

icom - 服务端基础环境部署文档

1、环境要求

名称	系统配置	环境	部署内容	中间件
流媒体服务器	系统：CentOS7.8 网络带宽：50M、预付费带宽（100M） 【硬性要求】 配置：2核8g	gcc、cmake	media-server（媒体服务）	无
应用服务器	系统：CentOS7.8 网络带宽：10M 配置：4核32G 100G（ssd盘，可动态）	jdk1.8、docker以及docker-compose（版本19以上）	icomAsst（辅助服务） icomFile（云存储） icomManage（管理服务） web静态页面（nginx）	MongoDB rocketMQ Minio MySQL

生产环境中，严禁用docker安装部署数据库中间件服务

2、环境安装

2.1流媒体服务安装

运行要求：

gcc 环境 \geq 4.8

端口放行：

1985、8080、8000（udp）、10080（udp）

安装部署：

- 1、解压编译好的压缩包 (media-CentOS7-x86_64-v0.0.1.zip)
- 2、授权文件权限

```
chmod -R 755 media-CentOS7-x86_64-v0.0.1
cd media-CentOS7-x86_64-v0.0.1
```

- 3、执行INSTALL脚本命令

```
./INSTALL
```

安装成功如下图所示：

```
[root@izbp1j8gc244si87pj3c14Z media-CentOS7-x86_64-v0.0.1]# ./INSTALL
argv[0]=./INSTALL
execute scripts in file: ./INSTALL
create log( /root/media-CentOS7-x86_64-v0.0.1/logs/package.1659421015.log ) success [ OK ]
see detail log: tailf /root/media-CentOS7-x86_64-v0.0.1/logs/package.1659421015.log [ OK ]
check tools [ OK ]
check previous install [ OK ]
previous install checked [ OK ]
backup old srs [ OK ]
backup installed dir, version=4.0.253 [ OK ]
to=/usr/local/srs.2022-08-02_14-16-55.v-4.0.253 [ OK ]
backup installed dir success [ OK ]
old srs backedup [ OK ]
prepare files [ OK ]
prepare files success [ OK ]
copy core components [ OK ]
copy core components success [ OK ]
install init.d scripts [ OK ]
install init.d scripts success [ OK ]
install srs.service for systemctl [ OK ]
install srs.service for systemctl success [ OK ]
install system service for CentOS [ OK ]
install system service success [ OK ]

hlep to renyanru@sina.com
install success
CentOS7:
    sudo systemctl start srs
srs root is /usr/local/srs
[root@izbp1j8gc244si87pj3c14Z media-CentOS7-x86_64-v0.0.1]#
```

4. 配置打洞服务ip

```
cd /usr/local/srs/conf
vim srs.conf
```

配置文件说明

```
# config for media.
listen                1935;
max_connections       4000;
#日志文件
srs_log_tank          file;
srs_log_file          ./objs/srs.log;
#开启后台守护
daemon                on;
#api接口端口
http_api {
    enabled            on;
    listen              1985;
}
```

```

http_server {
    enabled          on;
    #管理页面端口，后期会统一迁移，目前开启
    listen          8080;
    dir              ./objs/nginx/html;
}
rtc_server {
    enabled on;
    #流媒体协议流端口
    listen 8000; # UDP port
    #填写打洞服务ip 注意：此处只能填写ip不能填写域名
    candidate 101.43.120.102;
}
vhost icom {
#以下注释地方为接口回调，暂时不用开启
#    http_hooks {
#        enabled          on;
#        on_connect       http://172.28.0.102:8085/connect;
#        on_close         http://172.28.0.102:8085/close;
#        on_publish        http://172.28.0.102:8085/publish;
#        on_unpublish      http://172.28.0.102:8085/unpublished;
#        on_play           http://172.28.0.102:8085/play;
#        on_stop           http://172.28.0.102:8085/stop;
#        on_dvr            http://172.28.0.102:8085/api/v1/dvrs
http://localhost:8085/api/v1/dvrs;
#        on_hls            http://172.28.0.102:8085/api/v1/hls
http://localhost:8085/api/v1/hls;
#        on_hls_notify
http://172.28.0.102:8085/api/v1/hls/[server_id]/[app]/[stream]/[ts_u
rl][param];
#    }
    hls {
        enabled          on;
    }
    #关闭flv视频流播放协议
    http_remux {
        enabled          off;
        mount            [vhost]/[app]/[stream].flv;
    }
    #rtc协议配置
    rtc {
        #开启rtc协议
        enabled          on;
        #开启rtmp协议转rtc协议
        rtmp_to_rtc on;
        #开启rtc协议转rtmp协议
        rtc_to_rtmp on;
    }
}
}

```

5. 命令

```
启动服务
systemctl start srs
查看状态
systemctl status srs
停止服务
systemctl stop srs
```

2.2 MongoDB安装部署

环境要求：

MongoDB>=5.0

端口放行：

27017

1、安装

```
rpm -ivh mongodb-org-shell-5.0.10-1.el7.x86_64.rpm
```

systemctl start mongod.service并设置为开机自启动systemctl enable mongod.service MongoDB数据库默认的监听端口27017

2、创建账户（管理和普通账户）

输入mongo登录数据库进入shell命令行

```
#进入mongo
mongo
#使用管理员数据库
use admin
#创建管理员用户
db.createUser({user:"xxxx", pwd:"xxxx", roles:
[{"role:"readWriteAnyDatabase", db:"admin"}]});
exit;
#创建普通用户数据库
#以刚建立的管理用户登录数据库 创建icom用户
#登录mongo
mongo --port 27017 -u admin -p 123456 --authenticationDatabase admin
#使用icomdb库
use icom
#是创建账户
db.createUser({user: "icom",pwd: "xxxx",roles: [{ role: "root", db:
"icomdb" }]});
exit;
#验证
```

```
#以刚创建的icom用户登录
mongo -u icom -p xxxx --authenticationDatabase icomdb
```

2.3 rocketMq安装 (docker-compose部署)

环境要求 :

rocketMq版本=4.9.4 (只能是这个版本)

端口放行 :

broker : 10909、10911-10912、9876

console : 19867

1、创建挂载目录

```
mkdir -p /docker_data/rocketmq/namesrv/logs
mkdir -p /docker_data/rocketmq/namesrv/store
mkdir -p /docker_data/rocketmq/broker1/conf
mkdir -p /docker_data/rocketmq/broker1/logs
mkdir -p /docker_data/rocketmq/broker1/store
```

2、创建配置文件

```
cd /docker_data/rocketmq/broker1/conf
touch broker.conf
vi broker.conf
```

```
# 所属集群名称，如果节点较多可以配置多个
brokerClusterName = DefaultCluster
#broker名称，master和slave使用相同的名称，表明他们的主从关系
brokerName = broker1
#0表示Master，大于0表示不同的slave
brokerId = 0
#表示几点做消息删除动作，默认是凌晨4点
deleteWhen = 04
#在磁盘上保留消息的时长，单位是小时
fileReservedTime = 48
#有三个值：SYNC_MASTER，ASYNC_MASTER，SLAVE；同步和异步表示Master和Slave之间同步数据的机制；
brokerRole = ASYNC_MASTER
#刷盘策略，取值为：ASYNC_FLUSH，SYNC_FLUSH表示同步刷盘和异步刷盘；
SYNC_FLUSH消息写入磁盘后才返回成功状态，ASYNC_FLUSH不需要；
flushDiskType = ASYNC_FLUSH
# 设置broker节点所在服务器的ip地址 (**这个非常重要，主从模式下，从节点会根据主节点的brokerIP2来同步数据，如果不配置，主从无法同步，brokerIP1设置为自己外网能访问的ip，服务器双网卡情况下必须配置，比如阿里云这种，主节点需要配置ip1和ip2，从节点只需要配置ip1即可)
```

```
# 此ip由使用环境决定 本机使用 127 局域网使用 192 外网使用 外网ip
brokerIP1 = 192.168.8.102
#nameServer地址,分号分割
namesrvAddr = 192.168.8.102:9876
#Broker 对外服务的监听端口,
listenPort = 10911
#是否允许Broker自动创建Topic
autoCreateTopicEnable = false
#是否允许 Broker 自动创建订阅组
autoCreateSubscriptionGroup = false
#linux开启epoll
useEpollNativeSelector = true
```

3、分配权限(最重要的一步)

为所有目录包括子目录分配读写权限 没有写权限无法存储 会报错无法启动

```
chmod -R 777 /docker_data/rocketmq
```

4、docker-compose 编排

```
version: '3'
services:
  mqnamesrv:
    image: apache/rocketmq:4.9.4
    container_name: mqnamesrv
    restart: always
    ports:
      - 9876:9876
    environment:
      JAVA_OPT: -server -Xms512m -Xmx512m
    command: ["sh", "mqnamesrv"]
    volumes:
      - ./namesrv/logs:/home/rocketmq/logs
      - ./namesrv/store:/home/rocketmq/store
    networks:
      rocketmq_net:
        aliases:
          - rocketmq-namesrv

  mqbroker1:
    image: apache/rocketmq:4.9.4
    container_name: mqbroker1
    restart: always
    ports:
      - 10911:10911
      - 10909:10909
      - 10912:10912
    environment:
```

```

    JAVA_OPT_EXT: "-server -Xms2048M -Xmx2048M -Xmn512M"
    command: sh mqbroker -c /home/rocketmq/rocketmq-
4.9.4/conf/broker.conf
    depends_on:
      - mqnamesrv
    volumes:
      - ./broker1/conf/broker.conf:/home/rocketmq/rocketmq-
4.9.4/conf/broker.conf
      - ./broker1/logs:/home/rocketmq/logs/rocketmqlogs
      - ./broker1/store:/home/rocketmq/store
    networks:
      rocketmq_net:
        aliases:
          - rocketmq-broker

mqconsole:
  image: styletang/rocketmq-console-ng
  container_name: mqconsole
  restart: always
  ports:
    - 19876:19876
  environment:
    JAVA_OPTS: "-Dserver.port=19876 -
Drocketmq.namesrv.addr=rocketmq-namesrv:9876 -
Dcom.rocketmq.sendMessageWithVIPChannel=false"
  depends_on:
    - mqnamesrv
  networks:
    rocketmq_net:
      aliases:
        - rocketmq-console

networks:
  rocketmq_net:
    name: rocketmq_net
    driver: bridge

```

2.4 minio安装部署

之前部署过，这里不做详细说明 docker安装即可，需要注意的是做好配置文件、存储卷以及日志目录的挂载

端口放行：

9000、50000

2.5 java应用部署

环境要求：

jdk : 1.8

java应用后面根据脚本进行启动和部署。目前暂时不参与文档编写