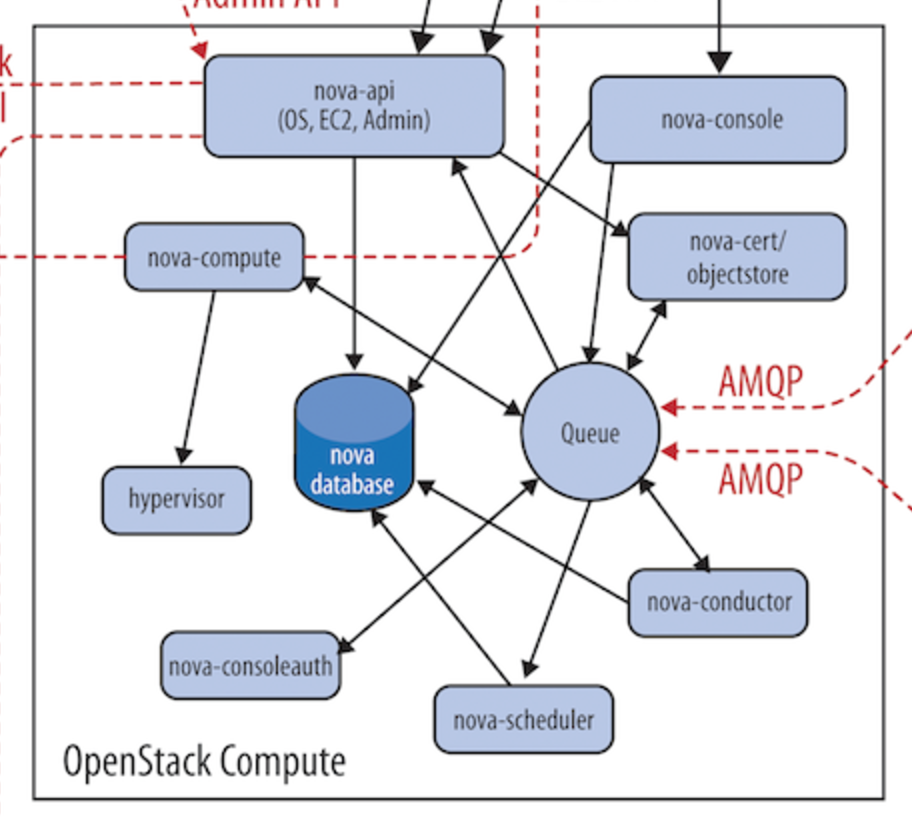
这是 OpenStack 实施经验分享系列的第 8 篇。

先来看张图：  
  
  
  
这是 Nova 的架构图，我们可以看到有两个组件处于架构的中心位置：数据库和Queue。数据库保存状态信息，而几乎所有的 nova-\* 服务都直接依赖于 Queue 实现服务之间的通信和调用。  
  
OpenStack 通常用 RabbitMQ 实现消息队列，几乎所有的 OpenStack 模块都会用到 RabbitMQ，如果 RabbitMQ 挂了，OpenStack 也就瘫了，可以说它是最重要的组件。  
  
本节我们就来讨论如何监控 RabbitMQ 的状态，介绍一个非常简单高效的方法。

**启用 RabbitMQ 管理 plugin**

默认安装中，我们只能用命令 rabbitmqctl 监控 RabbitMQ，比如：rabbitmqctl list\_queues，rabbitmqctl list\_exchanges 等子命令。这种方式不太直观，效率不高。  
  
好在 RabbitMQ 有一个管理 plugin，提供了图形管理界面，可以在运行 RabbitMQ 的节点（一般是控制节点）执行下面的命令启用。  
  
rabbitmq-plugins enable rabbitmq\_management

然后还需要创建一个 用户，用来登录管理控制台了。

rabbitmqctl add\_user  user\_admin  passwd\_admin

rabbitmqctl set\_user\_tags user\_admin administrator

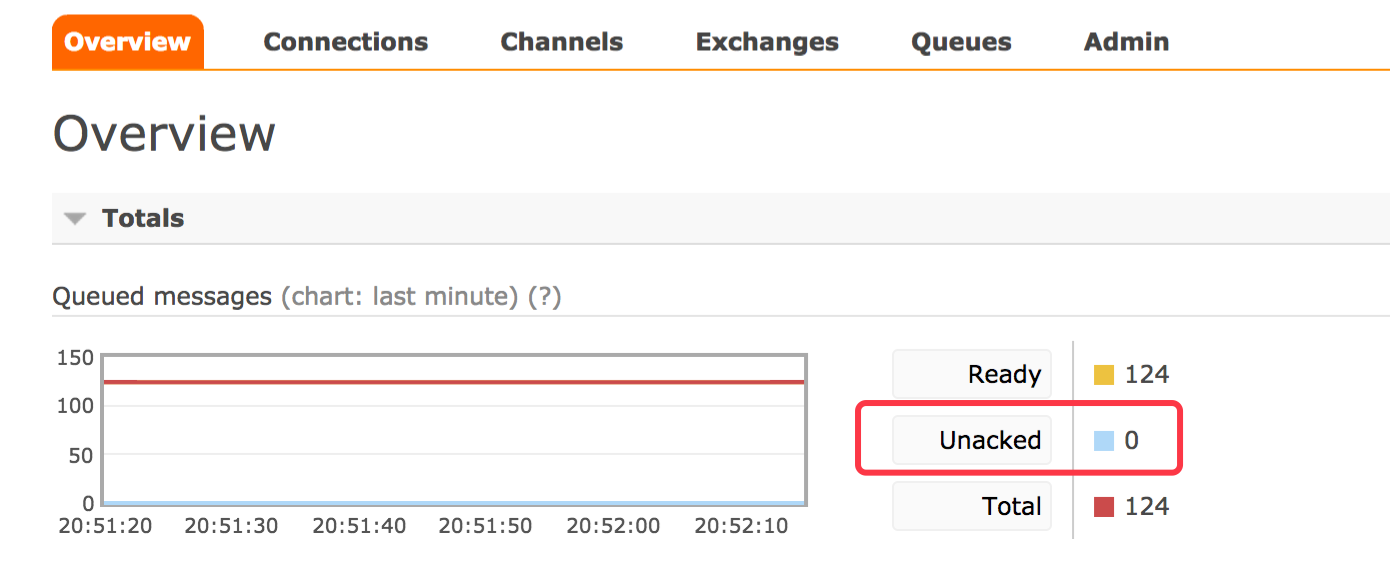
rabbitmqctl set\_permissions -p / user\_admin ".\*" ".\*" ".\*"

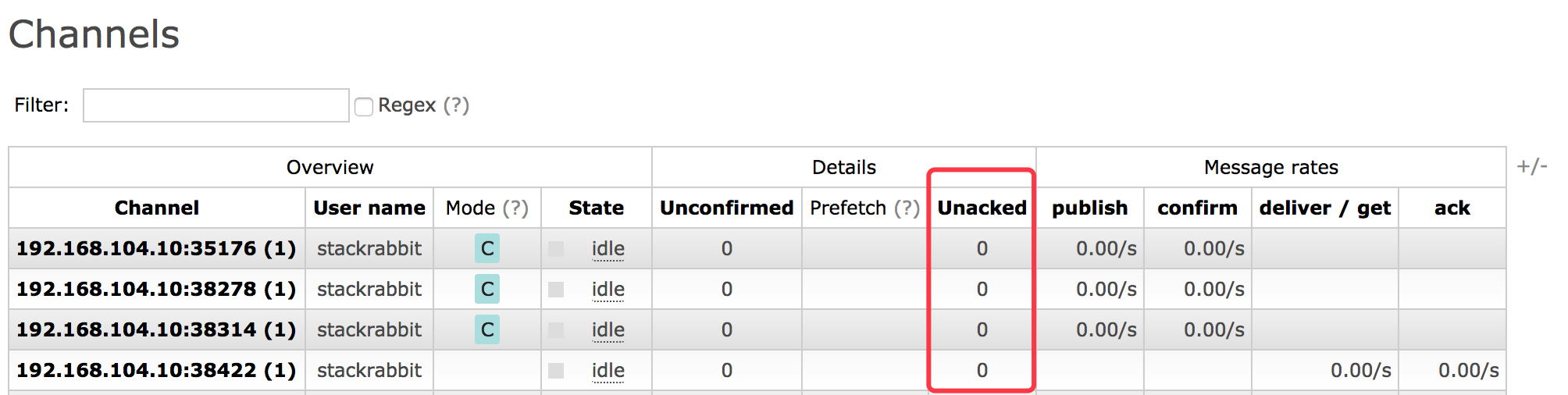
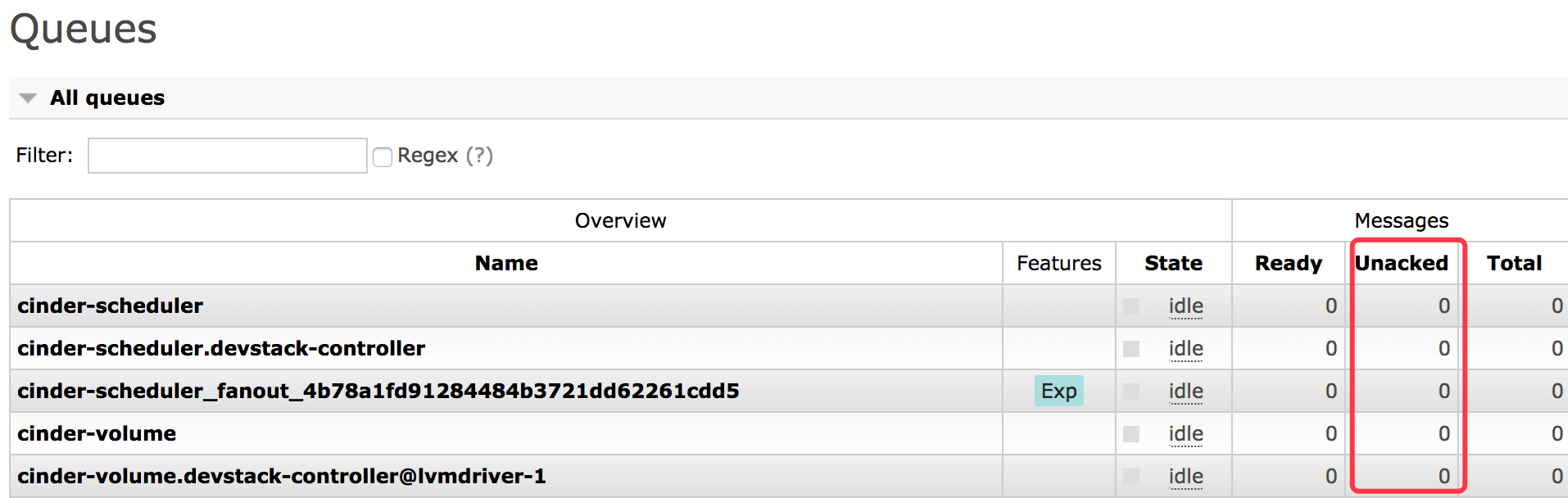
然后就可以用 user\_admin（密码 passwd\_admin）登录了，地址是

http://server-name:15672/



**最简单高效的监控方法**

Web 控制台会展示很多 RabbitMQ 信息，但最最重要的就一个：**Unacked Message。**这个数据会直接显示在登录之后的 Overview 标签中，第一眼就能看到。  
  
  


Unacked Message 指的是还没有被处理的消息。正常情况下，这个值应该为 0。如果这个值不是 0，并且持续增长，那你就得注意了，这意味着 RabbitMQ 出现了问题，队列开始积压，消息开始堆积，是一个严重的信号。  
  
接下来怎么办呢？  
  
这个时候就可以点开 Overview 后面的标签，查看到底消息是在哪个或者哪些 Connection，Channel，Exchange，Queues 中堆积，进而分析问题的根源并解决。  
  
  
  


**一个真实案例**

1. 客户的 OpenStack 在正常运行了一个月后突然挂了。  
  
2. 日志分析发现 nova，neutron 等模块都报告找不到相关的 queue。因为多个模块的日志都指向 RabbitMQ，看来 RabbitMQ 有最大嫌疑。  
  
3. RabbitMQ 日志中 Error 已经在持续刷屏，但信息很笼统。这时 RabbitMQ 已经处于无法工作的状态，只能重启 RabbitMQ。  
  
4. RabbitMQ 重启后，OpenStack 自动恢复。  
  
5. 打开 RabbitMQ Web 控制台，发现 Unacked Message > 0。  
  
6. 观察一段时间，发现 Unacked Message 以固定的速度持续增长。  
  
7. 定位 Message 增长的原因，发现均来自 Ceilometer 相关的 Queue。  
  
8. 检查 Ceilometer，发现了一个配置错误，导致 Ceilometer 发送到 Queue 的数据没有被处理。  
  
9. 修改配置，重启 Ceilometer，Unacked Message 开始下降，最后保持为 0。  
  
这个问题就像内存泄漏一样，Unacked Message 逐渐积累，最终压跨了整个 OpenStack。  
  
好了，以上就是今天的内容，下一节我们继续分享实施经验。