jumpserver安装配置

简介：

Jumpserver是全球首款完全开源的堡垒机，使用 GNU GPL v2.0 开源协议，是符合4A 的专业运维审计系统。官方网站：http://www.jumpserver.org/

Jumpserver 使用 Python /Django 进行开发，遵循 Web 2.0 规范，配备了业界领先的 Web Terminal 解决方案，交互界面美观、用户体验好。

Jumpserver 采纳分布式架构，支持多机房跨区域部署，中心节点提供 API，各机房部署登录节点，可横向扩展、无并发访问限制。

特点：

完全开源，GPL授权

Python编写，方便二次开发

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

组件说明：

- Jumpserver 为管理后台，管理员可以通过Web页面进行资产管理、用户管理、资产授权等操作
- Coco为SSH Server和Web Terminal Server 。用户可以通过使用自己的账户登录SSH或者WebSocket 接口直接访问被授权的资产。不需要知道服务器的账户密码
- Luna为Web Terminal Server 前端页面，用户使用Web Terminal方式登录所需要的组件
- Guacamole为Windows组件，用户可以通过Web Terminal来连接 Windows 资产 （暂时只能通过 Web Terminal 来访问）。Apache 跳板机项目，Jumpserver 使用其组件实现 RDP 功能，Jumpserver 并没有修改其代码而是添加了额外的插件，支持 Jumpserver 调用。

端口说明：

- Jumpserver 默认端口为 8080/tcp 配置文件在 jumpserver/config.py

- Coco 默认SSH 端口为 2222/tcp， 默认Web Terminal 端口为 5000/tcp 配置文件在 coco/conf.py
- Guacamole 默认端口为 8081/tcp 在 docker run 时指定
- Nginx 默认端口为 80/tcp 配置在 nginx/nginx.conf 中指定

Jumpserver提供的堡垒机必备功能			
身份验证 Authentication	登录认证	资源统一登录和认证	✓
		LDAP认证	✓
	多因子认证	MFA（Google Authenticator）	✓
账号管理 Account	集中账号管理	管理用户管理	✓
		系统用户管理	✓
	统一密码管理	资产密码托管	✓
		自动生成密码	✓
		密码自动推送	✓
	批量密码变更	定期批量修改密码	✓
		生成随机密码	✓
授权控制 Authorization	资产授权管理	资产树	✓
		资产或资产组灵活授权	✓
		节点内资产自动继承授权	✓
	多维度授权	可对用户、用户组或系统角色授权	✓
	指令限制	支持限制特权指令使用	✓
	统一文件传输	SFTP 文件上传/下载	✓
安全审计 Audit	会话管理	在线会话管理	✓
		历史会话管理	✓
	录像管理	Linux 录像支持	✓
		Windows 录像支持	✓
	指令审计	指令记录	✓
	文件传输审计	上传/下载记录审计	✓
Jumpserver提供的特色功能			
		完全开源	✓
		无并发及资产数量限制	✓
		分布式架构、灵活扩展	✓
		支持混合云中物理隔离资产的管理	✓
		容器化部署	✓
		体验极佳的Web Terminal	✓
		提供商业支持服务	✓

安装实现：

系统环境：centOS7

内核版本：

[root@jumpserver ~]# uname -r

3.10.0-862.el7.x86_64

关闭selinux和防火墙

修改字符集，否则可能报 input/output error 的问题，因为日志里打印了中文

[root@jumpserver ~]# localedef -c -f UTF-8 -i zh_CN zh_CN.UTF-8

[root@jumpserver ~]# export LC_ALL=zh_CN.UTF-8

[root@jumpserver ~]# echo 'LANG="zh_CN.UTF-8"' > /etc/locale.conf

1、准备 Python3 和 Python 虚拟环境

1.1 安装依赖包

[root@jumpserver ~]# yum -y install wget sqlite-devel xz gccautomake zlib-devel openssl-devel epel-release git

1.2 编译安装　这里必须执行编译安装，否则在安装Python 库依赖时会有问题

[root@jumpserver ~]# wget https://www.python.org/ftp/python/3.6.1/Python-3.6.1.tar.xz

[root@jumpserver ~]#tar xvf Python-3.6.1.tar.xz && cd Python-3.6.1

[root@jumpserver Python-3.6.1]# ./configure && make && make install

1.3 建立 Python 虚拟环境

因为CentOS 6/7 自带的是Python2，而Yum等工具依赖原来的Python，为了不扰乱原来的环境，所以使用Python虚拟环境

[root@jumpserver Python-3.6.1]# cd /opt/

[root@jumpserver opt]# python3 -m venv py3

[root@jumpserver opt]# source /opt/py3/bin/activate

(py3) [root@jumpserver opt]#

看到上面的提示符代表成功，以后运行 Jumpserver 都要先运行以上source命令，以下所有命令均在该虚拟环境中运行。为了防止运行 Jumpserver 时忘记载入Python虚拟环境导致程序无法运行，可以使用 autoenv

(py3)[root@jumpserver ~]# git clone git://github.com/kennethreitz/autoenv.git /opt/autoenv

(py3)[root@jumpserver ~]# echo 'source /opt/autoenv/activate.sh' >> ~/.bashrc

(py3)[root@jumpserver ~]# source ~/.bashrc

2、安装 Jumpserver

2.1 下载或clone项目　　你也可以选择去Github 项目页面直接下载 zip包。

[root@jumpserver autoenv]# cd /opt/

[root@jumpserver opt]# git clone --depth=1 https://github.com/jumpserver/jumpserver.git && cd jumpserver&& git checkout master

为了进入 jumpserver目录时将自动载入python虚拟环境，可以按照以下建立.env文件

[root@jumpserver jumpserver]# echo "source /opt/py3/bin/activate" > /opt/jumpserver/.env

2.2 安装依赖RPM包

[root@jumpserver jumpserver]# cd /opt/jumpserver/requirements/

autoenv:

autoenv: WARNING:

autoenv: This is the first time you are about to source /opt/jumpserver/.env:

autoenv:

autoenv: --- (begin contents) ---------------------------------------

autoenv: source /opt/py3/bin/activate$

autoenv:

autoenv: --- (end contents) ---

autoenv:

autoenv: Are you sure you want to allow this? (y/N) y　　#回答YES

(py3) [root@jumpserver requirements]# yum install -y `cat rpm_requirements.txt`

2.3 安装Python库依赖

(py3) [root@jumpserver requirements]# pip install -r requirements.txt

2.4 安装Redis, Jumpserver使用redis做cache和celery broker（celery是python做异步通信的工具包,celery使用了非常简单的方式实现了异步。主要是celery sender broker 到consumer的过程)

(py3) [root@jumpserver requirements]# yum install redis -y

(py3) [root@jumpserver requirements]# systemctl enable redis.service

(py3) [root@jumpserver requirements]# systemctl start redis.service

2.5 安装mariadb数据库

(py3) [root@jumpserver requirements]# yum -y install mariadb mariadb-devel mariadb-server

(py3) [root@jumpserver requirements]# systemctl enable mariadb.service

(py3) [root@jumpserver requirements]# systemctl start mariadb.service

2.6 创建数据库Jumpserver并授权

设置安全数据库访问，设置root密码为123.com

(py3) [root@jumpserver requirements]# mysql_secure_installation

连接数据库：

(py3) [root@jumpserver requirements]# mysql -u root -p123.com

MariaDB [(none)]> create database jumpserver default character set 'utf8' collate utf8_general_ci;

MariaDB [(none)]> grant all on jumpserver.* to 'jumpserver'@'127.0.0.1' identified by '123.com';

MariaDB [(none)]> grant all on jumpserver.* to 'jumpserver'@'%' identified by '123.com';

MariaDB [(none)]> grant all on jumpserver.* to 'jumpserver'@'192.168.154.%' identified by '123.com';

2.7 修改jumpserver配置文件

(py3) [root@jumpserver requirements]# cd /opt/jumpserver/

(py3) [root@jumpserver jumpserver]# cp config_example.py config.py

(py3) [root@jumpserver jumpserver]# vim config.py

注意: 配置文件是Python格式，不要用TAB，而要用空格

修改DevelopmentConfig中的配置，因为默认Jumpserver使用该配置，它继承自Config，配置内容根据实际情况进行修改。

```
class DevelopmentConfig(Config):
    DEBUG = True
    DB_ENGINE = 'mysql'
    DB_HOST = '127.0.0.1'
    DB_PORT = 3306
    DB_USER = 'jumpserver'
    DB_PASSWORD = '123.com'
    DB_NAME = 'jumpserver'
    REDIS_HOST = '127.0.0.1'
    REDIS_PORT = 6379
    REDIS_PASSWORD = ''
    BROKER_URL = 'redis://%(password)s%(host)s:%(port)s/3' % {
                'password': REDIS_PASSWORD,
                'host': REDIS_HOST,
                'port': REDIS_PORT,
    }


# Default using Config settings, you can write if/else for different env
config = DevelopmentConfig()
```

2.8 生成数据库表结构和初始化数据

(py3) [root@jumpserver jumpserver]# cd /opt/jumpserver/utils/

(py3) [root@jumpserver utils]# bash make_migrations.sh

2.9 运行 Jumpserver

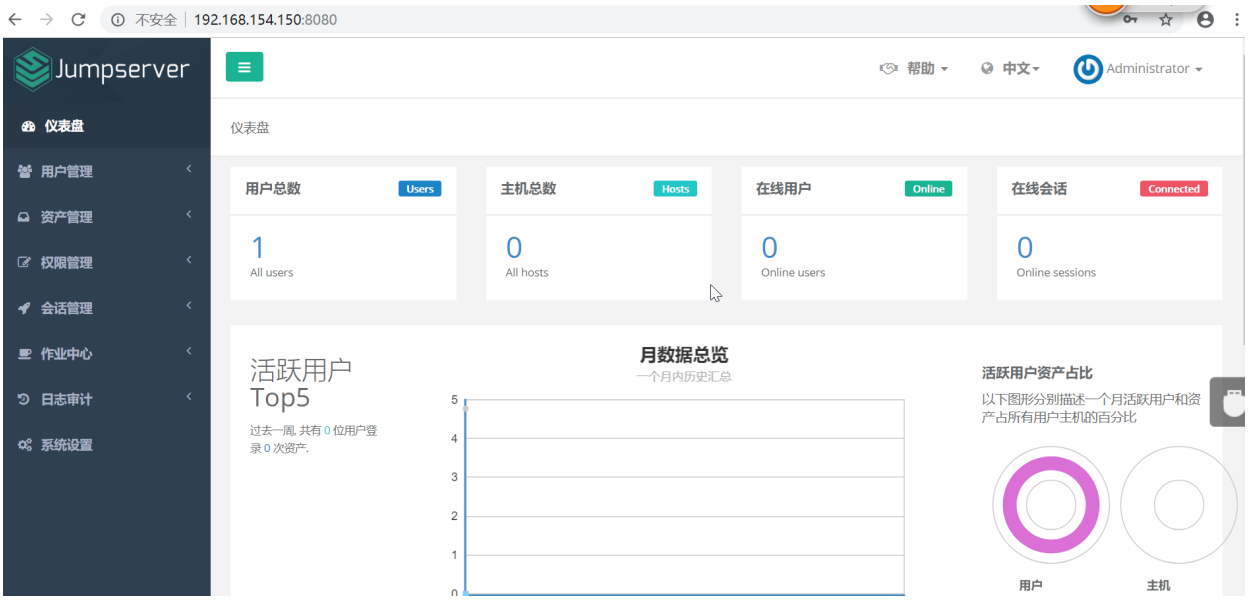
(py3) [root@jumpserver utils]# cd /opt/jumpserver/

(py3) [root@jumpserver jumpserver]# ./jms start all

后台运行使用 -d 参数./jms start all -d

重启命令：./jms restart

访问：运行不报错，请浏览器访问 http://192.168.154.150:8080/ 默认账号: admin 密码: admin 页面显示不正常先不用处理，搭建 nginx 代理就可以正常访问了



3、安装 SSH Server 和 Web Socket Server:Coco

3.1 下载或 Clone 项目

可以在jumpserver服务器上进行安装，也可以分开，需要source /opt/py3/bin/activate

[root@jumpserver ~]# cd /opt/

[root@jumpserver opt]# source /opt/py3/bin/activate

(py3) [root@jumpserver opt]# git clone https://github.com/jumpserver/coco.git && cd coco && git checkout master

```
(py3) [root@jumpserver opt]# git clone https://github.com/jumpserver/coco.git && cd coco && git checkout master
正克隆到 'coco'...
remote: Enumerating objects: 75, done.
remote: Counting objects: 100% (75/75), done.
remote: Compressing objects: 100% (52/52), done.
remote: Total 2203 (delta 38), reused 47 (delta 22), pack-reused 2128
接收对象中: 100% (2203/2203), 1.43 MiB | 496.00 KiB/s, done.
处理 delta 中: 100% (1460/1460), done.
已经位于 'master'
(py3) [root@jumpserver coco]# pwd
/opt/coco
```

为了进入coco目录时将自动载入python虚拟环境，创建.env文件

(py3) [root@jumpserver coco]# echo "source /opt/py3/bin/activate" > /opt/coco/.env

3.2 安装依赖包

(py3) [root@jumpserver coco]# cd /opt/coco/requirements/

(py3) [root@jumpserver requirements]# yum -y install $(cat rpm_requirements.txt)

(py3) [root@jumpserver requirements]# pip install -r requirements.txt -i https://pypi.org/simple

3.3 生成配置文件并运行

(py3) [root@jumpserver requirements]# cd /opt/coco/

(py3) [root@jumpserver coco]# cp conf_example.py conf.py

(py3) [root@jumpserver coco]# ./cocod start -d #后台启动

启动成功后去Jumpserver会话管理-终端管理接受coco的注册，如果页面不正常可以等部署完成后再处理。

4、安装Web Terminal前端: Luna

Luna已改为纯前端，需要 Nginx 来运行访问，访问（https://github.com/jumpserver/luna/releases）下载对应版本的release包，直接解

压，不需要编译。

(py3) [root@jumpserver opt]# pwd

/opt

(py3) [root@jumpserver opt]# wget

https://github.com/jumpserver/luna/releases/download/1.4.3/luna.tar.gz

(py3) [root@jumpserver opt]# tar xvf luna.tar.gz

(py3) [root@jumpserver opt]# chown -R root:root luna

5、安装Windows支持组件

如果不需要管理 windows资产，可以直接跳过这一步。因为手动安装guacamole [ˌgwækəˈmoʊleɪ]组件比较复杂，这里使用打包好的docker镜像部署启动guacamole

5.1 安装docker环境

```
(py3) [root@jumpserver opt]#yum remove docker \
                docker-client \
                docker-client-latest \
                docker-common \
                docker-latest \
                docker-latest-logrotate \
                docker-logrotate \
                docker-selinux \
                docker-engine-selinux \
                docker-engine
(py3) [root@jumpserver opt]#yum install -y yum-utils  device-mapper-persistent-data  lvm2
(py3) [root@jumpserver opt]# yum-config-manager \
    --add-repo \
    https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
(py3) [root@jumpserver opt]# yum install docker-ce -y
(py3) [root@jumpserver opt]# systemctl enable docker.service
(py3) [root@jumpserver opt]# systemctl start docker.service
(py3) [root@jumpserver opt]# systemctl status docker.service
```

5.2 启动 Guacamole

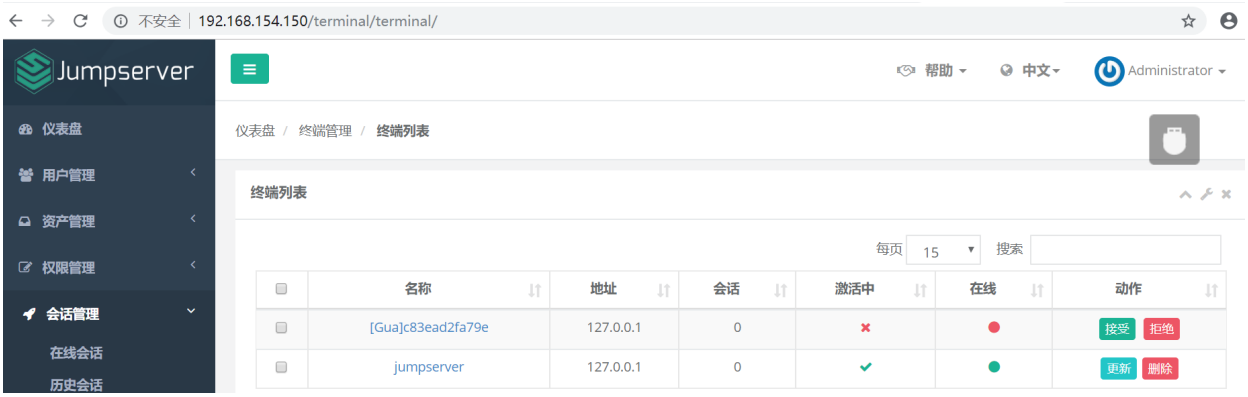
拉取docker镜像，也可以使用registry.jumpserver.org/public/guacamole:latest这个镜像

```
(py3) [root@jumpserver opt]# docker pull jumpserver/guacamole
```

这里所需要注意的是 guacamole暴露出来的端口是8081，若与主机上其他端口冲突请自定义

修改，JUMPSERVER_SERVER 环境变量的配置，填上Jumpserver 的内网地址, 启动成功后去 Jumpserver 会话管理-终端管理接受[Gua]开头的一个注册，如果页面显示不正常可以等部署完成后再处理。注意：这里一定要改写一下本机的IP地址, 不能使用 127.0.0.1，否则会出错。

```
(py3) [root@jumpserver opt]# docker run --name jms_guacamole -d \
> -p 8081:8080 -v /opt/guacamole/key:/config/guacamole/key \
> -e JUMPSERVER_KEY_DIR=/config/guacamole/key \
> -e JUMPSERVER_SERVER=http://192.168.154.150 \
> --restart always  jumpserver/guacamole:latest
(py3) [root@jumpserver opt]# docker ps
CONTAINER ID        IMAGE                          COMMAND             CREATED
STATUS              PORTS                    NAMES
c83ead2fa79e        jumpserver/guacamole:latest    "/init"             7 seconds ago
Up 6 seconds        0.0.0.0:8081->8080/tcp   jms_guacamole
(py3) [root@jumpserver opt]#
```



6、配置 Nginx 整合各组件

6.1 安装nginx

```
(py3) [root@jumpserver key]# yum install nginx -y
```

6.2 修改nginx主配置文件，主要是修改server块区域

```
http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 2048;
    client_max_body_size 100m;  #录像上传大小限制
```

代理到jumpserver的访问URL

```
server {
    listen       80 default_server;
    listen       [::]:80 default_server;
    server_name  _;
    root         /usr/share/nginx/html;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
        proxy_pass http://192.168.154.150:8080; #代理到jumpserver的访问URL
    }
```

luna配置

```
location /luna/ {
    try_files $uri / /index.html;
    alias /opt/luna/;  # luna路径，要按照实际安装路径来添加
}
```

录像位置：

```
 location /media/ {
     add_header Content-Encoding gzip;
     root /opt/jumpserver/data/;  #录像位置，如果修改安装目录，此处需要修改
 }
```

静态资源：

```
location /static/ {
    root /opt/jumpserver/data/; # 静态资源，如果修改安装目录，此处需要修改
}
```

代理coco配置

```
location /socket.io/ {
      proxy_pass http://192.168.154.150:5000/socket.io/; # coco安装服务器IP地址
      proxy_buffering off;
      proxy_http_version 1.1;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection "upgrade";
      proxy_set_header X-Real-IP $remote_addr;
      proxy_set_header Host $host;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      access_log off;
  }
```

代理guacamole配置

```
  location /guacamole/ {
        proxy_pass http://192.168.154.150:8081/; # guacamole服务器访问地址
        proxy_buffering off;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $http_connection;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        access_log off;
  }
```

6.3 启动nginx服务

(py3) [root@jumpserver ~]# systemctl enable nginx

(py3) [root@jumpserver ~]# systemctl start nginx

nginx配置文件：（server 块区域）

```
 location /luna/ {
        try_files $uri / /index.html;
        alias /opt/luna/;  # luna路径，要按照实际安装路径来添加
    }
    location /media/ {
        add_header Content-Encoding gzip;
        root /opt/jumpserver/data/;  #录像位置，如果修改安装目录，此处需要修改
    }
    location /static/ {
        root /opt/jumpserver/data/; # 静态资源，如果修改安装目录，此处需要修改
     }
  location /socket.io/ {
        proxy_pass http://192.168.154.150:5000/socket.io/; # coco安装服务器IP地址
        proxy_buffering off;
        proxy_http_version 1.1;
```

```
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        access_log off;
     }
    location /guacamole/ {
          proxy_pass http://192.168.154.150:8081/; # guacamole服务器访问地址
          proxy_buffering off;
          proxy_http_version 1.1;
          proxy_set_header Upgrade $http_upgrade;
          proxy_set_header Connection $http_connection;
          proxy_set_header X-Real-IP $remote_addr;
          proxy_set_header Host $host;
          proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
          access_log off;
    }
  location / {
        proxy_pass http://192.168.154.150:8080; #代理到jumpserver的访问URL
    }
```

6.4 防护墙要放行的端口

```
$ firewall-cmd --zone=public  --add-port=8080/tcp --permanent  #jumpserver 端口
$ firewall-cmd --zone=public  --add-port=80/tcp --permanent  # nginx 端口
$ firewall-cmd --zone=public  --add-port=2222/tcp --permanent  # 用户SSH登录端口 coco
$ firewall-cmd --zone=public  --add-port=5000/tcp --permanent  # 用户HTTP/WS登录端口 coco
$ firewall-cmd --zone=public  --add-port=8081/tcp --permanent  #guacamole端口 docker
$ firewall-cmd –reload  # 重载防火墙配置生效
```

7、测试连接

```
# 如果登录客户端是 macOS 或 Linux， 登录语法如下
```

$ ssh -p2222 admin@192.168.154.150

$ sftp -P2222 admin@192.168.154.150

密码: admin

如果登录客户端是 Windows， XshellTerminal 登录语法如下

$ ssh admin@192.168.154.150 2222

$ sftp admin@192.168.154.150 2222

密码: admin

如果能登陆代表部署成功

```
                Administrator, 欢迎使用Jumpserver开源跳板机系统

        1) 输入 ID 直接登录 或 输入部分 IP,主机名,备注 进行搜索登录(如果唯一).
        2) 输入 / + IP, 主机名 or 备注 搜索. 如: /ip
        3) 输入 p 显示您有权限的主机.
        4) 输入 g 显示您有权限的节点.
        5) 输入 g + 组ID 显示节点下主机. 如: g1
        6) 输入 s 中/英文切换.
        7) 输入 h 帮助.
        0) 输入 q 退出.

Opt> p
```

sftp默认上传的位置在资产的 /tmp 目录下

基本应用图解：

官方文档中心：http://docs.jumpserver.org/zh/docs/

1、修改管理员个人信息：

2、用户管理

2.1 添加用户组

用户组： 为了方便进行授权,可以将多个用户组成用户组。

在授权中使用组授权,那么该组中的用户就拥有所有授权的主机权限。

创建两个用户组：network-devices和web-site

2.2. 添加用户

用户： 用户是授权和登陆的主体,管理员为每个员工创建一个用户帐号用来登录跳板机。另外用户分为普通用户和超级管理员，后者可以审计查看用户登陆记录、命令历史等。

2.2.1 创建普通用户web和net，分别加入web-site组和network-devices组

2.2.2创建用户后，需要更新密码：

3、资产管理

3.1 创建资产树：

3.2 添加管理用户

管理用户是服务器的 root，或拥有 NOPASSWD: ALL sudo 权限的用户，Jumpserver 使用该用户来推送系统用户、获取资产硬件信息等。

名称可以按资产树来命名。用户名 root。密码和 SSH 私钥必填一个。

3.3 添加资产

点击页面左侧的“资产管理”菜单下的“资产列表”按钮，查看当前所有的资产列表。
点击页面左上角的“创建资产”按钮，进入资产创建页面，填写资产信息。
IP 地址和管理用户要确保正确，确保所选的管理用户的用户名和密码能"牢靠"地登录指定的 IP 主机上。资产的系统平台也务必正确填写。公网 IP 信息只用于展示，可不填，Jumpserver 连接资产主机使用的是 IP 信息。

3.4 添加系统用户

系统用户是 Jumpserver 跳转登录资产时使用的用户，可以理解为登录资产用户，如 web, sa, dba(ssh web@some-host), 而不是使用某个用户的用户名跳转登录服务器（ssh xiaoming@some-host）；简单来说是 用户使用自己的用户名登录 Jumpserver, Jumpserver 使用系统用户登录资产。
系统用户创建时，如果选择了自动推送 Jumpserver 会使用 Ansible 自动推送系统用户到资产中，如果资产（交换机、Windows）不支持 Ansible, 请手动填写账号密码。
Linux 系统协议项务必选择 ssh 。如果用户在系统中已存在，请去掉自动生成密钥、自动推送勾选。

4、权限管理 -------资产授权

创建授权规则

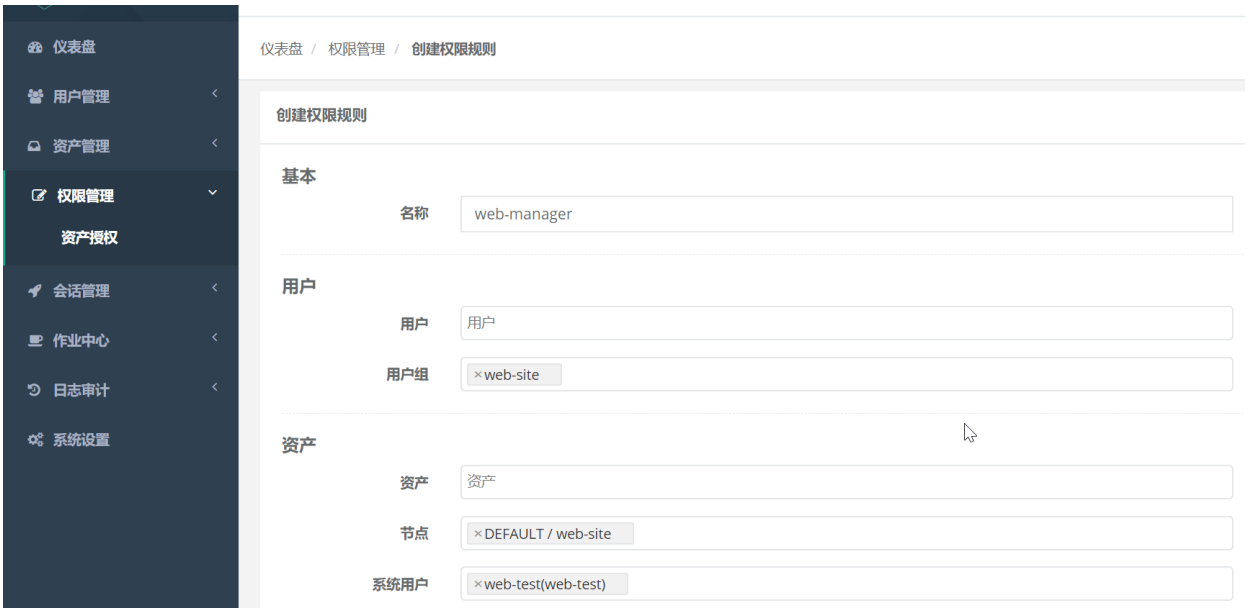
节点，对应的是资产，代表该节点下的所有资产。

用户组，对应的是用户，代表该用户组下所有的用户。

系统用户，及所选的用户组下的用户能通过该系统用户使用所选节点下的资产。

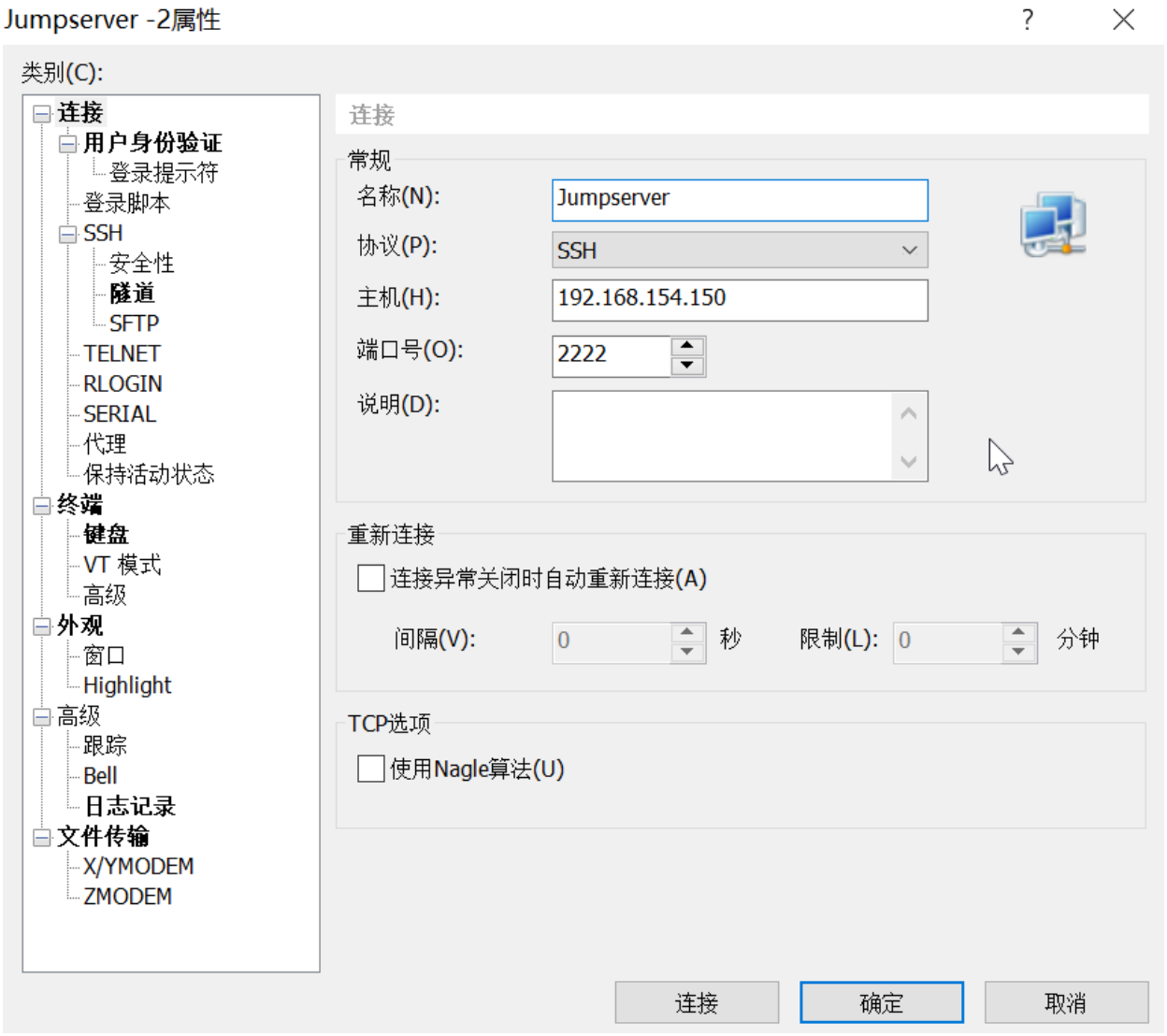
节点，用户组，系统用户是一对一的关系，所以当拥有 Linux、Windows 不同类型资产时，应该分别给 Linux 资产和 Windows 资产创建授权规则。

资产授权与节点授权的区别请参考下面示例，一般情况下，资产授权给个人，节点授权给用户组，一个授权只能选择一个系统用户

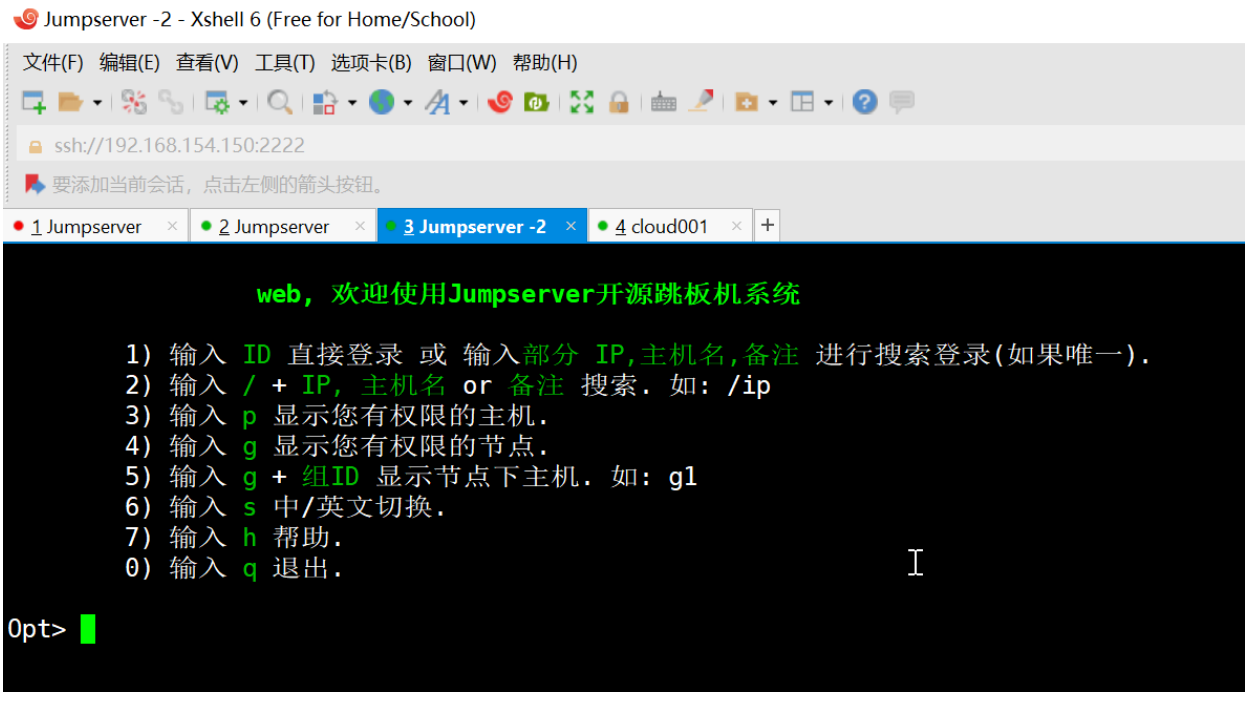


5、测试管理服务器：

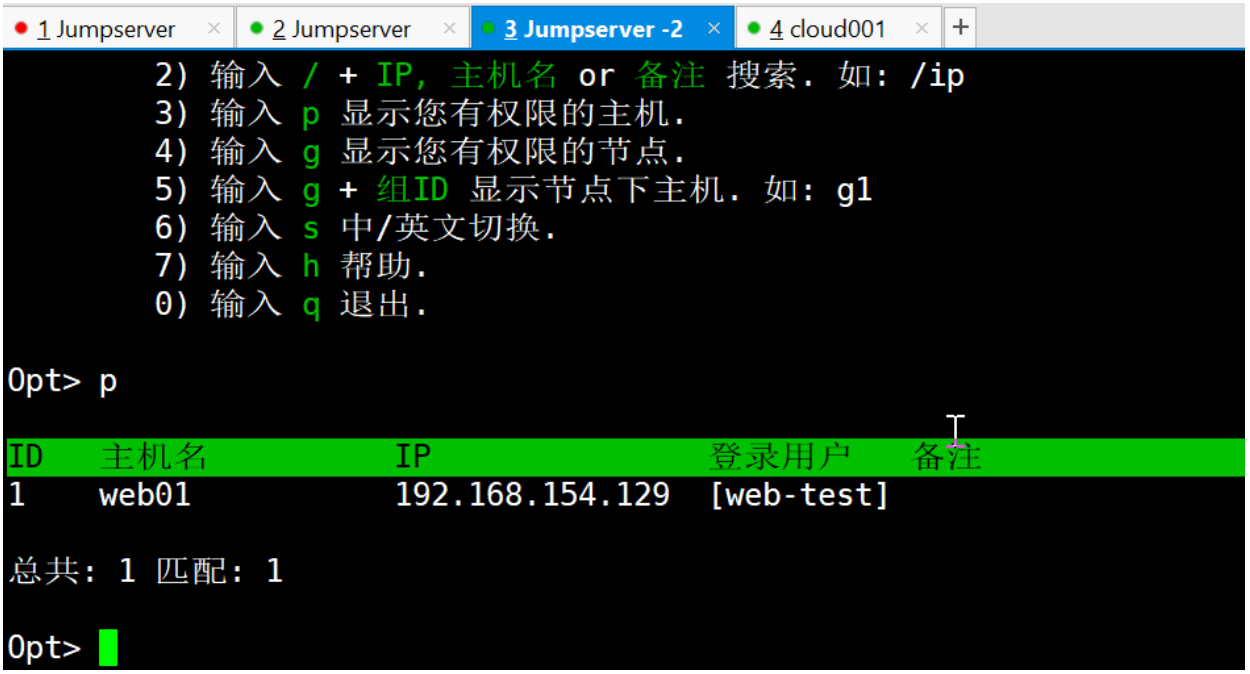
5.1 使用web用户ssh登录jumpserver跳板机，设置Xshell连接信息



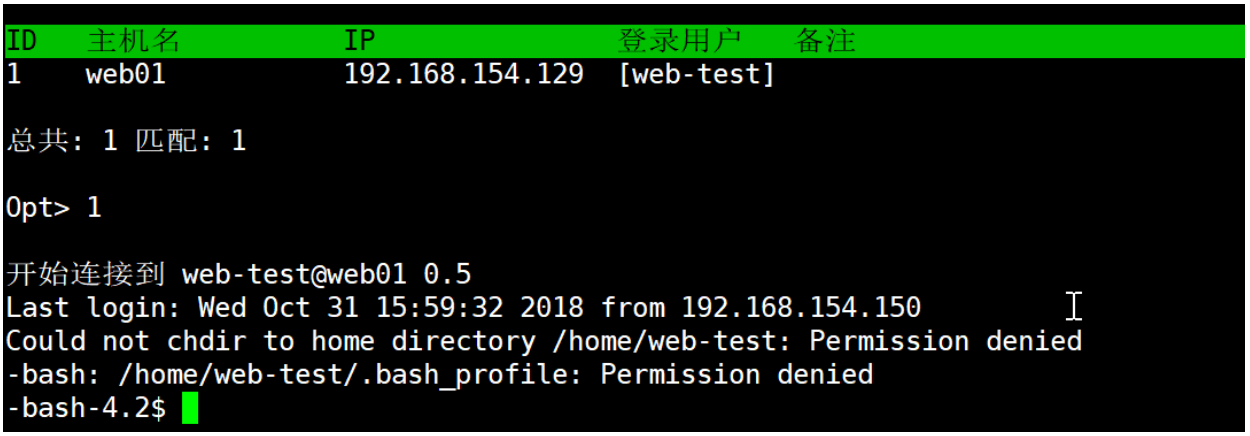
使用web用户登录成功后：



安装提示输入p



然后输入ID号1，显示登录成功



5.2 使用web页面登录也可以管理：

点击右边“web终端”，然后选择有权限的资产，也可以连接上去：

备注：

如何忘记了管理用户admin的密码，可以这样来修改：

[root@jumpserver ~]# cd /opt/jumpserver/apps/

(py3) [root@jumpserver apps]# python manage.py changepassword admin

也可以创建新管理用户：

(py3) [root@jumpserver apps]# python manage.py createsuperuser --username=admin2 --email=2345@qq.com