**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。**

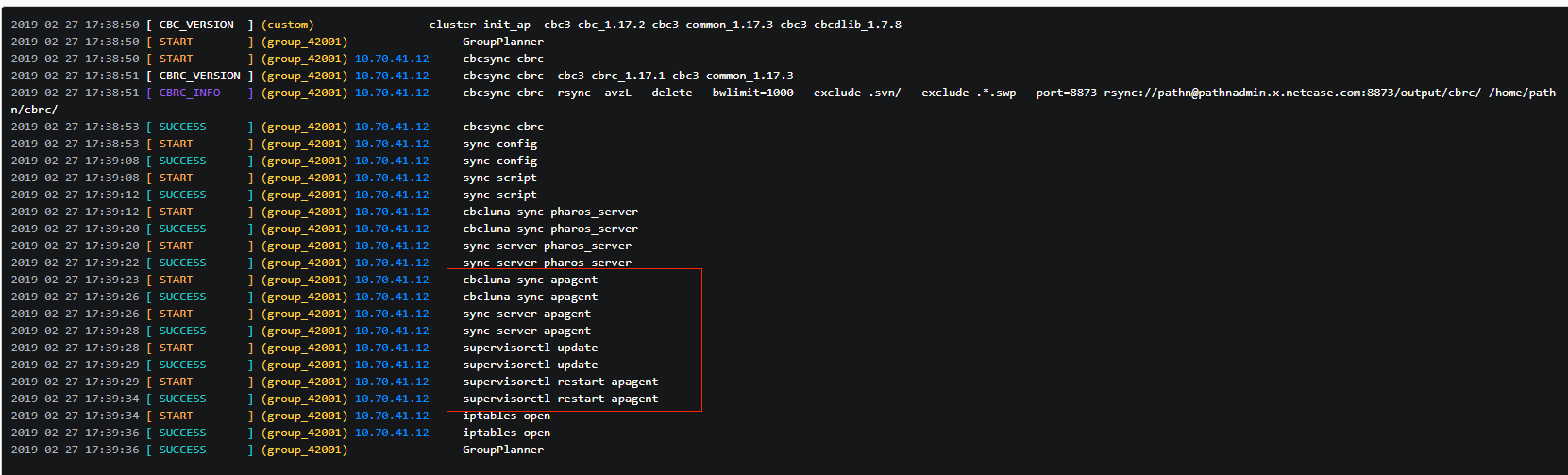
**群组缩容/故障机器下线：修改机器状态0下载，如果是ap，需要刷新对应列表**{{v3}}/portmaps/self\_trigger**指定proxy或者项目刷新。**

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。

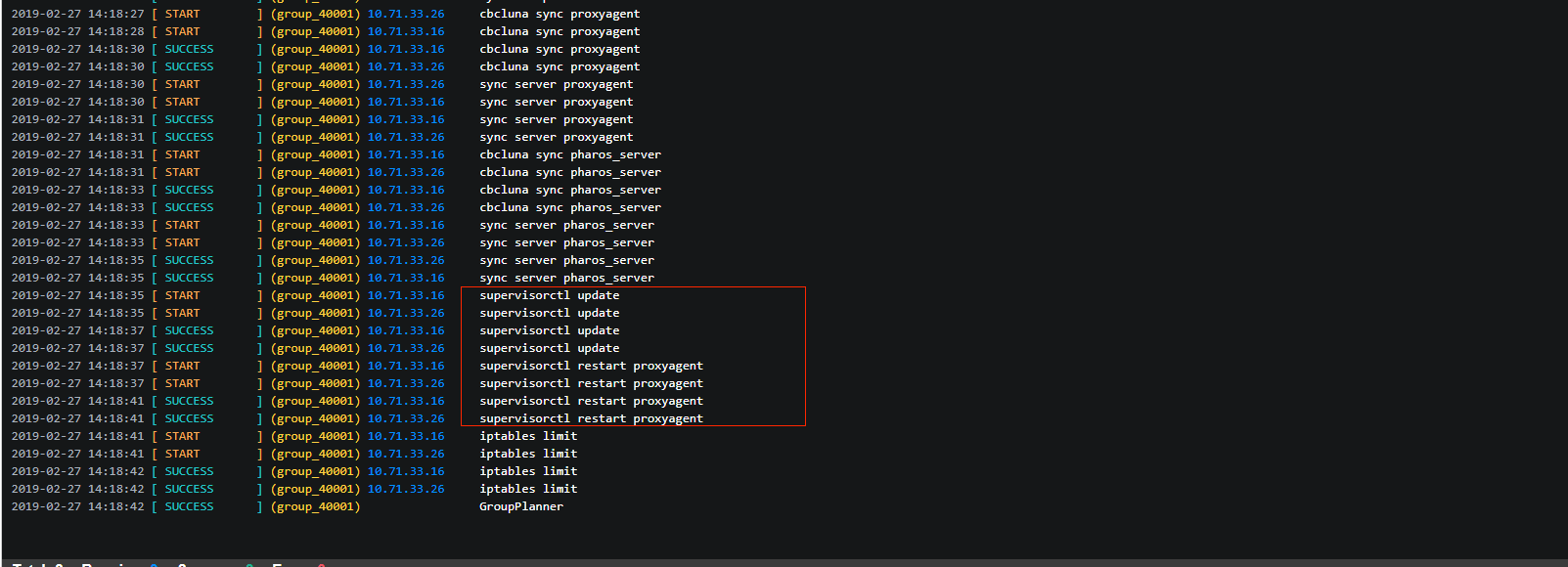
对于旧集群, 新上线proxy机器, 需要使用 sync 接口强制同步历史数据

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:



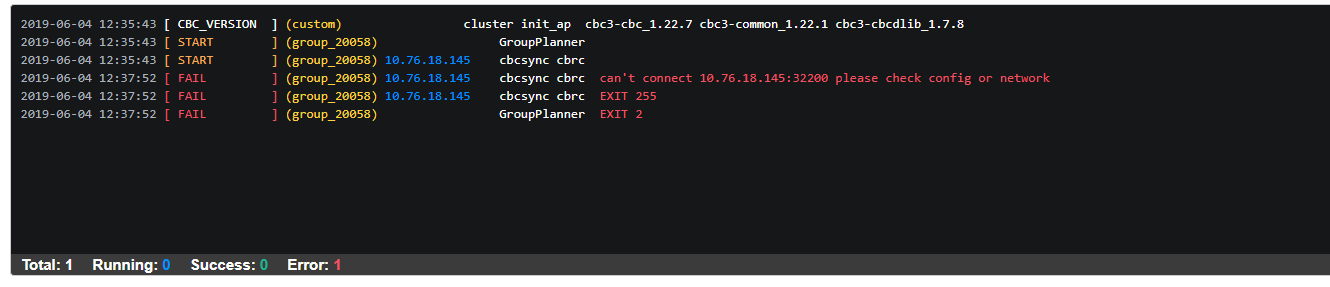
初始化proxy集群，同步代码，重启proxyagent：



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）。

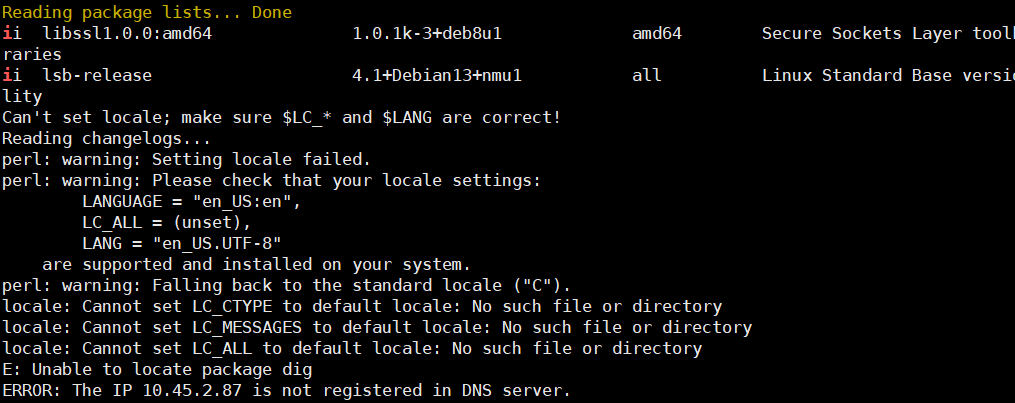
问题二：在旧的ap群组下创建实例，galaxy上显示状态为离线，阿拉丁上找不到对应的群组。需要修改galaxy状态为在线，之后可以在阿拉丁操作ap初始化。（昨天有个操作失误，在下线ap42001群组的时候，设置了ap离线状态，之后下线机器，回收实例，没有操作删除galaxy上面ap群组（ap42001群组还在，但是是离线状态，没有绑定机器）；后面再操作上线流程，通过api群组创建，用的还是旧的ap群组名称和参数；之后galaxy上42001显示有新的机器；但是阿拉丁上找不到这个42001群组）为避免这个情况，需要按照下线流程把ap删除，URL：DELETE /aps/{gid}。

**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）

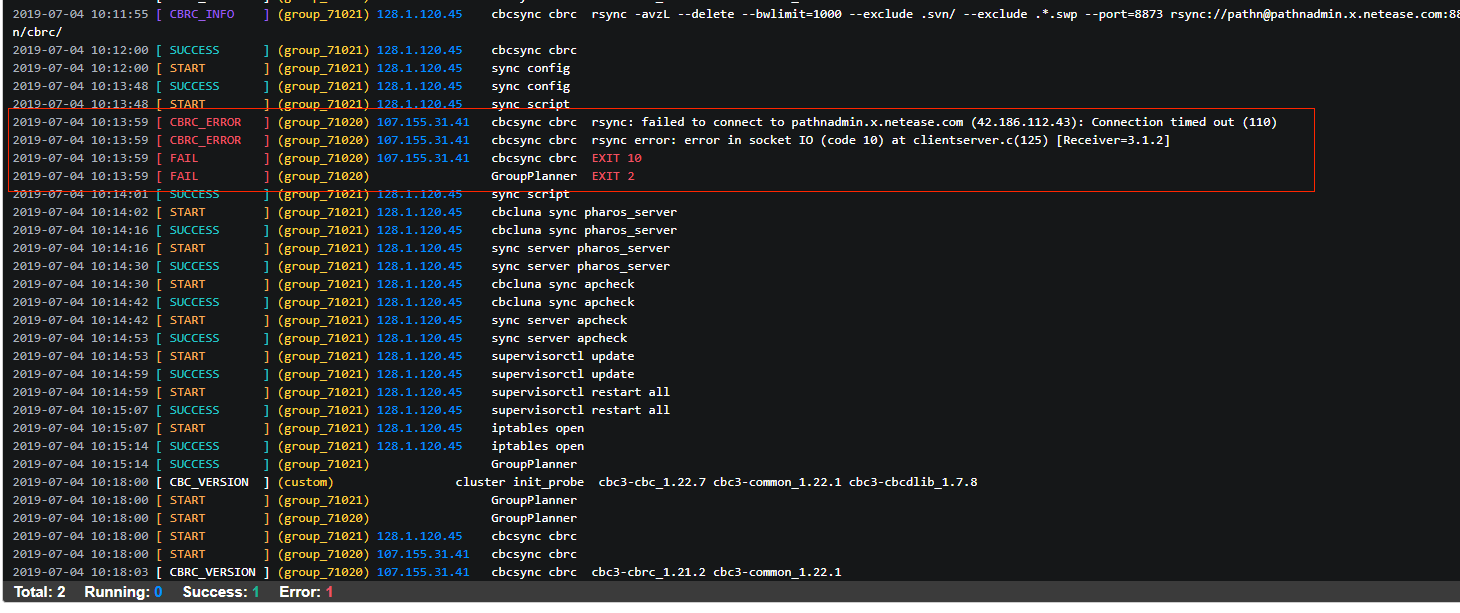
或者在阿拉丁直接走“初始化网络测试节点”流程。（33009)

 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

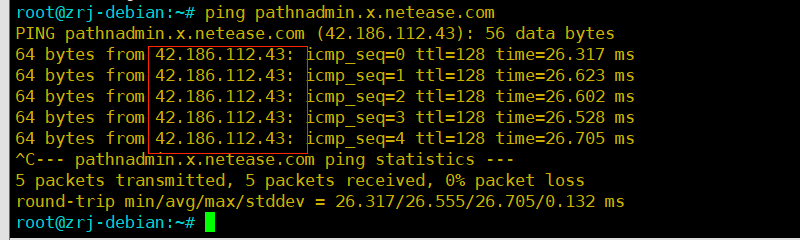
--bwlimit=1000 #限制传输带宽1000kps

--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：**



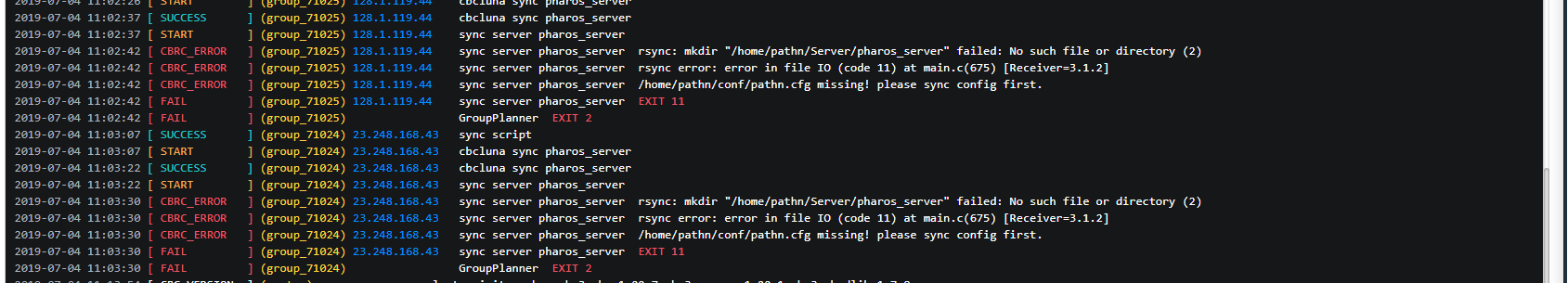
telnet 42.186.112.43 8873不通，网络问题，修改hosts，###该ip为pathn管理机外网ip



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。

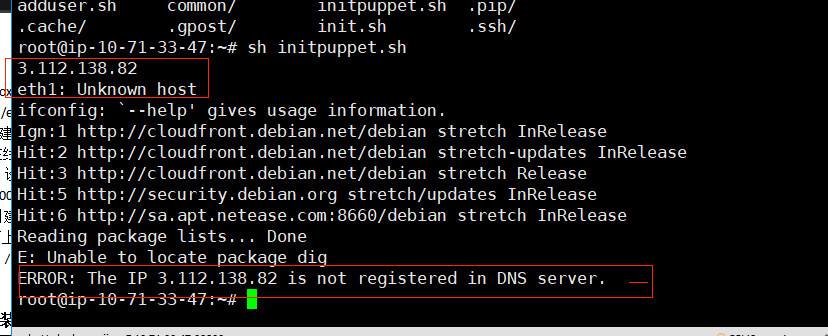
之后重跑正常。

**节点71024和71025报错，exit 11：**

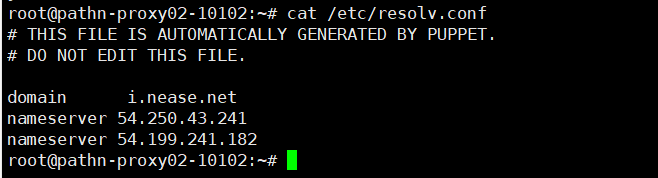
****

实为puppet未跑完整导致，重跑puppet后初始化正常。

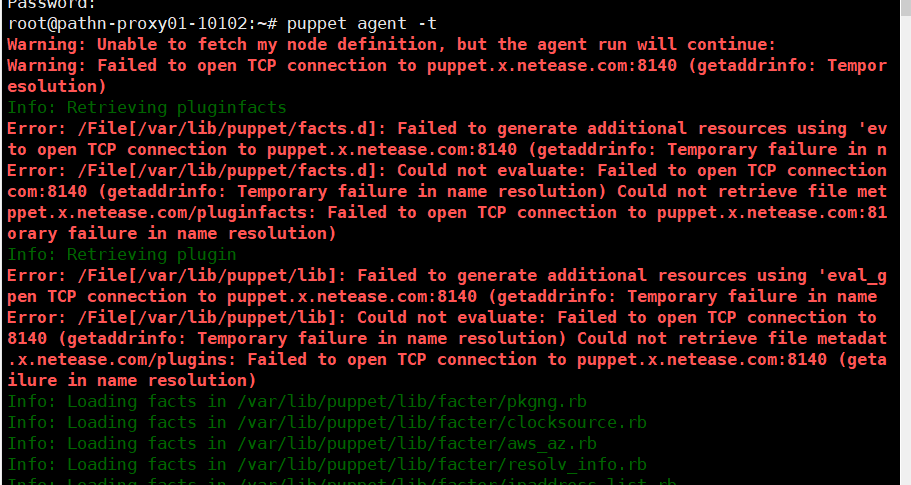
**Proxy01-10102安装puppet报错：**



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：

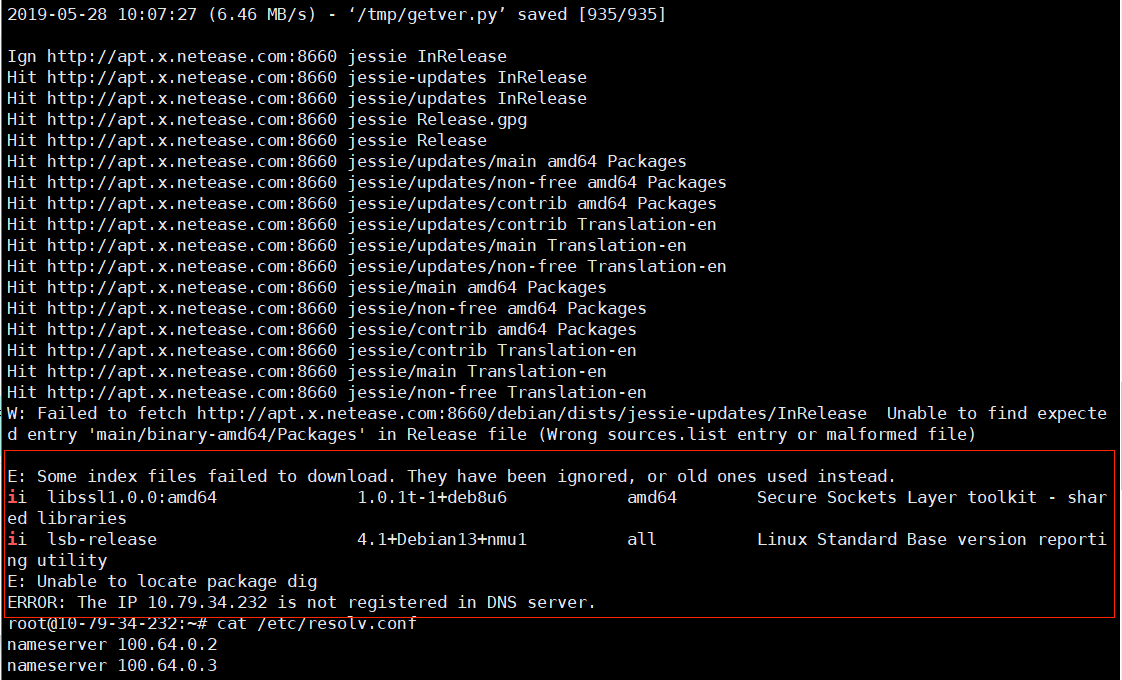


安装过后，puppet agent -t 运行极慢，有报错：



重复跑puppet后正常。

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

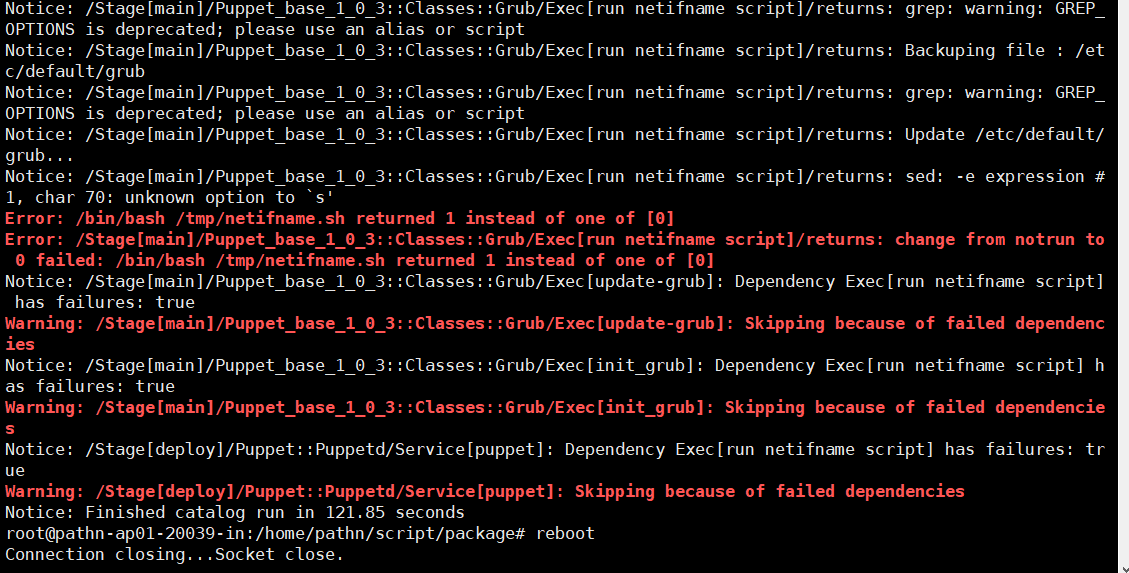
apt update

apt upgrade

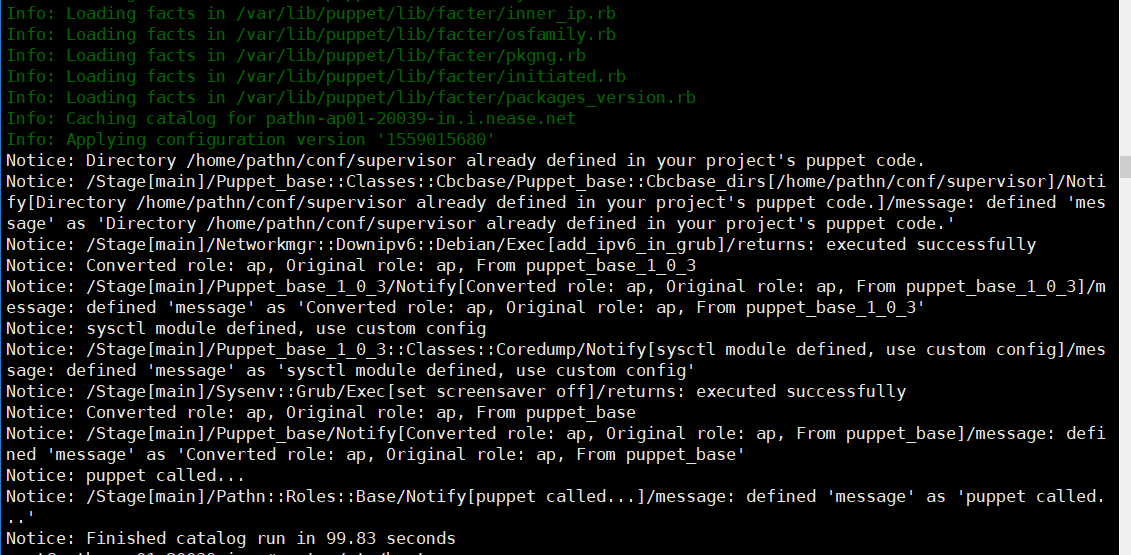
apt-get dist-upgrade

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：



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

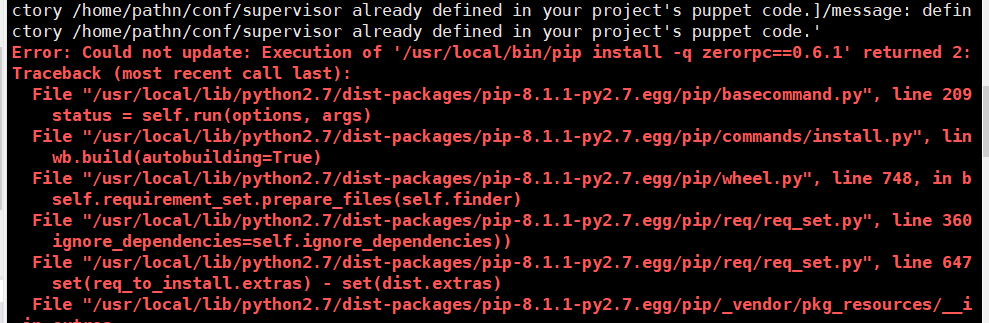
3.爱尔兰aws内网段不通，savpn申请添加新网段（加完网段再不通再找SESA值班检查一下aws的vpc设置）

**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

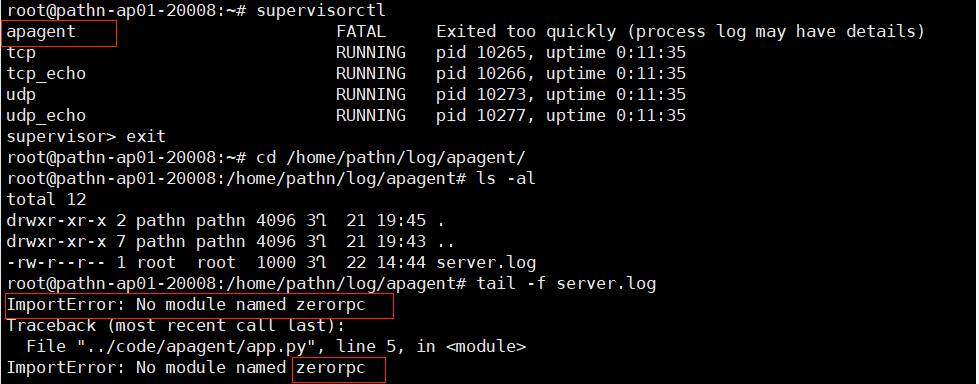
pip install pip==9.0.1 -U

apt install cl-cffi

手动安装zerorpc

pip install -q zerorpc

arm实例排查



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

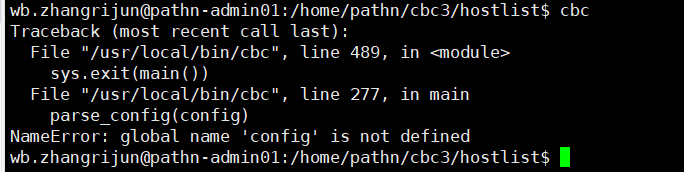
安装puppet后一般要确认finish再重启，tail -f /var/log/syslog ，在确认跑完一次catalog后再重启，不然重启过后，puppet agent -t 时会显示有puppet applying进程，需要等待。

四、apcheck(部署ap后进行验证)

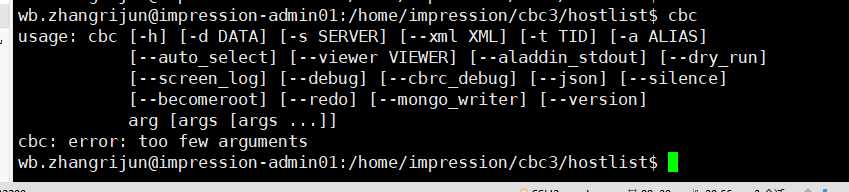
cbc xxx cbcluna sync apcheck

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.

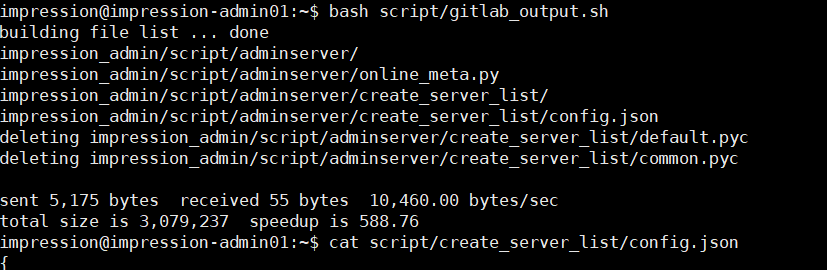
ap 的gid字段可以留空，pathn自动分配可用gid，如果再次指定已经使用过的gid，会把之前的群组覆盖。

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。



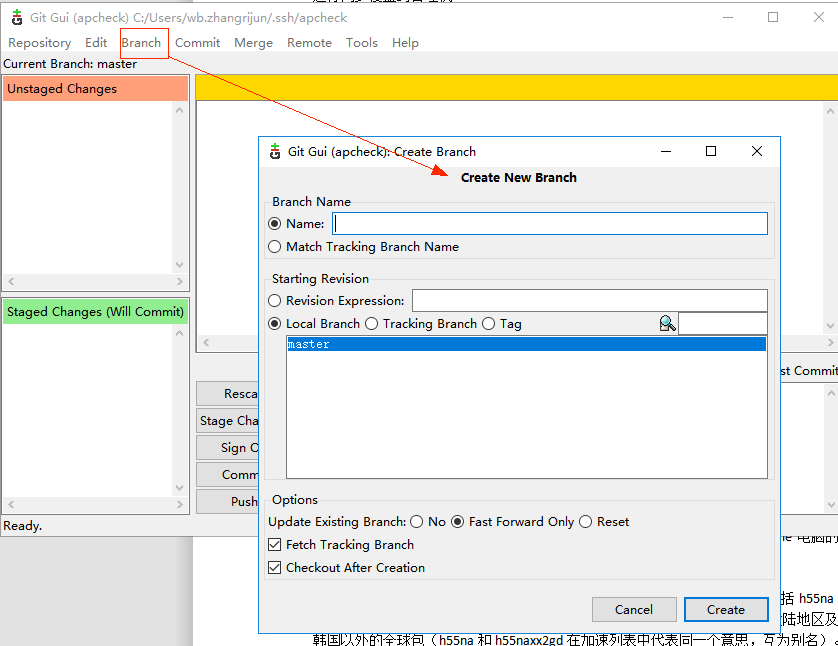
**Gitlab的key说明：**

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

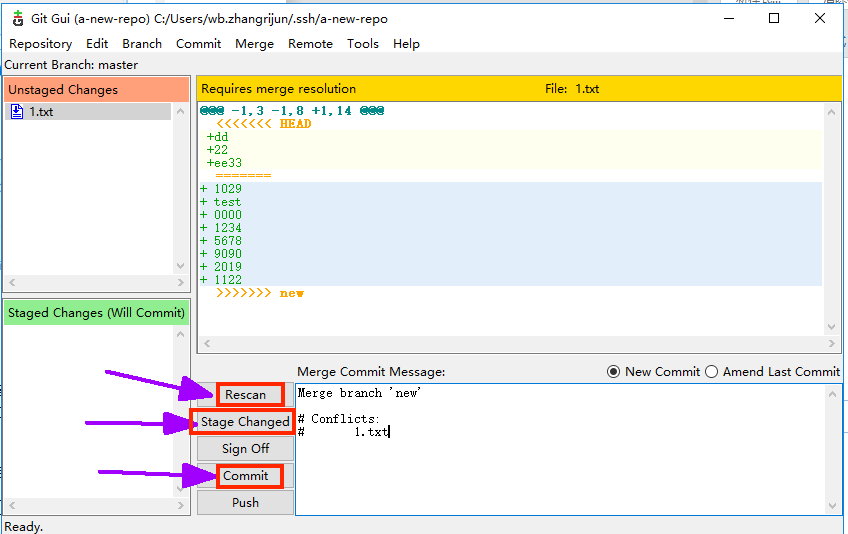
2.Xshel的key登记后，用于xshell登录其他机器（公钥在workflow登记后会同步到所在项目下的机器中，实现key的录入，之后才能ssh远程登录）。

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。

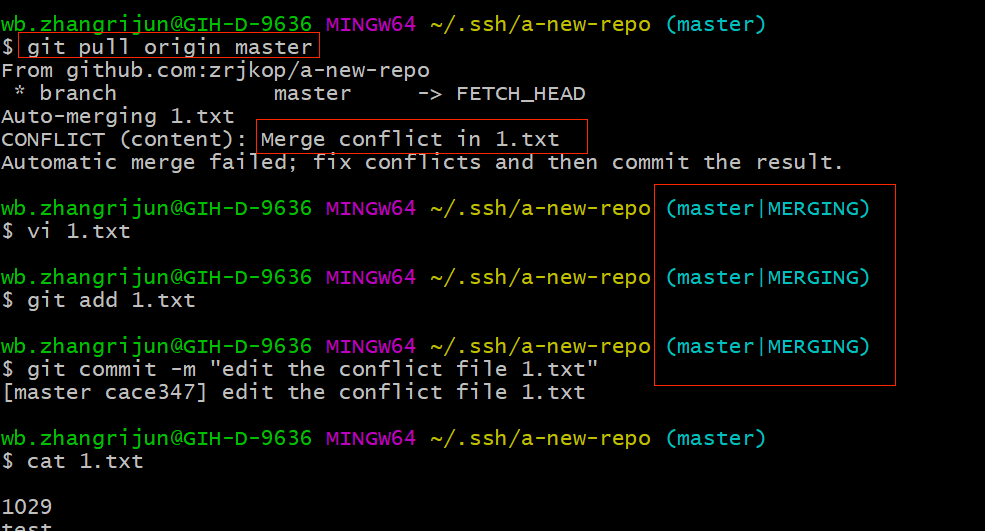
**Git-gui通过clone复制远端库到本地**，会同时把库的几个分支拷贝下来，一般会保留master,其他的分支，可以通过create branch创建，勾选match tracking branch name即同步远端分支名称和文件，之后可以切换分支和提交。



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]

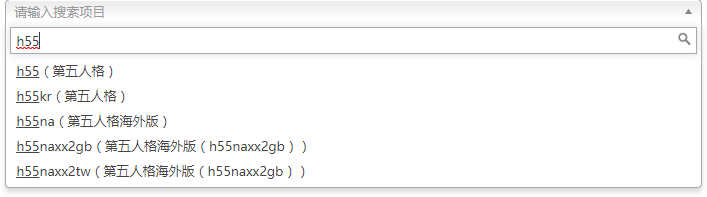
然后push

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。

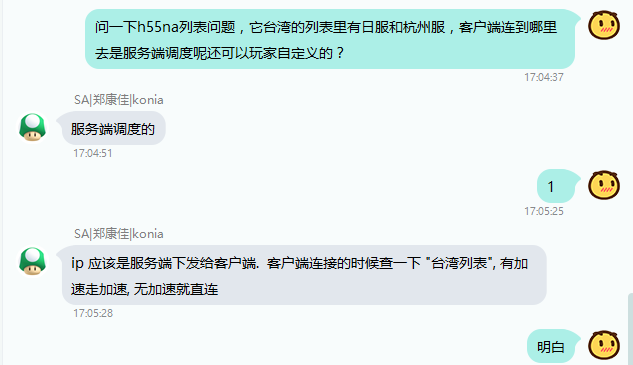
**项目在杭州和日本都有游戏服**，如h55na和g103，台湾地区的列表里面包含杭州和日本的游戏服：<https://impression.update.netease.com/lighten/pathn_g103_taiwan.txt>

<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。

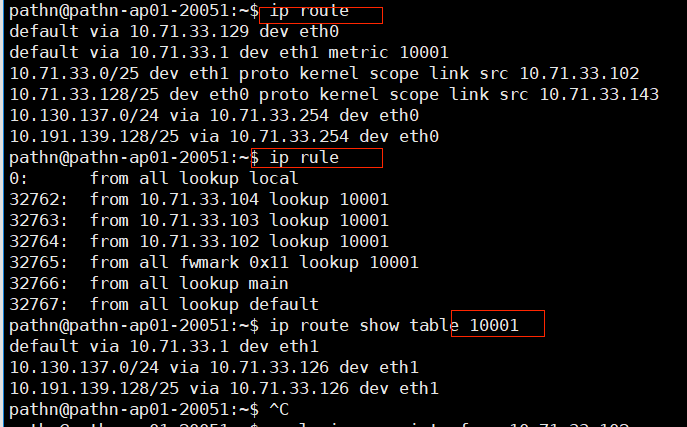
2：开启松散的反向路径校验。对每个进来的数据包，校验其源地址是否可达，即反向路径是否能通（通过任意网口），如果反向路径不同，则直接丢弃该数据包。

**~~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。

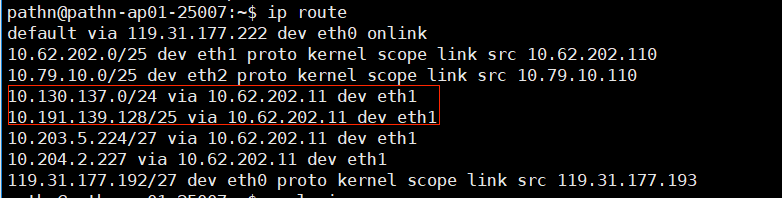
~~之前创建的节点，都统一使用了orgin：0x00000010和reply：0x00000011这个标签值，如果这么设置，需要rp\_filter开启，且会有性能瓶颈。rp\_filter默认关闭。~~可能理解错误，再查证。**下次验证新增指向10001节点时，没加aws标签时，看看路由的配置信息对比（需要阿拉丁初始化）；之后再添加数据包标记，测试一下origin使用0x11的情况，记录每步操作后apcheck的情况。结果**：20051，添加galaxy标签aws，（1）没添加标记时，初始化后有对应路由10001表生成，默认是通过.1的网关，但是端口测试不通,可能是.1的默认网关无法连通后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

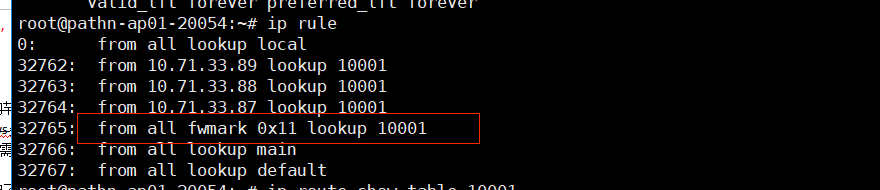
root@pathn-ap01-20085:~# ip route show table 10001

default via 10.68.24.1 dev eth1

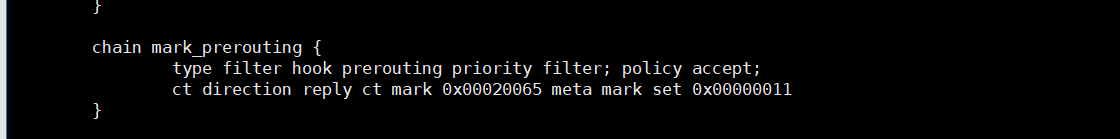
10.130.137.0/24 via 10.68.24.126 dev eth1

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

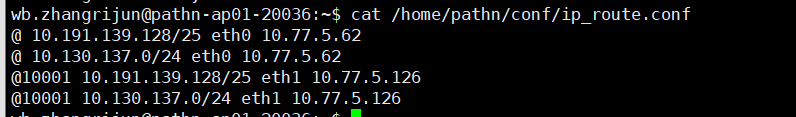
ct direction reply ct mark 0x00020008 meta mark set 0x00000011 }

**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：



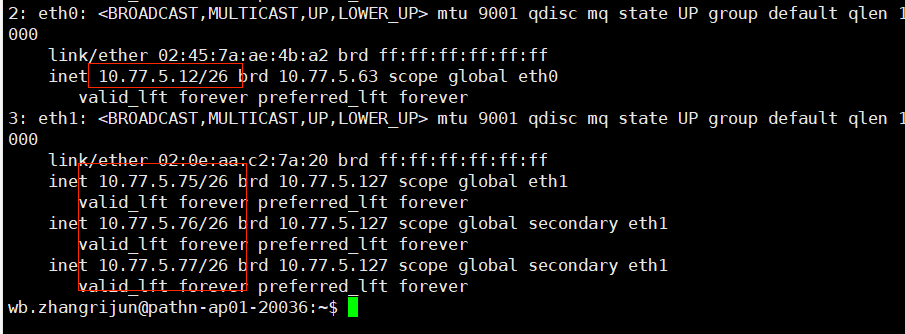
**节点20036指向10001**，初始化后，对应路由表10001中只有一条default规则default via 10.77.5.65 dev eth1， 测试不通，需要修改路由配置：

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

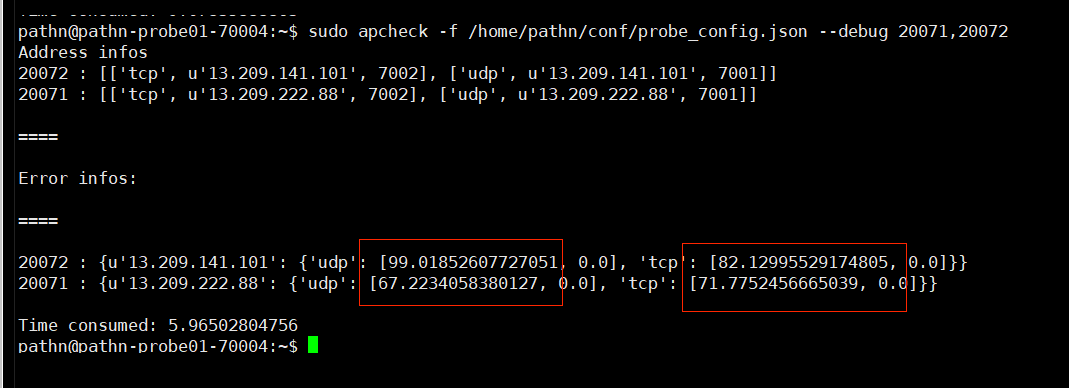


其他节点使用掩码25位/25,对应的gw分别为.126和.254

**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。**

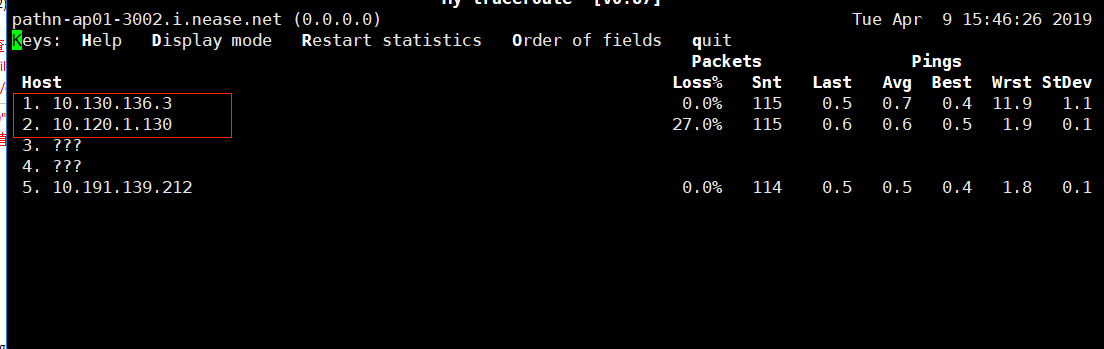
**20071,20072指向10003**，apcheck显示两个节点延迟有差异，



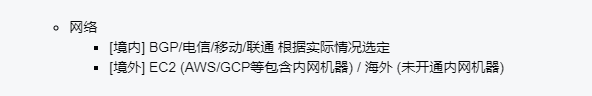
在20072ping后端，延迟比20071大，mtr跳转看不出原因，ip route 显示对应路由路径正常。排查为路由问题，是因为ping包去程走专线内网，回程绕道香港导致延迟大。康佳操作后端proxy刷新，之后正常。

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。



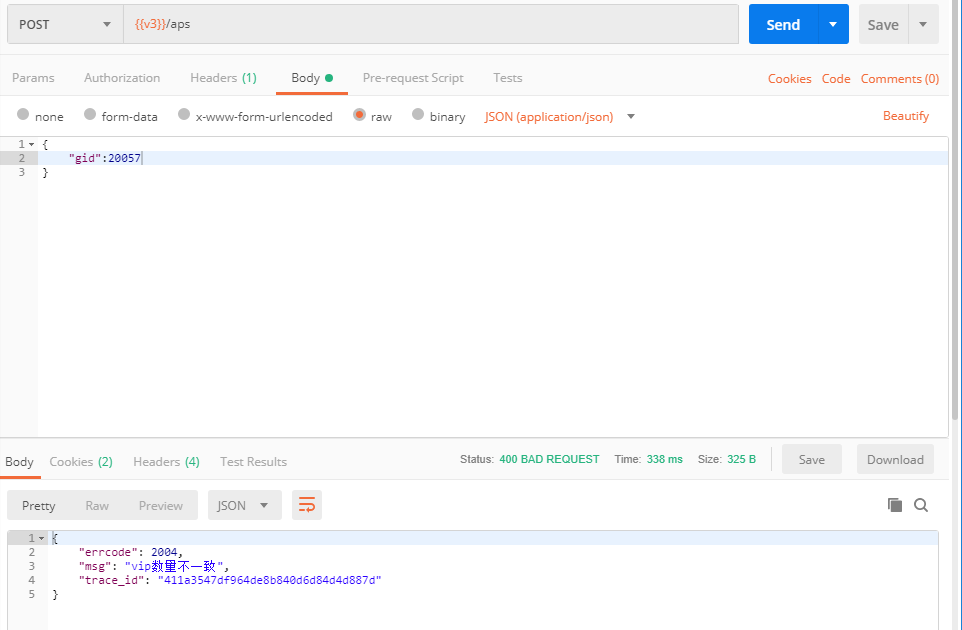
**专线节点机器多为虚拟机，多个节点共享一个实体机，登录一个节点即可验证走内网情况，查看backend.conf看后端ip，分别做mtr和traceroute验证。**

**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。**

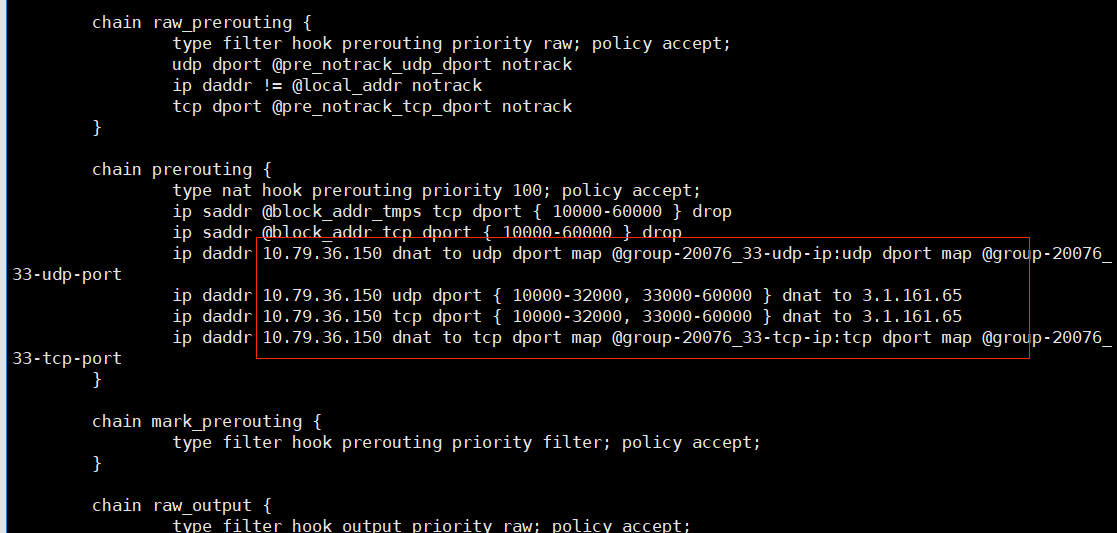
**Galaxy添加10202和20076映射：**

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。

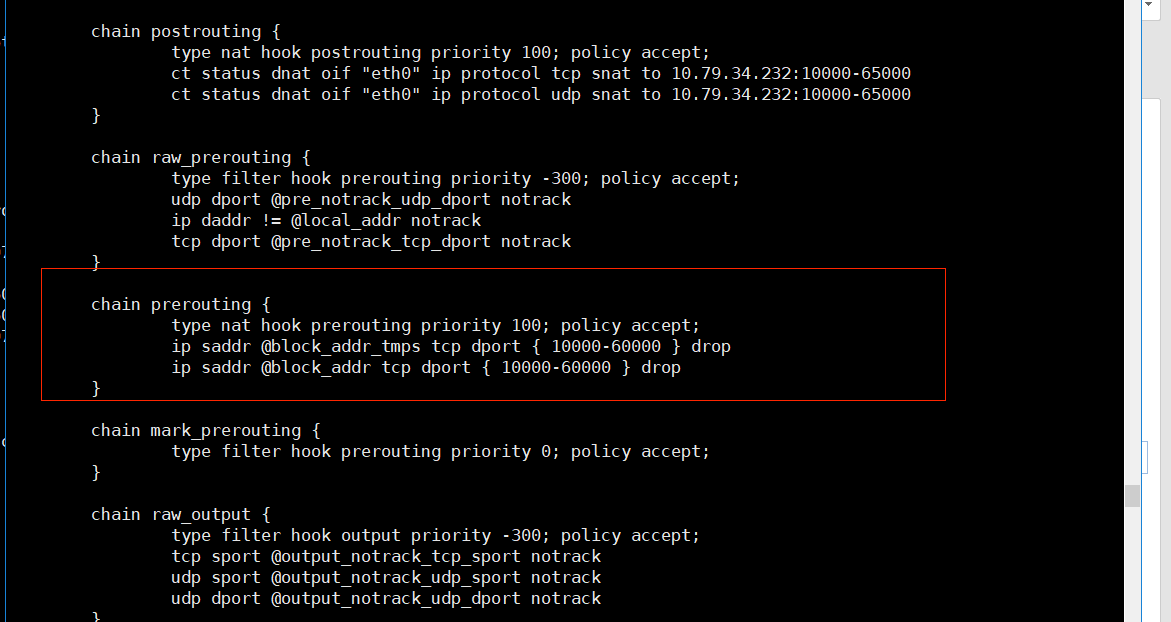
Galaxy生成映射，源ip使用内网ip，目的ip使用外网ip。

**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：**

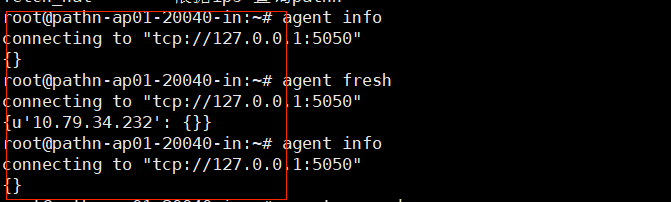
**20076：**



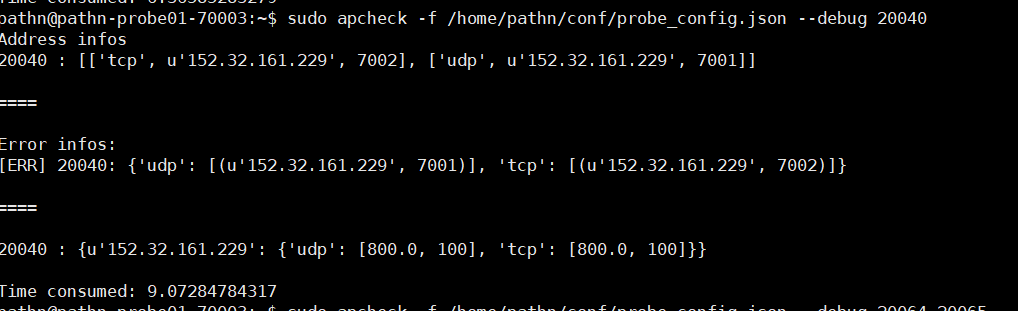
20039-20040:



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}

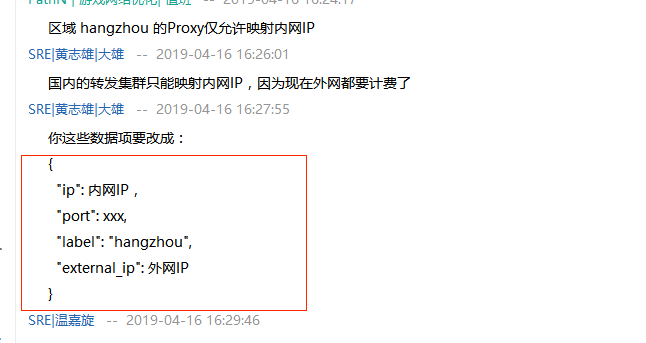
****

**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

**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)。

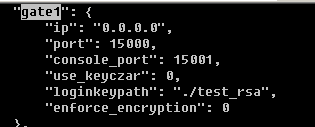
enable\_public 可选，国内机房默认仅允许映射内网IP，如果机器无法支持，将这个值设为1可允许映射外网IP。

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)。

**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

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。

Impression项目的echo机器，这个脚本具体目录待查。

**cbc操作：**

## 同步代码，初始化网络探测节点

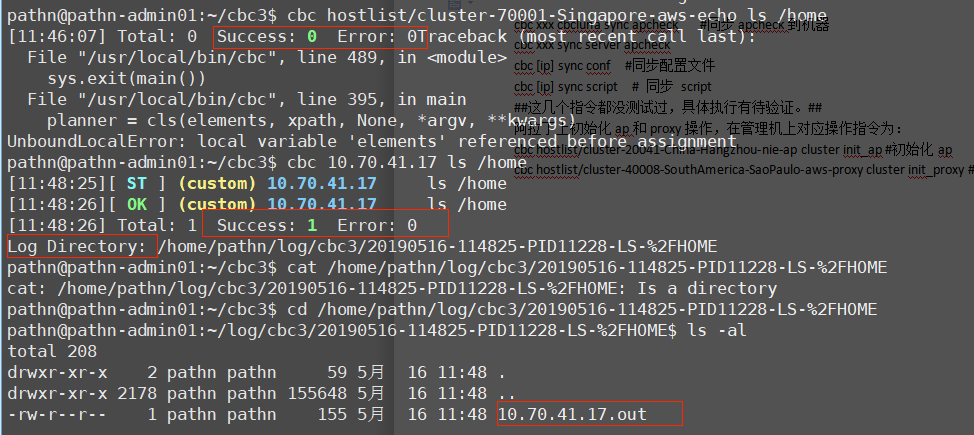
cbc hostlist/cluster-330xx cluster init\_probe # 探测机器

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 conf #同步配置文件

cbc [ip] sync script # 同步 script

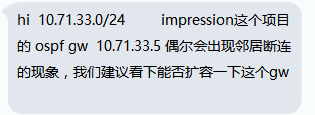
##这几个指令都没在管理机操作过，具体执行有待验证。##

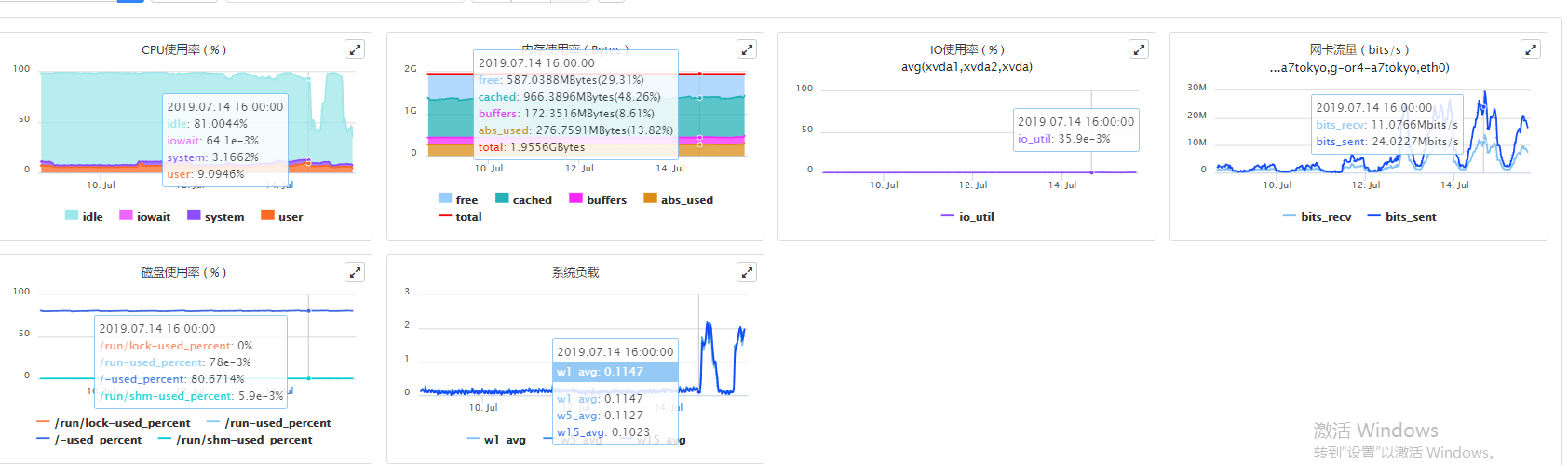
阿拉丁上初始化ap和proxy操作，在管理机上对应操作指令为：

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：**

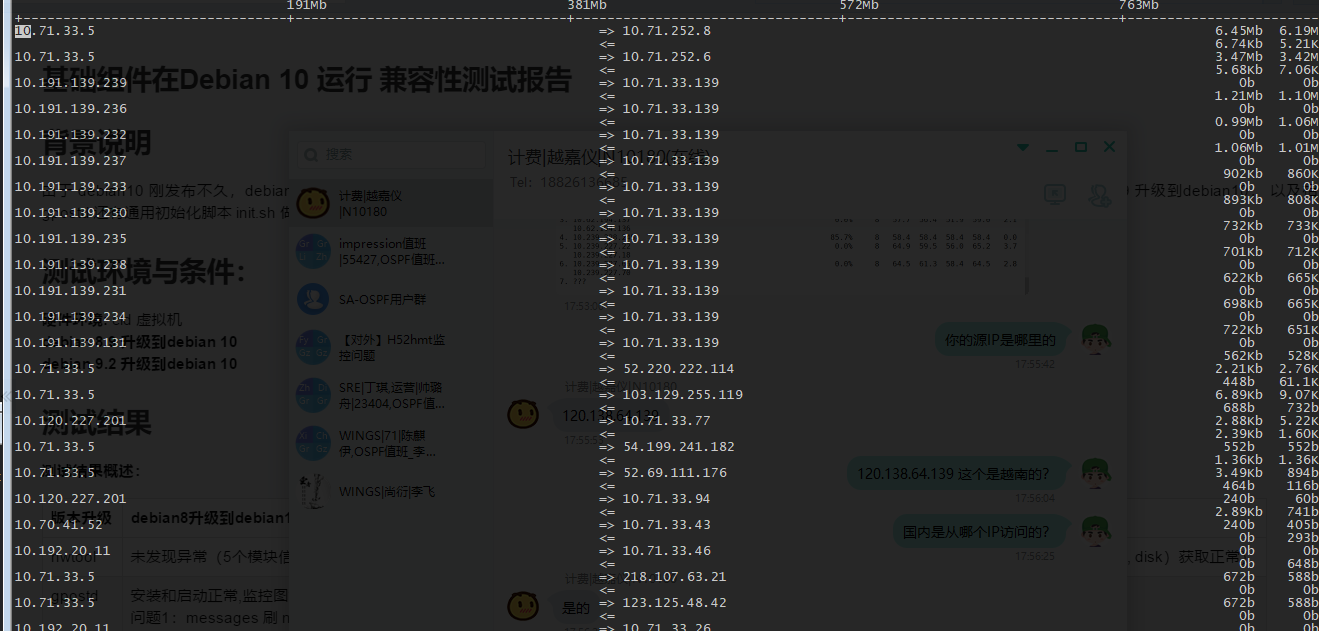




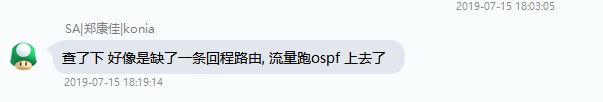
扩容步骤：skyline关机gw，对应网段会断网；skyline更改实例类型，实现扩容，之后重新开机。扩容不需要重新部署机器，会有开关机过程中网段的断网。

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：

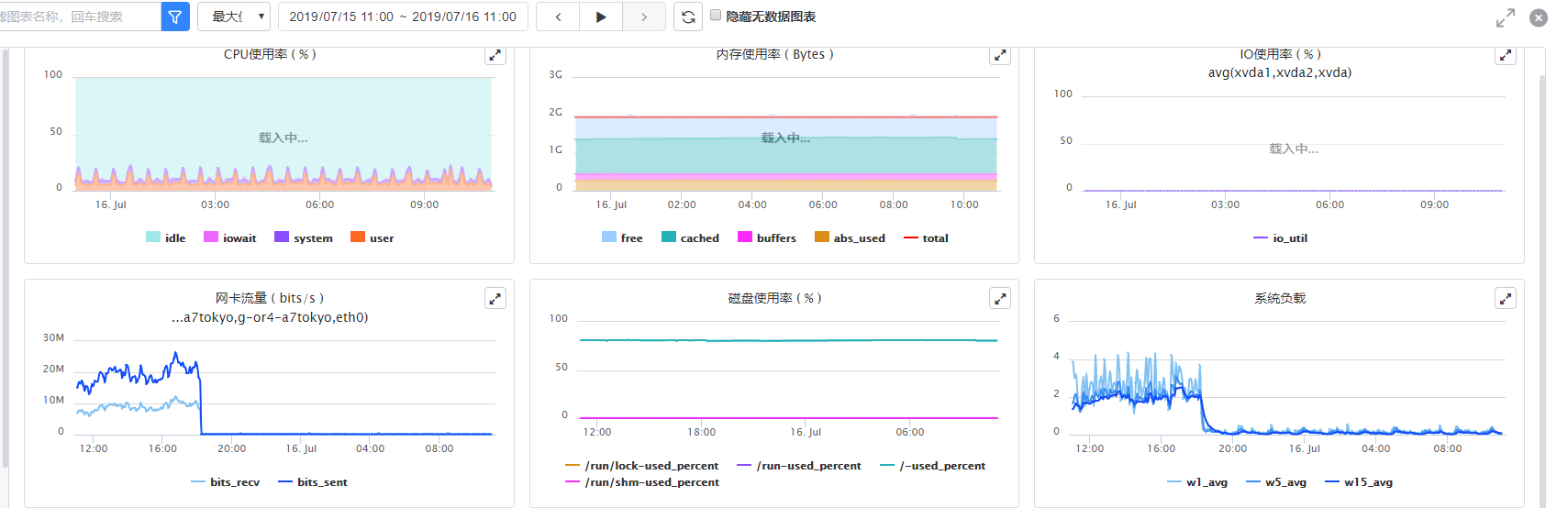


主要是10.71.33.139机器pathn-proxy01-10102，经查为缺少回程路由，导致流量走ospf

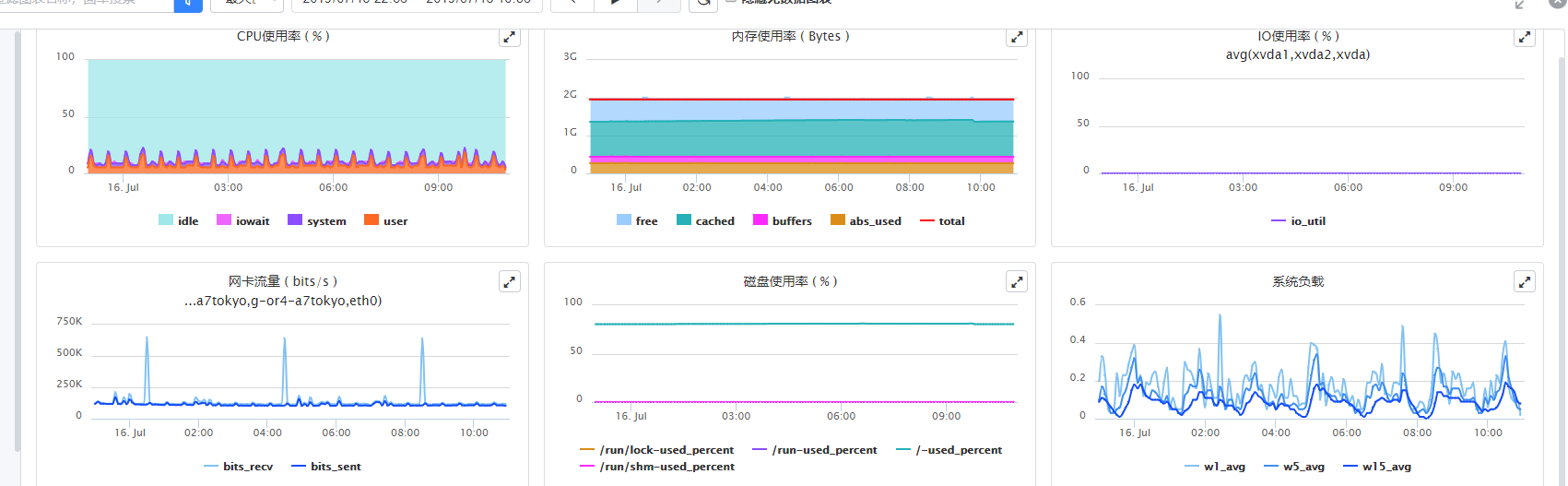


1. 调整机器路由，第二天再让ospf值班查看机器负载情况：

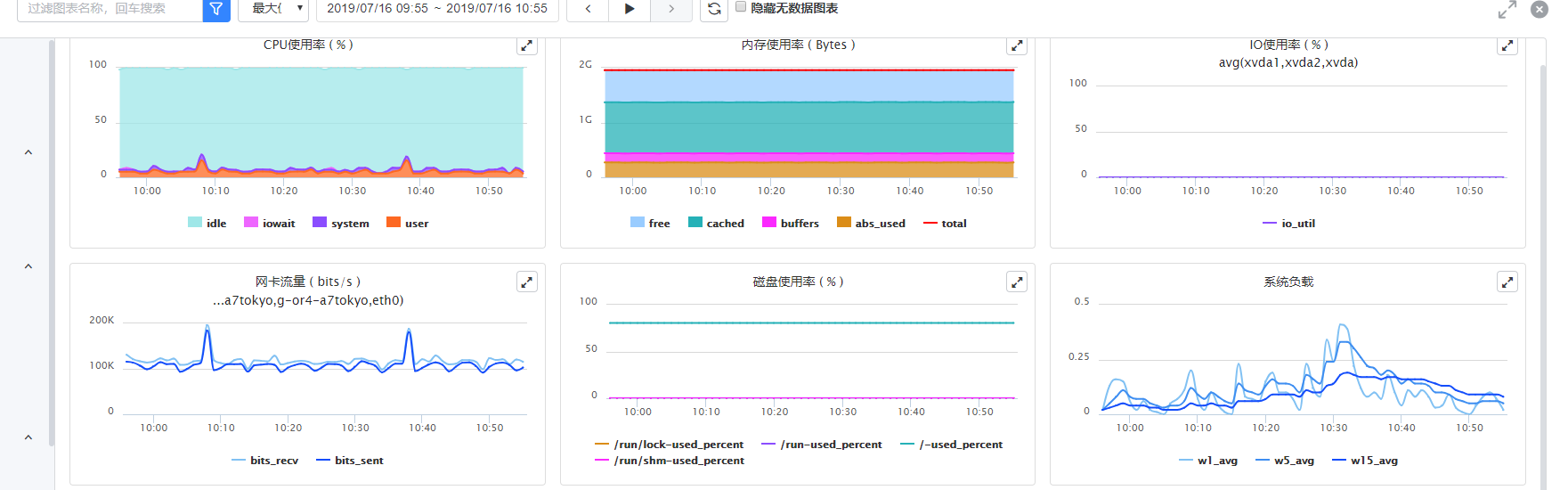
过去24小时：



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

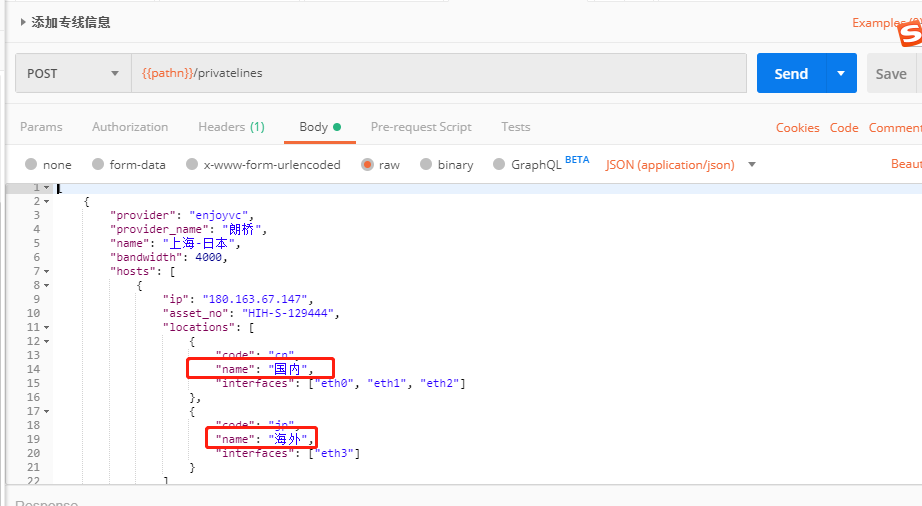
.5 是 ospf 搭的专线网关, 能通网易机房 10.x的网络

.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

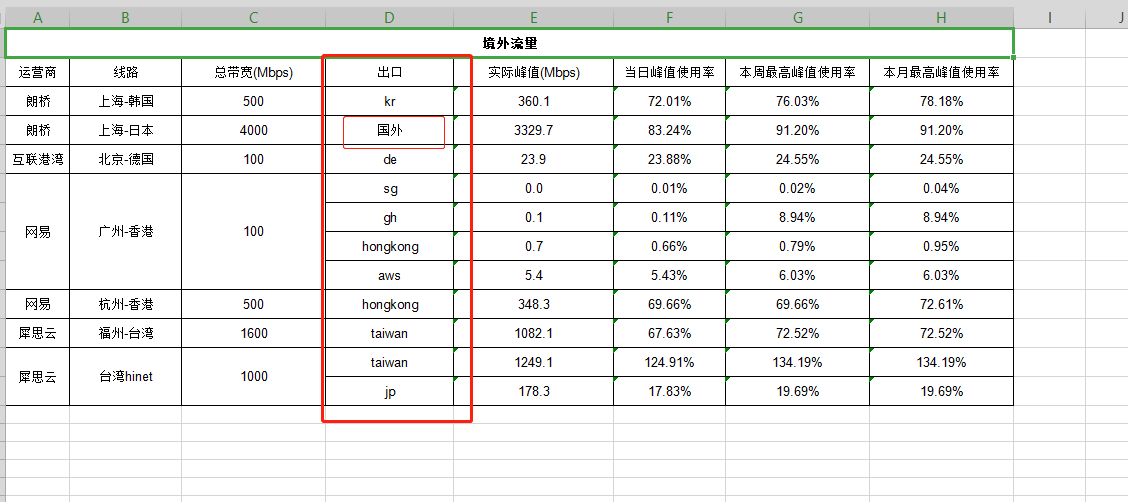
没资源就走ospf

**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：

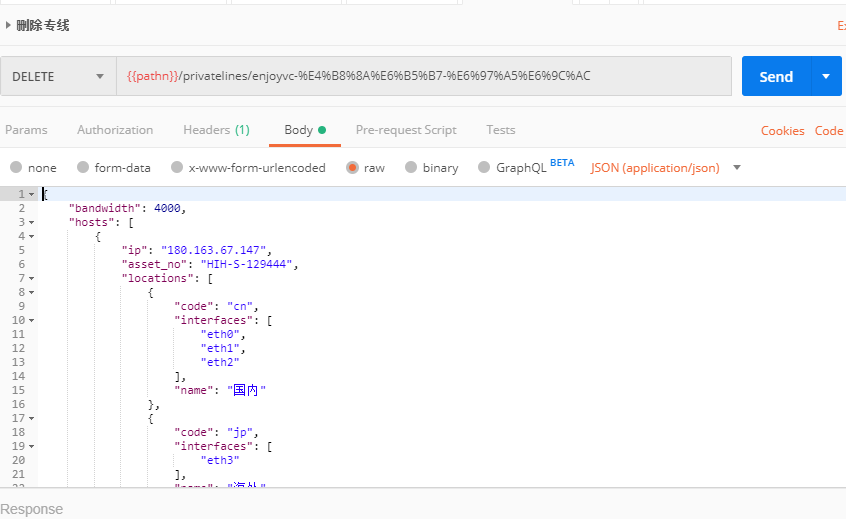


导致朗桥-日本方向之前配置被覆盖，只剩下180.163.64.4这台机器，需要重新导入配置文件。导入excel配置后，发现180.163.64.4这台机器的配置中name的字段值跟其他机器不一样，其他机器显示为null（未定义），这台显示为国内和国外，导入峰值数据是出口那一列也是显示“国内”“国外”，跟其他机器显示的“cn”“jp”不一致。



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。

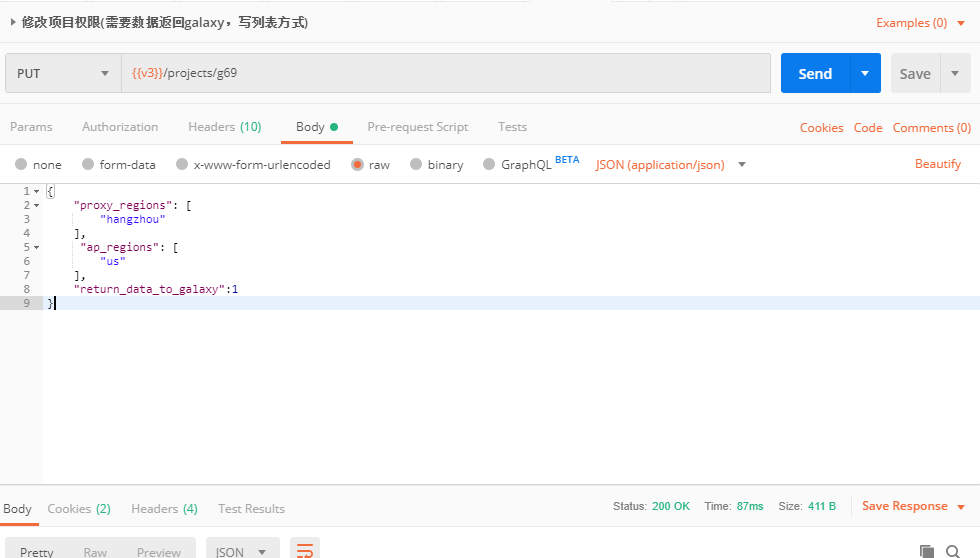


**g69申请临时北美加速入口，采用写列表方式接入：**

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。