PCI固定IP实现流程说明和测试建议

1. Deployment
2. 创建。

AdmissionWebHook截获deployment创建请求，解析出网络参数（网络域ID，子网ID，子网网关地址，子网CIDR，集群ID）。向IPResMgrSrv发起地址资源创建请求。

IPResMgrSrv收到请求后，依据网络参数向NSP请求副本数量的IP、Mac、PortID地址资源，将资源写入表tbl\_K8SResourceIPBind，每个资源是一条记录，DB写入完毕返回给WebHook表明地址池创建成功，否则失败。

1. 删除。

AdmissionWebHook截获消息后，通知IPResMgrSrv释放deployment对应的地址资源。

IPResMgrSrv在表tbl\_K8SResourceIPRecycle插入一条记录，记录实际要回收时间（在配置文件中配置了延期的时间，可动态reload），同时在服务的定时器队列中插入一条记录。如果服务重启，会从数据表中恢复属于该服务的定时回收任务。

租期到期，如果到期之前没有相同的deployment创建，IPResMgrSrv将会按tbl\_K8SResourceIPRecycle的记录信息将IP地址释放给NSP。

1. 租期未到时创建。

Deployment被删除后，其对应的地址资源会被保留一段时间，以便让用户能重复使用。这时会在tbl\_K8SResourceIPRecycle表中插入一条对应记录。

这时如果用户重新创建这个deployment，IPResMgrSrv会判断tbl\_K8SResourceIPRecycle是否有待回收的数据，如果有就会分三种情况来处理。

1. 副本数相同。直接恢复，将tbl\_K8SResourceIPRecycle中记录删除。
2. 当前请求的副本数小于原有的副本数。在表tbl\_K8SResourceIPBind表中选择未绑定的IP且数量为差额的记录，将其归还给NSP，同时删除该记录。

将tbl\_K8SResourceIPRecycle中记录删除。

1. 当前请求的副本数大于原有的副本数。计算所需增量，向NSP请求地址数据，将多条记录插入tbl\_K8SResourceIPBind。

将tbl\_K8SResourceIPRecycle中记录删除。

1. IPAM申请IP。

这是个并发操作，多个IPAM会同时向多个IPResMgrSrv发起申请IP的请求。资源池的mysql事务串行化是实现的重点。

这时tbl\_K8SResourceIPBind中会有副本数量的多条记录，状态都是未绑定。

表操作重点：

1. Transaction的隔离等级设置为LevelSerializable。
2. 使用SELECT … FOR UPDATE语句。
3. 选择后，设置该条记录为绑定状态。

测试重点：

1. IP是否有重复分配，多个pod得到相同的IP。
2. 是否有数据库死锁，这个通过log可以观察，也可以通过以下命令观察：

show status like '%lock%';

show OPEN TABLES where In\_use > 0;

show processlist;

1. IP分配是否和副本数对应。
2. 绑定标志是否为1，绑定PODID是否和kubectl中一致。

优化：

由于使用了LevelSerializable，这相当于是表锁，而不是gap lock，这样会影响申请的高并发。咨询了数据库专家，mysql8.0有skip locked，SELECT … FOR UPDATE SKIP LOCKED，同时保持原有的事务隔离等级repeatable\_read。

IPAM申请IP成功后，tbl\_K8SResourceIPBind表中对应记录绑定标志设置为1，同时设置绑定的PODID和绑定时间。

异常：这块对应逻辑还没开发，要有异常环境测试后决定如何处理。只做了node、pod状态查询。

1. 相同的deployment，一个pod被delete失败，k8s同时创建一个pod。IP被挂住没有正常释放，而新的POD因为无法获取IP而启动失败。
2. kubelet crash，节点没问题，网络没问题。这个时候pod还是可以服务的，但是5m之后k8s会重新创建pod，这时要手工释放原有pod。需要验证。
3. 节点失效。Pod是没有运行的，也没有解绑，这时需要手工释放pod对应IP。5m之后k8s会重新创建pod，需要验证。
4. IPAM释放IP。

POD被删除时，IPAM会调用IPResMgrSrv的释放接口，IPResMgrSrv收到请求后，根据关键字去表tbl\_K8SResourceIPBind中找到对应的记录，将bind标志设置为0。

1. 数据表。
2. Job、CronJob
3. 相关资料

Kubelet状态更新机制。<https://www.qikqiak.com/post/kubelet-sync-node-status/>