

kmq配置目录

首先，以在本机部署kmq node1节点为例，讲解如何准备node1节点所需要的配置目录，默认node1节点的配置目录为/home/w/conf/kmq/

一、配置目录

配置目录是kmq配置文件很重要的一个概念，每个kmq服务在启动的时候必须指定一个配置目录，这个配置目录包含了整个kmq运行所需要的信息，包括如何启动

```
/home/w/bin/kmqd -c /home/w/conf/kmq/node1 -s start
```

配置目录下包含的文件有：

```
appsconf
log4cpp.conf
monitor.conf
kmq.conf
```

二、log4cpp.conf配置文件

在配置log4cpp.conf之前，首先需要为kmq节点指定一个单独的日志保存目录，和配置目录一样，所有kmq节点的默认日志目录推荐放到/home/w/logs/kmq/下，例如目录：

```
/home/w/logs/kmq/node1/
```

log4cpp.conf内容如下：

```
# --- categories ---
log4cpp.rootCategory=NOTICE, main
log4cpp.category.kmq=NOTICE, kmq
log4cpp.category.monitor=WARN, monitor
# --- root Appender ---
log4cpp.appender.main=org.apache.log4cpp.RollingFileAppender
log4cpp.appender.main.fileName=/home/w/logs/kmq/node1/root.log
log4cpp.appender.main.maxFileAge=86400
log4cpp.appender.main.layout.ConversionPattern=%d{%Y-%m-%d %H:%M:%S} [%p]: %m%n
log4cpp.appender.main.layout=org.apache.log4cpp.PatternLayout
# --- App ---
log4cpp.appender.monitor=org.apache.log4cpp.RollingFileAppender
log4cpp.appender.monitor.fileName=/home/w/logs/kmq/node1/monitor.log
log4cpp.appender.monitor.maxFileAge=86400
log4cpp.appender.monitor.maxBackupIndex=5
log4cpp.appender.monitor.maxFileSize=1024000000
log4cpp.appender.monitor.backupPattern=%Y-%m-%d
log4cpp.appender.monitor.layout.ConversionPattern=%d{%Y-%m-%d %H:%M:%S} [%p]: %m%n
log4cpp.appender.monitor.layout=org.apache.log4cpp.PatternLayout
# --- App ---
log4cpp.appender.kmq=org.apache.log4cpp.RollingFileAppender
log4cpp.appender.kmq.fileName=/home/w/logs/kmq/node1/kmq.log
log4cpp.appender.kmq.maxFileAge=86400
log4cpp.appender.kmq.maxBackupIndex=5
log4cpp.appender.kmq.maxFileSize=1024000000
log4cpp.appender.kmq.backupPattern=%Y-%m-%d
log4cpp.appender.kmq.layout.ConversionPattern=%d{%Y-%m-%d %H:%M:%S} [%p]: %m%n
```

```
log4cpp.appender.kmq.layout=org.apache.log4cpp.PatternLayout
```

三、kmq.conf配置文件

配置内容如下，主要包括两部分，一个分是regmgr的配置，一部分是kmq全局的配置

```
[regmgr]
epoll_timeout_msec=10
regmgr_listen_addrs=:1510
register_interval_msec=1000
[kmq]
log4conf=log4cpp.conf
monitor=monitor.conf
apps_configdir=appsconf/
appconfupdate_interval_sec=1
```

对于regmgr部分：

1. epoll_timeout_msec=10 epoll_wait超时时间
2. regmgr_listen_addrs=:1510 regmgr监听的端口
3. register_interval_msec=1000
4. regmgr对于需要自动重连的连接（kmq与kmq节点之间的连接注册工作，是由regmgr来代理完成的），regmgr会在连接注册失败的情况，定期重连。

对于kmq部分

1. log4conf=log4cpp.conf log4cpp配置文件
2. monitor=monitor.conf kmq接入监控中心monitor的配置文件
3. apps_configdir=appsconf/ kmq存放appgroup context配置文件的目录，kmq可以同时服务于多个appgroup
4. context之间的通信，每个appgroup context在apps_configdir目录下都有一个单独的配置文件
5. appconfupdate_interval_sec=1 kmq定期扫描更新所有appgroup_context配置文件目录的时间间隔（单位为秒s）

五、apps_configdir目录

apps_configdir包含了所有appgroup context的配置文件，每个appgroup context对应一个后缀为.conf的配置文件，kmq主线程会定期（appconfupdate_interval_sec参数）扫描这个目录：

1. 如果配置文件对应的appgroup已经启动，则执行配置更新操作
2. 如果配置文件对应的appgroup还没启动，则执行启动操作

1) appgroup配置解释

apps_configdir目录默认包含一个appname=testapp的配置文件testapp.conf，内容如下：

```
{
"app_id": "kmq",
"monitor_server":[
{
"ip": "127.0.0.1",
"port": 21000
}
],
"send_interval_secs": 1,
"eth": "eth0"
```

```

}

[app]
appid=testapp
msg_balance_factor=50
msg_queue_size=1024
msg_timeout_msec=100
msg_io_events=500
epoll_timeout_msec=1
monitor_record_stats_msec=1000
reconnect_timeout_msec=10000
connect_to_apps=127.0.0.1:1520

```

appgroup contex的配置文件主要包括app相关的配置

1. appid=testapp // appgroup的名称
2. msg_balance_factor=50 //
3. 负载均衡因子，表示当角色相应的底层socket有读写事件（EPOLLIN|EPOLLOUT）时，每次只读取或者只写多少个消息
4. msg_queue_size=1024 // 队列大小，超过此大小，数据包会被丢弃，并返回一个deliver error类别的控制报文，将何地，何原因通知消息的producer
5. msg_timeout_msec=100 //消息的声明周期，不能超过此值，单位为毫秒ms，在api层或者任意一个kmq节点，都会检查数据包的声明周期是否结束，如果结束了，丢弃数据包，并error控制报文给producer
6. msg_io_events=500 // epoll_wait 的最大描述符数量参数
7. epoll_timeout_msec=1 // epoll_wait 超时的时间
8. monitor_record_stats_msec=1000 //将kmq系统信息更新到monitor-center的时间间隔
reconnect_timeout_msec=10000 //kmq检测到连接异常后，相应的角色（receiver或者dispatcher）会进入time_wait状态，并等待对端重连，等待时间为此值，如果超过此时间，没有连
connect_to_apps=127.0.0.1:1520 // kmq交叉互联的情况下，appgroup context需要连接到下一个kmq节点的ip地址

2) 动态添加appgroup context配置文件

如上所述，添加appgroup配置文件并不需要重启kmq，kmq会自动监控apps_configdir目录并实时启动新增的appgroup appgroup配置文件添加完毕之后，需要检查当前kmq节点的日志，观察新增的appgroup是否成功启动，例如：

在kmq_server上应该看到类似的如下日志：

```
2014-09-24 15:43:10 [NOTICE]: tid:4661 appctx.cc:120 Start(), "app front-ms start ok with pid 4661"
```

其中