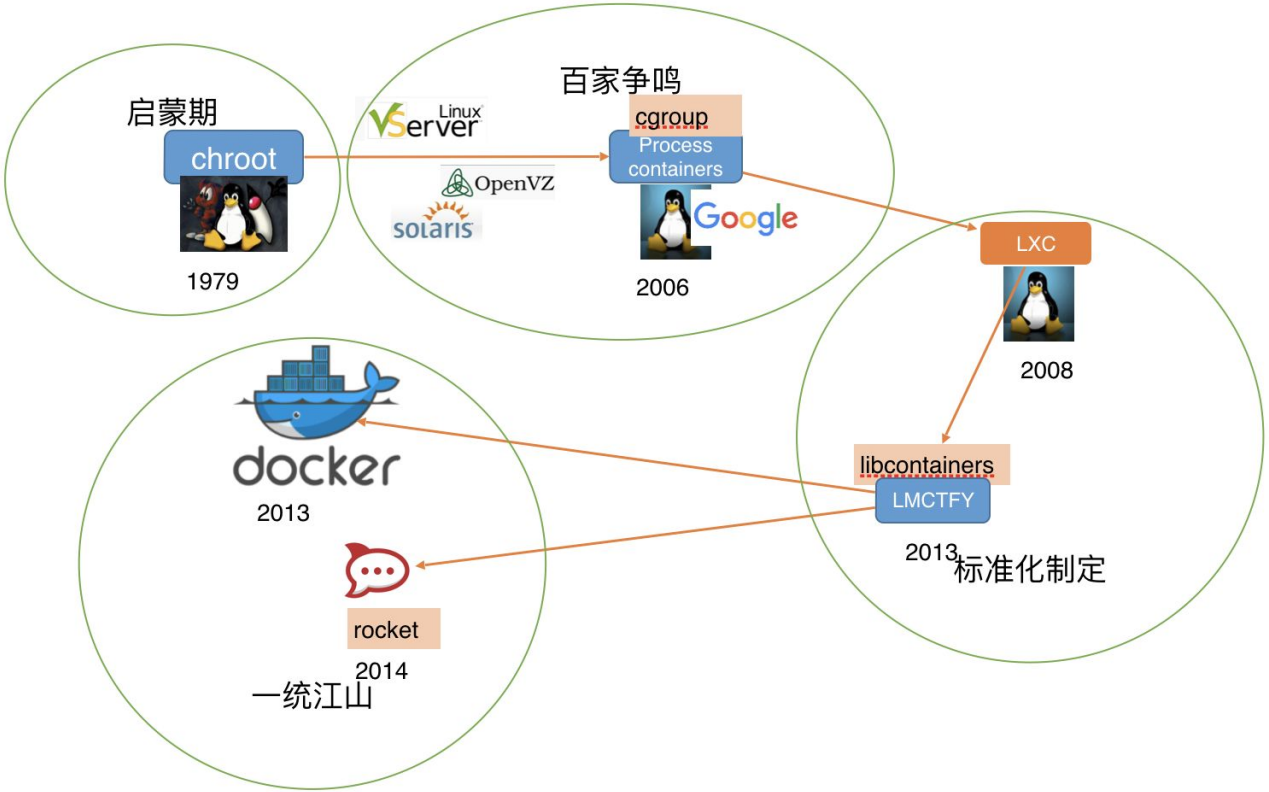
Caas与Iaas网络融合实践

By niusmallnan

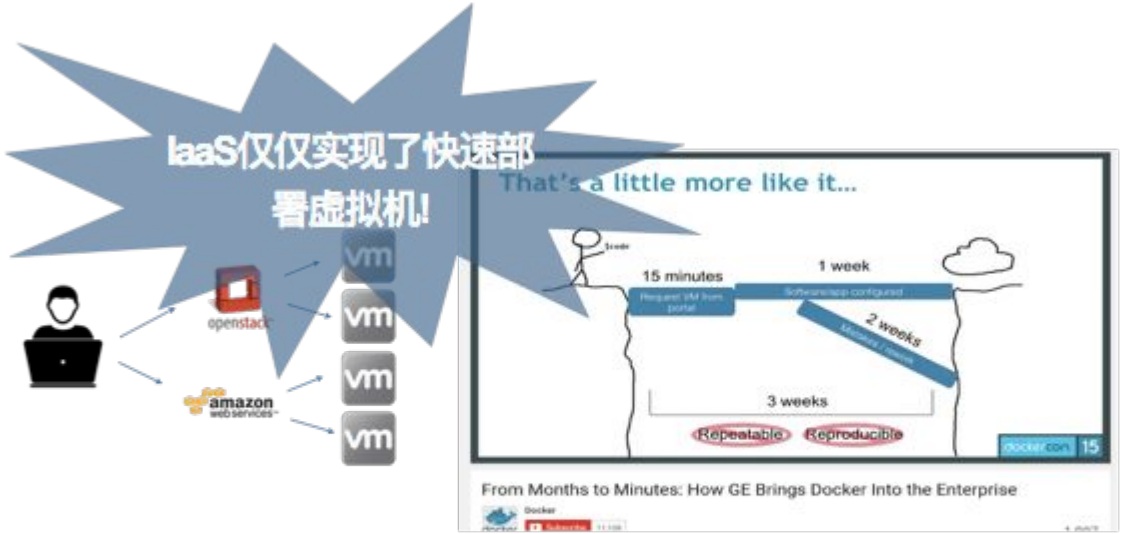
议程

1. Caas正突入云服务领域
2. Caas网络的需求和隐患
3. 解决之道与实践指南
4. Q&A

容器历史演进，技术日渐成熟



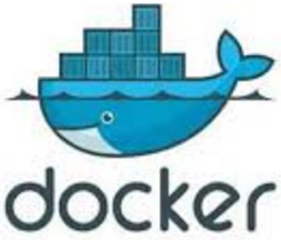
容器（Docker）解决什么问题



美国通用电气公司（GE）的经验是，用了IaaS后，部署VM只要15分钟，而应用的部署配置却仍要3个星期以上。

Container as a service

AWS ECS

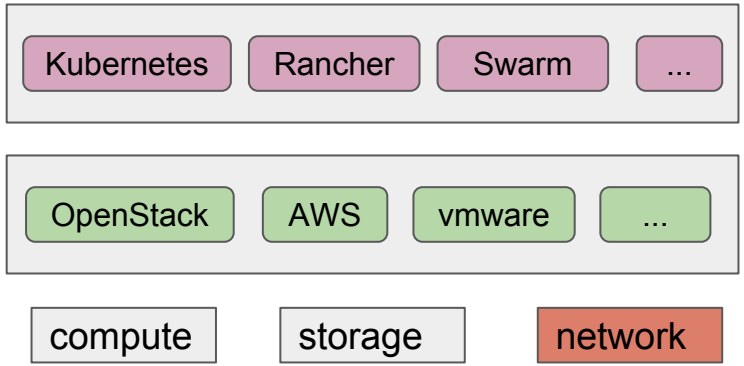


MAGNUM

Containers as a Service

kubernetes

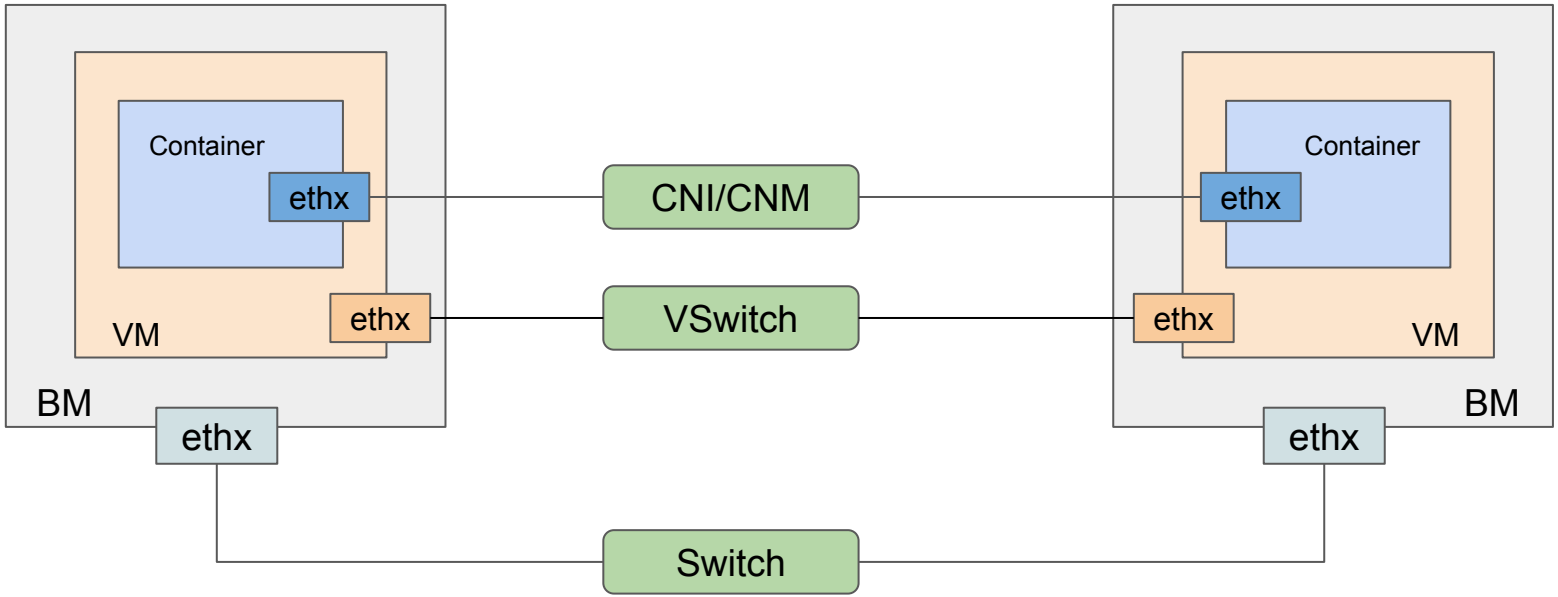
大部分Caas选择构建在Iaas之上



1. Caas的成熟度远没有达到可以管控物理 资源
2. 功能角度，Caas只是对Iaas的补充，无法替代
3. 主流的Iaas都在扩展对Caas的支持能力
4. 每一层架构都有自身的编排调度，每一层都依托上一层提供的基础设施资源，物理资源的实际使用率在不断递减

Caas网络的需求和隐患

全局网络概览


Container
ethx
VM
ethx
BM
ethx
CNI/CNM
VSwitch
Switch
Container
ethx
VM
ethx
BM
ethx

构建Caas网络两种模式

Caas内自治：
这种模式下，Caas不用关心底层设施服务，完全依靠自身编排引擎对网络的管理。

好处：
对于用户来说，管理非常简单，无需各种干预。

劣势：
对于用户来说，几乎不可控，有些网络性能较差，且很难与基础设施服务联动。

代表：
overlay(vxlan/ipsec)、Calico

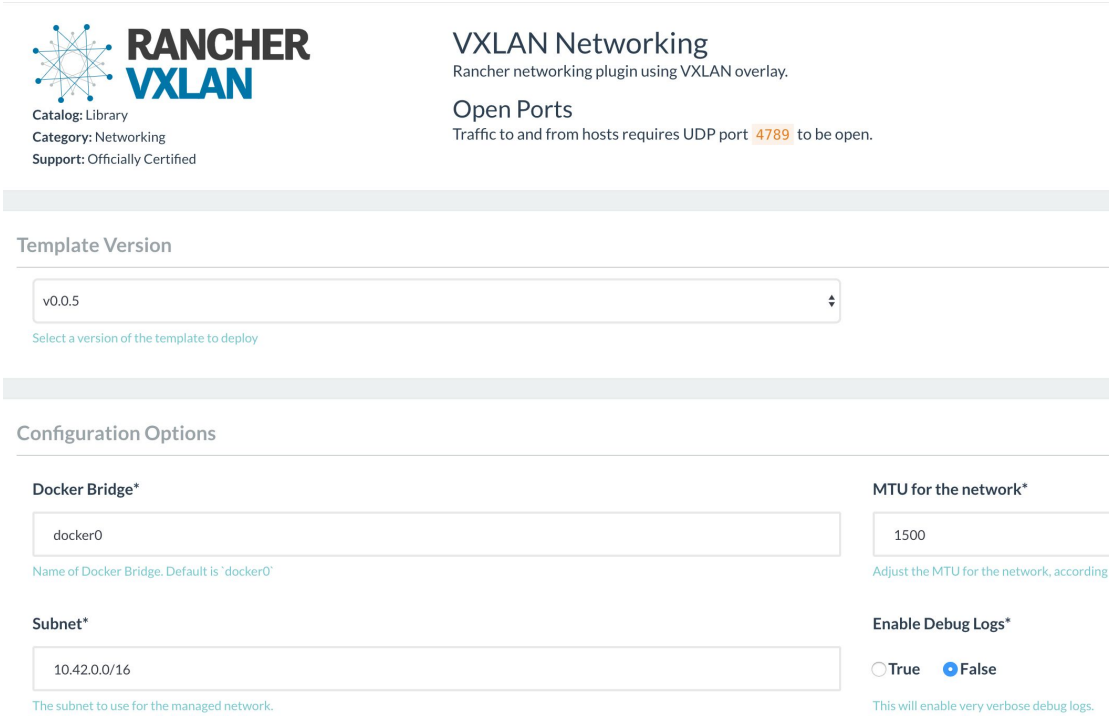
依托Iaas：
这种模式下，网络资源依然是依托Iaas来分配，Caas只是资源的使用方。

好处：
完全可控，物理使用率比较高

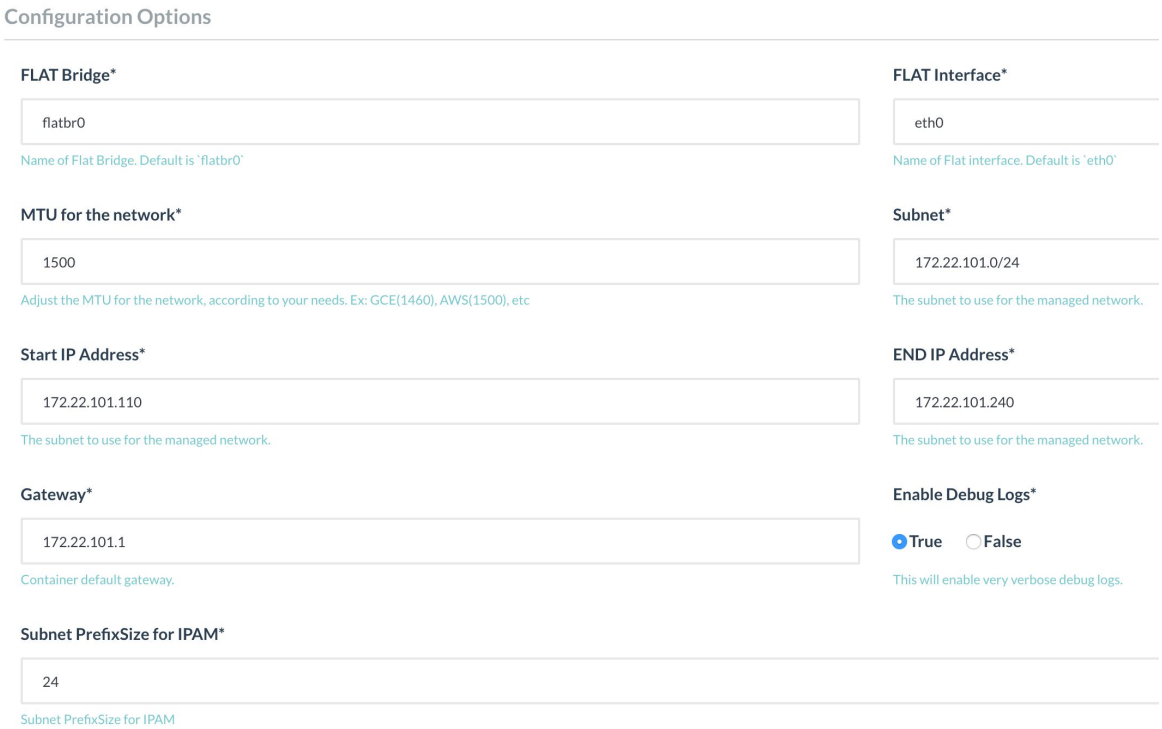
劣势：
需要用户本身有较高的水准，使用起来相对困难。

代表：
Kuryr(OpenStack)
结合CNI/CNM进行自定义

以Rancher为例（Caas内自治）



以Rancher为例（依托Iaas）

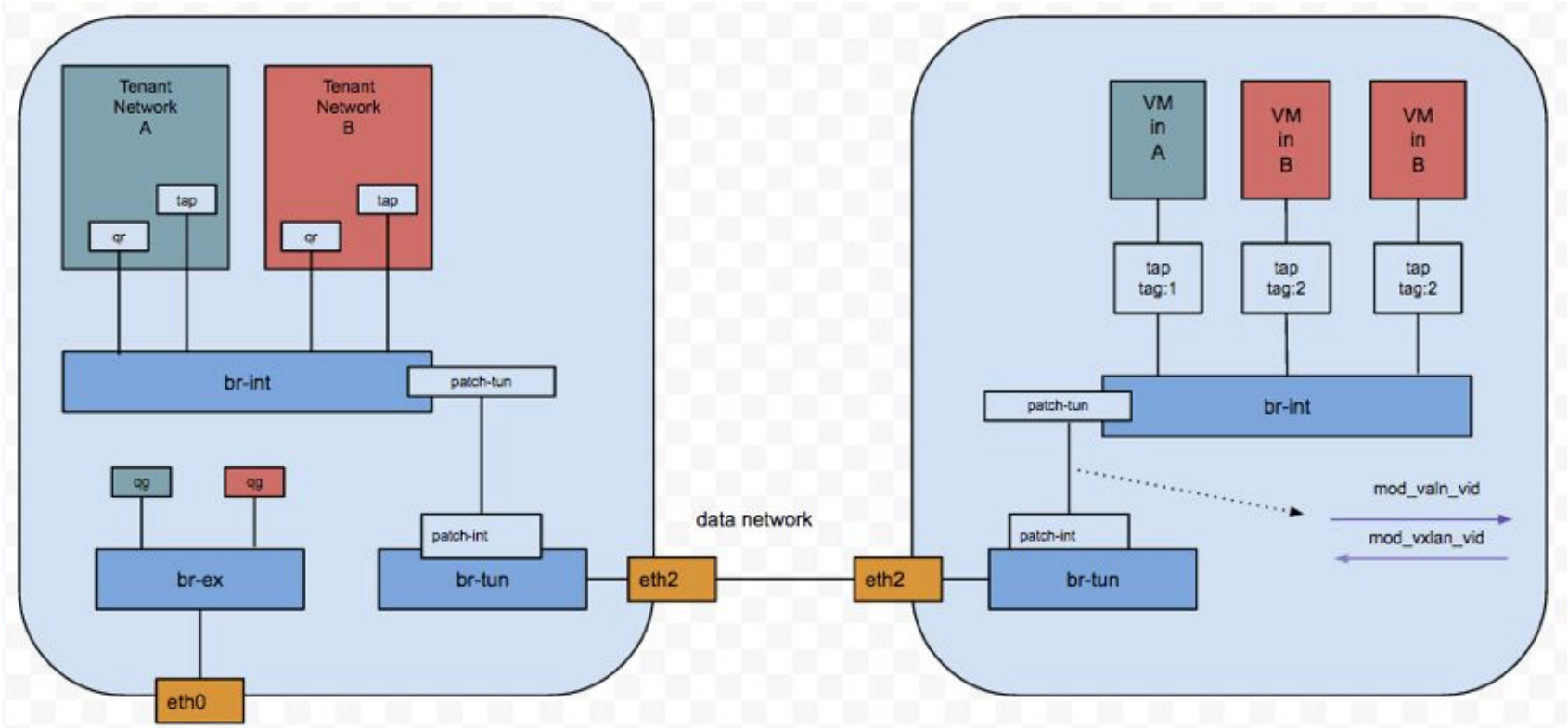


来自社区与用户的声音

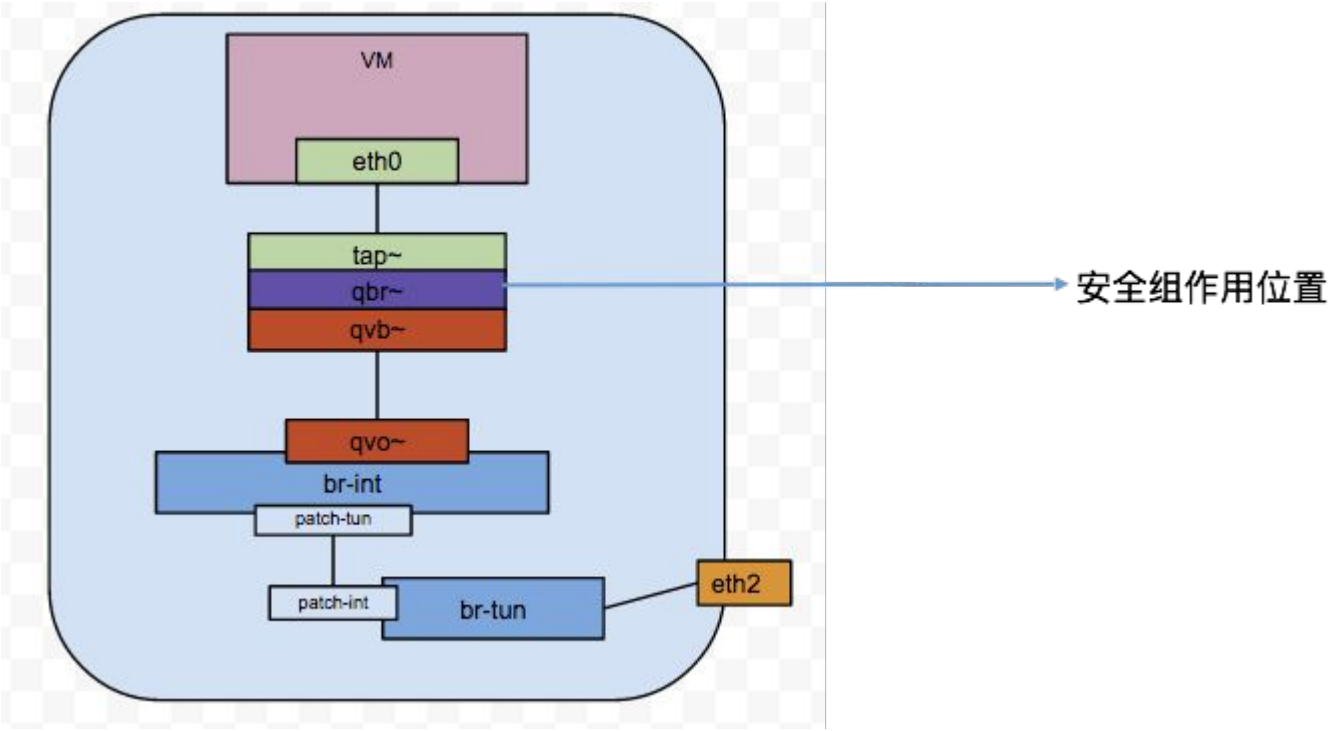
1. Caas还在不断演化中，用户还没有形成固定模式的使用习惯，Iaas上的遗留习惯依然会保留
2. 一容器一IP，很多时候还是像虚拟机一样使用容器
3. 如何构建高性能Caas网络
4. ...

与Iaas网络融合的解决之道
Rancher & OpenStack

OpenStack VM网络



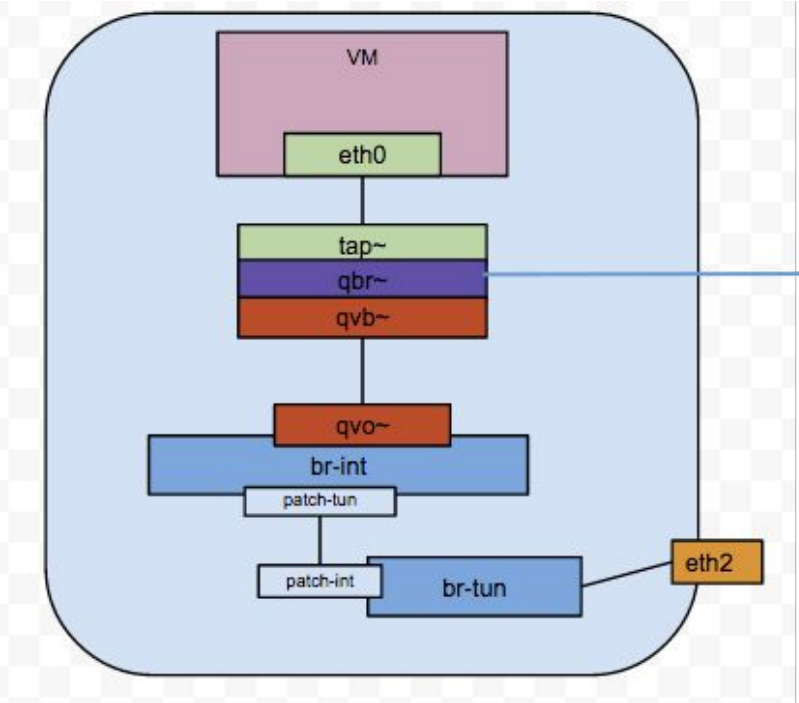
OpenStack Security Group



Caas网络痛点

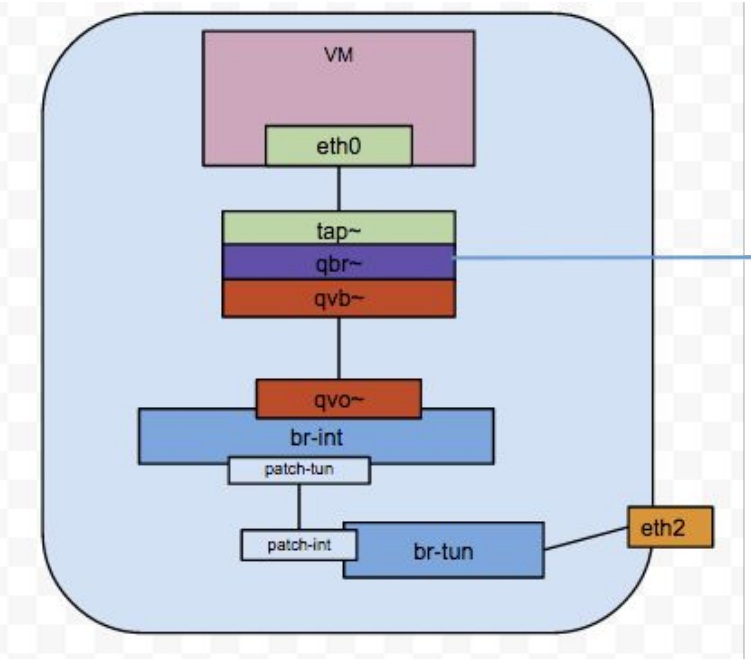
1. 需要嫁接在Iaas网络之上
2. Overlay on Overlay损耗巨大
3. 构建与Iaas subnet在同一网络中，需要“伪造”IP和MAC
4. “伪造”IP和MAC数据包默认不被通过

Disable Port Security



1. 安全规则accept all
2. 所有流量全通，所以安全组失效

Port Address Pairs

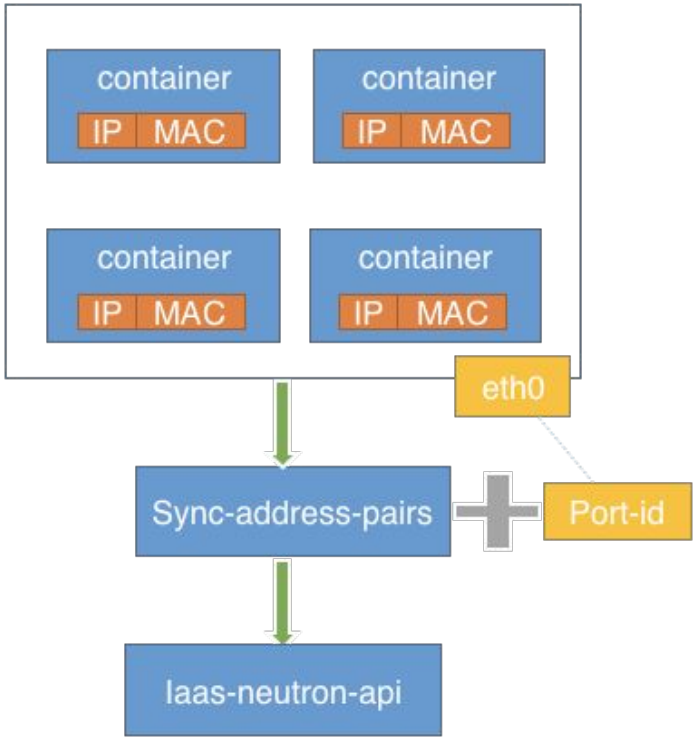


1. 默认只通过VM中eth0的ip和mac流量（anti-spoofing）
2. Address pairs就是可以额外添加允许通过的ip和mac

```
# neutron port-create net1 --allowed-address-pairs type=dict list=true
mac_address=<mac_address>,ip_address=<ip_cidr>
```

```
compute1:~$ iptables -L neutron-openvswi-sebf586a5-a -n -v
Chain neutron-openvswi-sebf586a5-a (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 RETURN     all  --  *      *       10.0.1.100           0.0.0.0/0            MAC FA:16:3E:42:A1:D4
   47     0 RETURN     all  --  *      *       10.0.1.10            0.0.0.0/0            MAC FA:16:3E:42:A1:C4
    3   252 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

容器扁平网络嫁接方案(Rancher)



方案一：

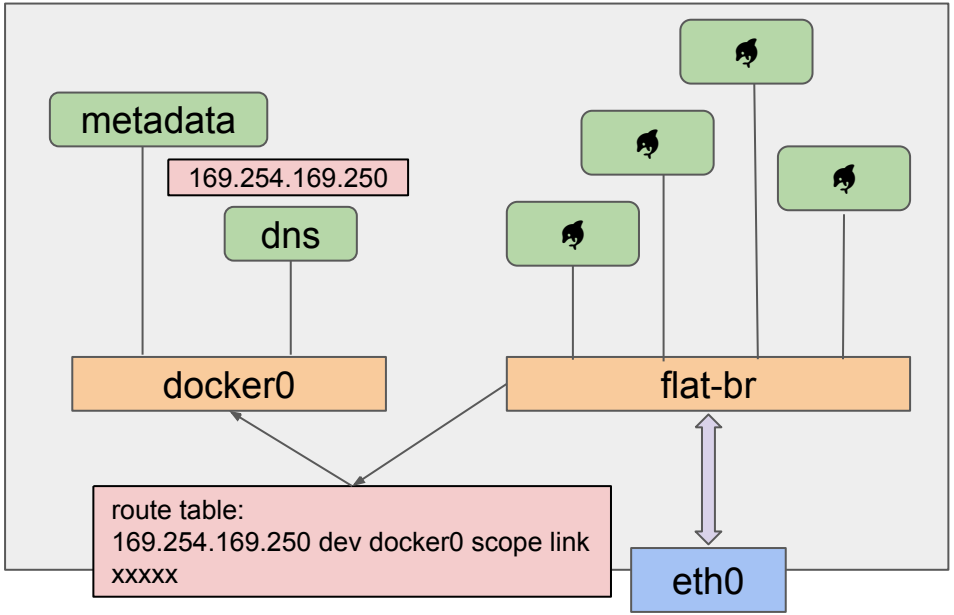
Caas内的服务程序自动将容器的ip&mac动态的设置到laas port中（手动测试验证已经通过）：

1. laas 的api能否支持这种频繁调用是个瓶颈
2. 每个laas port支持多少address pairs是个关键

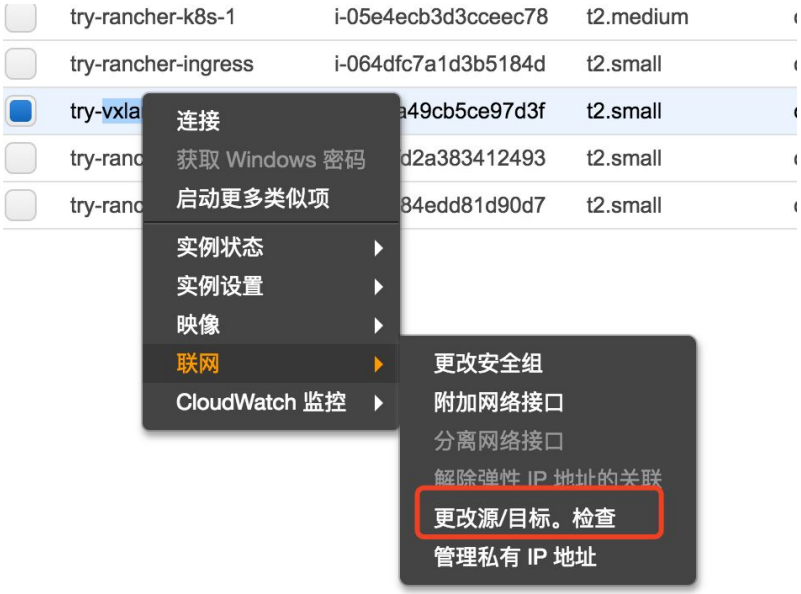
方案二：

laas层面能够支持安全组存在的情况下，可以伪造IP和MAC，目前OpenStack的原型并不能很好的支持，需要laas定制

虚拟机内部网络结构

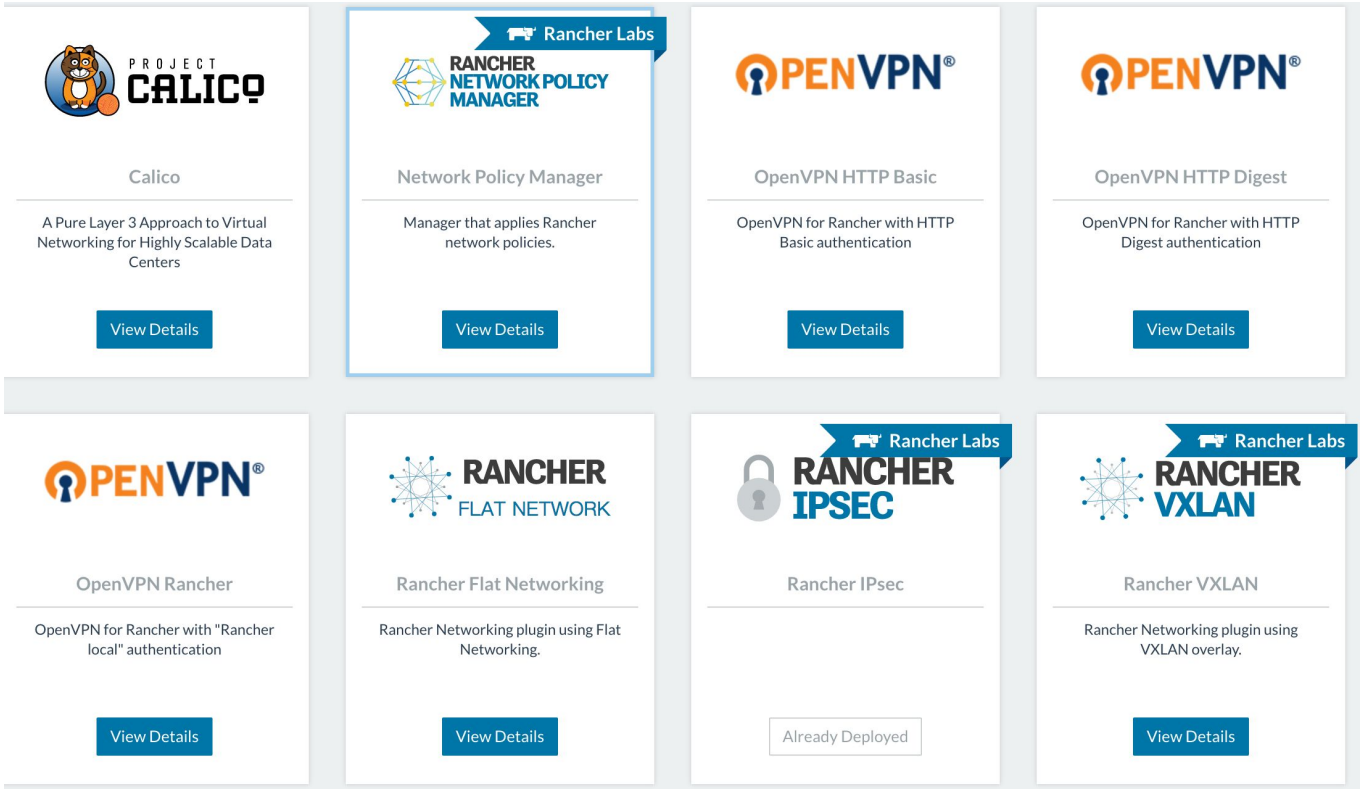

metadata
169.254.169.250
dns
docker0
flat-br
route table:
169.254.169.250 dev docker0 scope link
xxxxx
eth0

AWS的关键设置



Disable src/dst check

Caas网络插件式管理



Q&A