如何实现不中断业务的情况下替换MOS 控制节点

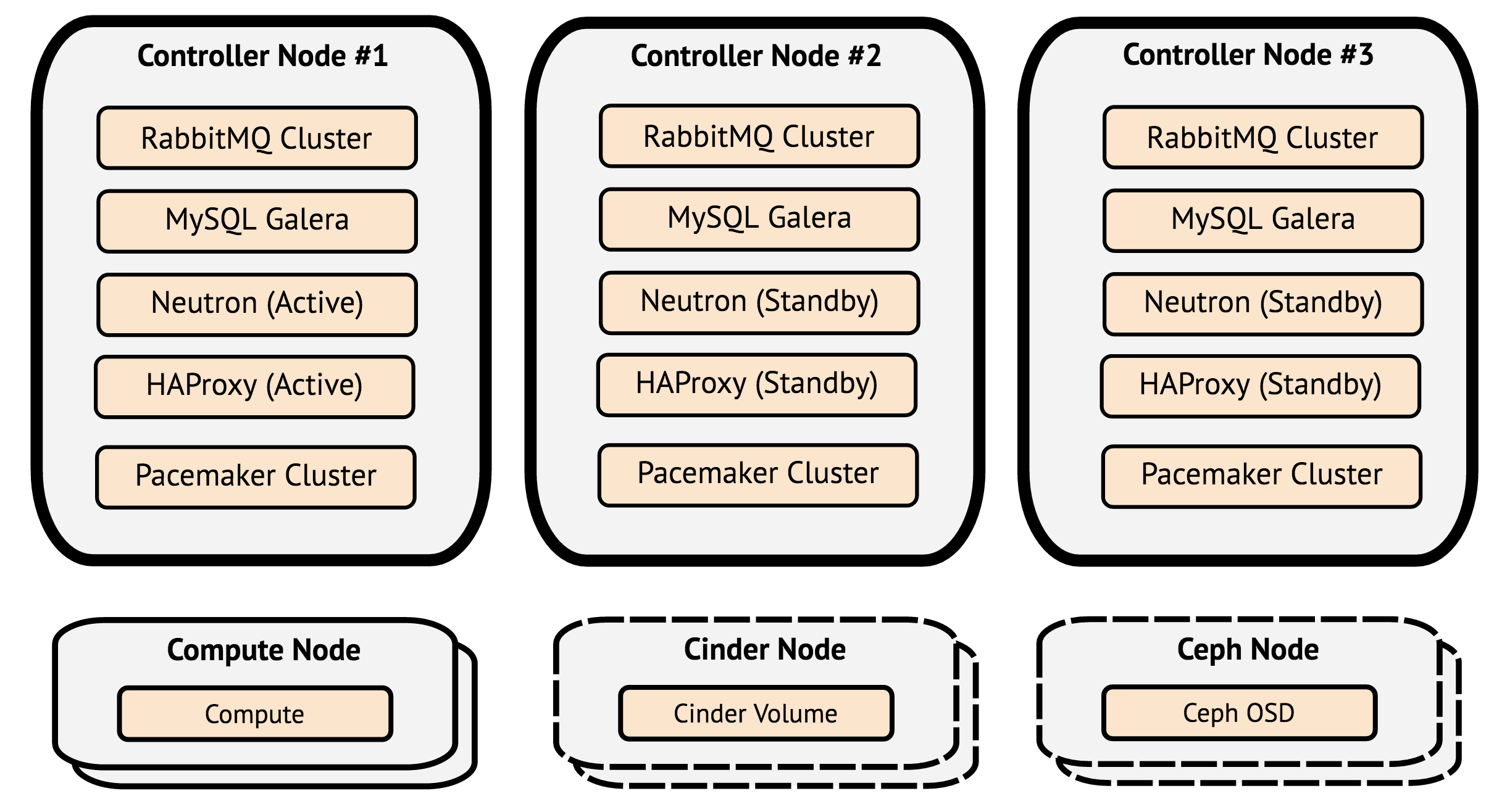
摘要： 在OpenStack的架构设计中，控制节点是使用通用的高可用组件实现的，正是由于组件多而且配置复杂，一旦发生故障，是令管理员非常头痛的问题，幸亏有Mirantis OpenStack Fuel 提供的自动化机制，帮助运维人员无缝、无痛的毋需间断业务的情况下随时替换控制节点。

## 常见场景：

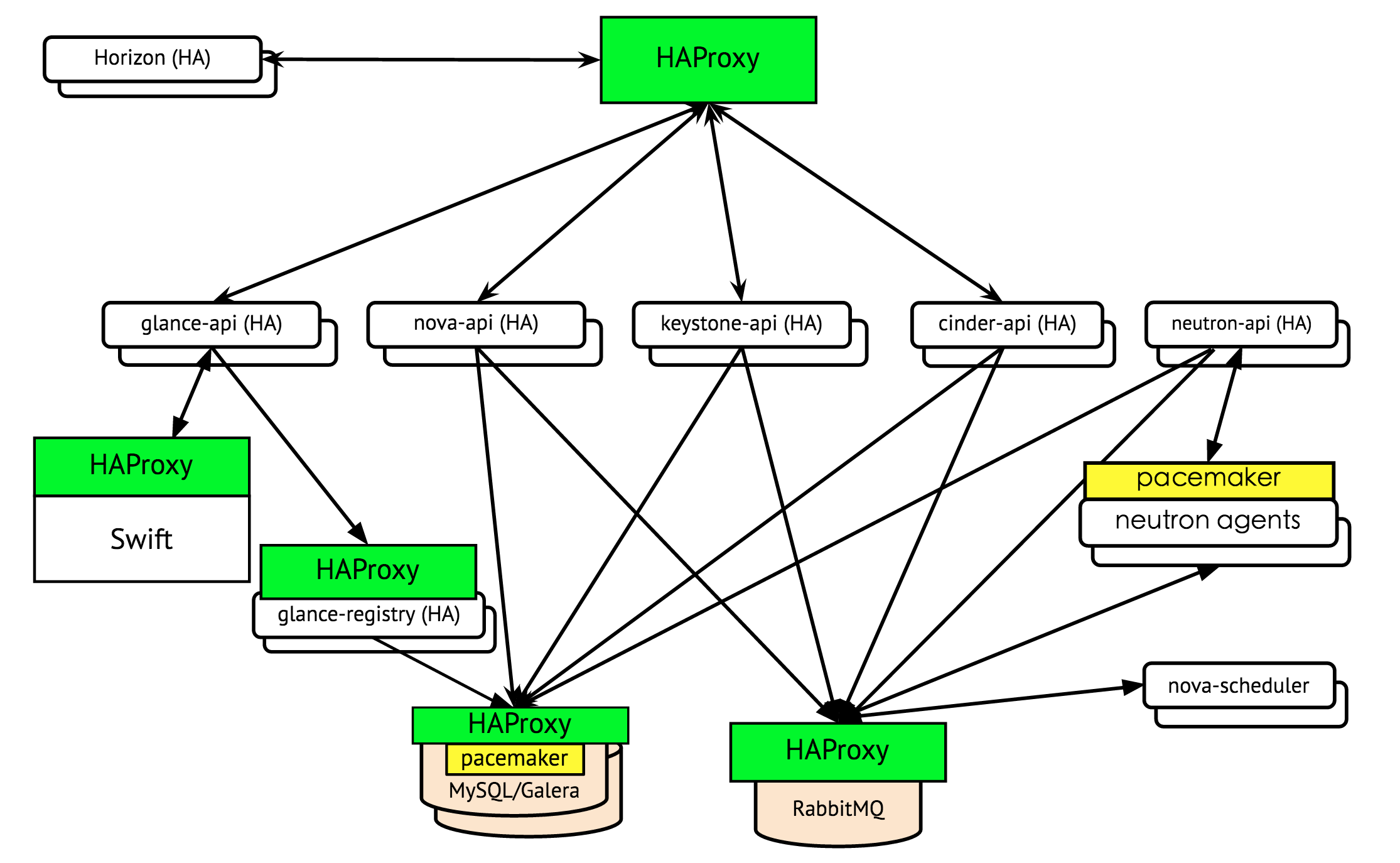
在一个利旧项目中，因为前期设备不够，用户使用了2台即将过保的服务器作为了控制节点，后期采购了新的设备，为保证云平台的稳定性和健壮性，需要把2台过保的服务器替换成新购的服务器。

控制节点是云平台的核心，上面跑着OpenStack多种关键服务，Mirantis OpenStack通过配置3个控制节点来保证服务的高可用，而替换控制节点是一个高风险操作，一旦操作失误可能造成集群起不来，云平台无法使用。而 Mirantis OpenStack使用自动化部署和管理工具Fuel，可以轻松在线替换控制节点，不影响业务虚拟机运行。只需要使用鼠标轻松点击，无需复杂操作就可完成替换作业。

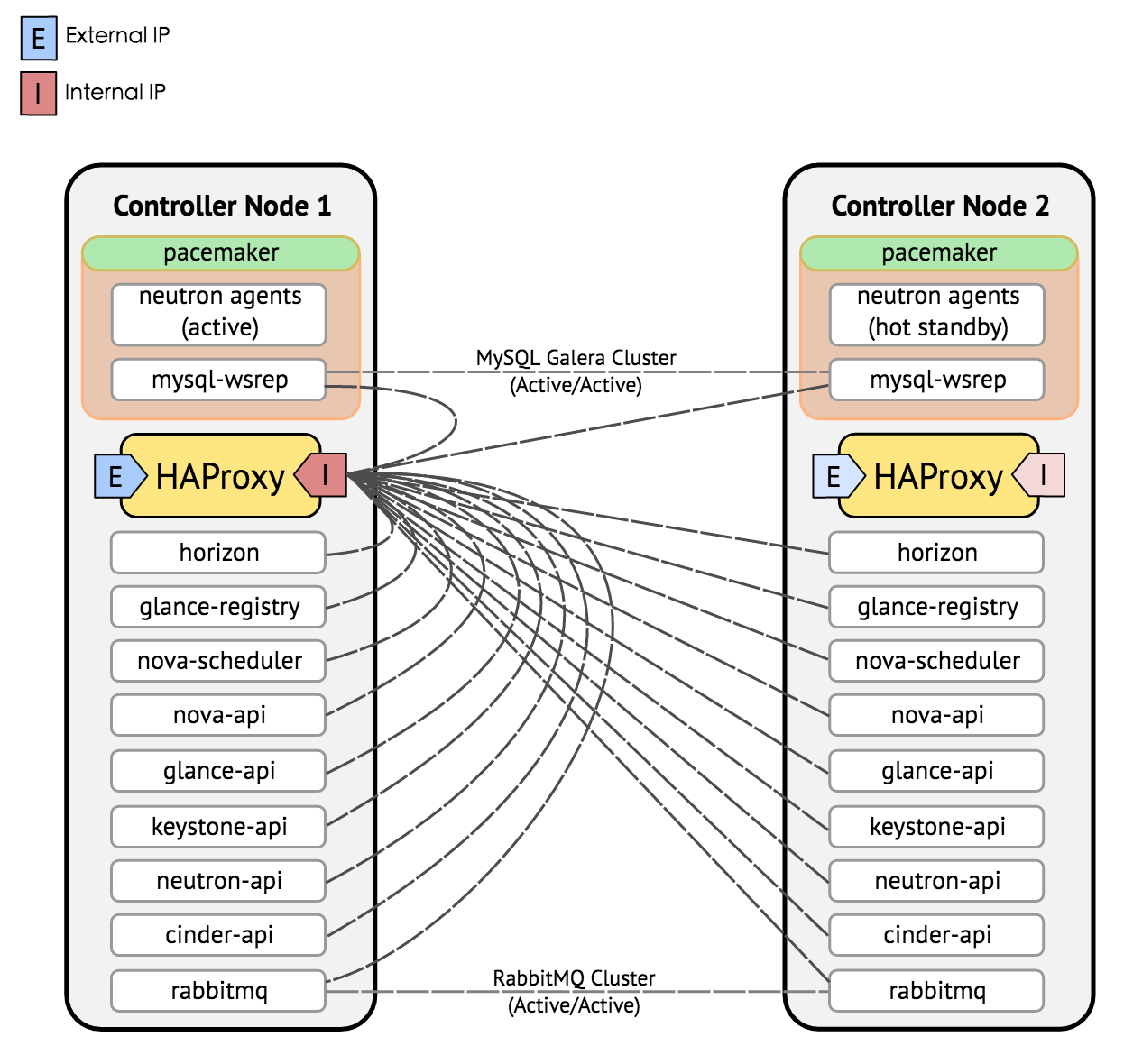
## 背景知识之：控制节点高可用架构介绍：



OpenStack各组件间通过基于HTTP协议的 RESTful API和基于 AMQP消息进行通信。无状态的API服务由pacemaker通过管理VIP来实现高可用，通过HAProxy实现负载均衡。有状态的服务，比如数据库和消息队列，使用各自的active/active或active/passive方案实现高可用。RabbitMQ使用内置的集群方案，MySQL使用Galera集群实现高可用。



Fuel 使用Pacemaker脚本在独立的网络 namespace中部署 HAProxy。多个控制节点才能实现高可用，为了能够选举出主控制节点，节点数必须为奇数。



每个控制节点都运行HAProxy，并提供如下服务

1. 所有控制节点管理唯一一个external VIP和一个Internal VIP
2. 提供四层（TCP）和七层（HTTP）的负载均衡

各个组件/服务的HA机制是不同的

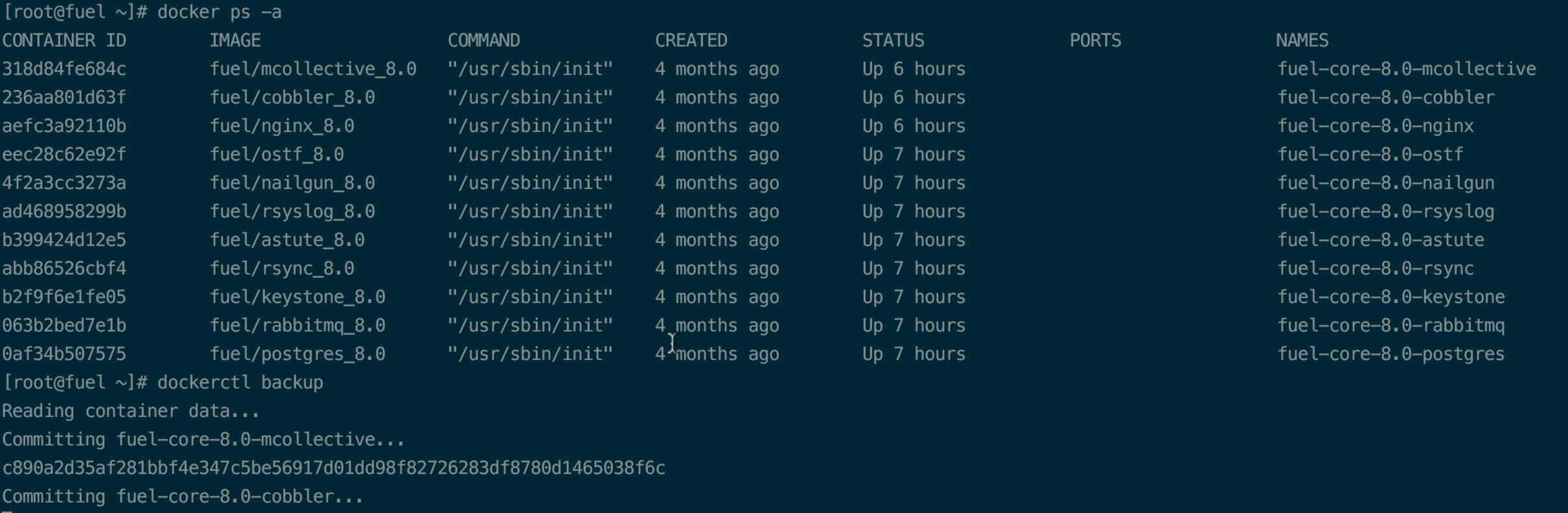
* 无状态服务，比如nova-api, glance-api, neutron-api, keystone-api, nova-scheduler, cinder-api，HAProxy接收到请求后直接转发即可。
* Horizon，作为典型的web应用，需要在haproxy中配置粘性会话（sticky sessions），确保同一用户的请求会被分配到同一后端。
* RabbitMQ提供基于镜像队列（mirrored queues）的active/active HA。
* MySQL使用MySQL/Galera的高可用方案。
* Neutron agent是active/passive方案
* Ceph monitor有自己的quorum-based HA机制。它要求所有节点上的时钟保持一致，如果偏移大于50ms，可能会导致服务出现问题

## 操作步骤：

1、在进行替换操作之前，需要进行备份操作，备份内容包括Fuel 平台，云平台及云平台数据库。

（1）备份Fuel 平台

通过Fuel 自带的dockerctl backup备份组件容器，备份到 /var/backup/fuel/目录中。



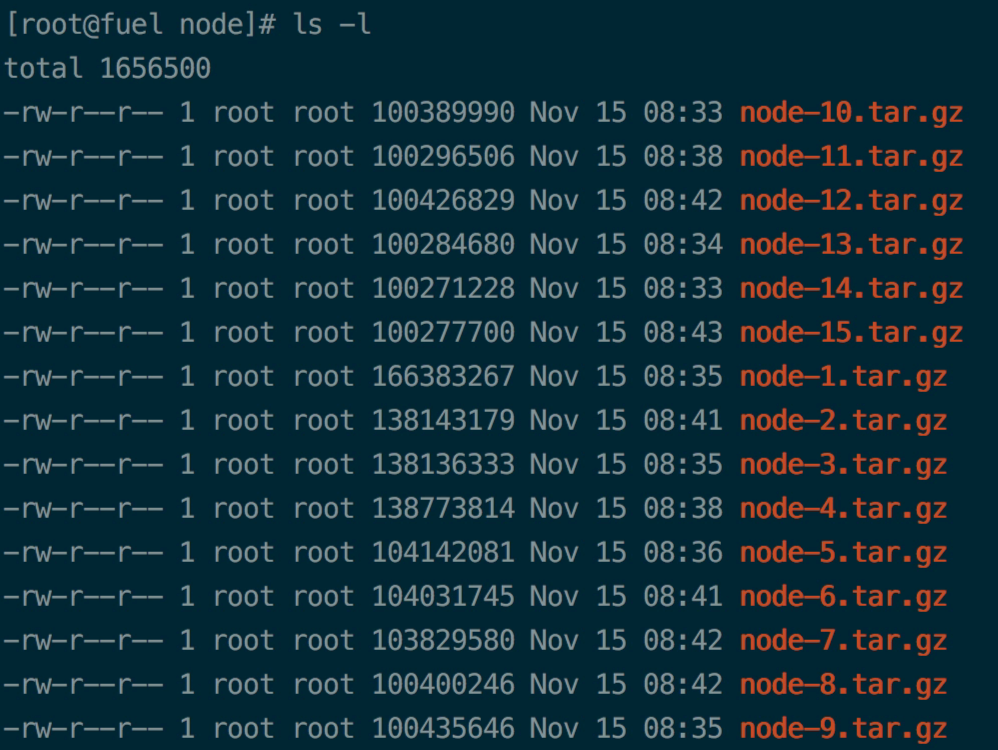
（2）备份云平台

对各个节点进行备份，包括控制节点，存储节点，计算节点，监控节点，备份/etc/，/usr/share目录，将备份节点备份到/var/backup/node目录。

登录fuel 节点，运行云平台备份脚本，脚本内容如下：

|  |
| --- |
| count=$(fuel node --env=1 |grep ready |wc -l)  for ((i=3;i<=$count;i++));  do  id=`fuel node --env=1 | awk "NR==$i" | awk '{print$1}'`  nodename=node-$id  ssh $nodename 'tar cvzfp /tmp/`hostname`.tar.gz /etc/ /usr/share/;'  scp $nodename:/tmp/$nodename.domain.tld.tar.gz /var/backup/node/  done |

查看/var/backup/node/目录下的文件，如下所示：



（3）备份云平台数据库

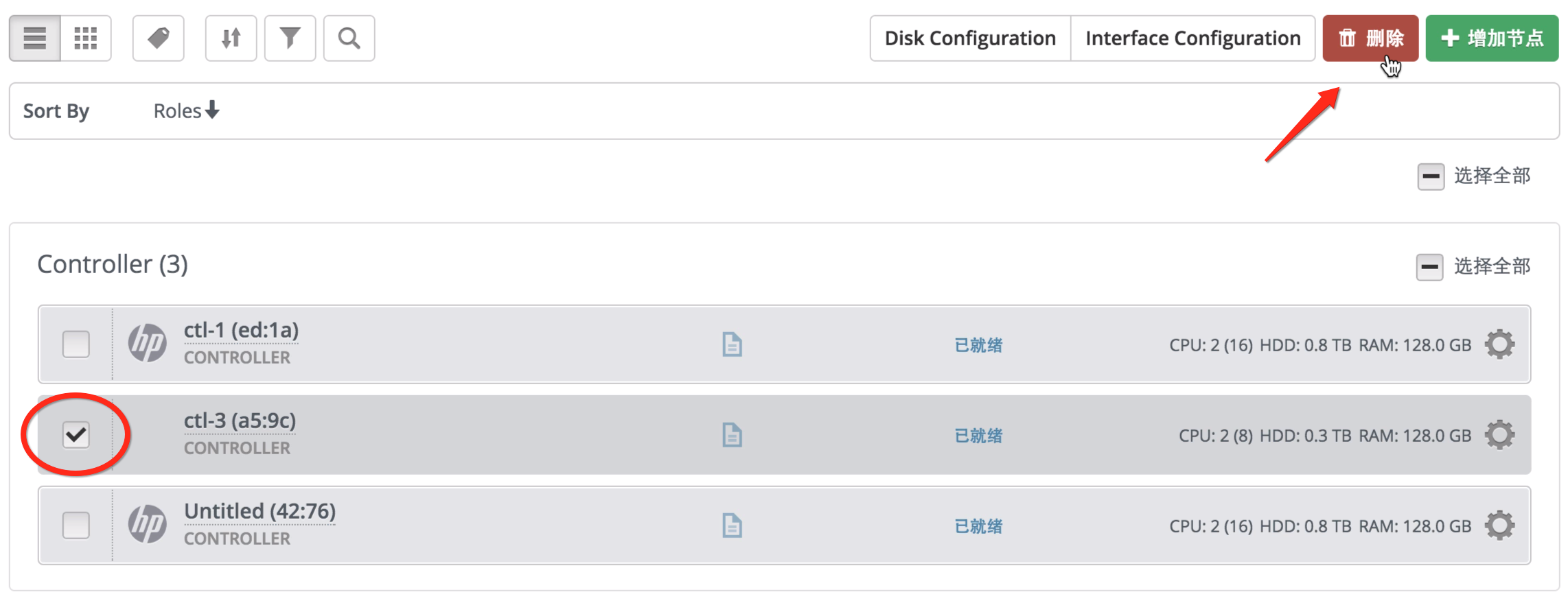
对云平台的数据库进行备份，首先登录到fuel节点，通过ssh连接到控制节点，在控制节点上备份数据库，同时将备份结果通过scp传输到fuel节点的/var/backup/mysql目录。

登录fuel 节点，运行云平台数据库备份脚本，脚本内容如下：

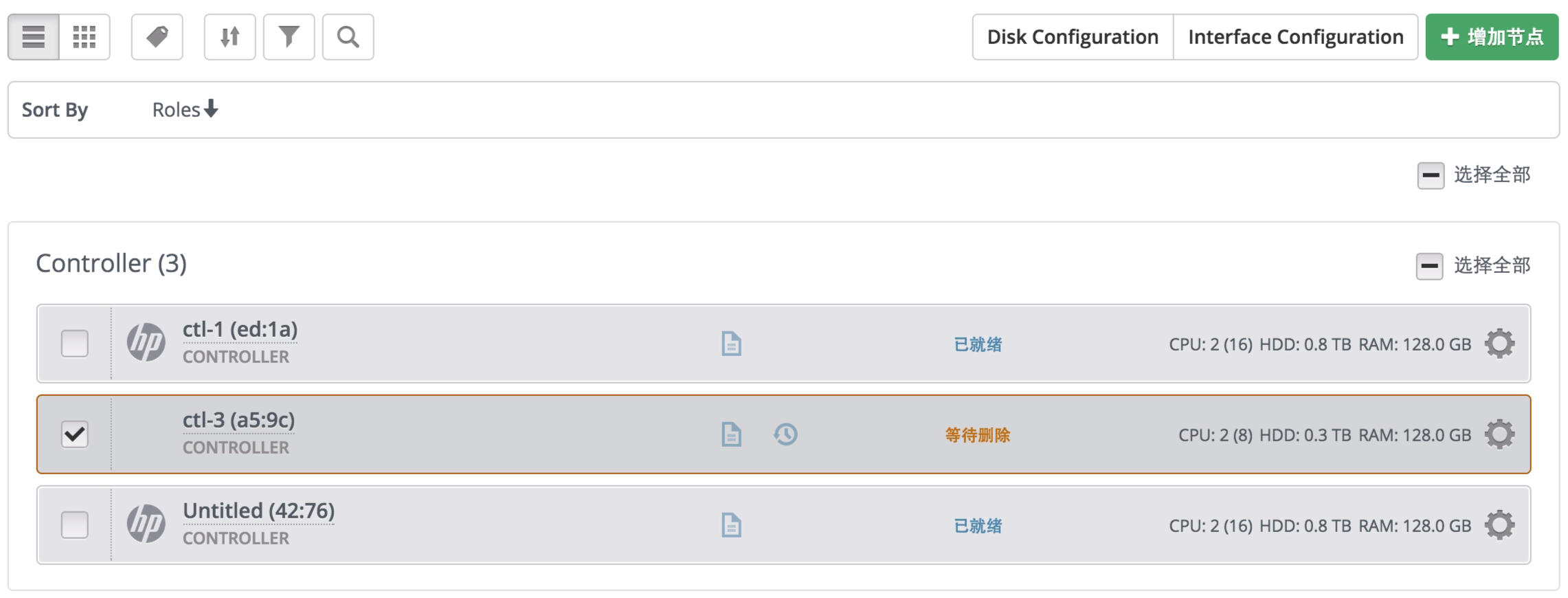
|  |
| --- |
| sql\_bak=$(date +%Y-%m-%d-%H.%M.%S)  ssh node-1 'mysqldump --events --all-databases --lock-all-tables --flush-logs -uroot -hlocalhost -pssodKZ1m6IcxGZAHnXF3FYmu |gzip -c | cat >/var/log/backup  /mysql\_backup.gz'  scp node-1:/var/log/backup/mysql\_backup.gz /var/backup/mysql/mysql\_backup\_$sql\_bak.gz  ssh node-1 'rm -f /var/log/backup/mysql\_backup.gz' |

2、移除控制节点

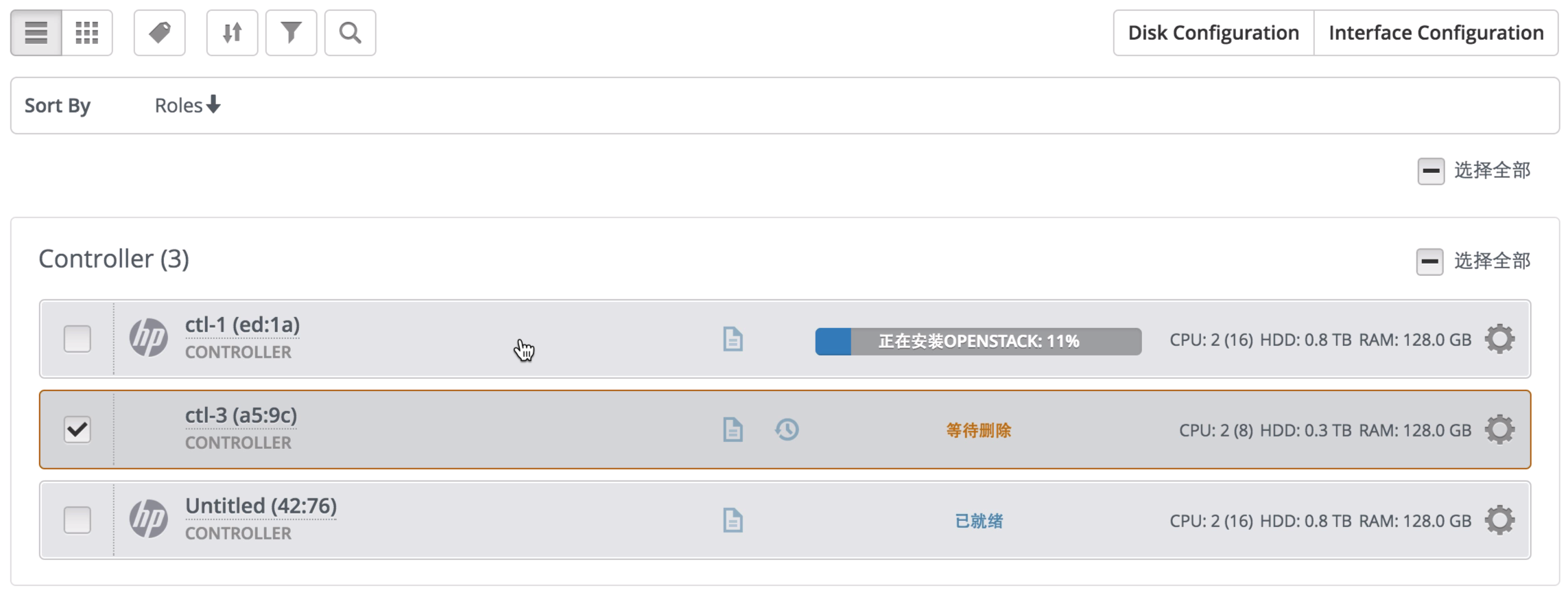
登录Fuel Web管理平台，选择需要移除的控制节点，点击“删除节点”



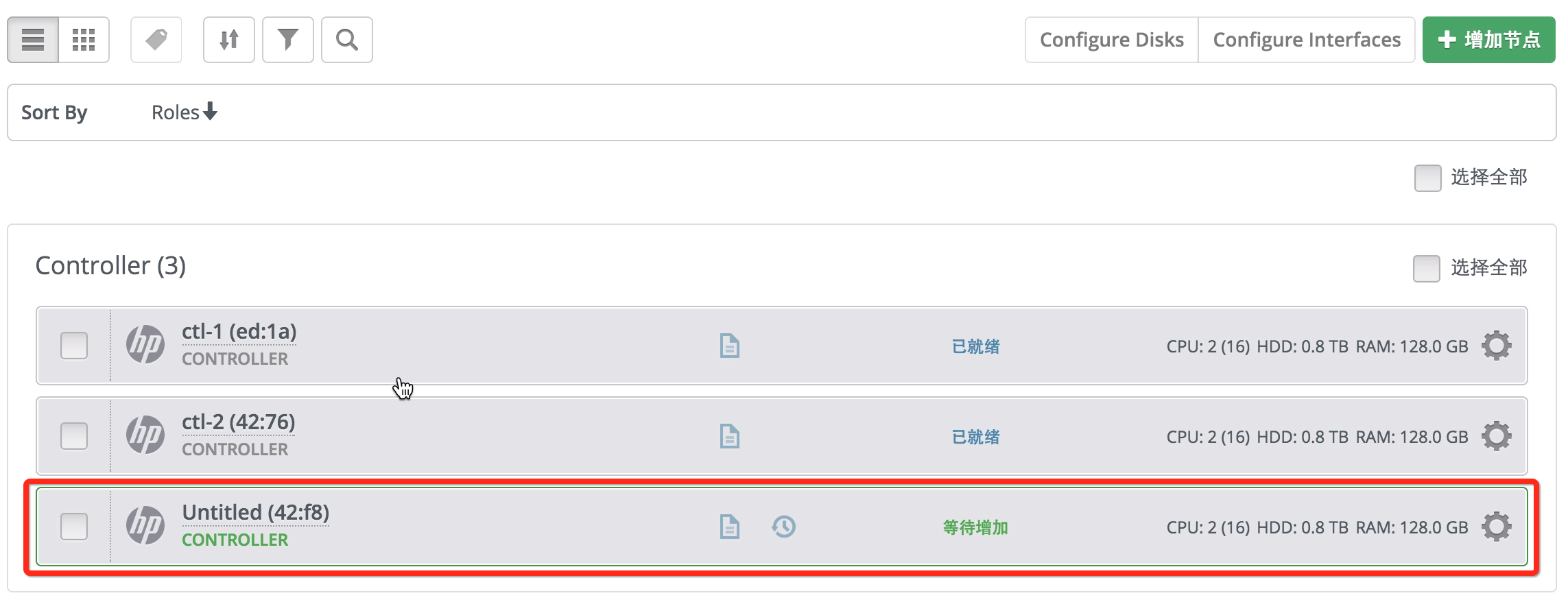
节点处于“等待删除”状态



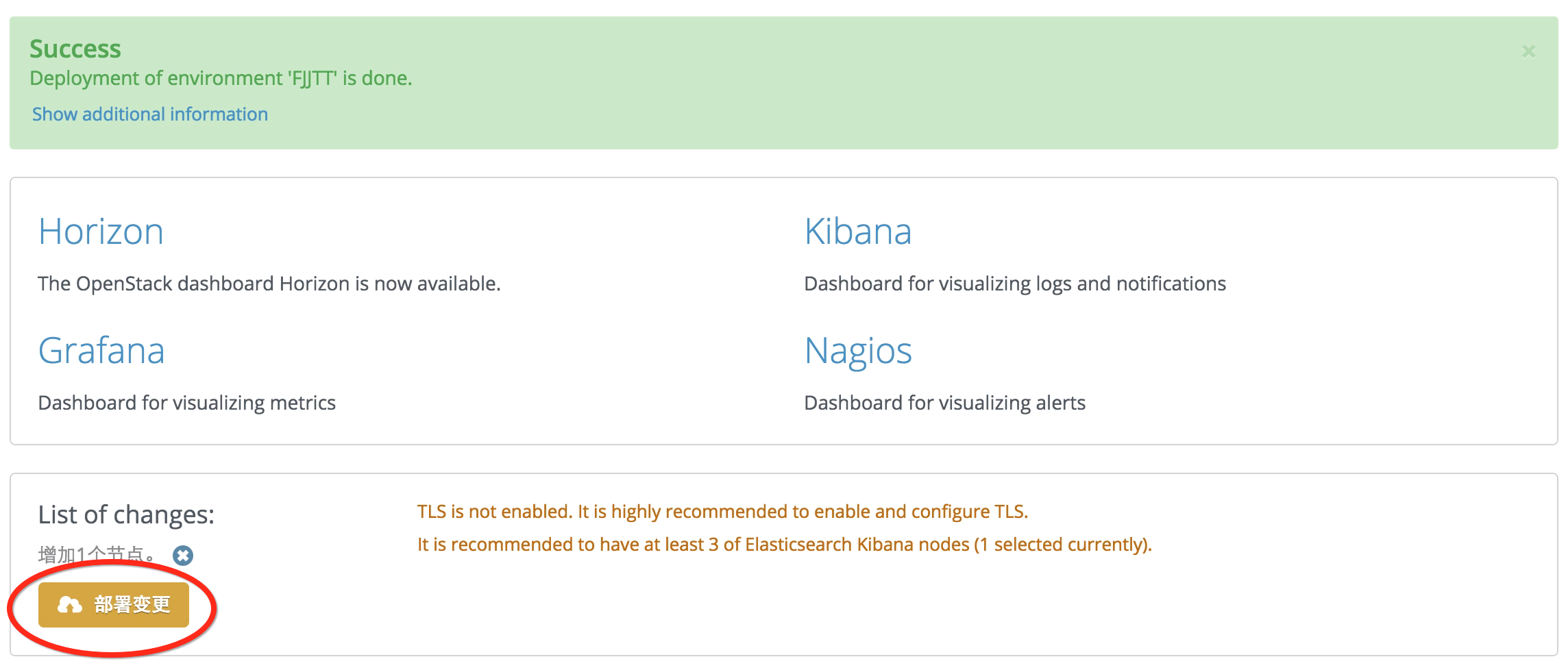
切换到“Dashboard”选项卡，点击“部署变更”，开始删除控制节点-3，集群各节点同步配置。



等待集群同步完成，删除结束，这时候就引导新的控制节点，添加到集群中：



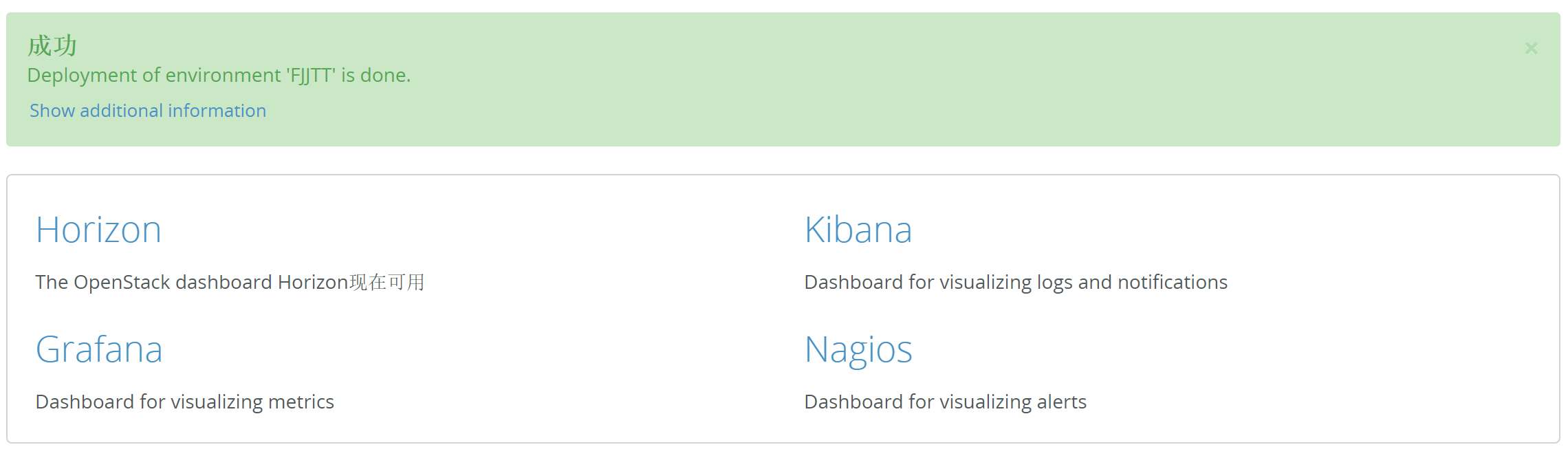
调整相关配置，验证网络，部署变更



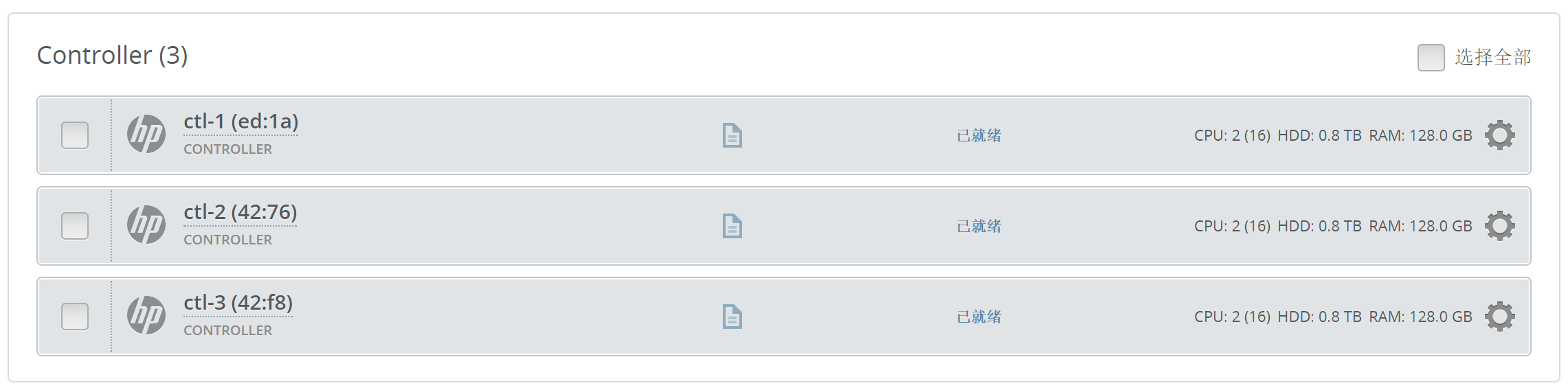
开始给新加入的控制节点安装操作系统



Fuel 会自动将新的节点加入集群，并同步其他节点信息，其他节点也会相信更新控制节点信息，同步完成后：



可以切换到“节点”选项卡看各节点状态，所有的控制节点已就绪：



这样，我们就轻松的完成了控制节点的替换。

注意：在我们做替换操作的同时，Mirantis OpenStack云平台上，管理员依旧在Horizon上进行各种操作，200多台业务虚拟机正运行的很欢畅。

# 总结：

替换控制节点是有风险的，不过Mirantis OpenStack大大简化了操作的难度，我们只需要点点鼠标就完成控制节点的替换，而且替换过程中不影响业务的运行。当然，这些操作都是需要你有一定的OpenStack基础，没有的话，也没有关系，UMCloud 工程师会帮助你！