???????

□□□□□□□□
#□□□□□
yum update -y
#□□□□□
sudo hostnamectl set-hostname rulr-1
#□□□□
ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime &&\
yum install ntpdate -y &&\
(crontab -l 2>/dev/null; echo "*/5 * * * * /usr/sbin/ntpdate ntp.aliyun.com >/dev/null 2>&1") |
crontab - &&\
crontab -l
#□□ 1
cat >> /etc/sysctl.conf << EOF
kernel.pid_max = 4194303
#net.ipv4.tcp_tw_recycle = 0
net.ipv4.tcp_tw_reuse = 1
net.ipv4.ip_local_port_range = 1024 65000
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_tw_buckets = 20480
net.ipv4.tcp_max_syn_backlog = 20480
net.core.netdev_max_backlog = 262144
net.ipv4.tcp_fin_timeout = 20
vm.swappiness = 0
net.core.somaxconn = 20480
vm.overcommit_memory = 1
net.core.rmem_default = 262144
net.core.rmem_max = 16777216
net.core.wmem_default = 262144
net.core.wmem_max = 16777216
net.ipv4.tcp_wmem = 4096 131072 1048576
net.ipv4.tcp_rmem = 4096 131072 1048576
net.ipv4.tcp_mem = 786432 1048576 1572864
net.ipv4.tcp_max_orphans = 262144
net.ipv4.tcp_fastopen = 3
fs.aio-max-nr = 1048576
net.ipv4.tcp_keepalive_time = 1800
net.ipv4.tcp_keepalive_probes = 3
net.ipv4.tcp_keepalive_intvl = 15
net.ipv4.route.gc_timeout = 100
net.ipv4.tcp_syn_retries = 1
net.ipv4.tcp_synack_retries = 1

```
EOF

sysctl -p

#优化  2
cat >> /etc/security/limits.conf << EOF
* soft     nproc          65535
* hard     nproc          65535
* soft     nofile         65535
* hard     nofile         65535
root soft     nproc          65535
root hard     nproc          65535
root soft     nofile         65535
root hard     nofile         65535
EOF

su -
ulimit -n

# 查看系统语言
echo $LANG
en_US.UTF-8
# 如果系统语言不是 "en_US.UTF-8"时按下面的步骤进行修改，选中 en_US.UTF-8 UTF-8
# dpkg-reconfigure locales
上下键 ：连续内容上下移动光标 tab键 ：不连续内容移动光标 space键 ：选中内容，确认

docker安装包下载目录: wget http://mirrors.aliyun.com/docker-ce/linux/static/stable/x86_64/docker-26.1.4.tgz
docker-compose安装包下载目录: wget https://github.com/docker/compose/releases/download/1.29.2/docker-compose-Linux-x86_64

上传 docker-skylin.zip
cd /home/shuncom &&\
unzip docker-skylin.zip &&\
mkdir -p /docker/data &&\
mkdir -p /etc/docker &&\
mv /home/shuncom/docker-26.1.4.tgz /docker/ &&\
cd /docker &&\
tar -zxvf docker-26.1.4.tgz &&\
mv docker/* /usr/bin/ &&\
dockerd

cat > /usr/lib/systemd/system/docker.service << EOF
[Unit]
Description=Docker Application Container Engine
Documentation=https://docs.docker.com
```

```
After=network-online.target firewalld.service
Wants=network-online.target
[Service]
Type=notify
ExecStart=/usr/bin/dockerd
ExecReload=/bin/kill -s HUP $MAINPID
LimitNOFILE=infinity
LimitNPROC=infinity
TimeoutStartSec=0
Delegate=yes
KillMode=process
Restart=on-failure
StartLimitBurst=3
StartLimitInterval=60s
[Install]
WantedBy=multi-user.target
EOF

# 为 docker.service添加执行权限
chmod +x /usr/lib/systemd/system/docker.service

# 重新加载系统控制器
systemctl daemon-reload


cat > /etc/docker/daemon.json << EOF
{
      "data-root":"/docker/data"
}
EOF

# 启动  docker
systemctl start docker

# 开机自启动
systemctl enable docker

# 验证  docker 查看  docker 版本:
docker --version
docker run hello-world

#安装  docker-compose
cp /home/shuncom/docker-compose-Linux-x86_64_1.29.2 /usr/bin/docker-compose && chmod +x
/usr/bin/docker-compose &&\
groupadd docker &&\
usermod -aG docker shuncom &&\
```

newgrp docker &&\
groups shuncom &&\
id shuncom

chown root:docker /var/run/docker.sock &&\
chmod 660 /var/run/docker.sock

rulr-1
docker network create --subnet=172.18.0.0/16 rulr-network &&\
docker network ls

sudo firewall-cmd --permanent --add-port=443/tcp &&\
sudo firewall-cmd --permanent --add-port=80/tcp &&\
sudo firewall-cmd --permanent --add-port=8070/tcp &&\
sudo firewall-cmd --permanent --add-port=20881/tcp &&\
sudo firewall-cmd --permanent --add-port=8082/tcp &&\
sudo firewall-cmd --permanent --add-port=20835/tcp &&\
sudo firewall-cmd --permanent --add-port=8086/tcp &&\
sudo firewall-cmd --permanent --add-source=0.0.0.0/0 &&\
sudo firewall-cmd --reload &&\
sudo firewall-cmd --list-all

□□□□□□□ GitLab□□□
□□ www□ servers□ 30.16□□

cd /home/shuncom
tar -zxvf servers2024xxxx.tar.gz
tar -zxvf www2024xxxx.tar.gz

mkdir -p /home/shuncom/log/dubbo/registry/cache &&\
mkdir -p /home/shuncom/log/nginx &&\
mkdir -p /home/shuncom/log/shuncom-oauth2-server/tomcat &&\
mkdir -p /home/shuncom/log/web-maintain &&\
mkdir -p /home/shuncom/log/web-things &&\
mkdir -p /home/shuncom/log/web-user

cat > /home/shuncom/compose/.env << EOF
RULR_MYSQL=192.168.30.97
RULR_ZOOKEEPER=192.168.30.95
RULR_REDIS=192.168.30.95
RULR_INFLUXDB=192.168.30.96
RULR_RABBITMQ=192.168.30.95
HOSTIP=192.168.30.91
WEB_HOST=192.168.30.91
WEB_PORT=443

```
RULR_1=192.168.30.91
RULR_3=192.168.30.93
RULR_4=192.168.30.94
JAVA_IMAGE=openjdk:8-alpine3.9-tini
EOF

sudo chown -R root.root /home/shuncom/www &&\
sudo chown -R shuncom.shuncom /home/shuncom/www/rulr

cd /home/shuncom/compose &&\
docker image load -i nginx_1.20.2.tar &&\
docker image load -i openjdk_8-alpine3.9.tar &&\
docker image load -i openjdk_8-alpine3.9-tini.tar

docker-compose build &&\
docker-compose up -d &&\
docker-compose ps

rulr-2
docker network create --subnet=172.18.0.0/16 rulr-network &&\
docker network ls

sudo firewall-cmd --permanent --add-port=20818/tcp &&\
sudo firewall-cmd --permanent --add-port=8040/tcp &&\
sudo firewall-cmd --permanent --add-port=20817/tcp &&\
sudo firewall-cmd --permanent --add-port=8041/tcp &&\
sudo firewall-cmd --permanent --add-port=20836/tcp &&\
sudo firewall-cmd --permanent --add-port=9922/tcp &&\
sudo firewall-cmd --permanent --add-port=20856/tcp &&\
sudo firewall-cmd --permanent --add-source=0.0.0.0/0 &&\
sudo firewall-cmd --reload &&\
sudo firewall-cmd --list-all

□□□□□□□          GitLab□□□
□□  www□ servers□ 30.17□□

cd /home/shuncom
tar -zxvf servers2024xxxx.tar.gz
tar -zxvf www2024xxxx.tar.gz

mkdir -p /home/shuncom/log/dtumt-connect &&\
mkdir -p /home/shuncom/log/engine-action-server &&\
mkdir -p /home/shuncom/log/engine-alarm-server &&\
mkdir -p /home/shuncom/log/engine-timing-server &&\
mkdir -p /home/shuncom/log/engine-trigger-server &&\
mkdir -p /home/shuncom/log/model-converter-server &&\
mkdir -p /home/shuncom/log/model-event-server &&\
```

```
mkdir -p /home/shuncom/log/model-proxy-server &&\
mkdir -p /home/shuncom/log/model-task-schedule

cat > /home/shuncom/compose/.env << EOF
RULR_MYSQL=192.168.30.97
RULR_ZOOKEEPER=192.168.30.95
RULR_REDIS=192.168.30.95
RULR_INFLUXDB=192.168.30.96
RULR_RABBITMQ=192.168.30.95
HOSTIP=192.168.30.92
WEB_HOST=192.168.30.91
WEB_PORT=443
RULR_1=192.168.30.91
RULR_3=192.168.30.93
RULR_4=192.168.30.94
JAVA_IMAGE=openjdk:8-alpine3.9-tini
EOF

cd /home/shuncom/compose &&\
docker load -i openjdk_8-ffmpeg.tar &&\
docker load -i openjdk_8-alpine3.9.tar &&\
docker load -i openjdk_8-alpine3.9-tini.tar

docker-compose build &&\
docker-compose up -d &&\
docker-compose ps

rulr-3
docker network create --subnet=172.18.0.0/16 rulr-network
docker network ls

sudo firewall-cmd --permanent --add-port=443/tcp &&\
sudo firewall-cmd --permanent --add-port=80/tcp &&\
sudo firewall-cmd --permanent --add-port=20815/tcp &&\
sudo firewall-cmd --permanent --add-port=9911/tcp &&\
sudo firewall-cmd --permanent --add-port=20821/tcp &&\
sudo firewall-cmd --permanent --add-port=20829/tcp &&\
sudo firewall-cmd --permanent --add-port=8074/tcp &&\
sudo firewall-cmd --permanent --add-port=8083/tcp &&\
sudo firewall-cmd --permanent --add-port=20846/tcp &&\
sudo firewall-cmd --permanent --add-source=0.0.0.0/0 &&\
sudo firewall-cmd --reload &&\
sudo firewall-cmd --list-all

□□□□□□          GitLab□□□
□□  www□ servers□ 30.18□□
```

```
cd /home/shuncom
tar -zxvf servers2024xxxx.tar.gz
tar -zxvf www2024xxxx.tar.gz

mkdir -p /home/shuncom/log/advice-server &&\
mkdir -p /home/shuncom/log/batch-proxy-server &&\
mkdir -p /home/shuncom/log/dtumt-connect &&\
mkdir -p /home/shuncom/log/dubbo/registry/cache &&\
mkdir -p /home/shuncom/log/model-data-resolver &&\
mkdir -p /home/shuncom/log/model-task-batch &&\
mkdir -p /home/shuncom/log/nginx &&\
mkdir -p /home/shuncom/log/pph-proxy-handler/tomcat &&\
mkdir -p /home/shuncom/log/push-tripartite &&\
mkdir -p /home/shuncom/log/shuncom-onbon-server


cat > /home/shuncom/compose/.env << EOF
RULR_MYSQL=192.168.30.97
RULR_ZOOKEEPER=192.168.30.95
RULR_REDIS=192.168.30.95
RULR_INFLUXDB=192.168.30.96
RULR_RABBITMQ=192.168.30.95
HOSTIP=192.168.30.93
WEB_HOST=192.168.30.91
WEB_PORT=443
RULR_1=192.168.30.91
RULR_3=192.168.30.93
RULR_4=192.168.30.94
JAVA_IMAGE=openjdk:8-alpine3.9-tini
EOF

cd /home/shuncom/compose/ &&\
docker image load -i nginx_1.20.2.tar &&\
docker image load -i openjdk_8-alpine3.9.tar &&\
docker image load -i openjdk_8-alpine3.9-tini.tar &&\
docker-compose build &&\
docker-compose up -d &&\
docker-compose ps

rulr-4
docker network create --subnet=172.18.0.0/16 rulr-network &&\
docker network ls

sudo firewall-cmd --permanent --add-port=8092/tcp &&\
sudo firewall-cmd --permanent --add-port=8093/tcp &&\
sudo firewall-cmd --permanent --add-port=20810/tcp &&\
```

```
sudo firewall-cmd --permanent --add-port=8096/tcp &&\
sudo firewall-cmd --permanent --add-port=20826/tcp &&\
sudo firewall-cmd --permanent --add-port=8094/tcp &&\
sudo firewall-cmd --permanent --add-port=20827/tcp &&\
sudo firewall-cmd --permanent --add-port=8060/tcp &&\
sudo firewall-cmd --permanent --add-port=8070/tcp &&\
sudo firewall-cmd --permanent --add-port=20877/tcp &&\
sudo firewall-cmd --permanent --add-port=8087/tcp &&\
sudo firewall-cmd --permanent --add-port=20824/tcp &&\
sudo firewall-cmd --permanent --add-port=20811/tcp &&\
sudo firewall-cmd --permanent --add-port=20825/tcp &&\
sudo firewall-cmd --permanent --add-port=20828/tcp &&\
sudo firewall-cmd --permanent --add-port=6011/tcp &&\
sudo firewall-cmd --permanent --add-port=6021/tcp &&\
sudo firewall-cmd --permanent --add-port=6031/tcp &&\
sudo firewall-cmd --permanent --add-port=6041/tcp &&\
sudo firewall-cmd --permanent --add-port=6051/tcp &&\
sudo firewall-cmd --permanent --add-port=6061/tcp &&\
sudo firewall-cmd --permanent --add-port=6071/tcp &&\
sudo firewall-cmd --permanent --add-port=6081/tcp &&\
sudo firewall-cmd --permanent --add-port=6091/tcp &&\
sudo firewall-cmd --permanent --add-source=0.0.0.0/0 &&\
sudo firewall-cmd --reload &&\
sudo firewall-cmd --list-all
```

□□□□□□ GitLab□□□
□□ www□ servers□ 30.41□□

```
cd /home/shuncom
tar -zxvf servers2024xxxx.tar.gz
```

```
mkdir -p /home/shuncom/log/dtumt-connect &&\
mkdir -p /home/shuncom/log/dtumt-handler &&\
mkdir -p /home/shuncom/log/dtumt-transmission &&\
mkdir -p /home/shuncom/log/dubbo/registry/cache &&\
mkdir -p /home/shuncom/log/gateway-connect &&\
mkdir -p /home/shuncom/log/gateway-handler &&\
mkdir -p /home/shuncom/log/gateway-proxy-handler &&\
mkdir -p /home/shuncom/log/light-connect-gateway &&\
mkdir -p /home/shuncom/log/light-connect-tcp &&\
mkdir -p /home/shuncom/log/light-proxy-handler &&\
mkdir -p /home/shuncom/log/mt-event-server &&\
mkdir -p /home/shuncom/log/nginx &&\
mkdir -p /home/shuncom/log/tripartite-connect &&\
mkdir -p /home/shuncom/log/tripartite-connect-sensor
```

```
cat > /home/shuncom/compose/.env << EOF
RULR_MYSQL=192.168.30.97
RULR_ZOOKEEPER=192.168.30.95
RULR_REDIS=192.168.30.95
RULR_INFLUXDB=192.168.30.96
RULR_RABBITMQ=192.168.30.95
HOSTIP=192.168.30.94
WEB_HOST=192.168.30.91
WEB_PORT=443
RULR_1=192.168.30.91
RULR_3=192.168.30.93
RULR_4=192.168.30.94
JAVA_IMAGE=openjdk:8-alpine3.9-tini
EOF

cd /home/shuncom/compose/ &&\
docker image load -i nginx-tcp-24.08.19.tar &&\
docker image load -i nginx-tcp.tar &&\
docker image load -i openjdk_8-alpine3.9.tar &&\
docker image load -i openjdk_8-alpine3.9-tini.tar

□□  docker-compose.yml
#  nginx-tcp:
#    image: nginx-tcp
#    build:
#      context: ./
#      dockerfile: NginxTcp

162□□□     image: nginx-tcp:24.08.19

docker-compose build &&\
docker-compose up -d &&\
docker-compose ps

rulr-5
docker network create --subnet=172.18.0.0/16 rulr-network &&\
docker network ls

sudo firewall-cmd --permanent --add-port=5672/tcp &&\
sudo firewall-cmd --permanent --add-port=15672/tcp &&\
sudo firewall-cmd --permanent --add-port=6379/tcp &&\
sudo firewall-cmd --permanent --add-port=2181/tcp &&\
sudo firewall-cmd --permanent --add-source=0.0.0.0/0 &&\
sudo firewall-cmd --reload &&\
sudo firewall-cmd --list-all
```

```
mkdir -p /home/shuncom/rulr/conf/redis &&\
mkdir -p /home/shuncom/logs/rabbitmq &&\
mkdir -p /home/shuncom/rulr/conf/rabbitmq &&\
chmod 757 /home/shuncom/logs/rabbitmq &&\
chmod 757 /home/shuncom/rulr/conf/rabbitmq &&\
mkdir -p /home/shuncom/rulr/conf/zookeeper &&\
mkdir -p /home/shuncom/compose


cat > /home/shuncom/compose/.env << EOF
RULR_MYSQL=192.168.30.97
RULR_ZOOKEEPER=192.168.30.95
RULR_REDIS=192.168.30.95
RULR_INFLUXDB=192.168.30.96
RULR_RABBITMQ=192.168.30.95
HOSTIP=192.168.30.95
WEB_HOST=192.168.30.91
WEB_PORT=443
RULR_1=192.168.30.91
RULR_3=192.168.30.93
RULR_4=192.168.30.94
JAVA_IMAGE=openjdk:8-alpine3.9-tini
EOF
```

3 上传应用配置文件

```
cd /home/shuncom/compose &&\
cp redis-5.0.14/6379.conf ../rulr/conf/redis/ &&\
cp rabbitmq-3.7.26/rabbitmq.conf ../rulr/conf/rabbitmq/ &&\
chmod 667 /home/shuncom/rulr/conf/rabbitmq/rabbitmq.conf &&\
cp zookeeper-3.5.8/log4j.properties ../rulr/conf/zookeeper/ &&\
cp zookeeper-3.5.8/zoo.cfg ../rulr/conf/zookeeper/
```

(4) compose

```
# 配置文件上传          (docker-compose.yml)
# cd /home/shuncom/compose

docker image load -i rabbitmq_3.7.26.tar &&\
docker image load -i redis_5.0.14.tar &&\
docker image load -i zookeeper_3.5.8.tar

vi docker-compose.yml
```

删除 mysql infuxdb部分内容

4 启动应用服务并查看服务状态

```
cd /home/shuncom/compose &&\
docker-compose up -d &&\
```

docker-compose ps

5 rabbitmq开启 web管理
docker exec -it rulr-rabbit bash
rabbitmq-plugins enable rabbitmq_management
exit

6 应用访问测试
(1) 访问 redis
docker exec -it rulr-redis bash
redis-cli -h 127.0.0.1 -a 'sz_clighting' -p 6379
127.0.0.1:6379> CONFIG get maxclients
1) "maxclients"
2) "10000"
127.0.0.1:6379> keys *
(empty list or set)
127.0.0.1:6379> exit
root@f39b51dab023:/data# redis-cli --version
redis-cli 5.0.14
root@f39b51dab023:/data# exit

(2) 访问 rabbitmq
URL: http://10.0.3.115:15672/
Username: shuncom_mq
Password: sz_clighting_mq

需要开启 WEB管理。可通过以下方式（查看服务状态）：
docker exec -it rulr-rabbit rabbitmqctl status

(3) 访问 zookeeper
docker exec -it rulr-zookeeper bash
root@2ad68fbb302e:/apache-zookeeper-3.5.8-bin# zkCli.sh -server 127.0.0.1:2181
[zk: 127.0.0.1:2181(CONNECTED) 0] ls /
[zookeeper]
[zk: 127.0.0.1:2181(CONNECTED) 1] quit
root@2ad68fbb302e:/apache-zookeeper-3.5.8-bin# exit

rulr-6

3.4 部署 InfluxDB

docker network create --subnet=172.18.0.0/16 rulr-network &&\
docker network ls

sudo firewall-cmd --permanent --add-port=8086/tcp &&\
sudo firewall-cmd --permanent --add-port=111/udp &&\
sudo firewall-cmd --permanent --add-port=2049/tcp &&\

```
sudo firewall-cmd --permanent --add-source=0.0.0.0/0 &&\
sudo firewall-cmd --reload &&\
sudo firewall-cmd --list-all

□□□□□□□□□              ip□□□□
sudo firewall-cmd --permanent --zone=public --add-source=0.0.0.0/0 &&\
sudo firewall-cmd --permanent --zone=public --add-port=1-65535/tcp &&\
sudo firewall-cmd --permanent --zone=public --add-port=1-65535/udp &&\
sudo firewall-cmd --reload &&\
sudo firewall-cmd --list-all

mkdir -p /home/shuncom/rulr/conf/influxdb &&\
mkdir -p /home/shuncom/influxdb/var/lib/influxdb/meta &&\
mkdir -p /home/shuncom/influxdb/var/lib/influxdb/wal &&\
mkdir -p /home/shuncom/influxdb/var/lib/influxdb/data &&\
mkdir -p /home/shuncom/influxdb/var/log/influxdb &&\
mkdir -p /home/shuncom/influxdb/etc/logrotate.d

tar -zxvf influxdb-1.8.10_linux_amd64.tar.gz &&\
mv influxdb-1.8.10-1 influxdb &&\
useradd -M -s /usr/sbin/nologin influxdb

id influxdb
uid=1601(influxdb) gid=1601(influxdb) groups=1601(influxdb)

3 □□   influxdb□□□□
cat > /home/shuncom/influxdb/etc/influxdb/influxdb.conf << EOF
reporting-disabled = false
bind-address = "127.0.0.1:8088"
[meta]
  dir = "/home/shuncom/influxdb/var/lib/influxdb/meta"
[data]
  dir = "/home/shuncom/influxdb/var/lib/influxdb/data"
  wal-dir = "/home/shuncom/influxdb/var/lib/influxdb/wal"
  series-id-set-cache-size = 100
[coordinator]
[retention]
[shard-precreation]
[monitor]
[http]
  enabled = true
  bind-address = ":8086"
  auth-enabled = false
[logging]
[subscriber]
[[graphite]]
```

```
[[collectd]]
[[opentsdb]]
[[udp]]
[continuous_queries]
[tls]
EOF

4 将 InfluxDB服务加入       systemd
cat > /lib/systemd/system/influxdb.service << EOF
[Unit]
Description=InfluxDB is an open-source, distributed, time series database
Documentation=https://docs.influxdata.com/influxdb/
After=network-online.target

[Service]
#以 influxdb用户启动无执行权限要用                    root用户启动
User=root
Group=root
LimitNOFILE=65536
ExecStart=/home/shuncom/influxdb/usr/bin/influxd -config
/home/shuncom/influxdb/etc/influxdb/influxdb.conf \$INFLUXD_OPTS
KillMode=control-group
Restart=on-failure
RestartSec=2s
[Install]
WantedBy=multi-user.target
Alias=influxdb.service
EOF

5 设置  InfluxDB日志
cat > /etc/rsyslog.d/influxdb.conf << EOF
### InfluxDB Rules ###
if \$programname == 'influxd' then {
  action(type="omfile" file="/home/shuncom/influxdb/var/log/influxdb/influxd.log")
  stop
}
EOF

# 重启  rsyslog服务
systemctl restart rsyslog.service

# 配置日志切割

# 修改配置文件并加入系统定时日志切割任务
cat > /home/shuncom/influxdb/etc/logrotate.d/influxdb << EOF
/home/shuncom/influxdb/var/log/influxdb/influxd.log {
```

```
    daily
    rotate 7
    missingok
    dateext
    copytruncate
    compress
}
EOF

chown -R influxdb.influxdb /home/shuncom/influxdb/ &&\
chmod 757 /home/shuncom/influxdb/var/log/influxdb/

6 启动  InfluxDB
systemctl daemon-reload &&\
systemctl start influxdb.service &&\
systemctl enable influxdb.service &&\
systemctl status influxdb.service

cat >> /etc/profile << EOF
export PATH=/home/shuncom/influxdb/usr/bin:\$PATH
EOF

source /etc/profile

cat >> ~/.bashrc << EOF
export PATH=/home/shuncom/influxdb/usr/bin:\$PATH
EOF

source ~/.bashrc


influx -version
InfluxDB shell version: 1.8.10

7 InfluxDB管理员创建
(1) 创建管理员
influx            # 进入命令行
> show users        # 展示用户及其权限
user admin
---- -----
> create user root with password 'root_influxdb' with all privileges        # 添加一个管理员用户
> show users                                                  # 展示用户及其权限
user admin
---- -----
root true
> exit
```

(2) 开启 influxdb认证
修改 influxdb.conf配置文件，将 [http]标签下 "auth-enabled = false"改为 "auth-enabled = true"
vim /home/shuncom/influxdb/etc/influxdb/influxdb.conf

(3) 重启 InfluxDB服务
systemctl restart influxdb.service

(4) InfluxDB认证登录
influx -host 127.0.0.1 -port 8086 -username root -password 'root_influxdb'
> show users
user admin
---- -----
root true

create database rulr_things;

2 设置库数据保留规则
use rulr_things;
CREATE RETENTION POLICY three_days ON rulr_things DURATION 3d REPLICATION 1;
CREATE RETENTION POLICY seven_days ON rulr_things DURATION 7d REPLICATION 1;
CREATE RETENTION POLICY half_year ON rulr_things DURATION 180d REPLICATION 1;
show retention policies;

3 创建用户
create user shuncom with password 'shuncom_influxdb_passwd';
grant all privileges on rulr_things to shuncom;
show grants for shuncom;

rulr-7

sudo firewall-cmd --permanent --add-port=3306/tcp &&\
sudo firewall-cmd --permanent --add-source=0.0.0.0/0 &&\
sudo firewall-cmd --reload &&\
sudo firewall-cmd --list-all

针对 oceanbase数据库

7 修改 mysql数据库密码
grep 'temporary password' /home/shuncom/mysql/data/mysqld.log
2022-09-06T07:24:22.857237Z 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: x2z?tj79y,dI

mysql -uroot -p'kfcmyi0.g)lZ'
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'Sz_clighting';
mysql> flush privileges;
mysql> exit;

1 创建库、创建用户并赋权

```
mysql -u root -p'Sz_clighting'
mysql> CREATE DATABASE rulr_things DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci;
mysql> create user "shuncom"@"%" identified by "sh_clighting";
mysql> grant all on rulr_things.* to "shuncom"@"%";
```

2 导入表结构和表数据

```
mkdir -p /home/shuncom/update_mysql/ &&\
chmod +x /home/shuncom/update_mysql/update_mysql.sh
```

2、如果初始化脚本低于当前部署版本，应用迁移工具（ rulr-db-migrate_vx.x.x.jar）进行升级

(2) 部署 jdk

```
cd update_mysql &&\
tar -xzf jdk-8u251-linux-x64.tar.gz &&\
mv jdk1.8.0_251/ /usr/local/jdk/
```

(3) 将 jdk加入环境变量并生效

```
cat >>/etc/profile<< EOF
export JAVA_HOME=/usr/local/jdk
export CLASSPATH=\$JAVA_HOME/lib/tools.jar:\$JAVA_HOME/jre/lib/rt.jar
export PATH=\$JAVA_HOME/bin:\$PATH
EOF

source /etc/profile

cat >> ~/.bashrc << EOF
export JAVA_HOME=/usr/local/jdk
export CLASSPATH=\$JAVA_HOME/lib/tools.jar:\$JAVA_HOME/jre/lib/rt.jar
export PATH=\$JAVA_HOME/bin:\$PATH
EOF

source ~/.bashrc
```

(4) 查看 jdk版本

```
java -version
java version "1.8.0_251"
```

当前 SQL导入与维护方式已更新!

1、执行初始化 sql脚本

```
mysql -u shuncom -p'sh_clighting'
use rulr_things
source /home/shuncom/update_mysql/v3.11.0__aliyun_init_db_dump.sql
```

cd /home/shuncom/update_mysql
vi update_mysql.sh
修改初始版本（本例是 v3.11.0），升级版本 v3.11.1
修改 jdk路径 /usr/local/jdk/

NOT_MYSQL，如果非 mysql数据库，值填写 true， mysql数据库则写 false
现有的本地化部署的脚本不需要更新，这个字段是针对 oceanbase做兼容用的
NOT_MYSQL="true"

保存退出

./update_mysql.sh

mkdir -p log/shuncom-db-migrate

手动升级 v3.11.2
mysql -u shuncom -p'sh_clighting'
use rulr_things
source /home/shuncom/update_mysql/v3.11.2/mysql__v3.11.2_ddl.sql
source /home/shuncom/update_mysql/v3.11.2/mysql__v3.11.2_dml.sql

---

Revision #1
Created 14 May 2025 06:14:42 by Admin
Updated 14 May 2025 06:20:02 by Admin