RabbitMQ集群简介

第1章引言

1.1目的

实现OpenStack RabbitMQ消息高可用性，实现MQ的负载均衡，缓解MQ压力，提高性能

1.2说明

> Rabbitmq的集群是依附于erlang的集群来工作的，所以必须先构建起erlang的集群景象。Erlang的集群中各节点是经由过程一个magic cookie来实现的，这个cookie存放在/var/lib/rabbitmq/.erlang.cookie中(本人采用的是yum
> 安装)，文件是400的权限。所以必须包管各节点cookie对峙一致，不然节点之间就无法通信

1.3 MQ

> MQ全称为Message Queue,消息队列（MQ）是一种应用程序对应用程序的通信方法。应用程序通过读写出入队列的消息（针对应用程序的数据）来通信，而无需专用连接来链接它们。消息传递指的是程序之间通过在消息中发送数据进行通信，而不是通过直接调用彼此来通信，直接调用通常是用于诸如远程过程调用的技术。排队指的是应用程序通过队列来通信。队列的使用除去了接收和发送应用程序同时执行的要求。其中较为成熟的MQ产品有IBM WEBSPHERE MQ

1.4概念

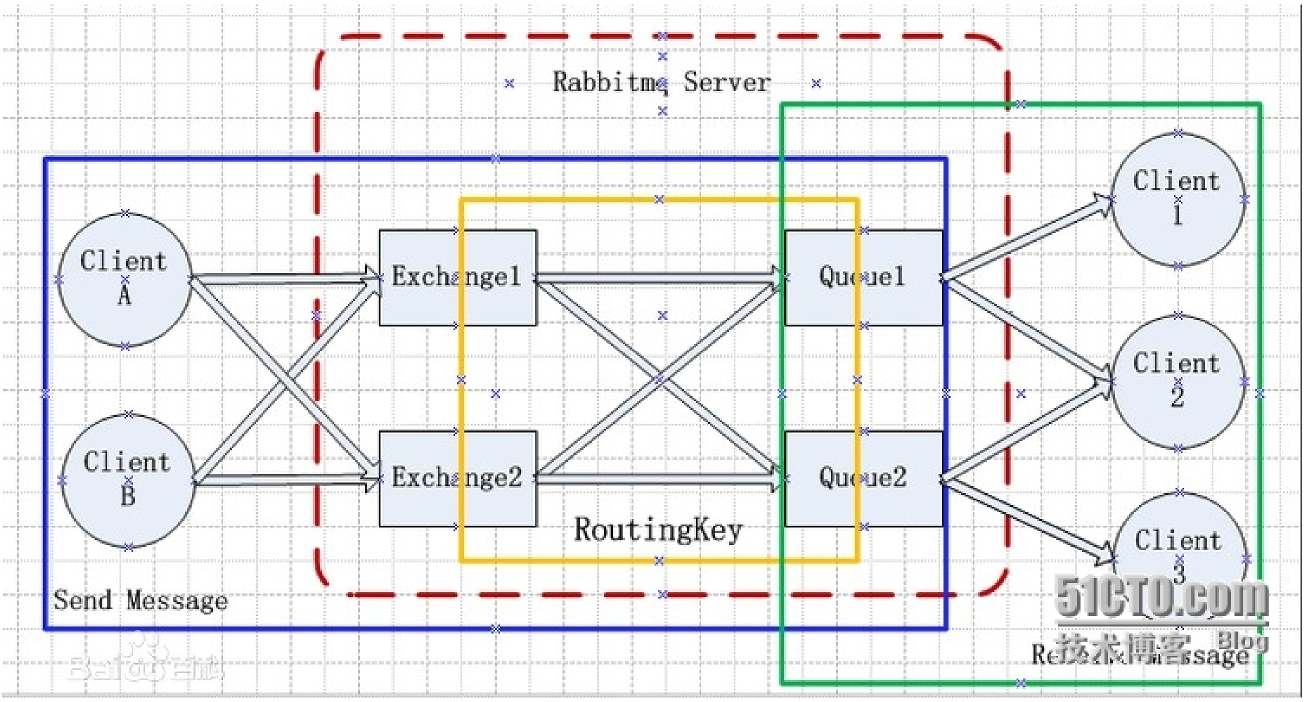
Broker：简单来说就是消息队列服务器实体。
Exchange：消息交换机，它指定消息按什么规则，路由到哪个队列。
Queue：消息队列载体，每个消息都会被投入到一个或多个队列。
Binding：绑定，它的作用就是把exchange和queue按照路由规则绑定起来。
Routing Key：路由关键字，exchange根据这个关键字进行消息投递。
vhost：虚拟主机，一个broker里可以开设多个vhost，用作不同用户的权限分离。
producer：消息生产者，就是投递消息的程序。
consumer：消息消费者，就是接受消息的程序。
channel：消息通道，在客户端的每个连接里，可建立多个channel，每个channel代表一个会话任务。

1.5MQ特点

> MQ是消费-生产者模型的一个典型的代表，一端往消息队列中不断写入消息，而另一端则可以读取或者订阅队列中的消息。MQ和JMS类似，但不同的是JMS是SUN JAVA消息中间件服务的一个标准和API定义，而MQ则是遵循了AMQP协议的具体实现和产品。

在项目中，将一些无需即时返回且耗时的操作提取出来，进行了异步处理，而这种异步处理的方式大大的节省了服务器的请求响应时间，从而提高了系统的吞吐量。

1.6工作流程



消息队列的使用过程大概如下：

（1）客户端连接到消息队列服务器，打开一个channel。
（2）客户端声明一个exchange，并设置相关属性。
（3）客户端声明一个queue，并设置相关属性。
（4）客户端使用routing key，在exchange和queue之间建立好绑定关系。
（5）客户端投递消息到exchange。

Exchange接收到消息后，就根据消息的key和已经设置的binding，进行消息路由，将消息投递到一个或多个队列里。

Exchange也有几个类型，完全根据key进行投递的叫做Direct交换机，例如，绑定时设置了routing key为”abc”，那么客户端提交的消息，只有设置了key为”abc”的才会投递到队列。对key进行模式匹配后进行投递的叫做Topic交换机，符号”#”匹配一个或多个词，符号”*”匹配正好一个词。例如”abc.#”匹配”abc.def.ghi”，”abc.*”只匹配”abc.def”。还有一种不需要key的，叫做Fanout交换机，它采取广播模式，一个消息进来时，投递到与该交换机绑定的所有队列。

RabbitMQ支持消息的持久化，也就是数据写在磁盘上，为了数据安全考虑，我想大多数用户都会选择持久化。消息队列持久化包括3个部分：
（1）exchange持久化，在声明时指定durable => 1
（2）queue持久化，在声明时指定durable => 1
（3）消息持久化，在投递时指定delivery_mode=> 2（1是非持久化）

如果exchange和queue都是持久化的，那么它们之间的binding也是持久化的。如果exchange和queue两者之间有一个持久化，一个非持久化，就不允许建立绑定。

1.7系统环境

角色	主机名	IP地址	系统版本
MQ Node	rabbit@athController	192.168.8.180	CentOS6.5 + RabbitMQ3.1.5
MQ Node	rabbit@athBackup87	192.168.8.87	CentOS6.5 + RabbitMQ3.1.5

MQ Node	rabbit@athBackup53	192.168.8.53	CentOS6.5 + RabbitMQ3.1.5

第2章 RabbitMQ部署

本环境用于测试，采用三台节点作为MQ的集群，系统OS采用的是CentOS 6.5 x64,该技术服务于OpenStack，卸载CenOS自带MQ产品QPID,采用的是RabbitMQ作为我们的首选高级消息队列服务

2.1系统环境基本配置

1）install epel yum

a) wgethttp://mirrors.yun-idc.com/epel/6/i386/epel-release-6-8.noarch.rpm

b)rpm -vihepel-release-6-8.noarch.rpm

2）install rabbitMQ

a)yum -y install rabbitmq-server

3）Configure /etc/hosts （这步需要配置）

a)192.168.8.180athController.8.180.abs.com.cnathController

b)192.168.8.53athBackup53.abs.com.cnathBackup53

c)192.168.8.87athBackup87.abs.com.cnathBackup87

4）Configure /etc/sysconfig/network setting(Setting hostname, login other node Configure)

a)sed -i 's/HOSTNAME=.*/HOSTNAME=hostname/' /etc/sysconf/network

5）Disabled selinux Configure and iptables

a)sed -i's/SELINUX=.*/SELINUX=disabled/' /etc/selinux/config

b)service iptables stop

c)chkconfig –level 123456iptables off

6）以上操作，在三台节点同时执行操作

2.2 RabbitMA配置

1)Configure /etc/rabbitmq/rabbitmq.config

```
[root@athControllerrabbitmq]# cat rabbitmq.config
% Thisfile managed by Puppet
%Template Path: rabbitmq/templates/rabbitmq.config
[
{rabbit,[
{default_user,<<"guest">>},
{default_pass,<<"guest">>}
]},
{kernel,[

]}
].
% EOF
[root@athControllerrabbitmq]#
```

2)Configure/etc/rabbitmq/rabbitmq-env.config

```
[root@athController rabbitmq]# cat/opt/rabbitmq-env.conf
RABBITMQ_NODE_PORT=5672
    [root@athController rabbitmq]#
```

3)Create .erlang.cookie,consistent

```
 [root@athControllerrabbitmq]# vim /var/lib/rabbitmq/.erlang.cookie
 YAGISQRAHKOFCZMWRFMT
 [root@athController rabbitmq]# chmod 400 /var/lib/rabbitmq/.erlang.cookie
 [root@athController rabbitmq]#chown rabbitmq:rabbitmq/var/lib/rabbitmq/.erlang.cookie
```

4)Restart rabitmq-server

a)Service rabbitmq-server restart

5)Check RabbitMQ server and Addautostart

```
[root@athControllernova]# netstat -tnpl|grep 5672
tcp0 0 0.0.0.0:15672 0.0.0.0:*LISTEN 26946/beam.smp
tcp0 0 0.0.0.0:55672 0.0.0.0:*LISTEN 26946/beam.smp
tcp0 0 0.0.0.0:5672 0.0.0.0:*LISTEN 26946/beam.smp
   [root@athController nova]#chkconfig –level 123456 rabbitmq-server on
```

6)三台节点同时配置并启动MQ服务

2.3 RabbitMQ集群配置

1）主节点配置

```
[root@athControllernova] rabbitmqctl stop_app
[root@athController nova] rabbitmqctl reset

[root@athController nova]
/usr/lib/rabbitmq/bin/rabbitmq-plugins enable rabbitmq_management
    [root@athController nova]
     /usr/lib/rabbitmq/bin/rabbitmq-plugins enable rabbitmq_management_agent
[root@athController nova] rabbitmqctl start_app
```

2）节点配置

```
[root@athControllerrabbitmq] rabbitmqctl stop_app
[root@athControllerrabbitmq] rabbitmqctl reset
[root@athControllerrabbitmq] rabbitmqctl start_app
```

```
[root@athControllerrabbitmq]
/usr/lib/rabbitmq/bin/rabbitmq-pluginsenable rabbitmq_management
[root@athControllerrabbitmq]
    /usr/lib/rabbitmq/bin/rabbitmq-plugins enable rabbitmq_management_agent
   [root@athController rabbitmq] rabbitmqctl join_cluster--ram athController
[root@athControllernova] rabbitmqctl start_app
```

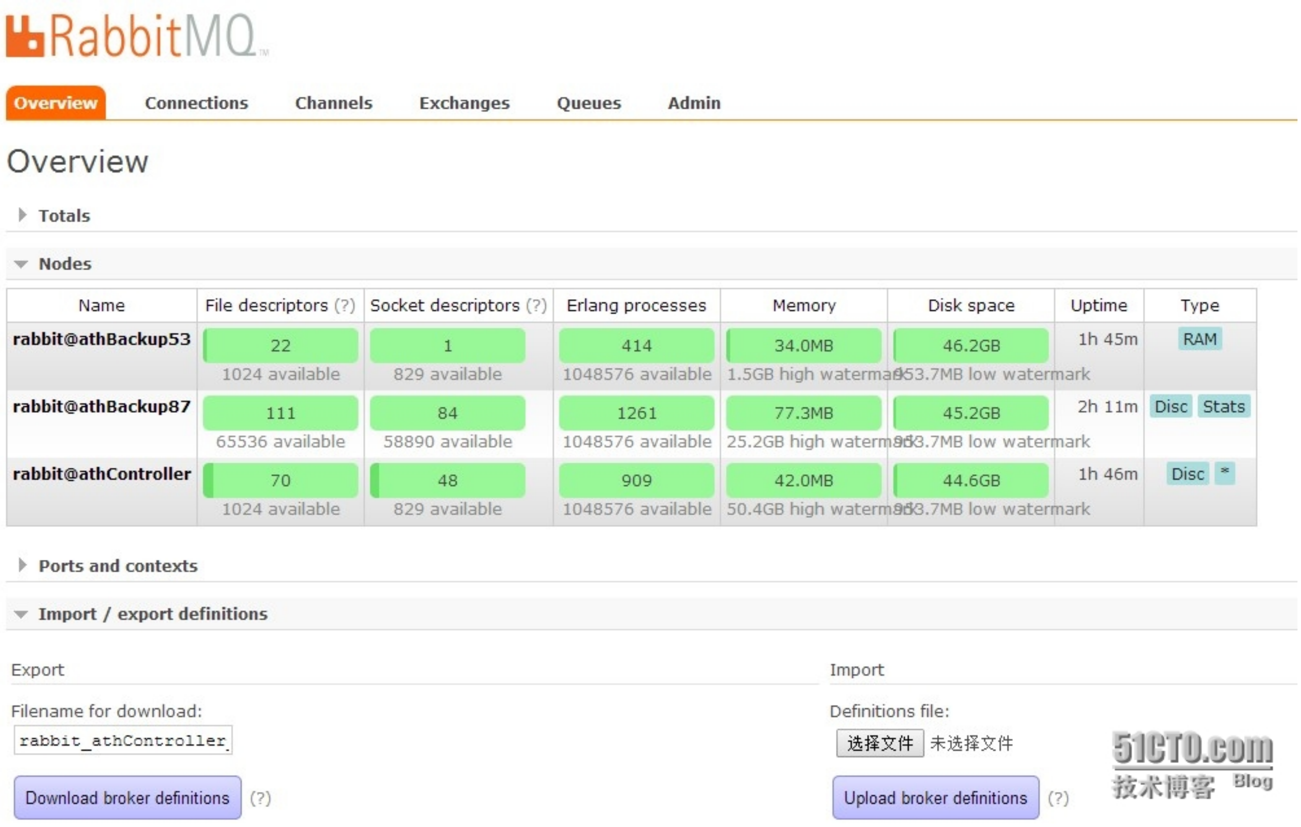
3)添加RabbitMQ用户(三台机器同时配置)

```
rabbitmqctl add_user username guest
 rabbitmqctl change_password guest guest
```

4) RabbitMQ镜像设置

```
rabbitmqctl set_policy ha-all"^" '{"ha-mode":"all"}'
```

3)浏览RabbitMQ WEB监控



Overview

Totals

Nodes

Name	File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Uptime	Type
rabbit@athBackup53	22 1024 available	1 829 available	414 1048576 available	34.0MB 1.5GB high waterma	46.2GB 953.7MB low watermark	1h 45m	RAM
rabbit@athBackup87	111 65536 available	84 58890 available	1261 1048576 available	77.3MB 25.2GB high waterm	45.2GB 953.7MB low watermark	2h 11m	Disc Stats
rabbit@athController	70 1024 available	48 829 available	909 1048576 available	42.0MB 50.4GB high waterm	44.6GB 953.7MB low watermark	1h 46m	Disc *

Ports and contexts

Import / export definitions

Export

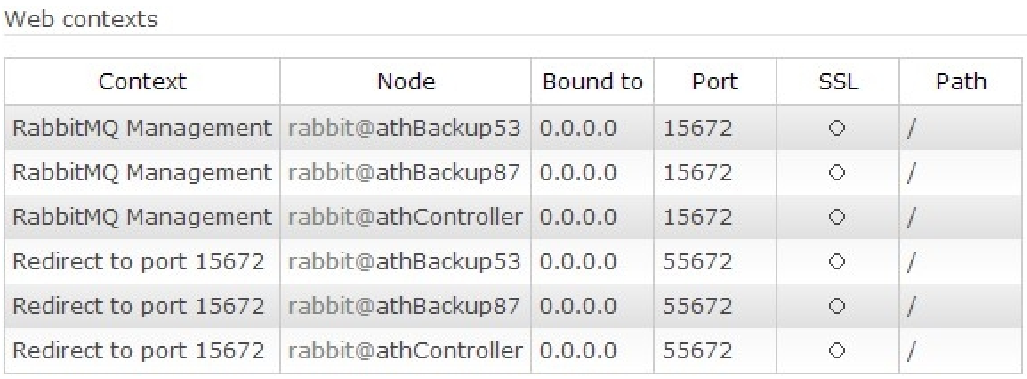
Filename for download:
rabbit_athController

Download broker definitions (?)

Import

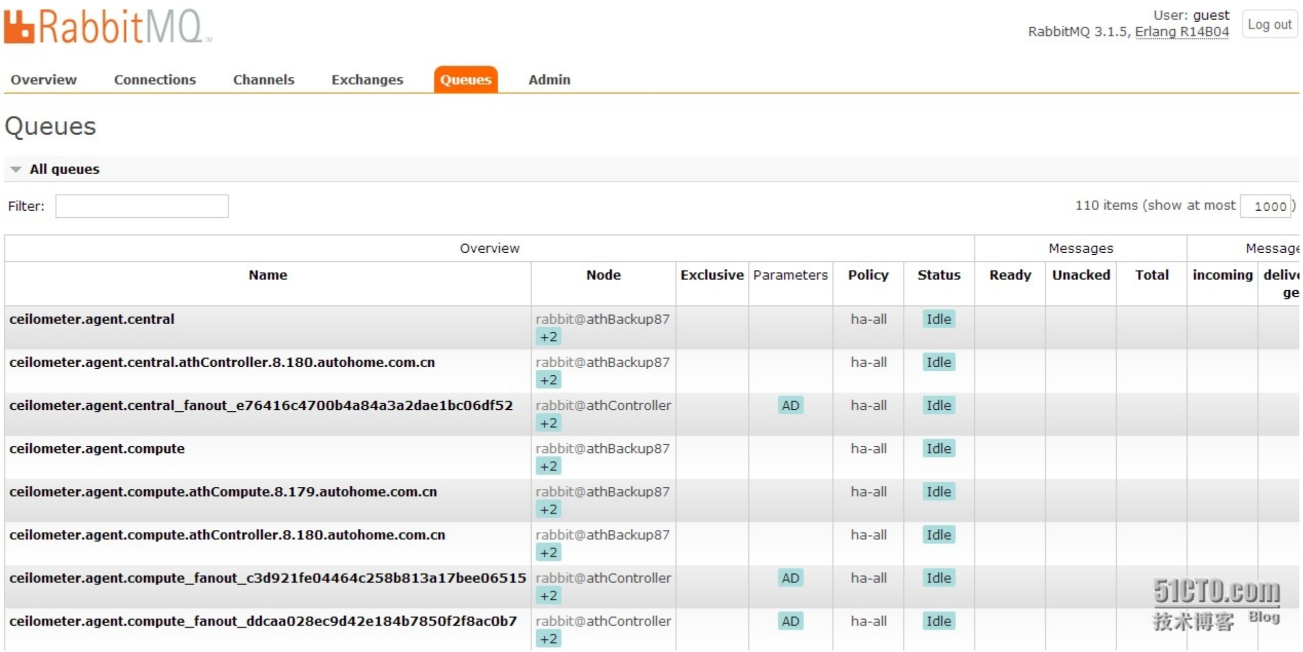
Definitions file:
选择文件 未选择文件

Upload broker definitions (?)



Web contexts

Context	Node	Bound to	Port	SSL	Path
RabbitMQ Management	rabbit@athBackup53	0.0.0.0	15672	○	/
RabbitMQ Management	rabbit@athBackup87	0.0.0.0	15672	○	/
RabbitMQ Management	rabbit@athController	0.0.0.0	15672	○	/
Redirect to port 15672	rabbit@athBackup53	0.0.0.0	55672	○	/
Redirect to port 15672	rabbit@athBackup87	0.0.0.0	55672	○	/
Redirect to port 15672	rabbit@athController	0.0.0.0	55672	○	/



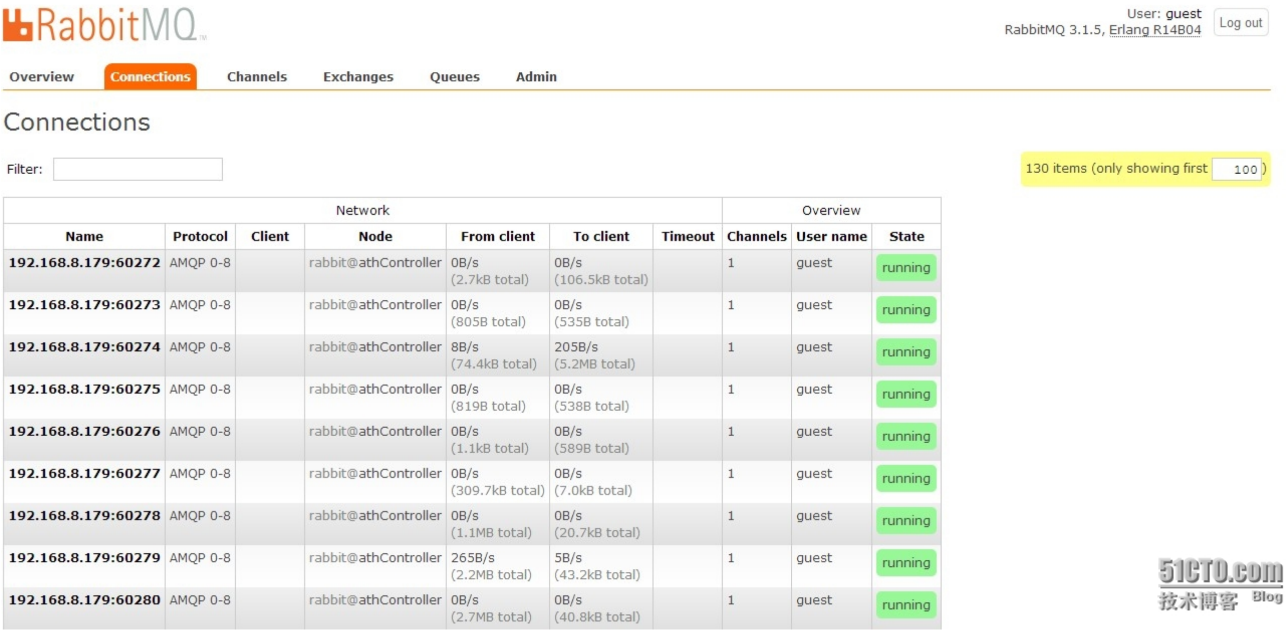
Queues

All queues

Filter:

110 items (show at most 1000)

Overview						Messages			Message	
Name	**Node**	**Exclusive**	**Parameters**	**Policy**	**Status**	**Ready**	**Unacked**	**Total**	**incoming**	**delive ge**
ceilometer.agent.central	rabbit@athBackup87 +2			ha-all	Idle					
ceilometer.agent.central.athController.8.180.autohome.com.cn	rabbit@athBackup87 +2			ha-all	Idle					
ceilometer.agent.central_fanout_e76416c4700b4a84a3a2dae1bc06df52	rabbit@athController +2		AD	ha-all	Idle					
ceilometer.agent.compute	rabbit@athBackup87 +2			ha-all	Idle					
ceilometer.agent.compute.athCompute.8.179.autohome.com.cn	rabbit@athBackup87 +2			ha-all	Idle					
ceilometer.agent.compute.athController.8.180.autohome.com.cn	rabbit@athBackup87 +2			ha-all	Idle					
ceilometer.agent.compute_fanout_c3d921fe04464c258b813a17bee06515	rabbit@athController +2		AD	ha-all	Idle					
ceilometer.agent.compute_fanout_ddcaa028ec9d42e184b7850f2f8ac0b7	rabbit@athController +2		AD	ha-all	Idle					

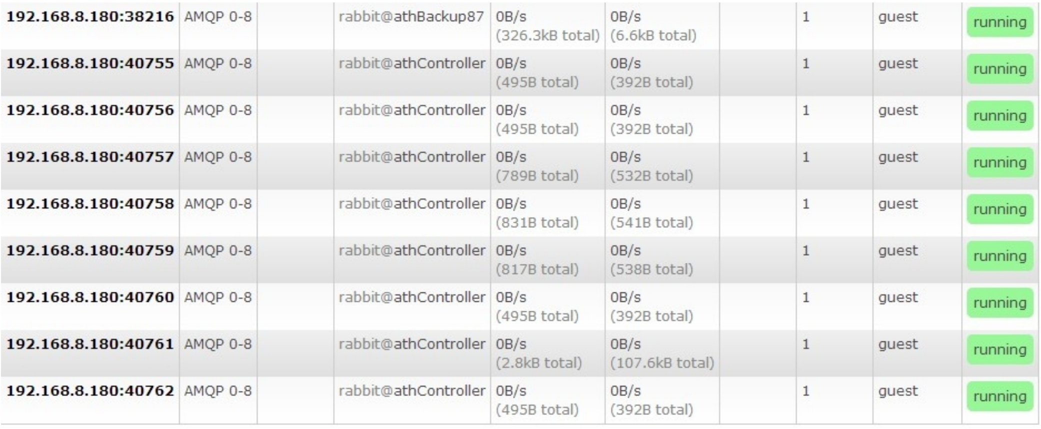


Connections

Filter:

130 items (only showing first 100)

Network							Overview		
Name	**Protocol**	**Client**	**Node**	**From client**	**To client**	**Timeout**	**Channels**	**User name**	**State**
192.168.8.179:60272	AMQP 0-8		rabbit@athController	0B/s (2.7kB total)	0B/s (106.5kB total)		1	guest	running
192.168.8.179:60273	AMQP 0-8		rabbit@athController	0B/s (805B total)	0B/s (535B total)		1	guest	running
192.168.8.179:60274	AMQP 0-8		rabbit@athController	8B/s (74.4kB total)	205B/s (5.2MB total)		1	guest	running
192.168.8.179:60275	AMQP 0-8		rabbit@athController	0B/s (819B total)	0B/s (538B total)		1	guest	running
192.168.8.179:60276	AMQP 0-8		rabbit@athController	0B/s (1.1kB total)	0B/s (589B total)		1	guest	running
192.168.8.179:60277	AMQP 0-8		rabbit@athController	0B/s (309.7kB total)	0B/s (7.0kB total)		1	guest	running
192.168.8.179:60278	AMQP 0-8		rabbit@athController	0B/s (1.1MB total)	0B/s (20.7kB total)		1	guest	running
192.168.8.179:60279	AMQP 0-8		rabbit@athController	265B/s (2.2MB total)	5B/s (43.2kB total)		1	guest	running
192.168.8.179:60280	AMQP 0-8		rabbit@athController	0B/s (2.7MB total)	0B/s (40.8kB total)		1	guest	running



192.168.8.180:38216	AMQP 0-8		rabbit@athBackup87	0B/s (326.3kB total)	0B/s (6.6kB total)		1	guest	running
192.168.8.180:40755	AMQP 0-8		rabbit@athController	0B/s (495B total)	0B/s (392B total)		1	guest	running
192.168.8.180:40756	AMQP 0-8		rabbit@athController	0B/s (495B total)	0B/s (392B total)		1	guest	running
192.168.8.180:40757	AMQP 0-8		rabbit@athController	0B/s (789B total)	0B/s (532B total)		1	guest	running
192.168.8.180:40758	AMQP 0-8		rabbit@athController	0B/s (831B total)	0B/s (541B total)		1	guest	running
192.168.8.180:40759	AMQP 0-8		rabbit@athController	0B/s (817B total)	0B/s (538B total)		1	guest	running
192.168.8.180:40760	AMQP 0-8		rabbit@athController	0B/s (495B total)	0B/s (392B total)		1	guest	running
192.168.8.180:40761	AMQP 0-8		rabbit@athController	0B/s (2.8kB total)	0B/s (107.6kB total)		1	guest	running
192.168.8.180:40762	AMQP 0-8		rabbit@athController	0B/s (495B total)	0B/s (392B total)		1	guest	running

第3章 RabbitMQ集群验证

3.1 Nova配置MQ HA

1）NovaRabbitMQ配置

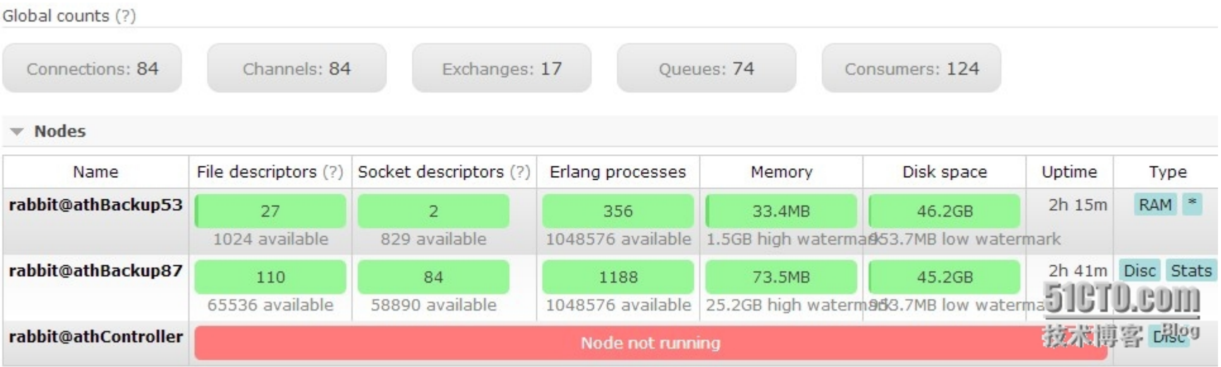
```
[root@athControllernova]# cat /etc/nova/nova.conf|grep rabbit
rabbit_host=192.168.8.180
rabbit_port=5672
rabbit_hosts=192.168.8.180:5672,192.168.8.53:5672,192.168.8.87:5672
rabbit_use_ssl=False
rabbit_userid=guest
rabbit_password=guest
rabbit_virtual_host=/
rabbit_retry_interval=1
rabbit_retry_backoff=2
rabbit_max_retries=0
rabbit_ha_queues=True
    [root@athController nova]#
```

3.2 RabbitMQ HA验证

2）stop RabbitMQ集群

```
a)[root@athController nova]#/etc/init.d/rabbitmq-server stop
b)Stopping rabbitmq-server:rabbitmq-server.
c)[root@athController nova]#
```

3）RabbitMQ监控报警



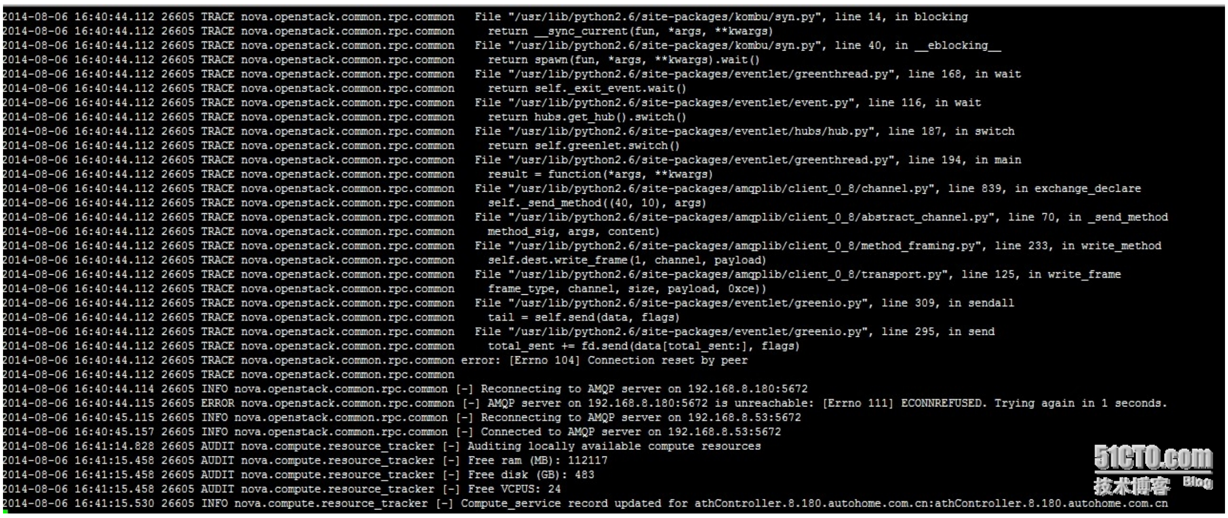
Global counts (?)

Connections: 84 | Channels: 84 | Exchanges: 17 | Queues: 74 | Consumers: 124

Nodes

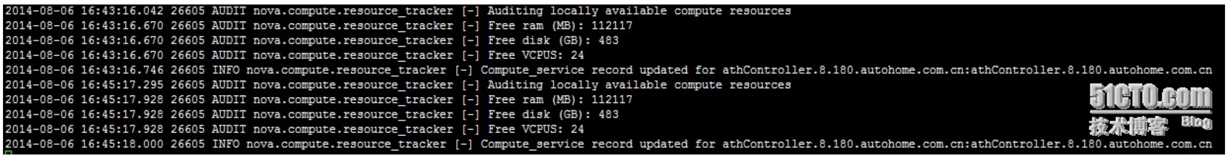
Name	File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Uptime	Type
rabbit@athBackup53	27 1024 available	2 829 available	356 1048576 available	33.4MB 1.5GB high watermark	46.2GB 953.7MB low watermark	2h 15m	RAM *
rabbit@athBackup87	110 65536 available	84 58890 available	1188 1048576 available	73.5MB 25.2GB high waterm	45.2GB 953.7MB low waterma	2h 41m	Disc Stats
rabbit@athController	Node not running						Disc

4）Nova错误日志



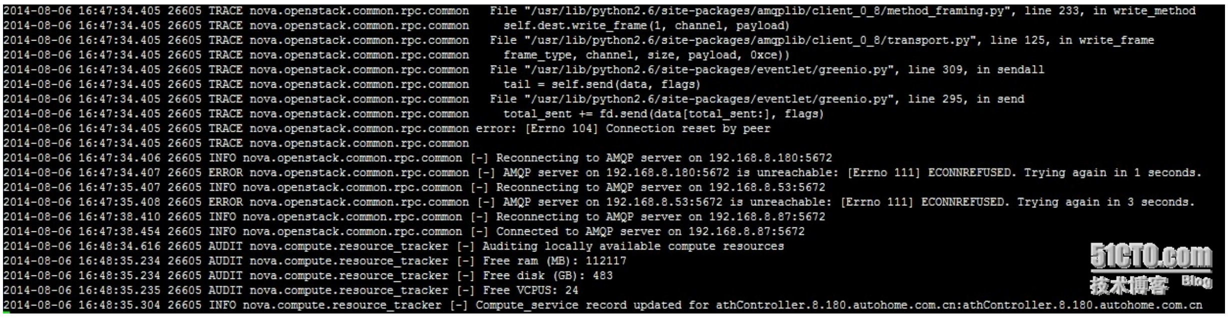
```
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/kombu/syn.py", line 14, in blocking
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common     return __sync_current(fun, *args, **kwargs)
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/kombu/syn.py", line 40, in __eblocking__
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common     return spawn(fun, *args, **kwargs).wait()
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/eventlet/greenthread.py", line 168, in wait
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common     return self._exit_event.wait()
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/eventlet/event.py", line 116, in wait
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common     return hubs.get_hub().switch()
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/eventlet/hubs/hub.py", line 187, in switch
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common     return self.greenlet.switch()
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/eventlet/greenthread.py", line 194, in main
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common     result = function(*args, **kwargs)
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/amqplib/client_0_8/channel.py", line 839, in exchange_declare
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common     self._send_method((40, 10), args)
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/amqplib/client_0_8/abstract_channel.py", line 70, in _send_method
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common     method_sig, args, content)
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/amqplib/client_0_8/method_framing.py", line 233, in write_method
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common     self.dest.write_frame(1, channel, payload)
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/amqplib/client_0_8/transport.py", line 125, in write_frame
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common     frame_type, channel, size, payload, 0xce))
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/eventlet/greenio.py", line 309, in sendall
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common     tail = self.send(data, flags)
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/eventlet/greenio.py", line 295, in send
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common     total_sent += fd.send(data[total_sent:], flags)
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common error: [Errno 104] Connection reset by peer
2014-08-06 16:40:44.112 26605 TRACE nova.openstack.common.rpc.common
2014-08-06 16:40:44.114 26605 INFO nova.openstack.common.rpc.common [-] Reconnecting to AMQP server on 192.168.8.180:5672
2014-08-06 16:40:44.115 26605 ERROR nova.openstack.common.rpc.common [-] AMQP server on 192.168.8.180:5672 is unreachable: [Errno 111] ECONNREFUSED. Trying again in 1 seconds.
2014-08-06 16:40:45.115 26605 INFO nova.openstack.common.rpc.common [-] Reconnecting to AMQP server on 192.168.8.53:5672
2014-08-06 16:40:45.157 26605 INFO nova.openstack.common.rpc.common [-] Connected to AMQP server on 192.168.8.53:5672
2014-08-06 16:41:14.828 26605 AUDIT nova.compute.resource_tracker [-] Auditing locally available compute resources
2014-08-06 16:41:15.458 26605 AUDIT nova.compute.resource_tracker [-] Free ram (MB): 112117
2014-08-06 16:41:15.458 26605 AUDIT nova.compute.resource_tracker [-] Free disk (GB): 483
2014-08-06 16:41:15.458 26605 AUDIT nova.compute.resource_tracker [-] Free VCPUS: 24
2014-08-06 16:41:15.530 26605 INFO nova.compute.resource_tracker [-] Compute_service record updated for athController.8.180.autohome.com.cn:athController.8.180.autohome.com.cn
```

5）192.168.8.53：5672接管消息队列，已正常工作



```
2014-08-06 16:43:16.042 26605 AUDIT nova.compute.resource_tracker [-] Auditing locally available compute resources
2014-08-06 16:43:16.670 26605 AUDIT nova.compute.resource_tracker [-] Free ram (MB): 112117
2014-08-06 16:43:16.670 26605 AUDIT nova.compute.resource_tracker [-] Free disk (GB): 483
2014-08-06 16:43:16.670 26605 AUDIT nova.compute.resource_tracker [-] Free VCPUS: 24
2014-08-06 16:43:16.746 26605 INFO nova.compute.resource_tracker [-] Compute_service record updated for athController.8.180.autohome.com.cn:athController.8.180.autohome.com.cn
2014-08-06 16:45:17.295 26605 AUDIT nova.compute.resource_tracker [-] Auditing locally available compute resources
2014-08-06 16:45:17.928 26605 AUDIT nova.compute.resource_tracker [-] Free ram (MB): 112117
2014-08-06 16:45:17.928 26605 AUDIT nova.compute.resource_tracker [-] Free disk (GB): 483
2014-08-06 16:45:17.928 26605 AUDIT nova.compute.resource_tracker [-] Free VCPUS: 24
2014-08-06 16:45:18.000 26605 INFO nova.compute.resource_tracker [-] Compute_service record updated for athController.8.180.autohome.com.cn:athController.8.180.autohome.com.cn
```

6）依次停止8.53:5672消息服务，8.87:5672正常接管



```
2014-08-06 16:47:34.405 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/amqplib/client_0_8/method_framing.py", line 233, in write_method
2014-08-06 16:47:34.405 26605 TRACE nova.openstack.common.rpc.common     self.dest.write_frame(1, channel, payload)
2014-08-06 16:47:34.405 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/amqplib/client_0_8/transport.py", line 125, in write_frame
2014-08-06 16:47:34.405 26605 TRACE nova.openstack.common.rpc.common     frame_type, channel, size, payload, 0xce))
2014-08-06 16:47:34.405 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/eventlet/greenio.py", line 309, in sendall
2014-08-06 16:47:34.405 26605 TRACE nova.openstack.common.rpc.common     tail = self.send(data, flags)
2014-08-06 16:47:34.405 26605 TRACE nova.openstack.common.rpc.common   File "/usr/lib/python2.6/site-packages/eventlet/greenio.py", line 295, in send
2014-08-06 16:47:34.405 26605 TRACE nova.openstack.common.rpc.common     total_sent += fd.send(data[total_sent:], flags)
2014-08-06 16:47:34.405 26605 TRACE nova.openstack.common.rpc.common error: [Errno 104] Connection reset by peer
2014-08-06 16:47:34.405 26605 TRACE nova.openstack.common.rpc.common
2014-08-06 16:47:34.406 26605 INFO nova.openstack.common.rpc.common [-] Reconnecting to AMQP server on 192.168.8.180:5672
2014-08-06 16:47:34.407 26605 ERROR nova.openstack.common.rpc.common [-] AMQP server on 192.168.8.180:5672 is unreachable: [Errno 111] ECONNREFUSED. Trying again in 1 seconds.
2014-08-06 16:47:35.407 26605 INFO nova.openstack.common.rpc.common [-] Reconnecting to AMQP server on 192.168.8.53:5672
2014-08-06 16:47:35.408 26605 ERROR nova.openstack.common.rpc.common [-] AMQP server on 192.168.8.53:5672 is unreachable: [Errno 111] ECONNREFUSED. Trying again in 3 seconds.
2014-08-06 16:47:38.410 26605 INFO nova.openstack.common.rpc.common [-] Reconnecting to AMQP server on 192.168.8.87:5672
2014-08-06 16:47:38.454 26605 INFO nova.openstack.common.rpc.common [-] Connected to AMQP server on 192.168.8.87:5672
2014-08-06 16:48:34.616 26605 AUDIT nova.compute.resource_tracker [-] Auditing locally available compute resources
2014-08-06 16:48:35.234 26605 AUDIT nova.compute.resource_tracker [-] Free ram (MB): 112117
2014-08-06 16:48:35.234 26605 AUDIT nova.compute.resource_tracker [-] Free disk (GB): 483
2014-08-06 16:48:35.235 26605 AUDIT nova.compute.resource_tracker [-] Free VCPUS: 24
2014-08-06 16:48:35.304 26605 INFO nova.compute.resource_tracker [-] Compute_service record updated for athController.8.180.autohome.com.cn:athController.8.180.autohome.com.cn
```

3.3 RabbitMQ恢复

1)主节点恢复启动MQ

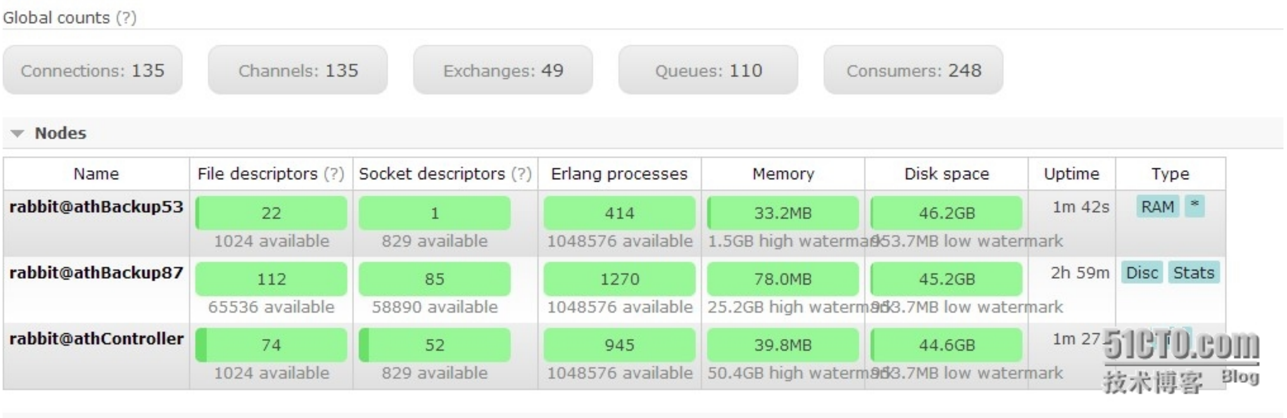
```
[root@athControllernova]# /etc/init.d/rabbitmq-server start

Startingrabbitmq-server: SUCCESS

rabbitmq-server.

    [root@athController nova]#
```

2)RabbitMQ监控显示

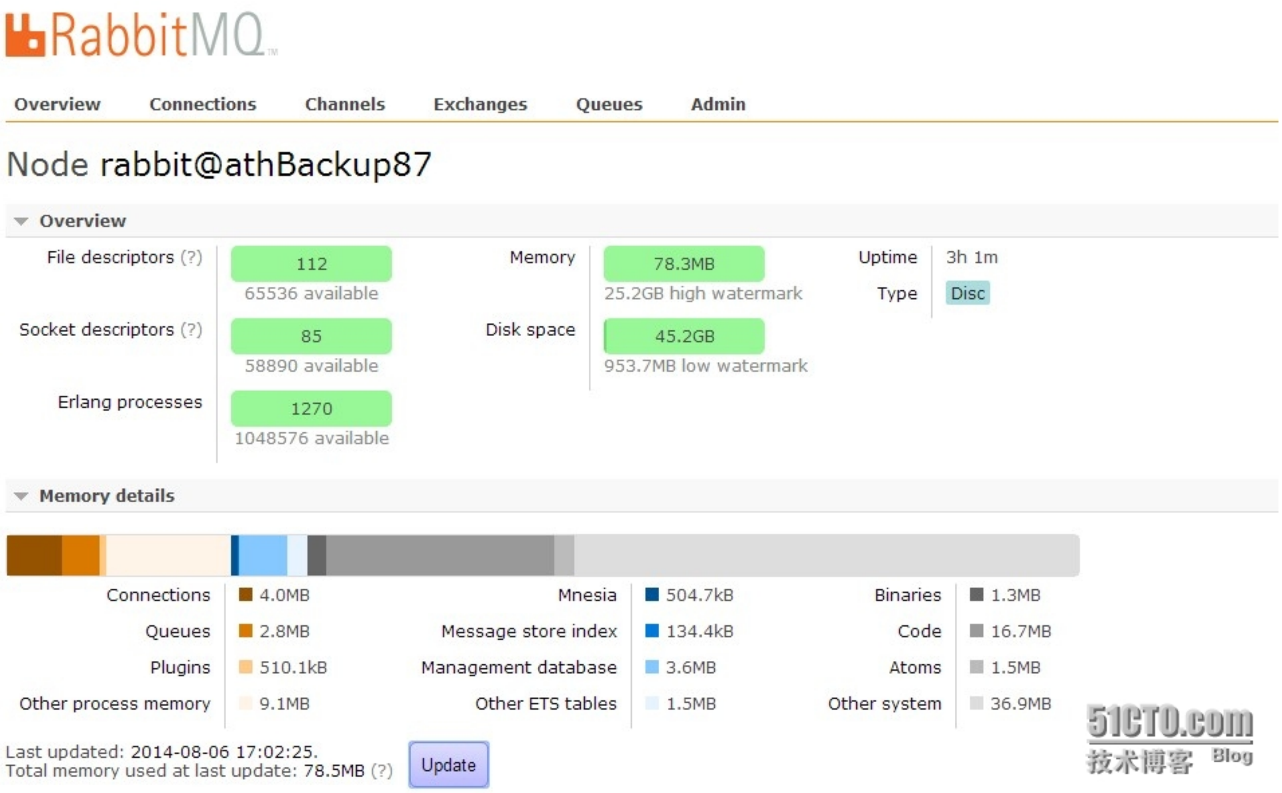


Global counts (?)

Connections: 135 | Channels: 135 | Exchanges: 49 | Queues: 110 | Consumers: 248

Nodes

Name	File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Uptime	Type
rabbit@athBackup53	22 1024 available	1 829 available	414 1048576 available	33.2MB 1.5GB high waterma	46.2GB 953.7MB low watermark	1m 42s	RAM *
rabbit@athBackup87	112 65536 available	85 58890 available	1270 1048576 available	78.0MB 25.2GB high waterm	45.2GB 953.7MB low watermark	2h 59m	Disc Stats
rabbit@athController	74 1024 available	52 829 available	945 1048576 available	39.8MB 50.4GB high waterm	44.6GB 953.7MB low watermark	1m 27	



Node rabbit@athBackup87

Overview

File descriptors (?)	112 65536 available	Memory	78.3MB 25.2GB high watermark	Uptime	3h 1m
				Type	Disc
Socket descriptors (?)	85 58890 available	Disk space	45.2GB 953.7MB low watermark		
Erlang processes	1270 1048576 available				

Memory details

Connections	■ 4.0MB	Mnesia	■ 504.7kB	Binaries	■ 1.3MB
Queues	■ 2.8MB	Message store index	■ 134.4kB	Code	■ 16.7MB
Plugins	■ 510.1kB	Management database	■ 3.6MB	Atoms	■ 1.5MB
Other process memory	■ 9.1MB	Other ETS tables	■ 1.5MB	Other system	■ 36.9MB

Last updated: 2014-08-06 17:02:25.
Total memory used at last update: 78.5MB (?)

第4章 RabbitMQ知识普及

4.1 RabbitMQ用户权限

1）创建用户，配置密码

a)add_user <username><password>

2）删除用户

a)delete_user <username>

3）改变用户密码

a)change_password<username> <newpassword>

4）清除用户密码

a)clear_password <username>

5）设置用户权限

a)set_user_tags <username><tag>

b)rabbitmqctl set_user_tags rootadministrator

6）查询用户权限

a)list_users

4.2 RabbitMQ集群

1）添加RabbitMQ节点加入集群

a)join_cluster<clusternode> [--ram]

2）查看集群状态

a)cluster_status

3）修改集群node存储方式

a)change_cluster_node_type disc |ram

b)rabbitmqctl change_cluster_node_typeram

4）跟新集群节点

a)update_cluster_nodesclusternode

5）RabbitMQ同步消息队列

a)sync_queue queue

b)rabbitmqctl sync_queue compute

4.3 RabbitMQ查询

1）查询消息队列信息

a)list_queues [-p<vhostpath>] [<queueinfoitem> ...]

2）查询消息交换机信息

a)list_exchanges [-p<vhostpath>] [<exchangeinfoitem> ...]

3）查询exchanges队列直接bind信息

a)list_bindings [-p<vhostpath>] [<bindinginfoitem> ...]

4）查询消息连接信息，如果是集群，将显示node bind信息

a)list_connections[<connectioninfoitem> ...]

4.4 RabbitMQ其他

1)RabbitMQ进程停止

a)stop [<pid_file>]

b)service rabbitmq-srver

2)RabbitMQ应用停止与启动

a)stop_app，start_app

3)RabbitMQ重置/强制重置节点到原始状态

a)reset ， force_reset

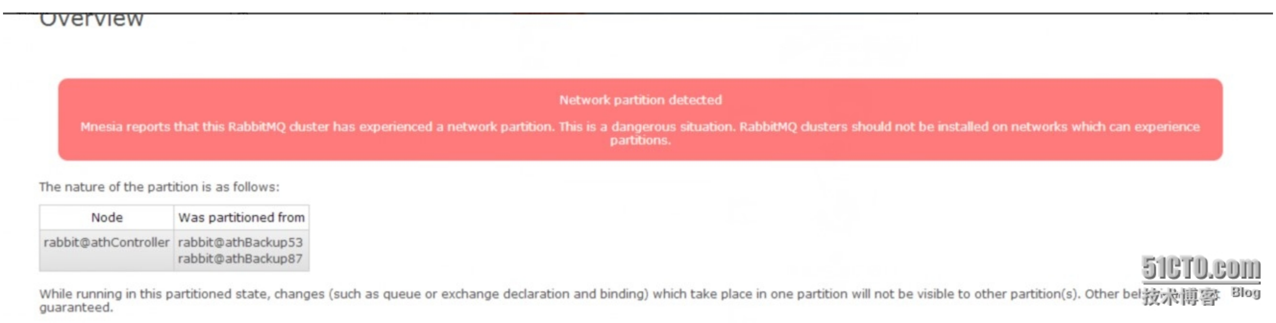
4)查看RabbitMQ基础信息

5)Rabbbitmqctl status

第5章 FAQ

5.1 RabbitMQ集群-网络分区

1)RabbitMQ集群网络分区问题



Overview

Network partition detected

Mnesia reports that this RabbitMQ cluster has experienced a network partition. This is a dangerous situation. RabbitMQ clusters should not be installed on networks which can experience partitions.

The nature of the partition is as follows:

Node	Was partitioned from
rabbit@athController	rabbit@athBackup53 rabbit@athBackup87

While running in this partitioned state, changes (such as queue or exchange declaration and binding) which take place in one partition will not be visible to other partition(s). Other bel guaranteed.

abbitMQ clusters do not tolerate network partitions well.If you are thinking of clusteringacross a WAN, don't.You shouldusefederationor theshovelinstead.

rabbitmq一共有三种处理方式：ignore，autoheal，pause_minority。默认的处理方式是ignore，即什么也不做

1）ignore默认处理方式

```
[root@athControllerrabbitmq]# cat rabbitmq.config

% This file managed by Puppet

% Template Path: rabbitmq/templates/rabbitmq.config

[

 {rabbit, [

 {default_user,<<"guest">>},

 {default_pass,<<"guest">>},

 {tcp_listeners,[5672]},

{cluster_partition_handling, ignore}

 ]},

 {kernel, [


 ]}

].

% EOF

[root@athController rabbitmq]#
```

2）autoheal的处理方式：简单来讲就是当网络分区恢复后，rabbitmq各分区彼此进行协商，分区中客户端连接数最多的为胜者，其余的全部会进行重启，这样也就恢复到同步的状态了

3）pause_minority的处理方式：rabbitmq节点感知集群中其他节点down掉时，会判断自己在集群中处于多数派还是少数派，也就是判断与自己形成集群的节点个数在整个集群中的比例是否超过一半。如果是多数派，则正常工作，如果是少数派，则会停止rabbit应用并不断检测直到自己成为多数派的一员后再次启动rabbit应用。注意：这种处理方式集群通常由奇数个节点组成。在CAP中，优先保证了CP

RabbitMQ集群的网络分区容错性并不是非常高，在网络经常发生分区时会有些问题，最明显的就是脑裂问题，官方给出的处理方式

Clustering and Network Partitions

RabbitMQ clusters do not tolerate network partitions well. Ifyou are thinking of clustering across a WAN, don't. You should usefederationor theshovelinstead.

However, sometimes accidents happen. This page documents how todetect network partitions, some of the bad effects that may happen duringpartitions, and how to recover from them.

RabbitMQ stores information about queues, exchanges, bindingsetc in Erlang's distributed database,Mnesia.Many of the details of what happens around network partitions are related to Mnesia'sbehaviour.

第6章参考

6.1 URL参考

http://www.rabbitmq.com/ha.html

http://www.rabbitmq.com/shovel.html

http://www.rabbitmq.com/partitions.html

http://www.rabbitmq.com/partitions.html

http://www.rabbitmq.com/clustering.html

http://www.bbtang.info/linux/fuwu/610.html

http://www.rabbitmq.com/federation.html

http://baike.baidu.com/view/4095865.htm?fr=aladdin

http://www.kankanews.com/ICkengine/archives/71918.shtml

http://lists.rabbitmq.com/pipermail/rabbitmq-discuss/2014-January/033412.html

http://rabbitmq.1065348.n5.nabble.com/Node-statistics-not-available-v3-2-0-td32937.html

这是一篇网上的文章，为了重复造轮子，这里直接粘过来了，我们之后，可以根据自己的需求，在这个基础上进行修改。

连接地址：http://swq499809608.blog.51cto.com/797714/1540205