
│   │   └── haproxy.cfg
│   ├── keepalived
│   │   └── etc
│   ├── mysql
│   │   ├── data
│   │   └── etc
│   ├── nginx
│   │   ├── etc
│   │   └── html
│   └── teleport
│       ├── etc
│       ├── log
│       └── replay
├── docker-compose.yml
├── Dockerfile
├── libs
│   └── teleport-server-linux-x64-3.3.1.tar.gz
├── restart.log
└── restart.sh
```

配置文件

docker-compose配置文件

```
# cat /root/docker-compose/teleport/docker-compose.yml

version: '3.1'
services:
  mysql:
    image: mysql:5.7
    container_name: mysql
    tty: true
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - $PWD/data/mysql/etc/my.cnf:/etc/mysql/my.cnf:ro
      - $PWD/data/mysql/data:/var/lib/mysql
    restart: always
    command: [
      "--character-set-server=utf8mb4",
      "--collation-server=utf8mb4_unicode_ci",
      "--innodb_flush_log_at_trx_commit=1",
      "--sync_binlog=1"
      ]
    environment:
      MYSQL_ROOT_PASSWORD: 12wsxCDE#
      MYSQL_DATABASE: teleport
      MYSQL_USER: teleport
      MYSQL_PASSWORD: 12wsxCDE#
    ports:
      - 3306:3306
    restart: always

  teleport:
    build: .
    image: teleport:v3.3.1
    container_name: teleport
    tty: true
    depends_on:
      - mysql
    command: bash -c "/usr/local/teleport/start.sh && tail -f
/usr/local/teleport/data/log/*.log"
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - $PWD/data/teleport/etc:/usr/local/teleport/data/etc
      - $PWD/data/teleport/replay:/usr/local/teleport/data/replay
      - $PWD/data/teleport/log:/usr/local/teleport/data/log
    ports:
      - 7190:7190
      - 127.0.0.1:52080:52080
      - 52089:52089
      - 52189:52189
      - 52389:52389
    restart: always

  nginx:
    build: .
    image: nginx:latest
    container_name: nginx
    tty: true
    depends_on:
      - teleport
```

```
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - $PWD/data/nginx/etc/nginx.conf:/etc/nginx/nginx.conf
      - $PWD/data/nginx/etc/conf.d:/etc/nginx/conf.d
      # - /NWJXZ:/NWJXZ
      # - /opt/nginx/html:/opt/nginx/html
      - $PWD/data/nginx/html:/opt/nginx/html
      - /var/log/nginx:/var/log/nginx
    ports:
       - 8000:80
       - 8080:8080
    restart: always

  haproxy:
     image: haproxy
     container_name: haproxy
     tty: true
     volumes:
       - $PWD/data/haproxy/haproxy.cfg:/usr/local/etc/haproxy/haproxy.cfg:ro
     depends_on:
       - nginx
     ports:
       - "80:80"
     restart: always

  keepalived:
    image: osixia/keepalived:2.0.19
    container_name: keepalived
    tty: true
    depends_on:
      - nginx
    network_mode: "host"
    #pid: "host"
    #pid: "container:nginx"
    pid: "service:nginx"
    cap_drop:
      - NET_ADMIN
    privileged: true
    volumes:
      - $PWD/data/keepalived/etc/keepalived.conf:/usr/local/etc/keepalived/keepalived.conf:ro
      - $PWD/data/keepalived/etc/nginx_check.sh:/container/service/keepalived/assets/nginx_check.sh:ro
    restart: always
```

keepalived配置文件

192.168.56.13

```
# cat /root/docker-compose/teleport/data/keepalived/etc/keepalived.conf

global_defs {
    #这里邮件配置只对内网
```

```
    notification_email {
      #xxx@qq.com #邮件报警
    }
    #notification_email_from xxx@163.com #指定发件人
    #smtp_server smtp.163.com   #指定smtp服务器地址
    #smtp_connect_timeout 30    #指定smtp连接超时时间
    router_id LVS_DEVEL  #负载均衡标识，在局域网内应该是唯一的。
}

vrrp_script chk_nginx {
    script "/container/service/keepalived/assets/nginx_check.sh"
    #每2秒检测一次nginx的运行状态
    interval 2
    #失败一次，将自己的优先级调整为-20
    weight  -20
}

# virtual_ipaddress vip
# vrrp-虚拟路由冗余协议
# vrrp_instance 用来定义对外提供服务的VIP区域及其相关属性
vrrp_instance VI_1 {
    state BACKUP #指定该keepalived节点的初始状态
    interface enp0s8 #vrrp实例绑定的接口，用于发送VRRP包
    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
    priority 149 #指定优先级，优先级高的将成为MASTER
    nopreempt #设置为不抢占。默认是抢占的
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    ###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
    unicast_src_ip 192.168.56.13  ##本机ip
    unicast_peer { #采用单播的方式发送VRRP通告，指定单播邻居的IP地址
        192.168.56.14
    }
    virtual_ipaddress { #指定VIP地址
        192.168.56.15
    }
    #nginx存活状态检测脚本
    track_script {
        nginx_check
    }
    notify "/container/service/keepalived/assets/notify.sh"
}
```

192.168.56.14

```
# cat docker-compose/teleport/data/keepalived/etc/keepalived.conf

global_defs {
    #这里邮件配置只对内网
    notification_email {
      #xxx@qq.com #邮件报警
    }
    #notification_email_from xxx@163.com #指定发件人
    #smtp_server smtp.163.com   #指定smtp服务器地址
```

```
    #smtp_connect_timeout 30    #指定smtp连接超时时间
    router_id LVS_DEVEL  #负载均衡标识，在局域网内应该是唯一的。
}

vrrp_script chk_nginx {
    script "/container/service/keepalived/assets/nginx_check.sh"
    #每2秒检测一次nginx的运行状态
    interval 2
    #失败一次，将自己的优先级调整为-20
    weight  -20
}

# virtual_ipaddress vip
# vrrp-虚拟路由冗余协议
# vrrp_instance 用来定义对外提供服务的VIP区域及其相关属性
vrrp_instance VI_1 {
    state BACKUP #指定该keepalived节点的初始状态
    interface enp0s8 #vrrp实例绑定的接口，用于发送VRRP包
    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
    priority 149 #指定优先级，优先级高的将成为MASTER
    nopreempt #设置为不抢占。默认是抢占的
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    ###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
    unicast_src_ip 192.168.56.14  ##本机ip
    unicast_peer { #采用单播的方式发送VRRP通告，指定单播邻居的IP地址
        192.168.56.13
    }
    virtual_ipaddress { #指定VIP地址
        192.168.56.15
    }
    #nginx存活状态检测脚本
    track_script {
        nginx_check
    }
    notify "/container/service/keepalived/assets/notify.sh"
}
```

keepalived检测脚本

```
# cat /root/docker-compose/teleport/data/keepalived/etc/nginx_check.sh

#!/bin/bash
#counter=$(ps -C nginx --no-heading|wc -l)
counter=$(ps -ef|grep nginx|grep -v grep|wc -l)
if [ "${counter}" = "0" ]; then
    sleep 3
    #counter=$(ps -C nginx --no-heading|wc -l)
    counter=$(ps -ef|grep nginx|grep -v grep|wc -l)
    if [ "${counter}" = "0" ]; then
        #systemctl stop keepalived
        #pkill keepalived
        ps -ef|grep keepalived|grep -v grep|awk '{print $1}'|xargs kill -9
    fi
```

```
fi
```

haproxy配置文件

```
# cat /root/docker-compose/teleport/data/haproxy/haproxy.cfg

global
        log 127.0.0.1 local0
        log 127.0.0.1 local1 notice
defaults
        log global
        mode http
        option httplog
        option dontlognull
        timeout connect 5000ms
        timeout client 50000ms
        timeout server 50000ms
        stats uri /status

frontend http-in
    bind             *:80
    acl url_static   path_beg  -i / /static /mirrors /gitbook /pypi
    acl url_static   path_end  -i .jpg .jpeg .gif .png .ico .bmp .html

    use_backend      static_group   if url_static
    default_backend  dynamic_group

backend static_group
        balance roundrobin
    option httpchk  GET /
        http-check expect  status 200
        server ngx_server_1 192.168.56.13:8000 check
        server ngx_server_2 192.168.56.14:8000 check

backend dynamic_group
        balance roundrobin
        cookie tp_server insert indirect nocache
        server tp_server_1 192.168.56.13:8080 check cookie tp_server_1
        server tp_server_2 192.168.56.14:8080 check cookie tp_server_2
```

nginx配置文件

nginx.conf

```
# cat /root/docker-compose/teleport/data/nginx/etc/nginx.conf

user  nginx;
worker_processes  1;

error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;


events {
```

```
    worker_connections  1024;
}


http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request"
'
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    keepalive_timeout  65;

    #gzip  on;

    include /etc/nginx/conf.d/*.conf;
}
```

conf.d/*.conf

conf.d/default.conf

```
# cat /root/docker-compose/teleport/data/nginx/etc/conf.d/default.conf

server {
    listen       80;
    server_name  localhost;

    #charset koi8-r;
    access_log  /var/log/nginx/access.log  main;
    error_log  /var/log/nginx/error.log;

    location / {
        root   /opt/nginx/html;
        index  index.html index.htm;
    }

    location /nginx_status {
        stub_status on;
        access_log off;
        allow 10.75.12.143;
        allow 10.75.12.145;
        deny all;
    }

    location /mirrors {
        root /opt/nginx/html/;
        autoindex on;
        autoindex_format html;
        autoindex_exact_size off;
```

```
        autoindex_localtime on;
        charset utf-8,gbk;
    }

    location /gitbook {
       root /opt/nginx/html/;
       autoindex on;
       autoindex_format html;
       autoindex_exact_size off;
       autoindex_localtime on;
       charset utf-8,gbk;
    }

    location /pypi/web {
       root /opt/nginx/html/;
       autoindex on;
       autoindex_format html;
       autoindex_exact_size off;
       autoindex_localtime on;
       charset utf-8,gbk;
    }

    location /mirrors/backup/ {
      root /opt/nginx/html/;
      autoindex on;
      autoindex_format html;
      autoindex_exact_size off;
      autoindex_localtime on;
      allow 10.75.12.143;
      allow 10.75.12.145;
      deny all;
    }

    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

}
```

conf.d/proxy.conf

```
# cat /root/docker-compose/teleport/data/nginx/etc/conf.d/proxy.conf

upstream teleport  {
    server teleport:7190;
}

server {
    listen 8080;
    server_name  www.mxnet.io;

    access_log  /var/log/nginx/access.log  main;
    error_log  /var/log/nginx/error.log;

    location /nginx_status {
        stub_status on;
```

```
        access_log off;
        allow 10.75.12.143;
        deny all;
    }

    location / {
        proxy_pass  http://teleport;

        #Proxy Settings
        proxy_redirect     off;
        proxy_set_header   Host             $host;
        proxy_set_header   X-Real-IP        $remote_addr;
        proxy_set_header   X-Forwarded-For  $proxy_add_x_forwarded_for;
        proxy_next_upstream error timeout invalid_header http_500 http_502
http_503 http_504;
        proxy_max_temp_file_size 0;
        proxy_connect_timeout      90;
        proxy_send_timeout         90;
        proxy_read_timeout         90;
        proxy_buffer_size          4k;
        proxy_buffers              4 32k;
        proxy_busy_buffers_size    64k;
        proxy_temp_file_write_size 64k;

        #以下三行是websocket需要的
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
   }

}
```

teleport配置文件

web.ini

```
# cat /root/docker-compose/teleport/data/teleport/etc/web.ini

; codec: utf-8

[common]

; ip=0.0.0.0

; port listen by web server, default to 7190.
; DO NOT FORGET update `common::web-server-rpc` in core.ini if you modified
this setting.
port=7190

; log file of web server, default to /var/log/teleport/tpweb.log
log-file=/usr/local/teleport/data/log/tpweb.log

; `log-level` can be 0 ~ 4, default to 2.
; LOG_LEVEL_DEBUG     0   log every-thing.
; LOG_LEVEL_VERBOSE   1   log every-thing but without debug message.
; LOG_LEVEL_INFO      2   log information/warning/error message.
```

```
; LOG_LEVEL_WARN      3   log warning and error message.
; LOG_LEVEL_ERROR     4   log error message only.
log-level=0

; 0/1. default to 0.
; in debug mode, `log-level` force to 0 and display more message for debug purpose.
debug-mode=0

; `core-server-rpc` is the rpc interface of core server.
; default to `http://127.0.0.1:52080/rpc`.
; DO NOT FORGET update this setting if you modified rpc::bind-port in core.ini.
core-server-rpc=http://127.0.0.1:52080/rpc

[database]

; database in use, should be sqlite/mysql, default to sqlite.
type=mysql

; sqlite-file=/usr/local/teleport/data/db/teleport.db

mysql-host=mysql

mysql-port=3306

mysql-db=teleport

mysql-prefix=tp_

mysql-user=teleport

mysql-password=12wsxCDE#
```

core.ini

```
# cat /root/docker-compose/teleport/data/teleport/etc/core.ini

; codec: utf-8

[common]
; 'log-file' define the log file location. if not set, default locate
; to $INSTDIR%/data/log/tpcore.log
;log-file=/var/log/teleport/tpcore.log

; log-level can be 0 ~ 4, default value is 2.
; LOG_LEVEL_DEBUG     0   log every-thing.
; LOG_LEVEL_VERBOSE   1   log every-thing but without debug message.
; LOG_LEVEL_INFO      2   log infomation/warning/error message.
; LOG_LEVEL_WARN      3   log warning and error message.
; LOG_LEVEL_ERROR     4   log error message only.
log-level=2

; 0/1. default to 0.
; in debug mode, `log-level` force to 0 and display more message for debug purpose.
debug-mode=0
```

```
; 'replay-path' define the replay file location. if not set, default locate
; to `$INSTDIR%/data/replay`
;replay-path=/var/lib/teleport/replay

; `web-server-rpc` is the rpc interface of web server.
; default to `http://127.0.0.1:7190/rpc`.
; DO NOT FORGET update this setting if you modified common::port in
web.ini.
web-server-rpc=http://127.0.0.1:7190/rpc

[rpc]
; Request by web server. `bind-ip` should be the ip of core server. If web
server and
; core server running at the same machine, it should be `127.0.0.1`.
; DO NOT FORGET update `common::core-server-rpc` in web.ini if you modified
this setting.
bind-ip=127.0.0.1
bind-port=52080

[protocol-ssh]
enabled=true
lib=tpssh
bind-ip=0.0.0.0
bind-port=52189

[protocol-rdp]
enabled=true
lib=tprdp
bind-ip=0.0.0.0
bind-port=52089

[protocol-telnet]
enabled=true
lib=tptelnet
bind-ip=0.0.0.0
bind-port=52389
```

mysql配置文件

192.168.56.13

```
# cat /root/docker-compose/teleport/data/mysql/etc/my.cnf

[mysql]

[mysqld]
pid-file    = /var/run/mysqld/mysqld.pid
socket      = /var/run/mysqld/mysqld.sock
datadir     = /var/lib/mysql
#log-error  = /var/log/mysql/error.log
# By default we only accept connections from localhost
#bind-address   = 127.0.0.1
# Disabling symbolic-links is recommended to prevent assorted security
risks
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

```
#查看master状态
docker exec -it mysql bash -c "mysql -uroot -p12wsxCDE# -se 'show master status\G;'"

#查看slave状态
docker exec -it mysql bash -c "mysql -uroot -p12wsxCDE# -se 'show slave status\G;'"

#查看grep过滤后的信息
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'

#查看uuid
docker exec -it mysql bash -c "mysql -uroot -p12wsxCDE# -se 'show slave hosts\G;'"
```

登录mysql

```
# master上查看master状态
mysql> show master status;
+------------------+----------+--------------+------------------+-------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+------------------+----------+--------------+------------------+-------------------+
| mysql-bin.000121 |      154 | teleport     |                  |                   |
+------------------+----------+--------------+------------------+-------------------+
1 row in set (0.01 sec)
mysql>

# slave上停止已有同步
mysql> stop slave;
Query OK, 0 rows affected (0.00 sec)
mysql>

# slave上配置同步
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;
Query OK, 0 rows affected, 2 warnings (0.16 sec)
mysql>

```

```
# slave上开启同步
mysql> start slave;
Query OK, 0 rows affected (0.00 sec)
mysql>

# slave上查看slave同步状态
mysql> show slave status\G;
...
Master_Host: 192.168.56.13
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
...
mysql>

# slave上查看master状态
mysql> show master status;
+------------------+----------+--------------+------------------+-------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+------------------+----------+--------------+------------------+-------------------+
| mysql-bin.000114 |     1237 | teleport     |                  |                   |
+------------------+----------+--------------+------------------+-------------------+
1 row in set (0.00 sec)
mysql>

# master上停止已有同步
mysql> stop slave;
Query OK, 0 rows affected (0.00 sec)
mysql>

# master上配置同步
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;
Query OK, 0 rows affected, 2 warnings (0.01 sec)
mysql>

# master上开启同步
mysql> start slave;
Query OK, 0 rows affected (0.00 sec)
mysql>

# master上查看slave同步状态
mysql> show slave status\G;
...
Master_Host: 192.168.56.14
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
...
mysql>
```

mysql报错解决

报错1:
mysql主从复制报错:The slave I/O thread stops because master and slave have equal MySQL server UUIDs

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

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。

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。

报错2:
mysql主从复制报错:A slave with the same server_uuid as this slave has connected to the master

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.'

原因：Slave的server_uuid 有冲突，导致一主多从的主从复制有问题

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

```
# cat /var/lib/mysql/auto.cnf
[auto]
server-uuid=d186c795-4a2a-11e9-a756-0242ac140002
```