

- [一文搞懂 RabbitMQ 的重要概念以及安装](#)
 - [一 RabbitMQ 介绍](#)
 - [1.1 RabbitMQ 简介](#)
 - [1.2 RabbitMQ 核心概念](#)
 - [1.2.1 Producer\(生产者\) 和 Consumer\(消费者\)](#)
 - [1.2.2 Exchange\(交换器\)](#)
 - [1.2.3 Queue\(消息队列\)](#)
 - [1.2.4 Broker \(消息中间件的服务节点\)](#)
 - [1.2.5 Exchange Types\(交换器类型\)](#)
 - [① fanout](#)
 - [② direct](#)
 - [③ topic](#)
 - [④ headers\(不推荐\)](#)
 - [二 安装 RabbitMQ](#)
 - [2.1 安装 erlang](#)
 - [2.2 安装 RabbitMQ](#)

一文搞懂 RabbitMQ 的重要概念以及安装

一 RabbitMQ 介绍

这部分参考了《RabbitMQ实战指南》这本书的第 1 章和第 2 章。

1.1 RabbitMQ 简介

RabbitMQ 是采用 Erlang 语言实现 AMQP(Advanced Message Queuing Protocol, 高级消息队列协议) 的消息中间件, 它最初起源于金融系统, 用于在分布式系统中存储转发消息。

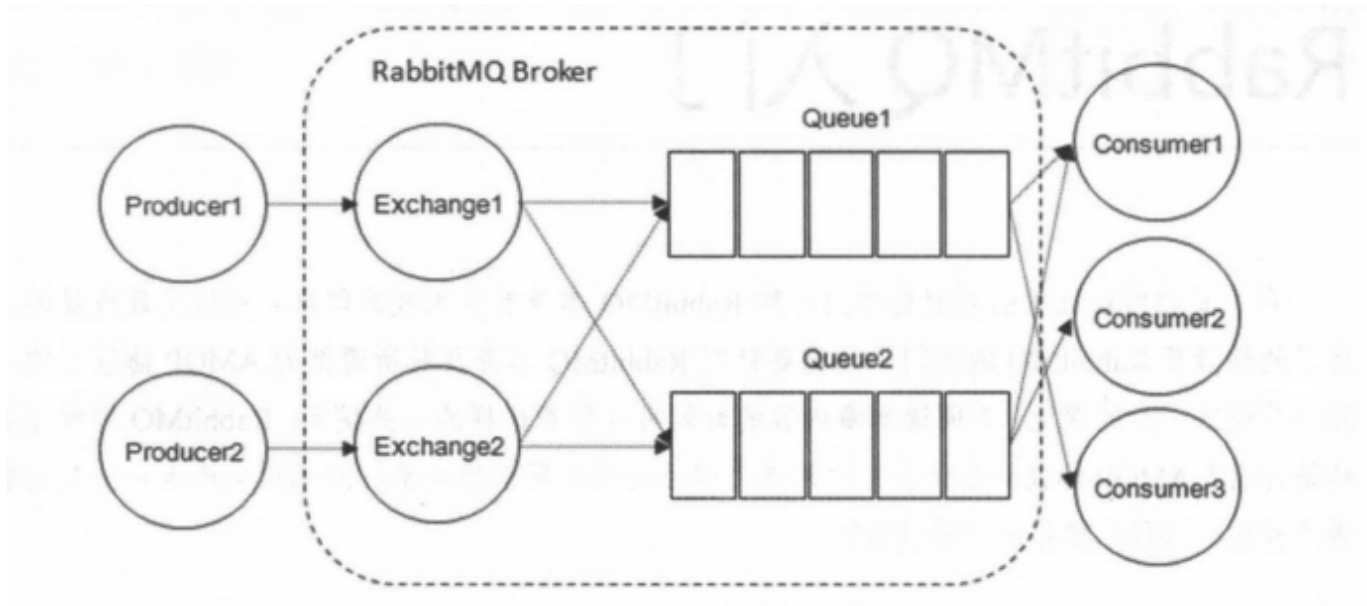
RabbitMQ 发展到今天, 被越来越多的人认可, 这和它在易用性、扩展性、可靠性和高可用性等方面的卓著表现是分不开的。RabbitMQ 的具体特点可以概括为以下几点:

- **可靠性:** RabbitMQ使用一些机制来保证消息的可靠性, 如持久化、传输确认及发布确认等。
- **灵活的路由:** 在消息进入队列之前, 通过交换器来路由消息。对于典型的路由功能, RabbitMQ 已经提供了一些内置的交换器来实现。针对更复杂的路由功能, 可以将多个交换器绑定在一起, 也可以通过插件机制来实现自己的交换器。这个后面会在我们将 RabbitMQ 核心概念的时候详细介绍到。
- **扩展性:** 多个RabbitMQ节点可以组成一个集群, 也可以根据实际业务情况动态地扩展集群中节点。
- **高可用性:** 队列可以在集群中的机器上设置镜像, 使得在部分节点出现问题的情况下队列仍然可用。
- **支持多种协议:** RabbitMQ 除了原生支持 AMQP 协议, 还支持 STOMP、MQTT 等多种消息中间件协议。
- **多语言客户端:** RabbitMQ几乎支持所有常用语言, 比如 Java、Python、Ruby、PHP、C#、JavaScript 等。
- **易用的管理界面:** RabbitMQ提供了一个易用的用户界面, 使得用户可以监控和管理消息、集群中的节点等。在安装 RabbitMQ 的时候会介绍到, 安装好 RabbitMQ 就自带管理界面。
- **插件机制:** RabbitMQ 提供了许多插件, 以实现从多方面进行扩展, 当然也可以编写自己的插件。感觉这个有点类似 Dubbo 的 SPI机制。

1.2 RabbitMQ 核心概念

RabbitMQ 整体上是一个生产者与消费者模型，主要负责接收、存储和转发消息。可以把消息传递的过程想象成：当你将一个包裹送到邮局，邮局会暂存并最终将邮件通过邮递员送到收件人的手上，RabbitMQ就好比由邮局、邮箱和邮递员组成的一个系统。从计算机术语层面来说，RabbitMQ 模型更像是一种交换机模型。

下面再来看看图1—— RabbitMQ 的整体模型架构。



下面我会——介绍上图中的一些概念。

1.2.1 Producer(生产者) 和 Consumer(消费者)

- **Producer(生产者)** :生产消息的一方（邮件投递者）
- **Consumer(消费者)** :消费消息的一方（邮件收件人）

消息一般由 2 部分组成：**消息头**（或者说是标签 Label）和 **消息体**。消息体也可以称为 payload ,消息体是不透明的，而消息头则由一系列的可选属性组成，这些属性包括 routing-key（路由键）、priority（相对于其他消息的优先权）、delivery-mode（指出该消息可能需要持久性存储）等。生产者把消息交由 RabbitMQ 后，RabbitMQ 会根据消息头把消息发送给感兴趣的 Consumer(消费者)。

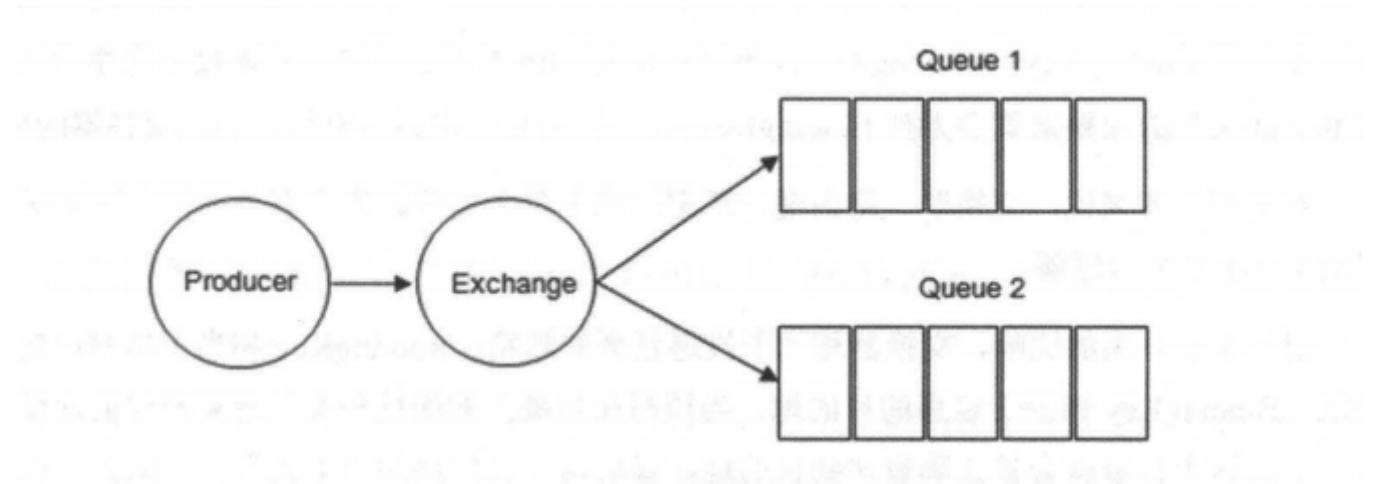
1.2.2 Exchange(交换器)

在 RabbitMQ 中，消息并不是直接被投递到 **Queue(消息队列)** 中的，中间还必须经过 **Exchange(交换器)** 这一层，**Exchange(交换器)** 会把我们的消息分配到对应的 **Queue(消息队列)** 中。

Exchange(交换器) 用来接收生产者发送的消息并将这些消息路由给服务器中的队列中，如果路由不到，或许会返回给 **Producer(生产者)**，或许会被直接丢弃掉。这里可以将RabbitMQ中的交换器看作一个简单的实体。

RabbitMQ 的 Exchange(交换器) 有4种类型，不同的类型对应着不同的路由策略：direct(默认)，fanout，topic，和 headers，不同类型的Exchange转发消息的策略有所区别。这个会在介绍 **Exchange Types(交换器类型)** 的时候介绍到。

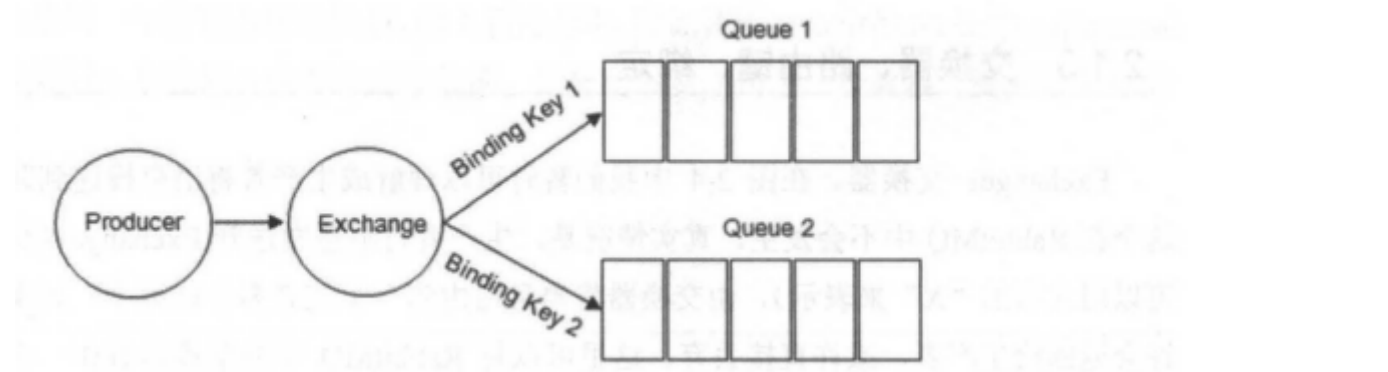
Exchange(交换器) 示意图如下：



生产者将消息发给交换器的时候，一般会指定一个 **RoutingKey(路由键)**，用来指定这个消息的路由规则，而这个 **RoutingKey** 需要与交换器类型和绑定键(**BindingKey**)联合使用才能最终生效。

RabbitMQ 中通过 **Binding(绑定)** 将 **Exchange(交换器)** 与 **Queue(消息队列)** 关联起来，在绑定的时候一般会指定一个 **BindingKey(绑定键)**，这样 RabbitMQ 就知道如何正确将消息路由到队列了,如下图所示。一个绑定就是基于路由键将交换器和消息队列连接起来的路由规则，所以可以将交换器理解成一个由绑定构成的路由表。Exchange 和 Queue 的绑定可以是多对多的关系。

Binding(绑定) 示意图：



生产者将消息发送给交换器时，需要一个RoutingKey,当 BindingKey 和 RoutingKey 相匹配时，消息会被路由到对应的队列中。在绑定多个队列到同一个交换器的时候，这些绑定允许使用相同的 BindingKey。BindingKey 并不是在所有的情况下都生效，它依赖于交换器类型，比如fanout类型的交换器就会无视，而是将消息路由到所有绑定到该交换器的队列中。

1.2.3 Queue(消息队列)

Queue(消息队列) 用来保存消息直到发送给消费者。它是消息的容器，也是消息的终点。一个消息可投入一个或多个队列。消息一直在队列里面，等待消费者连接到这个队列将其取走。

RabbitMQ 中消息只能存储在 **队列** 中，这一点和 **Kafka** 这种消息中间件相反。Kafka 将消息存储在 **topic (主题)** 这个逻辑层面，而相对应的队列逻辑只是topic实际存储文件中的位移标识。RabbitMQ 的生产者生产消息并最终投递到队列中，消费者可以从队列中获取消息并消费。

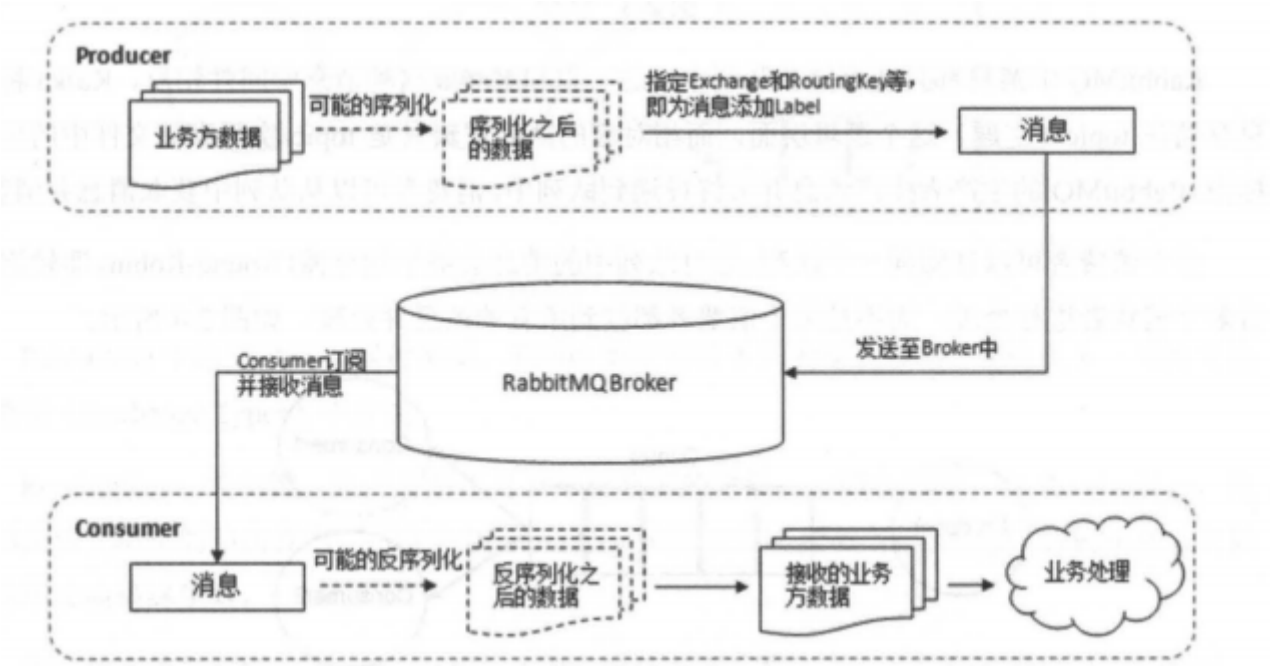
多个消费者可以订阅同一个队列，这时队列中的消息会被平均分摊（Round-Robin，即轮询）给多个消费者进行处理，而不是每个消费者都收到所有的消息并处理，这样避免的消息被重复消费。

RabbitMQ 不支持队列层面的广播消费,如果有广播消费的需求,需要在其上进行二次开发,这样会很麻烦,不建议这样做。

1.2.4 Broker（消息中间件的服务节点）

对于 RabbitMQ 来说,一个 RabbitMQ Broker 可以简单地看作一个 RabbitMQ 服务节点,或者RabbitMQ服务实例。大多数情况下也可以将一个 RabbitMQ Broker 看作一台 RabbitMQ 服务器。

下图展示了生产者将消息存入 RabbitMQ Broker,以及消费者从Broker中消费数据的整个流程。



这样图1中的一些关于 RabbitMQ 的基本概念我们就介绍完毕了,下面再来介绍一下 **Exchange Types(交换器类型)**。

1.2.5 Exchange Types(交换器类型)

RabbitMQ 常用的 Exchange Type 有 **fanout**、**direct**、**topic**、**headers** 这四种 (AMQP规范里还提到两种 Exchange Type, 分别为 system 与 自定义, 这里不予以描述)。

① fanout

fanout 类型的Exchange路由规则非常简单, 它会把所有发送到该Exchange的消息路由到所有与它绑定的Queue中, 不需要做任何判断操作, 所以 fanout 类型是所有的交换机类型里面速度最快的。fanout 类型常用来广播消息。

② direct

direct 类型的Exchange路由规则也很简单, 它会把消息路由到那些 Bindingkey 与 RoutingKey 完全匹配的 Queue 中。

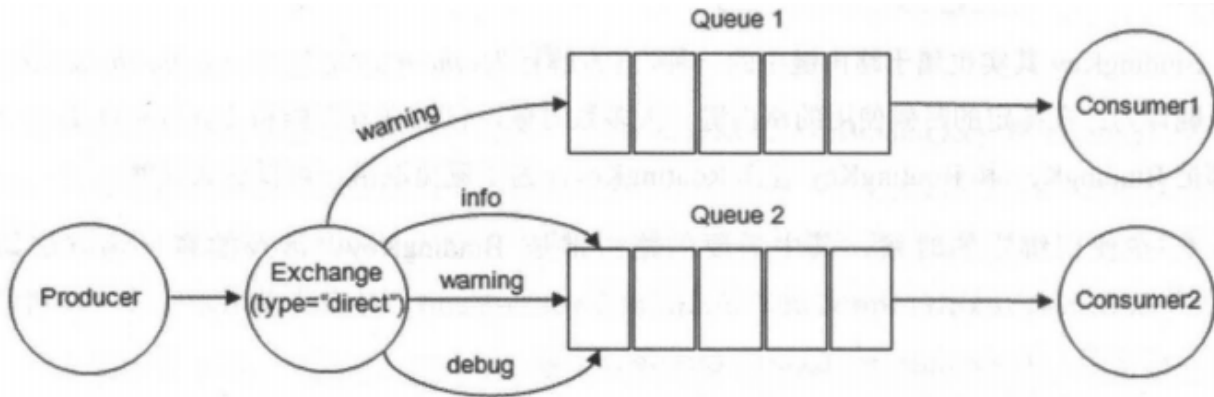


图 2-7 direct 类型的交换器

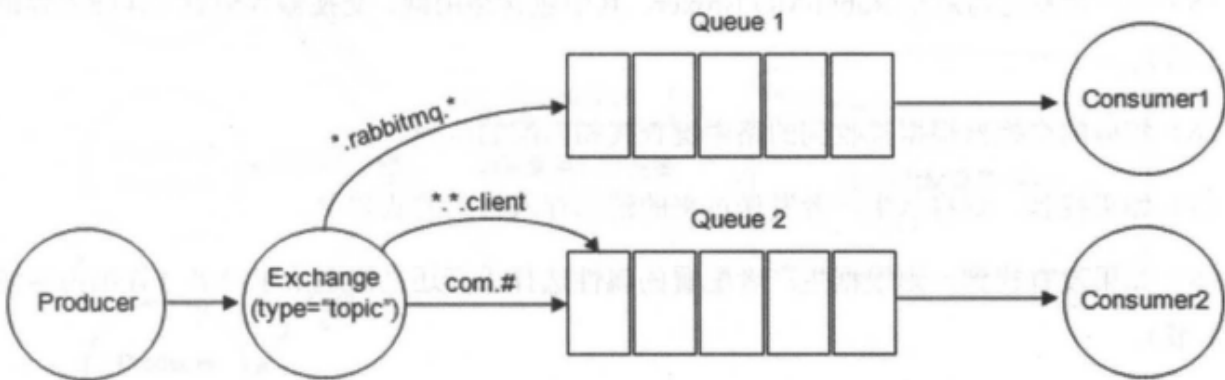
以上图为例，如果发送消息的时候设置路由键为“warning”,那么消息会路由到 Queue1 和 Queue2。如果在发送消息的时候设置路由键为“Info”或者“debug”，消息只会路由到Queue2。如果以其他的路由键发送消息，则消息不会路由到这两个队列中。

direct 类型常用在处理有优先级的任务，根据任务的优先级把消息发送到对应的队列，这样可以指派更多的资源去处理高优先级的队列。

③ topic

前面讲到direct类型的交换器路由规则是完全匹配 BindingKey 和 RoutingKey ，但是这种严格的匹配方式在很多情况下不能满足实际业务的需求。topic类型的交换器在匹配规则上进行了扩展，它与 direct 类型的交换器相似，也是将消息路由到 BindingKey 和 RoutingKey 相匹配的队列中，但这里的匹配规则有些不同，它约定：

- RoutingKey 为一个点号“.”分隔的字符串（被点号“.”分隔开的每一段独立的字符串称为一个单词），如 “com.rabbitmq.client”、“java.util.concurrent”、“com.hidden.client”；
- BindingKey 和 RoutingKey 一样也是点号“.”分隔的字符串；
- BindingKey 中可以存在两种特殊字符串“*”和“#”，用于做模糊匹配，其中“*”用于匹配一个单词，“#”用于匹配多个单词(可以是零个)。



以上图为例：

- 路由键为 “com.rabbitmq.client” 的消息会同时路由到 Queue1 和 Queue2；
- 路由键为 “com.hidden.client” 的消息只会路由到 Queue2 中；
- 路由键为 “com.hidden.demo” 的消息只会路由到 Queue2 中；
- 路由键为 “java.rabbitmq.demo” 的消息只会路由到Queue1中；

- 路由键为 "java.util.concurrent" 的消息将会被丢弃或者返回给生产者（需要设置 mandatory 参数），因为它没有匹配任何路由键。

④ headers(不推荐)

headers 类型的交换器不依赖于路由键的匹配规则来路由消息，而是根据发送的消息内容中的 headers 属性进行匹配。在绑定队列和交换器时制定一组键值对，当发送消息到交换器时，RabbitMQ会获取到该消息的 headers（也是一个键值对的形式）对比其中的键值对是否完全匹配队列和交换器绑定时指定的键值对，如果完全匹配则消息会路由到该队列，否则不会路由到该队列。headers 类型的交换器性能会很差，而且也不实用，基本上不会看到它的存在。

二 安装 RabbitMq

通过 Docker 安装非常方便，只需要几条命令就好了，我这里是只说一下常规安装方法。

前面提到了 RabbitMQ 是由 Erlang语言编写的，也正因如此，在安装RabbitMQ 之前需要安装 Erlang。

注意：在安装 RabbitMQ 的时候需要注意 RabbitMQ 和 Erlang 的版本关系，如果不注意的话会导致出错，两者对应关系如下：

在安装rabbitmq的时候需要注意rabbitmq和erlang版本的关系，如果两者关系不——对应的话，那么会导致出错的。rabbitmq版本和erlang的关系

[Compatibility Matrix](#).

RabbitMQ and Erlang/OTP Compatibility Matrix

RabbitMQ version	Minimum required Erlang/OTP	Maximum supported Erlang/OTP	Notes
<ul style="list-style-type: none">3.7.123.7.113.7.103.7.93.7.83.7.7	<ul style="list-style-type: none">20.3.x	<ul style="list-style-type: none">21.x	<ul style="list-style-type: none">Erlang/OTP 19.3.x support was discontinued as of Jan 1st, 2019For the best TLS support, the latest version of Erlang/OTP 21.x is recommendedOn Windows, Erlang/OTP 20.2 changed default cookie file location
<ul style="list-style-type: none">3.7.63.7.53.7.43.7.33.7.23.7.13.7.0	<ul style="list-style-type: none">19.3	<ul style="list-style-type: none">20.3.x	<ul style="list-style-type: none">For the best TLS support, the latest version of Erlang/OTP 20.3.x is recommendedErlang versions 19.3.4, 19.3.5, 19.3.6, 19.3.7, 19.3.8, 19.3.9, 19.3.10, 19.3.11, 19.3.12, 19.3.13, 19.3.14, 19.3.15, 19.3.16, 19.3.17, 19.3.18, 19.3.19, 19.3.20, 19.3.21, 19.3.22, 19.3.23, 19.3.24, 19.3.25, 19.3.26, 19.3.27, 19.3.28, 19.3.29, 19.3.30, 19.3.31, 19.3.32, 19.3.33, 19.3.34, 19.3.35, 19.3.36, 19.3.37, 19.3.38, 19.3.39, 19.3.40, 19.3.41, 19.3.42, 19.3.43, 19.3.44, 19.3.45, 19.3.46, 19.3.47, 19.3.48, 19.3.49, 19.3.50, 19.3.51, 19.3.52, 19.3.53, 19.3.54, 19.3.55, 19.3.56, 19.3.57, 19.3.58, 19.3.59, 19.3.60, 19.3.61, 19.3.62, 19.3.63, 19.3.64, 19.3.65, 19.3.66, 19.3.67, 19.3.68, 19.3.69, 19.3.70, 19.3.71, 19.3.72, 19.3.73, 19.3.74, 19.3.75, 19.3.76, 19.3.77, 19.3.78, 19.3.79, 19.3.80, 19.3.81, 19.3.82, 19.3.83, 19.3.84, 19.3.85, 19.3.86, 19.3.87, 19.3.88, 19.3.89, 19.3.90, 19.3.91, 19.3.92, 19.3.93, 19.3.94, 19.3.95, 19.3.96, 19.3.97, 19.3.98, 19.3.99, 19.3.100, 19.3.101, 19.3.102, 19.3.103, 19.3.104, 19.3.105, 19.3.106, 19.3.107, 19.3.108, 19.3.109, 19.3.110, 19.3.111, 19.3.112, 19.3.113, 19.3.114, 19.3.115, 19.3.116, 19.3.117, 19.3.118, 19.3.119, 19.3.120, 19.3.121, 19.3.122, 19.3.123, 19.3.124, 19.3.125, 19.3.126, 19.3.127, 19.3.128, 19.3.129, 19.3.130, 19.3.131, 19.3.132, 19.3.133, 19.3.134, 19.3.135, 19.3.136, 19.3.137, 19.3.138, 19.3.139, 19.3.140, 19.3.141, 19.3.142, 19.3.143, 19.3.144, 19.3.145, 19.3.146, 19.3.147, 19.3.148, 19.3.149, 19.3.150, 19.3.151, 19.3.152, 19.3.153, 19.3.154, 19.3.155, 19.3.156, 19.3.157, 19.3.158, 19.3.159, 19.3.160, 19.3.161, 19.3.162, 19.3.163, 19.3.164, 19.3.165, 19.3.166, 19.3.167, 19.3.168, 19.3.169, 19.3.170, 19.3.171, 19.3.172, 19.3.173, 19.3.174, 19.3.175, 19.3.176, 19.3.177, 19.3.178, 19.3.179, 19.3.180, 19.3.181, 19.3.182, 19.3.183, 19.3.184, 19.3.185, 19.3.186, 19.3.187, 19.3.188, 19.3.189, 19.3.190, 19.3.191, 19.3.192, 19.3.193, 19.3.194, 19.3.195, 19.3.196, 19.3.197, 19.3.198, 19.3.199, 19.3.200, 19.3.201, 19.3.202, 19.3.203, 19.3.204, 19.3.205, 19.3.206, 19.3.207, 19.3.208, 19.3.209, 19.3.210, 19.3.211, 19.3.212, 19.3.213, 19.3.214, 19.3.215, 19.3.216, 19.3.217, 19.3.218, 19.3.219, 19.3.220, 19.3.221, 19.3.222, 19.3.223, 19.3.224, 19.3.225, 19.3.226, 19.3.227, 19.3.228, 19.3.229, 19.3.230, 19.3.231, 19.3.232, 19.3.233, 19.3.234, 19.3.235, 19.3.236, 19.3.237, 19.3.238, 19.3.239, 19.3.240, 19.3.241, 19.3.242, 19.3.243, 19.3.244, 19.3.245, 19.3.246, 19.3.247, 19.3.248, 19.3.249, 19.3.250, 19.3.251, 19.3.252, 19.3.253, 19.3.254, 19.3.255, 19.3.256, 19.3.257, 19.3.258, 19.3.259, 19.3.260, 19.3.261, 19.3.262, 19.3.263, 19.3.264, 19.3.265, 19.3.266, 19.3.267, 19.3.268, 19.3.269, 19.3.270, 19.3.271, 19.3.272, 19.3.273, 19.3.274, 19.3.275, 19.3.276, 19.3.277, 19.3.278, 19.3.279, 19.3.280, 19.3.281, 19.3.282, 19.3.283, 19.3.284, 19.3.285, 19.3.286, 19.3.287, 19.3.288, 19.3.289, 19.3.290, 19.3.291, 19.3.292, 19.3.293, 19.3.294, 19.3.295, 19.3.296, 19.3.297, 19.3.298, 19.3.299, 19.3.300, 19.3.301, 19.3.302, 19.3.303, 19.3.304, 19.3.305, 19.3.306, 19.3.307, 19.3.308, 19.3.309, 19.3.310, 19.3.311, 19.3.312, 19.3.313, 19.3.314, 19.3.315, 19.3.316, 19.3.317, 19.3.318, 19.3.319, 19.3.320, 19.3.321, 19.3.322, 19.3.323, 19.3.324, 19.3.325, 19.3.326, 19.3.327, 19.3.328, 19.3.329, 19.3.330, 19.3.331, 19.3.332, 19.3.333, 19.3.334, 19.3.335, 19.3.336, 19.3.337, 19.3.338, 19.3.339, 19.3.340, 19.3.341, 19.3.342, 19.3.343, 19.3.344, 19.3.345, 19.3.346, 19.3.347, 19.3.348, 19.3.349, 19.3.350, 19.3.351, 19.3.352, 19.3.353, 19.3.354, 19.3.355, 19.3.356, 19.3.357, 19.3.358, 19.3.359, 19.3.360, 19.3.361, 19.3.362, 19.3.363, 19.3.364, 19.3.365, 19.3.366, 19.3.367, 19.3.368, 19.3.369, 19.3.370, 19.3.371, 19.3.372, 19.3.373, 19.3.374, 19.3.375, 19.3.376, 19.3.377, 19.3.378, 19.3.379, 19.3.380, 19.3.381, 19.3.382, 19.3.383, 19.3.384, 19.3.385, 19.3.386, 19.3.387, 19.3.388, 19.3.389, 19.3.390, 19.3.391, 19.3.392, 19.3.393, 19.3.394, 19.3.395, 19.3.396, 19.3.397, 19.3.398, 19.3.399, 19.3.400, 19.3.401, 19.3.402, 19.3.403, 19.3.404, 19.3.405, 19.3.406, 19.3.407, 19.3.408, 19.3.409, 19.3.410, 19.3.411, 19.3.412, 19.3.413, 19.3.414, 19.3.415, 19.3.416, 19.3.417, 19.3.418, 19.3.419, 19.3.420, 19.3.421, 19.3.422, 19.3.423, 19.3.424, 19.3.425, 19.3.426, 19.3.427, 19.3.428, 19.3.429, 19.3.430, 19.3.431, 19.3.432, 19.3.433, 19.3.434, 19.3.435, 19.3.436, 19.3.437, 19.3.438, 19.3.439, 19.3.440, 19.3.441, 19.3.442, 19.3.443, 19.3.444, 19.3.445, 19.3.446, 19.3.447, 19.3.448, 19.3.449, 19.3.450, 19.3.451, 19.3.452, 19.3.453, 19.3.454, 19.3.455, 19.3.456, 19.3.457, 19.3.458, 19.3.459, 19.3.460, 19.3.461, 19.3.462, 19.3.463, 19.3.464, 19.3.465, 19.3.466, 19.3.467, 19.3.468, 19.3.469, 19.3.470, 19.3.471, 19.3.472, 19.3.473, 19.3.474, 19.3.475, 19.3.476, 19.3.477, 19.3.478, 19.3.479, 19.3.480, 19.3.481, 19.3.482, 19.3.483, 19.3.484, 19.3.485, 19.3.486, 19.3.487, 19.3.488, 19.3.489, 19.3.490, 19.3.491, 19.3.492, 19.3.493, 19.3.494, 19.3.495, 19.3.496, 19.3.497, 19.3.498, 19.3.499, 19.3.500, 19.3.501, 19.3.502, 19.3.503, 19.3.504, 19.3.505, 19.3.506, 19.3.507, 19.3.508, 19.3.509, 19.3.510, 19.3.511, 19.3.512, 19.3.513, 19.3.514, 19.3.515, 19.3.516, 19.3.517, 19.3.518, 19.3.519, 19.3.520, 19.3.521, 19.3.522, 19.3.523, 19.3.524, 19.3.525, 19.3.526, 19.3.527, 19.3.528, 19.3.529, 19.3.530, 19.3.531, 19.3.532, 19.3.533, 19.3.534, 19.3.535, 19.3.536, 19.3.537, 19.3.538, 19.3.539, 19.3.540, 19.3.541, 19.3.542, 19.3.543, 19.3.544, 19.3.545, 19.3.546, 19.3.547, 19.3.548, 19.3.549, 19.3.550, 19.3.551, 19.3.552, 19.3.553, 19.3.554, 19.3.555, 19.3.556, 19.3.557, 19.3.558, 19.3.559, 19.3.560, 19.3.561, 19.3.562, 19.3.563, 19.3.564, 19.3.565, 19.3.566, 19.3.567, 19.3.568, 19.3.569, 19.3.570, 19.3.571, 19.3.572, 19.3.573, 19.3.574, 19.3.575, 19.3.576, 19.3.577, 19.3.578, 19.3.579, 19.3.580, 19.3.581, 19.3.582, 19.3.583, 19.3.584, 19.3.585, 19.3.586, 19.3.587, 19.3.588, 19.3.589, 19.3.590, 19.3.591, 19.3.592, 19.3.593, 19.3.594, 19.3.595, 19.3.596, 19.3.597, 19.3.598, 19.3.599, 19.3.600, 19.3.601, 19.3.602, 19.3.603, 19.3.604, 19.3.605, 19.3.606, 19.3.607, 19.3.608, 19.3.609, 19.3.610, 19.3.611, 19.3.612, 19.3.613, 19.3.614, 19.3.615, 19.3.616, 19.3.617, 19.3.618, 19.3.619, 19.3.620, 19.3.621, 19.3.622, 19.3.623, 19.3.624, 19.3.625, 19.3.626, 19.3.627, 19.3.628, 19.3.629, 19.3.630, 19.3.631, 19.3.632, 19.3.633, 19.3.634, 19.3.635, 19.3.636, 19.3.637, 19.3.638, 19.3.639, 19.3.640, 19.3.641, 19.3.642, 19.3.643, 19.3.644, 19.3.645, 19.3.646, 19.3.647, 19.3.648, 19.3.649, 19.3.650, 19.3.651, 19.3.652, 19.3.653, 19.3.654, 19.3.655, 19.3.656, 19.3.657, 19.3.658, 19.3.659, 19.3.660, 19.3.661, 19.3.662, 19.3.663, 19.3.664, 19.3.665, 19.3.666, 19.3.667, 19.3.668, 19.3.669, 19.3.670, 19.3.671, 19.3.672, 19.3.673, 19.3.674, 19.3.675, 19.3.676, 19.3.677, 19.3.678, 19.3.679, 19.3.680, 19.3.681, 19.3.682, 19.3.683, 19.3.684, 19.3.685, 19.3.686, 19.3.687, 19.3.688, 19.3.689, 19.3.690, 19.3.691, 19.3.692, 19.3.693, 19.3.694, 19.3.695, 19.3.696, 19.3.697, 19.3.698, 19.3.699, 19.3.700, 19.3.701, 19.3.702, 19.3.703, 19.3.704, 19.3.705, 19.3.706, 19.3.707, 19.3.708, 19.3.709, 19.3.710, 19.3.711, 19.3.712, 19.3.713, 19.3.714, 19.3.715, 19.3.716, 19.3.717, 19.3.718, 19.3.719, 19.3.720, 19.3.721, 19.3.722, 19.3.723, 19.3.724, 19.3.725, 19.3.726, 19.3.727, 19.3.728, 19.3.729, 19.3.730, 19.3.731, 19.3.732, 19.3.733, 19.3.734, 19.3.735, 19.3.736, 19.3.737, 19.3.738, 19.3.739, 19.3.740, 19.3.741, 19.3.742, 19.3.743, 19.3.744, 19.3.745, 19.3.746, 19.3.747, 19.3.748, 19.3.749, 19.3.750, 19.3.751, 19.3.752, 19.3.753, 19.3.754, 19.3.755, 19.3.756, 19.3.757, 19.3.758, 19.3.759, 19.3.760, 19.3.761, 19.3.762, 19.3.763, 19.3.764, 19.3.765, 19.3.766, 19.3.767, 19.3.768, 19.3.769, 19.3.770, 19.3.771, 19.3.772, 19.3.773, 19.3.774, 19.3.775, 19.3.776, 19.3.777, 19.3.778, 19.3.779, 19.3.780, 19.3.781, 19.3.782, 19.3.783, 19.3.784, 19.3.785, 19.3.786, 19.3.787, 19.3.788, 19.3.789, 19.3.790, 19.3.791, 19.3.792, 19.3.793, 19.3.794, 19.3.795, 19.3.796, 19.3.797, 19.3.798, 19.3.799, 19.3.800, 19.3.801, 19.3.802, 19.3.803, 19.3.804, 19.3.805, 19.3.806, 19.3.807, 19.3.808, 19.3.809, 19.3.810, 19.3.811, 19.3.812, 19.3.813, 19.3.814, 19.3.815, 19.3.816, 19.3.817, 19.3.818, 19.3.819, 19.3.820, 19.3.821, 19.3.822, 19.3.823, 19.3.824, 19.3.825, 19.3.826, 19.3.827, 19.3.828, 19.3.829, 19.3.830, 19.3.831, 19.3.832, 19.3.833, 19.3.834, 19.3.835, 19.3.836, 19.3.837, 19.3.838, 19.3.839, 19.3.840, 19.3.841, 19.3.842, 19.3.843, 19.3.844, 19.3.845, 19.3.846, 19.3.847, 19.3.848, 19.3.849, 19.3.850, 19.3.851, 19.3.852, 19.3.853, 19.3.854, 19.3.855, 19.3.856, 19.3.857, 19.3.858, 19.3.859, 19.3.860, 19.3.861, 19.3.862, 19.3.863, 19.3.864, 19.3.865, 19.3.866, 19.3.867, 19.3.868, 19.3.869, 19.3.870, 19.3.871, 19.3.872, 19.3.873, 19.3.874, 19.3.875, 19.3.876, 19.3.877, 19.3.878, 19.3.879, 19.3.880, 19.3.881, 19.3.882, 19.3.883, 19.3.884, 19.3.885, 19.3.886, 19.3.887, 19.3.888, 19.3.889, 19.3.890, 19.3.891, 19.3.892, 19.3.893, 19.3.894, 19.3.895, 19.3.896, 19.3.897, 19.3.898, 19.3.899, 19.3.900, 19.3.901, 19.3.902, 19.3.903, 19.3.904, 19.3.905, 19.3.906, 19.3.907, 19.3.908, 19.3.909, 19.3.910, 19.3.911, 19.3.912, 19.3.913, 19.3.914, 19.3.915, 19.3.916, 19.3.917, 19.3.918, 19.3.919, 19.3.920, 19.3.921, 19.3.922, 19.3.923, 19.3.924, 19.3.925, 19.3.926, 19.3.927, 19.3.928, 19.3.929, 19.3.930, 19.3.931, 19.3.932, 19.3.933, 19.3.934, 19.3.935, 19.3.936, 19.3.937, 19.3.938, 19.3.939, 19.3.940, 19.3.941, 19.3.942, 19.3.943, 19.3.944, 19.3.945, 19.3.946, 19.3.947, 19.3.948, 19.3.949, 19.3.950, 19.3.951, 19.3.952, 19.3.953, 19.3.954, 19.3.955, 19.3.956, 19.3.957, 19.3.958, 19.3.959, 19.3.960, 19.3.961, 19.3.962, 19.3.963, 19.3.964, 19.3.965, 19.3.966, 19.3.967, 19.3.968, 19.3.969, 19.3.970, 19.3.971, 19.3.972, 19.3.973, 19.3.974, 19.3.975, 19.3.976, 19.3.977, 19.3.978, 19.3.979, 19.3.980, 19.3.981, 19.3.982, 19.3.983, 19.3.984, 19.3.985, 19.3.986, 19.3.987, 19.3.988, 19.3.989, 19.3.990, 19.3.991, 19.3.992, 19.3.993, 19.3.994, 19.3.995, 19.3.996, 19.3.997, 19.3.998, 19.3.999, 19.4.0.0, 19.4.0.1, 19.4.0.2, 19.4.0.3, 19.4.0.4, 19.4.0.5, 19.4.0.6, 19.4.0.7, 19.4.0.8, 19.4.0.9, 19.4.0.10, 19.4.0.11, 19.4.0.12, 19.4.0.13, 19.4.0.14, 19.4.0.15, 19.4.0.16, 19.4.0.17, 19.4.0.18, 19.4.0.19, 19.4.0.20, 19.4.0.21, 19.4.0.22, 19.4.0.23, 19.4.0.24, 19.4.0.25, 19.4.0.26, 19.4.0.27, 19.4.0.28, 19.4.0.29, 19.4.0.30, 19.4.0.31, 19.4.0.32, 19.4.0.33, 19.4.0.34, 19.4.0.35, 19.4.0.36, 19.4.0.37, 19.4.0.38, 19.4.0.39, 19.4.0.40, 19.4.0.41, 19.4.0.42, 19.4.0.43, 19.4.0.44, 19.4.0.45, 19.4.0.46, 19.4.0.47, 19.4.0.48, 19.4.0.49, 19.4.0.50, 19.4.0.51, 19.4.0.52, 19.4.0.53, 19.4.0.54, 19.4.0.55, 19.4.0.56, 19.4.0.57, 19.4.0.58, 19.4.0.59, 19.4.0.60, 19.4.0.61, 19.4.0.62, 19.4.0.63, 19.4.0.64, 19.4.0.65, 19.4.0.66, 19.4.0.67, 19.4.0.68, 19.4.0.69, 19.4.0.70, 19.4.0.71, 19.4.0.72, 19.4.0.73, 19.4.0.74, 19.4.0.75, 19.4.0.76, 19.4.0.77, 19.4.0.78, 19.4.0.79, 19.4.0.80, 19.4.0.81, 19.4.0.82, 19.4.0.83, 19.4.0.84, 19.4.0.85, 19.4.0.86, 19.4.0.87, 19.4.0.88, 19.4.0.89, 19.4.0.90, 19.4.0.91, 19.4.0.92, 19.4.0.93, 19.4.0.94, 19.4.0.95, 19.4.0.96, 19.4.0.97, 19.4.0.98, 19.4.0.99, 19.4.0.100, 19.4.0.101, 19.4.0.102, 19.4.0.103, 19.4.0.104, 19.4.0.105, 19.4.0.106, 19.4.0.107, 19.4.0.108, 19.4.0.109, 19.4.0.110, 19.4.0.111, 19.4.0.112, 19.4.0.113, 19.4.0.114, 19.4.0.115, 19.4.0.116, 19.4.0.117, 19.4.0.118, 19.4.0.119, 19.4.0.120, 19.4.0.121, 19.4.0.122, 19.4.0.123, 19.4.0.124, 19.4.0.125, 19.4.0.126, 19.4.0.127, 19.4.0.128, 19.4.0.129, 19.4.0.130, 19.4.0.131, 19.4.0.132, 19.4.0.133, 19

erlang 官网下载: <http://www.erlang.org/downloads>

2 解压 erlang 安装包

```
[root@SnailClimb local]#tar -xvzf otp_src_19.3.tar.gz
```

3 删除 erlang 安装包

```
[root@SnailClimb local]#rm -rf otp_src_19.3.tar.gz
```

4 安装 erlang 的依赖工具

```
[root@SnailClimb local]#yum -y install make gcc gcc-c++ kernel-devel m4 ncurses-devel openssl-devel unixODBC-devel
```

5 进入erlang 安装包解压文件对 erlang 进行安装环境的配置

新建一个文件夹

```
[root@SnailClimb local]# mkdir erlang
```

对 erlang 进行安装环境的配置

```
[root@SnailClimb otp_src_19.3]#  
./configure --prefix=/usr/local/erlang --without-javac
```

6 编译安装

```
[root@SnailClimb otp_src_19.3]#  
make && make install
```

7 验证一下 erlang 是否安装成功了

```
[root@SnailClimb otp_src_19.3]# ./bin/erl
```

运行下面的语句输出"hello world"

```
io:format("hello world~n", []).
```

```
[root@SnailClimb otp_src_19.3]# ./bin/erl
Erlang/OTP 19 [erts-8.3] [source] [64-bit] [async-threads:10] [hipe] [kernel-poll:false]

Eshell V8.3 (abort with ^G)
1> io:format("hello world~n", []).
hello world
ok
2> █
```

大功告成，我们的 erlang 已经安装完成。

8 配置 erlang 环境变量

```
[root@SnailClimb etc]# vim profile
```

追加下列环境变量到文件末尾

```
#erlang
ERL_HOME=/usr/local/erlang
PATH=$ERL_HOME/bin:$PATH
export ERL_HOME PATH
```

运行下列命令使配置文件profile生效

```
[root@SnailClimb etc]# source /etc/profile
```

输入 erl 查看 erlang 环境变量是否配置正确

```
[root@SnailClimb etc]# erl
```

```
[root@SnailClimb etc]# erl
Erlang/OTP 19 [erts-8.3] [source] [64-bit] [async-threads:10] [hipe] [kernel-poll:false]

Eshell V8.3 (abort with ^G)
1> █
```

2.2 安装 RabbitMQ

1. 下载rpm

```
wget https://www.rabbitmq.com/releases/rabbitmq-server/v3.6.8/rabbitmq-server-3.6.8-1.el7.noarch.rpm
```


或者直接在官网下载

<https://www.rabbitmq.com/install-rpm.html>

2. 安装rpm

```
rpm --import https://www.rabbitmq.com/rabbitmq-release-signing-key.asc
```

紧接着执行：

```
yum install rabbitmq-server-3.6.8-1.el7.noarch.rpm
```

中途需要你输入"y"才能继续安装。

3 开启 web 管理插件

```
rabbitmq-plugins enable rabbitmq_management
```

4 设置开机启动

```
chkconfig rabbitmq-server on
```

4. 启动服务

```
service rabbitmq-server start
```

5. 查看服务状态

```
service rabbitmq-server status
```

6. 访问 RabbitMQ 控制台

浏览器访问：<http://你的ip地址:15672/>

默认用户名和密码： guest/guest;但是需要注意的是：guest用户只是被容许从localhost访问。官网文档描述如下：

```
“guest” user can only connect via localhost
```


解决远程访问 RabbitMQ 远程访问密码错误

新建用户并授权

```
[root@SnailClimb rabbitmq]# rabbitmqctl add_user root root
Creating user "root" ...
[root@SnailClimb rabbitmq]# rabbitmqctl set_user_tags root administrator

Setting tags for user "root" to [administrator] ...
[root@SnailClimb rabbitmq]#
[root@SnailClimb rabbitmq]# rabbitmqctl set_permissions -p / root ".*" ".*" ".*"
Setting permissions for user "root" in vhost "/" ...
```

再次访问:http://你的ip地址:15672/ ,输入用户名和密码： root root



User: root

Cluster: rabbit@SnailClimb ([change](#))

RabbitMQ 3.6.8, [Erlang R16B03-1](#)

Overview

Connections

Channels

Exchanges

Queues

Admin

Overview

Totals

Queued messages (chart: last minute) (?)

Currently idle

Message rates (chart: last minute) (?)

Currently idle

Global counts (?)

Connections: 0

Channels: 0

Exchanges: 8

Queues: 0

Consumers: 0

Node

Node: rabbit@SnailClimb (More about this node)

File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Rates mode	Info	Reset stats DB
<div>52</div> <div>1024 available</div>	<div>0</div> <div>829 available</div>	<div>322</div> <div>1048576 available</div>	<div>53MB</div> <div>735MB high watermark46MB low watermark</div>	<div>31GB</div>	basic	<div>Disc 1</div>	<div>Reset</div>

Reset stats on all nodes