EMQTT 分布物联网消息队列

2019年03月

**修改记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修订日期 | 修订人员 | 修订纪要 |
| V1.0 | 2019-03-25 | 徐强辉 | 初始版本 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**文档目录**

[1. 一级Title1](#_Toc17809)

[1.1. 二级Title1](#_Toc1549)

[1.2. 二级Title2](#_Toc25507)

[1.2.1. 三级Title1](#_Toc27634)

[1.2.2. 三级Title2](#_Toc13629)

[1.2.2.1. 四级Title](#_Toc8803)

[2. 一级Title2](#_Toc28876)

[2.1. 二级Title](#_Toc12598)

[2.2. 二级Title](#_Toc25982)

[2.2.1. 三级Title](#_Toc16068)

[2.2.2. 三级Title](#_Toc3207)

[2.2.2.1. 四级Title](#_Toc8877)

[2.2.2.2. 四级Title](#_Toc32092)

# EMQ2.0 简介

EMQ (Erlang/Enterprise/Elastic MQTT Broker) 是基于 Erlang/OTP 平台开发的开源物联网 MQTT 消息服务器。Erlang/OTP 是出色的软实时(Soft-Realtime)、低延时(Low-Latency)、分布式(Distributed) 的语言平台。MQTT 是轻量的(Lightweight)、发布订阅模式(PubSub) 的物联网消息协议。

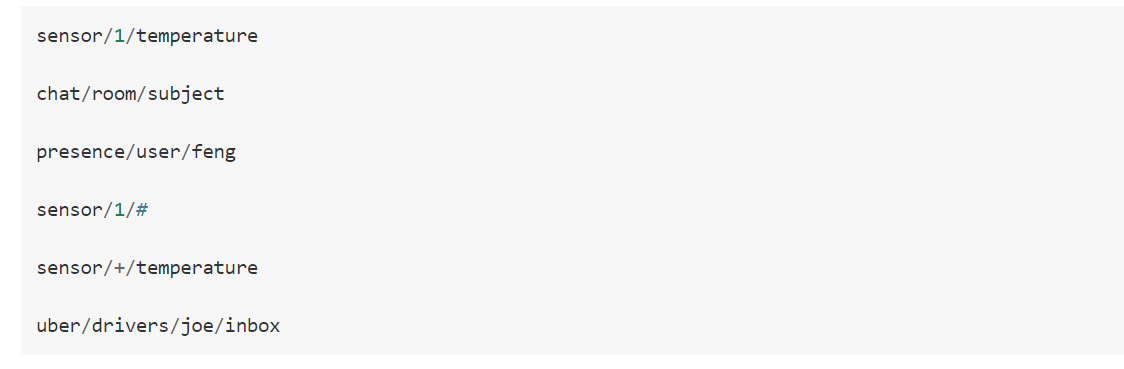
*EMQ* 项目设计目标是承载移动终端或物联网终端海量 MQTT 连接，并实现在海量物联网设备间快速低延时消息路由:

1. 稳定承载大规模的 MQTT 客户端连接，单服务器节点支持50万到100万连接。
2. 分布式节点集群，快速低延时的消息路由，单集群支持1000万规模的路由。
3. 消息服务器内扩展，支持定制多种认证方式、高效存储消息到后端数据库。
4. 完整物联网协议支持，MQTT、MQTT-SN、CoAP、WebSocket 或私有协议支持。

# MQTT 发布订阅模式

MQTT 是发布订阅(Publish/Subscribe) 模式的消息协议，与 HTTP 协议请求响应(Request/Response) 模式不同。

MQTT 发布者与订阅者之间通过”主题”(Topic) 进行消息路由，主题(Topic) 格式类似 Unix 文件路径，例如:



MQTT 主题(Topic) 支持’+’, ‘#’的通配符，’+’通配一个层级，’#’通配多个层级(必须在末尾)。

MQTT 消息发布者(Publisher) 只能向特定’名称主题’(不支持通配符)发布消息，订阅者(Subscriber)通过订阅’过滤主题’(支持通配符)来匹配消息。

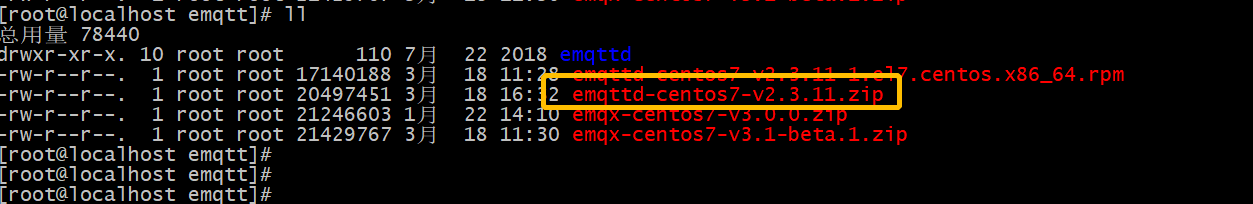
# MQTT安装部署

我这里是选择 安装包安装, 还可以 docker, rpm 等安装方式.

## 安装启动

### 下载版本 v2.3.11

下载地址: <http://www.emqtt.com/downloads>



### 虚拟机准备

|  |  |
| --- | --- |
| emq虚拟机 | 192.168.188.131, 192.168.188.132 |
| haproxy虚拟机 | 192.168.188.133 |

### 解压安装包

131, 132两台虚拟机分别解压安装包到对应安装目录.

unzip emqttd-centos7-v2.3.11.zip

### 配置参数

#### Erlang虚拟机参数

##vim emqttd/etc/emq.conf

#节点名称

node.name = emq@192.168.188.131

#Erlang 虚拟机允许的最大进程数，EMQ 一个连接会消耗2个Erlang进程,

#参数值 > 最大允许连接数 \* 2

node.process\_limit = 2097152

#Erlang 虚拟机允许的最大 Port 数量，EMQ 一个连接消耗1个 Port.

#Erlang 的 Port 非 TCP 端口，可以理解为文件句柄。

#参数值 > 最大允许连接数

node.max\_ports = 1048576

#### EMQ消息服务器参数

##vim emqttd/etc/emq.conf

listener.tcp.external = 0.0.0.0:1883

#设置 TCP 监听器的 Acceptor 池大小

listener.tcp.external.acceptors = 64

#emq最大允许连接数

listener.tcp.external.max\_clients = 1000000

#### linux操作系统配置

1. vim /etc/sysctl.conf

#系统全局允许分配的最大文件句柄数

fs.file-max=2097152

fs.nr\_open=2097152

#并发连接 backlog 设置

net.core.somaxconn=32768

net.ipv4.tcp\_max\_syn\_backlog=16384

net.core.netdev\_max\_backlog=16384

#可用知名端口范围

net.ipv4.ip\_local\_port\_range='1000 65535'

#TCP Socket 读写 Buffer 设置

net.core.rmem\_default=262144

net.core.wmem\_default=262144

net.core.rmem\_max=16777216

net.core.wmem\_max=16777216

net.core.optmem\_max=16777216

net.ipv4.tcp\_rmem='1024 4096 16777216'

net.ipv4.tcp\_wmem='1024 4096 16777216'

#TCP 连接追踪设置

net.nf\_conntrack\_max=1000000

net.netfilter.nf\_conntrack\_max=1000000

net.netfilter.nf\_conntrack\_tcp\_timeout\_time\_wait=30

#TIME-WAIT Socket 最大数量、回收与重用设置

net.ipv4.tcp\_max\_tw\_buckets=1048576

# 注意: 不建议开启該设置，NAT模式下可能引起连接RST

# net.ipv4.tcp\_tw\_recycle = 1

# net.ipv4.tcp\_tw\_reuse = 1

#FIN-WAIT-2 Socket 超时设置

net.ipv4.tcp\_fin\_timeout = 15

2. vim /etc/systemd/system.conf

DefaultLimitNOFILE=1048576

3. vim /etc/security/limits.conf

#持久化设置允许用户/进程打开文件句柄数

\* soft nofile 1048576

\* hard nofile 1048576

#### 其他配置

echo 2097152 > /proc/sys/fs/nr\_open

#允许当前会话/进程打开文件句柄数

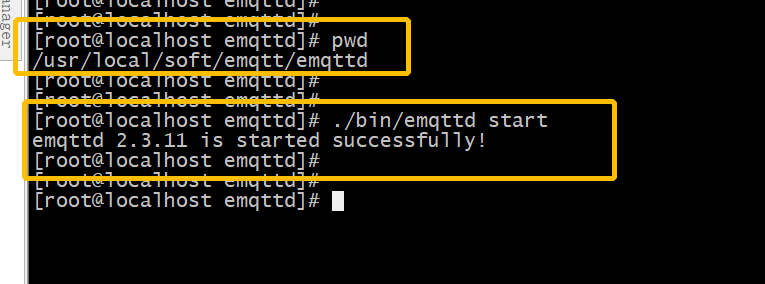
ulimit -n 1048576

### 启动emq集群

131,132两台虚拟机修改好配置

#### 启动单节点

./bin/emqttd start



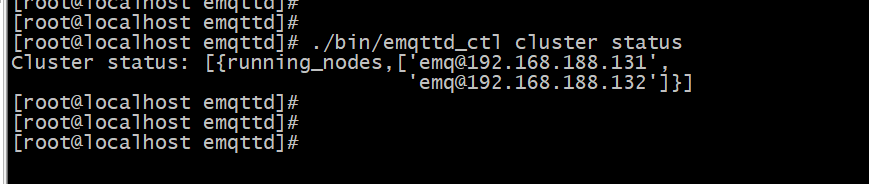
#### 加入emq集群

在132节点 执行. (也可以在131节点执行)

./bin/emqttd\_ctl cluster join [emq@192.168.188.131](mailto:emq@192.168.188.131)

#### 查看集群状态

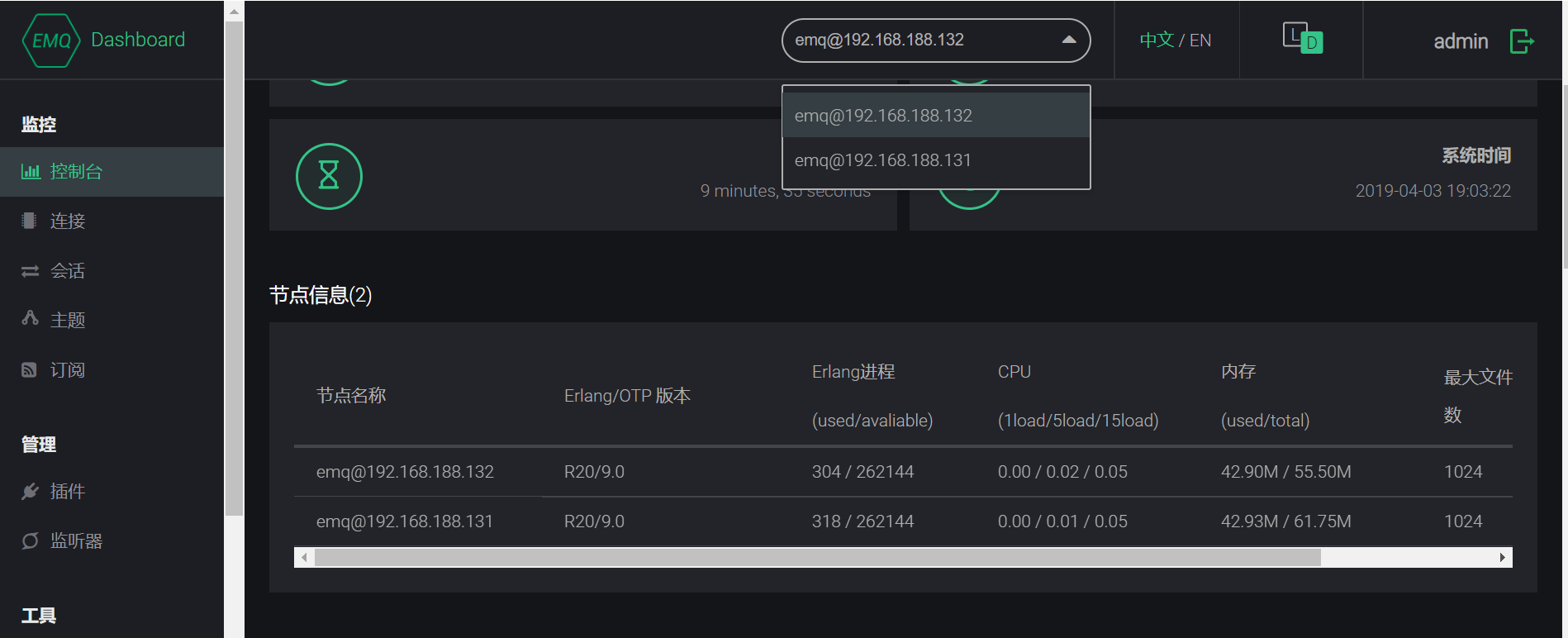
./bin/emqttd\_ctl cluster status



#### web控制台

默认web控制台端口为18083 账号 admin /public

<http://192.168.188.131:18083/#/>



# 安装Haproxy

根据官网私有云部署架构推荐, 使用haproxy 做 负载均衡.

在192.168.188.133安装好 haproxy 并做好对应配置.

监听tcp端口为 1883

# 压测工具

## 依赖安装

### erlang依赖

yum -y install make gcc gcc-c++ kernel-devel m4 ncurses-devel openssl-devel unixODBC-devel

### 安装erlang

#### 下载解压

wget [http://erlang.org/download/](http://erlang.org/download/otp_src_21.0.tar.gz)otp\_src\_20.0.tar.gz

tar -xvzf otp\_src\_21.0.tar.gz

#### 配置

进入 解压缩目录

./configure --prefix=/usr/local/erlang --with-ssl -enable-threads -enable-smmp-support -enable-kernel-poll --enable-hipe --without-javac

#### 安装

make && make install

#### 配置profile

vim /etc/profile

export ERLANG\_HOME=/usr/local/erlang

export PATH=$PATH:$JAVA\_HOME/bin:$ERLANG\_HOME/bin

使配置生效

source /etc/profile

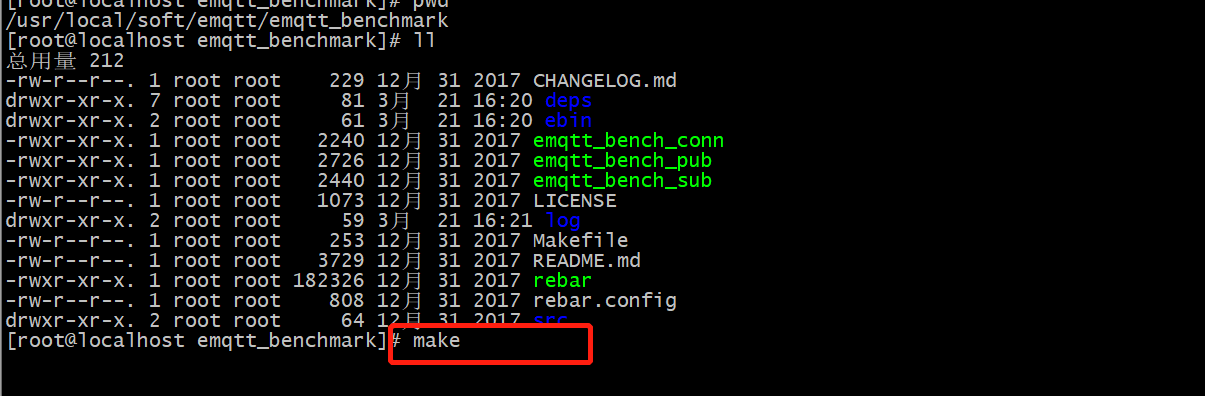
## 安装测试工具

### 从github上下载源码

<https://github.com/emqtt/emqtt_benchmark/tree/benchmark>

解压到对应目录 进入对应目录

执行 make (这里对erlang版本有要求, 20.0版本的才能用)



make执行完之后

### 参数说明

#### clean session

true: 表示服务端会创建一个持久会话, 在客户端断开连接时, 会话仍然会保持并保存离线消息. 直到会话超时注销.

false: 创建一个新的临时会话, 客户端断开是, 会话自动销毁.

#### retained

true发送的消息会保存在服务端, 后续的订阅者会接收到该消息.

#### Qos

0: 最多一次

1: 最少一次

2: 只有一次

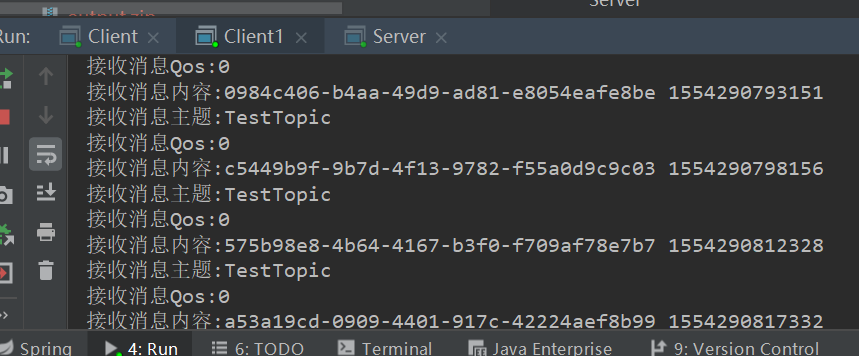
发布和订阅都需要设置qos, 发布client发送消息到 broker, broker 发送消息到对应订阅者.

示例:

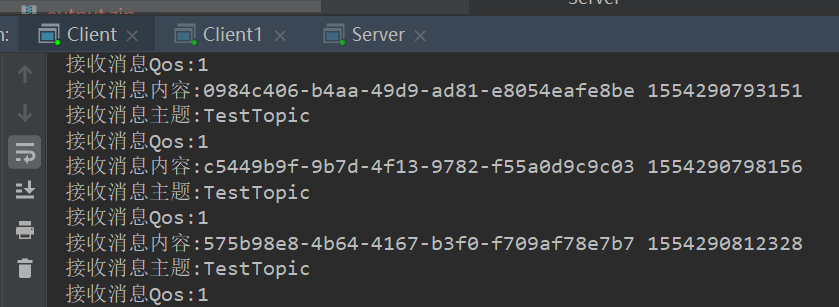
发布者qos设置为2



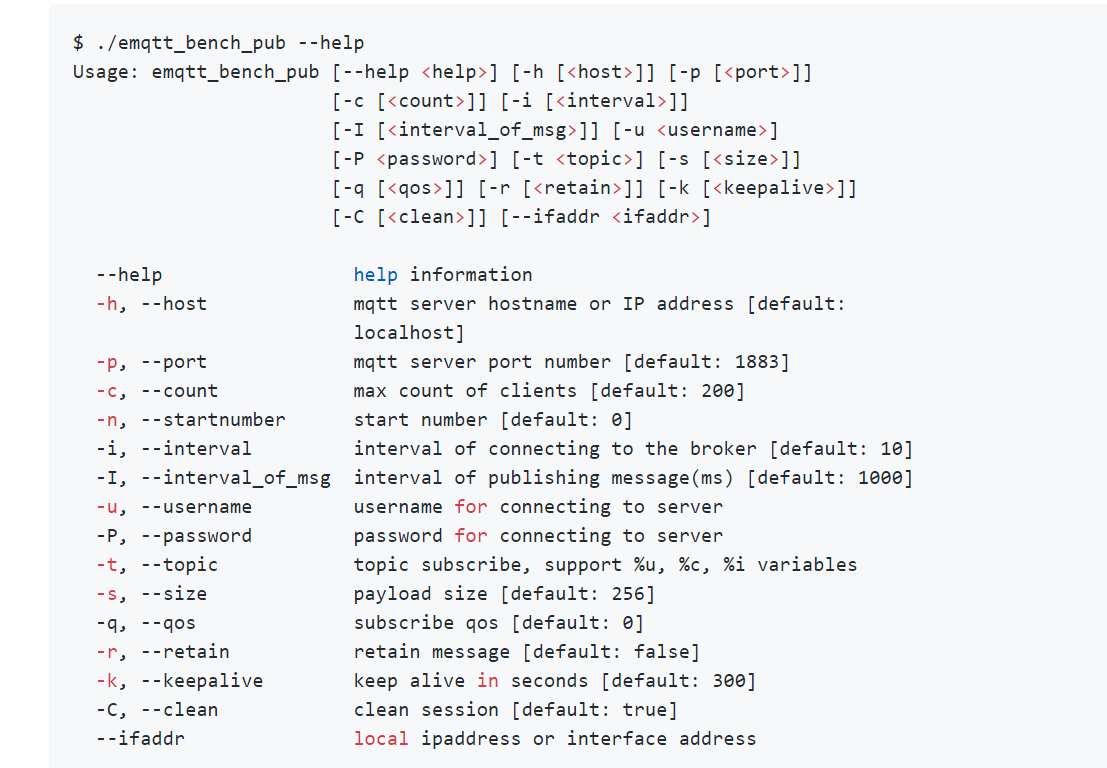
订阅者0 qos设置为0



订阅者1 qos设置为 1



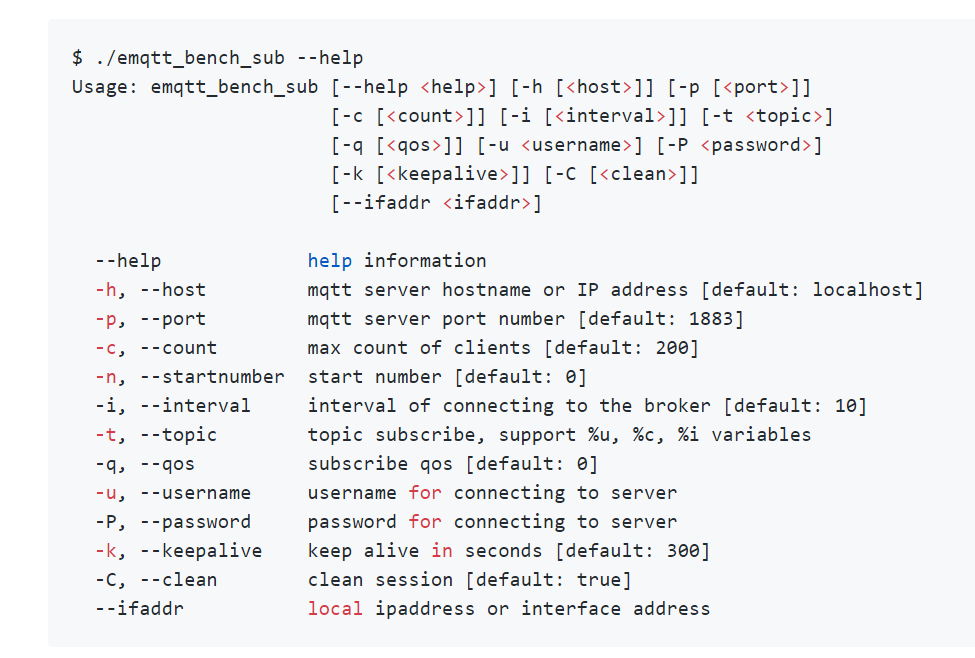
### 发布测试指令

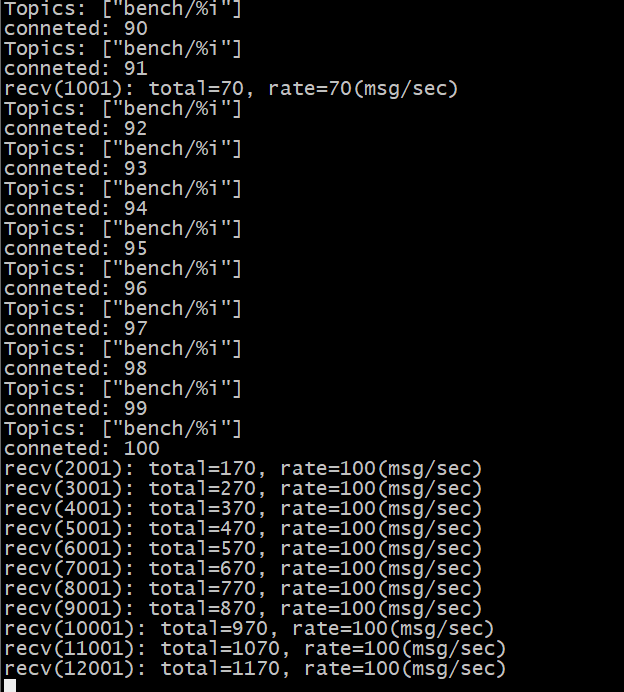
./emqtt\_bench\_pub -h 192.168.188.133 -c 100 -i 10 -t bench/%i



### 订阅测试指令

./emqtt\_bench\_sub –h 192.168.188.133 –c 100 –i 10 –t bench/%i





### 管理控制后台

