**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，vip为5

},

"quantity": 1,

"ebs\_size": 20

}

pathn-ap01-20060，只有一张网卡eth0，含有4个pip，1个eip，在galaxy有一对映射，对应的vip数量为1.（对应参数可能是：

"eth0": {

"pip\_num": 4,

"eip\_num": 1,

"add\_ip\_mapping": true ##galaxy生成映射

},

）

######################################################################

**Proxy10102的IP地址需求: 主Ip eth0 5个EIP, eth1 1个EIP, 将eth1 的IP入库 pathn（导入pathn时，关键参数是ip的数量，以及是否生成映射）**

{

"cloud\_type" : "aws",

"region" : "[ap-northeast-1]Tokyo",

"instance\_type": "c5.large 2core 4GB",

"eth0": {

"pip\_num": 5,

"eip\_num": 5,

"add\_ip\_mapping": false ##galaxy不生成映射，不会导入到pathn

},

"eth1": {

"pip\_num": 1,

"eip\_num": 1,

"add\_ip\_mapping": true ###galaxy生成映射，导入pathn，vip数量为1；如果

##是eth0生成映射，会被导入到pathn，vip数量会是5个。

},

"quantity": 2,

"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，nft的检查，对应proxy新增的机器，需要接口触发postman指定机器同步映射才会同步到机器上。多网卡机器，检查rp\_filter状态，修改为0，apcheck测试，设置ap上线/备用/维护状态。**

**群组扩容新增机器：proxy群组接口**{{v3}}/toolbox/create\_galaxy\_group**接口新增machine id，之后post，重新post一次proxy，之后可以get获取到proxy所有机器，新机器会显示为3维护状态；进行puppet安装，阿拉丁初始化，同步端口映射，测试端口正常后再修改机器状态，设置上线。proxy机器上下线不需要对列表进行刷新。**

**群组缩容/故障机器下线：修改机器状态0下载，如果是ap，需要刷新对应列表**{{v3}}/portmaps/self\_trigger**指定proxy或者项目刷新。**

**机房物理机proxy机器nft安装：**

**cd /home/pathn/script && bash nft\_install.sh**

**Debian10 跑脚本无法安装，估计是源的问题，手动安装路径/home/pathn/script/package**

**dpkg -i libnftnl\_1.1.4-20190830\_amd64.deb**

**dpkg -i nftables\_0.9.2-20190830\_amd64.deb**

**pathn上线新环境：**

**创建云实例（创建机器暂时使用test环境），使用tasks里面的创建实例接口,群组直接在galaxy上手动创建，直接把机器关联到对应的服务即可；根据gid导入机器后，post创建cluster，获取到cluster的id，创建虚拟群组，使用vgroup里面的创建ap和proxy，这里需要用到cluster\_id和gid。**

11003集群

**proxy**

**a1.large 2core 4GB，2台，单网卡，10个EIP（使用主ip）## "use\_main\_ip": true**

**ap**

**a1.large 2core 4GB，2台，双网卡，每个网卡10个EIP（不使用主ip）## "use\_main\_ip": false**

**galaxy上新建群组后添加机器，通过服务和ip筛选机器，机器名会被命名为项目（pathn）加上绑定的服务（proxy或者ap），后面接上群组的代号，比如**pathn-ap01-15002，pathn-proxy01-11001

**创建cluster(公共集群)**

**{**

**"name": "public-frankfurt-cluster",**

**"region": "frankfurt",**

**"description": "欧洲公用集群群",**

**"vip\_num": 1,**

**"projects": [], ###未替换前先把projects置空，如果添加项目，会报错 - 消息：项目 g78na,g37,h50,g103,g78,pathn 在区域 tokyo 已有转发集群**

**"enable\_proxy\_entry": false,**

**"only\_mapping\_internal\_ip": false,**

**"status": 1,**

**"region\_configs": [**

**{**

**"region": "cn"**

**},**

**{**

**"region": "jg"**

**},**

**{**

**"region": "sg"**

**},**

**{**

**"region": "us",**

**"ap\_regions": ["virginia"]**

**}**

**]**

**}**

###创建proxy

{

"cluster\_id": "c7",

"role": "proxy",

"isp\_cls": 0,

"preference": {

"gid": "11007" ###关联gid，群组内有ap和proxy，不会相互干扰

},

"status": 0

}

####创建ap

{

"cluster\_id": "c7",

"role": "ap",

"exchange": false,

"region": "virginia",

"projects": [],

"isp\_cls": 0,

"preference": {

"gid": "11005"

},

"status": 0

}

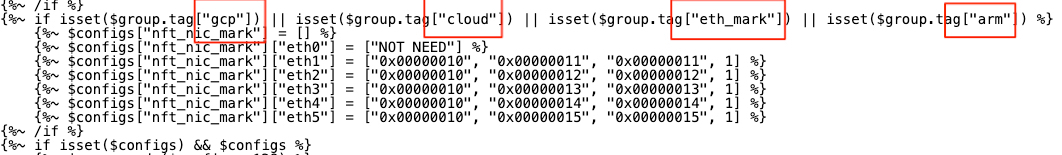
##apcheck 使用a\*\*的节点名称

端口不通时进行以下检查：

双网卡实例先检查rp\_filter：sysctl -a | grep rp\_filter

检查nft\_mark，专线节点ap和ep需要检查mark，双网卡检查;云机器需要给群组加对应aws，arm，gcp标签，会生成对应ct\_mark标记。##40011自动加上mark，后面的好像没加？手动加的mark还未撤销，撤销了40006上a7的标记，结果又不通。待查验证。##使用台湾gcp新建40011-02测试。

**打 aws/gcp/arm 标签**



为配置模板 /config/{%$host.ip%}.apagent.conf里面内容。

存放路径为ap机器的/home/pathn/conf/apagent.conf

**台湾专线入口ap12009上面不设置mark**，因为eth0为119.31.177.213(a112)和119.31.177.211(a111),流量从eth0进来，dnat改为10.79.10.37后走内网eth1到后端ep(a110),流量从ep返回来时从eth1进，走默认路由eth0返回到客户端，形成环路，所以不用mark。

a110@12009-01的eth1为内网，eth0为电信，eth2为联通，机器12009上设置了tunnel，通过tunnel控制流量从eth1进来最后再从eth1返回a111和a112？

**10102下线流程，cluster\_id:oc-10102:**

通过{{pathn}}/vgroups?cluster\_id=oc-10102查看集群上面创建的vgroup，先删除虚拟群组，报错码1017 400 有其他的虚拟群组的后端配置指向该虚拟群组，不能直接删除，先从AP的agent\_config里移除相应后端；

删除集群/api/v3/clusters/oc-10102，错误码：1014 - 消息：集群 Cluster(id='oc-10102') 仍有映射, 不能删除；

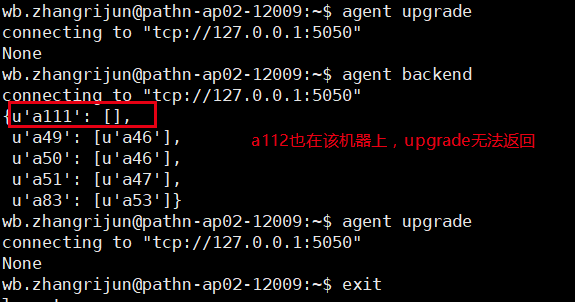
使用portmap接口指定cluster使映射过期（"expire\_days": 0），并立即回收映射；之后可以删除cluster，最后通过机器id删除机器。

查询节点id：由gid在galaxy查ip，通过ip在{{pathn}}/ip\_configs?ip=45.40.60.13，返回backend和 "vgroup\_id": "a2",

通过machine\_id查询机器上的vgroup：get {{pathn}}/machines/135638/vgroups

**ap exchange的状态要设置为2**

台湾节点a111和a112，指向福州中转节点a110，由于a110状态设置为3,导致不可用，端口测试a111和a112不通，且宿主机12009-02上无法upgrade：



台湾，韩国，日本专线的ap一般需要设置ct\_mark

{

"nft\_mark": {

"origin": "0x10",

"reply": "0x14",

"enable": true

},

"nft\_ct\_mark": "0x1002", ####根据线路机器区分，目前1001和1002##后面根据vgroup\_id

"nft\_access": []

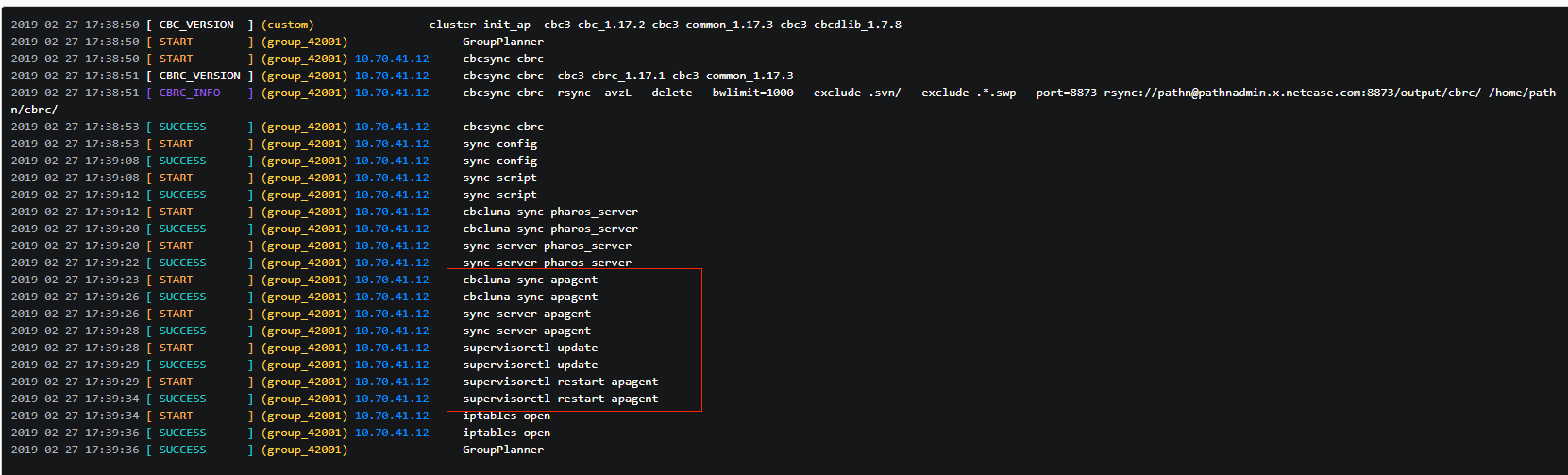
}

ap入库后，走内网的ap设置meta mark（指向杭州的ap，以及杭州ap指向海外proxy），海外a p指向海外proxy，走外网，不设置mata mark；~~指向杭州的ap需要设置静态路由，设置走pathn内网100.\*.\*.\*~~ 在galaxy添加特定标签，aws节点添加aws节点，海外节点（20076,20029之类）设置island标签，生成配置，同步，更新，跑初始化后会把路由配置更新到机器。

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}。

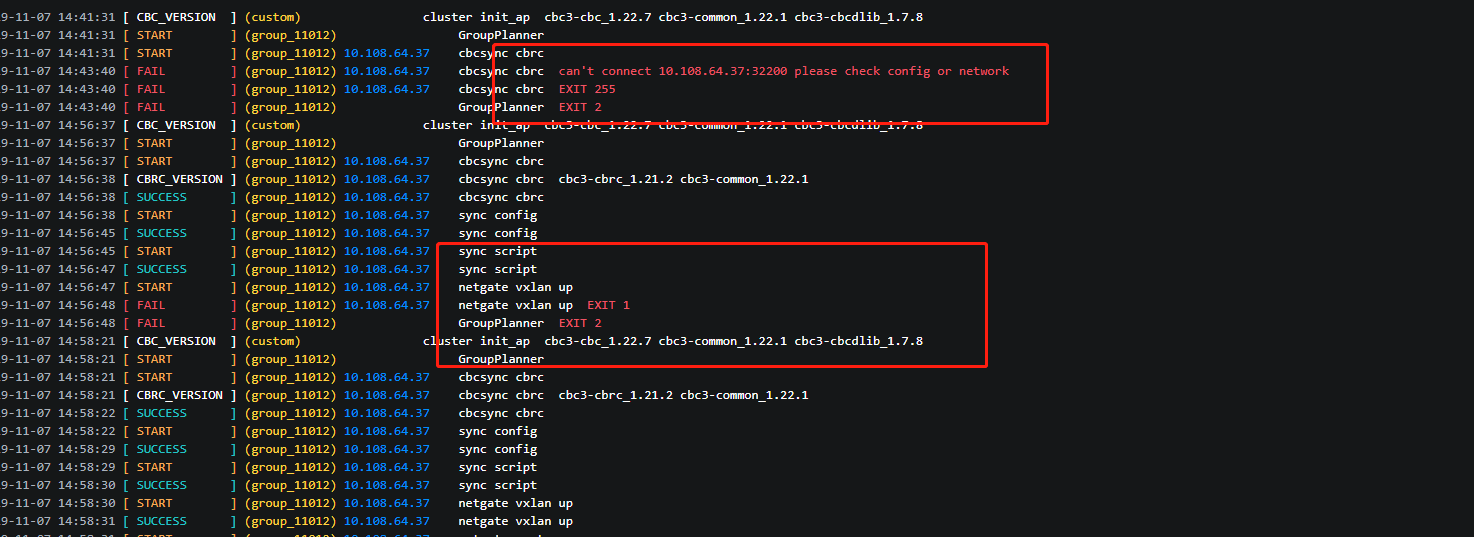
**新管理机下初始化失败：**

cbcsync cbrc can't connect 10.71.33.165:32200 please check config or network

需要在机器手动 sudo ipset add local-net 10.200.122.20 添加新管理机ip

后续通过galaxy同步到全局机器。

**中经云机器11012初始化失败：**



第一个报错是内网路由没加，/sbin/ip route add 10.0.0.0/8 dev bond1 via 10.108.64.1(期间ipset list local-net显示local-net不存在，估计需要阿拉丁推送到机器上)；路由存在的情况下，如果还报错network，考虑ipset add local-net 10.200.122.20。

第二个报错为缺少gpost，pip安装后正常。

**Debian10常见阿拉丁初始化报错：**

2019-11-01 11:13:38 [CBRC\_ERROR ] nft upgrade failed!

2019-11-01 11:22:19 [CBRC\_ERROR ] backend null

2019-11-01 11:22:19 [CBRC\_ERROR ] save iptables and ipset to /home/pathn/conf failed, please check!

2019-11-01 11:29:51 [CBRC\_ERROR ] backend null

2019-11-01 11:29:52 [CBRC\_ERROR ] iptables-restore v1.8.2 (nf\_tables): Set pb-ssh doesn't exist.

2019-11-01 11:29:52 [CBRC\_ERROR ] Error occurred at line: 20

2019-11-01 11:29:52 [CBRC\_ERROR ] Try `iptables-restore -h' or 'iptables-restore --help' for more information.

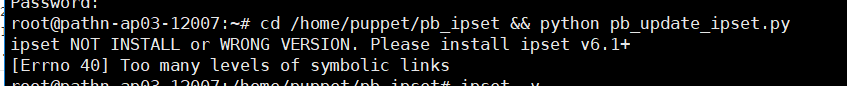
2019-11-01 11:29:52 [CBRC\_ERROR ] Update ipset and iptables failed! Restore to old.

2019-11-01 11:32:05 [CBRC\_ERROR ] backend null

**解决方法：cd /home/puppet/pb\_ipset && python pb\_update\_ipset.py**

**刷新防火墙报错，** iptables open save iptables and ipset to /home/pathn/conf failed, please check!

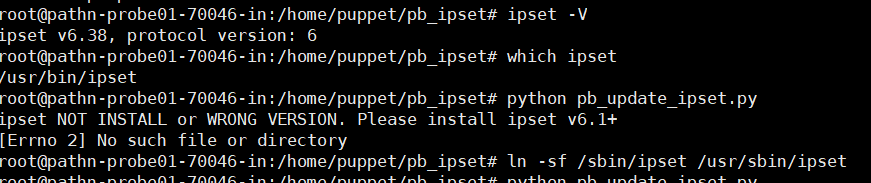
机器上ipset报错：



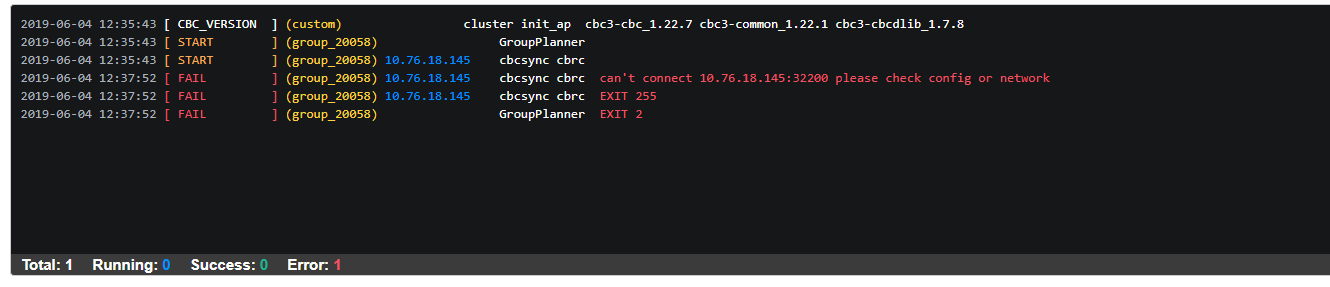
**处理：apt remove ipset && apt update && apt install -y ipset**

**之后再cd /home/puppet/pb\_ipset && python pb\_update\_ipset.py**

**ucloud**机器ipset路径有问题，通知大雄改下镜像，需要替换ln -sf /sbin/ipset /usr/sbin/ipset

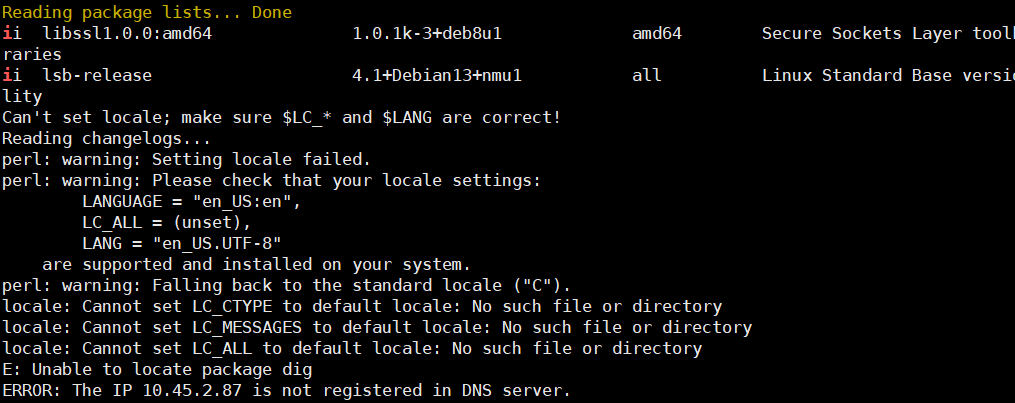


**20058ap初始化失败：**



为网络连接异常，对应内网无法登录和ping，需要找aws管理员添加对应路由。

**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）

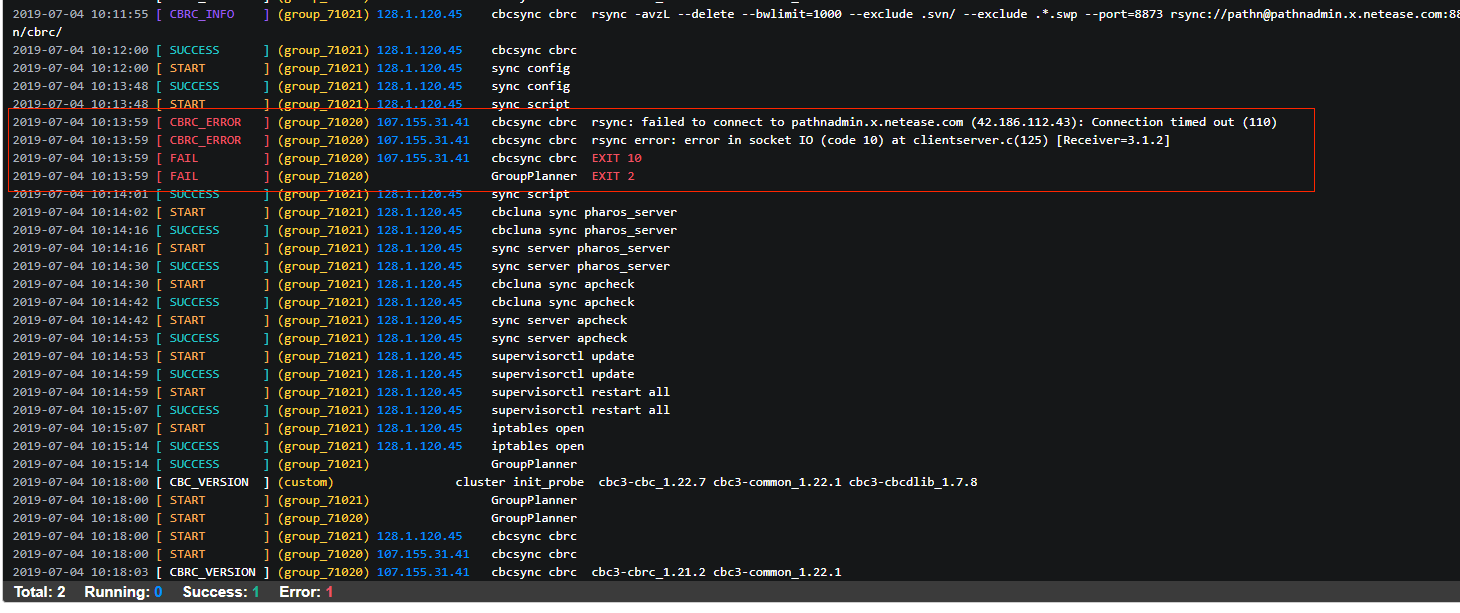
或者在阿拉丁直接走“初始化网络测试节点”流程。（33009)

 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

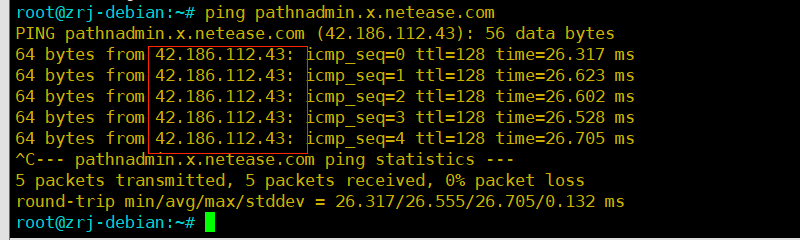
--bwlimit=1000 #限制传输带宽1000kps

--exclude .svn/ ##排除.svn/目录

**Pathn初始化网络测试节点71020，阿拉丁初始化报错：**



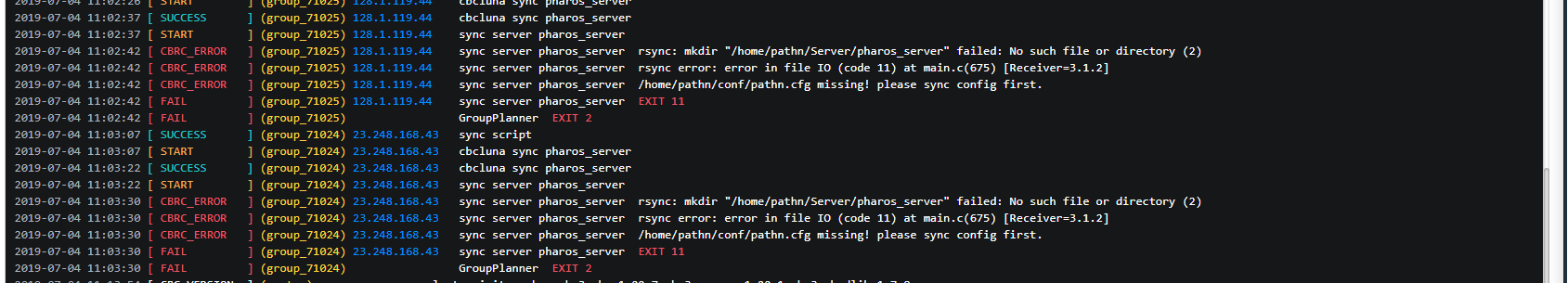
telnet 42.186.112.43 8873不通，网络问题，修改hosts，###该ip为pathn管理机外网ip



echo '18.138.59.204 pathnadmin.x.netease.com' >> /etc/hosts ##这个ip为新加坡echo节点70001的外网ip。

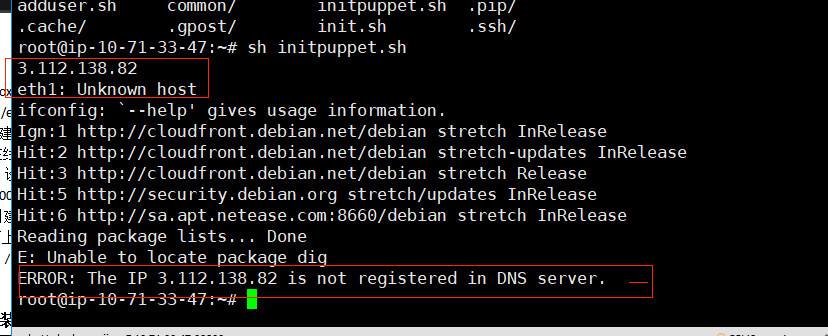
之后重跑正常。

**节点71024和71025报错，exit 11：**

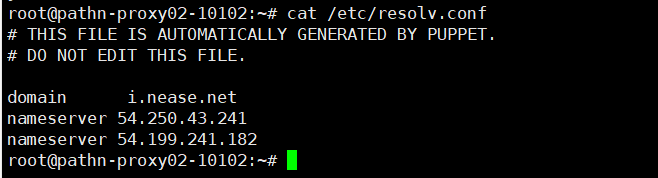
****

实为puppet未跑完整导致，重跑puppet后初始化正常。

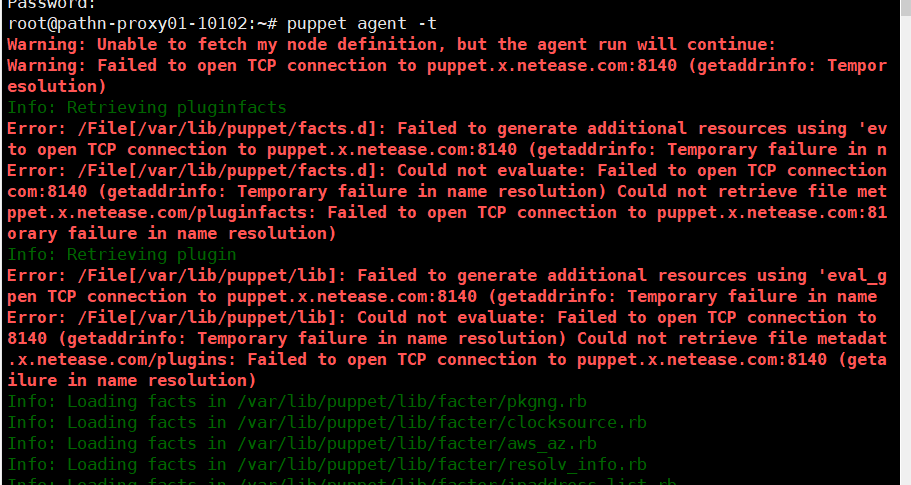
**Proxy01-10102安装puppet报错：（not registered in DNS server）**



确认galaxy生成配置和同步后，dns的同步大概需要十分钟（甚至半个小时），另一台proxy02-10102安装正常，直接参照proxy02-10102这台修改/etc/resolv.conf，之后可以安装：



安装过后，puppet agent -t 运行极慢，有报错：

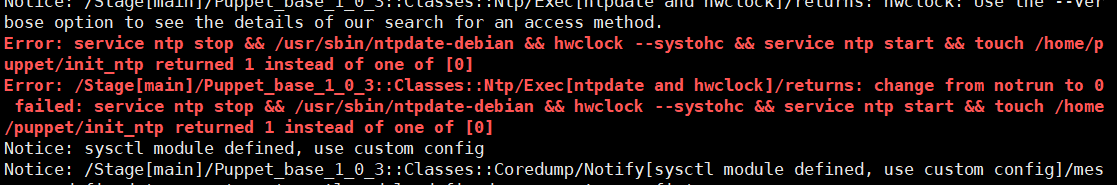


重复跑puppet后正常。

**安装puppet提示：ip is not registered in DNS server**

张家口阿里云探测机器，机器有内网ip，但未在galaxy登记内网ip，安装时会有报错，需要补充内网到galaxy，生成配置再同步。

**台湾gcp 35.201.252.147 跑puppet 报错hwclock**



CDN|郑康佳|konia|58623 15:04:14

去touch 一下

CDN|郑康佳|konia|58623 15:04:21

@SRE|张日军 按照出错

CDN|郑康佳|konia|58623 15:04:39

touch 一下那个文件 绕过 puppet

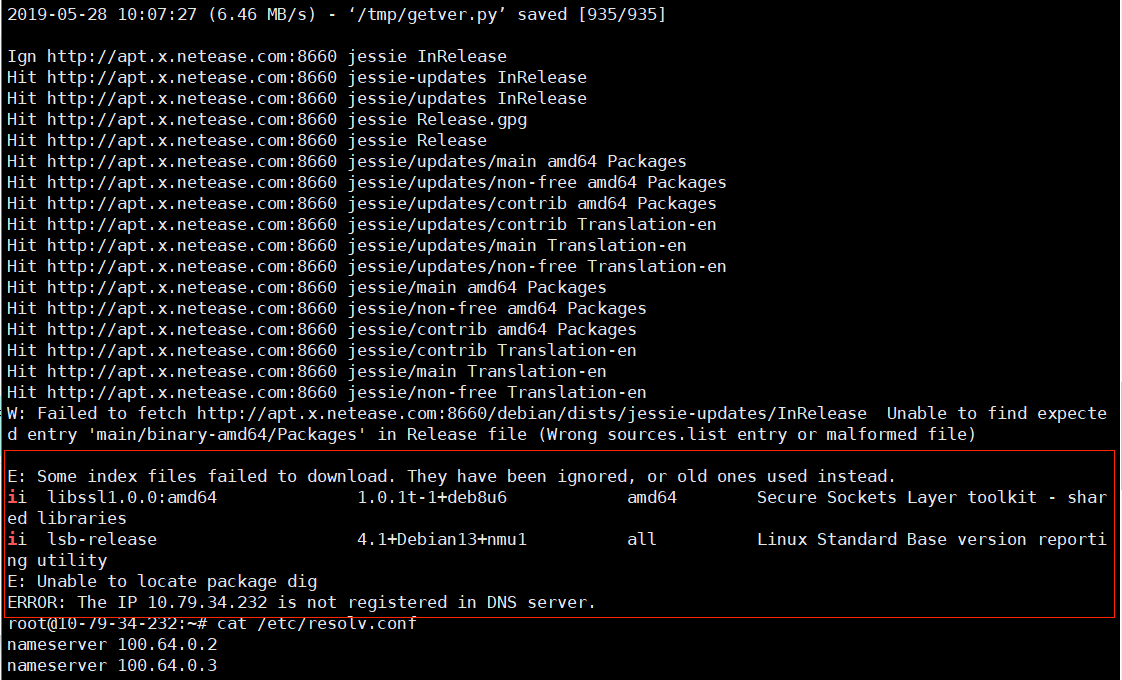
SRE|张日军 15:04:50

1

SRE|黄志雄|大雄 15:05:17

这是镜像问题，已经反馈给了skyline。现在先touch /home/puppet/init\_ntp来跳过这个步骤

20039-20040雅加达和胡志明海外节点安装puppet报错(ucloud机器)：



需要先把机器系统由debian8.8升级9：

修改apt源

cat > /etc/apt/sources.list <<EOF

deb http://cloudfront.debian.net/debian stretch main non-free contrib

deb-src http://cloudfront.debian.net/debian stretch main non-free contrib

deb http://security.debian.org/ stretch/updates main non-free contrib

deb-src http://security.debian.org/ stretch/updates main non-free contrib

deb http://cloudfront.debian.net/debian stretch-updates main non-free contrib

EOF

之后

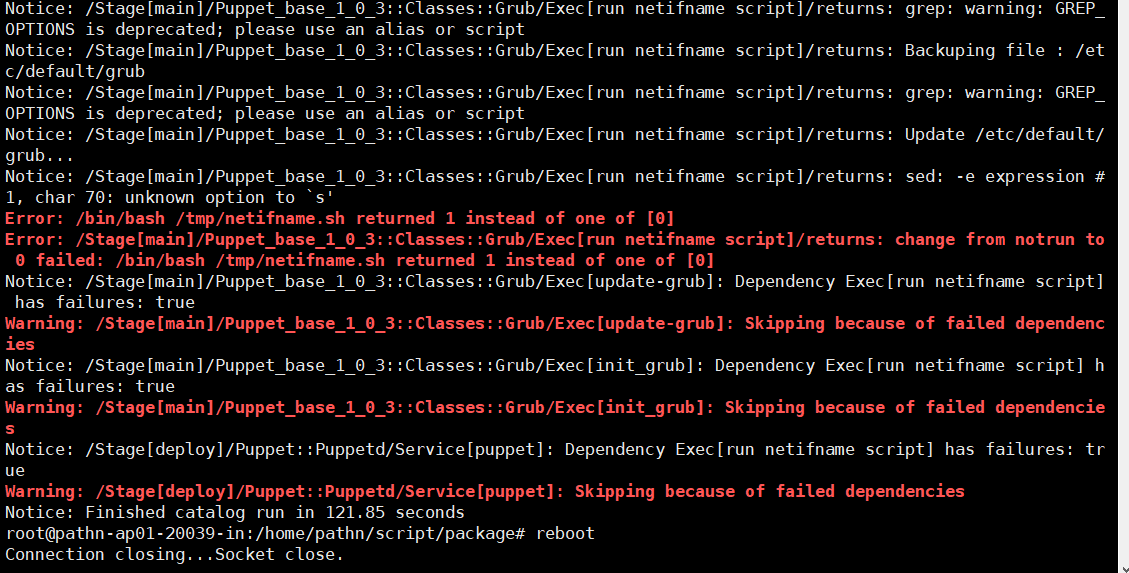
apt update

apt upgrade

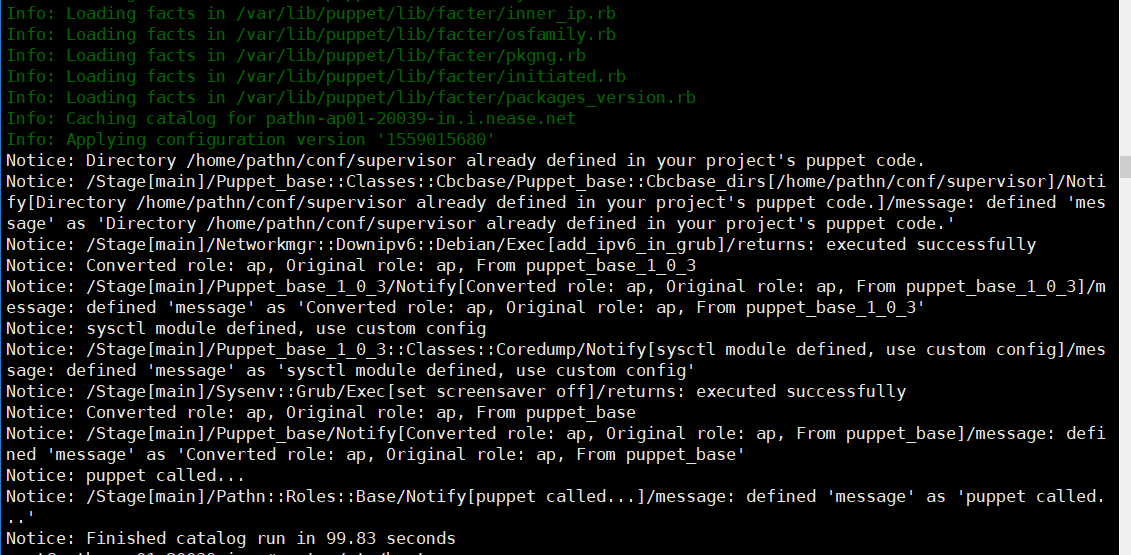
apt-get dist-upgrade

先升级重启, 再安装puppet

升级后，重启会root（端口22）登录不上，应该是因为ssh\_config配置被刷新覆盖(安装步骤中有一步是提示ssh配置文件是否更新)，不允许root直接登录，需要联系管理员在网页登录ssh进去修改PermitRootLogin yes。为避免这个情况，在重启之前，先手动修改ssh\_config,允许root登录，或者在安装中，ssh覆盖选择否，保留旧的配置文件。（没重启直接安装puppet，重启进去后跑puppet报错很多，配置没修改成功）。当时的操作步骤：先升级了系统，之后wget和安装puppet，有提示完成安装，之后重启，ssh无法连上，联系管理员网页shell修改ssh\_config 允许root登录，之后登录进去，跑puppet报错，期间有重新sh 安装脚本，而后再重启过，这次登录需要使用public key，进去后再跑puppet，多次后通过。



重启过后，安装puppet后，ssh\_config会被再次修改，重启后，机器的端口会被修改为32200，而且会把登录方式改为publickey，root的密码还是之前的。切换root，再跑puppet时间很长，但没有报错：



**galaxy上机器名与登录机器看到的名称不一致（15002）：**

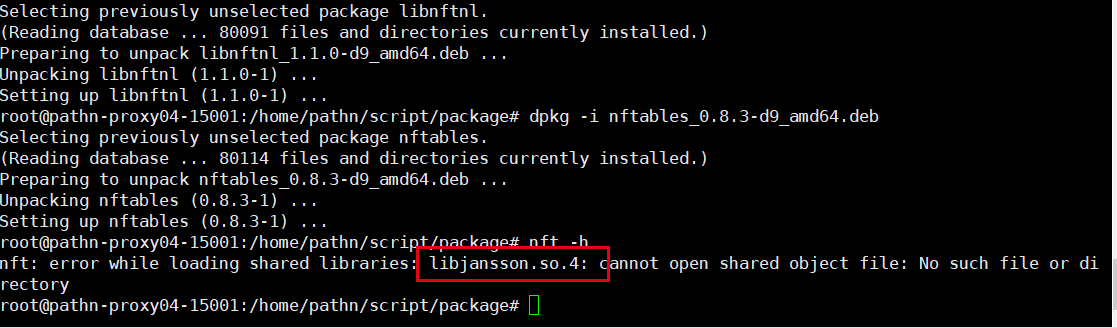


15002节点两个机器名不一致，是因为备用群组里重复绑定了机器：



需要把备用群组里的机器删除，重新生成配置。

**犯的低级错误：**没有生成配置和同步，直接跑puppet，导致nft安装包没有同步到机器，机器的名称也没有同步，各种配置不生效。nft安装后报依赖包问题，需要重新跑一下puppet。



**15002-02初始化失败：** Lost remote after 10s heartbeat

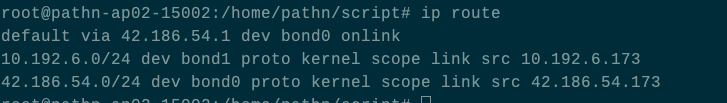
检查路由，检查网络，内网不通http://int.api-pathn.nie.netease.com/api/v3 pathn服务端就会在最后agagent upgrade报错；未接通内网之前，可以通过打island标签先走外网。



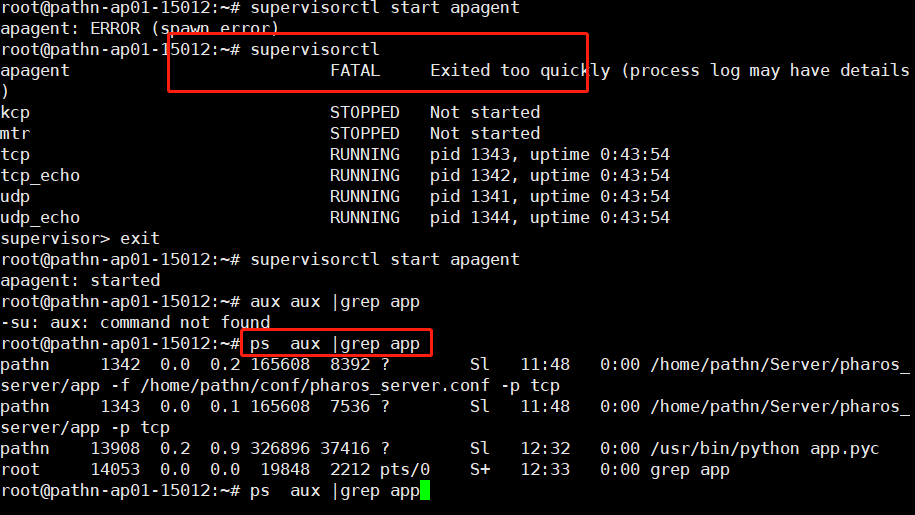
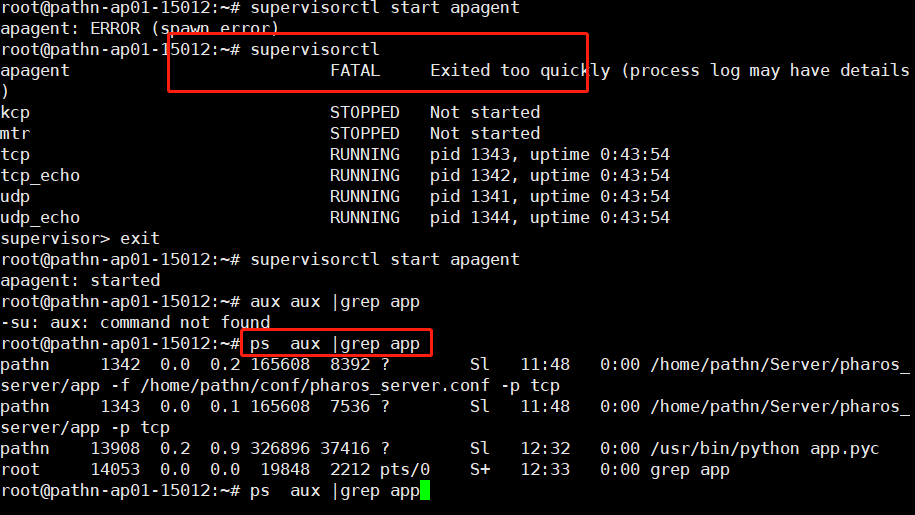
缺了10.0.0.0/8 默认路由，需要手动添加上，参考15002-01的/etc/network/interfaces,写入

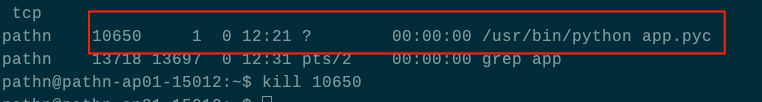
up /sbin/ip route add 10.0.0.0/8 dev bond1 via 10.192.6.1之后阿拉丁正常。

up /sbin/ip route add 10.0.0.0/8 via 10.200.40.1 dev bond1 ##两种写法



**15012-01初始化失败：supervisorctl restart apagent EXIT 111**





kill掉10650，重启apagent。

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修改群组为在线状态，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

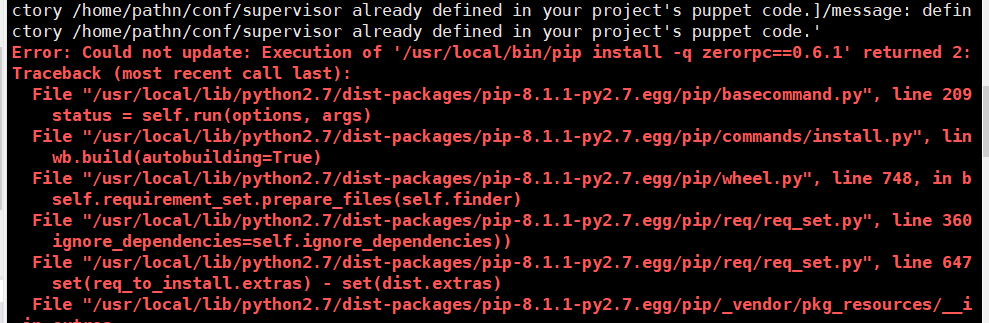
3.爱尔兰aws内网段不通，savpn申请添加新网段（加完网段再不通再找SESA值班检查一下aws的vpc设置）

**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

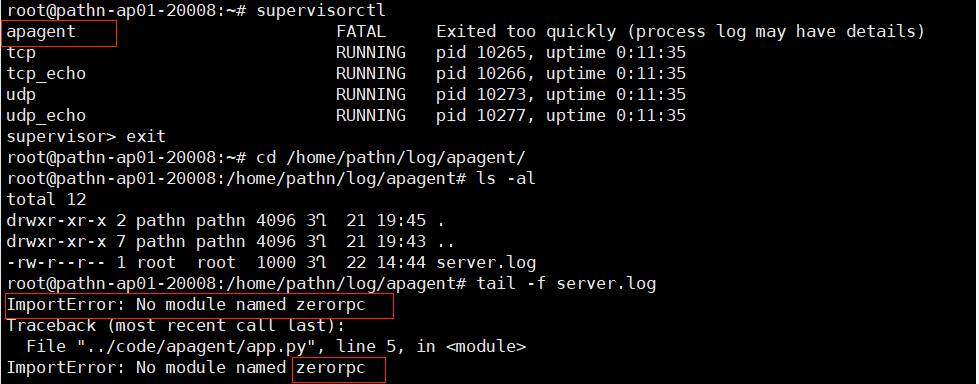
pip install pip==9.0.1 -U

apt install cl-cffi

手动安装zerorpc

pip install -q zerorpc

arm实例排查



Arm实例和t3实例的rp\_filter没关，需要检查。

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：**

**原因：应该是已经有Nftables 列表了. 但是重启了apagent 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)

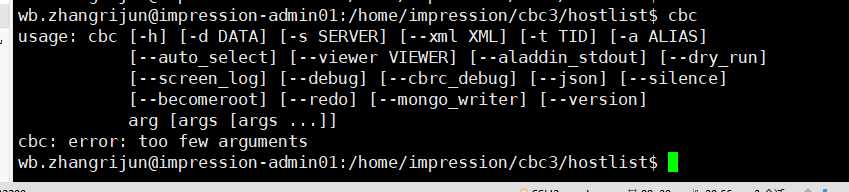
cbc xxx cbcluna sync apcheck

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.

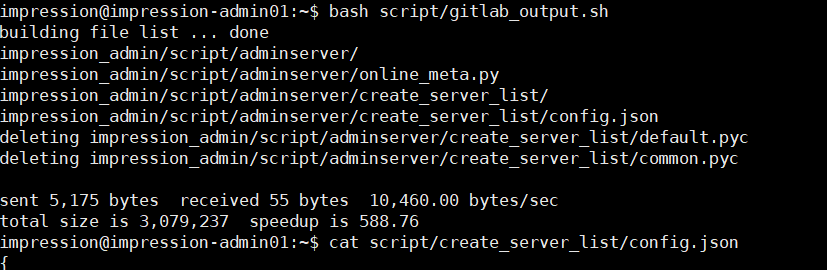
ap 的gid字段可以留空，pathn自动分配可用gid，如果再次指定已经使用过的gid，会把之前的群组覆盖。

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（/home/impression/script/create\_server\_list）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。



目前的控制列表操作还是在impression管理机下操作，对应impression的gitlab操作，因为游戏SDK的列表还是用的 https://impression.update.netease.com。



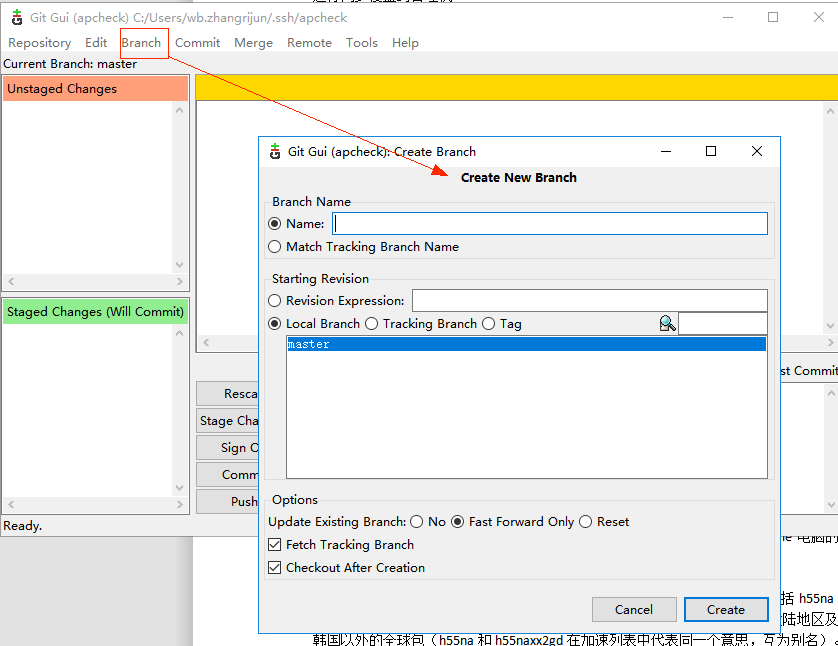
**Gitlab的key说明：**

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

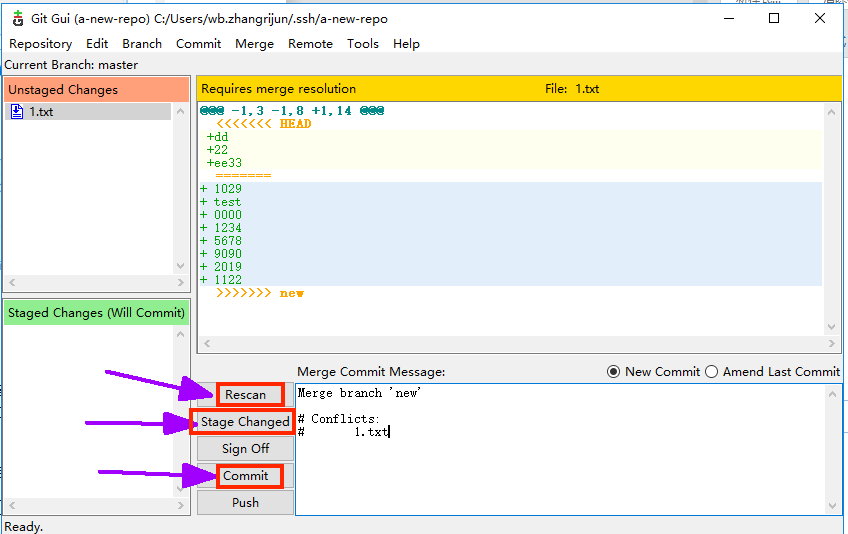
2.Xshel的key登记后，用于xshell登录其他机器（公钥在workflow登记后会同步到所在项目下的机器中，实现key的录入，之后才能ssh远程登录）。

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。

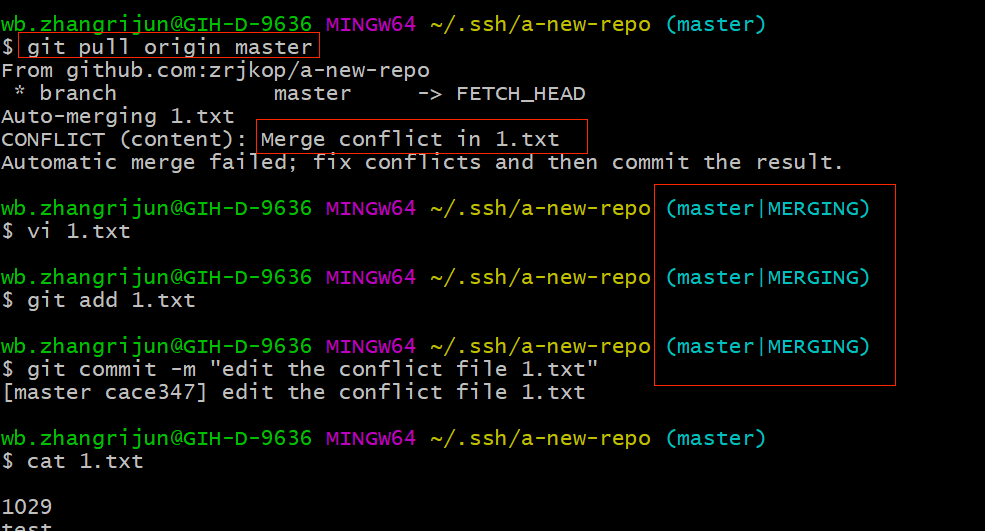
**Git-gui通过clone复制远端库到本地**，会同时把库的几个分支拷贝下来，一般会保留master,其他的分支，可以通过create branch创建，勾选match tracking branch name即同步远端分支名称和文件，之后可以切换分支和提交。



在本地编辑修改库里面的文件后，git-gui界面里通过rescan会把所有修改过的文件添加到图形界面框内，右上角会提示文件改动对比，需要staged changed保存修改，之后commit，最后通过push同步到远端。



在远端和本地文件一致的情况下，如远端和本地都有1.txt，且内容一样的文件，此时在本地修改1.txt，之后push到远端，能正常提交；但是如果，远端和本地的1.txt不一致，即1.txt在远端被修改过，本地的1.txt也被修改过，这时候直接push会报错，提示文件不一致，冲突无法提交，解决方案：一是 git pull origin master，先同步远端文件到本机，此时会提示有文件冲突，1.txt，vi进入文件会看到冲突内容，重新编辑后可以git add 和git commit，之后push正常。



二是使用强制push的方法，会把远程库数据强制清除，提交新数据

git push -u origin master -f

三是若不想merge远程和本地修改，可以先创建新的分支：

git branch [name]

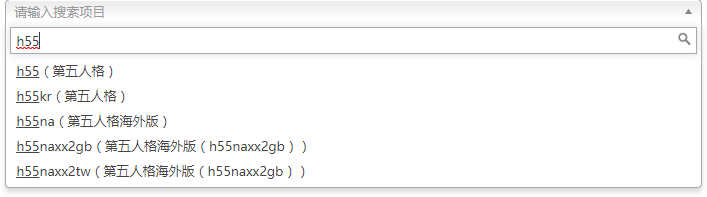
然后push

git push -u origin [name] ###以新分支提交后，如果需要合并到master，在web端同样需要解决冲突问题，需要合并差异部分。

###编辑不同分支，删除不同内容，能否直接合并（删除相同内容时合并正常，在不同分支编辑相同部分内容，做相同修改）：不同编辑，合并报错，需要合并冲突内容才能提交。

Key：公钥public用于服务端加密，所以需要上传公钥给服务端；私钥private，用于本机解密，个人保管，如果其他机器拿到你的私钥，也就可以跟服务端建立连接（办公电脑的私钥导入到home电脑使用，就是这种形式，公钥还是同一个；或者重新登记home电脑的公钥到服务端）。

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~~，###特定地区的包，意味着region是固定的，只会去读取特定的列表。###如果台湾的包在韩国地区进入游戏，又是如何处理（这种情况会不会提示重新下载游戏包？）。~~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？？？？~~这点说法不对，特定区域的发行包，比如台湾包，韩国包，应该region是确定的，项目应该是针对特定地区发布，不需要额外的region判断，直接读取对应的加速列表。对于全球包（项目也不清楚玩家在哪各地区），会对region进行判断，之后读取对应的列表。灯塔的sdk，用于region的判断，对应特定地区的发行包，不接sdk也是可以，无需判断region，项目通过客户端能确定到region，就能拼接加速列表的url去获取ip和端口。项目也可以自己开发sdk，基于ip库等逻辑，去判断region，获取正确url。

灯塔sdk，集成region判断（网络优化，指引接入加速节点）以及网络探测，项目如果不接灯塔sdk，采取自开发sdk，网络探测模块是不是用不了？Sdk的方式：1是使用unisdk，Android接入需要在unipack选择渠道后，勾选Pharos二级渠道，IOS接入需要将SDK文件下载更新到母包；2不接入unisdk的，可以通过直接调用PharosSDK.framework的接口。

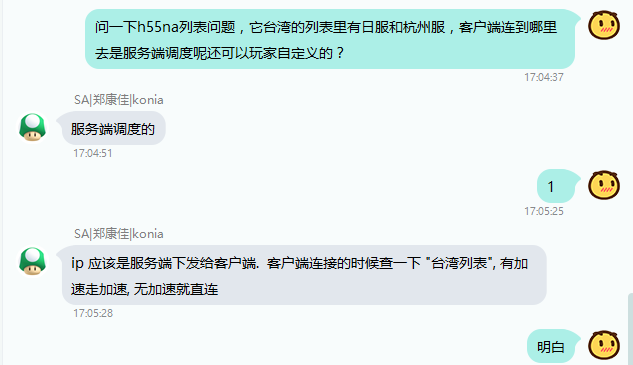
**项目在杭州和日本都有游戏服**，如h55na和g103，台湾地区的列表里面包含杭州和日本的游戏服：<https://impression.update.netease.com/lighten/pathn_g103_taiwan.txt>

<https://impression.update.netease.com/lighten/pathn_h55na_taiwan.txt>

台湾玩家的分配，是基于什么逻辑，是有客户端玩家自定义还是服务端调度？~~H55na的情况，如果列表清除日本战斗服，台湾玩家会直连过去日本，目测是玩家手动选择的战斗服。在台湾hinet到日本aws专线线路故障时，是否应该把列表中日本的服下掉，这情况下玩家如果选择日本的战斗服，是否会直连过去？或者端口探测异常后，客户端自动选择其他服（杭州服），甚至会直接直连？~~

~~按照h55na的说法，列表如果下掉日本战斗服，客户端会选择直连，那服务器选择应该是客户端玩家自定义，由玩家去选择进到哪里的战斗服。~~

以上说法错误，连接杭州服或日服，基于服务端调度，ip由服务端下发到客户端，客户端拿到ip会对比台湾列表，有加速入口就走加速，没有加速就直连。



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

**打不打标记：走内网的ap，双网卡多IP，设置标记；走外网的ap，不打标记（暂定）**

aws节点需要galaxy添加标签aws，海外节点如ucloud的节点，需要添加island标签，生成配置，同步，之后跑初始化节点（会影响机器路由的添加和阿拉丁初始化）

**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）：sysctl -a | grep rp\_filter或者cat /proc/sys/net/ipv4/conf/\*/rp\_filter

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

}

}

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。

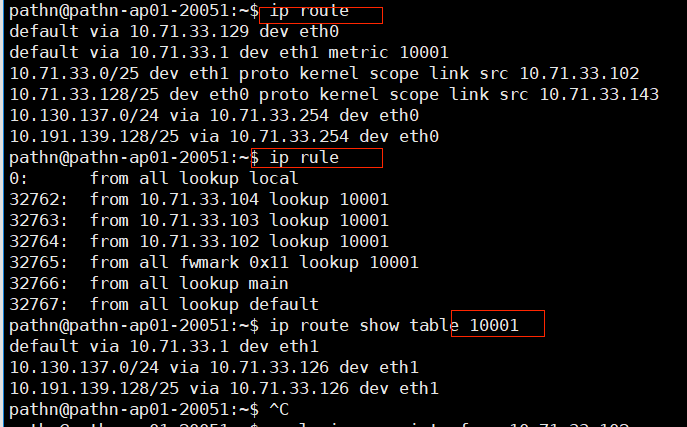
2：开启松散的反向路径校验。对每个进来的数据包，校验其源地址是否可达，即反向路径是否能通（通过任意网口），如果反向路径不同，则直接丢弃该数据包。

**~~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（这种情况需要把rp\_filter=0）；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；~~不打标记的情况下，需要aws节点的走后端默认路由是走eth1，进出都走eth1.

14014（韩国朗桥节点）bond0上海电信ip，bond1韩国ip，打了标记，海外流量走bond1进来，nft链的dnat后，走默认main路由，通过bond0转发到杭州机房，杭州机房回包发送到bond0，通过标记恢复，使流量走路由表kr，通过bond1回包到韩国玩家。

20036，孟买节点，指向10001，打了标记，流量的转发只通过eth1实现，路由表10001写有指向后端ip的路由，通过eth1发到后端backend，后端回包时，conntrack恢复标记，还是走10001，通过eth1发回给玩家客户端。

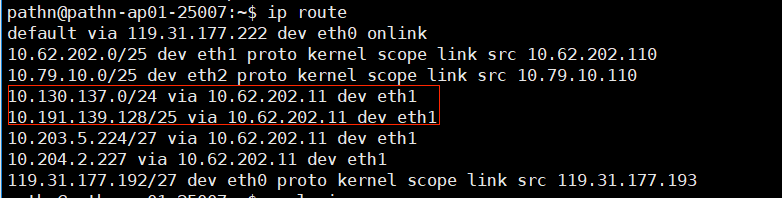
~~之前创建的节点，都统一使用了orgin：0x00000010和reply：0x00000011这个标签值，如果这么设置，需要rp\_filter开启，且会有性能瓶颈。rp\_filter默认关闭。~~可能理解错误，再查证。**下次验证新增指向10001节点时，没加aws标签时，看看路由的配置信息对比（需要阿拉丁初始化）；之后再添加数据包标记，测试一下origin使用0x11的情况，记录每步操作后apcheck的情况。结果**：20051，添加galaxy标签aws，（1）没添加标记时，初始化后有对应路由10001表生成，默认是通过.1的网关，但是端口测试不通,可能是.1的默认网关无法连通后端；（2）添加origin使用0x11的情况，端口可通；（3）添加 "origin": "0x00000010", "reply": "0x00000011",端口通。后两种情况，10001路由表生成了访问后端使用的默认gw为.126（/25掩码）



之前20054，20083未加aws标签，不打标记，端口不通；

下次验证：不加aws标签，直接打标记，目测也是不通；或者添加aws标签后看路由生成情况，再添加标记，对比路由生成情况，再把标记enable参数设为false看看。

**台湾节点20056**，3个vip，导入pathn使用3个外网ip，后端网络类型使用10000，走内网，没有使用标记。机器上eth0为台湾ip119.31.177.193，eth1为pathn内网 10.62.202.110，



客户端转发流量从eth0进来，经过eth1回到proxy后端；proxy的回包返回到ap时，查询默认路由，回包再次从eth0出去，（关键在于其默认路由走eth0？能让回包返回原来的链路）发回去给客户端。

对于双网卡aws节点，如果不标记，不加eth1策略路由，流量从eth1进来，通过eth0走默认路由转发到proxy，proxy回包时，还是查询默认路由，走eth0发出去给客户端（源ip是eth1而被丢掉）？这种情况下回包会被丢掉，无法回到客户端。

问题是如果aws设置eth1路由，使流量转发只经过eth1，是否可以不设置标记？这一步需要怎样设置路由？之前的测试20051，不加标记，端口测试不通，转发有问题，原因何在？不加标记，是不是导致回包查询默认路由时走eth0出去，导致客户端无法收到？下次新建指向10001的aws节点，在不加标记的情况下，手动添加走eth1默认路由看看，为何加了标记会10001路由表会添加.126或者.62的路由？海外---->海外的情况，走外网，10001路由表没有添加.126或.62的路由，走的是默认ip所在段的gw。

目前可以确认，海外---海外的情况，走外网，对应/home/pathn/conf/ip\_route.conf为空，对应10001路由表的默认网关为eth1的gw。海外---国内，或国内---海外，走内网的情况，对应/home/pathn/conf/ip\_route.conf里面写有eth0的默认网关，以及10001路由表gw。

目前不加标记的情况：20056这种，vip是默认路由对应的ip，即eth0，流量从eth0进来，proxy返回回包时，查询的默认路由也是eth0出去；

另外就是单网卡，进出就是一个网卡，不需要标记，不需要策略路由。

日本aws节点20054,20083指向杭州10001，需要galaxy添加aws标签，之后生成配置，同步更新（把aws对应路由配置同步到机器），设置在线，在阿拉丁初始化ap(下发配置到机器)。另外需要设置标签值：

{

"nft\_mark": {

"origin": "0x00000010",

"reply": "0x00000011",

"enable": true

}

}

初始化后，节点机器上10001路由表：

root@pathn-ap01-20085:~# ip route show table 10001

default via 10.68.24.1 dev eth1

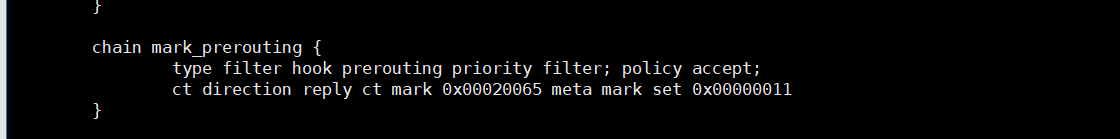
10.130.137.0/24 via 10.68.24.126 dev eth1

10.191.139.128/25 via 10.68.24.126 dev eth1

ip rule输出：



Nft的配置：



回包会被恢复0x11标记，然后走10001路由表，即从eth1进来的数据包，还能从eth1出去。

**目前走内网的aws都打了标记，专线ap如20058（enjoyvc）没打标记，由于只有一张网卡？**

**走海外的一般不打标记，但前期创建的打了，如20007指向10101，origin使用0x10，reply使用0x11，会影响性能？**

**单网卡的ap可以确定不打标记（数据进入只用一个网卡，走默认路由）**

**20008指向10101，打了标记，用0x11相同的标记**

"nft\_mark": {

"enable": true,

"origin": "0x00000011",

"reply": "0x00000011"

},

Nft输出：

chain mark\_prerouting {

type filter hook prerouting priority filter; policy accept;

ct direction original ct mark 0x00020008 meta mark set 0x00000011

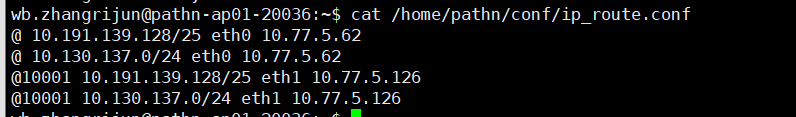
ct direction reply ct mark 0x00020008 meta mark set 0x00000011 }

**ap端口测试正常前，先把ap状态设置为3或者0**，此时阿拉丁上会不显示机器，如果额外给机器添加标签，需要阿拉丁初始化时，需要galaxy设置机器在线状态，之后生成配置和同步，然后阿拉丁添加操作对象时才会显示对应的机器。待端口测试正常后才更新ap状态为1，不然echo探测会一直报警：



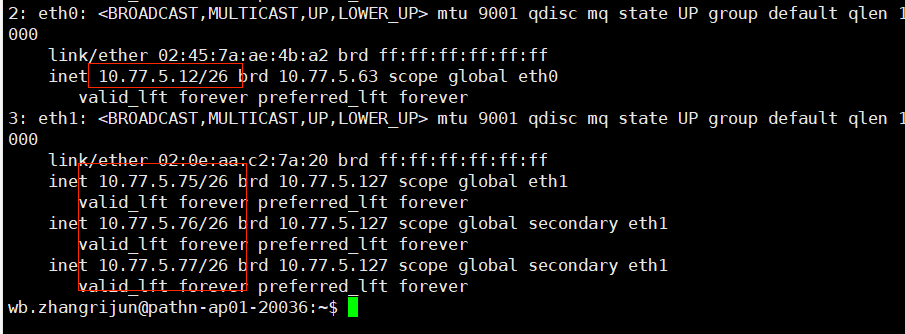
**节点20036指向10001**，初始化后，对应路由表10001中只有一条default规则default via 10.77.5.65 dev eth1， 测试不通，需要修改路由配置：

路由配置路径在conf目录下ip\_route.conf，修改gw后，route\_check -f 修改配置后刷新路由



eth1是10.77.5.75/26，网关用10.77.5.126

eth0是10.77.5.12/26，网关用10.77.5.62

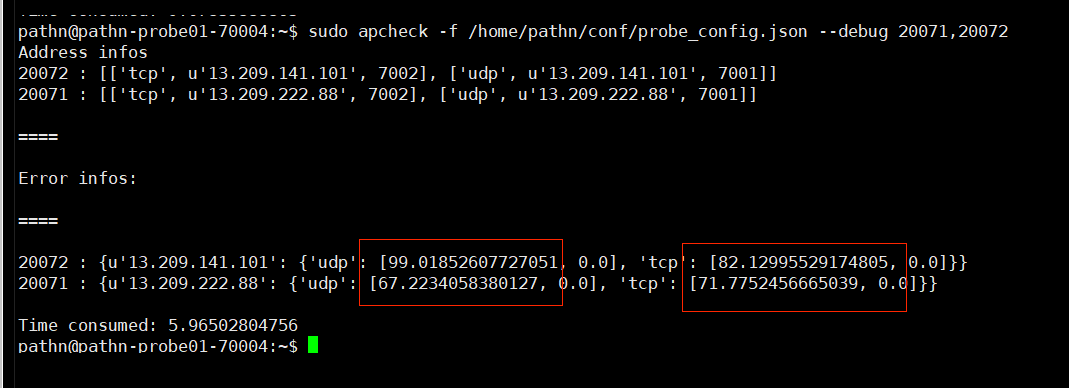


其他节点使用掩码25位/25,对应的gw分别为.126和.254

**Pathn网关设置：一般情况下内网使用子网最后一个可用ip作为网关，如上面的/26,使用62和126作为网关ip，走内网节点ap上面基本是这么设置，子网的第一个可用ip如.1作为默认网关，也是可以通；外网的网关（运营商ip），例如台湾福州25009，福州移动183.251.62.221**

**/29和联通36.250.7.253/29,使用子网第一个可用ip，183.251.62.217 和 36.250.7.249作为网关ip。**

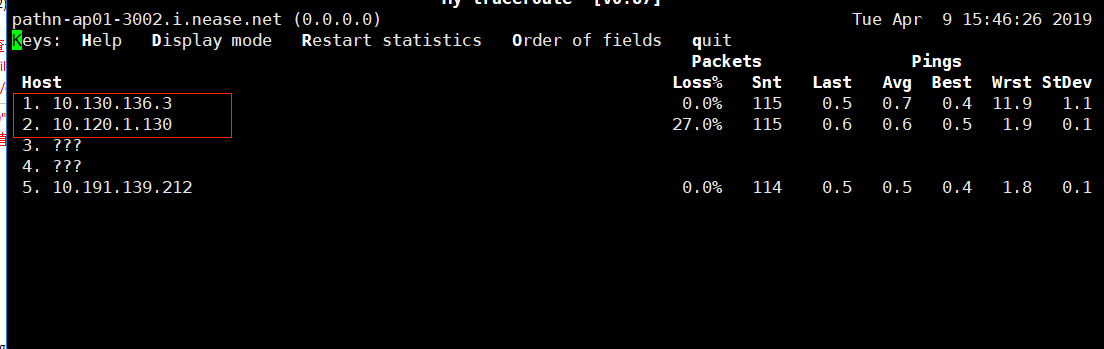
**20071,20072指向10003**，apcheck显示两个节点延迟有差异，



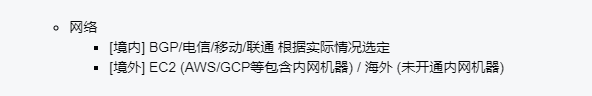
在20072ping后端，延迟比20071大，mtr跳转看不出原因，ip route 显示对应路由路径正常。排查为路由问题，是因为ping包去程走专线内网，回程绕道香港导致延迟大。康佳操作后端proxy刷新路由，之后正常。

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。



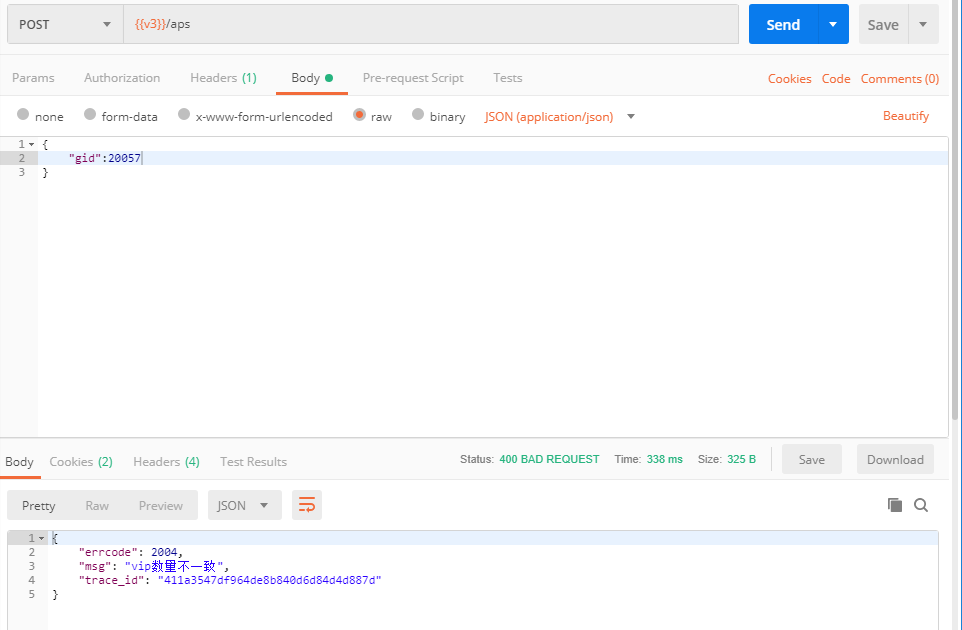
**专线节点机器多为虚拟机，多个节点共享一个实体机，登录一个节点即可验证走内网情况，查看backend.conf看后端ip，分别做mtr和traceroute验证。**

**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(台湾probe节点)去测试端口。**

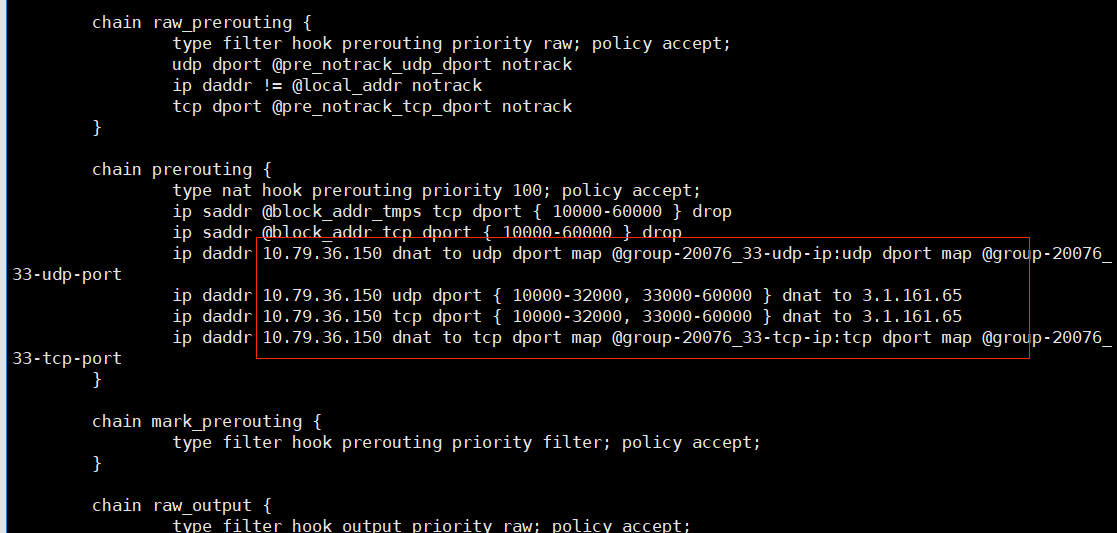
**Galaxy添加10202和20076映射：**

Ap20076指向proxy10202，各有一对内外网ip，使用1个vip，入库前需要galaxy生成映射。上午创建10202时，错误操作生成映射，把源ip选择外网ip，目的ip选择为内网ip，导致入库后，postman使用get显示proxy的ip刚刚相反了，外网ip显示类型为10000，内网ip显示类型为10001；所以在下午操作ap入库时，生成映射操作同样错误，ap显示的ip类型也相反，修改转发后端默认ip类型为10001时，agent backend显示异常，nft显示异常，端口测试报错。需要把相关机器先下线，出库，然后再入库。

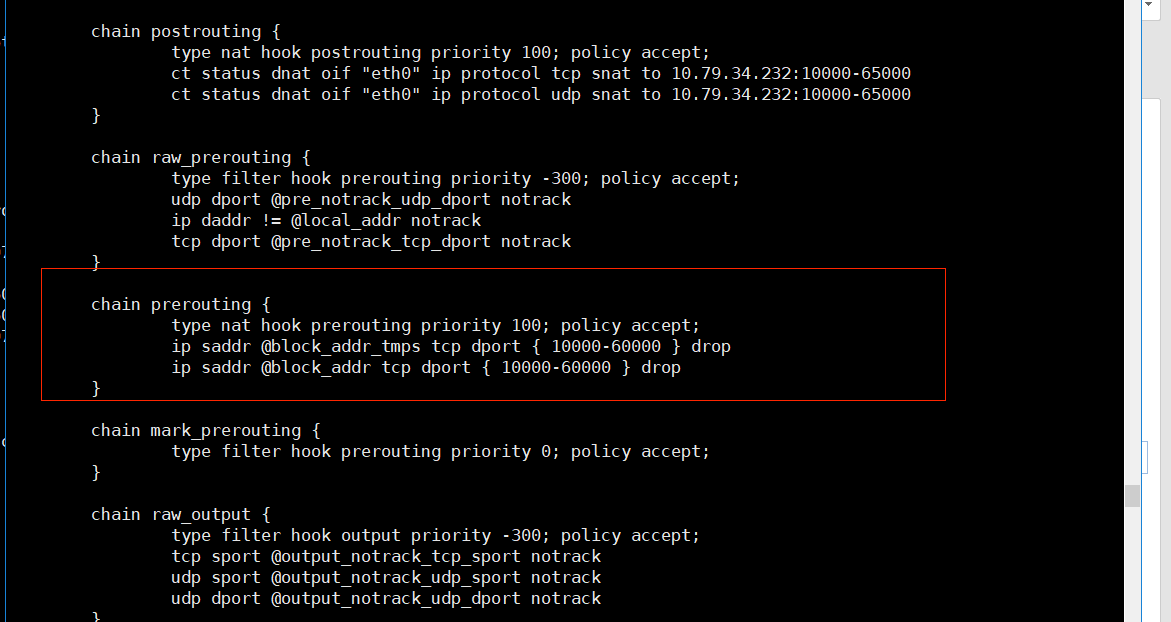
Galaxy生成映射，源ip使用内网ip，目的ip使用外网ip。

**海外节点20039,20040指向10202，使用海外网络11，刚开始入库只导入外网ip，没有生成映射，ap的get显示bind\_infos里面ips只有外网ip，这种情况下，设置后端网络类型走外网10001，但是阿拉丁初始化后nft list ruleset的输出跟20076有差异：**

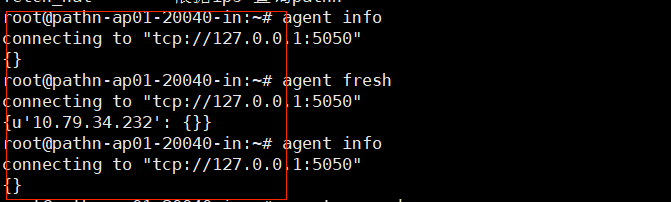
**20076：**



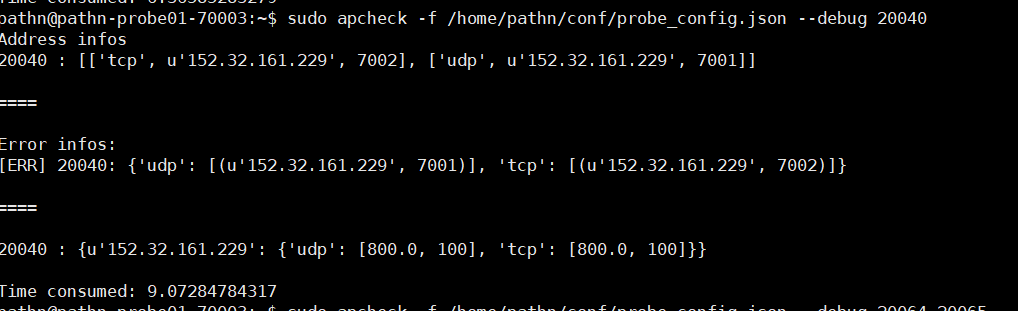
20039-20040:



相比20076，输出少很多东西；而且agent backend没输出，upgrade没反应，version没输出。



探测机器上使用apcheck检查节点不通。



解决方法：先把ap出库删除，在galaxy上把映射添加，再重新入库，之后agent upgrade后显示版本号，nft输出正常。

**针对采用写列表项目迁移v3（目前g103，g37）：**

**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。~~目前还有g37，g37gb，h18，g83这几个使用写列表方式。**

****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}

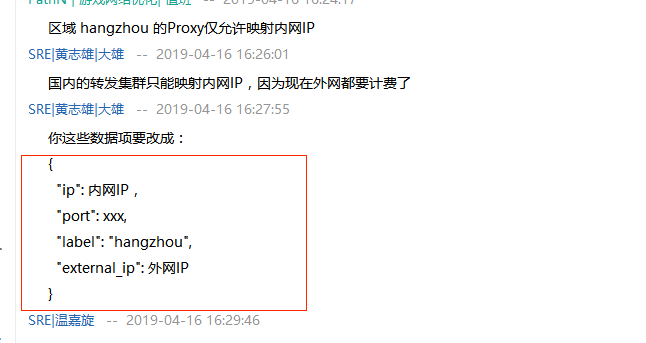
****

**v3优势：**

1. 增强稳定性, galaxy 数据项不容易出现出错

2. 高可用的网络互连





**g37迁移：**

1. 触发的EXT 数据项 需要修改label, 原来的 cn => hangzhou, us => virginia, jp => tokyo

2. 列表读取回调的格式 有调整

v2 是region => ip => port

v3 是ip => $port => $region

数据项参数：

对于容器, 都填外网Ip + 一个enable\_public =1 参数

非容器, ip 为内网ip, external\_ip 为外网IP

**G37日本列表，pathn节点是否生效，参考这个url**

https://g37.update.netease.com/mini\_server\_list\_jp.txt

**U2na接入问题：**

通过列表访问服务器登录失败，节点telnet测试（游戏服单独开tcp端口）正常；排查为游戏服没有监听内网ip，需要额外配置数据项，允许使用外网映射(bigworld引擎游戏项目)。

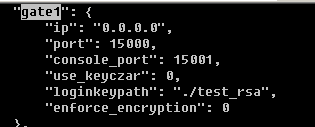
enable\_public 可选，国内机房默认仅允许映射内网IP，如果机器无法支持，将这个值设为1可允许映射外网IP。

修改数据项，添加enable\_public=1（int）参数后转发正常。

大部分的项目都监听内网，proxy--->游戏服这一段都内网，不用计费；对应没监听内网的，这一段只能走外网，计费。

**g70迁移问题：**

v2迁移v3，v3要求杭州机房监听内容，已经监听内网：



gate监听的0.0.0.0应该支持内网访问。

5月3号项目反馈，台湾地区使用WiFi无法连接，4g连接正常。v2使用的是台湾gcp节点，抓包测试能看到客户端建立连接且有数据交互，但客户端卡在登录界面；根据udid在elk看到测试客户端的网络连接正常，rtt数值较小；手动修改列表，添加台湾hinet接入节点，客户端测试WiFi正常，怀疑是玩家到谷歌gcp入口这一段有问题（gcp的原因？）。项目添加v3外部数据源，pathn分配hinet接入点，重新触发生成新列表，之后覆盖v3覆盖v2。

**Pathn testing接口创建试炼场节点：**

42002登录进去显示42007，backend显示40002，这是因为之前测试时创建过42001--->40002，后面对机器回收销毁，重新跑puppet和初始化后正常

**圣保罗新建aws节点内网无法登陆：**

因为内网ip段海外接入ospf，需要创建实例机器(skyline按ospf模板创建)，~~提供给ospf(galaxy上需要操作变更机器所属项目pathn---->ospf)，作为接入ospf的gw。~~

~~找ospf的值班 grp.ospfsa@corp.netease.com，提供机器ip，申请接入ospf，可能需要添加人员的操作权限sh adduser.sh [xxx@corp.com(账号)。](mailto:xxx@corp.com(账号)。)~~

###操作已更新，skyline上创建ospf角色机器后会直接转移到ospf项目下，不需要galaxy操作，不需要手动添加人员。

**端口测试：**

1.服务端开启simple http

进入一个空目录下/tmp/xxxx, python -m SimpleHTTPServer 25000 ## SimpleHTTPServer为python2的模块

客户端可以通过telnet测试端口25000的状态；

2.pathn或其他游戏服，服务器里应该有echo服务，可以通过直接 ./echo -p 24000开启监听24000端口，之后在客户端发echo请求验证；利用telnet或nc，应该可以接收到echo返回的输入信息。

3.用netcat测试

udp：服务端 nc -ul -p 1080 ###监听1080端口

客户端nc -u服务端ip 1080，建立连接后可以通信，两端可以收到输出内容

tcp：服务端 nc -l -p 1080

客户端 nc 192.168.65.132 1080，建立连接，能收到两端输入的信息

或者使用telnet 192.168.65.132 1080，建立连接，也可实现输入输出。

4.echo探测机器里有pathnping脚本：

Pathn项目probe机器在/home/pathn/Server/apcheck里面，通过命令:

./pathnping -u https://impression.update.netease.com/lighten/pathn\_h50\_jp.txt -o probe.txt

可以探测这个列表的ip和端口，并把结果输出到probe.txt中。

proxy机器的心跳开启 sudo supervisorctl start tcp

ap机器的心跳开启sudo supervisorctl start udp

節點上線前檢查，對節點ip和端口進行telnet測試

<http://impressionadmin.x.netease.com:8892/lighten/pathn_h50_jp.txt>

1. 服务端未开服或者防火墙未放通情况下，加速列表里的端口无法telnet

<https://impression.update.netease.com/lighten/pathn_popo_jp.txt>



入口机器13.230.192.193机器上通过netstat看不到监听10000-10002，在nft规则里面有对应规则，

ip daddr 10.71.33.26 ct mark set 0x00000011 ip protocol tcp dnat to 10.200.122.60:tcp dport map @a249\_62-ports

其中a249\_62-ports {

type inet\_service : inet\_service

elements = { 7001 : 9999, 7002 : 9998 }

~~节点机器使用tcp7002和9998（映射关系），udp7001和9999（映射关系）去转发流量？客户端通过ip：port连接到ap，ap转发到proxy（通过70001这几个端口转发？）感觉是随机端口，nft里面开放端口范围10000--30000，33000-60000；proxy上面nft里面记录了与游戏服的端口映射443:10000；443:10001；443:10002，应该是在这里进行筛选，对于不是请求10000-10002端口的数据丢弃。~~有待商榷。

参考15030

ip daddr 10.70.41.148 ct mark set 0x00000011 ip protocol tcp dnat to 103.135.241.26:tcp dport map @a206\_58-ports

ip daddr 10.70.41.148 ct mark set 0x00000011 ip protocol udp dnat to 103.135.241.26:udp dport map @a206\_58-ports

这两条规则，是给探测机器探测链路使用的，指定了7001到7001映射，7002到7002

map a206\_58-ports {

type inet\_service : inet\_service

elements = { 7001 : 7001, 7002 : 7002 }

业务流量应该是这两条，放通所有的端口（实际是列表里出现的端口才有效）

ip daddr 10.70.41.148 ct mark set 0x00000011 udp dport { 10000-32000, 33000-60000 } dnat to 103.135.241.26

ip daddr 10.70.41.148 ct mark set 0x00000011 tcp dport { 10000-32000, 33000-60000 } dnat to 103.135.241.26

dport是列表里的端口，sport则是客户端随机指定去连ap，或者ap随机指定去连后端，conntrack ipput链路的dport才是需要关注的端口。在虚拟机telnet 15030的35229端口（列表开放端口），

**15030用tcpdump看**

15:14:26.320314 IP 10.200.112.26.53566 > 10.70.41.148.35229: Flags [.], ack 1, win 515, length 0

15:14:26.374611 IP 10.70.41.148.35229 > 10.200.112.26.53566: Flags [.], ack 4, win 54, length 0

**12002上看**

15:14:46.407441 IP 10.230.233.26.35229 > 18.139.91.11.53566: Flags [.], ack 4, win 54, length 0

15:14:46.572075 IP 18.139.91.11.53566 > 10.230.233.26.35229: Flags [.], ack 1, win 515, length 0

**15030会使用跟客户端源端口一样的端口53566作为源端口去连12002**

ap和proxy能telnet通的端口都是列表里ap这一列对应端口，游戏服能telnet的端口为列表游戏服对应的这一列端口，属于真正的业务端口。

客户端(源ip源端口----dst\_ip=ap,dst\_port=10003)---->ap(src\_ip=客户端，src\_port=10000-60000随机----dst\_ip=proxy,dst\_port=10003)---->proxy(src\_ip=客户端，src\_prot=10000-65000,dst\_ip=游戏服，dst\_port=443)---->游戏服（这是conntrack的记录，会记录客户端源ip；tcpdump的记录应该是每一段的src和dst；也不是很准确，gw机器上看到的是转发连接（不用gw机器处理直接转发?），是ap到proxy的ip和端口）

**节点机器15030上conntrack记录**conntrack -L ##中转ep

tcp 6 299 ESTABLISHED src=49.48.228.3 dst=10.70.41.148 sport=48410 dport=35066 src=103.135.241.26 dst=10.70.41.23 sport=35066 dport=48410 [ASSURED] mark=17 use=1

客户端49.48.228.3，15030-eth1-10.70.41.148；节点backend12002-103.135.241.26，15030-eth0-10.70.41.23，dport=35066为列表对外开放端口

**sudo conntrack.py nat**

{"cluster\_region": "hangzhou", "project": "h55na", "use": "1", "nft\_access": "", "protocol": "tc

p", "network": "ipv4", "assured": true, "gid": "a206", "timestamp": 1587698026.949795, "region":

"singapore", "protocol\_number": "6", "proxy\_port": 4090, "mark": "17", "network\_number": "2", "

state": "ESTABLISHED", "expire": "3599", "cluster\_id": "c15", "proxy\_host": "10.200.103.42", "ou

tput": {"dport": "58012", "src": "103.135.241.26", "dst": "10.70.41.23", "sport": "35046"}, "inp

ut": {"dport": "35046", "src": "202.185.131.114", "dst": "10.70.41.148", "sport": "58012"}, "mac

hine\_id": 253179}

**韩国15032-ap03，a227指向日本proxy**

{"cluster\_region": "tokyo", "project": "h55na", "use": "1", "nft\_access": "", "protocol": "tcp", "network": "ipv4", "assured": true, "gid": "a227", "timestamp": 1587698961.495953, "region": "seoul", "protocol\_number": "6", "proxy\_port":20047, "mark": "17", "network\_number": "2", "state": "ESTABLISHED", "expire": "299", "cluster\_id": "c3", "proxy\_host": "10.71.100.38", "output": {"dport": "51252", "src": "10.71.100.112", "dst": "10.76.18.39", "sport": "10934"}, "input": {"dport": "10934", "src": "223.62.173.179", "dst": "10.76.18.142", "sport": "51252"}, "machine\_id": 267782}

日本proxy15001-10.71.100.112，15032-eth0-10.76.18.39；客户端223.62.173.179，15032-eth1-10.76.18.142

conntrack -L

tcp 6 3599 ESTABLISHED src=14.34.112.65 dst=10.76.18.142 sport=43280 dport=34111 src=10.71.100.112 dst=10.76.18.42 sport=34111 dport=43280 [ASSURED] mark=17 use=1

dport=34111 这个端口在韩国列表里34102:1:14

"dport": "10934"对应列表端口10919:1:39

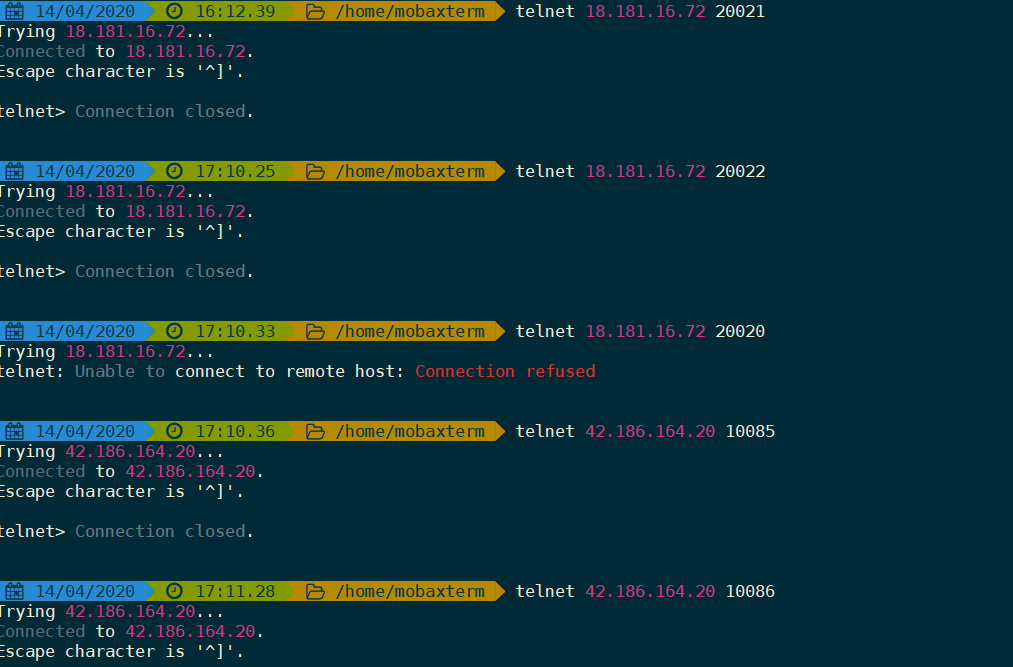
对15030的tcp70002端口进行telnet，15030上能看到来自虚拟机源ip，在15030的后端12002是tcpdump可以看到来自15030的请求，而不是客户端虚拟机的ip（请求经15030转发）。同理proxy机器上的tcpdump应该也是看到ap的ip，而不是客户端的ip，但是conntrack里会保留客户端ip信息，

1. 端口映射一一对应，例如g97na，列表里测试服18.181.16.72只使用20021和20022端口，列表里多开放了端口，但多出来的端口无法连通。

18.181.16.72:20015:1:19 cn 42.186.164.20:10079:1:19 cn-1:42.186.164.30

<https://impression.update.netease.com/lighten/pathn_g97na_cn.txt>

telnet 18.181.16.72 20021/20022能通，对应的telnet 42.186.164.20 10085/10086能通



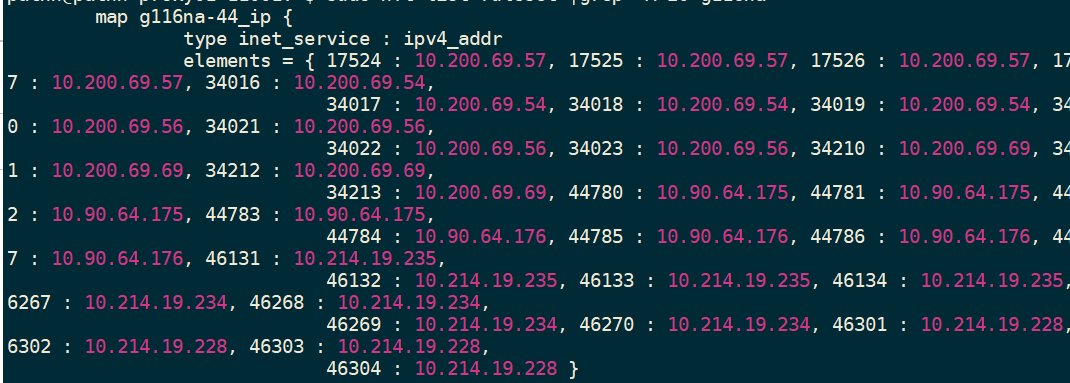
**列表出现内网**

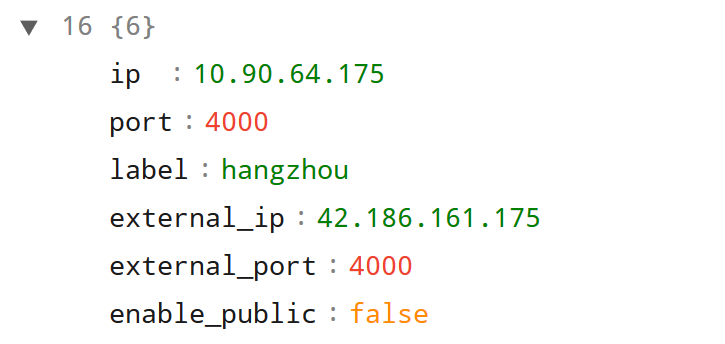
lighten列表的ip和数据源返回的ip都用的是external\_ip，没有external\_ip的时候会用数据项的ip字段；

pathn节点连的是ip，但在列表上显示的是external\_ip（没有external\_ip时会用ip）, 数据源返回的数据跟列表是一致的；

<https://impression.update.netease.com/lighten/pathn_g116na_sg.txt>

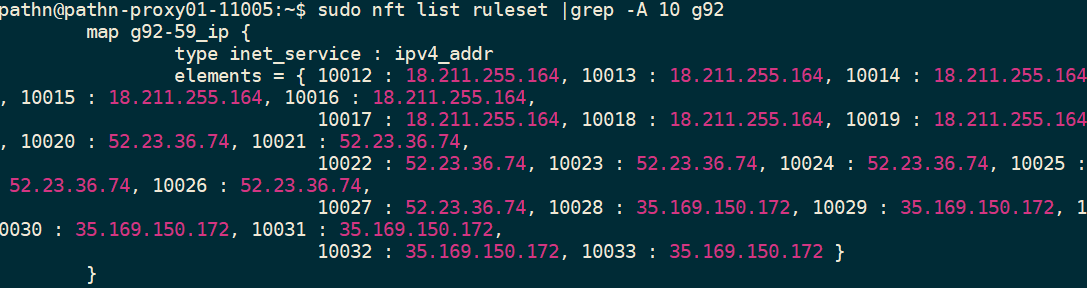
**g116na列表里同时出现内网和外网**ip，proxy11001-01里nft对应g116na的ip集合都是内网，即proxy连接项目机器会使用内网（排除proxy的nft未同步的情况，应该是sync过了。每次galaxy修改数据项，都会触发同步到proxy机器）。





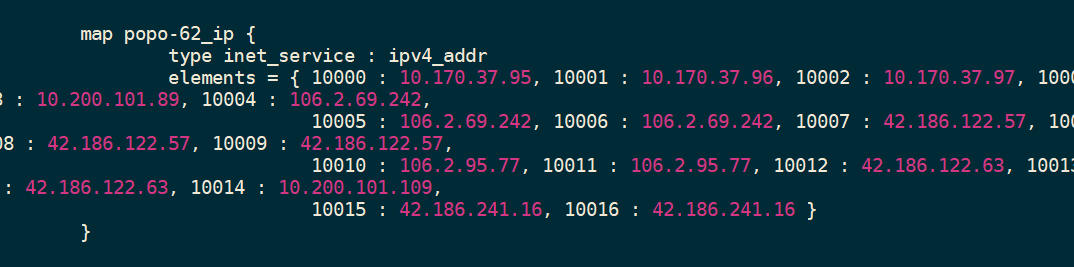
应该是有些服务器ip没有提external\_ip的值，里面的10.214的内网出现在列表，应该是这部分没有外网。

**g92海外集群，列表只有外网**https://impression.update.netease.com/lighten/pathn\_g92\_eu.txt，应该是enable\_public=true



popo集群，列表里显示都是外网，但proxy里面有内网ip

<https://impression.update.netease.com/lighten/pathn_popo_eu.txt>



是数据项哪一个参数的作用？？？？会不会是内网服务端的external\_ip没设置，ip字段为内网；另外有外网的服务段external\_ip设置了外网ip，这样列表会取了external\_ip字段生成列表，但节点连接还是会使用ip字段。

#external\_ip配置的情况下，列表里都会出现外网，至于会不会有内网，得看部分数据项是不是存在只配置内网不配置外网的情况。如果enable\_public=true，则nft里应该出现的是外网ip，但如果有些数据项没给外网ip，应该会造成这部分内网出现在nft里，这部分是proxy会去连接的ip。

**cbc操作：**

## 同步代码，初始化网络探测节点

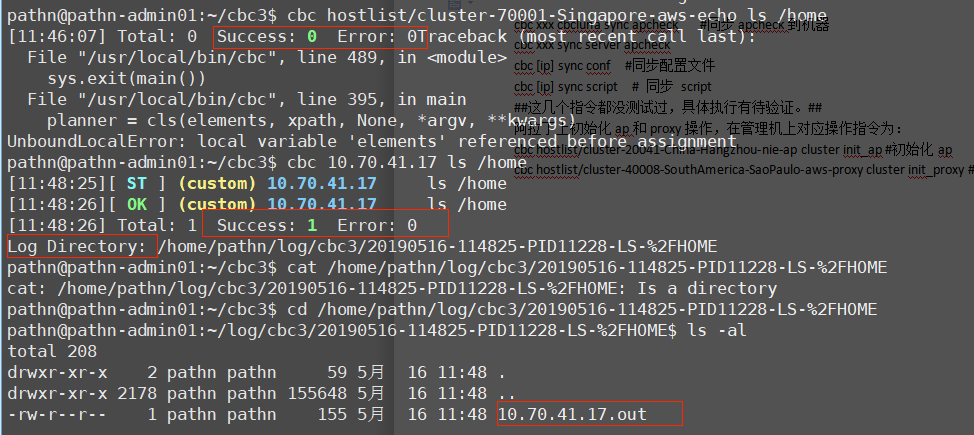
cbc hostlist/cluster-330xx cluster init\_probe # 探测机器

一般在管理机操作（阿拉丁的操作请求也是会提交到管理机），需要进到/home/pathn/cbc3目录下，一般指令格式：

cbc [操作对象ip] [操作指令]

cbc hostlist/cluster-70001-Singapore-aws-echo ls /home #使用主机名称指定时，操作没有执行

cbc 10.70.41.17 ls /home #直接使用ip时，提示操作完成，输出日志



cbc xxx cbcluna sync apcheck #同步apcheck到机器

cbc xxx sync server apcheck

cbc [ip] sync config #同步配置文件

cbc [ip] sync script # 同步 script

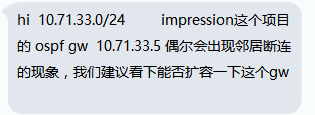
##这几个指令都没在管理机操作过，具体执行有待验证。##

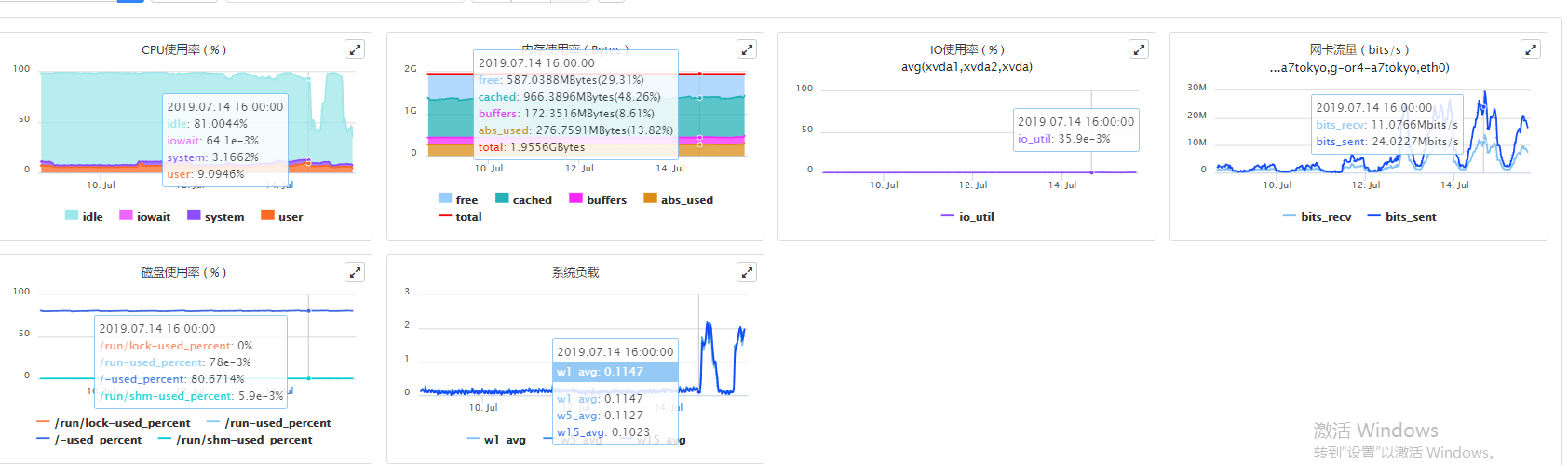
阿拉丁上初始化ap和proxy操作，在管理机上对应操作指令为：

cbc hostlist/cluster-20041-China-Hangzhou-nie-ap cluster init\_ap #初始化ap

cbc hostlist/cluster-40008-SouthAmerica-SaoPaulo-aws-proxy cluster init\_proxy #初始化proxy

**ospf值班反馈gw机器10.71.33.5负载高，偶发重连情况，建议扩容：**

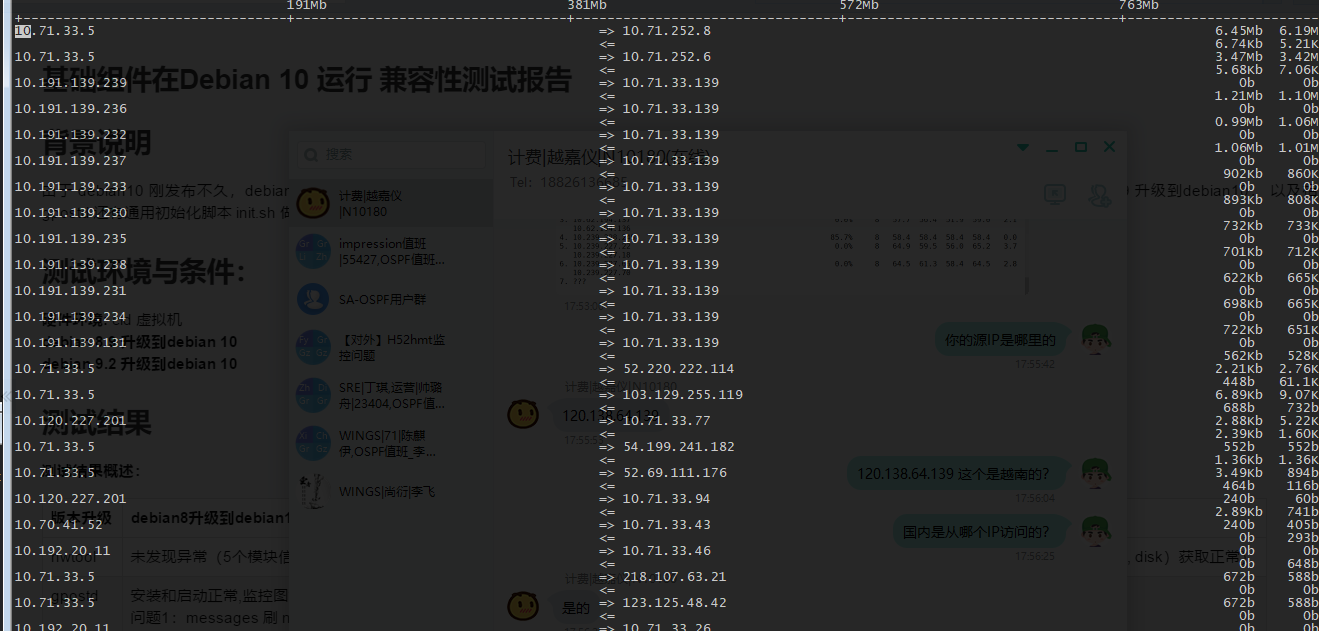




扩容步骤：skyline关机gw，对应网段会断网；skyline更改实例类型，实现扩容，之后重新开机。扩容不需要重新部署机器，会有开关机过程中网段的断网。

实际处理流程：

1. 让ospf值班帮忙查看gw上主要的流量来源，看是哪些机器流量走在gw上：



主要是10.71.33.139机器pathn-proxy01-10102，经查为缺少回程路由，导致流量走ospf

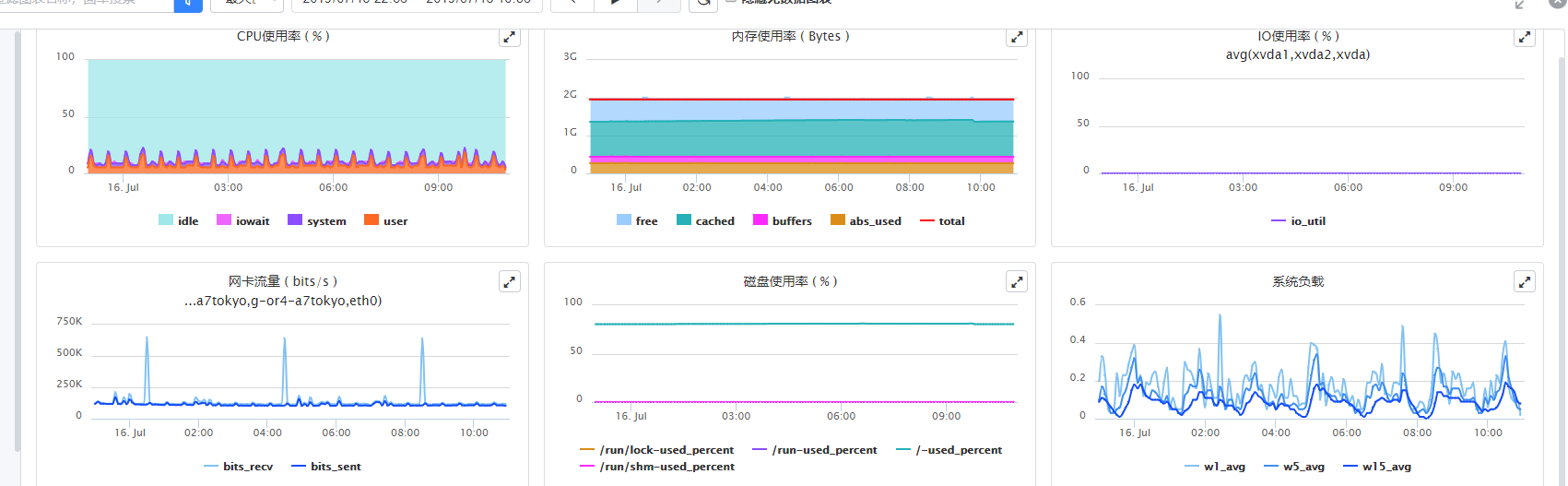


1. 调整机器路由，第二天再让ospf值班查看机器负载情况：

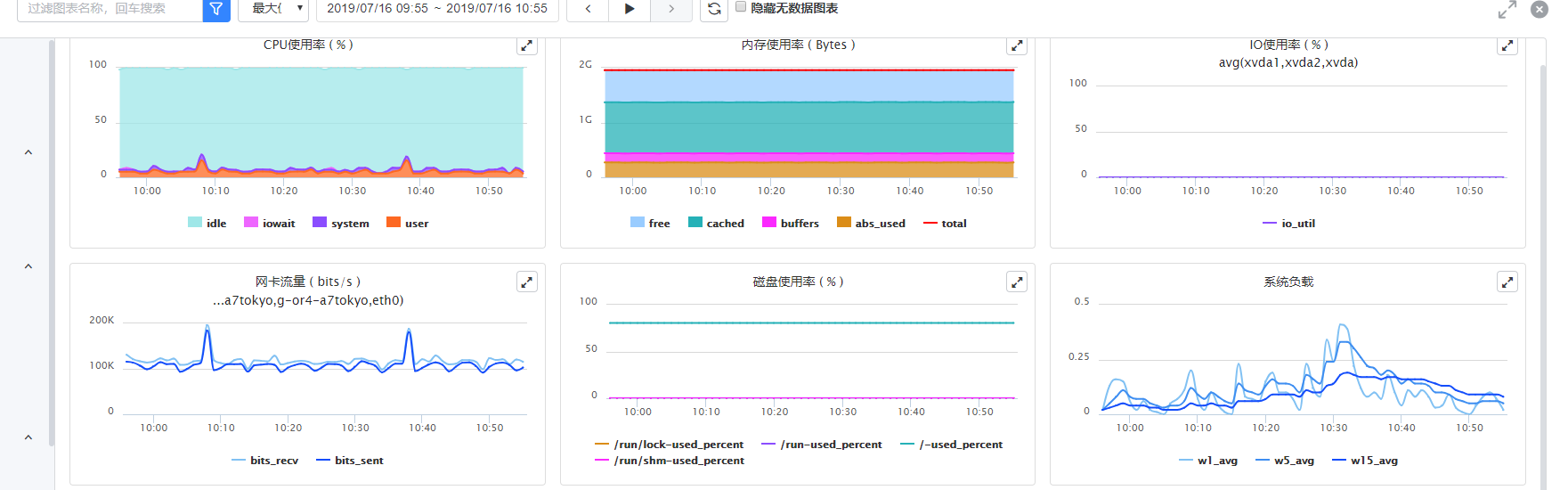
过去24小时：



过去12小时：



过去1小时：



1. 负载降下去了，路由调整生效，暂时不需要再扩容。

**aws机器上默认网关.1和ospf专线网关.5,以及pathn搭建网关.126/254说明：**

SRE|张日军 11:04:45

问个问题，aws节点机器，业务的转发流量一般是走.62/126/254这些pathn搭建的网关，但机器上默认的网关.1和.129gw跟ospf的gw机器是什么关系的，例如这个ospf的机器10.71.33.5

SA|郑康佳|konia 11:06:46

.1 是 AWS 自带的网关, 一般是公网流量出口

.5 是 ospf 搭的专线网关, 能通网易机房 10.x的网络

.126/254 是pathn 的专线搭的网关, 线路比ospf要好, 一般我们的流量要优先走自建网关

没资源就走ospf

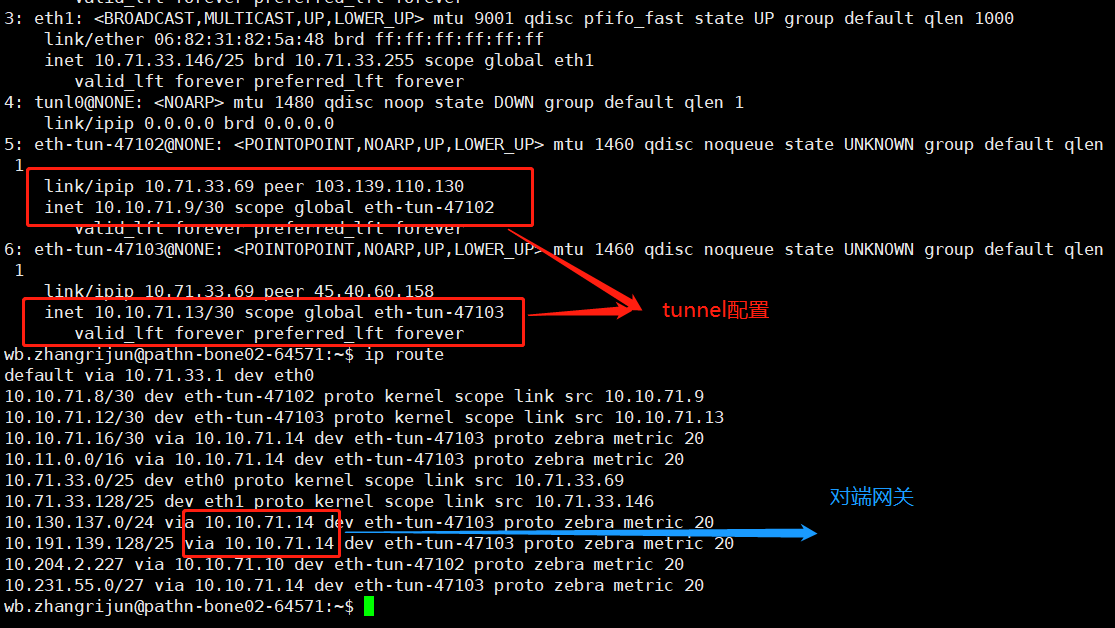
##v3后使用.120/250网关。

**海外aws与杭州机房互联**

**一**般是通过上海的出口，北京的出口，香港出口与aws互联（海外回杭州走内网时，出口网络丢包也会导致线路报警）

gw互联：

日本aws gw机器pathn-bone02-64571

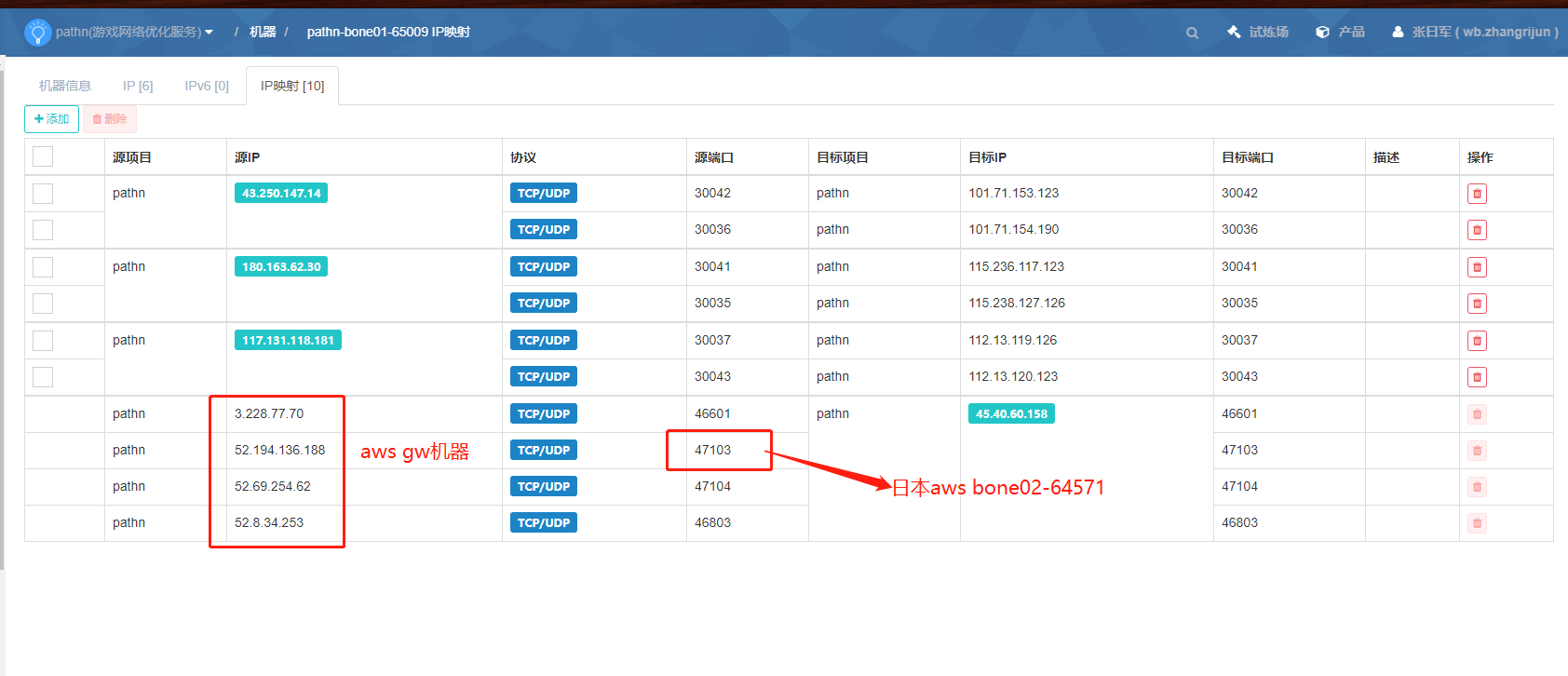


里面有路由

10.130.137.0/24 via 10.10.71.14 dev eth-tun-47103 proto zebra metric 20

10.191.139.128/25 via 10.10.71.14 dev eth-tun-47103 proto zebra metric 20

10.10.71.14在zenlayer机器bone01-65009上。

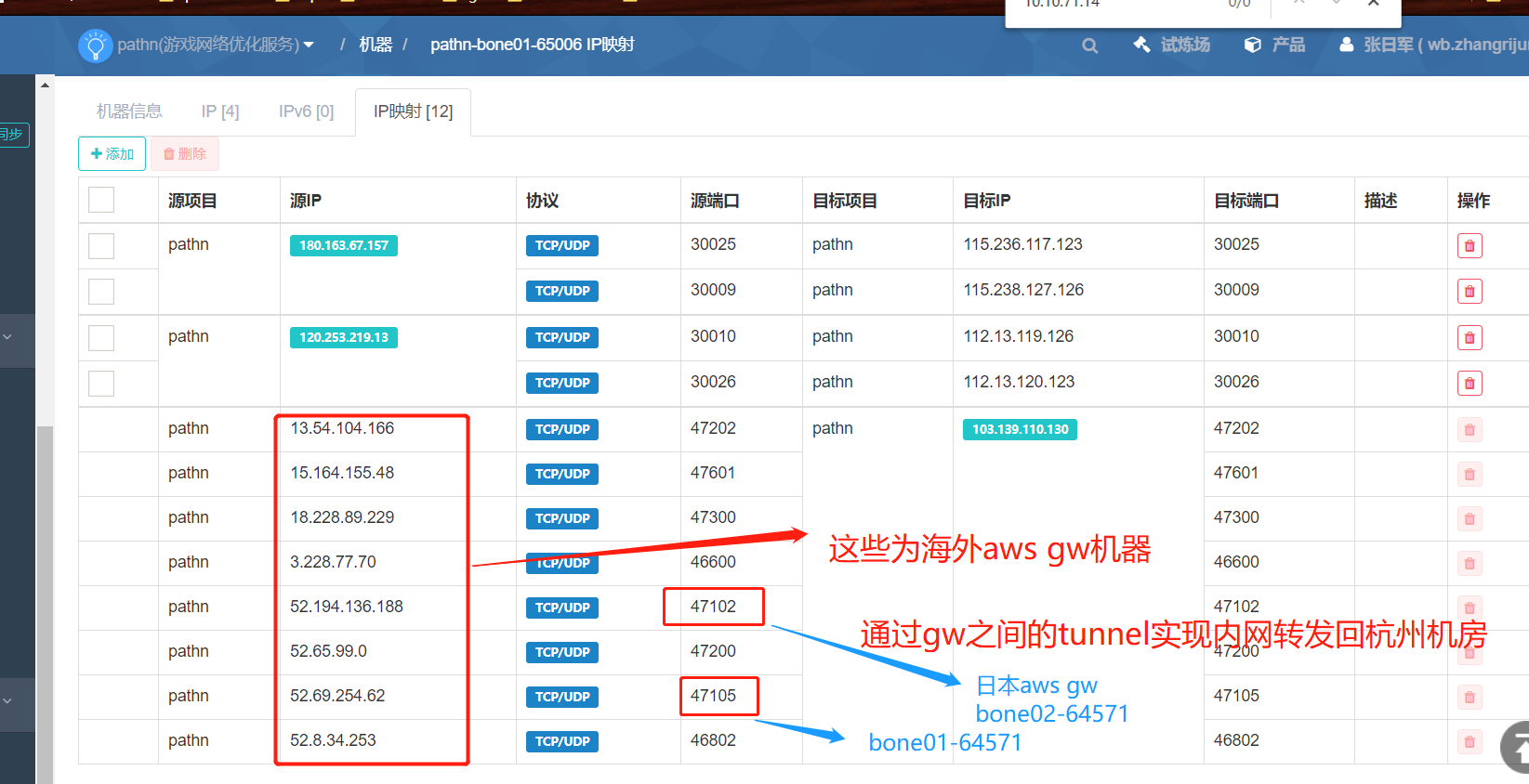


日本另一台gw bone01-64571，当前路由有：

10.130.137.0/24 via 10.10.71.22 dev eth-tun-47105 proto zebra metric 20

10.191.139.128/25 via 10.10.71.22 dev eth-tun-47105 proto zebra metric 20

对端gw为上海朗桥gw bone01-65006（10.10.71.22）



即日本aws回杭州有两个入口，一个是zenlayer，一个是朗桥。如果是日本到杭州内网线路告警，需要确认两个入口是否正常。

其他aws回国线路同理，需要查看入口的状况。法兰克福入口应该是中港（待确认）。

Peer如果是内网ip，如法兰克福的bone01-64575：

link/ipip 10.75.24.126 peer 10.200.122.21 ##bone02-65003，中港入口

inet 10.10.75.9/30 scope global eth-tun-47502

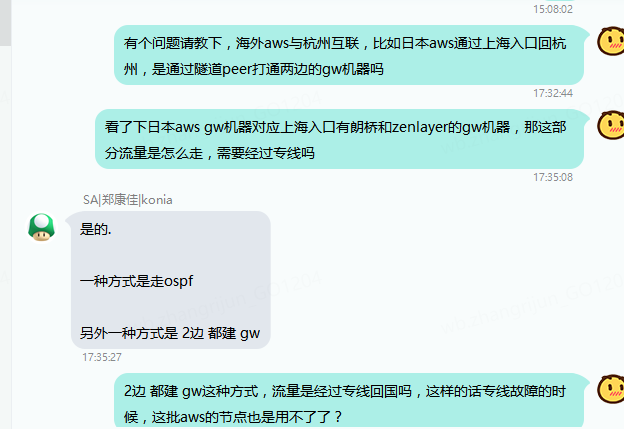
是不是就是走中港的aws dx，

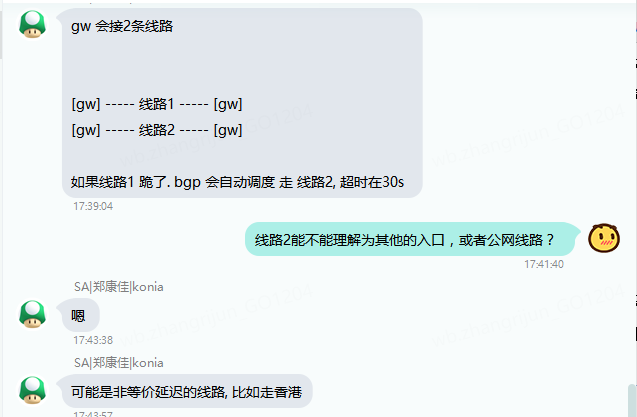
Peer如果是外网（一般情况下都是外网），走公网：

link/ipip 10.75.24.126 peer 103.135.241.62 ##peer的配置对应galaxy上面的ip映射

inet 10.10.75.13/30 scope global eth-tun-47503 ##这是ip route里面当前生效的tunnel\_id 47503

gw说明：



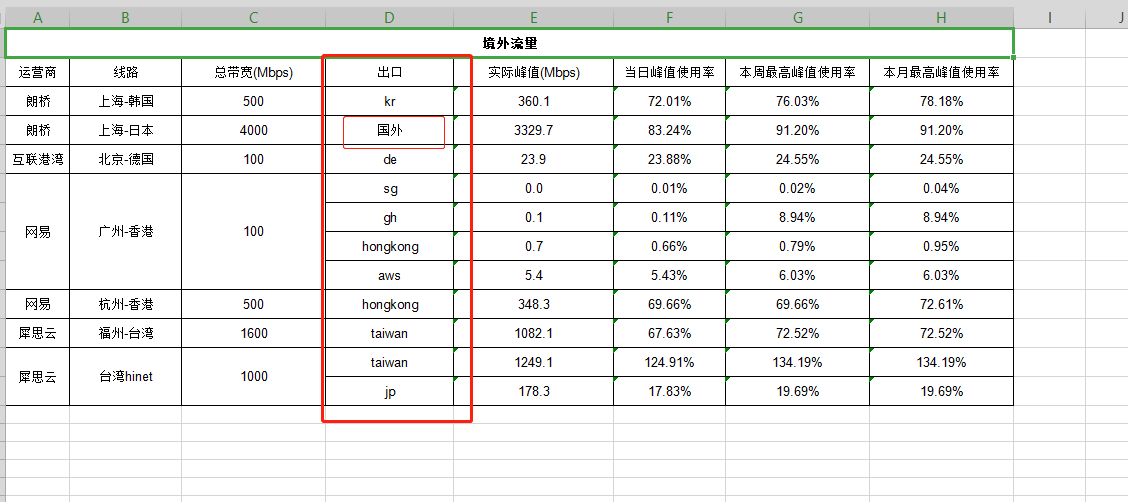


**海外带宽统计postman操作**

在新增朗桥日本线路180.163.64.4机器时，错误使用了添加专线信息的这个方法：

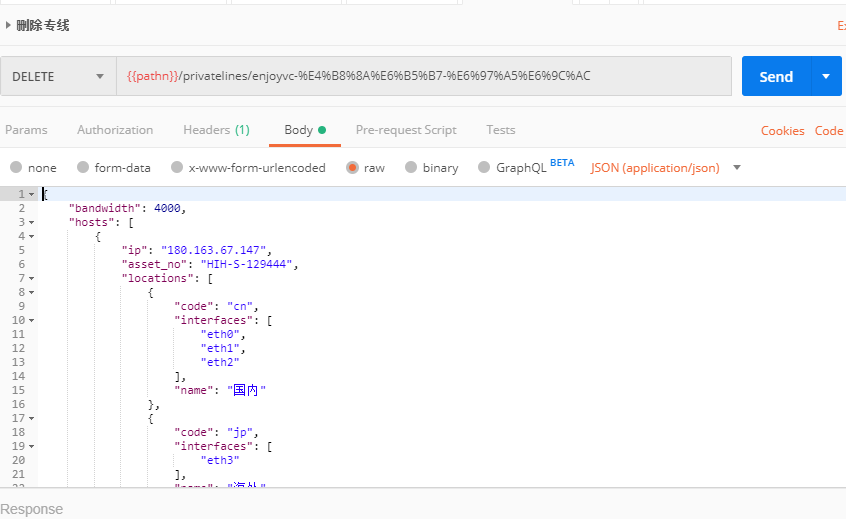


导致朗桥-日本方向之前配置被覆盖，只剩下180.163.64.4这台机器，需要重新导入配置文件。导入excel配置后，发现180.163.64.4这台机器的配置中name的字段值跟其他机器不一样，其他机器显示为null（未定义），这台显示为国内和国外，导入峰值数据是出口那一列也是显示“国内”“国外”，跟其他机器显示的“cn”“jp”不一致。



之后又使用更新专线机器主机名这个方法，首先将name的值设为“null”，更新后显示name：“null”（字符串），跟其他机器显示name：null（未定义）有差异；出口这列显示为null，不符合预期；之后再把name字段直接删除，不做定义，继续post，发现无法覆盖，还是保留了name：“null”（字符串）这个值。

最后使用删除专线这个接口，指定清除180.163.64.4这个机器，指定重新导入配置excel，之后出口这一列显示符合预期。



**g69申请临时北美加速入口，采用写列表方式接入：**

更新Project

URL：PUT /projects/{project\_code}

1.Postman开通项目权限，关键参数 "return\_data\_to\_galaxy":1，允许项目从galaxy拿到返回数据。



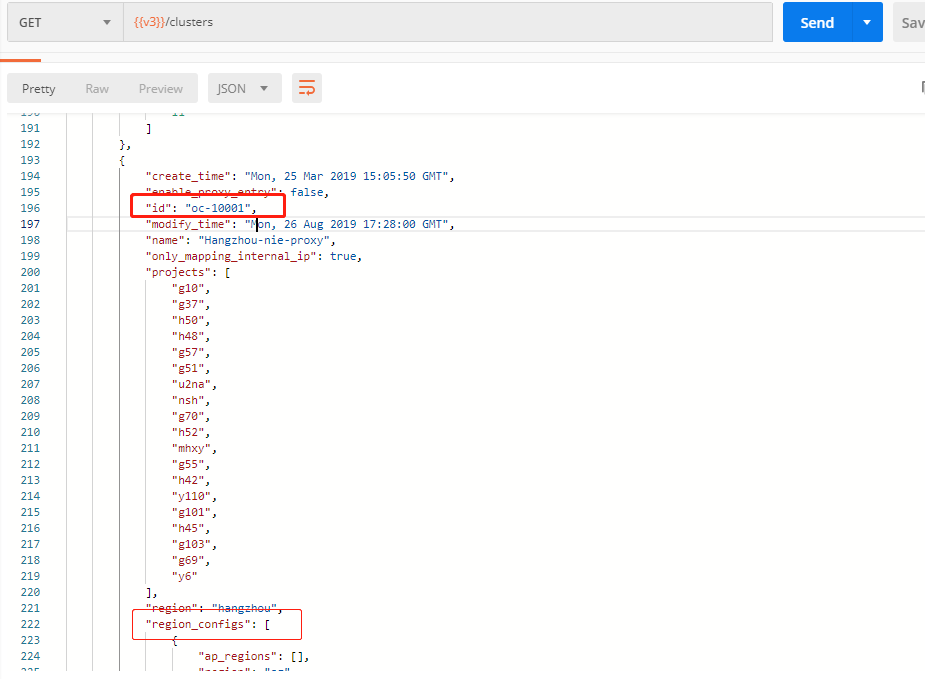
2.项目需要在galaxy准备好数据项（服务器ip端口列表），之后pathn开通权限后，项目新建外部数据源，url选择pathn\_v3,数据源选择之前准备的数据项，提交后可以预览返回的加速节点ip和端口信息。

3.之后回复项目邮件，告知开通完成。

变更：先把项目添加到对应cluster的project里面，之后使用更新project接口修改return\_data\_to\_galaxy参数为1.

对于项目使用容器或者bigworld项目，需要指定：外网Ip + enable\_public 参数。

**Pathn新环境相关操作变动：**



SRE|张日军 15:01:22

如果需要指定项目使用特定的region，是需要另外创建cluster吗，像y6只想使用us，sg和taiwan这几个region这种情况

SRE|黄志雄|大雄 15:02:50

不需要，之前这个限制是通过ap\_regions限定的，现在先取消这个限制吧

SRE|黄志雄|大雄 15:03:30

只要确保项目想开通的区域都在cluster里就好

新增项目使用patch方法，添加cluster的projects列表。

移除cluster的区域配置，使用DELETE /clusters/{cluster\_id}/region\_configs

# 请求体

{

"region\_configs": [

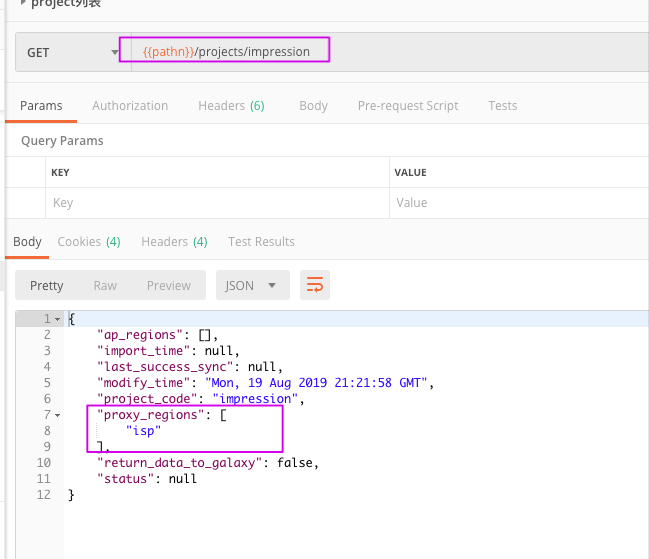
{ # sg的配置将从该Cluster的regiong\_configs中移除

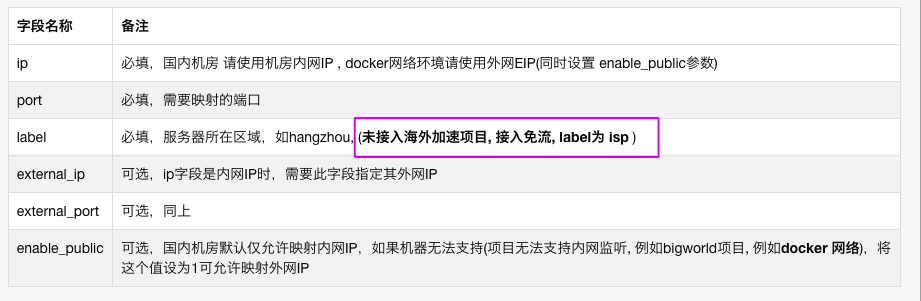
"region": "sg"

}]

}

**项目接入移动联通免流：**





如果项目 没有开通过 hangzhou region, 申请免流的话, 就给开通 ( label isp)

"proxy\_regions": ["isp"]

如果项目开通过 hangzhou region的话, 申请接入免流, 就给开通 ( label hangzhou)

"ap\_regions": ["isp"]

SA|郑康佳|konia 2019-08-19 21:35:34

已经接过pathn 海外加速什么的

SA|郑康佳|konia 2019-08-19 21:35:42

就只能 开通ap

如果没有接过, 就直接设置 isp. isp集群也就是你下午搞的6台集群. 专门做这个的.

**2019-8-26引入pathn新环境**，杭州共享集群10001，将项目加入到cluster10001的projects列表里即可，默认项目可以使用cluster下region\_configs里的所有reign。

免流接入：如果项目之前接入过pathn，则把项目添加到10001集群的projects里即可，一般项目已经在里面（使用10001proxy的项目），项目发起免流流程时，不需要要额外的操作，需要把列表发出来给到项目即可；目前使用免流节点20061和20062，（新节点为12007）；

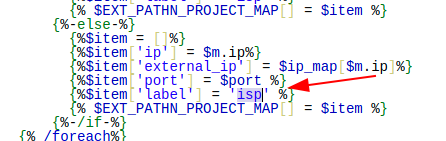
对于没接过pathn的项目，需要把项目添加到免流集群cluster-10006的projects去，之后通知项目sa把label设置为isp，提供列表。

**如果项目不需要使用海外加速，只是使用免流，可以不接sdk，直接开发。**

###20061和20062已经下线，免流节点使用12007，新项目（针对已经接入过海外加速的项目）接入使用c1公共集群，添加项目到c1的projects里面。

未接入海外加速的新项目接入免流，使用免流集群oc-10006,label值填isp，该集群直接使用proxy作为接入点。添加项目到projects字段后，以cluster\_id触发生成映射。

**g18之前接入了免流，label=isp，再接入海外加速**



1.跟项目确认两个问题：游戏服是否一样，免流是否继续使用----同服，免流继续用

2.workflow申请流程，pathn这边重新给项目开一个集群

项目免流同服，新建一个独立集群，数据项修改label=hangzhou，免流使用ap\_region=isp接入。

**东南亚机器实现ssh跳转登录：**

需要下线群组1004机器，为东南亚zenlayer的ip，直接登录不了，选择从新加坡aws70001跳转：

登录70001，复制本地私钥到~.ssh/目录下，即id\_rsa\_2048，修改权限chmod 600 id\_rsa\_2048

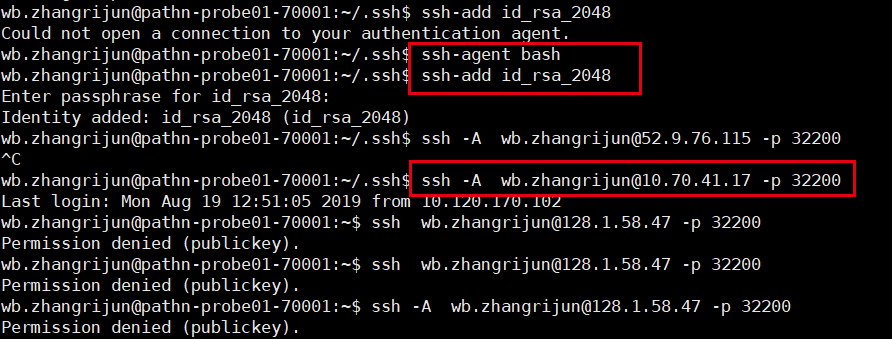
之后ssh-add id\_rsa\_2048,如果报错：

wb.zhangrijun@pathn-probe01-70001:~/.ssh$ ssh-add id\_rsa\_2048

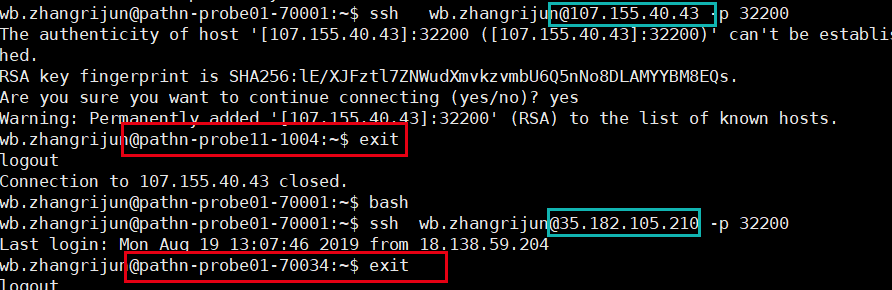
Could not open a connection to your authentication agent.

执行ssh-agent bash 开另外一个进程

重复ssh-add id\_rsa\_2048 ：



之后ssh -A 登录本机70001，此时的bash可以进行转发ssh

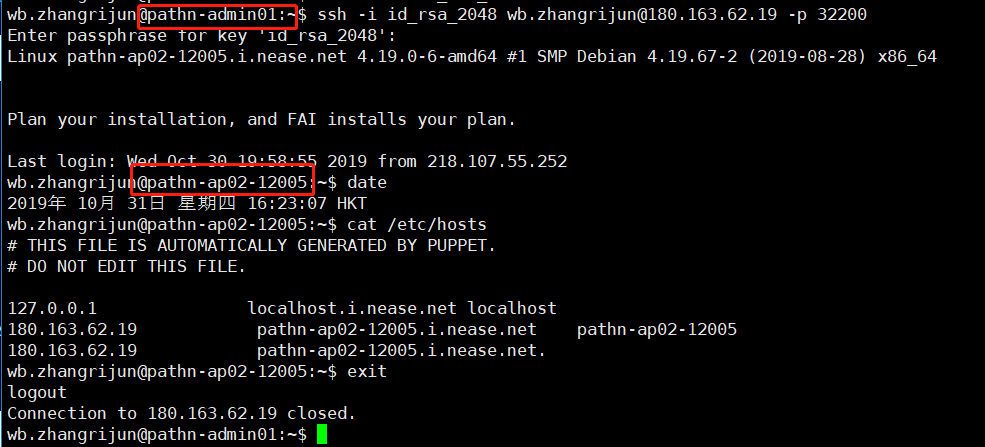


这个方式会导致ps aux |grep ssh里面ssh-agent的进程数越来越多，pstree里面的ssh-agent越来越多。？？？？？能不能直接kill进程？



有更简单的方法，通过-i 指定私钥文件去登陆：

ssh -i id\_rsa\_2048 wb.zhangrijun@180.163.62.19 -p 32200



**ansible批量操作：**

为每台ARM机器跑以下命令：`systemctl enable puppet && systemctl start puppet`

可以用ansible完成，将所有arm机器ip放到一个hosts文件，然后运行：

ansible -i hosts all -b --become-method=su -K -m shell -a 'systemctl enable puppet && systemctl start puppet'

1. 需要在hosts文件里指定ssh端口，后面添加两行

[all:vars]

ansible\_ssh\_port=32200

2.直接运行是会提示publickey问题，无法连接，需要通过跳转（使用新加坡70001机器），ssh-add id\_rsa\_2048添加自身私钥，再次登录自身后执行ansible正常。

3.登录后su至root用户操作

写在host文件里

# hosts 配置

[xx]

10.0.0.1 ansible\_su\_pass={passwd}

或

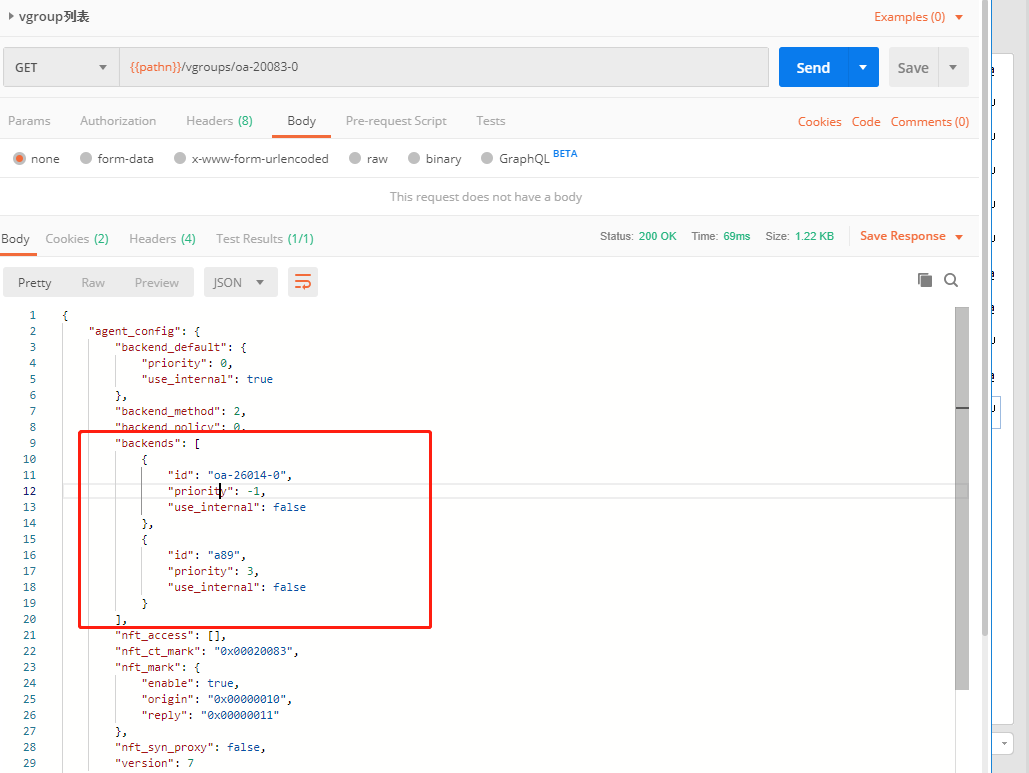
[xx:vars]

ansible\_su\_pass={passwd}

**10001集群项目使用aws转发到专线ep说明：**



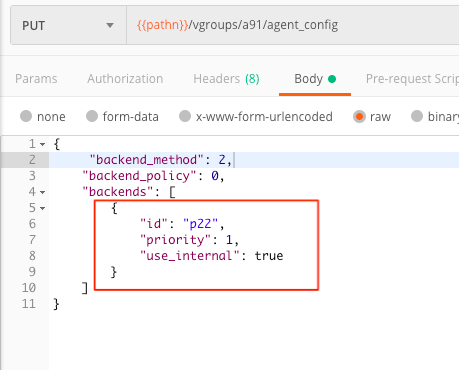
ep一般设置为备用状态，status=2,不影响使用。20084/20083/20053的backend里有多个ep，通过priority指定优先级，值越大，优先级越高。当前20083指定后端为a89。接口对agent\_config进行修改后(修改前对节点重新跑一次初始化)，需要在机器上进行agent upgrade同步配置，或者管理机上cbc xxx apagent upgrade。



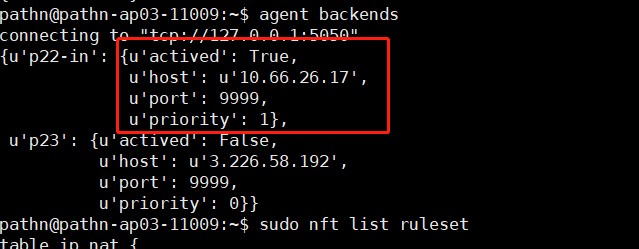
**海外机器互联如果有报警**

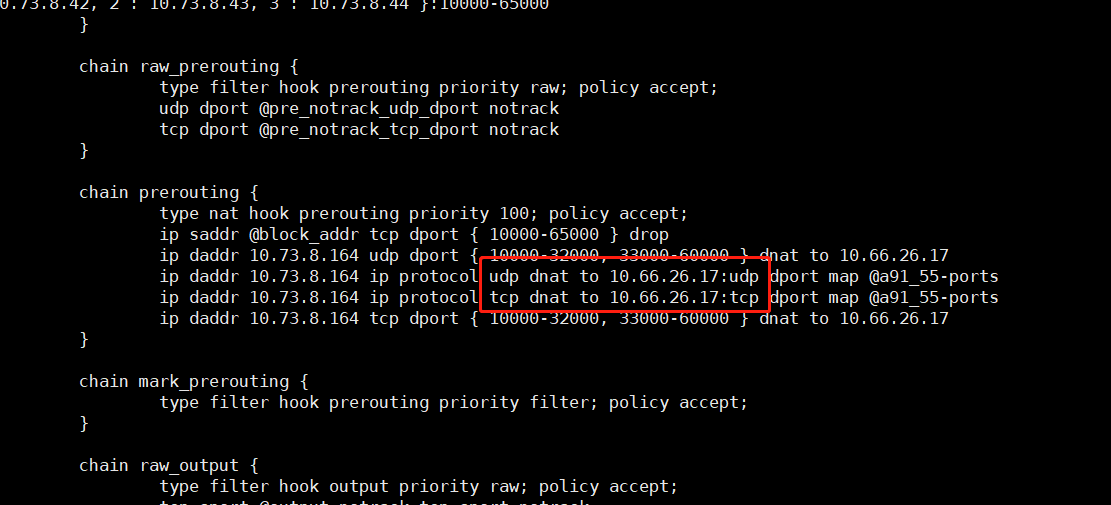
1. 先把aws peering 打通. mtr测试 没有多余跳数 (下一跳就到, 多跳数 一般有问题)

2.把 其中一个 proxy 设置为 use\_internal: true

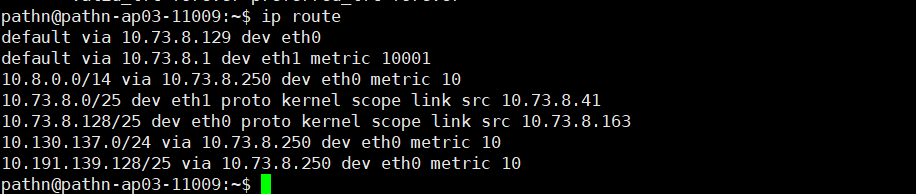


a91@11009，此时的对应后端和nft为：





对应路由：



并没有设置去往10.66.26.17的策略路由，走的是默认路由.129，其内网实现是通过aws 的peering，不是走pathn的自建gw（海外--杭州的情况）。

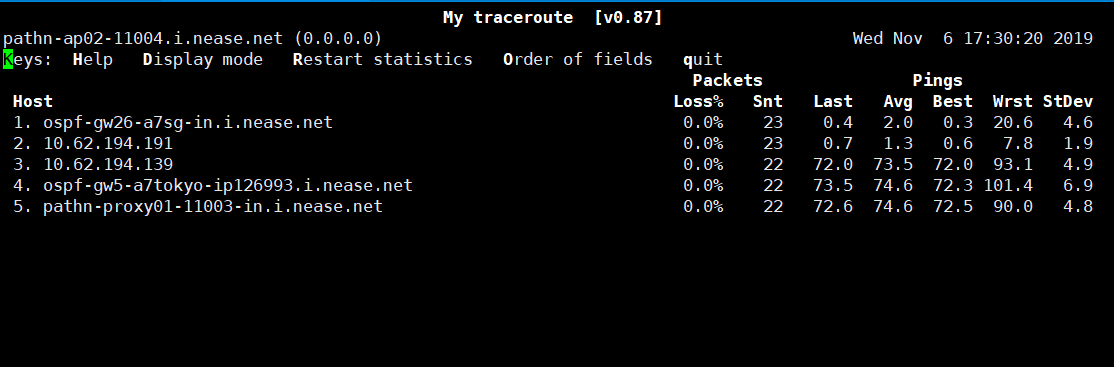
**之前未在aws建立peering时，海外---海外直接使用了内网，**流量是走在ospf的内网上，导致流量上涨后出现大面积丢包和延时报警：



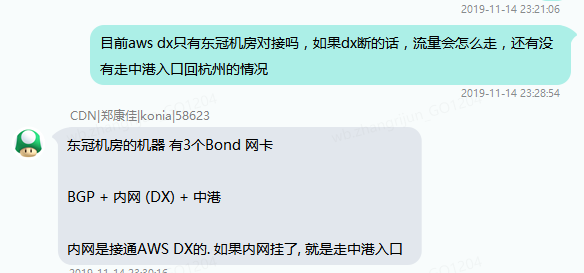
未建立peering时mtr为：

新加坡11004上mtr到日本11003，会经过ospf

pathn@pathn-ap02-11004:~$ mtr 10.71.33.176



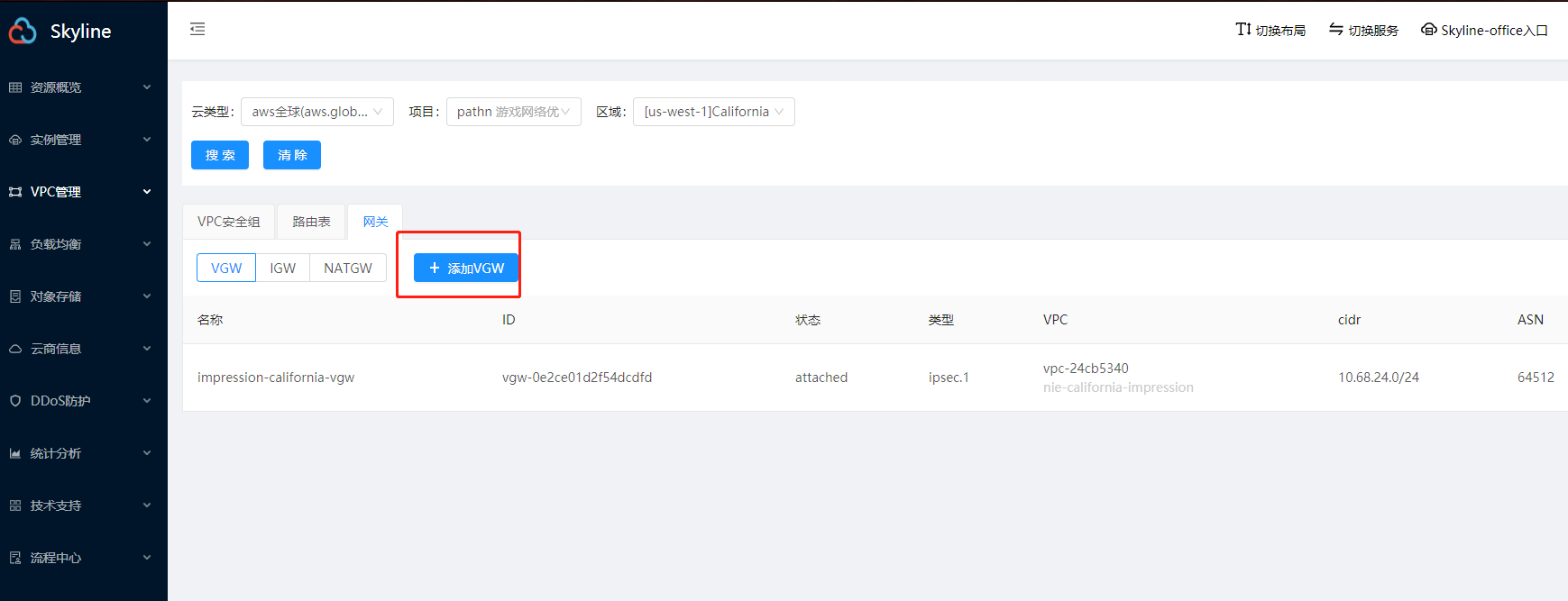
**aws dx 说明**

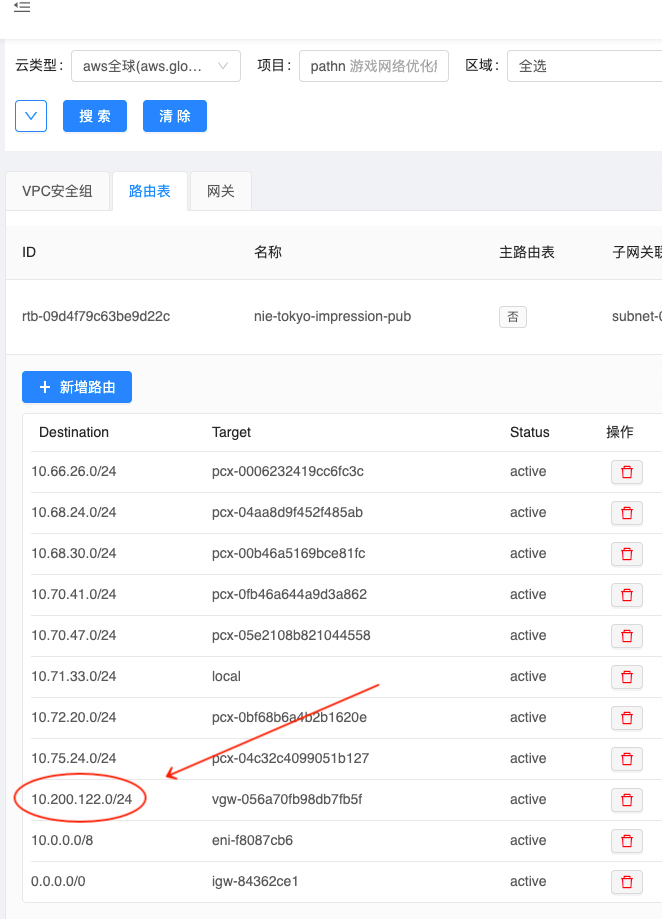


接DX 不用peering,. 是在vpc 加路由

<https://skyline.nie.netease.com/aws/network>

<https://g.126.fm/01GJUv6目前aws> region对应dx的网关情况vgw，在vpc管理界面，点击网关进去，有添加VGW按钮





CDN|郑康佳|konia|58623 2019-11-14 23:39:46

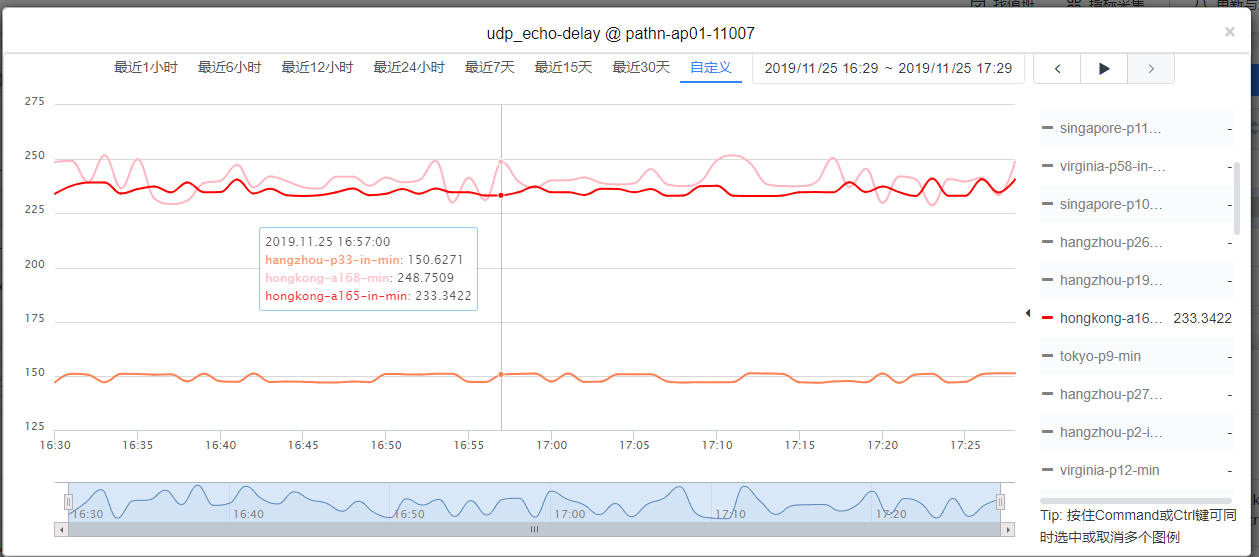
要手动去每个vpc 的路由表里面 加上杭州的内网网段 走向 vgw

CDN|郑康佳|konia|58623 2019-11-14 23:40:16

可以实现 统一走内网回杭州

CDN|郑康佳|konia|58623 2019-11-14 23:40:39

不过走香港 延迟并不是最佳的



法兰克福11007ap-01，流量回杭州，a165走dx；a168走中港；p33走pathn的自搭gw，dx和中港的延时比较差。

**5月27号California全线节点回杭州中断**

[pathn][2020-05-27 16:13:24][udp\_echo监控-丢包率超出阈值]：

当前共8个，异常8个，新增影响8个

处理：https://g.126.fm/01nQBt6

影响:

pathn-bone02-65003 | 42.186.27.21 | Hangzhou-hk-nie-gw (异常)

- 后端：TO-64568\_California-aws-gw\_46804

- 当前值：100

pathn-ap01-15048 | 54.219.14.50 | California-aws-popo (异常)

- 后端：hangzhou-p69-in

- 当前值：100

pathn-bone01-64568 | 52.8.34.253 | California-aws-gw (异常)

- 后端：TO-65003\_Hangzhou-hk-nie-gw\_46804

- 当前值：100

pathn-ap01-15048 | 54.219.14.50 | California-aws-popo (异常)

- 后端：hangzhou-p68-in

- 当前值：100

pathn-ap02-40006 | 13.56.167.87 | California-aws-public (异常)

- 后端：hangzhou-p7-in

- 当前值：100

pathn-ap02-40006 | 13.56.167.87 | California-aws-public (异常)

- 后端：hangzhou-p8-in

- 当前值：100

pathn-ap02-15048 | 54.176.132.28 | California-aws-popo (异常)

- 后端：hangzhou-p69-in

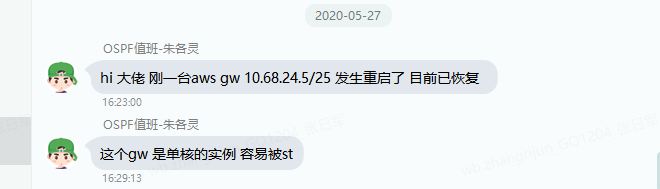
- 当前值：100

pathn-ap02-15048 | 54.176.132.28 | California-aws-popo (异常)

- 后端：hangzhou-p68-in

- 当前值：100

Ospf反馈有网关重启



但是正常流量不会走ospf，正常不会受影响才对；

**排查为区域vpc没加dx路由；**

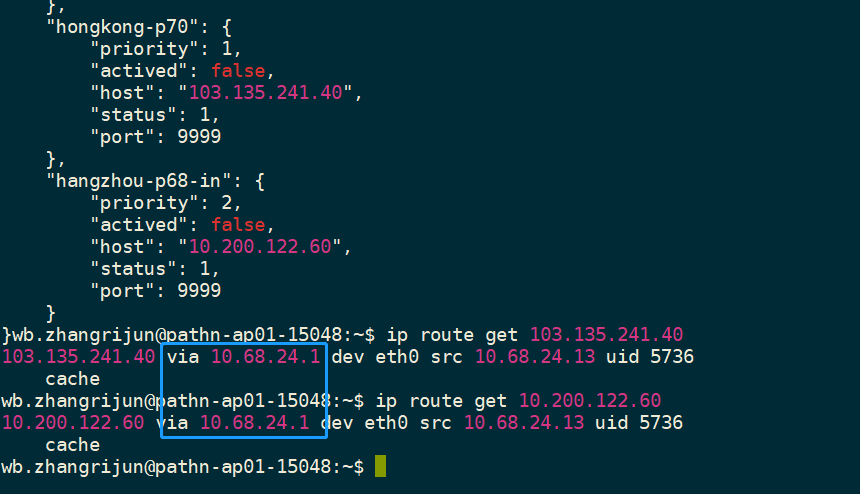
之前为什么未发现？----g37的美国节点后端使用了专线日本中转节点，默认的proxy，以p开头的后端，才是走gw--gw的线路，所以gw--gw线路平常流量很小，而且pathn自建aws gw，正常流量一般不会造成压力。只有内网走ospf gw时，在流量暴涨的情况会出现丢包，受限于gw实例类型。小流量走在ospf线路正常，除非检查路由，mtr，不然只有出问题才发现。所以故障出现时，g37的流量是不受影响的，只有香港后端a196（dx内网10.200.122.50）有报警。--去往10.200.122.50匹配到ospf。

popo群组目前没有流量，后端p68,p69为内网，走的是gw--gw，正常应该走pathn的gw才对，为什么走到了ospf gw？----15048机器eth1入库对外，eth0不入库，流量经过eth0转发到后端，未加dx的路由前，应该是内网后端10.200.122.60匹配到了ospf网关，经ospf到中港入口，所以ospf gw异常重启时影响到了线路。

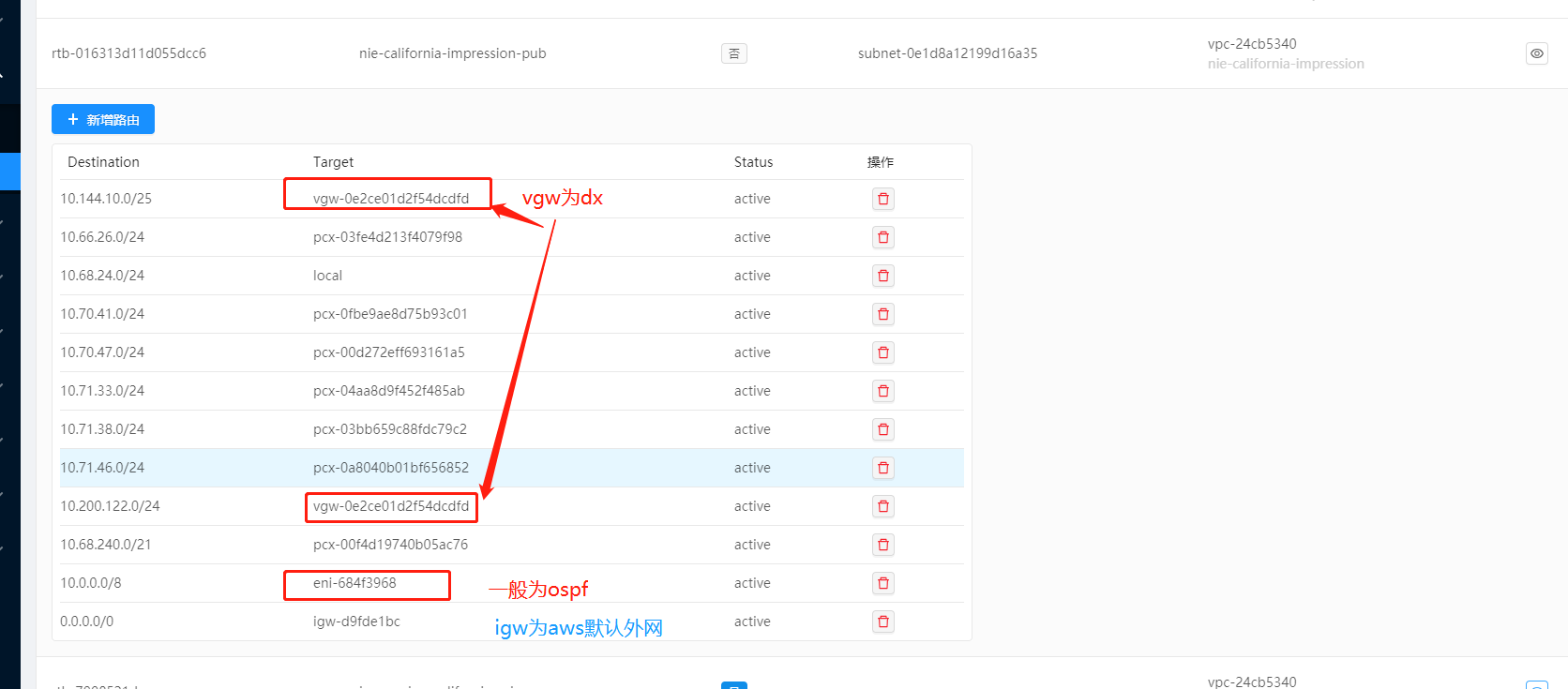
15048跟11006公共集群的节点路由情况不一样，11006里面有指定回杭州内网走 10.68.24.250这个64568这个pathn的gw，15048里eth0走.1网关，eth1走.129，没有指向64568。（15048eth0和11006eth1是10.68.24.0/25段，15048eth1和11006eth0是10.68.24.128/25段）

10.130.137.0/24 via 10.68.24.250 dev eth0 metric 10 ##11006

10.191.139.128/25 via 10.68.24.250 dev eth0 metric 10



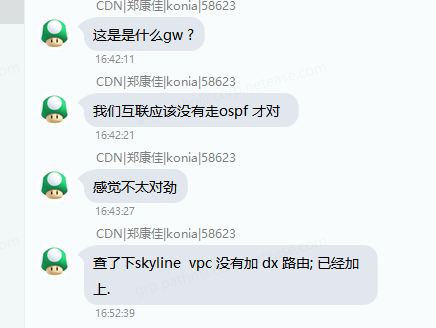
.1为aws默认外网网关，跟vpc路由表相关，下图为california vpc-24cb5340 subnet-0e1d8a12199d16a35的路由情况。未加dx路由之前，默认内网会匹配到10.0.0.0/8这条ospf路由。



skyline上面vpc-24cb5340 （10.68.24.0/24）分拆了两个子网，subnet-b4024fec（10.68.24.0/25）和subnet-0e1d8a12199d16a35（10.68.24.128/25），两个子网都有各自的路由表。

aws机器上有路由，skyline上也有路由，如何区分和理解？----机器上的路由是直达路由，如pathn的gw，是同一个region下的内网，直接指向。skyline上面的路由，把aws的数据封装，通过宿主机层面去路由，即机器上没有指定的，一般是走.1（或者.129）默认网关，再根据skyline上面的路由去匹配。优先级是先从机器上路由匹配，再到skyline路由表匹配。

15048能不能指定走pathn gw？----可以，但没必要。后端的内网是10.200.122的中港段，直接有dx路由，没必要走pathn的这条路。



刚开始aws region的内网跟杭州机房内网是不通的，需要接入ospf，后面有了dx，多了一种互联情况，dx不适合跨云互联的情况。

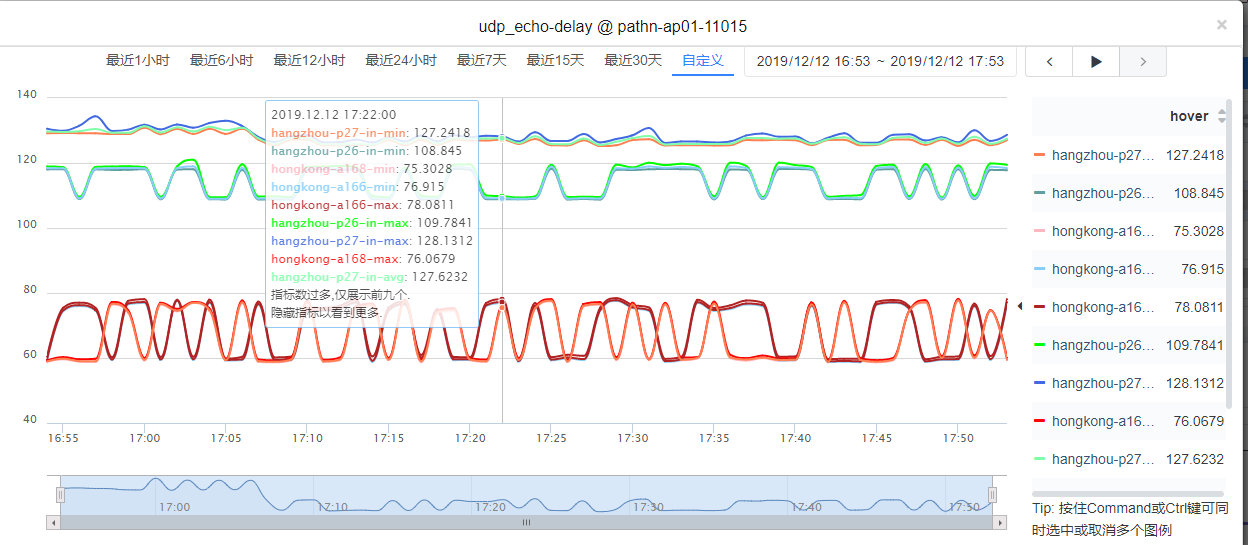
**Skyline页面上操作peering和路由表：**

Peering通过/24对应vpc的id去添加，路由则需要具体分.0/25和.128/25去分别添加路由表。

**东南亚ucloud节点11014-11016**

目前节点都是通过走中港12002中转回杭州机房，东南亚ucloud--->外网--->中港中转ep a165或a168--->杭州中港专线---->杭州

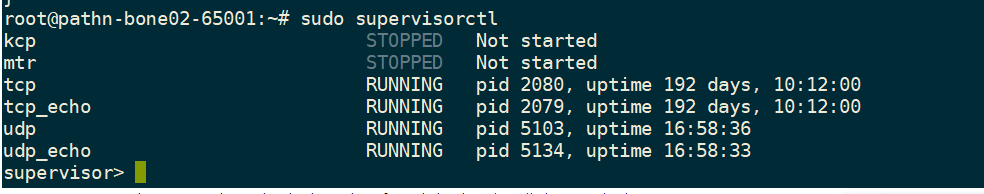
越南ucloud11015当前后端延时情况



几个节点上ping管理机，内网10.200.122.20的延时比外网103.135.241.10（中港）延时差，但比42.186.27.20好，主要当前ospf链路情况为先到新加坡，然后到杭州核心交换机，再到具体机房（之前是到新加坡 --上海 --杭州核心）。就当前来说，直接走外网都中港，再中转回杭州是最佳线路。

**重启echo服务**

sudo supervisorctl ；supervisor help查看帮助信息



之后 restart/start/stop udp操作

应该还可以使用supervisorctl restart apagent的形式调用；

**Debian 10 arm实例eth0 ip无法登录，eth1正常**

这个是路由问题，可以先在管理机上cbc ... sync script（先在galaxy上生成配置），然后在/home/pathn/script/ec2net目录下执行make install命令。

跑Puppet需要先安装zerorpc

初始化报错，首次报错ec2net目录下有文件无权限删除，手动删除后，流程往下走；

进行到apagent upgrade 报错lost remote 10s，路由问题？目前nft能有输出；

arm对应puppet4链接wget -O initpuppet\_pp4.sh <http://puppet.x.netease.com/installpp/initpuppet.sh.pp4.arm>

Debian10 arm问题：

1. eth0 ip无法使用，同步脚本ec2net后正常；
2. puppet有报错，zerorpc相关，手动安装后puppet能完整跑完；
3. cbc初始化，第一次报错是ec2net脚本目录下build里有新文件导致rsync失败，删除后流程往后走；进行到apagent upgrade 报错 **Lost remote after 10s heartbeat**

zerorpc的问题，需要重装：

pip install --upgrade pip

pip uninstall future gevent greenlet msgpack-python pyzmq -y

pip install zerorpc

15041修改ip，主ip，5个内网ip使用中，日本ip使用中----网卡报修中，已修改ip

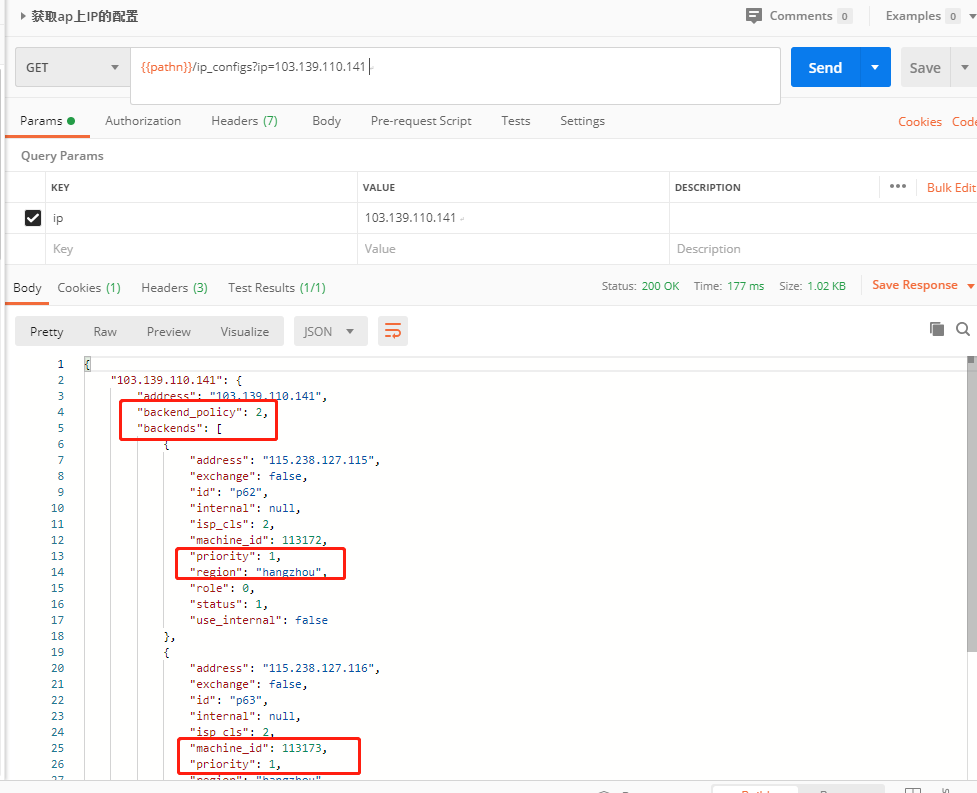
**15041galaxy修改更新ip时，修改主ip后**，hwtool信息无法正常上传，sn序列号等信息丢失，需要生成配置后同步，再不行，机器上执行删除/var/cache/hwtool里的内容，再python /usr/local/hwtool/script/hwcheck.py -d

**12011机器上a221的backend里同时存在两个proxy，p62和p63**

应该是由"backend\_policy": 2指定

后端选择策略（0：RAMDOM，1：RTT，2：RR，3：WRD）

RR是load balance，WRD，带权重的分配



****

"backend\_method"(默认为0，即ap会使用集群内所有的proxy作为后端；1则是使用backends里指定的后端；2为混合。

**朗桥网关机器更新移动ip后tunnel互联报警（65001--->65006）**

[pathn][2020-04-09 19:05:05][udp\_echo监控-丢包率超出阈值]：

当前共2个，异常2个，新增影响2个

处理：https://g.126.fm/03lPHeU

影响:

pathn-bone01-65001 | 115.236.117.123 | Hangzhou-multiisp-nie-gw (异常)

- 后端：TO-65006\_Shanghai-enjoyvc-gw\_30026

- 当前值：100

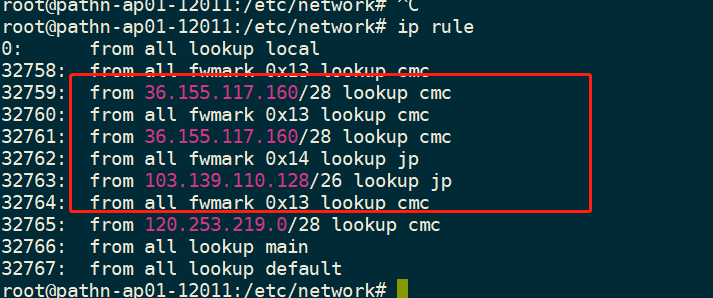
pathn-bone01-65001 | 115.236.117.123 | Hangzhou-multiisp-nie-gw (异常)

- 后端：TO-65006\_Shanghai-enjoyvc-gw\_30028

- 当前值：100

9号下午操作更新ip，ifdown 网卡，修改interface文件，等noc调整vlan后ifup网卡，新ip测试连通性正常。

问题一：ifconfig eth3 up 网卡不显示，随后ifup eth3，导致ip rule 里面显示重复的规则



**处理：使用ip rule del 删除旧的和重复的路由。**

**规范点操作：ifdown 网卡后，直接ip rule del 旧的相关路由，之后再修改interfaces文件，之后ifup 网卡，应该会报ipset的问题，不影响。**

问题二：65006网关机器的tunnel提前已拆除，但后续一直有丢包报警触发。

**处理：cbc sync config ##机器互联探测配置同步/home/pathn/conf/probe\_udp.conf**

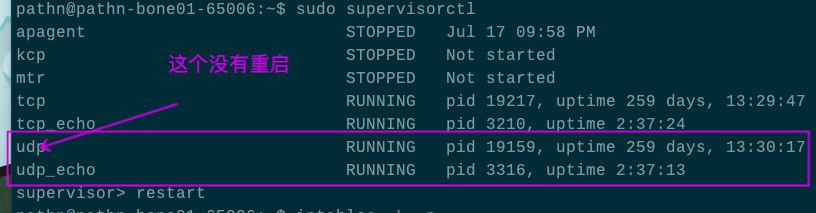
**如果没同步，机器的监控上bone会一直探测旧配置的目的ip和端口，线路异常时也会触发报警**



11点同步配置后，65006-bone02不再探测30016和30028移动这两条线路

**换IP的机器 要 sudo supervisorctl restart udp\_echo**

**sudo supervisorctl restart udp**



疑问：对端65001还是保留旧的tunnel信息，有65006旧的移动ip信息，好像不影响。探测逻辑是怎样呢？----65006重启echo和udp\_echo后，新ip的9999端口会重新监听，因此不影响探测；65001主要依赖sync conf这一步同步到新配置，然后探测新配置里的新ip即可。

至于隧道里旧的ip（相当于起的虚拟网卡），不影响探测，应该属于残留的配置记录，更新下sudo tunnel down eth-tun-47603 && sudo tunnel up eth-tun-47603应该就会不会再显示，真正配置记录在

/home/pathn/conf/bgpd.conf和/home/pathn/conf/tunnel.conf。

机器上探测是基于配置文件 cat /home/pathn/conf/probe\_udp.conf，读取这个配置去探测；

对端的机器监听是通过cat /home/pathn/conf/supervisor/pathn.echo.conf这里去启动；修改ip后需要重启服务，添加新ip的9999端口监听。

**相关问题zenlayer泰国和印尼配置了.250的内网，作tunnel探测开放udp 9999端口**

香港探测节点64577对这两个机器探测有报警

[pathn][2020-07-08 18:03:37][udp\_echo监控-丢包率超出阈值]：

当前共2个，异常2个，新增影响2个

处理：https://g.126.fm/00duufe

影响:

pathn-bone01-64577 | 18.162.196.247 | Hongkong-aws-gw (异常)

- 后端：TO-12017\_Jakarta-zenlayer-public\_47704

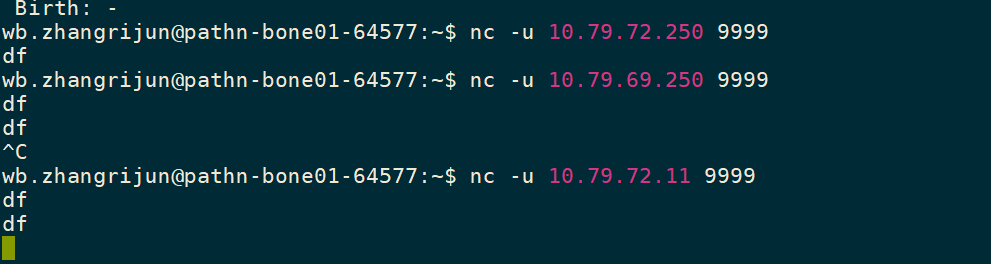
- 当前值：100

pathn-bone01-64577 | 18.162.196.247 | Hongkong-aws-gw (异常)

- 后端：TO-12016\_Bangkok-zenlayer-public\_47703

- 当前值：100

在64577测试12017的udp 9999端口不通



12017查看udp监听netstat -aunlp，.250的内网9999没有监听



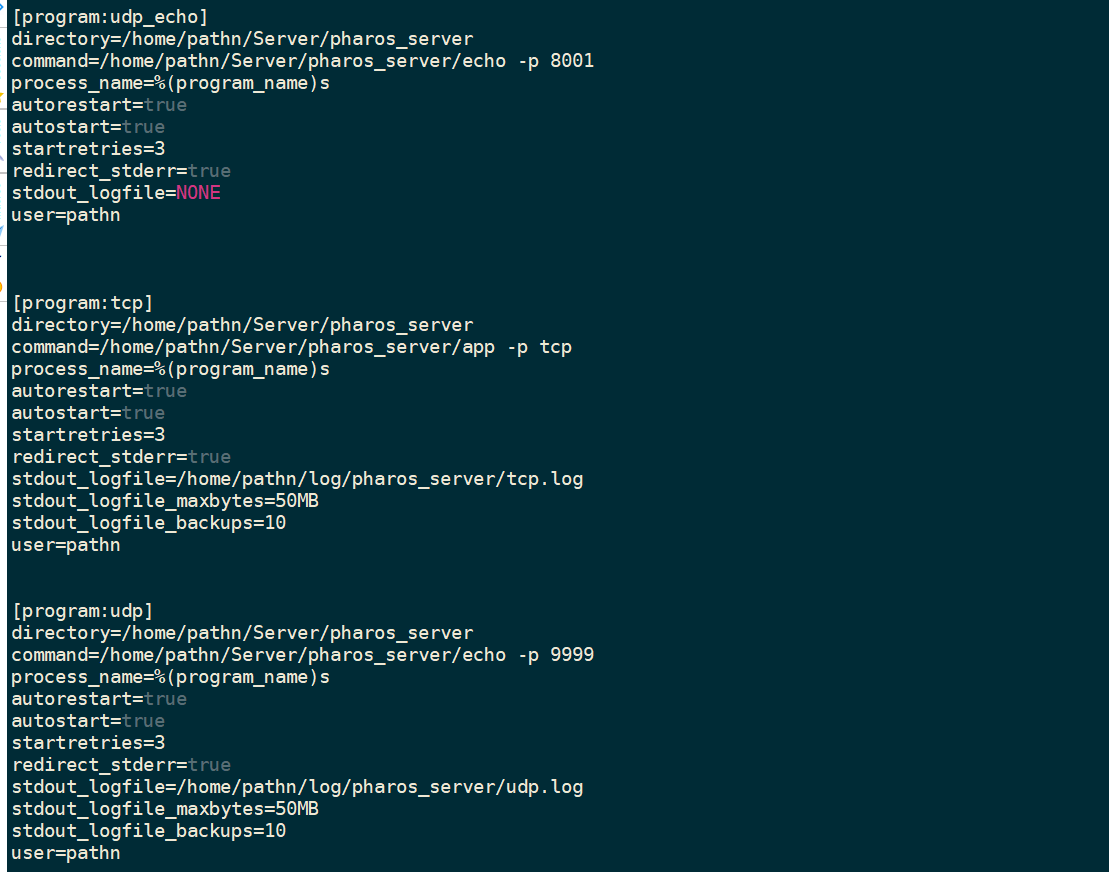
原因是12016和12017的udp udp\_echo没有重启，需要登录到机器重启；

sudo supervisorctl

restart udp

restart udp\_echo

udp和udp\_echo配置cat /home/pathn/conf/supervisor/pathn.echo.conf，指定了9999端口和8001端口----对应路径的echo不支持cat查看，是编译过的so文件；盲猜重启服务时会把本机所有ip都开启监听。

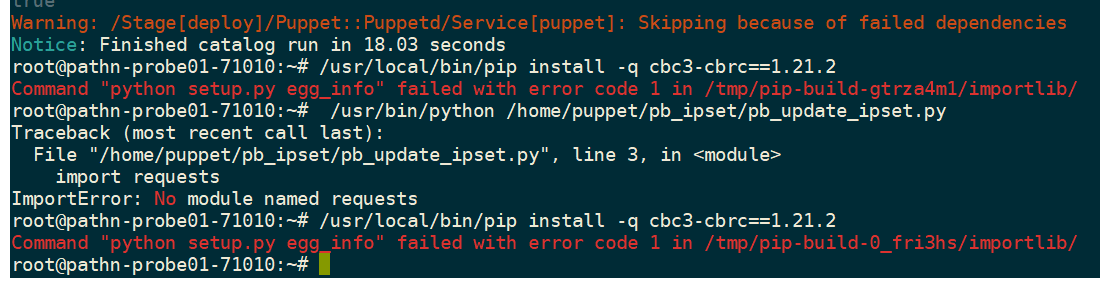


**机器安装跑puppet，需要先把galaxy是ip的region修改正确**

**张家口阿里云探测机器安装puppet**

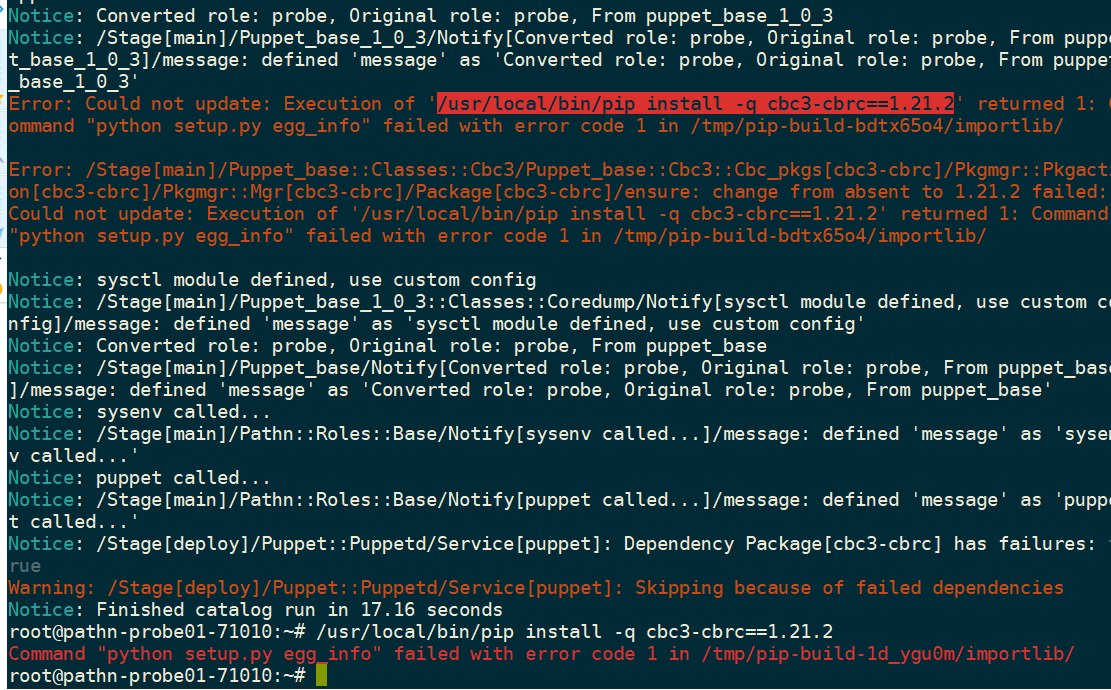
1. ipset问题，需要安装requests库，因为机器升级到python3，需要指定pip2安装；

pip2 install requests



2.cbrc安装失败，pip 版本不对。

ln -sf /usr/bin/pip2 /usr/local/bin/pip



**puppet agent -t报错**

Error: pip install setuptools==34.4.1 returned 1 instead of one of [0]

Error: /Stage[setup]/Puppet\_base\_1\_0\_3::Classes::Packages/Exec[pip update]/returns: change from notrun to 0 failed: pip install setuptools==34.4.1 returned 1 instead of one of [0]

Notice: /Stage[setup]/Puppet\_base\_1\_0\_3::Classes::Packages/Exec[update\_pip]: Dependency Exec[pip update] has failures: true

Warning: /Stage[setup]/Puppet\_base\_1\_0\_3::Classes::Packages/Exec[update\_pip]: Skipping because of failed dependencies

Error: Could not prefetch package provider 'pip': [nil, nil, [#<Puppet::Type::Package::ProviderPip:0x00558914999ba0 @property\_hash={:ensure=>"8.1.1', 'console\_scripts', 'pip')()", :name=>" load\_entry\_point('pip", :provider=>:pip}>], nil, nil, nil, nil, nil]

Error: /Stage[setup]/Puppet\_base\_1\_0\_3::Classes::Packages/Pkgmgr::Pkgaction[pip install]/Pkgmgr::Mgr[glances]/Package[glances]: Could not evaluate: [nil, nil, [#<Puppet::Type::Package::ProviderPip:0x005589149a4190 @property\_hash={:ensure=>"8.1.1', 'console\_scripts', 'pip')()", :name=>" load\_entry\_point('pip", :provider=>:pip}>], nil, nil, nil, nil, nil]

Error: Could not find command 'aptitude'

Error: /Stage[setup]/Puppet\_base\_1\_0\_3::Classes::Packages/Exec[system update]/returns: change from notrun to 0 failed: Could not find command 'aptitude'

单独跑pip install setuptools==34.4.1提示locale.Error: unsupported locale setting，之后export LC\_ALL=C正常。----暂时解决问题，如果业务本身需要使用特定语言包，locale显示语言包，locale -a显示本机当前支持的本地配置。直接设置为C可能会有影响，需要安装对应语言包，例如zh\_CN.UTF-8，安装指令：apt-get install language-pack-zh-hans -y

**马尼拉zenlayer12015 puppet报错**

Error: Could not update: Execution of '/usr/bin/pip install -q zerorpc==0.6.1' returned 1: Command "/usr/bin/python -m pip install --ignore-installed --no-user --prefix /tmp/pip-build-env-WXs7i7 --no-warn-script-location --no-binary :none: --only-binary :none: -i https://pip.nie.netease.com/simple -- "setuptools >= 40.8.0" wheel "Cython >= 3.0a5" "cffi >= 1.12.3 ; platform\_python\_implementation == 'CPython'" "greenlet >= 0.4.16 ; platform\_python\_implementation == 'CPython'"" failed with error code 1 in None

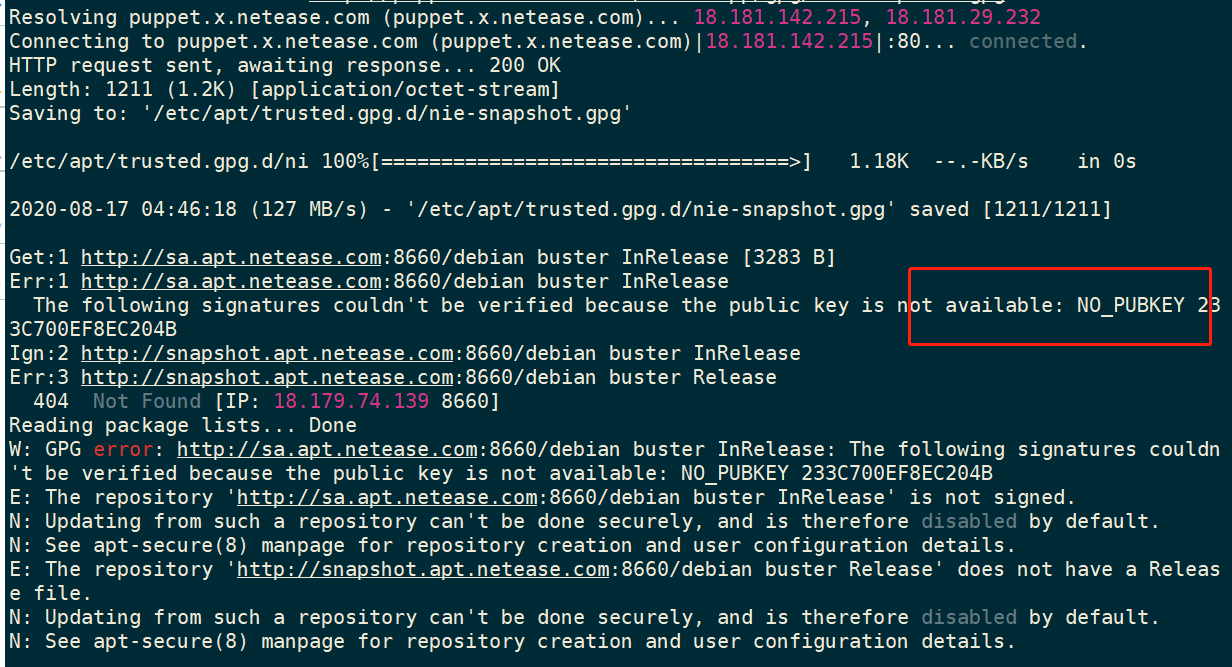
Error: /Stage[main]/Puppet\_base\_1\_0\_3::Params/Puppet\_base\_1\_0\_3::Create\_rs[pkg]/Puppet\_base\_1\_0\_3::Devide[ap\_pkg]/Pkgmgr::Pkgaction[zerorpc]/Pkgmgr::Mgr[zerorpc]/Package[zerorpc]/ensure: change from absent to 0.6.1 failed: Could not update: Execution of '/usr/bin/pip install -q zerorpc==0.6.1' returned 1: Command "/usr/bin/python -m pip install --ignore-installed --no-user --prefix /tmp/pip-build-env-WXs7i7 --no-warn-script-location --no-binary :none: --only-binary :none: -i https://pip.nie.netease.com/simple -- "setuptools >= 40.8.0" wheel "Cython >= 3.0a5" "cffi >= 1.12.3 ; platform\_python\_implementation == 'CPython'" "greenlet >= 0.4.16 ; platform\_python\_implementation == 'CPython'"" failed with error code 1 in None

单独运行这一串命令，会报错“No matching distribution found for Cython>=3.0a5”

搜索相关问题，**有说需要升级pip， pip install pip==9.0.1 -U之后**（之前arm实例安装puppet的奇奇怪怪问题也用这个安装解决一部分问题）puppet agent -t 正常

**印尼12017安装puppet报错**

**The following signatures couldn't be verified because the public key is not available: NO\_PUBKEY 233C700EF8EC204B**



修改/etc/apt/source.list参考同一区域的初始化完成的机器。之后apt-get update，安装puppet的过程很慢，下载包速度非常慢；

下次试下这个：

gpg --keyserver keys.gnupg.net --recv-keys 8C718D3B5072E1F5  
gpg --armor --export 8C718D3B5072E1F5 | apt-key add -

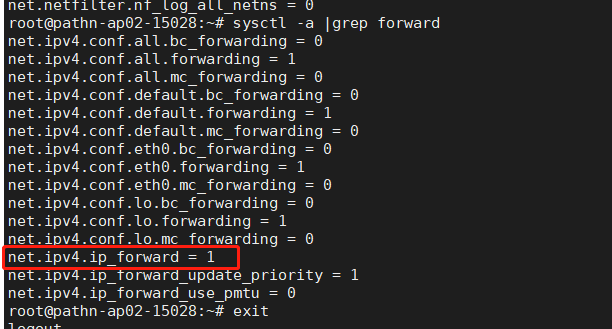
apt-get update

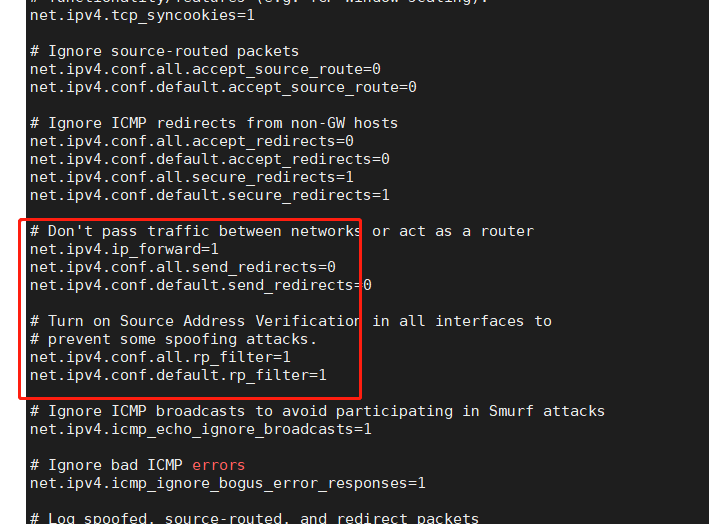
机器没有python，需要安装python 2.7 之后建立软连 ln -s /usr/bin/python2.7 /usr/bin/python

**台湾15028 gcp实例由于宿主机故障发生重启，引发节点探测异常报警**

经查为参数net.ipv4.ip\_forward 的影响，被重置为0。修改为1，通过/etc/sysctl.d/11-gce-network-security.conf固化配置。推测可能是由于gcp旧镜像当时没配置/etc/sysctl.d/11-gce-network-security.conf配置，实例没有进行该参数固化导致重启后出现问题。

多网卡检查 sysctl -a |grep filter；单网卡检查 sysctl -a |grep forward;sysctl -w临时性修改，固化需要写配置文件。



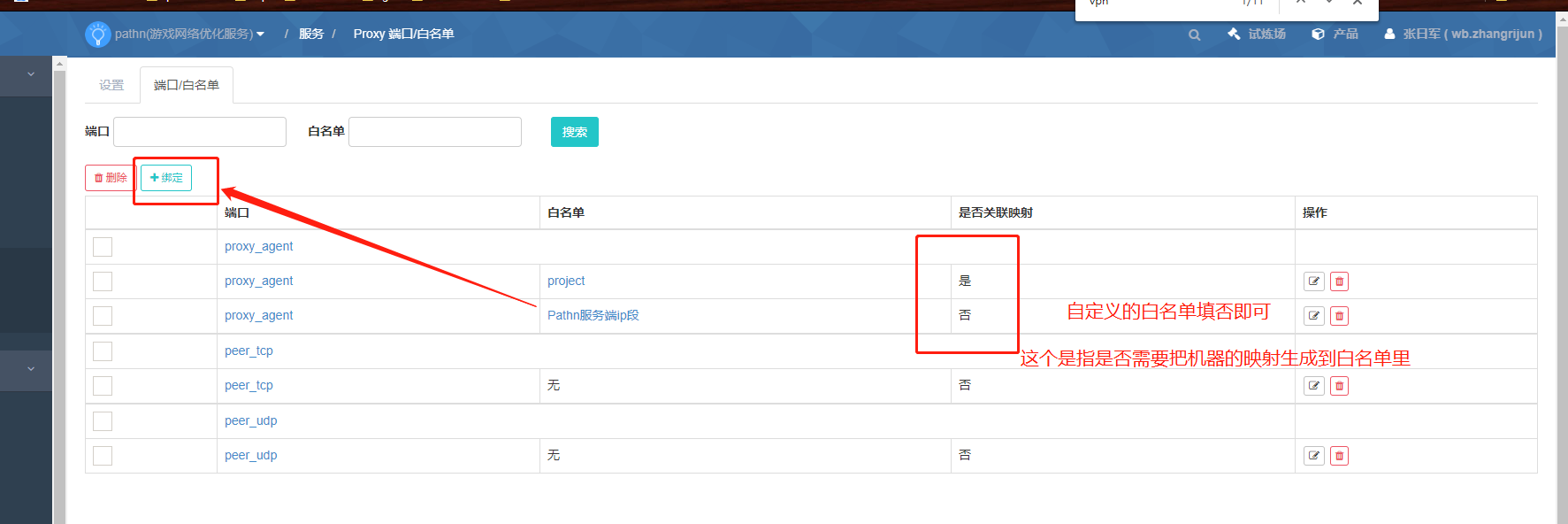


**proxy机器更新agent监听白名单**

所有proxy的agent监听端口添加网段白名单:10.203.5.224/27

1. 先加一个白名单--galaxy-服务-白名单，添加一个自定义的白名单

2. 修改端口白名单--galaxy-服务，进入proxy服务，添加绑定proxy\_agent端口对应的白名单。



之后生成同步配置，如果勾上这个，会自动刷新机器防火墙，不需要cbc操作同步。

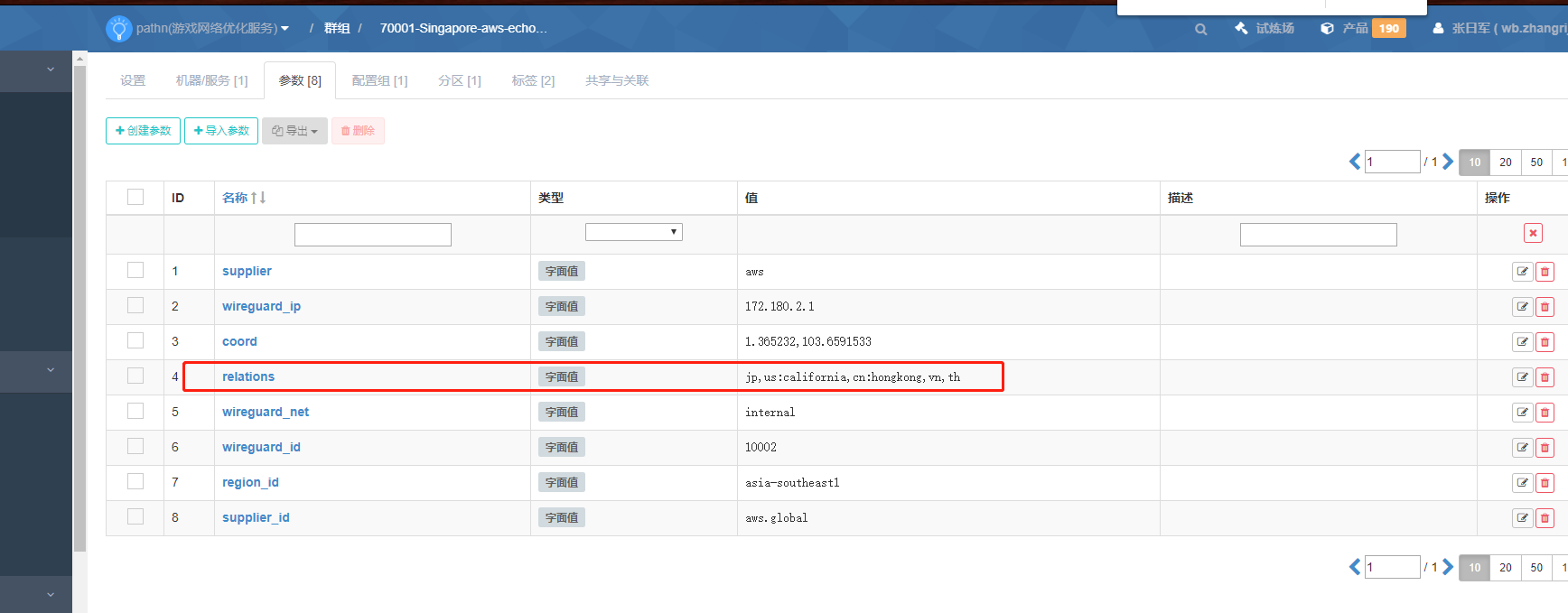


一开始想直接修改project这个白名单，但是这样会影响到关联了这个白名单的其他服务，避免影响其他服务，需要另外添加一个白名单。操作上有点像skyline的安全组。

**探测节点的配置**

/home/pathn/conf/probe\_udp.conf和/home/pathn/conf/probe\_tcp.conf指定了探测目的端ip和端口。由galaxy配置模板/config/{%$host.ip%}.tcp.probe.conf和/config/{%$host.ip%}.udp.probe.conf生成。

例如新加坡70001，群组参数里写好了探测的区域：

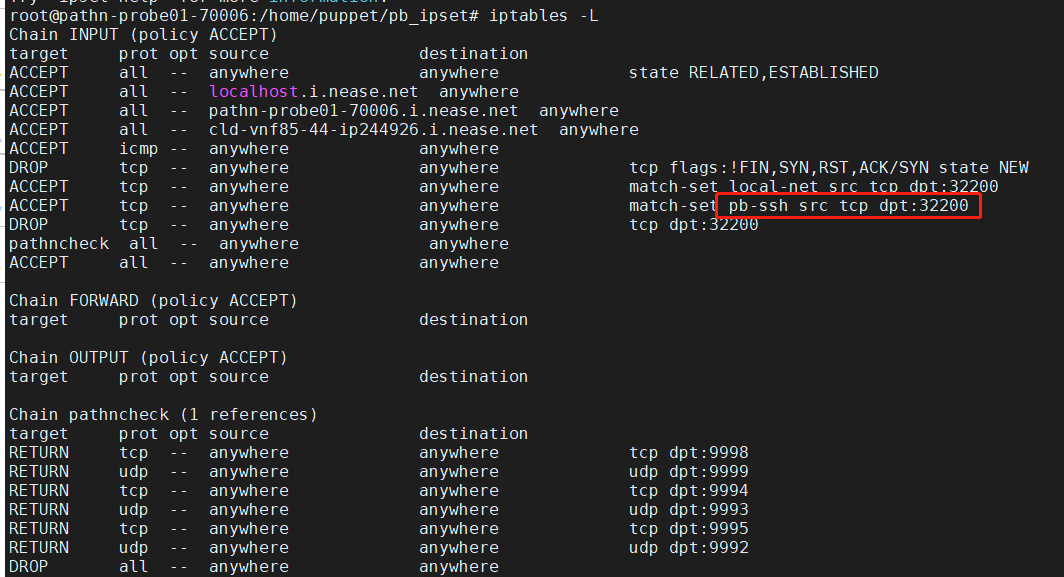


探测节点，一方面可以作为区域互联之间的探测，表示各链路直接的网络状况，也是机房互联网络情况；另一方面作为客户端的探测rap，可以表示边缘节点网络质量，

（写到探测配置文件https://impression.update.netease.com/explore\_jp.txt这些文件里），表示该区域该供应商的网络覆盖情况，玩家到这个供应商入口的网络情况。

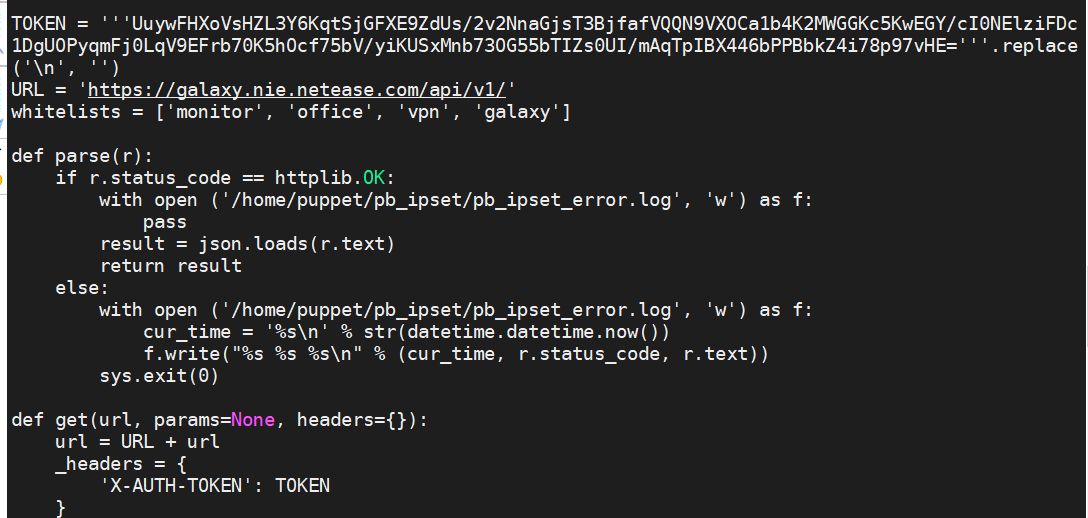
**机器刷新iptables以及ipset说明**

Iptables管理模式为galaxy生成配置，通过puppet或者管理机cbc同步的各群组机器。



galaxy上关于ipset的配置模板主要是common-ipset；pathn的common-ipset作为一个公共ipset可以被其他服务群组调用，比如ap服务，proxy以及probe服务机器实现定制化生成ipset。

机器上local-net一般包括了更多的ip，监控系统的ip是在pb-ssh里面，local-net里面没有，pb-ssh应该是/home/puppet/pb\_ipset/pb\_update\_ipset.py生成，在galaxy拉取对应白名单生成ipset。----这个应该是puppetbase生成的一样ipset，所有项目公用。Local-net则属于pathn自身使用的集合（local-net跟pb-ssh会有重合的ip，因为common-ipset里也会引用到monitor的白名单ip）。



例如monitor白名单ip引用：{%foreach $global.whitelist.monitor as $ip%}

probe机器对应 /ipsets/{%$host.ip%}.ipset配置模板，通过这句去引用common-ipset，

{%~include file={%real\_path name='common-ipset'%} sets='local-net,nie-office'~%}

所有ip都在/home/pathn/conf/ipset目录下的ipset和ipset.pathn两个文件里。

一般跑一下puppet agent -t会同步local-net这个集合,pb-ssh这个一般是在puppet初始化生成，防火墙也会同步更新。

完整的同步刷新防火墙步骤：适用于大批量操作，个别操作直接puppet agent -t也行。

cbc hostlist/list-all sync config

cbc hostlist/list-all iptables open

如果ip有疏漏，可以考虑看下绑定的白名单是否有问题，看下配置模板预览内容是否符合预期。

案例：probe01-70006机器刷新防火墙，需要同步监控region新增ip白名单；一开始上机器看ipset命令显示异常，直接**重装apt remove ipset && apt update && apt install -y ipset**

**之后再cd /home/puppet/pb\_ipset && python pb\_update\_ipset.py**，之后pb-ssh会同步到最新的ip，如果需要local-net同步，应该要跑一下puppet agent -t；

另一台机器10.82.101.22，开始机器上pb-ssh和local-net都没有监控的新ip，直接跑了一次puppet后local-net同步到新ip，pb-ssh也会同步。----比较奇怪，15030这些机器，pb-ssh里面有新ip，但local-net没有，单独跑过python pb\_update\_ipset.py？

--所有新的机器，比如71030/71031，puppet跑过，阿拉丁初始化也跑过，机器上的pb-ssh和local-net都有了新监控ip。

**puppet流程**

当前嬴政系统使用的是 master agent 架构，该架构工作流程简述如下：

1. 所有的 puppet 代码只存在 master 上，agent 上无 puppet 代码
2. agent 将自身的系统状况（facts）上传给 puppet.conf 指定的 master 并请求 catalog
3. master 收到 agent 数据后开始编译 puppet 代码并生成 catalog 传回 agent
4. catalog 是一个表示系统最终状态的文件，agent 收到 catalog 后 apply 对机器生效

**galaxy触发/同步--生成配置文件，生效到机器还需要puppet和cbc**

防火墙、报警、线上配置、监控、DNS需要手动同步，没有自动更新的机制--cbc去推送配置

puppet自动同步，每半个钟运行一下（或者手动puppet agent -t）--puppet去拉配置

Puppet管理了哪些配置----大部分在common模块里，加上项目自定义的模块。哪些配置要cbc，哪些要puppet，要捋一捋。**例如防火墙，直接手动puppet agent -t，是可以刷新防火墙，但是这样只能一台一台操作，或者等半个钟puppet定期跑一遍，利用cbc则可以批量的推送更新，对大批量操作更方便。**

**管理机-pathn服务端**

管理机是1001，代号admin，白名单开放是办公区和savpn

服务端是2001，代号pathn，存放了配置数据，rsyncd端口白名单开放对应的galaxy，cache，gitlab等，serverlist端口开放savpn和办公区。initsite8660端口和vxlan9990端口开放白名单是project

**nft\_access节点白名单控制**

一般节点全开，不做限制，个别节点使用白名单，如15020上a92，只对办公网和savpn开放。

 "nft\_access": [

            "office",

            "vpn"

        ]

通过postman patch {{pathn}}/vgroups/a92/agent\_config来修改。

之后节点nft里面会有对应ip集合：



**下线探测节点触发了一堆报警**

[处理][pathn][2020-05-25 10:49:12][udp\_echo监控-丢包率超出阈值]：

当前共1个，异常1个，新增影响1个

处理：https://g.126.fm/01vD6Q8

影响:

pathn-probe01-70025 | 3.9.254.202 | London-aws-echo (异常)

- 后端：TO-eu-frankfurt-gcp-bgp

- 当前值：100

Galaxy生成配置和同步后，需要**cbc同步配置cbc hostlist/all-70016-Frankfurt-aws-echo sync config**，之后机器上的/home/pathn/conf/probe\_udp.conf和probe\_tcp.conf才会更新。

同一个region，比如tokyo-gcp探测节点有两个，新加坡70001是如何确定去探测哪一个？

----一开始70001的探测配置里是日本gcp对应的是71020的ip，操作71020下线后，同步70001的配置，其日本gcp对应ip变成了70005，另一个日本的gcp。

**galaxy添加标签alert\_disable，同步后可以屏蔽机器报警**

[pathn][异常][2020-06-03 09:46:25][pathn服务端请求延迟超出阈值]

当前共1个，异常1个，新增影响1个

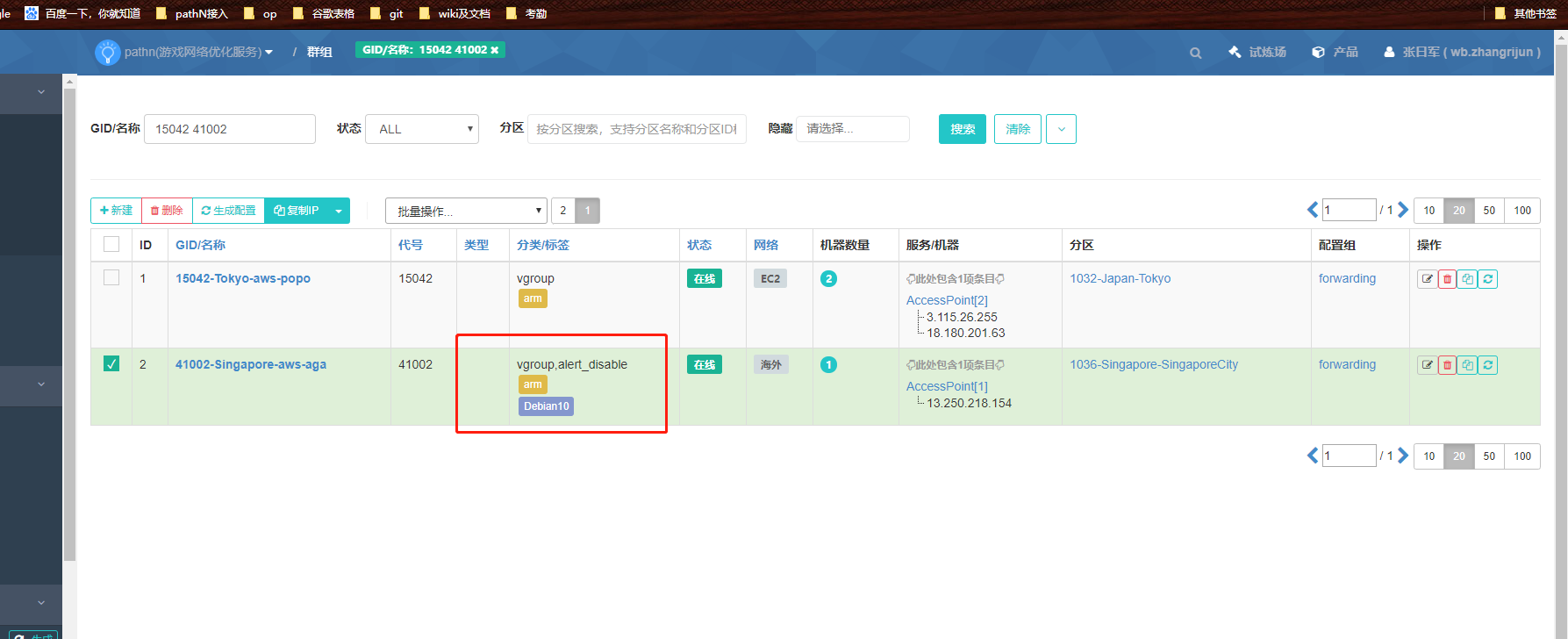
当前值:

pathn-ap03-15042 | 13.250.218.154 | Singapore-aws-aga: 10K (异常) ##机器名跟ip没对上，实际机器名为pathn-ap01-41002

如未及时处理，请联系值班:

姓名: 张日军 手机号: 17608990421

处理: https://g.126.fm/03FtNQI



**东南亚ucloud故障处理预案**

1. 回杭州公共集群c1，如果节点问题，将香港节点覆盖其入口，刷新列表；

修改集群cluster的region\_configs， PATCH {{pathn}}/clusters/c6， 例如将测试集群c6台湾列表替换为香港入口ip。

{

"region\_configs": [

{

"ap\_regions": ["hongkong"], ##可以指定该区域使用哪些区域的AP生成列表

"enable": true,

"isps": [],

"region": "taiwan", ###台湾区域

"scopes": [],

"svrlists": []

}

]

}

中港入口有问题可以考虑经新加坡朗桥中转，如11014-ap01后端除了香港中转，也有新加坡中转a246（目前下线维护状态），越南和印尼的11015和11016目前节点后端没有指向新加坡a246。但是中港入口有问题时，朗桥新加坡到香港入口会不会也有问题？需要确认后再调整。

1. 到新加坡c5公共集群，g78na的c14集群，下线ucloud节点，刷新列表；

@pathnctl vgroup <ID> -s 3

@pathnctl sync serverlist -c/--cluster <cluster\_id>

或者单独刷新项目列表,@pathnctl sync serverlist -p/--project <project\_code>

**Pathn外部数据源同步**

SRE|李健华 12:03:39

path的外部数据项， 群组或机器有变化的时候每次都得手动去点生成吗？

SRE|李健华 12:03:48

还是生成全部就会自动跟着生成？

CDN|郑康佳|konia|58623 12:04:58

SA管理的那个数据项 会随着 `群组或机器有变化` 而变化的话,

在galaxy生成任意文件都会提交到我们这边开通.

CDN|郑康佳|konia|58623 12:05:06

不需要人肉去刷新

SRE|李健华 12:05:47

SA管理的那个数据项 会随着 `群组或机器有变化` 而变化的话,

嗯， 这个可以保证

CDN|郑康佳|konia|58623 12:07:19

**那只要galaxy 有生成操作 都会同步数据. 问题不大**

**[pathn][2020-07-06 15:02:34][服务端warning]：**

- 环境：production

- 报警模块：instance:10.203.5.250,process:api

- 调用者：wb.zhangrijun ( 10.200.112.27 )

- 接口：POST /api/v3/portmaps/self\_trigger

- trace\_id：5e2b31559a5446cca0df310247e21643

- 消息：g67ena 在以下集群在某些列表中缺少映射IP: c23(taiwan)

- 详情：<https://g.126.fm/02KF7XO>

这是因为c23集群region\_configs里面配置了taiwan入口，但是没有提供台湾ap；

一了百了，可以更新c23，去掉taiwan 这个region。

**Puppet异常**

Could not retrieve local facts: invalid byte sequence in UTF-8 #2. [pathn-dev04-1102.i.nease.net] Failed to apply catalog: Could not retrieve local facts: invalid byte sequence in UTF-8



修复命令：systemctl set-environment LANG=zh\_CN.UTF-8 && systemctl restart puppet

Puppet指引文档https://g.126.fm/00dxyDe

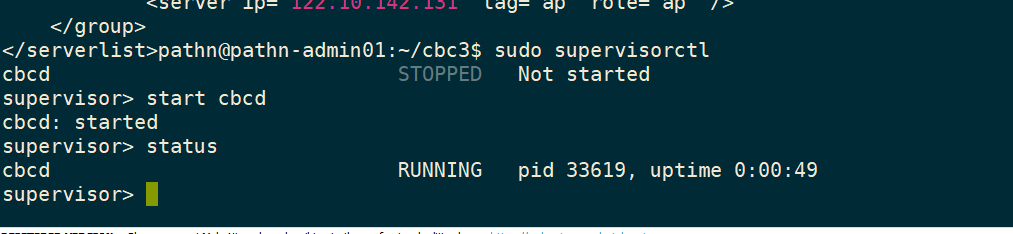
**应用配置失败：**pathn-bone01-65002 42.186.79.249

修改编码，观察后续邮件是否还报警

**12018越南zenlayer在阿拉丁不显示**

操作流程里无法显示12018对象

排查为管理机cbcd没起来，应该是前几天重启机器后进程没起来。



**[12018-HoChiMinh-zenlayer-public](https://galaxy.nie.netease.com/_pathn/galaxy/groups/91176/services)** [pathn服务端请求延迟超出阈值](https://blackhole.nie.netease.com/_pathn/issue/detail?issue_id=5f1ad20e4b378d936979fc42)报警

ap01-122.10.142.130 机器配置了隧道内网，也作为ap02的内网下一跳网关.250



ping 10.204.2.227不通，ping10.77.40.120和10.10.77.21（隧道对端ip）正常

sudo nft list table ip nat

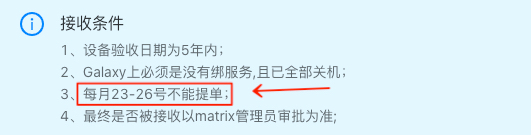
sudo tunnel up

之后ap01这台可以通内网；

但是ap02这台一直不通，ap01抓包看ping只有request，没有reply，telnet 80端口也是没有响应。----**需要在香港aws的gw机器上添加静态路由**

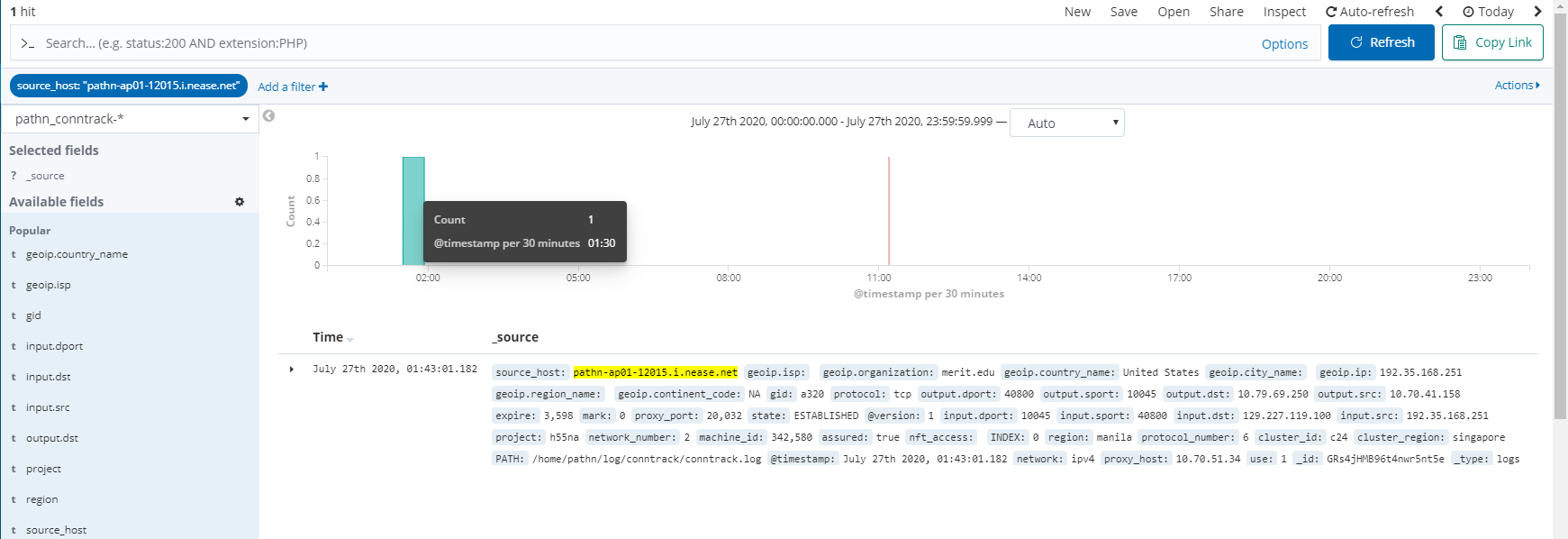
**下周任务：27号**

@op pathn 在matrix系统归还备机 42.186.57.59 @SRE|张日军

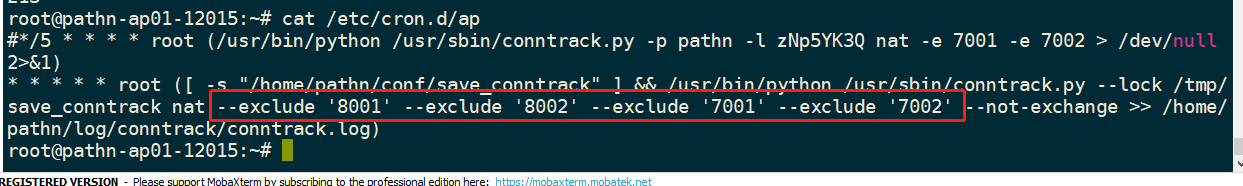


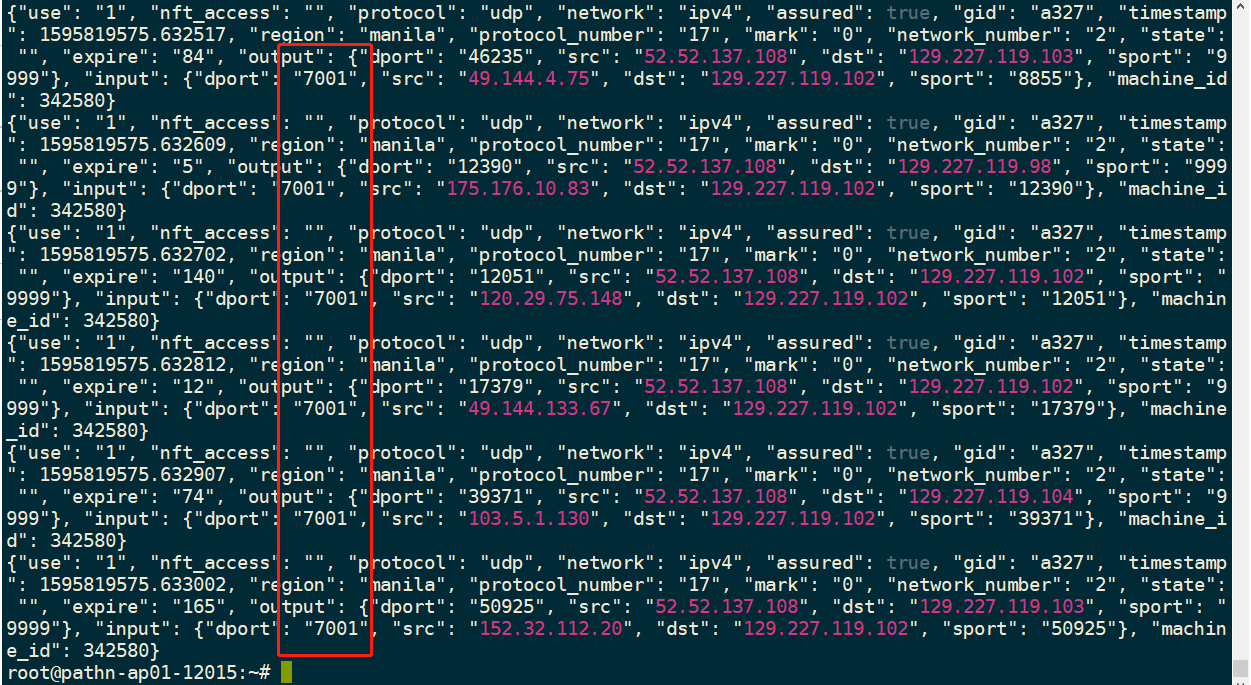
**pathn-ap01-12015.i.nease.net**

elk上面conntrack日志只有一个记录，对应机器上/home/pathn/log/conntrack/conntrack.log也是一个记录；



conntrack.py nat |wc -l显示有两百多记录，未记录到日志是应该属于探测端口7001，被排除了





**机器需要下线关机更换内存，也需要确认这部分连接，为g96的探测，指定了机器上a327-129.227.119.102，修改配置改为另一台机器的a328推送配置迁移流量。**

CDN|郑康佳|konia|58623 11:04:06

我查一下网络探测哈

CDN|郑康佳|konia|58623 11:05:44

{% $IMPRESSION\_ECHO["\_all"]["test\_g96\_ph"] = [

"ip" => "129.227.119.102", "udp\_echo" => "129.227.119.102:7001", "tcp\_echo" => "",

"ips" => ["129.227.119.102"], "udp\_echoes" => ["129.227.119.102:7001"], "tcp\_echoes" => []

]

%}

这部分的内容应该是impression下数据项内容，这个数据项在哪里被引用到，有待确认

<https://galaxy.nie.netease.com/_impression/galaxy/dataentrys/6407/content>

**探测目的ip汇总，探测节点都在里面**

<https://pathn.update.netease.com/echo.impression.txt>

**[pathn][elk\_syslog告警][2020-07-15 12:54:26]**

异常主机:116.93.17.135 pathn-probe01-70032

源日志: [513100.356653] print\_req\_error: I/O error, dev fd0, sector 0

详情页: https://g.126.fm/04gXabe

这个报警，看了网上有个比较相似的

https://askubuntu.com/questions/719058/blk-update-request-i-o-error-dev-fd0-sector-0/884026

实测有效

sudo rmmod floppy

echo "blacklist floppy" | sudo tee /etc/modprobe.d/blacklist-floppy.conf

sudo dpkg-reconfigure initramfs-tools

**70021探测节点替换新实例**

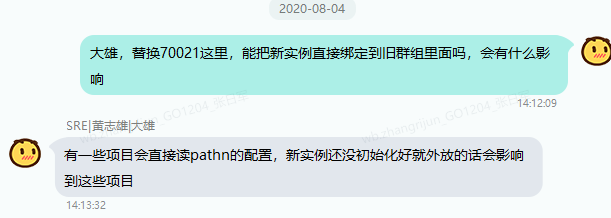
1. 创建一个临时群组（复制70021群组）

2. 创建一个c5实例，放到临时群组中，然后进行初始化

3. 初始化完成后，把70021的旧实例从群组中移除，并把新实例移到70021

4. 重新生成配置并推送（pathn项目和impression项目都要重新生成）

5. 旧实例下线





galaxy直接复制群组

**G92替换aga触发c16生成映射无数据返回**

因为项目没有有效的映射，失效的映射是不会在接口返回的，但是会出现在列表中

SRE|张日军 17:30:11

aga的节点不会作为映射返回接口吗

SRE|黄志雄|大雄 17:36:03

映射是分配在vip上的，然后vip再解析成aga或ap的ip，如果连有效映射都没有，那当然就不会返回了

**[升级][pathn][异常][2020-08-07 11:42:10][pathn服务端请求延迟超出阈值]**

当前共1个，异常1个

当前值:

pathn-ap02-12005 | 180.163.62.19 | Shanghai-zenlayer-public: 10K (异常)

8月7号上海电信故障，内网隧道不通，访问服务端延迟过高

处理：galaxy群组添加island标签后同步配置；

或者直接vi conf/apagent.conf，改为外网"api": "https:\/\/api-pathn.nie.netease.com\/api\/v3"

**udp\_echo监控-完全丢包，台湾机器下线到日本节点后，仍然发出到日本后端的报警**

[处理][pathn][2020-08-13 10:26:37][udp\_echo监控-完全丢包]：

当前共4个，异常4个，新增影响4个

影响:

pathn-ap01-15004 | 103.105.134.111 | Taiwan-syscloud-h55na (异常)

- 后端：tokyo-p7-in

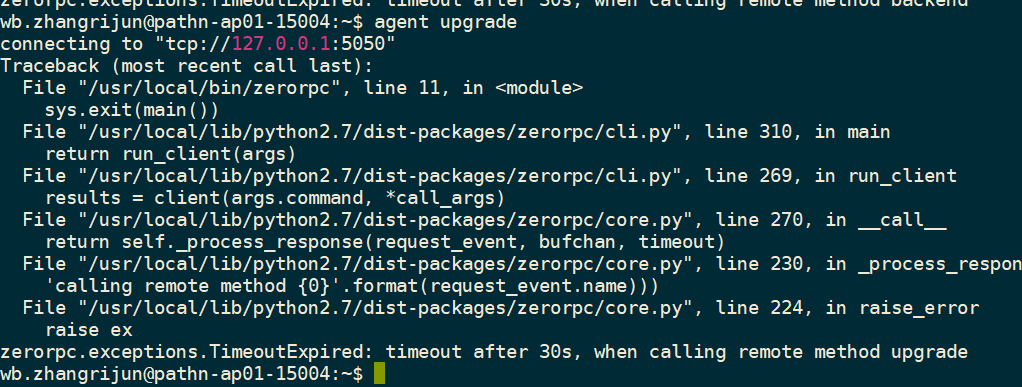
- 当前值：100

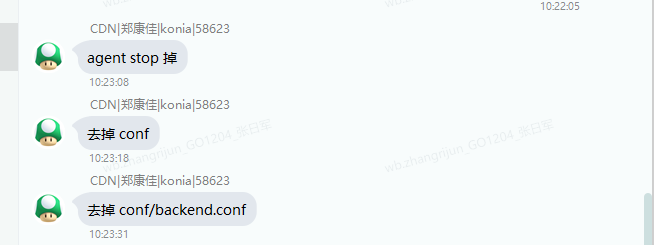
pathn-ap01-15004 | 103.105.134.111 | Taiwan-syscloud-h55na (异常)

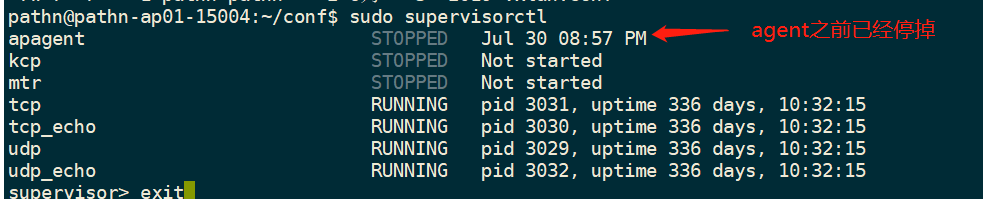
- 后端：tokyo-p5-in

- 当前值：100

机器上agent upgrade会显示超时，cat /home/pathn/conf/backend.conf里还是保留节点后端信息。







再需要删除backend.conf即可

**udp\_echo监控-丢包率超出阈值----probe机器互联，部分机器下线导致报警**

[pathn][2020-08-14 20:41:49][udp\_echo监控-丢包率超出阈值]：

当前共1个，异常1个，新增影响1个

影响:

pathn-bone01-64577 | 18.162.196.247 | Hongkong-aws-gw (异常)

- 后端：TO-12017\_Jakarta-zenlayer-public\_47704

- 当前值：100

处理：<https://g.126.fm/00Ub5R6>

需要修改64577机器上/home/pathn/conf/probe\_udp.conf，将对应12017的探测配置去掉，这个操作应该跟galaxy下线测试节点后，生成同步配置，再通过管理机cbc sync conf对报警的机器更新配置一样。

**备用资源上线流程**

机器开机

同步proxy映射 ##@pathnctl sync proxy p110

proxy节点状态设为1 ##@pathnctl vgroup p110 -s 1

等待几分钟后查看AGA后端是否已经更新 ##postman查看aga后端情况

**节点下线机器回收**

1. 下线前确认节点上的conntrack记录，在节点完全没有conntrack记录的情况下再进行回收；根据剩余conntrack记录的情况，跟项目sa确认旧节点的使用情况。

2. 旧节点机器回收前，保留节点ip，不直接删除；先刷空机器上nftables规则，观察使用情况，如果有项目异常可以及时回滚**；**

**[pathn][异常][2020-08-31 18:31:01][CPU空闲使用率]**

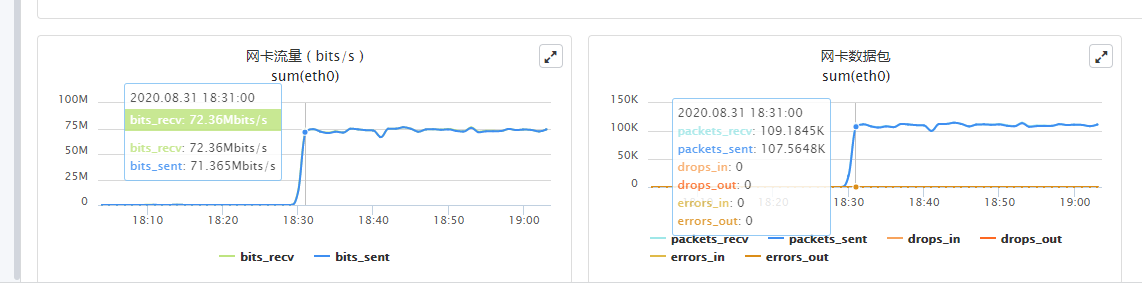
**当前共4个，异常4个，新增影响4个**

**当前值:**

**pathn-proxy02-11006.0 | 3.101.4.104 | California-aws-public: 1.7241 (异常)**

**pathn-proxy01-11006.1 | 52.52.137.108 |**

美国proxy11006报警，负载过高，6点半后有大流量----**流量平滑上升考虑业务流量，暴涨考虑被攻击**



本来想问项目有无调整上流量，项目否定

机器上iftop -n 有异常ip 72.21.81.85（非aga ip）

conntrack -L | grep 72.21.81.85 确认全是7001 echo\_live服务监听的端口，应该是被扫描打流量；

ss -lnp|grep 7001 ##查看7001是什么服务监听端口

conntrack -L | grep 7001 |grep -v 72.21.81.85确认没有其他ip连接这个端口；通过关闭echo\_live服务临时关闭

Supervisorctl echo\_live stop

大雄手动加iptables规则封堵 72.21.81.85，之后把echo\_live开回去。

iptables -I INPUT -s 72.21.81.85 -j DROP

**G37切换aga**

au，us，jp三个区域切换aga列表，eu保留；

切换后，美国玩家连aga，优先选择了欧洲aws节点作为后端，所以15017群组在切换后流量上升，特别是ap02这台，分摊一部分的流量（之前g37欧洲列表只用ap01上的节点）；

**代理游戏接入Pathn**

SRE|梁从龙|24022 17:50:21

有代理游戏接入过Pathn吗

SRE|梁从龙|24022 17:51:07

pathn是否要提供SDK给合作方的研发人员使用呢

SRE|梁从龙|24022 17:51:21

想大致了解下这个信息。

合作方游戏我们有对接. maxxx 类的项目

ma83,ma96,ma102na

**skyline上面网络延迟数据说明**

<https://skyline.nie.netease.com/clouds/monitor>

probedata查的是玩家地区到机房的网络数据，skyline上的这个是机房到机房的网络数据，不一样的

skyline显示的是机房互联数据，对应小蜜@pathn net singapore virginia ##源区域，目的区域，对应http://netmonitor.nie.netease.com/business/btopo/205这个配置

probedata则是@pathn probedata "United States" singapore ##玩家区域，目的区域

**gw机器隧道操作----观脉65011 GW调整**

**GW调整**

galaxy设置群组类型为 其他-故障群组, 并生成配置推送, 使用cbc同步gw配置

cbc hostlist/list-bone sync config

断开隧道

sudo tunnel down eth-tun-47603

**回滚操作**

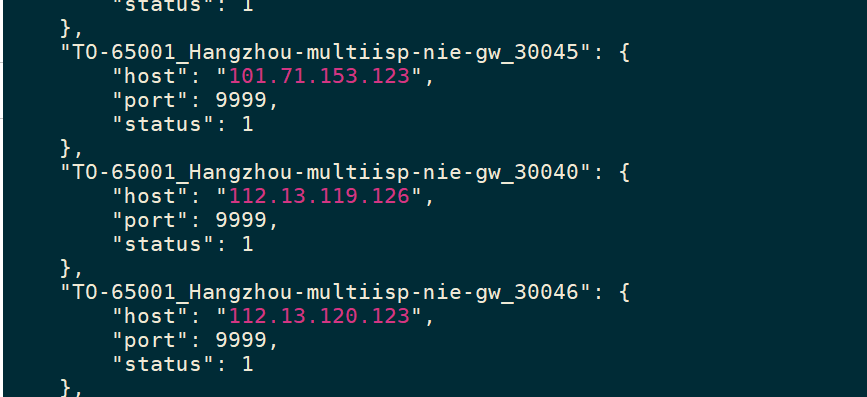
galaxy移除群组类型为 其他-故障群组, 并生成配置推送, 使用cbc同步gw配置

cbc hostlist/list-bone sync config

恢复隧道

sudo tunnel up

张日军 14:51:44



像观脉网关这个，到杭州有三线，假如是其中一条有问题，应该可以down掉问题线路吧

Net(郑康佳) | CDN |58623 14:55:02

这个阔以

Net(郑康佳) | CDN |58623 14:55:14

但如果全down 可能就不好了..

Net(郑康佳) | CDN |58623 14:55:31

另外如果要屏蔽的话 把 status改成 3 就不会有报警了

**日升欧洲被攻击断线，切到中港**

嗯. 但具体群组改一下

群组是 65005-Beijing-nie-gw 设置为离线

然后去 -64575 down 一下 sudo tunnel down eth-tun-47500

**9月22经日本自建和zenlayer去往美国aws丢包**

张日军 16:56:34

美国两个gw 64566和64568，都有跟中港65003内网互联线路；

切换的话是直接在这两台机器上关闭zenlayer和自建对应隧道吗

Net(郑康佳) | CDN |58623 16:57:05

嗯..理论上是

Net(郑康佳) | CDN |58623 16:57:13

但切换有风险, 不确定会不会彻底断了..

Net(郑康佳) | CDN |58623 16:57:50

有些时候 可能 内网互联没有学到路由.

Net(郑康佳) | CDN |58623 16:57:54

直接切就全挂了.

**东冠国际出口机器上架HIH-S-93070 HIH-S-93071**

(需要添加Intel X520 10Gb 2-port网卡\*2 ，搬迁至国际出口机柜)

1、提交【硬件变更工单】，扩容网卡。

2、提交【网络机架调整工单】，通知机房帮你搬迁到你要的机柜

3、如果需要更换IP，还需要galaxy修改对应的IP

09-18机房反馈网卡缺，等调拨，下周到

张日军 11:20:57

大佬请教个问题，后续机器会有6个万兆网卡，两两做bond，一个接外网，一个内网，一个接国际日本；

想问下机房接线是按照我们做好的bond去接线，还是机房接好线我们再做bond？@李林

李林 11:21:38

都是可以的，你配置好IP，告诉机房那些端口配置了那些IP，机房就会知道如何接线的。

李林 11:22:04

eth0+eth1=bonding0——外网

这样的信息给到机房就好

目前机器eth0和eth2是两个千兆卡，bond0接外网；-----从新增的万兆卡拿两个做新的bond0

eth1和eth3两个万兆卡，bond1接内网；

剩下的万兆卡做bond2，接日本出口；

**12014 unreplied\_conntrack取源ip数据**

报警机器上/home/pathn/log/conntrack\_status\_log目录

cat unreplied\_20200924\_171544.log|awk -F "=" '{print $2}' |awk -F " " '{print $1}'|sort | uniq -c | sort -nr | head -30

取次数前30的ip

sort 是默认以第一个数据来排序，而且默认是以字符串形式来排序,所以由字母 a 开始升序排序

uniq命令可以去除排序过的文件中的重复行，因此uniq经常和sort合用。也就是说，为了使uniq起作用，所有的重复行必须是相邻的

**11012中经云机房机器没有在线节点，流量却比较高，为探测流量**

一开始按ip查galaxy，只显示属于11012，没看到70039，需要通过机器里通过ip去定位到这个虚拟机，绑定了ap和echo服务

机器上tcpdump看大部分都是9999端口

<https://pathn.update.netease.com/echo.impression.txt>机器ip出现在列表里，关联配置模板为echo.impression.txt

<https://galaxy.nie.netease.com/_pathn/galaxy/configtpls/54662/params>

模板里用到的数据项为ECHO

<https://galaxy.nie.netease.com/_pathn/galaxy/dataentrys/10005/content>

**C13集群artct广州集群**

项目服务端218.107.55.11在高塘，需要开放给海外用户访问，所以使用hongkong加速入口，节点a92，同时也配置了杭州的节点a103，开放给国内访问

http://pathn-in.x.netease.com:8892/lighten/artct/hongkong.txt

**C17集群ma83杭州.大陆加速集群**

开通了广州和北京的入口，北京的入口使用的是杭州11001的proxy作为入口（p60-42.186.79.135），这个得研究下是哪里指定的----可能是对应region没有提供ap的话，会去proxy作为入口？

<http://pathn-in.x.netease.com:8892/lighten/ma83/beijing.txt>

集群配置允许proxy作为入口，"enable\_proxy\_entry": **true**,

正是这样，文档有说明https://g.126.fm/03HukHS

