Jumpserver 是一款由**[Python](http://lib.csdn.net/base/python" \o "Python知识库" \t "_blank)**编写开源的跳板机(堡垒机)系统，实现了跳板机应有的功能。基于ssh协议来管理，客户端无需安装agent。

特点：

完全开源，GPL授权   
Python编写，容易再次开发   
实现了跳板机基本功能，认证、授权、审计   
集成了Ansible，批量命令等   
支持WebTerminal   
Bootstrap编写，界面美观   
自动收集硬件信息   
录像回放   
命令搜索   
实时监控   
批量上传下载

注意：   
在使用jumpserver过程中，有一步是系统用户推送，要推送成功，client（后端服务器）要满足以下条件：

 1）后端服务器需要有python、sudo环境才能使用推送用户，批量命令等功能

2）后端服务器如果开启了selinux，请安装libselinux-python

六、更新代码   
cd /opt/jumpserver   
git pull

一、用户管理

1）添加用户

点击用户管理 —> 用户 —> 创建用户

输入要添加的名称，用户名，邮件，用户组，密码，角色 —> 提交。

查看添加的用户

查看用户邮件   
邮件中包含了用户名，权限，web密码，ssh 密钥密码，以及密钥下载地址。

2）添加用户组

点击用户管理 —> 用户组 —> 创建用户组

名称，用户，备注—> 确认。

二、资产管理

1）创建资产组

点击资产管理 —> 资产组 —> 创建资产组

输入组名称，资产，备注—> 提交。

2）创建资产

点击资产管理 —> 资产 —> 创建资产

输入主机名，IP，端口，集群，公网IP，系统类型，资产环境，管理用户，资产组，标签，备注 —> 提交

3）添加集群

点击资产管理 —> 集群 —> 创建集群

输入名称，地址，联系人，手机，管理用户，系统用户，运营商，内网，外网 —> 提交。

4）添加管理用户

管理用户是服务器上已存在的特权用户，Jumpserver使用该用户来 `推送系统用户`、`获取资产硬件信息`等。可以设置主机级别管理用户，也设置集群级别管理用户，这样资产可以不用再单独设置。Linux服务器上的ROOT用户绑定的是资产物理主机。

点击资产管理 —> 管理用户 —> 创建管理用户

输入名称，用户名，密码，ssh私钥 —> 提交。

5）添加系统用户

系统用户是用户登录资产(服务器)时使用的用户，如 web, sa, dba等具有特殊功能的用户。系统用户创建时，如果选择了自动推送 Jumpserver会使用ansible自动推送到系统用户所在集群的资产中，如果资产(交换机)不支持ansible, 请手动填写账号密码。

点击资产管理 —> 系统用户 —> 创建系统用户

输入名称，用户名，优先级，协议，集群 —> 提交。

ssh-keygen -f admin

cat admin.pub

把此私钥上传到浏览器中。

6）添加标签管理

点击资产管理 —> 标签管理 —> 创建标签

输入名称，值，资—> 提交。

三、权限管理

1）资产授权

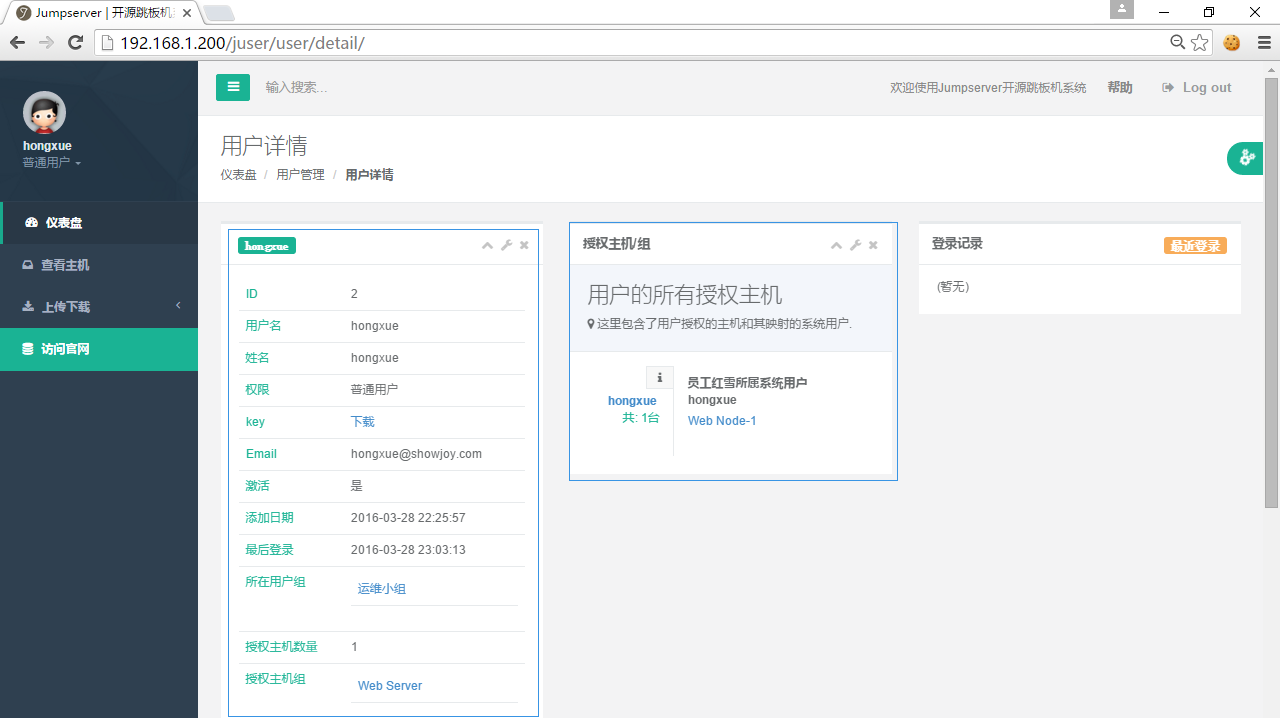
点击权限管理 —> 资产授权 —> 创建授权规则

输入名称，用户/用户组，资产/资产组，系统用户，备注 —> 提交。

四、登录

这时候创建已经完成，下一步用户”hongxue”通过web和密钥登录堡垒机 192.168.1.200，并实现跳转到服务器 192.168.1.210

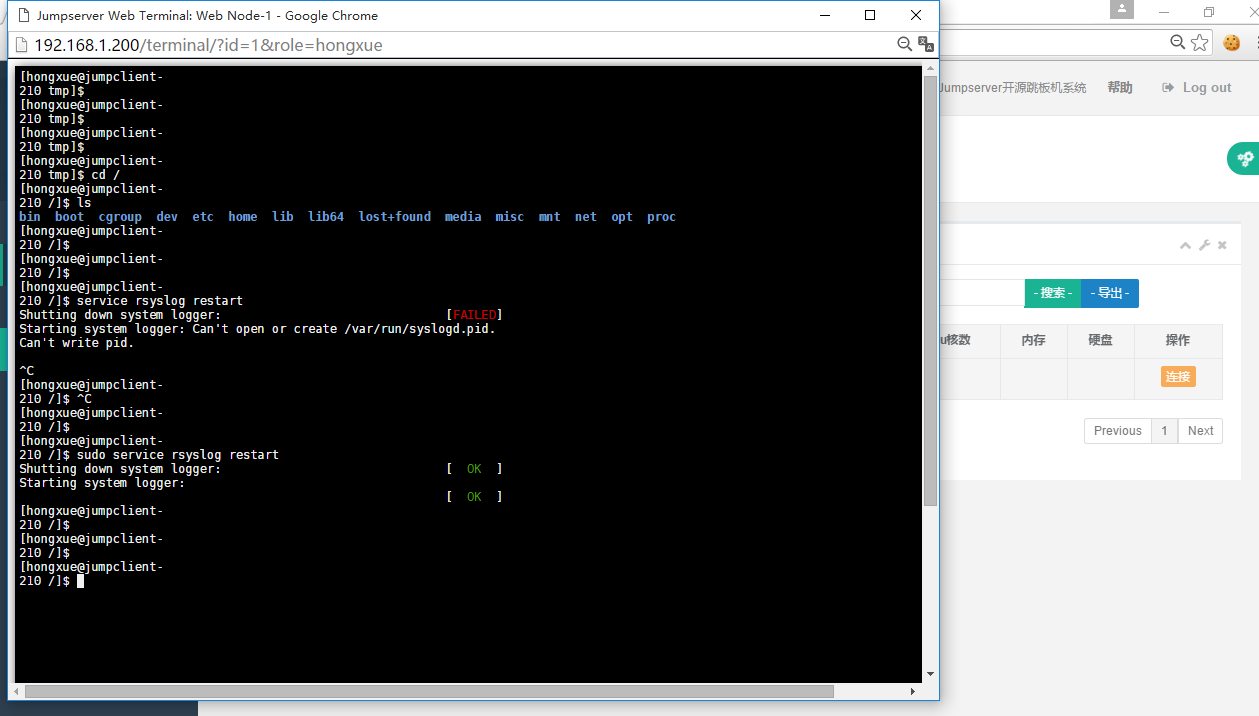
1）通过web登录   
可以看到用户ID，用户名，权限，key，最后登录，用户组，授权主机数，以及主机信息。



单击查看主机 —> 连接



连接成功，可以对该主机进行操作。

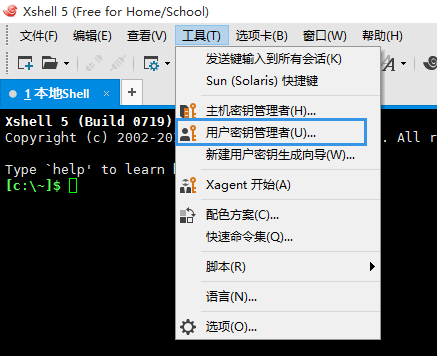


2）通过ssh登录

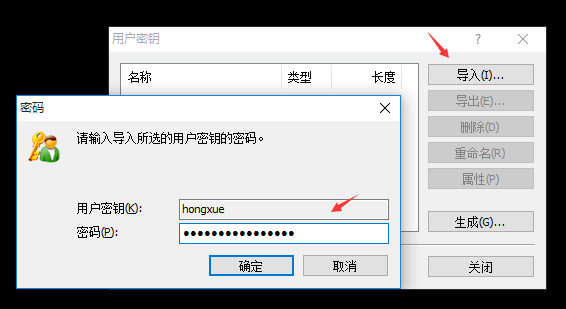
通过邮件中收到的地址，下载key



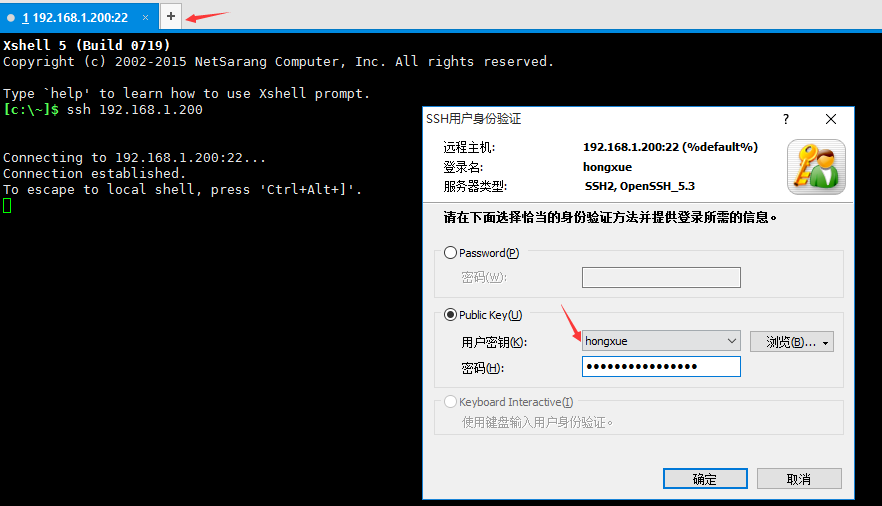
点击工具 —> 用户密钥管理者



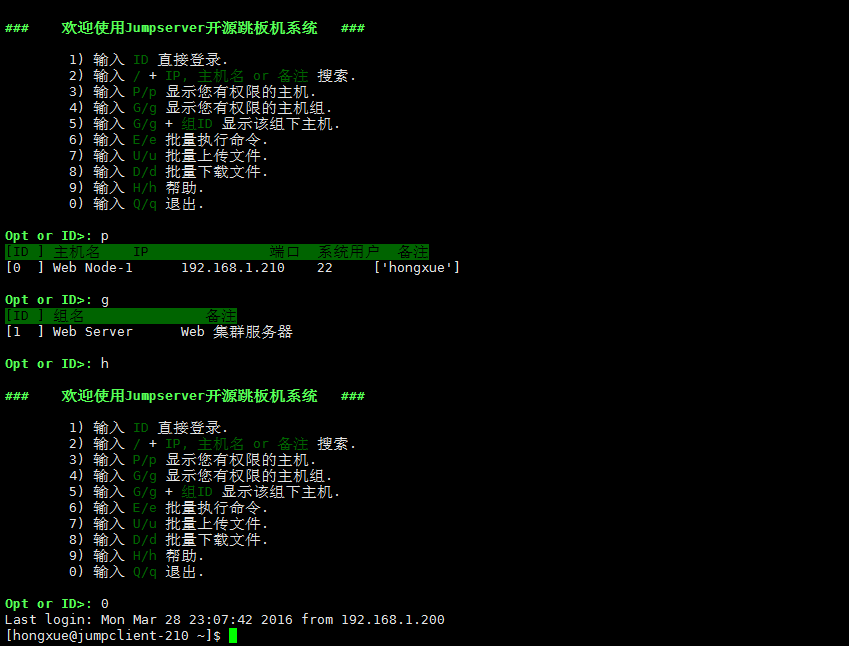
单击导入 —> 输入用户名，密码



连接 jumpserver 堡垒机 192.168.1.200，输入密钥密码进行登录



登录成功！！ 从授权的列表中连接到服务器 192.168.1.210



！！为了安全，建议配置nginx 反向代理 jumpserver

log\_format jumpserver '$remote\_addr - $remote\_user [$time\_local] "$request" $status $body\_bytes\_sent "$request\_time" $request\_body "$http\_referer" "$http\_user\_agent" "$http\_x\_forwarded\_for" $scheme $http\_host';  
  
  
server {  
        listen       8080 ssl;  
        listen       80;  
deny all;  
        server\_name jumpserver.xxxx.com;  
        index index.html index.htm index.php;  
  
        ssl\_certificate      ssl/xxxx.com.crt;  
        ssl\_certificate\_key  ssl/xxxx.com.key;  
ssl\_protocols TLSv1 TLSv1.1 TLSv1.2;  
ssl\_ciphers HIGH:!aNULL:!MD5;  
ssl\_prefer\_server\_ciphers on;  
  
  
if ( $scheme = http ) {  
   rewrite ^(.\*)$ https://$host:8090$request\_uri? permanent;  
  }  
  
location / {  
    proxy\_set\_header Connection "";  
    proxy\_http\_version 1.1;  
    proxy\_pass      http://10.43.12.31:8090;  
 }  
  
location ^~ /ws/ {  
proxy\_pass http://10.43.12.31:8090/ws/;  
proxy\_set\_header X-Real-IP $remote\_addr;  
proxy\_set\_header Host $host;  
proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
proxy\_http\_version 1.1;  
proxy\_set\_header Upgrade $http\_upgrade;  
proxy\_set\_header Connection "upgrade";  
  }  
            
  
          
        access\_log  /data0/logs/jumpserver.log jumpserver;  
        error\_log  /data0/logs/jumpserver\_error.log debug;  
}

一. Jumpserver启动 Python: 版本 3.6

**1.1 启动Jumpserver**

先进入Python虚拟环境

[root@localhost ~]# source /usr/local/python/py3/bin/activate

(py3) [root@localhost ~]# service redis start

(py3) [root@localhost ~]# cd /usr/local/jumpserver/

(py3) [root@localhost jumpserver]# service nginx start

(py3) [root@localhost jumpserver]# python run\_server.py all

**1.2 启动coco**

先进入Python虚拟环境

[root@localhost ~]# source /usr/local/python/py3/bin/activate

(py3) [root@localhost ~]# cd /usr/local/jumpserver/coco/

(py3) [root@localhost coco]# python run\_server.py

2018-01-28 22:06:47 [service DEBUG] Initial app service

2018-01-28 22:06:47 [service DEBUG] Load access key

2018-01-28 22:06:47 [service INFO] No access key found, register it

2018-01-28 22:06:47 [service INFO] "Terminal was not accepted yet"

2018-01-28 22:06:50 [service INFO] "Terminal was not accepted yet"

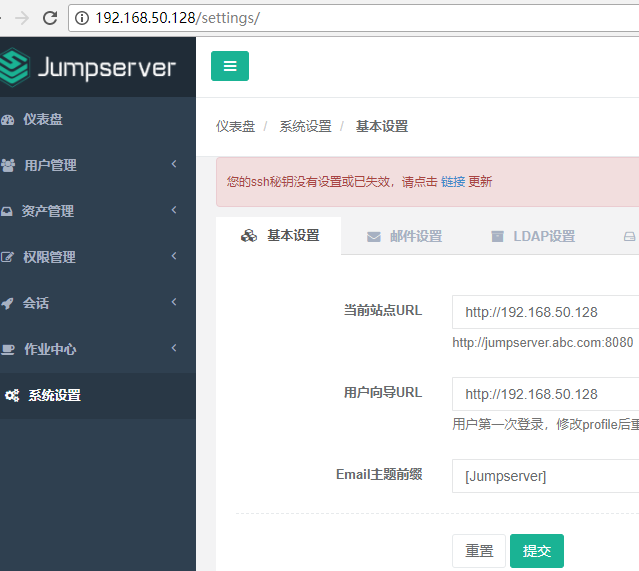
...

**1.3 访问jumpserver管理后台接受coco的注册**

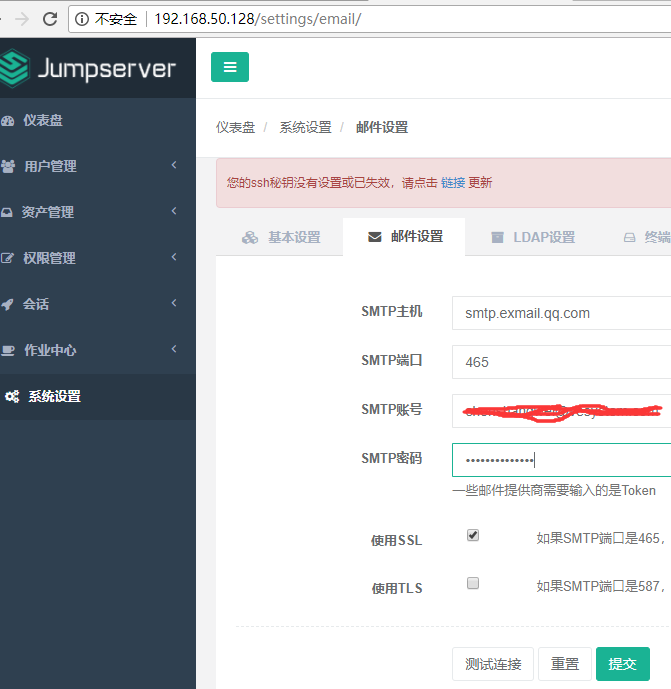
<http://ip> 账号密码： admin admin

**1.4 系统设置**

**1.4.1 基本设置**



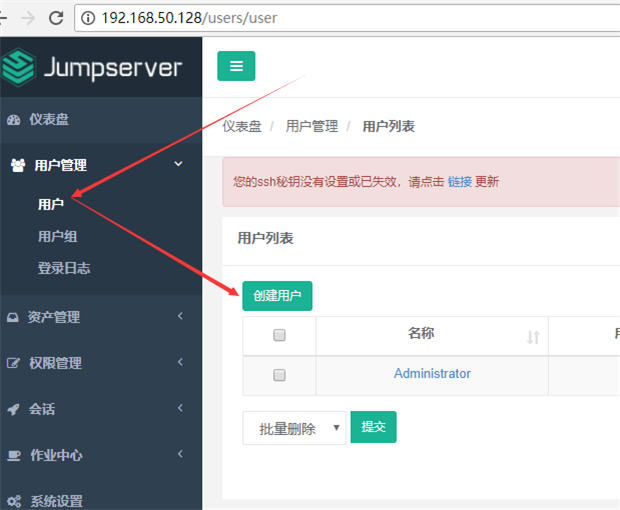
**1.4.2 邮件设置**



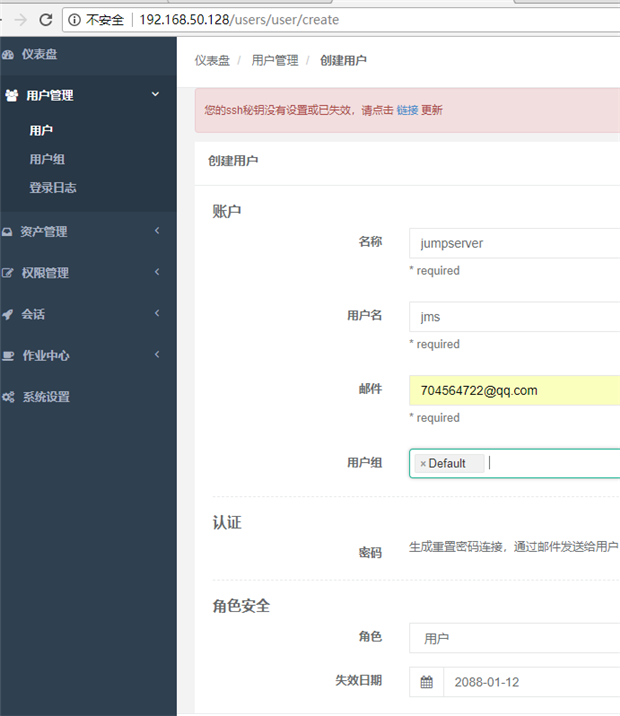
**1.5 创建用户并登录**

创建用户会发送邮件，需要设置密码，登录

**1.5.1. 创建用户**



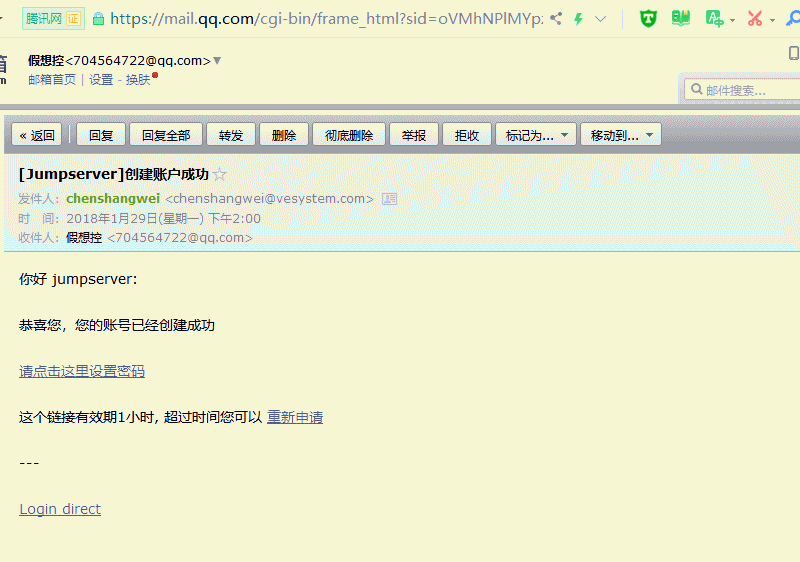
**1.5.2. 创建用户jms**



**1.5.3. 发送邮件创建帐户成功**



**1.5.4. 打开邮件点设置密码 登录用户 jms**



**1.6 创建管理用户**

创建一个管理用户， 创建资产时需要关联

**1.6.1. 创建管理用户**



**1.6.2. 管理用户为root**





**1.7 创建资产**

创建一个资产,关联刚创建的管理用户

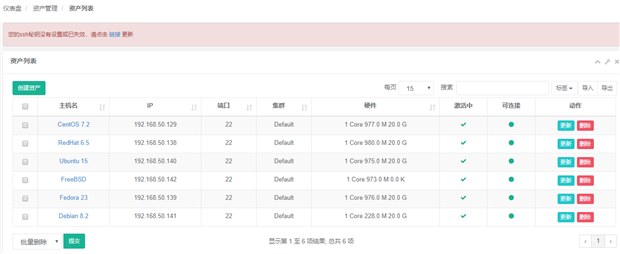
**1.7.1. 创建资产**



**1.7.2. 添加资产并关联管理用户**



**1.7.3. 添加完资产会自动更新获取硬件信息**



**1.8 创建系统用户**

系统用户是用来登录资产的，授权时需要

**1.8.1. 创建系统用户**



**1.8.2. 创建系统用户为Dev**



**1.8.3. 创建完成**



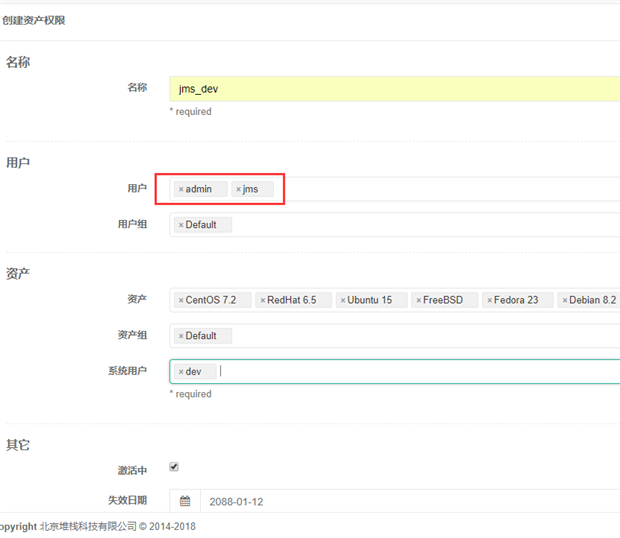
**1.9 创建授权规则**

授权规则 关联用户，资产，系统用户 形成授权规则，授权的系统用户会自动推送到资产上

**1.9.1. 创建授权规则**



**1.9.2. 创建授权规则jms\_Dev**



**1.9.3. 完成创建授权规则**



**1.10 SSH连接终端**

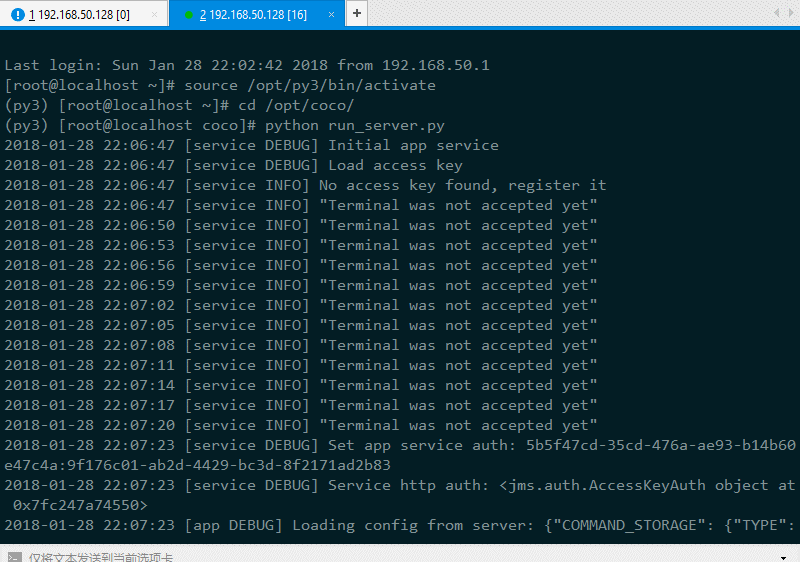
$ ssh -p2222 admin@192.168.244.144

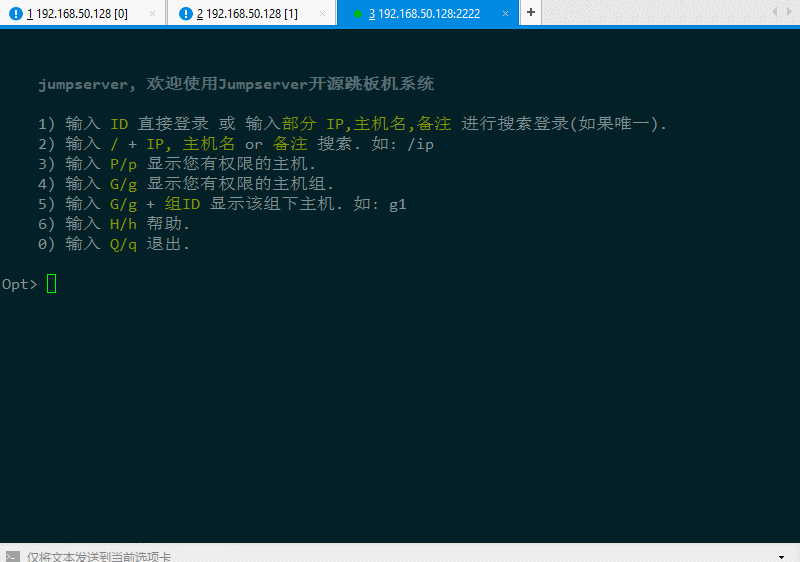
密码: admin

如果是用在windows下，Xshell terminal登录语法如下

$ssh admin@192.168.244.144 2222

密码: admin





**1.11 WEB连接终端**

administrator 先切换到用户界面

