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 3](#_Toc17809)

[1.1. 二级Title1 3](#_Toc1549)

[1.2. 二级Title2 3](#_Toc25507)

[1.2.1. 三级Title1 3](#_Toc27634)

[1.2.2. 三级Title2 3](#_Toc13629)

[1.2.2.1. 四级Title 4](#_Toc8803)

[2. 一级Title2 4](#_Toc28876)

[2.1. 二级Title 4](#_Toc12598)

[2.2. 二级Title 4](#_Toc25982)

[2.2.1. 三级Title 5](#_Toc16068)

[2.2.2. 三级Title 5](#_Toc3207)

[2.2.2.1. 四级Title 5](#_Toc8877)

[2.2.2.2. 四级Title 5](#_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。

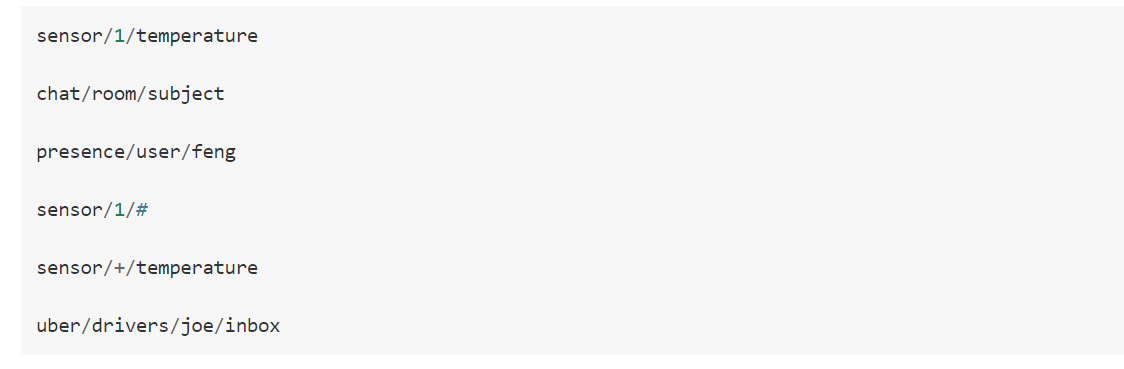
*EMQ* 项目设计目标是承载移动终端或物联网终端海量 MQTT 连接，并实现在海量物联网设备间快速低延时消息路由:

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。

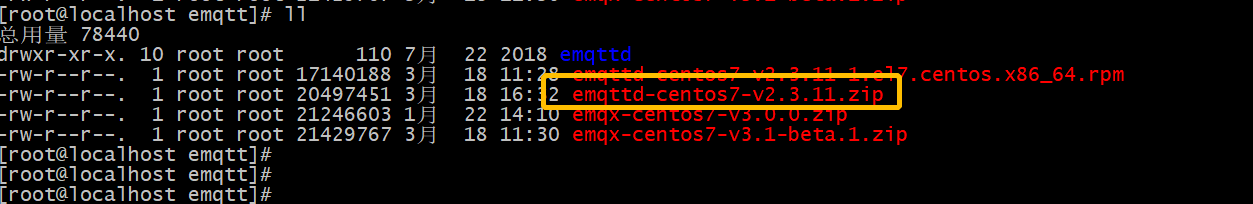
# MQTT安装部署

## 下载地址

<http://www.emqtt.com/downloads>

## 安装启动

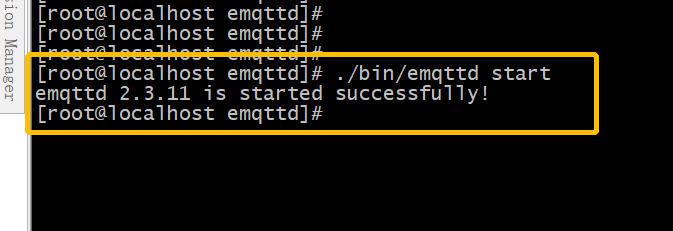
下载版本 v2.3.11



unzip emqttd-centos7-v2.3.11.zip

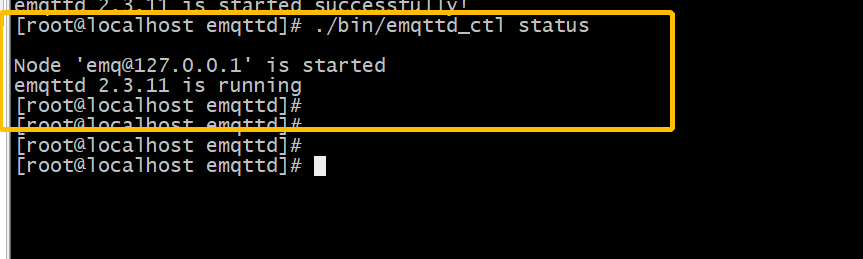
启动

./bin/emqttd start



查看状态

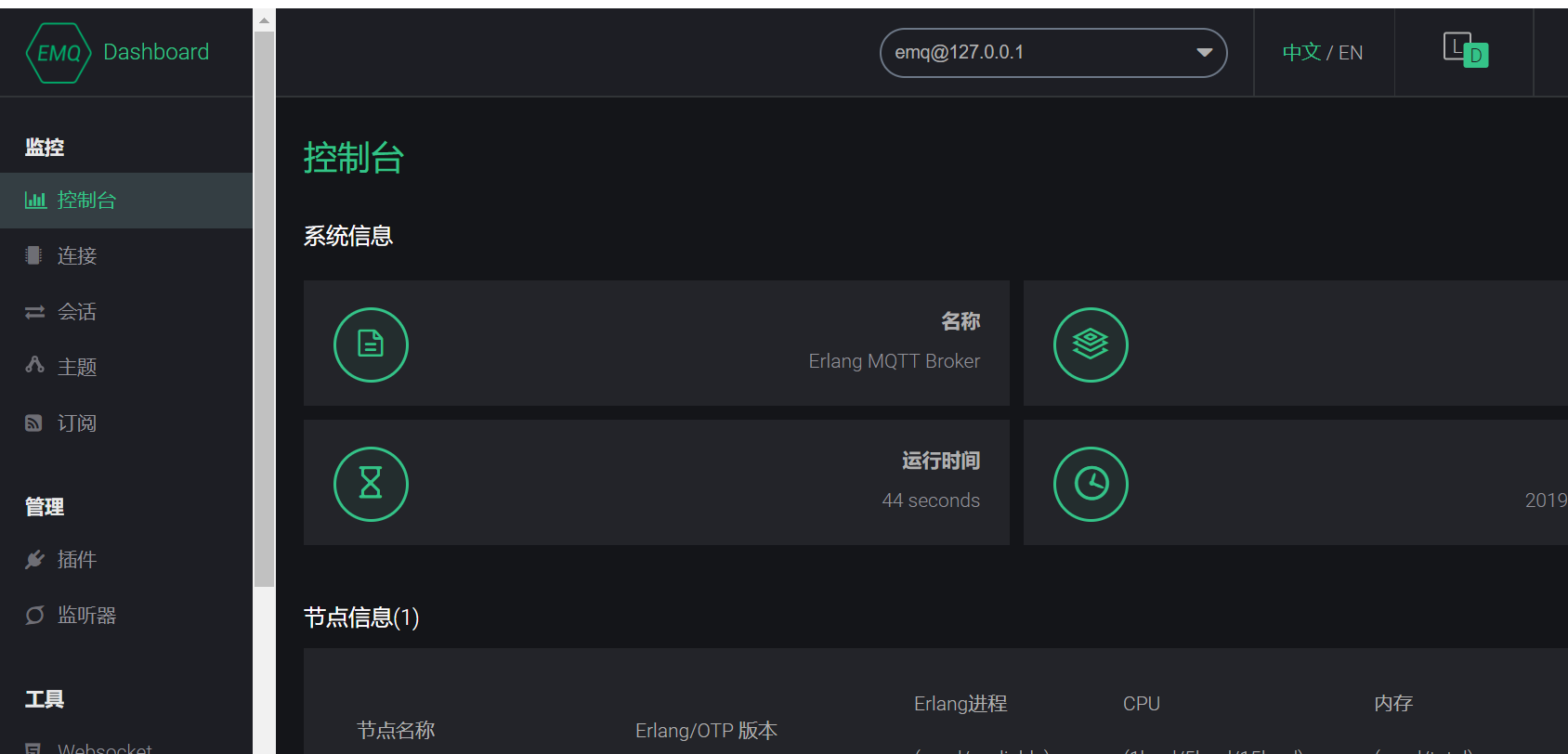
./bin/emqttd\_ctl status



管理控制台Dashboard

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

默认用户 admin/public



# EMQ 2.0功能列表

* 完整的 MQTT V3.1/V3.1.1 协议规范支持
* QoS0, QoS1, QoS2 消息支持
* 持久会话与离线消息支持
* Retained 消息支持
* Last Will 消息支持
* TCP/SSL 连接支持
* MQTT/WebSocket/SSL 支持
* HTTP消息发布接口支持
* $SYS/# 系统主题支持
* 客户端在线状态查询与订阅支持
* 客户端 ID 或 IP 地址认证支持
* 用户名密码认证支持
* LDAP 认证
* Redis、MySQL、PostgreSQL、MongoDB、HTTP 认证集成
* 浏览器 Cookie 认证
* 基于客户端 ID、IP 地址、用户名的访问控制(ACL)
* 多服务器节点集群(Cluster)
* 多服务器节点桥接(Bridge)
* mosquitto 桥接支持
* Stomp 协议支持
* MQTT-SN 协议支持
* CoAP 协议支持
* Stomp/SockJS 支持
* 通过 Paho 兼容性测试
* 2.0新功能: 本地订阅($local/topic)
* 2.0新功能: 共享订阅($share/<group>/topic)
* 2.0新功能: sysctl 类似 k = v 格式配置文件

# EMQ2.0 扩展插件列表

*EMQ* 2.0 支持丰富的扩展插件，包括控制台、扩展模块、多种认证方式、多种接入协议等:

|  |  |
| --- | --- |
| [emq\_plugin\_template](https://github.com/emqtt/emq_plugin_template) | 插件模版与演示代码 |
| [emq\_retainer](https://github.com/emqtt/emq_retainer) | Retain 消息存储插件 |
| [emq\_modules](https://github.com/emqtt/emq_modules) | Presence, Subscription 扩展模块插件 |
| [emq\_dashboard](https://github.com/emqtt/emq_dashboard) | Web 管理控制台，默认加载 |
| [emq\_auth\_clientid](https://github.com/emqtt/emq_auth_clientid) | ClientId、密码认证插件 |
| [emq\_auth\_username](https://github.com/emqtt/emq_auth_username) | 用户名、密码认证插件 |
| [emq\_auth\_ldap](https://github.com/emqtt/emq_auth_ldap) | LDAP 认证插件 |
| [emq\_auth\_http](https://github.com/emqtt/emq_auth_http) | HTTP 认证插件 |
| [emq\_auth\_mysql](https://github.com/emqtt/emq_auth_mysql) | MySQL 认证插件 |
| [emq\_auth\_pgsql](https://github.com/emqtt/emq_auth_pgsql) | PostgreSQL 认证插件 |
| [emq\_auth\_redis](https://github.com/emqtt/emq_auth_redis) | Redis 认证插件 |
| [emq\_auth\_mongo](https://github.com/emqtt/emq_auth_mongo) | MongoDB 认证插件 |
| [emq\_sn](https://github.com/emqtt/emq_sn) | MQTT-SN 协议插件 |
| [emq\_coap](https://github.com/emqtt/emq_coap) | CoAP 协议插件 |
| [emq\_stomp](https://github.com/emqtt/emq_stomp) | Stomp 协议插件 |
| [emq\_recon](https://github.com/emqtt/emq_recon) | Recon 优化调测插件 |
| [emq\_reloader](https://github.com/emqtt/emq_reloader) | 热升级插件(开发调试) |
| [emq\_sockjs](https://github.com/emqtt/emq_sockjs) | SockJS 插件(废弃) |

扩展插件通过 ‘bin/emqttd\_ctl’ 管理命令行，或 Dashboard 控制台加载启用。例如启用 PostgreSQL 认证插件:

./bin/emqttd\_ctl plugins load emq\_auth\_pgsql

# 测试工具

## 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

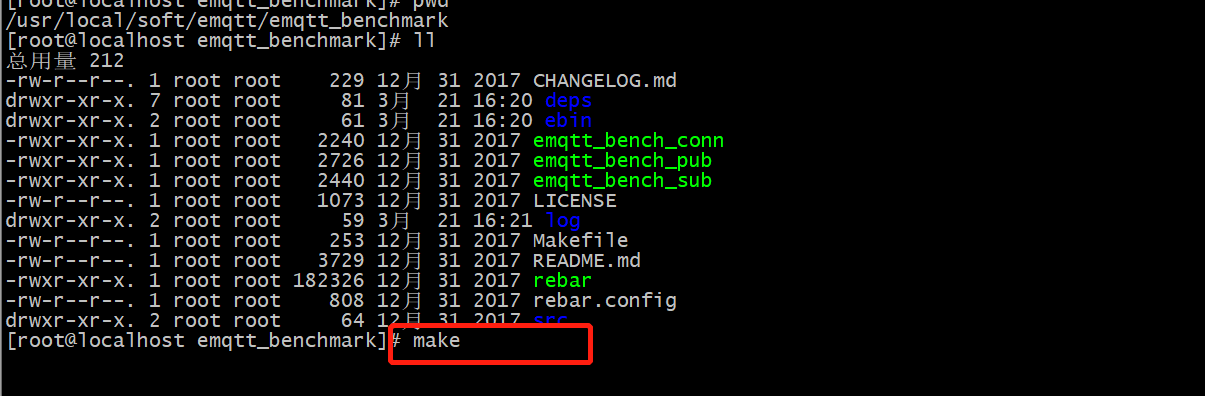
## 安装测试工具

### 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

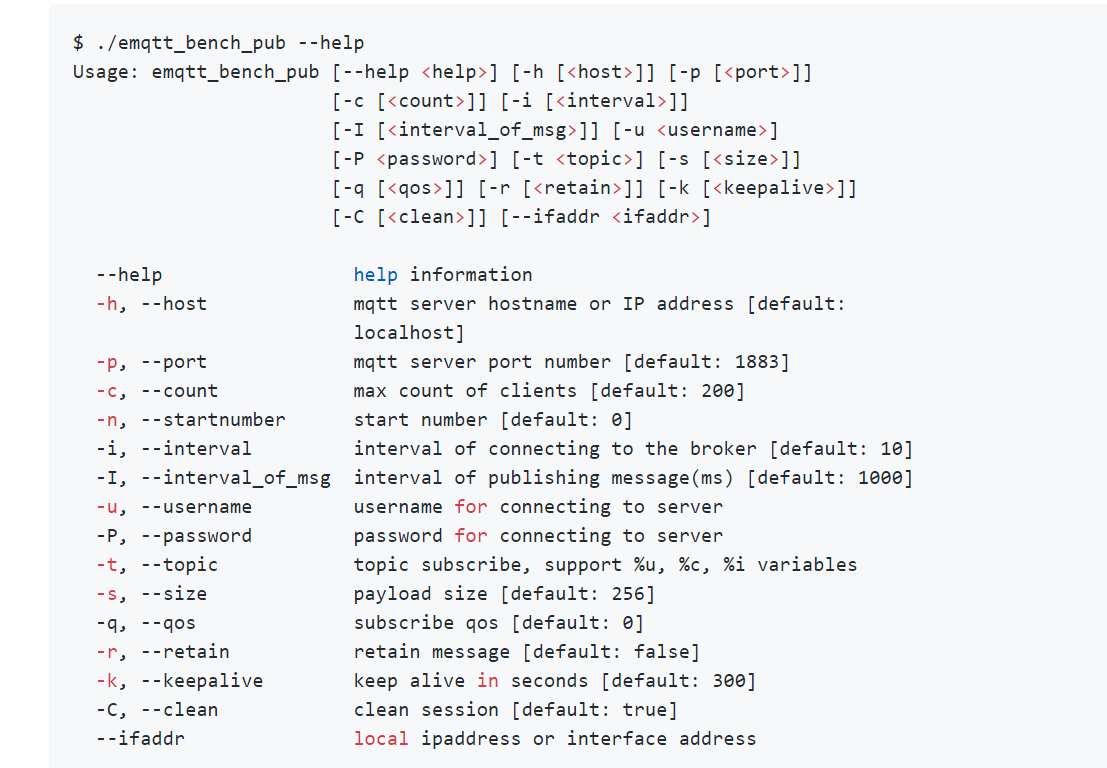
true: 表示服务端会创建一个持久会话, 在客户端断开连接时, 会话仍然会保持并保存离线消息. 直到会话超时注销.

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.

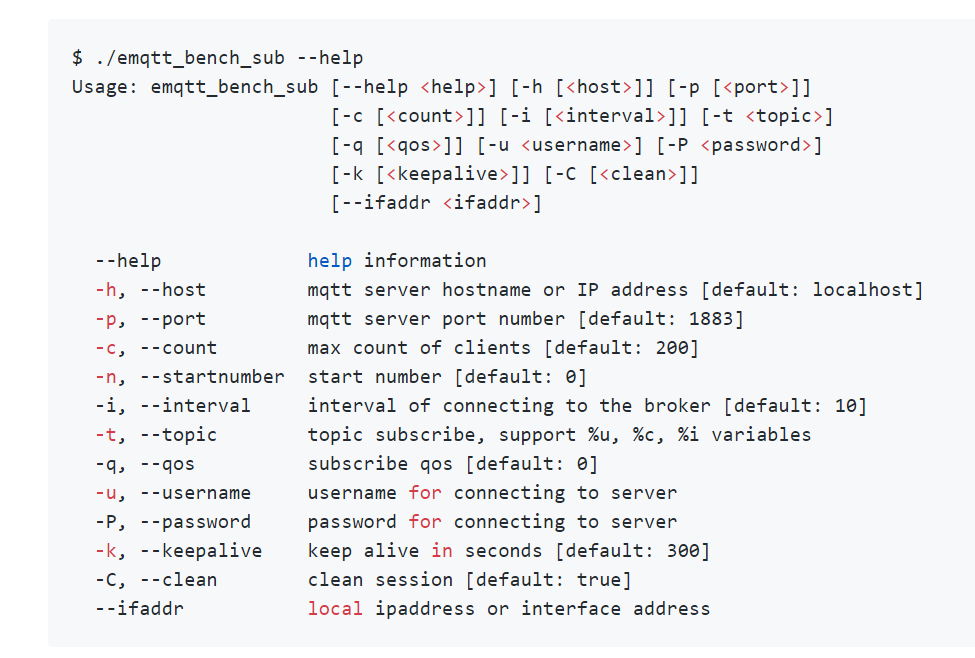
### 发布测试指令

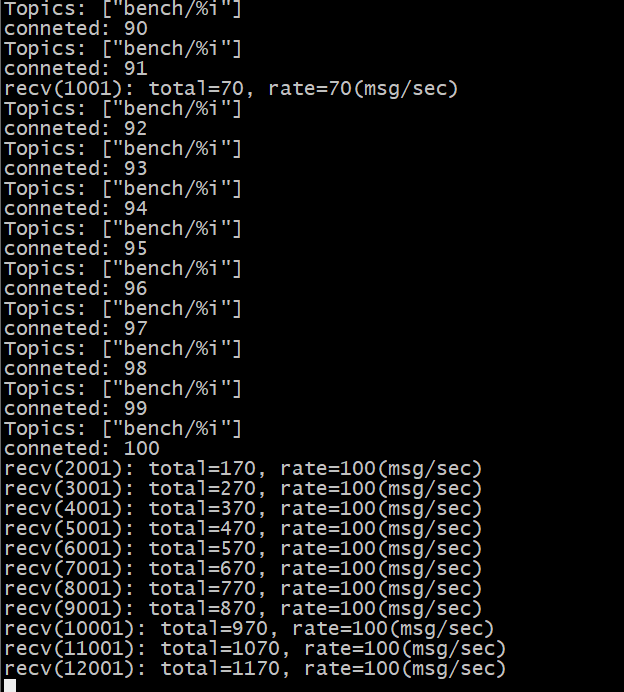
./emqtt\_bench\_pub -h 192.168.188.131 -c 100 -i 10 -t bench/%i



### 订阅测试指令

./emqtt\_bench\_sub –h 192.168.188.131 –c 100 –i 10 –t bench/%i





### 管理控制后台

