链滴

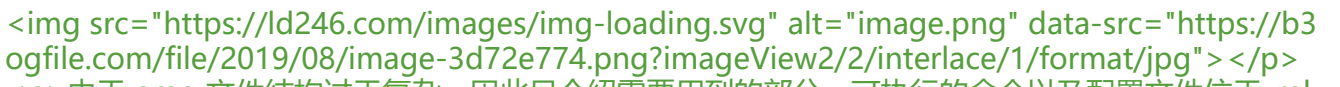
emq 的部署、使用及维护

作者：Leif160519

原文链接：https://ld246.com/article/1566526027464

来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="">
1、下载 emq 源码
emq 使用 3.0 以上版本（emq 2.X 版本称为 emqttd，3.X 版本称为 emqx），并且自己编译源（方便做扩展性修改）。git 地址：https://github.co/emqx/emqx-rel.git。
emq 的编译依赖于 <code>Erlang</code> 环境（emqx 依赖于 Erlang R21.2+ 版本，emqttd 依赖于 Erlang R20+ 版本）。Erlang 安装: <a href="https://ld246.com/forward?goto=http%3A2F%2Fwww.erlang.org%2F" target="_blank" rel="nofollow ugc">http://www.erlang.org/</a。
源码下载完成后，进入 emqx-rel 主文件夹，然后直接 make 编译。编译完成后的文件结构如下


由于 emq 文件结构过于复杂，因此只介绍需要用到的部分。可执行的命令以及配置文件位于_rel emqx 中，插件位于 deps 中。
2、emq 配置、运行和部署
进入 <code>emqx-rel/_rel/emqx/bin</code> 文件夹下，用 <code>emqx start</code> 令即可运行 emq，用 <code>emqx stop</code> 课停止 emq， <code>emqx console</code> 进入控制台运行模式，在此模式下可看见诸多类似设备连接、服务器报错等信息。
emq 的配置文件是 <code>emqx-rel/_rel/emqx/etc/emqx.conf</code>，下面介绍配置文中比较重要的部分。
<code>node.name=emqx@192.168.4.31</code> 服务器的节点名，需要 <code>emqx@host< code> 的格式，注意 host 中最好使用本机真实 ip，尤其是在集群配置中，否则会出现预期之外的错。并且，emq 启动之后不要修改此配置，确保 emq 开启和关闭的时候此配置必须相同，否则会出现法关闭 emqx 的错误（就算一开始写的 <code>127.0.0.1</code>，然后启动之后发现错了，也要关闭 emqx 再修改配置！）。
<code>listener.tcp.external = 0.0.0.0:1885</code> : emq 对外监听的 tcp 端口，默认 188。
<code>listener.ssl.external = 8883</code> : emq 监听的 ssl 端口，默认 8883。
<code>listener.ssl.external.keyfile = etc/certs/key.pem</code> : 服务器的私钥（关于 ssl 证的具体介绍见另一篇文档）。
<code>listener.ssl.external.certfile = etc/certs/cert.pem</code> : 服务器的公钥。
<code>listener.ssl.external.cacertfile = etc/certs/cacert.pem</code> : ca 证书。
<code>listener.ssl.external.verify = verify_peer</code> : 开启双向认证。
<code>listener.ssl.external.fail_if_no_peer_cert = true</code> : 如果客户端证书错误，则止连接。
如果开启单向认证，只需要配置服务器公钥和私钥，如果开启双向认证，则需要配置 ca、开启向认证、并禁止客户端无证书连接。
另外，在 <code>emqx-rel/_rel/emqx/etc/acl.conf</code> 中，有一处配置需要注意。
<pre><code class="highlight-chroma">{allow, {user, "dashboard"}, subscribe, ["$SYS/#"]}.

{allow, {ipaddr, "1 7.0.0.1"}, pubsub, ["$SYS/#", "#"]}.

%%{deny, all, sub cribe, ["$SYS/#", {eq, "#"}]}.

{allow, all}.
</code></pre>
这里对 emq 客户端的权限作了一些配置，对某些 topic 禁止客户端订阅。我们需要将第三行的置注释掉，否则客户端无法订阅 <code>$SYS/#</code> 的 topic（具体作用见下文）。
3、emq 集群部署

emq 提供了多种去中心化的集群部署方案：

在 <code>emqx-rel/_rel/emqx/etc/emqx.conf</code> 中可以配置集群的创建方式，在此绍一下手动创建集群的方法。
<code>cluster.discovery = manual</code> : 开启手动创建集群模式。
启动两台 <code>emqx</code> 节点（物理节点，非逻辑节点），假设两个节点分别为 <cod>emqx@node1</code> 和 <code>emqx@node2</code>，在任意一台节点（以 <code>node</code> 为例）上，执行命令：<code>emqx-rel/_rel/emqx/bin/emqx_ctl clsuter join emqx@nde2</code>
节点退出集群：<code>mqx-rel/_rel/emqx/bin/emqx_ctl cluster leave</code>
4、emq 的管理和监控
<code>emq</code> 提供了方便用户管理监控的控制台、api 及系统消息。控制台可直接访问 <code>http://{host}:18083</code>，默认用户名 <code>admin</code>，密码 <code>public/code>。

此处是插件功能和 emq 提供的 HTTP API 列表，可直接在控制台中进行配置和插件的开关，emq 支持插件的热加载。
除此之外，emq 提供了一些用于监控 broker 状态的系统主题 $SYS。其中，我们的上下线通知用到了下列两个 topic：

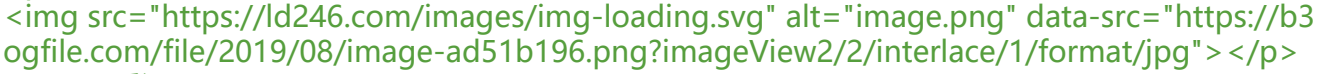
但是此处的上线通知流程存在问题，emq broker 会在客户端尝试连接（鉴权）时进行上线通知而不是连接成功之后通知。还有一种解决方案是用插件进行通知，但这样会使用 http 同步调用，代较大，因此暂时还是用系统消息进行异步通知。
5、插件的编写
插件位于 emqx-rel/deps 中，其中有若干已经由官方写好的插件，可以直接修改配置文件或直在控制台中进行配置。
如果需要自己写插件，emq 也提供了插件模板，在 <code>emqx-rel/deps/emqx_plugin_template</code> 中。打开 <code>emqx-rel/deps/emqx_plugin_template/src/emqx_plugin_templae.erl</code>，其中可以看到 emq 提供了一些钩子：
<pre><code class="highlight-chroma">   emqx:hook('client.authenticate', fun ?MODULE:on_client_authenticate/2, [Env]),
    emqx:hook('client.check_acl', fun ?MODULE:on_client_check_acl/5, [Env]),
    emqx:hook('client.connected', fun ?MODULE:on_client_connected/4, [Env]),
    emqx:hook('client.disconnected', fun ?MODULE:on_client_disconnected/3, [Env]),
    emqx:hook('client.subscribe', fun ?MODULE:on_client_subscribe/3, [Env]),
    emqx:hook('client.unsubscribe', fun ?MODULE:on_client_unsubscribe/3, [Env]),
    emqx:hook('session.created', fun ?MODULE:on_session_created/3, [Env]),
    emqx:hook('session.resumed', fun ?MODULE:on_session_resumed/3, [Env]),
    emqx:hook('session.subscribed', fun ?MODULE:on_session_subscribed/4, [Env]),
    emqx:hook('session.unsubscribed', fun ?MODULE:on_session_unsubscribed/4, [Env]),
    emqx:hook('session.terminated', fun ?MODULE:on_session_terminated/3, [Env]),

    emqx:hook('message.publish', fun ?MODULE:on_message_publish/2, [Env]),
    emqx:hook('message.deliver', fun ?MODULE:on_message_deliver/3, [Env]),
    emqx:hook('message.acked', fun ?MODULE:on_message_acked/3, [Env]),
    emqx:hook('message.dropped', fun ?MODULE:on_message_dropped/3, [Env]).
</code></pre>
当达成这些条件事，会触发钩子函数，用户可以自己定义钩子函数，例如**目前鉴权所用到的：
<pre><code class="highlight-chroma">on_client_authenticate(Credentials = #{client_id := ClientId, password := Password, username := Username}, _Env) ->
    io:format("Client(~s) authenticate, Password:~p ~n, Username:~p ~n", [ClientId, Password, Username]),
    inets:start(),
    {ok, {{Version, 200, ReasonPhrase}, Headers, Body}} = httpc:request(string:join(["http://www.example.com/api/mqtt/authenticate/login?username=", binary_to_list(base64:encode(binary_to_list(Username))), "&password=", binary_to_list(Password), "&clientId=", binary_to_list(base64:encode(binary_to_list(ClientId)))], "")),
        if
            Body == "" ->
            {stop, Credentials#{auth_result => success}};
        true ->
            {stop, Credentials#{auth_result => fail}}
        end.
</code></pre>
插件编写完成后，需要到主目录 emqx-rel 中重新 make。注意，<code>make</code> 之后置文件会重新生成，因此 make 之前最好备份一下配置文件。然后启动 emq，到控制台中开启插件可。
6、管理监控 API
emq 提供了一些 api，可以获取服务器的一些信息。

首先在 App 中新增用户

然后通过 <code>http://host:8080/{api}</code> 访问，访问需要 <code>http basic authentication</code>，用刚刚创建的用户可以访问。
7、emq 使用过程中可能遇到的问题及注意事项
1）<code>emqx start</code> 指令后，卡住不动，没有出现 <code>start successfully</code> 的字样，可能是端口被其他进程占用。emq 会占用的端口如下：

这些端口都可以在配置文件中自己修改，但是只要有一个冲突就无法正常启动。
2）<code>emqx stop</code> 时，显示 <code>emqx@xxx no response</code> 字样，

查配置文件中的节点名是否正确。
3）在开启集群模式时，如果要关闭一个节点，最好先手动让该节点退出集群，否则下一次启动节点时可能出现无法加入集群的情况（emq 会记住集群信息，重启时会自动恢复之前的集群状态，这个过程经常会出现一些问题，所以最好的办法就是手动退出集群，然后重启之后再手动加入集群）
4）emq 重启之后，会自动启动上一次启动过的插件，但是配置会回复初始化（除非修改配置文）。
5）emq 的启动进程为 <code>beam.smp</code>，守护进程为 <code>epmd</code>，是 Erlang 的进程结构比较奇怪，所以直接 <code>kill</code> 掉上述进程之后，emq 可能还在运中。所以最好执行 <code>emqx stop</code> 指令来停止 emq。
8、部署架构


9、后记
在官方文档中有更为全面的说明和指南，本文只列出了部分实际用到的比较重要的功能。如果后有扩展或本文中不详细之处，可参加官方文档：https://developer.emqx.io/docs/broker/v3/cn/getstarted.html
另附：插件编写指南 https://ww.cnblogs.com/wunaozai/p/8067621.html
官方文档对于部分内容也并不十分详细，本文对实际中遇到的问题和注意事项作了部分补充。</>