

---

# 长虹物联网平台消息系统安装部署说明书

***Release 1.0***

**<contact@emqtt.io>**

**Jan 11, 2018**



<b>1</b>	<b>安装部署2台Redis+Sentinel集群</b>	<b>1</b>
1.1	安装部署环境 . . . . .	1
1.2	部署Redis服务 . . . . .	1
<b>2</b>	<b>安装部署3台Zookeeper+Kafka集群</b>	<b>3</b>
2.1	安装部署环境 . . . . .	3
2.2	部署Zookeeper服务 . . . . .	3
2.3	部署Kafka服务 . . . . .	4
2.4	测试Zookeeper+Kafka . . . . .	5
<b>3</b>	<b>安装部署3台EMQX集群</b>	<b>7</b>
3.1	安装部署环境 . . . . .	7
<b>4</b>	<b>安装部署haproxy，实现3台emqx负载均衡</b>	<b>9</b>
4.1	安装部署环境 . . . . .	9
4.2	安装HAProxy服务 . . . . .	9
4.3	部署HAProxy服务 . . . . .	9



---

## 安装部署2台Redis+Sentinel集群

---

### 1.1 安装部署环境

2台物理机器:

```
节点1 192.168.1.101 centos7.3  
节点2 192.168.1.102 centos7.3
```

每个节点都安装Redis服务:

```
cd /opt  
wget http://download.redis.io/releases/redis-4.0.2.tar.gz  
tar xzf redis-4.0.2.tar.gz  
cd redis-4.0.2  
yum install gcc  
make  
make install
```

### 1.2 部署Redis服务

192.168.1.101节点创建主redis:

```
mkdir redis-master  
cp /opt/redis-4.0.2/redis.conf ./redis-master
```

修改master节点配置文件:

```
vim /opt/redis-master/redis.conf
bind 192.168.1.101
port 6680
```

192.168.1.102节点创建从redis:

```
mkdir redis-slave
cp /opt/redis-4.0.2/redis.conf ./redis-slave
```

修改slave节点配置文件:

```
vim redis-slave/redis.conf
bind 192.168.1.102
port 6680
slaveof 192.168.1.101 6680
```

192.168.1.101创建sentinel节点:

```
mkdir redis-sentinel
cp /opt/redis-4.0.2/sentinel.conf redis-sentinel
```

修改sentinel配置文件:

```
vim redis-sentinel/sentinel.conf
bind 192.168.1.101
port 26680
sentinel monitor mymaster 192.168.1.101 6680 1
```

启动redis主/从节点:

```
redis-server ./redis-master/redis.conf &
redis-server ./redis-slave/redis.conf &
```

启动sentinel:

```
redis-sentinel ./redis-sentinel/sentinel.conf &
```

使用sentinel的命令查看sentinel管理的redis master, 我们需要用redis-cli命令进入到sentinel:

```
redis-cli -h 192.168.1.101 -p 26680
sentinel get-master-addr-by-name mymaster
```

---

### 安装部署3台Zookeeper+Kafka集群

---

#### 2.1 安装部署环境

3台物理机器:

```
节点1 192.168.1.101 centos7.3  
节点2 192.168.1.102 centos7.3  
节点3 192.168.1.103 centos7.3
```

每个节点都安装java环境:

```
cd /opt/  
yum install java-1.8.0
```

每个节点都下载 kafka:

```
wget http://mirrors.hust.edu.cn/apache/kafka/1.0.0/kafka_2.12-1.0.0.tgz  
tar -xzf kafka_2.12-1.0.0.tgz
```

#### 2.2 部署Zookeeper服务

每个节点都创建zookeeper数据目录:

```
mkdir -p zookeeper/zkdata zookeeper/zkdatalog
```

优化JVM虚拟机参数:

```
vim kafka_2.12-1.0.0/bin/kafka-server-start.sh  
  
export KAFKA_HEAP_OPTS="-Xmx2G -Xms2G"
```

每个节点都修改zookeeper配置:

```
vim kafka_2.12-1.0.0/config/zookeeper.properties  
  
maxClientCnxns=0  
tickTime=2000  
initLimit=10  
syncLimit=5  
dataDir=/opt/zookeeper/zkdata  
dataLogDir=/opt/zookeeper/zkdatalog  
clientPort=2181  
server.1=192.168.1.101:2888:3888  
server.2=192.168.1.103:2888:3888  
server.3=192.168.1.103:2888:3888
```

192.168.1.101节点执行:

```
echo "1" > /opt/zookeeper/zkdata/myid
```

192.168.1.102节点执行:

```
echo "2" > /opt/zookeeper/zkdata/myid
```

192.168.1.103节点执行:

```
echo "3" > /opt/zookeeper/zkdata/myid
```

每个节点都启动Zookeeper服务器:

```
./kafka_2.12-1.0.0/bin/zookeeper-server-start.sh kafka_2.12-1.0.0/config/  
↪zookeeper.properties &
```

## 2.3 部署Kafka服务

每个节点都创建Kafka存储目录:

```
mkdir -p kafka/kafkalogs
```

每个节点都修改kafka配置文件:

```
vim kafka_2.12-1.0.0/config/server.properties  
  
192.168.1.101节点配置  
  
broker.id=1 #这里的数字和zookeeper配置的数字最好一致  
host.name=192.168.1.101
```



```
num.network.threads=9
num.io.threads=16
log.dirs=/opt/kafka/kafkalog/
message.max.byte=5242880
default.replication.factor=1
replica.fetch.max.bytes=5242880
zookeeper.connect=192.168.1.101:2181,192.168.1.102:2181,192.168.1.103:2181
```

192.168.1.102节点配置:

```
vim kafka_2.12-1.0.0/config/server.properties

broker.id=2 #这里的数字和zookeeper配置的数字最好一致
host.name=192.168.1.102
num.network.threads=9
num.io.threads=16
log.dirs=/opt/kafka/kafkalog/
message.max.byte=5242880
default.replication.factor=1
replica.fetch.max.bytes=5242880
zookeeper.connect=192.168.1.101:2181,192.168.1.102:2181,192.168.1.103:2181
```

192.168.1.103节点配置:

```
vim kafka_2.12-1.0.0/config/server.properties

broker.id=3 #这里的数字和zookeeper配置的数字最好一致
host.name=192.168.1.103
num.network.threads=9
num.io.threads=16
log.dirs=/opt/kafka/kafkalog/
message.max.byte=5242880
default.replication.factor=1
replica.fetch.max.bytes=5242880
zookeeper.connect=192.168.1.101:2181,192.168.1.102:2181,192.168.1.103:2181
```

每个节点都启动Kafka服务:

```
./kafka_2.12-1.0.0/bin/kafka-server-start.sh kafka_2.12-1.0.0/config/server.
↪properties &
```

## 2.4 测试Zookeeper+Kafka

测试集群是否成功:

```
./kafka_2.12-1.0.0/bin/kafka-topics.sh --zookeeper 192.168.1.101:2181 --
↪replication-factor 1 --partitions 1 --topic testTopic --create
./kafka_2.12-1.0.0/bin/kafka-console-consumer.sh --zookeeper 192.168.1.
↪102:2181 --topic testTopic --from-beginning
./kafka_2.12-1.0.0/bin/kafka-console-producer.sh --broker-list 192.168.1.
↪103:9092 --topic testTopic
```



---

### 安装部署3台EMQX集群

---

#### 3.1 安装部署环境

3台物理机器:

```
节点1 192.168.1.101 centos7.3
节点2 192.168.1.102 centos7.3
节点3 192.168.1.103 centos7.3
```

每个节点都安装emqx:

```
rpm -ivh --force emqx-changhong-centos7-v2.2-1.el7.centos.x86_64.rpm
```

192.168.1.101节点修改配置文件:

```
vim /etc/emqx/emqx.conf
node.name = emqx@192.168.1.101
```

192.168.1.102节点修改配置文件:

```
vim /etc/emqx/emqx.conf
node.name = emqx@192.168.1.102
```

192.168.1.103节点修改配置文件:

```
vim /etc/emqx/emqx.conf
node.name = emqx@192.168.1.103
```

每个节点都需要修改的配置文件:

```
vim /etc/emqx/plugins/emqx_changhong.conf

##redis sentinel服务器地址
changhong.redis.server = xxx.xxx.xxx.xxx:26680

## sentinel监听redis master的名字
changhong.redis.sentinel = mymaster

vim /etc/emqx/plugins/emqx_auth_jwt.conf
## JWT的密钥
auth.jwt.secret = xxxxxx

vim /etc/emqx/plugins/emqx_bridge_kafka.conf
bridge.kafka.pool1.server = xxx.xxx.xxx.xx1:9092 xxx.xxx.xxx.xx2:9092 xxx.xxx.
↪xxx.xx3:9092
```

192.168.1.101节点启动emqx服务:

```
service emqx start
```

192.168.1.102节点启动emqx服务并加入EMQX集群:

```
service emqx start
emqx_ctl cluster join emqx@192.168.1.101
```

192.168.1.103节点启动emqx服务并加入EMQX集群:

```
service emqx start
emqx_ctl cluster join emqx@192.168.1.101
```

查看emqx是否集群成功:

```
emqx_ctl cluster status
```

---

### 安装部署haproxy，实现3台emqx负载均衡

---

#### 4.1 安装部署环境

2台物理机器:

```
节点1 192.168.1.101 centos7.3
节点2 192.168.1.102 centos7.3
```

#### 4.2 安装HAProxy服务

编译安装HAProxy:

```
wget http://www.haproxy.org/download/1.7/src/haproxy-1.7.5.tar.gz
tar xf haproxy-1.7.5.tar.gz
cd haproxy-1.7.5
make TARGET=linux2628 USE_PCRE=1 USE_OPENSSL=1 USE_ZLIB=1 USE_CRYPT_H=1 USE_
↪LIBCRYPT=1 PREFIX=/usr/local/haproxy
make install PREFIX=/usr/local/haproxy
安装成功后，查看版本
/usr/local/haproxy/sbin/haproxy -v
```

#### 4.3 部署HAProxy服务

复制haproxy文件到/usr/sbin下:

```
cp /usr/local/haproxy/sbin/haproxy /usr/sbin/
```

复制haproxy脚本，到/etc/init.d下:

```
cp ./examples/haproxy.init /etc/init.d/haproxy
chmod 755 /etc/init.d/haproxy
```

创建系统账号:

```
useradd -r haproxy
```

创建配置文件:

```
mkdir /etc/haproxy
```

修改配置文件:

```
vi /etc/haproxy/haproxy.cfg

global
    log 127.0.0.1 local3 info
    chroot /usr/local/haproxy
    user haproxy
    group haproxy
    daemon
    maxconn 1024000

defaults
    log global
    mode tcp
    option httplog
    option dontlognull
    timeout connect 5000
    timeout client 50000
    timeout server 50000

frontend emqx_tcp
    bind *:1883
    option tcplog
    mode tcp
    default_backend emqx_tcp_back

frontend emqx_ws
    bind *:8083
    option tcplog
    mode tcp
    default_backend emqx_ws_back

frontend emqx_dashboard
    bind *:18083
    option tcplog
    mode tcp
    default_backend emqx_dashboard_back
```

```

frontend emqx_api
  bind *:8080
  option tcplog
  mode tcp
  default_backend emqx_api_back

frontend emqx_ssl
  bind *:8883 ssl crt /etc/emqx/certs/emqx.pem no-ssl3
  option tcplog
  mode tcp
  default_backend emqx_ssl_back

frontend emqx_wss
  bind *:8084 ssl crt /etc/emqx/certs/emqx.pem no-ssl3
  option tcplog
  mode tcp
  default_backend emqx_wss_back

backend emqx_tcp_back
  balance roundrobin
  server emqx_node_1 192.168.1.101:1883 check
  server emqx_node_2 192.168.1.102:1883 check
  server emqx_node_3 192.168.1.103:1883 check

backend emqx_ws_back
  balance roundrobin
  server emqx_node_1 192.168.1.101:8083 check
  server emqx_node_2 192.168.1.102:8083 check
  server emqx_node_3 192.168.1.103:8083 check

backend emqx_dashboard_back
  balance roundrobin
  server emqx_node_1 192.168.1.101:18083 check
  server emqx_node_2 192.168.1.102:18083 check
  server emqx_node_3 192.168.1.103:18083 check

backend emqx_api_back
  balance roundrobin
  server emqx_node_1 192.168.1.101:8080 check
  server emqx_node_2 192.168.1.102:8080 check
  server emqx_node_3 192.168.1.103:8080 check

backend emqx_ssl_back
  mode tcp
  balance source
  timeout server 50s
  timeout check 5000
  server emqx_node_1 192.168.1.101:1883 check inter 10000 fall 2 rise 5
↪weight 1
  server emqx_node_2 192.168.1.101:1883 check inter 10000 fall 2 rise 5
↪weight 1
  server emqx_node_3 192.168.1.101:1883 check inter 10000 fall 2 rise 5
↪weight 1

```

```
source 0.0.0.0 usesrc clientip

backend emqx_wss_back
    mode tcp
    balance source
    timeout server 50s
    timeout check 5000
    server emqx_node_1 192.168.1.101:8083 check inter 10000 fall 2 rise 5_
↪weight 1
    server emqx_node_2 192.168.1.101:8083 check inter 10000 fall 2 rise 5_
↪weight 1
    server emqx_node_3 192.168.1.101:8083 check inter 10000 fall 2 rise 5_
↪weight 1
    source 0.0.0.0 usesrc clientip
```

启动haproxy:

```
service haproxy start
```

支持与联系:

官网:	<a href="http://emqtt.com">http://emqtt.com</a>
电话:	400-696-5502
邮箱:	<a href="mailto:contact@emqtt.io">contact@emqtt.io</a>
微信:	emqtttd
微博:	<a href="http://weibo.com/emqtt">http://weibo.com/emqtt</a>
Twitter:	@emqtt
公司:	杭州映云科技有限公司