2017.11.9(Day4) cloudy

Vocabulary

1.LTR(long term retention保留) kind of product as 5340

2.legacy refactor(transfer perl into python)

3.D2C

4.unify driver(Qlogic FC driver update)

5.Firefly(auto Framework)

6.SAS cable?Raid?

7.all hands（全体会议）

Concatenation 一系列相关的事物；revert 恢复;credential 证明;deployment 部署;

STH

|  |  |
| --- | --- |
| git  是代码管理工具 |  |
| bitbucket （version control tool）是git的辅助工具，用于 code review. |  |
| 就是写完代码后，要同事看过，才能真正进入代码，这是为了保证质量 |  |
| jenkins 是持续集成 工具。 |  |
| JIRA 是任务分配，跟踪的工具。 |  |
| scurm 是开发模式。 |  |
| automation 是自动化测试工具。 |  |
| automation 就是robotframe work.firework是基于robotframe的 |  |
|  |  |
| 以上这一套，应该说还是当下主流的，比较流行的一整套工具和模式的集合 |  |
| 比较有效率。 Linux 更新python至3.5.4版本。 建立软链接：1.没有删除旧版本python，故建立新的软连接至新版本 2.系统自带pip3和pip3.5,建立pip至pip3.5的软链接才能使用pip命令，or只能使用pip3或pip3.5 2017.11.10(Day5) sunny Vocabulary suite套件；roster 花名册;usage使方法 2017.11.13 sunny with strong wind Vocabulary Hierarchy 分层结构；velocity 速度，速率; audit 审计；archive 存档;poll 检查 调查; binaries 二进制代码；overaching 首要的;embrace 信奉，接受; collaboration 合作，协作;versus(VS); SME(subject matter experts);shippable 可交付的 Accountability比responsibility更强调严肃感;investment 投资; corporate level战略层次; leveraging 促使..改变; STH crucible 代码走查工具 Docker 代码集成工具;CI continuous integration ；Artifactory 仓库管理器; know how the Jenkins works in CI Diagram. Discuss how the artifactory works in CI Diagram(is it responsible for the automatically make,build,upload of the Git’s changes? Is it use docker or sth) LINUX Download code from Git 2017.11.14 sunny with strong wind Vocabulary Rear后面的，培养; metadata 描述数据属性的数据（on master server）;GUI(graphical user interface 用户图形界面); dedupe快速去重;deduplication 重复数据删除；NAS（Network Attached Storage：网络附属存储） replication 复制;duplication 复制；两者的区别是：前者只注重外形的一致，比如邮票的金箔复制品；后者则需要复制品和原版完全相同，比如文件数据的复制粘贴。 Eliminate 消除，淘汰;Fingerprint 这里指look up 查找 STH Q： 1.Backup Appliance with deduplication , deduplication Application 的区别。 2.5030产品没了？5030是Deduplication Appliance,其他的是NetworkBackup Appliance. 根据自己的理解： Appliance的核心技术是deduplication,核心功能模块有Master server & Media server &Storage.中者要进行deduplication的功能。 其中Client根据与Appliance的传输或者技术方式不同，分为SAN client, LAN client, CPD client etc.  根据Appliance存储备份信息的媒介不同，又分为VTL option, Openstorage(needs specific API) option, Flexible disk option,Pure disk option. 感觉在机房里看到的Rbot，Ebot都是Basic tape&disk. LTR(Long Term Retention):支持D2C功能，可以将数据传到云端，要做自己的私有云端。Instella team在做5340产品中将disk分成不同大小lun的工作。之前的每个产品都需要这个工作，但是因为之前用的netapp的硬盘，现在用seagate的磁盘，在进行分lun的工作时要有不同的API等，而且硬盘的存储空间大小也不同。 Code repository: Git? Jenkins builder: Docker? Storage: artifactory? Note for Applianceoverview PPT: 1.Storage有两种：integrate with media server & split with media server 2.Backup,replication,duplication的含义: Backup:client to media server Duplication: media server to media server with the same master Replication: media server to media server with the different master(个人理解是，属性变了) 3.Pure Disk Appliance& MSDP Appliance,前者在哪里做去重呀？现在主要是MSDP. 2017.11.15 sunny with slight wind Vocabulary Hot spare热备用，空闲，加电待机的备用磁盘;HDD(hard disks drive)硬盘驱动器;HA(high available) STH Q：1.Appliance sever 和 RBOT EBOT里的disk类型一样吗?每个Drive Bay里都是SCSI盘 2.每个slot的连接类型是根据disk类型决定的还是disk类型都一样并且可以兼容各种类型只是根据不同的MODE决定的？ 犯了根本性错误，slot里插得是网卡或者光卡。 RBOD和EBOD之间通过sas连接主要是成本考虑，都是走的scsi协议 5330的RBOT和EBOT的存储结构： 1.6 data volume groups(每一个group里有9个disks,7个data disks+2parity disks) 2.1 metadata volume group(该group 里有4个disks,2个stripe+2个 mirror,两者的读写速度相反，前者最快，后者最慢（用来备份）) 3.2个disk用做Hotspare 4.EBOD和RBOT存储结构一样。 2017.11.16 sunny Vocabulary Convention 规定，习俗;CFT（Cross Functional Team） STH JIRA是任务分配追踪的工具。https://www.ibm.com/developerworks/cn/rational/1606\_wanghy\_saf/index.html How to defect….. 只有5330 Appliance的storage分为RBOD+EBOD，其他的都是JBOD（没有前端逻辑控制各个磁盘，即raid控制器，JBOD和EBOD几乎没什么区别）。 Code insurance是在进行check in(commit)之前需要做的： 该事项需要登录stashboard,在codeinsurance里在repository里建立自己的branch并选择Appliance,然后会自动生成相应的ISO，然后在选择的Appliance上测试自己修改后的code。 在熟练之后，可以先把修改的文件打成rpm包，直接mount到选择的设备上，手动执行firefly命令进行测试。 Attention：iso包含的文件比rpm多。 工位换了地方。 2017.11.16  Install appliance 看文档 2017.11.17 在虚机和物理机上装APPLIANCE  2017.11.20 sunny 1.git上存code source Atifactory 上存iso 2.用IPMI设置IP后就可以通过SSH登录并操作Appliance了 3.在物理机上install Appliance 4.明天继续在虚机上装Appliance 2017.11.21 cloudy 1.install Appliance 遇到socket written failed(24)问题。 2.install Pycharm on workstation 3.组会。 2017.11.22 sunny 1.FCR(fibre channel replication) 2.HBA(host bus adaptor) Sth Summary of installing appliance procedure: Problem： 两台appliance都装好后，在media上与master进行认证时失败，错误类型为24，执行bptestbpcd 命令时，socket write failed. 排查原因是因为作为media的n39-h21 appliance没有安装好，其ISO文件还是之前的版本，导致两台appliance的ISO不同，从而不能成功通信。 重新安装n39-h21,因为master上已经添加了media appliance,此时media与master通信需要证书的authentic,所以要进行清除证书操作。 在上述操作过程中，需要登录密码，在automation的配置文件里找到一个但是不能有，后来发现是因为我的appliance的clish里没有用户，要先建立一个用户并设置密码，用该密码登录才可以。 成功清除证书操作后，继续在media appliance上进行media role的设置，但是在name look up 步骤时failed 后来发现是因为在hosts add操作时，把设备的主机全名写错了（short name.域名），重新添加一遍之后，成功设置media role。（但是在此过程中，第一遍还是出现了错误类型为24的failed，在重新执行命令之后，没有复现，直到成功，难道是系统不稳定的原因？） Learning： 设置IP时四大要素：ip address+掩码+gateway+dns 查询各种相关版本方式。 在automation上找密码的过程中需要了解robotframework和代码的框架，暂时了解到一些命令的使用，例如fg,jobs,set nu,grep. Daemon概念。 主机的short name和long name。 2017.11.23 sunny Vocabulary nuts and bolts 一些具体事情的具体细节；debriefing 任务报告;template 模板,样板;galore of 很多;acceptance criteria 验收标准 sth 了解robot framework 2017.11.24 sunny 了解firefly 并运行testcases. 遗留问题：ENV\_ROBOT\_PATH 在哪定义 2017.11.27 cloudy What I have done today: view testcase teambuilding Nov. 2017.11.28 sunny with strong wind About testcase: backslash character反斜线字符 case-insensitive 不区分大小写 ： 转义字符 Reference linux commands today: Dirname; export; alias; echo; grep 知识点:1.vim快捷键2. python正则表达式中的re模块！ What I have done today: view the san testcase know sth about pull request for modifying automation team meeting(understand the goal about main pipeline run time & test case numbers etc.) 2017.11.29 sunny  About FCR: PD(PureDisk) AIR(Auto Image Replication) Opt-dup FTMS(between SAN clients and master or media)，FCR(between master and media) FCR diagram is as below: Sth important: 1.Appliance models’ template path: /opt/NBUAppliance/scripts/fcr/template 2.lspci ：可以找到目前主机上的所有通过总嫌连接的硬件设备。 Appliance pci info path:/usr/share/hwdata/ 3.查看设备搜slot信息： from vxplatform import \*dir()里面有各种查询Appliance信息的函数 Reference linux commands today: lspci; find; 关于find有点要说明：在结合正则表达式‘regex’搜索文件名时，需要注意匹配文件名的正则表达式前面不能用’^’而是用’.\*’,因为要搜索的整个路径，路径本身就是一个字符串，而文件名是在最后而不是最前面。 知识点： 在其他终端里杀掉本终端里死掉的进程： ps –U username 根据tty号和进程的名字查找出对应的PID号 ps –s 9 PID号 Done!! 2017.11.30 sunny About fcr:  1.try to install docker Reference linux commands today: git pull(更新代码) ‘ctrl+Z’把当前打开的文件挂起，之后用jobs+fg组合命令重新打开就好了。 sh. Linux下打开pycharm. 知识点：1.testcase 的Python 实现在lib文件里。 2.UMI(Unique Message Identifier),用于记录clish里的错误类型，帮助用户处理操作异常问题。 3.what’s base directory? 2017.12.1 sunny About installing docker:  用yum install docker-io安装时，报错“没有yum模块”：原因是/usr/bin/yum文件里使用的是python,主机指定的是python2.7.7版本，该版本里没有yum模块。于是用pip下载yum，又报错“Could not find a version that satisfies the requirement yum (from versions: ) No matching distribution found for yum”，暂时没有继续解决。于是用python2.6里的yum执行该命令后成功安装yum（具体操作是/usr/bin/yum文件里使用的python版本改了一下）。 Docker pull步骤总是在下载的最后环节报错“Error: image dev\_builder:latest not found”，在docker restart时报内存空间不足。于是df –h之后发现默认下载的位置内存不够。 因为/home的剩余空间很大，所以将/home的空间分给/一部分。具体操作步骤如下网址： http://blog.csdn.net/llnara/article/details/50205303 值得注意的是，操作过程中/home上不能有正在进行的进程，如果用“普通用户登录系统之后切换到root下操作”这种方式会导致/home上有很多进程，必须要直接以root用户登录。（具体原因好好查查） |  |

**Reference linux commands today:**

1. df –h
2. ctrl+r 快捷键
3. lsof
4. du
5. df
6. fuser
7. ps –aux
8. fdisk –l
9. vgdisplay
10. lvdisplay
11. service network restart
12. umount
13. /etc/sysconfig/network-script

知识点：

1. python文件格式 首行格式
2. docker

2017.12.4 sunny

Pseudo 假的，冒充的；

1．Meeting with new manager

2.docker 基于LXC实现，namespace&cgroup概念。

3.在git上下载，证书丢了所以不行；证书没问题的话，直接git clone就好了。

用python 安装firefly，docker和linux的挂载点（这里是git本地仓库）是同步的。

**Reference linux commands today:**

ethtool -i docker0 ll /sys/bus/pci/devices/0000\:0 lspci | grep -i eth ifplugstatus

Docker ps -a docker attach 08c2dd851a1b

2017.12.5 sunny

Prompt 提示;

1. Etrack用来给nbu说明appliance出现的跟nbu有关的bug.

2.access是私有云项目，低层是印度那边在做，我们负责shell部分。

IT(integration test) :利用脚本

E2E(end to end test)：模拟用户在clish的操作

UI(单元测试)

**Reference linux commands today:**

1. lsof：查看端口使用情况
2. find
3. rm+正则表达式

知识点：

1. 虚拟机网络设置与宿主机的链接模式：桥接,NAT, host-only
2. Docker原理详细介绍：<https://www.cnblogs.com/huatianbuyuexia/p/6540166.html>
3. 正则表达式的重要性！！！！！

2017.12.6 sunny

**What I have done today:**

1. 给自己的gitrepo设置本地仓库
2. 看python书
3. Linux的正则表达式没有看完。

2017.12.7 sunny

**Reference linux commands today:**

1. echo > ‘filename’:清空文件内容；
2. echo ‘info’ >> ‘filename’:往文件里写入信息
3. lsmod:其实就是list modules的缩写，可以显示内核已加载的模块。
4. dmesg:用来显示内核信息，kernel会把开机信息存储在ring buffer当中，开机时来不及查看的信息可以用dmesg查看。开机信息被存放在/var/log/dmesg里
5. modprobe：智能的往内核加载或删除模块
6. modinfo：查看设备详细信息
7. http://www.thegeekstuff.com/2015/02/rpm-build-package-example/

2017.12.8 sunny

**Learn sth new :**

1. 更新本地git仓库的代码：git pull
2. 程序入口，ymal文件？
3. 在windows上下载了代码

2017.12.10 sunny with strong wind

Macros 宏定义 ；font 字体；patch 补丁；

2017.12.11 sunny

Sth about program：

Makefile文件：基本功能是描述源程序之间的相互关系并自动维护编译工作。（主要通过冒号来描述文件依赖关系并通过比较有无更新进行command的执行）

spec文件：基本功能是描述将source code打包成RPM包的detail info

xml文件:程序配置文件。

**Reference linux commands today:**

Grep的用法：

1.ls –l |grep “location1”location2（如果搜索的正则表达式里没有空格，location1的双引号加不加无所谓，如果表达式里有命令，会先将命令执行输出后进行查找。但是跟单引号是有区别的，单引号是不会进行命令解析的。）

今天遇到的问题是：目的是在ltra的所有文件里找到unittest文件。进行了如下尝试：

1. ls –l |grep unittest \*，发现此时是在ltra里所有文件内容(注意只是在当前显示的这层目录的文件内容里)里进行文本的搜索。
2. ls –l |grep unittest，此时是在当前文件夹的列表也就是ls –l的标准输出内容里查找。

如果想在达到目的，可以使用“find 目录 –regex 正则表达式”来查找。

2017.12.12 sunny

Learn sth new:

1. 先简单的理解为设备的驱动需要作为模块加载到内核当中，而lspci显示的是总线方式连接的设备，lsmod显示的已经加载到内核的模块。
2. HBAs是光纤通道的适配器使用时需要driver，这里是qla2xx和qla2xxtgt。看fcr的结构图，这里的HBA的数据通信协议为scsi。
3. 传统上说的网卡其实就是adapter，根据数据通信协议的不同分为三种，其接口类型都有光口和电口，光口都是通过光纤线缆传输数据，电口的接口各有不同：
4. 以太网卡（Ethernet adapter）,通信协议为tcp/ip，与以太交换机连接；
5. FC网卡（Fibre Channel HBA）,和光纤交换机连接，通信协议为光纤通道数据协议；
6. ISCSI网卡（ISCSI HBA），通信协议为ISCSI数据协议。

2017.12.13 sunny

Sth about triage:

1. automation两个主要的分支：master和dev\_master,分别对应firefly和testrun里的testcase.
2. 根据failure case的report名称可以分辨出是哪个。

2017.12.14 sunny

Intermittent 间歇的，断断续续的；LDAP

* + 1. sunny

nbapp639：physical machine as media(MTV’s name rules)

sclmifaappd01v02:virtual machine as masterapp

2017.12.22 sunny

Install and configure d2c appliances：

**Reference linux commands :**

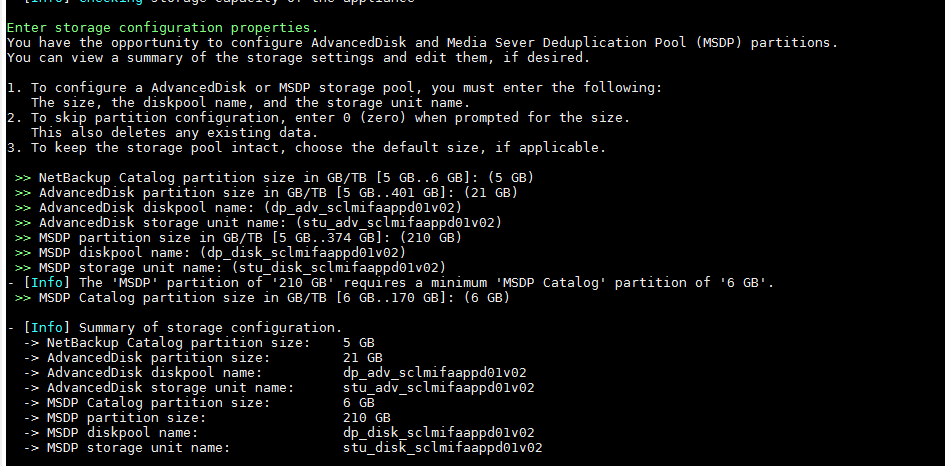
1. Ps –ef
2. Wc:

legacy problems:

1. nameserver,domain,dns,hosts,resolv.cofig
2. aggregate report & report
3. 如何设置才能在ssh时直接输入主机名就好了？今天用ping命令测试了/etc/hosts和/etc/resolv.conf的情况。

process：

1. 第一次安装之后遇到两个问题：以新建的cliuser身份登录appliance查询相关policy没有结果；执行TEST software 的CLI命令之后，新建的diskpool的status为DOWN。
2. 之后进行了factory reset，重新配media role，进行证书的remove，在新建bucketname时重新选择名字（带appliance的容易被清掉）。上面两个问题都解决了



2017.12.26 sunny

Learn sth new:

1. OS 在与别的主机通信时优先到/etc/hosts文件里查询，没有目的主机的时候再去/etc/resolv.config文件里查找。/etc/resolv.config里有nameserver和serchdomain，对于公司的专用网来说很重要，先去serchdomain里找目的主机，没有的话去nameserver里找。

综上，这两个文件夹在主机间通讯时与DNS服务有关的，和hostname没有关系。

1. 关于hostname，其实归根结底，OS会在查询到主机名后改变内核的一个参数从而显示出来，即 /proc/sys/kernel/hostname文件，但是该文件的值会在OS启动时根据/etc/hostname或者/etc/sysconfig/network里HOSTNAME的值而定。一般系统优先根据/etc/hostname文件来决定主机名。如果没有该文件的话，OS会去/etc/sysconfig/network里查询。所以如果直接修改/etc/sysconfig/network里的值而不重新启动的话，该修改是不生效的。而直接修改/proc/sys/kernel/hostname里的值，重新建立终端就会生效（执行hostname也会生效，因为该指令是去内核里查询），但是重新启动后就失效了，因为会系统会根据前两者的值修改内核的值。
2. 今天遇到的问题是，10.192.7.70的workstation与nbapp639建立连接之后，后者进行了re-install iso,在workstation再次用ssh试图与nbapp639建立连接时，rsa秘钥认证失败。

**原因是**：用OpenSSH的人都知ssh会把你每个你访问过计算机的公钥(public key)都记录在~/.ssh/known\_hosts。当下次访问相同计算机时，OpenSSH会核对公钥。如果公钥不同，OpenSSH会发出警告，避免你受到DNS Hijack之类的攻击。

**综上：**有两种方案：一是将保存在~/.ssh/known\_hosts里的公钥删除；而是在/etc/ssh/ssh\_config文件中修改StrictHostKeyChecking 后的ask设置为no。后者会一劳永逸但也会增加风险，但是对于公司内部的主机来说风险还好。

1. 用浏览器打开.html文件的设置方法（遗留）。
2. 对于打算放在后台运行的任务又想进行操作，可以在命令后面加上“&”
3. 关于Unicode。
4. 不同主机之间复制文件：scp命令

2017.12.27 sunny

Triage遇到的问题：

1. nbapp2eg出现了core dump，trace by story <https://jira.community.veritas.com/browse/APPSOL-71904>（ upgrde pipeline）

gdb命令打开core dump文件

1. i213-eng142 deployed failed, make lab ticket [REQV0169741](https://veritas.service-now.com/nav_to.do?uri=sc_request.do%3Fsys_id=4d242f5ddb178b40c9b3fb541d961982%26sysparm_stack=sc_request_list.do%3Fsysparm_query=active=true) to track.

2017.12.28 Cloudy

1.Support下的Storage Reset在appliance配role之后就没了。

2017.12.29 Cloudy

1.classic master 用来指做NBU，可以用来做master，此时只能用root身份登录，而nba只能用admin身份登录，这也是区分是不是appliance的方法。

2.Vim通过tabnew实现打开新的分页，并且用“n-g-t”或者“ctrl+6”组合键实现页面切换。

3.在i213-eng142 deployed failed后，查看control console看到http的response消息，此response是从一个数据库返回的，而不是从i213-eng142本身（遗留：什么数据库）

4.遗留：

1.修改完case跑firefly在新建linux user时失败，查看原因是master IP 解析失败。master ping不通，但是是台虚机，不能用IPMI登录查看信息。周二回来找人帮忙看看；

2.ENV\_MASTER\_NBU\_...变量是在哪里定义的？

2018.1.1 Sunny

1.ctrl+F网页内搜索

2018.1.2 Sunny with strong wind

1. csconfig cldinstance –i查看appliance支持的clouder provider和region。

2.正则表达式：在匹配有空格的string时，虽然正则pattern里设置为相应的空格数时也可以匹配成功，但是建议匹配空格时用‘\s+’来匹配，防止该string的间隔变了的时候匹配失败。

3.during configure d2c.different region has a same access key id but Beijing.

2018.1.3 Cloudy

D2C upgrade的3.1iso run automation case的过程：

在3.1iso上run automation suite，发现找不到一个.py文件

原因是在3.1.1iso里添加了该文件用来改变选择cloud provider和region的方式

修改suite里的配置方式的testcase，发现cloud provider在变量si定义时赋值为加引号的形式导致在匹配关键字时失败。

在选择cloud provider时，匹配默认answer失败，原因是3.1和3.1.1的iso里，provider list 不一致，而default answer在3.1iso里没有。

将default cloud provider修改为‘Beijing’,在选择acess key ID后failed，原因是different region has a same access key id but Beijing。

将defaut region修改为West California，在进行storage reset时failed，原因是在4步骤后，进程死掉了，重新kill掉该进程，并手动进行了storage reset。

再配置完storage class后failed，error为Unable to add the cloud storage server to the cloud instance 'amazon.com'. Please factory reset the appliance and retry the configuration。但是factory reset之后还是出现一样的问题。原因是在进行storage reset时master上有相应的信息没有清干净。

注：需要搞清楚factory reset和storage reset时master和media的关系。（遗留）

通过修改firefly里的RobotExc.py文件并重新安装firefly，将firefiy run的log实时输入到/tmp/test.log文件里，可以及时查看信息，节约时间。

Media appliance:

vxlogview -p 409 > test:

[Error] [145550] cmd:" /usr/openv/netbackup/bin/admincmd/csconfig cldinstance -as -in amazon.com -sts nbapp639\_amazon -ssl 2 -storage\_class STANDARD\_IA -lc 'us-west-1'"( 1 )

/log/netbackup/nbcssc/csconfig.log:

nbapp639\_amazon already exist.

Master appliance:

csconfig cldinstance -i|less

nbdevquery -h

nbdevquery -liststs

csconfig cldinstance -h

csconfig cldinstance -nbu\_clish -i

csconfig cldinstance -nbu\_clish -at s3

csconfig cldinstance -at s3

csconfig cldinstance -at s3 -help

csconfig cldinstance -at s3 -i

csconfig cldinstance -at s3 -i -nbu\_clish

csconfig cldinstance -at s3 -i -nbu\_clish -j

csconfig cldinstance -at s3 -i -nba\_clish

nbdevquery -liststs

csconfig cldinstance -h

csconfig cldinstance -at s3

csconfig cldinstance -at s3 -h

csconfig cldinstance -at s3 -rs nbapp639\_amazon -h

csconfig cldinstance -at s3 -rs nbapp639\_amazon

csconfig cldinstance -rs -at s3 -h

csconfig cldinstance -rs -in amazon.cn -sts nbapp639\_amazon:删除该文件。

csconfig cldinstance -at s3 -i|less

通过上述命令删除干净之后发现iso莫名变成3.1.1,于是将两个appliance重新装iso。

在设置完hostname之后开始run cases，但是因为没有提前配置master appliance导致失败。

原因是从log上看是因为解析master appliance的IP地址失败。原来appliance在配置为master之后会被添加到一个列表里，case里会调用get\_server\_master\_ip函数去这里面找，如果不事先设置好master当然找不到。不过也可以在case里添加配置master role。

重新配置了之后发现media appliance的iso版本又被升级为3.1.1。可能是有人也在用。

又重装了nbapp666作为media。

Linux command today：

Ps

1. tail

tail –f filename:监视文件的实时输入并显示到屏幕上。按ctrl+c退出。

tail –n 20 filename:显示文件最后20行。

Tail –r –n 10 filename:逆序显示文件最后10行。

类似的还有more（从第一行开始分页显示，空格往后翻，b往前翻）less相反。

2018.1.9 Sunny

1. smoke test 冒烟测试：用来正式集成之前找bug，尤其是对于其他模块的影响情况。

2.upgrad中途aborted后要把.lock文件删掉。路径为/inst/patch/appliance/upgrading/.

3.关于git。

4.sed命令：

修改或者替换关键词的时候，’s/原本的内容/修改后的内容/g’,这里的’/’也可以是’#’

5.vim注释掉多行：ctrl+v上下键选中多行按下i键进入插入模式输入#等想输入的符号按下ESC键

About upgrade：

echo "http://10.182.26.181/artifactory/release/nba/main/3.1.1-20180105095345/post\_release\_3.1.1-20180105103447/" | sed "s/http/https/g" > "/tmp/tmp\_smoke\_https\_url.txt";//将该网址中的http替换成https,然后写入/ tmp/tmp\_smoke\_https\_url.txt文件中。

if grep -q "/usr/openv/tmp/base\_url.txt" /opt/NBUAppliance/scripts/patch\_online.pm;

then cat "/tmp/tmp\_smoke\_https\_url.txt" > /usr/openv/tmp/base\_url.txt; ;//在后面的文件中匹配引号中的路径名，如果有。则将tmp/tmp\_smoke\_https\_url.txt的内容重定向到后面文件中

else if grep -q "/tmp/base\_url.txt" /opt/NBUAppliance/scripts/patch\_online.pm;

then echo "/tmp/tmp\_smoke\_https\_url.txt" > /tmp/base\_url.txt;

else sed -i "s#https.\*TECH#`cat /tmp/tmp\_smoke\_https\_url.txt`#g" /opt/NBUAppliance/scripts/patch\_online.pm;

fi;

fi;

rm /tmp/tmp\_smoke\_https\_url.txt;//移除该文件。

注意：

本地跑firefly的时候要记得先跑tail –f

/home/ben/git\_working\_tree/automation/resource/softwareupdates. softwareupdates\_constants.robot文件在本地进行了修改 不能提交。

http://10.182.26.181/artifactory/release/nba/main/3.1.1-20180105095345/post\_release\_3.1.1-20180105103447/

2018.1.12 Sunny

About add cases for d2c upgrade:

1. git operation
2. vim operation
3. write robot
4. find variable difine
5. 删除后缀相同的文件 命令？？

**之前一直纠结于rm命令后面用正则表达式匹配，但是并不成功。后来才意识到rm后面要用通配符。**

1. Startclish
2. 下周 FC准备

2018.1.14 Sunny

1.在一个文件夹下使用grep查找某个语句时，参数-rin里的r是指在各问文件里进行递归查询。

2.在robot文件了，在调用Robot 库函数和lib文件里的库函数时 都是前缀Library。

3. ENV\_TARGET\_IP 从哪里来？

4.ctrl+l清屏操作

5.scsi有几种不同的target，如scst，lio等。

6.miscunit是啥？

2018.1.15 Cloudy

1静态方法、类方法和实例方法

静态方法和类方法基本相同。他们都是一个类本身的函数，调用的时候可以是类.方法，也可以是实例.方法，也可以是直接调用。而实例方法只能通过实例.方法的方式进行调用，即使是用类本身直接调用也是不可以的。

2.在pycharm里新建一个.py文件并没有设置起始行（#!/usr/bin/env python）也可以正常RUN出最后结果的原因是起始行的作用是是在类UNIX环境下不调用解释器就能运行.py文件。

2018.1.16 Cloudy

scp kernel-firmware-20170530-7.9.1.noarch.rpm suz@10.192.7.71:

sync

reboot -i

cd

lspci -d1077:

dmesg |grep qla2xxx

cd /opt/IMAppliance/scripts/

./fibrechannel\_ui.py configuration show

lspci -d1077：

cd /sys/kernel/scst\_tgt/targets/qla2x00t/

cd /opt/NBUAppliance/xml/

vim fibre-admin.xml

vim fc\_hba\_config.pl

2018.1.17 Cloudy

1.有时间看看/etc/bashrc和/etc/profile！！

2.注意啦！！对于结果直接输出到console的命令（即stdout输出）的输出内容进行正则的时候，如果输出内容是文件名字信息，在使用grep的参数时要注意，如果只是在列出的信息中查找关键字，就不要用‘r’，因为这个参数是在文件里查找。如果要在显示的文件里查找关键字有如下两种方式，举例说明：

第一种：lspci |grep –rin ‘正则表达式’

第二种：lspci |grep –in ‘正则表达式’ \*

3.python的super()函数，可以不绑定子类的父类，在父类更换的时候，只在参数出修改父类的名称就好了。而且保证每个父类只被调用一次。<http://www.jb51.net/article/66912.htm>

4.source命令。修改/etc/profile之后要用source命令运行一遍。

5.读写文件。结合with as语句。

6.字典。在2.x版本里dic.values()或者dic.keys()返回的是一个列表。在3.x版本里返回的是字典视图，不是列表，所以如果想用list的内建函数时需要进行类型转换一下。要注意。

所谓视图就是说，结果不是一次性返回的，可以以迭代的方式依次返回。

2018.1.18 Cloudy

1.在appliance上测试了reset()函数，成功执行。

2.将make之后的RPM包在appliance上安装并测试。

3.

Pycharm小技巧：

1.返回上一个操作的地方：将view里的toolbar打开，就可以有一个向后的按钮了。

遗留：

1.python的logger模块及公司自己的vxlog

2.lspci -m -nn d37..明天来看看怎么回事

3.将code里的print改为log函数实现。

4.pycharm搜索文件路径？是指import的文件的路径都会去哪里找。

2018.1.19 Cloudy

1.了解一下i18n(国际化）与gettext的关系。.po文件与.mo文件的生成和翻译还不太懂。

2.哪个文件修改的环境变量？

3.重点看看SPEC文件和makefile文件，还有ut文件。

2018.1.22 Sunny

1.Target做scan返回True，initiator做scan调用我现在写的scan------reset操作只在target端操作有意义，clean操作只在initiator端有意义。

3.zone命令

4.为什么新建的zone 在之前的target端不覆盖？

5.为什么在67看到了0a的target，但是13上show出来是initiator？因为只有在clish上设置target才会在scan时显示。如果在root模式下手动操作，则不会显示。

看看那个security team

2018.1.29 Sunny

遗留：实现在ui.py执行reset和clean。

Readlink函数 posix.py 关系？



难道虽然模块直接调用了类方法，但是也要跟相应的类有交互？