**Postman：认证**

X-Auth-Token:（通过浏览器开发者模式里面取，打开galaxy或者其他页面需要token验证的网页获取）

XZRnK6y2%2BI17p86OuJ24JDZiYkuFRjov58kbb7PmalTpkBwOlN21Dky1LYthZ6%2BYgQ%2B0GAJeMlkE%0ArcWnhVddtWKVqywIDldnPd0P3rczPhvKt0IJQiDA4de8WJKVK0hf2058eiRBE5fIPy%2FovFWv2Fvh%0AuyV9gj2fdHghwHnVZTk%3D%0A

一是在header里添加X-Auth-Token；二是在cookie里面添加Auth-Token

**新建云实例（aws ap节点）**

{

"cloud\_type" : "aws",

"region" : "[ap-southeast-1]Singapore",

"instance\_type": "t2.medium 2core 4GB",

"eth0": {

"pip\_num": 1, #private ip

"eip\_num": 1, #external ip

"add\_ip\_mapping": false #galaxy生成映射与否（true生成）

},

"eth1": {

"pip\_num": 5, #与vip数量一致

"eip\_num": 5,

"add\_ip\_mapping": true

},

"quantity": 1, #创建实例得个数

"ebs\_size": 20 #实例磁盘容量

}

**Proxy机器**(proxy无特殊情况下一般为单网卡实例，网卡上的从IP和EIP数量与VIP数量相等)

{

"cloud\_type" : "aws",

"region" : "[ap-southeast-1]Singapore",

"instance\_type": "t2.medium 2core 4GB",

"eth0": {

"pip\_num": 5,

"eip\_num": 5,

"add\_ip\_mapping": true ###galaxy生成映射

},

"quantity": 1,

"ebs\_size": 20

}

**新建群组**

{

"role": "proxy",

"gid": "40001", ##对应ap，如果不指定gid的值，pathn会自动分配可用的值

"proxy\_id": "40001", ##ap的这个值，需要填其所属的proxy的id，不是自身的ap\_id

"name": "Japan-Tokyo-aws-proxy",

"isp\_id": "9", ###网络类型，9表示海外配置有内网的aws

"area": "tokyo",

"machine\_ids": [186258,186259]

}

ap机器状态上传接口：<http://10.203.5.247/api/v3/ap_agent_infos>

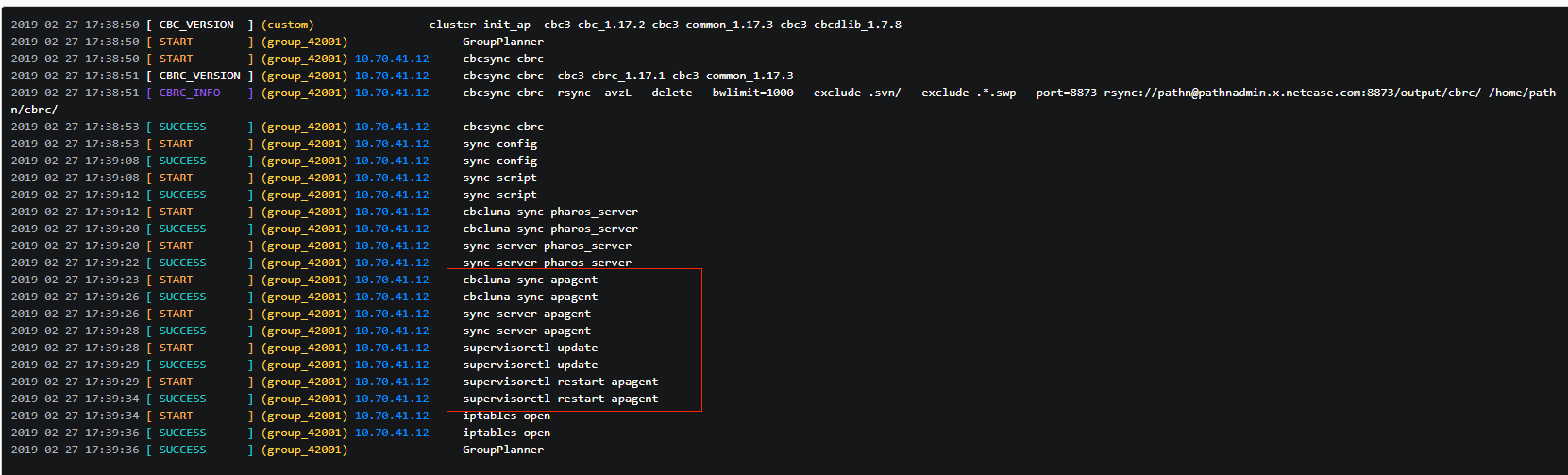
**创建云实例（postman）create\_instances（ap/proxy） ------->创建galaxy(ap)群组（postman）create\_galaxy\_group------>群组初始化（galaxy生成配置，同步，更新）------>入库，post------proxies/aps------>机器初始化：安装puppet，安装nft（检查是否已安装nft -v,aws实例模板已经安装nft模块，不需重复安装），确认puppet安装完成后reboot而后执行puppet agent -t；阿拉丁初始化群组----->检查proxy的nft（（检查sudo nft list ruleset）阿拉丁初始化后生效），ap机器状态ap\_agent\_infos，检查agent info和agent backend等能否正常输出。**

ap入库后设置meta mark；指向杭州的ap需要设置静态路由，设置走pathn内网100.\*.\*.\*

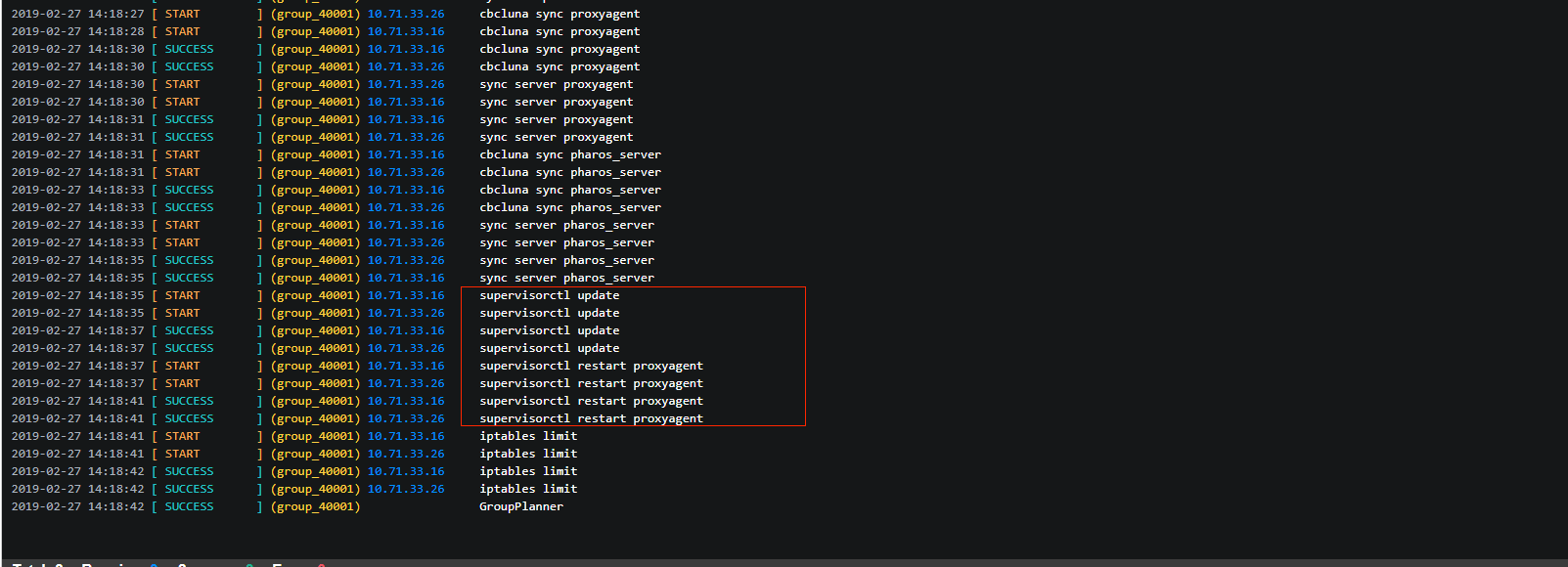
对于旧集群, 新上线proxy机器, 需要使用 sync 接口强制同步历史数据

国外aws之间修改起走外网连接，目前新建节点默认走内网（nft list ruleset中chain prerouting里面的dnat部分），修改后端配置PUT /aps/{gid}/agent\_config，设置ip\_type的值。

阿拉丁初始化ap群组，可以更新apagent代码，并重启apagent:



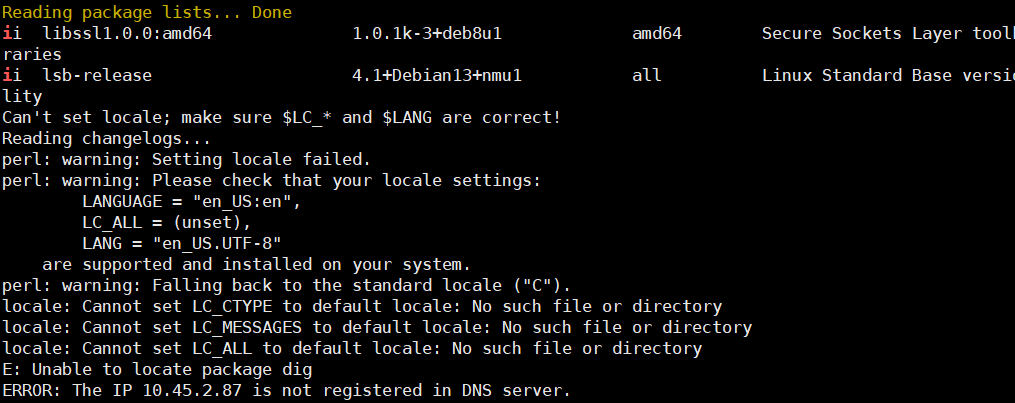
初始化proxy集群，同步代码，重启proxyagent：



问题一：阿拉丁初始化proxy出错，可能为机器的防火墙问题，echo '10.202.4.227 pathnadmin.x.netease.com' >> /etc/hosts后正常（暂时的修改，host会被puppet再次修改）。

问题二：在旧的ap群组下创建实例，galaxy上显示状态为离线，阿拉丁上找不到对应的群组。需要修改galaxy状态为在线，之后可以在阿拉丁操作ap初始化。（昨天有个操作失误，在下线ap42001群组的时候，设置了ap离线状态，之后下线机器，回收实例，没有操作删除galaxy上面ap群组（ap42001群组还在，但是是离线状态，没有绑定机器）；后面再操作上线流程，通过api群组创建，用的还是旧的ap群组名称和参数；之后galaxy上42001显示有新的机器；但是阿拉丁上找不到这个42001群组）为避免这个情况，需要按照下线流程把ap删除，URL：DELETE /aps/{gid}。

初始化网络测试节点，安装puppet报错：



处理：sed -i 's/# zh\_CN.UTF-8 UTF-8/zh\_CN.UTF-8 UTF-8/g' /etc/locale.gen

sed -i 's/# en\_US.UTF-8 UTF-8/en\_US.UTF-8 UTF-8/g' /etc/locale.gen

locale-gen

wget -O initpuppet.sh http://puppet.x.netease.com/installpp/initpuppet.sh && sh initpuppet.sh

如果puppet agent -t继续有报错，先跑apt update 和apt upgarde，再重新跑命令，报错重新再跑。

## 同步代码

cbc hostlist/cluster-330xx cluster init\_probe # 探测机器

在impression管理机操作cbc,330xx对应机器的hostname：

指令：cbc hostlist/cluster-330xx(详细机器名称) cluster init\_probe （管理机上tab自动填充时没有显示33009机器，奇怪）

或者在阿拉丁直接走“初始化网络测试节点”流程。（33009)

 cbcsync cbrc rsync -avzL --delete --bwlimit=1000 --exclude .svn/ --exclude .\*.swp --port=8873 rsync://impression@impressionadmin.x.netease.com:8873/output/cbrc/ /home/impression/cbrc/ ##日志显示的同步操作，将管理机的cbrc目录内容同步到33009下的/home/impression/cbrc目录

--bwlimit=1000 #限制传输带宽1000kps

--exclude .svn/ ##排除.svn/目录

测试节点上线问题：

1. ap集群 需设置 meta mark 标记：需要在ap入库后才能设置（put）
2. 阿拉丁找不到群组，需要galaxy同步配置;阿拉丁cbc成功跑完后proxy机器的nft才会正常输出内容。
3. 测试环境中arm（40005）使用puppet4，删除puppet3：aptitude purge -yP puppetall-3.4.3 && rm -rf /usr/local/puppetall-3.4.3 /var/lib/puppet;

其他使用puppet3

40003-01安装puppet4失败，之后重启机器监控系统gpost丢数据， /etc/init.d/gpostd restart

日志目录cd /var/lib/gpostd/log/，查看错误日志tail -f gpostd.stderr.log

最后一招，skyline关机重启漂移宿主，40003-01至此恢复。再不行就下线再创建。

1. arm实例puppet有对应安装包，下载链接(文件名)不一样。aws机器模板已经部署nft，不需要再重复安装。

# arm架构的机器

wget -O initpuppet\_pp4.sh http://puppet.x.netease.com/installpp/initpuppet.sh.pp4.arm

~~arm模板中实例的cbrc未安装，需要手动安装并指定版本：~~

~~pip install --upgrade -i https://pip.nie.netease.com/simple/ cbc3-cbrc==1.17.1~~

~~cbcbase手动安装~~

~~pip install --upgrade -i https://pip.nie.netease.com/simple/ cbcbase~~

后面模板已修复

正式环境节点问题：

1.创建的arm proxy实例，再阿拉丁上初始化一直报错，待解决。

2.~~arm不支持两个实例一起生成，quantity目前只支持1个；~~偶发异常，后面正常

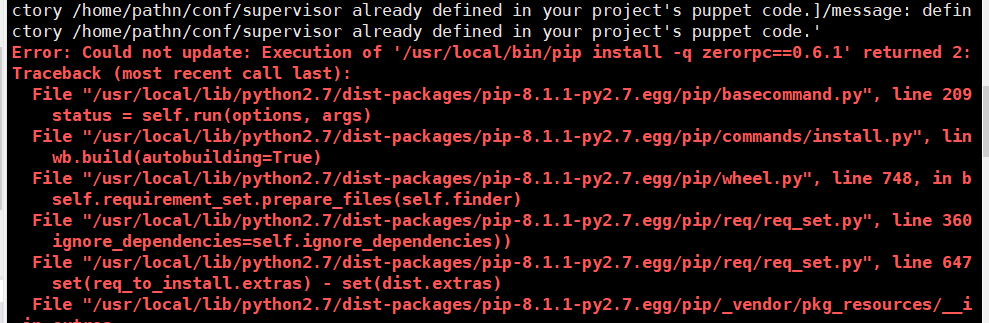
3.爱尔兰aws内网段不同，savpn申请添加新网段（加完网段再不通再找SESA值班检查一下aws的vpc设置）

**4.arm节点要在galaxy加一个arm标签：标签---关联--arm**

arm的实例要手动安装gpost，以后初始化arm实例的时候也跑一下这条命令：

pip install -i https://pip.nie.netease.com/simple gpost

Puppet agent -t 报错



升级

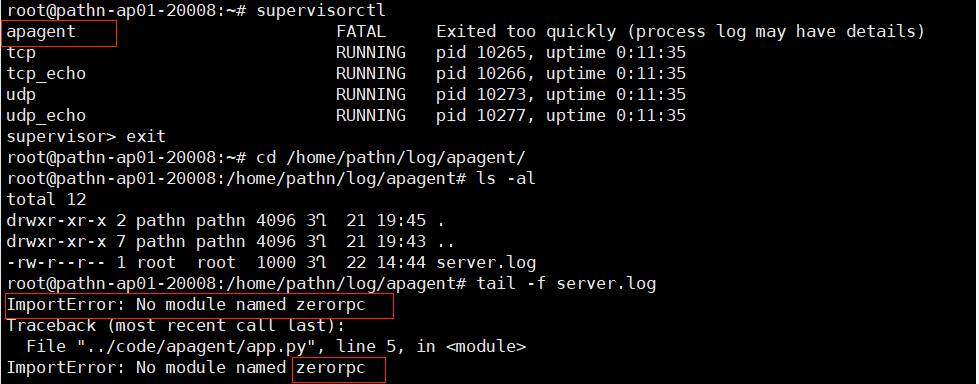
pip install pip==9.0.1 -U

apt install cl-cffi

手动安装zerorpc

pip install -q zerorpc

arm实例排查



1. 周四创建日本proxy10101，其ap20001-20008，初始化失败节点20007和20008arm实例，阿拉丁初始化失败，待解决。杭州ap实体机，未创建。
2. 周五创建新加坡，佛吉尼亚，法兰克福：

新加坡proxy10201创建完成，节点ap

**创建节点后，agent info没输出，接口ap\_agent\_infos没有ap数据：**

**数据：应该是已经有Nftables 列表了. 但是重启了apgent 进程. 这个时候, 不会进行配置, 除非upgrade。**

**处理：agent upgrade**

1. 指向杭州10001的节点20029--20036（一开始10001未入库，post节点显示proxy10001不存在，proxy10001正常入库后，显示维护状态离线，阿拉丁初始化nft list ruleset没输出，agent info/backend无输出，agent upgrade报错，继续修改为在线可用状态，Galaxy更新和同步后，agent upgrade正常后ap操作正常）

阿拉丁初始化后需要添加静态路由

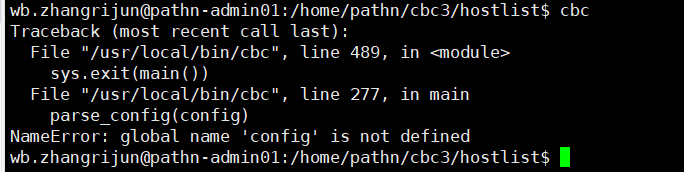
安装puppet后一般要确认finish再重启，tail -f /var/log/syslog ，在确认跑完一次catalog后再重启，不然重启过后，puppet agent -t 时会显示有puppet applying进程，需要等待。

四、apcheck(部署ap后进行验证)

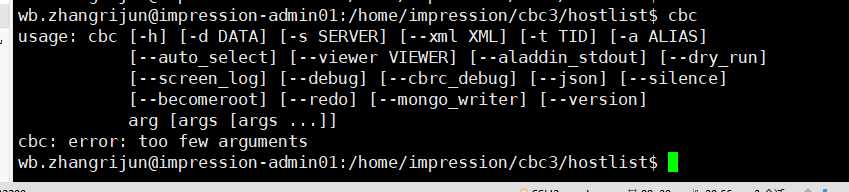
cbc xxx cbcluna sync apcheck

cbc xxx sync server apcheck ##同步到ap节点机器（~~pathn管理机运行cbc会报错，阿拉丁上调用初始化ap及proxy的时候正常，无解~~操作错误导致，cbc执行需要进入到对应目录/home/pathn/cbc3/，不是全局指令，cbc的初始化和同步操作需要在项目用户pathn或impression下执行，不然会有报错。）

pathn管理机下运行cbc报错：



Impression管理机：



这个工具一般在管理机上跑：/home/pathn/Server/apcheck，

如果管理机跟ap网络不通再找别的节点

4月3号台湾gce节点

SRE|黄志雄|大雄 4月2日 17:15:54

{

"cloud\_type": "gcp",

"region": "[asia-east1]GC\_Taiwan",

"instance\_type": "n1-highcpu-8 8core 8GB",

"gcp\_nic\_num": 4,

"quantity": 1,

"ebs\_size": 20

}

SRE|黄志雄|大雄 4月2日 17:16:37

"gcp\_nic\_num": 4, 这个是设置网卡个数的参数，包括主网卡

==============

@CDN|张日军 今天麻烦搭2个 台湾gcp的 AP集群,归属proxy 10001 实例采用 n1-highcpu-4 4core 4GB 4个网卡.

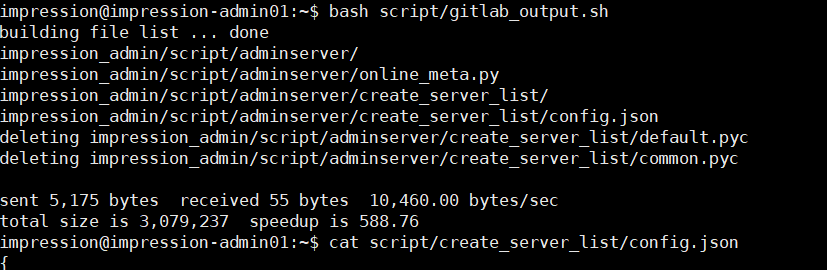
ap 的gid字段可以留空，pathn自动分配可用gid，如果再次指定已经使用过的gid，会把之前的群组覆盖。

1. **项目修改别名：https://g.126.fm/02VUwc8列表控制指引**

管理机上直接修改config.json是临时性操作，会被git的同步覆盖，需要在git上修改，然后进行同步覆盖到管理机。

路径：impression\_admin/script/adminserver/create\_server\_list下的config.json,通过edit进行修改，然后commit（加上commit说明），之后提交merge请求，等待通过。

最后在管理机上执行 bash script/gitlab\_output.sh 同步代码。



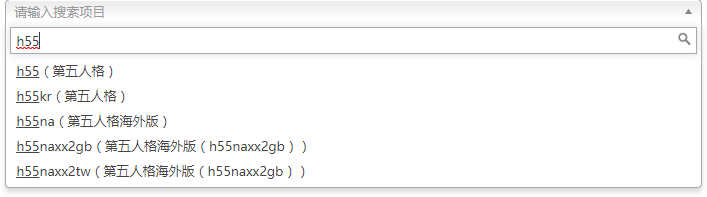
Gitlab的key说明：

1.git-gui生成的key导入到GitHub和gitlab后，能通过ssh克隆远程库

2.Xshell的key登记后，用于xshell登录其他机器（公钥在workflow登记后会同步到所在项目下的机器中，实现key的录入，之后才能ssh远程登录）。

3.虚拟机的key(存放路径为/root/.ssh下)导入到github后，能连接到github的远程库，这里不是使用xshell生成的公钥，是虚拟机下debian9生成的key。

别名设置：游戏在不同地区分区不同的包，对应的项目代号不同，如h55包括h55na表示国际版,h55naxx2gd表示全球版,主要有三种发行包，台湾，韩国，以及除了大陆地区及台湾韩国以外的全球包（h55na和h55naxx2gd在加速列表中代表同一个意思，互为别名）。



？？？？？：接入灯塔后，客户端判断所在region，去获取加速节点的ip和端口，这个跟包版本有无联系？台湾的包是不是优先去取台湾的节点，如果台湾的包在韩国地区进入游戏，又是如何处理。或者灯塔不管什么发行包，判断region后，拉取对应region的节点信息，那列表的别名是否有意义？？？？？待解惑

1. **节点初始化及端口检查，网络检查**
2. 回杭州的节点（走内网）：国内----海外，也是走内网

端口检查用apcheck；

网络检查：走内网，cat /home/pathn/conf/backend.conf查看后端ip，traceroute ip 和mtr -n ip；检查跳转是否是走中间节点100.66.\*.\*的pathn内网；杭州机房节点走机房内网，不需要走pathn内网。

1. 海外-海外节点（走外网）：

端口检查同上；

网络检查：nft list ruleset看chain prerouting中dnat是否为外网；接口get {{v3}}/aps/20009看ip\_type是否为10001

**Arm实例20007-20008等apcheck报错：**

ap端口检查：

[ERR] 20007: {'udp': [(u'63.35.123.225', 7001), (u'99.80.189.209', 7001), (u'99.80.215.163', 7001)], 'tcp': [(u'63.35.123.225', 7002), (u'99.80.189.209', 7002), (u'99.80.215.163', 7002)]}

[ERR] 20008: {'udp': [(u'63.32.24.249', 7001), (u'63.35.244.247', 7001), (u'99.80.196.35', 7001)], 'tcp': [(u'63.32.24.249', 7002), (u'63.35.244.247', 7002), (u'99.80.196.35', 7002)]}

[ERR] 20011: {'udp': [(u'99.80.190.34', 7001), (u'99.80.210.103', 7001), (u'99.80.236.96', 7001)], 'tcp': [(u'99.80.190.34', 7002), (u'99.80.210.103', 7002), (u'99.80.236.96', 7002)]}

[ERR] 20012: {'udp': [(u'63.32.186.81', 7001), (u'63.34.239.34', 7001), (u'63.35.238.204', 7001)], 'tcp': [(u'63.32.186.81', 7002), (u'63.34.239.34', 7002), (u'63.35.238.204', 7002)]}

[ERR] 20019: {'udp': [(u'63.35.253.180', 7001), (u'99.80.151.17', 7001), (u'99.80.217.245', 7001)], 'tcp': [(u'63.35.253.180', 7002), (u'99.80.151.17', 7002), (u'99.80.217.245', 7002)]}

[ERR] 20020: {'udp': [(u'99.80.142.18', 7001), (u'99.80.201.75', 7001), (u'99.80.216.127', 7001)], 'tcp': [(u'99.80.142.18', 7002), (u'99.80.201.75', 7002), (u'99.80.216.127', 7002)]}

[ERR] 20027: {'udp': [(u'3.209.24.167', 7001), (u'3.92.90.69', 7001), (u'52.86.126.104', 7001)], 'tcp': [(u'3.209.24.167', 7002), (u'3.92.90.69', 7002), (u'52.86.126.104', 7002)]}

[ERR] 20028: {'udp': [(u'100.25.122.194', 7001), (u'3.209.138.92', 7001), (u'34.226.120.118', 7001)], 'tcp': [(u'100.25.122.194', 7002), (u'3.209.138.92', 7002), (u'34.226.120.118', 7002)]}

估计是镜像没有关闭 rp\_filter，检查一下 ap 节点的 rp\_filter（reserve path filter反向路径过滤）：

for i in /proc/sys/net/ipv4/conf/\*/rp\_filter; do echo 0 > $i; done

echo "net.ipv4.conf.all.rp\_filter=0" >> /etc/sysctl.conf

echo "net.ipv4.conf.default.rp\_filter=0" >> /etc/sysctl.conf

或者设置标签值：（单单修改标签值好像没效果，apcheck仍然报错）

{{pathn}}/aps/2xxxx7/agent\_config

{

"nft\_mark": {

"origin": "0x00000011",

"reply": "0x00000011",

"enable": true

}

}

**设置标签，**主要作用于专线的节点，使流量从bond0进来，然后从bond1出去；对于aws这些节点，流量转换是在同一张网卡eth1上（eth0作为管理ip，不参与流量转发），不需要打标签，enable的值可以设置为false；

之前创建的节点，都统一使用了orgin：0x00000010和reply：0x00000011这个标签值，如果这么设置，需要rp\_filter开启，且会有性能瓶颈。rp\_filter默认关闭。

即rp\_filter参数有三个值，0、1、2，具体含义：

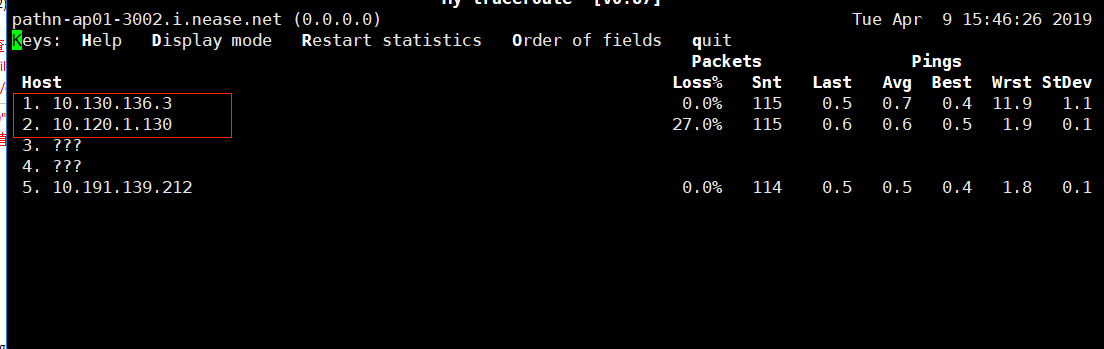
0：不开启源地址校验。

1：开启严格的反向路径校验。对每个进来的数据包，校验其反向路径是否是最佳路径。如果反向路径不是最佳路径，则直接丢弃该数据包。

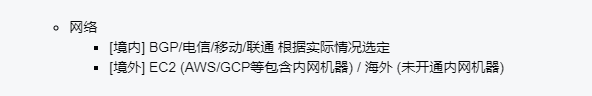
2：开启松散的反向路径校验。对每个进来的数据包，校验其源地址是否可达，即反向路径是否能通（通过任意网口），如果反向路径不同，则直接丢弃该数据包。

20019修改过后端口检查还是报错，原因是路由表10001默认路由没加上，估计是创建后没有重启过，导致默认路由没加上。

中港节点20037和20038，因为宿主机直接在杭州机房，走的是机房内网，没有走pathn另外搭建的内网100.\*.\*.\*的段：



**20037和20038（中港节点）等专线机器，阿拉丁初始化失败，是因为这些机器没有内网ip，或者有内网ip（专线机器），但这些内网与杭州机房不互通，网络类型应该为海外**，不同于aws/gcp网络类型的EC2，这些内网与杭州内网互通，需要修改galaxy节点的网络类型，之后生成配置，同步。



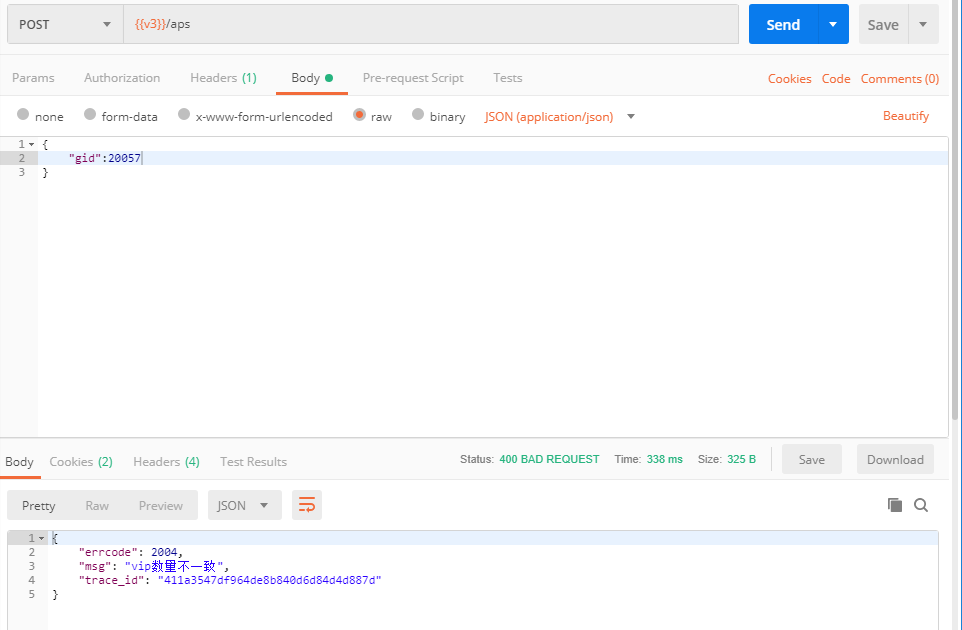
**专线节点机器多为虚拟机，多个节点共享一个实体机，登录一个节点即可验证走内网情况，查看backend.conf看后端ip，分别做mtr和traceroute验证。**

**20041-20042-20043（宿主为pathn-ap01-60006）,cat /proc/sys/net/ipv4/conf/\*/rp\_filter中bond0和bond1中的rp\_filter值为1，做了修改，暂时没改/etc/sysctl.conf,跟pathn-ap02-60006一样，目前这个文件为空，没有内容。**

**前面的节点机器，,cat /proc/sys/net/ipv4/conf/\*/rp\_filter的值都为0，但部分机器/etc/sysctl.conf是空，没有内容（20009,20010）。**

**原因：有另外的配置文件，/etc/sysctl.d/00-xx，里面有对应配置信息，所以不影响。**

**20057入库显示vip数量不一致：**



**需要把galaxy上20057节点的映射删除，外网ip设置为主ip使用中，内网ip设置为备用；delete节点后重新导入；这种实体机，登录进去可以看到内外网ip；如果有映射，入库时会把映射里的ip都读到bind\_infos中；aws和虚拟机ap（韩国互联港湾等专线ap）使用后一种方式，机器导入映射，在机器上只能看到内网ip。**

**10003入库显示vip数量不一致：**

**Proxy下10003有两台机器，**pathn-proxy01-10003有两个映射（一个vip，一对内外网ip组成映射），pathn-proxy02-10003没有映射，在galaxy上添加映射后导入正常。

**20055入库vip数量不一致：**

**Vip数量为1，但galaxy在用状态有两个ip，需要把其中一个设置为备用：**

****

**202.153.181.21为业务ip，设置为在用，103.105.134.112为管理ip，设置为备用。**

**20055端口测试不通，因为ip202.153.181.21，只允许本地ip访问，外部ip访问不了（犀思云用于解决专线攻击的方案）。**

**加了一个台湾测试节点：- 70002 Taiwan-gcp-echo**

**pathn-probe01-70002 35.234.24.55 10.78.30.13**

**台湾犀思云的节点通过70002去测试端口。**

**场景：适用于客户端无法接入灯塔的情况下适用，该方法直接将映射生成到 serverlist 中，一般情况下不推荐。g103目前使用这种方式。**

****默认情况下，外部数据源不会返回具体分配数据给项目，如果需要使用这个方法，请联系PathN的SA开通****

开通之后，项目 SA 可以在模板中通过一个数据项获取到机器 IP:PORT 在各个区域对应的映射关系

通过$EXT\_DATA.EXT\_PATHN\_PROXY来调用

$EXT\_DATA.{外部数据源名称}