

RabbitMQ 集群与高可用配置

作者: [88250](#)

原文链接: <https://ld246.com/article/1380179992896>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<h2>集群概述</h2>

<div>通过 Erlang 的分布式特性（通过 magic cookie 认证节点）进行 RabbitMQ 集群，各 Rabbit Q 服务为对等节点，即每个节点都提供服务给客户端连接，进行消息发送与接收。</div>

<div> </div>

<div>这些节点通过 RabbitMQ A 队列（镜像队列）进行消息队列结构复制。本方案中搭建 3 个节点，并且都是磁盘节点（所节点状态保持一致，节点完全对等），只要有任何一个节点能够工作，RabbitMQ 集群对外就能提供务。</div>

<h2>环境</h2>

<div>

CentOS 6.4, 64位

RabbitMQ 3.1.5

HAProxy 1.4.22

</div>

<div> </div>

<div>RabbitMQ 集群安装在 3 个节点上：192.168.1.1、192.168.1.2、192.168.1.3；</div>

<div>HAProxy 安装在 192.168.1.4 上，用于对外提供 RabbitMQ 均衡。</div>

<div> </div>

<div>所需安装包下载。</div>

<h2>配置步骤</h2>

<div>1. 安装 erlang、rabbitmq</div>

<div>在 192.168.1.1、192.168.1.2、192.168.1.3 三个节点上安装，然后开启 RabbitMQ 监控插件</div>

<div>

<pre class="brush: bash; gutter: false; toolbar: false; auto-links: false">rabbitmq-plugins enable rabbitmq_management</pre>

</div>

<div>2. 修改 /etc/hosts </div>

<div>加入集群 3 个节点的描述：</div>

<div>

<pre class="brush: bash; gutter: false; toolbar: false; auto-links: false">192.168.1.1 node1
192.168.1.2 node2
192.168.1.3 node3</pre>

</div>

<div>

<div> </div>

<div>3. 设置 Erlang Cookie </div>

<div>Erlang Cookie 文件：/var/lib/rabbitmq/.erlang.cookie。这里将 node1 的该文件复制到 node2、node3，由于这个文件权限是 400，所以需要先修改 node2、node3 中的该文件权限为 777：</div>

<div>

<div>

<pre class="brush: bash; gutter: false; toolbar: false; auto-links: false"># chmod 777 /var/lib/rabbitmq/.erlang.cookie</pre>

</div>

<div>然后将 node1 中的该文件拷贝到 node2、node3，最后将权限和所属用户/组修改回来：</div>

<div>

<div>

<pre class="brush: bash; gutter: false; toolbar: false; auto-links: false"># chmod 400 /var/lib/rabbitmq/.erlang.cookie

```
# chown rabbitmq /var/lib/rabbitmq/.erlang.cookie
# chgrp rabbitmq /var/lib/rabbitmq/.erlang.cookie</pre>
</div>
<div> </div>
<div><strong>4. 使用 -detached 参数运行各节点</strong></div>
<div>
<pre class="brush: bash; gutter: false; toolbar: false; auto-links: false"># rabbitmqctl stop
# rabbitmq-server -detached</pre>
</div>
<div> </div>
<div><strong>5. 组成集群</strong></div>
<div>将 node2、node3 与 node1 组成集群: </div>
<div> </div>
<div>
<pre class="brush: bash; gutter: false; toolbar: false; auto-links: false">node2 # rabbitmqctl st
p_app
node2 # rabbitmqctl join_cluster rabbit@node1
node2 # rabbitmqctl start_app

node3 # rabbitmqctl stop_app
node3 # rabbitmqctl join_cluster rabbit@node1
node3 # rabbitmqctl start_app</pre>

</div>
</div>
<div> </div>
<div></div>
<div> </div>
<div>此时 node2 与 node3 也会自动建立连接; 如果要使用内存节点, 则可以使用</div>
<div>
<div>node2 # rabbitmqctl join_cluster <em>--ram</em> rabbit@node1 加入集群。</div>
<div> </div>
<div>集群配置好后, 可以在 RabbitMQ 任意节点上执行 rabbitmqctl cluster_status 来查看是否集
配置成功。</div>
<div> </div>
<div><strong>6. 设置镜像队列策略</strong></div>
<div>在任意一个节点上执行: </div>
<div>
<pre class="brush: bash; gutter: false; toolbar: false; auto-links: false"># rabbitmqctl set_polic
ha-all "^" '{"ha-mode":"all"}'</pre>
</div>
<div>将所有队列设置为镜像队列, 即队列会被复制到各个节点, 各个节点状态保持一直。</div>
<div> </div>
<div>完成这 6 个步骤后, RabbitMQ 高可用集群就已经搭建好了, 最后一个步骤就是搭建均衡器。
</div>
<div> </div>
<div><strong>7. 安装并配置 HAProxy</strong></div>
<div>在 192.168.1.4 上安装 HAProxy, 然后修改 /etc/haproxy/haproxy.cfg: </div>
<div> </div>
<div>
<pre class="brush: bash; gutter: false; toolbar: false; auto-links: false">listen rabbitmq_cluster
0.0.0.0:5672

mode tcp
```

balance roundrobin

```
server node1 192.168.1.1:5672 check inter 2000 rise 2 fall 3
```

```
server node2 192.168.1.2:5672 check inter 2000 rise 2 fall 3
```

```
server node2 192.168.1.3:5672 check inter 2000 rise 2 fall 3</pre>
```

</div>

</div>