第5天-计划任务/远程管理

一、计划任务

1、计划任务概念解析

> 在Linux操作系统中，除了用户即时执行的命令操作以外，还可以配置在指定的时间、指定的日期执行预先计划好的系统管理任务（如定期备份、定期采集监测数据）。RHEL6系统中默认已安装了at、cronie软件包，通过atd和crond这两个系统服务实现一次性、周期性计划任务的功能，并分别通过at、crontab命令进行计划任务设置。

2、计划任务分类

> 一次性计划任务
>
> 循环型计划任务
>
> 系统级计划任务
>
> 用户级计划任务

3、Crontab详解

```
安装软件
    [root@qfedu.com ~]#  yum -y install crontabs

启动服务
    rhel5/6:
    [root@qfedu.com ~]#  /etc/init.d/crond status
    [root@qfedu.com ~]#  /etc/init.d/crond start

    rhel7:
    [root@qfedu.com ~]#  systemctl start crond.service
    [root@qfedu.com ~]#  systemctl status crond.service
    [root@qfedu.com ~]#  systemctl enable crond.service

    开机启动(rhel5/6)
        [root@qfedu.com ~]# chkconfig crond on

创建计划任务：用户级别的计划任务
[root@qfedu.com ~]# crontab  -u  用户   -e
-u 指定用户     默认不写就是root

[root@qfedu.com ~]# crontab -e
配置分两部分 拿空格分开
第一部分:时间
         分钟  小时      日          月            周
    范围  0-59 0-23       1-31        1-12        0-7
上面的时间范围可以查看man手册:  [root@qfedu.com ~]# man 5 crontab

各种时间写法:
     5                  10   *    *    *
     5                  10   8    *    *
     1                  5   7   *   5
     1,5,9          *   *   *   *
     8-12               *   *   *   *
     5-20,40    *   *   *   *
     8-12,20-25 *   *   *   *
     */5            *   *   *   *

ps:    *        表示每...
         ,      取不同的时间点
         -      表示范围
         */5    每5分钟

第二部分:动作
     把上面规定的时间要执行的命令写在这里，当然包括脚本(最常用)，命令最好要写绝对路径

查看计划任务:两种方法
    1)[root@qfedu.com ~]# crontab -l
      -u 用户名     查看某一个账户的计划任务
    2)[root@qfedu.com ~]# cat /var/spool/cron/root

计划任务删除:两种方法
    1)[root@qfedu.com ~]#  crontab -r -u wing
        -r 删除
        -u 指定用户
        [root@qfedu.com ~]# crontab -e -u tom
    2)[root@qfedu.com ~]#  rm -f /var/spool/cron/root

计划任务的权限控制
    [root@qfedu.com ~]# cat /etc/cron.deny
    如果这个文件存在，凡是写到这个文件里面的账户不允许执行crontab命令
    [root@qfedu.com ~]# cat /etc/cron.allow
    如果这个文件存在，没有写到这个文件里面的账户不允许执行crontab命令
    如果有allow文件，那不管deny是否存在，都是只允许allow文件里面的用户
```

二、计划任务实战

使用计划任务运行指定应用程序

三、远程管理

1、ssh服务

```
安装软件：
    openssh-server    提供服务
    openssh-clients    客户端
    openssh

    [root@qfedu.com ~]# yum install openssh*  -y

ssh  端口22
服务器端：
    启动服务：
    [root@qfedu.com ~]# systemctl start sshd

    查看：
     [root@qfedu.com ~]#  lsof -i:22

    关闭防火墙和selinux

客户端：
    远程登陆管理：
        [root@qfedu.com ~]#  ssh  -X  tom@10.18.44.208  -p 2222
        [root@qfedu.com ~]#  ssh  10.18.44.208
         如登陆果账户没有密码，默认不能

无密码登陆(ssh密钥认证)
client:
    产生公钥和私钥：
    [root@qfedu.com ~]#  ssh-keygen        //一路回车
    拷贝公钥给对方：
    [root@qfedu.com ~]#  ssh-copy-id  -i    10.18.44.208

直接执行远程命令：
    [root@qfedu.com ~]# ssh  10.18.44.208  "reboot"

远程拷贝：
    需要先安装客户端
    [root@qfedu.com ~]#  cp   源文件   目标路径

     谁是远程谁加IP
    [root@qfedu.com ~]# scp    /a.txt     192.168.2.108:/
    [root@qfedu.com ~]# scp   192.168.2.108:/a.txt     ./
     -P端口
     拷贝目录加-r选项

     [root@qfedu.com ~]# scp   192.168.2.108:/a.txt     192.168.2.109:/

修改端口号
   [root@qfedu.com ~]#  vim /etc/ssh/sshd_config

    Port 22
    ListenAddress 192.168.2.8
    PermitRootLogin yes
    MaxSessions 10        最大并发量
    PermitEmptyPasswords no
```

2、rz sz命令

```
安装
root 账号登陆后执行以下命令：
 [root@qfedu.com ~]#yum install -y lrzsz

使用
sz命令发送文件到本地：
 [root@qfedu.com ~]# sz filename
rz命令本地上传文件到服务器：
 [root@qfedu.com ~]## rz
执行该命令后，在弹出框中选择要上传的文件即可。
```

3、远程桌面管理

4、Jumpserver

四、Jumpserver详解

1、环境配置

```
[root@sdp-dev ~]# getenforce
Disabled
[root@sdp-dev ~]# systemctl stop firewalld.service
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@sdp-dev ~]# localedef -c -f UTF-8 -i zh_CN zh_CN.UTF-8
[root@sdp-dev ~]# export LC_ALL=zh_CN.UTF-8
[root@sdp-dev ~]# echo 'LANG="zh_CN.UTF-8"' > /etc/locale.conf
[root@sdp-dev ~]# yum -y install wget sqlite-devel xz gcc automake zlib-devel
openssl-devel epel-release git

[root@sdp-dev ~]# wget https://www.python.org/ftp/python/3.6.1/Python-
3.6.1.tar.xz
[root@sdp-dev ~]# tar xf Python-3.6.1.tar.xz  && cd Python-3.6.1
[root@sdp-dev Python-3.6.1]#  ./configure && make && make install
这里必须执行编译安装，否则在安装 Python 库依赖时会有麻烦...

[root@sdp-dev Python-3.6.1]# cd /opt
[root@sdp-dev opt]# python3 -m venv py3
[root@sdp-dev opt]# source /opt/py3/bin/activate
(py3) [root@sdp-dev opt]#

(py3) [root@sdp-dev opt]# git clone git://github.com/kennethreitz/autoenv.git
正克隆到 'autoenv'...
remote: Enumerating objects: 671, done.
remote: Total 671 (delta 0), reused 0 (delta 0), pack-reused 671
接收对象中: 100% (671/671), 103.92 KiB | 115.00 KiB/s, done.
处理 delta 中: 100% (356/356), done.
(py3) [root@sdp-dev opt]#
(py3) [root@sdp-dev opt]# echo 'source /opt/autoenv/activate.sh' >> ~/.bashrc
(py3) [root@sdp-dev opt]# source ~/.bashrc
(py3) [root@sdp-dev opt]#
```

2、下载Jumpserver

```
(py3) [root@sdp-dev opt]# git clone https://github.com/jumpserver/jumpserver.git
&& cd jumpserver && git checkout master
正克隆到 'jumpserver'...
remote: Enumerating objects: 79, done.
remote: Counting objects: 100% (79/79), done.
remote: Compressing objects: 100% (68/68), done.
remote: Total 41282 (delta 19), reused 20 (delta 5), pack-reused 41203
接收对象中: 100% (41282/41282), 52.05 MiB | 79.00 KiB/s, done.
处理 delta 中: 100% (28176/28176), done.
已经位于 'master'
(py3) [root@sdp-dev jumpserver]#
```

3、安装所需的python modules

```
(py3) [root@sdp-dev jumpserver]#  echo "source /opt/py3/bin/activate" >
/opt/jumpserver/.env
(py3) [root@sdp-dev jumpserver]# cd 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
(py3) [root@sdp-dev requirements]#

(py3) [root@sdp-dev requirements]#  yum -y install $(cat rpm_requirements.txt)
(py3) [root@sdp-dev requirements]# pip install --upgrade pip
(py3) [root@sdp-dev requirements]#  pip install -r requirements.txt
```

4、安装Redis

```
(py3) [root@sdp-dev requirements]#  yum -y install redis
(py3) [root@sdp-dev requirements]#  systemctl enable redis
Created symlink from /etc/systemd/system/multi-user.target.wants/redis.service
to /usr/lib/systemd/system/redis.service.
(py3) [root@sdp-dev requirements]# systemctl start redis
```

5、安装MySQL

```
(py3) [root@sdp-dev requirements]# yum -y install mariadb mariadb-devel mariadb-
server
(py3) [root@sdp-dev requirements]# systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service
to /usr/lib/systemd/system/mariadb.service.
(py3) [root@sdp-dev requirements]# systemctl start mariadb
(py3) [root@sdp-dev requirements]#

(py3) [root@sdp-dev requirements]# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.60-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>  create database jumpserver default charset 'utf8';
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> grant all on jumpserver.* to 'jumpserveradmin'@'127.0.0.1'
identified by 'jumpserverpwd';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> \q
Bye
(py3) [root@sdp-dev requirements]#
```

6、配置Jumpserver

```
(py3) [root@sdp-dev requirements]# pwd
/opt/jumpserver/requirements
(py3) [root@sdp-dev requirements]# cd ..
(py3) [root@sdp-dev jumpserver]# ls
apps      config_example.yml  Dockerfile  entrypoint.sh  LICENSE  README_EN.md  requirements  tmp
build.sh  data                docs        jms            logs     README.md     run_server.py  utils
(py3) [root@sdp-dev jumpserver]# cp config_example.yml config.yml
(py3) [root@sdp-dev jumpserver]#
(py3) [root@sdp-dev jumpserver]# SECRET_KEY=`cat /dev/urandom | tr -dc A-Za-z0-9 | head -c 50`
(py3) [root@sdp-dev jumpserver]# echo $SECRET_KEY
vFjo4WEMRWNinXMconEXodf3VeEaRStkDzo6SpIfNxphYEEMUZ
(py3) [root@sdp-dev jumpserver]# echo "SECRET_KEY=$SECRET_KEY" >> ~/.bashrc
(py3) [root@sdp-dev jumpserver]# BOOTSTRAP_TOKEN=`cat /dev/urandom | tr -dc A-Za-z0-9 | head -c 16`
(py3) [root@sdp-dev jumpserver]# echo "BOOTSTRAP_TOKEN=$BOOTSTRAP_TOKEN" >> ~/.bashrc
(py3) [root@sdp-dev jumpserver]# echo $BOOTSTRAP_TOKEN
yBCVQ9WHA9phTZ21
(py3) [root@sdp-dev jumpserver]# sed -i "s/SECRET_KEY:/SECRET_KEY: $SECRET_KEY/g" /opt/jumpserver/config.yml
(py3) [root@sdp-dev jumpserver]# sed -i "s/BOOTSTRAP_TOKEN:/BOOTSTRAP_TOKEN: $BOOTSTRAP_TOKEN/g" /opt/jumpserver/config.yml
(py3) [root@sdp-dev jumpserver]# sed -i "s/# DEBUG: true/DEBUG: false/g" /opt/jumpserver/config.yml
(py3) [root@sdp-dev jumpserver]# sed -i "s/# LOG_LEVEL: DEBUG/LOG_LEVEL: ERROR/g" /opt/jumpserver/config.yml
(py3) [root@sdp-dev jumpserver]#  sed -i "s/# SESSION_EXPIRE_AT_BROWSER_CLOSE: false/SESSION_EXPIRE_AT_BROWSER_CLOSE: true/g" /opt/jumpserver/config.yml
(py3) [root@sdp-dev jumpserver]# echo -e "\033[31m 你的SECRET_KEY是 $SECRET_KEY \033[0m"
 你的SECRET_KEY是 vFjo4WEMRWNinXMconEXodf3VeEaRStkDzo6SpIfNxphYEEMUZ
(py3) [root@sdp-dev jumpserver]# echo -e "\033[31m 你的BOOTSTRAP_TOKEN是 $BOOTSTRAP_TOKEN \033[0m"
 你的BOOTSTRAP_TOKEN是 yBCVQ9WHA9phTZ21
(py3) [root@sdp-dev jumpserver]# vi config.yml
(py3) [root@sdp-dev jumpserver]# sed -n '/^DB_/p' /opt/jumpserver/config.yml
DB_ENGINE: mysql
DB_HOST: 127.0.0.1
DB_PORT: 3306
DB_USER: jumpserveradmin
DB_PASSWORD: jumpserverpwd
DB_NAME: jumpserver
(py3) [root@sdp-dev jumpserver]#
```

7、启动/关闭Jumpserver

```
(py3) [root@sdp-dev jumpserver]# ./jms start
......

(py3) [root@sdp-dev jumpserver]# ./jms stop
Stop service: gunicorn
Stop service: celery
Stop service: beat
(py3) [root@sdp-dev jumpserver]#
```

后台启动

```
(py3) [root@sdp-dev jumpserver]# ./jms start -d
```

8、部署koko

支持终端管理，默认port为2222

```
[root@sdp-dev ~]# systemctl start docker
[root@sdp-dev ~]#
[root@sdp-dev ~]# Server_IP=192.168.20.32
[root@sdp-dev ~]# BOOTSTRAP_TOKEN=yBCVQ9WHA9phTZ21
[root@sdp-dev ~]# docker run --name jms_koko -d -p 2222:2222 -p 5000:5000 -e
CORE_HOST=http://$Server_IP:8080 -e BOOTSTRAP_TOKEN=$BOOTSTRAP_TOKEN
jumpserver/jms_koko:1.5.5
Unable to find image 'jumpserver/jms_koko:1.5.5' locally
1.5.2: Pulling from jumpserver/jms_koko
050382585609: Pull complete
f6e2d22aa00f: Pull complete
8c86c00c5332: Pull complete
6b9c6941a89d: Pull complete
a10054b94acf: Pull complete
4005724a64ff: Pull complete
446406ca2953: Pull complete
716a981c63ee: Pull complete
41a65efed49e: Pull complete
Digest: sha256:ac6258fe46165860289410970e124031aa74a380cb3e1ad97348feb2c9265cbc
Status: Downloaded newer image for jumpserver/jms_koko:1.5.5
31fc5862ea104946590c232f16dab366d55823e559e256c5208a3720be9406ba
[root@sdp-dev ~]#
```

手工部署koko（coco 目前已经被 koko 取代)

```
cd /opt
wget https://github.com/jumpserver/koko/releases/download/1.5.2/koko-master-37daa82-linux-amd64.tar.gz
tar xf koko-master-37daa82-linux-amd64.tar.gz
chown -R root:root kokodir
cd kokodir
chown -R root:root /opt/kokodir
cd /opt/kokodir
cp config_example.yml config.yml
vim config.yml  # BOOTSTRAP_TOKEN 需要从 jumpserver/config.yml 里面获取，保证一致
./koko
```

9、部署guacamole

基于 HTML 5 和 JavaScript 的 VNC 查看器

```
[root@sdp-dev ~]# docker run --name jms_guacamole -d -p 8081:8081 -e JUMPSERVER_SERVER=http://$Server_IP:8080 -e BOOTSTRAP_TOKEN=$BOOTSTRAP_TOKEN jumpserver/jms_guacamole:1.5.5
Unable to find image 'jumpserver/jms_guacamole:1.5.5' locally
1.5.5: Pulling from jumpserver/jms_guacamole
8ba884070f61: Pull complete
74b389e6937e: Pull complete
41f5461bfc2f: Pull complete
f693f2484212: Pull complete
246835158fe4: Pull complete
Digest: sha256:de0b74e33c9991181eb507d768df73fb05932f3b4722dc36ecdca4e358fdce8d
Status: Downloaded newer image for jumpserver/jms_guacamole:1.5.5
f4d0c314c5fb840e42ea7e284f5349c571039bb1e3af2f3f8377b7a2c5f53f82
[root@sdp-dev ~]#
```

手工部署guacamole

```
$ cd /opt
$ git clone --depth=1 https://github.com/jumpserver/docker-guacamole.git
$ cd /opt/docker-guacamole
$ tar xf guacamole-server-1.0.0.tar.gz
$ cd /opt/docker-guacamole/guacamole-server-1.0.0
# 根据 http://guacamole.apache.org/doc/gug/installing-guacamole.html 文档安装对应的依赖包
$ autoreconf -fi
$ ./configure --with-init-dir=/etc/init.d
$ make
$ make install

# 访问 https://tomcat.apache.org/download-90.cgi 下载最新的 tomcat9
$ mkdir -p /config/guacamole /config/guacamole/lib /config/guacamole/extensions /config/guacamole/data/log/
$ cd /config
$ wget http://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-9/v9.0.22/bin/apache-tomcat-9.0.22.tar.gz
$ tar xf apache-tomcat-9.0.22.tar.gz
$ mv apache-tomcat-9.0.22 tomcat9
$ rm -rf /config/tomcat9/webapps/*
$ sed -i 's/Connector port="8080"/Connector port="8081"/g' /config/tomcat9/conf/server.xml
$ echo "java.util.logging.ConsoleHandler.encoding = UTF-8" >>
/config/tomcat9/conf/logging.properties
$ ln -sf /opt/docker-guacamole/guacamole-1.0.0.war
/config/tomcat9/webapps/ROOT.war
$ ln -sf /opt/docker-guacamole/guacamole-auth-jumpserver-1.0.0.jar
/config/guacamole/extensions/guacamole-auth-jumpserver-1.0.0.jar
$ ln -sf /opt/docker-guacamole/root/app/guacamole/guacamole.properties
/config/guacamole/guacamole.properties
$ wget https://github.com/ibuler/ssh-forward/releases/download/v0.0.5/linux-
amd64.tar.gz
$ tar xf linux-amd64.tar.gz -C /bin/
$ chmod +x /bin/ssh-forward

# 设置 guacamole 环境
$ export JUMPSERVER_SERVER=http://127.0.0.1:8080  # http://127.0.0.1:8080 指
jumpserver 访问地址
$ echo "export JUMPSERVER_SERVER=http://127.0.0.1:8080" >> ~/.bashrc

# BOOTSTRAP_TOKEN 为 Jumpserver/config.yml 里面的 BOOTSTRAP_TOKEN 值
$ export BOOTSTRAP_TOKEN=******
$ echo "export BOOTSTRAP_TOKEN=******" >> ~/.bashrc
$ export JUMPSERVER_KEY_DIR=/config/guacamole/keys
$ echo "export JUMPSERVER_KEY_DIR=/config/guacamole/keys" >> ~/.bashrc
$ export GUACAMOLE_HOME=/config/guacamole
$ echo "export GUACAMOLE_HOME=/config/guacamole" >> ~/.bashrc

$ /etc/init.d/guacd start
$ sh /config/tomcat9/bin/startup.sh
```

10、部署luna

与nginx结合支持Web Terminal前端

```
[root@sdp-dev ~]# cd /opt/
[root@sdp-dev opt]# wget
https://github.com/jumpserver/luna/releases/download/1.5.5/luna.tar.gz
[root@sdp-dev opt]# tar xf luna.tar.gz
[root@sdp-dev opt]# chown -R root:root luna
```

11、配置nginx

```
[root@sdp-dev opt]# cd /usr/local/nginx/conf/
[root@sdp-dev conf]# ls
fastcgi.conf            koi-utf             nginx.conf          uwsgi_params
fastcgi.conf.default    koi-win             nginx.conf.default
uwsgi_params.default
fastcgi_params          mime.types          scgi_params         win-utf
fastcgi_params.default  mime.types.default  scgi_params.default
[root@sdp-dev conf]# mkdir conf.d
[root@sdp-dev conf]# cd conf.d/
[root@sdp-dev conf.d]# vim jumpserver.conf
[root@sdp-dev conf.d]# ls
jumpserver.conf
[root@sdp-dev conf.d]# cat jumpserver.conf
server {
    listen 80;
    # server_name _;
    server_name bastion.qf.com;

    client_max_body_size 100m;  # 录像及文件上传大小限制

    location /luna/ {
        try_files $uri / /index.html;
        alias /opt/luna/;  # luna 路径，如果修改安装目录，此处需要修改
    }

    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 /koko/ {
        proxy_pass       http://localhost:5000;
        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://localhost:8081/;
        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 /ws/ {
        proxy_pass http://localhost:8070;
        proxy_http_version 1.1;
        proxy_buffering off;
        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 / {
        proxy_pass http://localhost:8080;
        proxy_set_header X-Real-IP $remote_addr;
```

```
            proxy_set_header Host $host;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            access_log off;
        }
}
[root@sdp-dev conf.d]#
[root@sdp-dev conf.d]# cd ..
[root@sdp-dev conf]# vim nginx.conf
[root@sdp-dev conf]# grep -Pv "^($| *#)" nginx.conf
worker_processes  1;
events {
    worker_connections  1024;
}
http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    keepalive_timeout  65;
    include /usr/local/nginx/conf/conf.d/*.conf;
}
[root@sdp-dev conf]# cd ..
[root@sdp-dev nginx]# sbin/nginx -t
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
[root@sdp-dev nginx]#
```

12、Jumpserver 登录测试

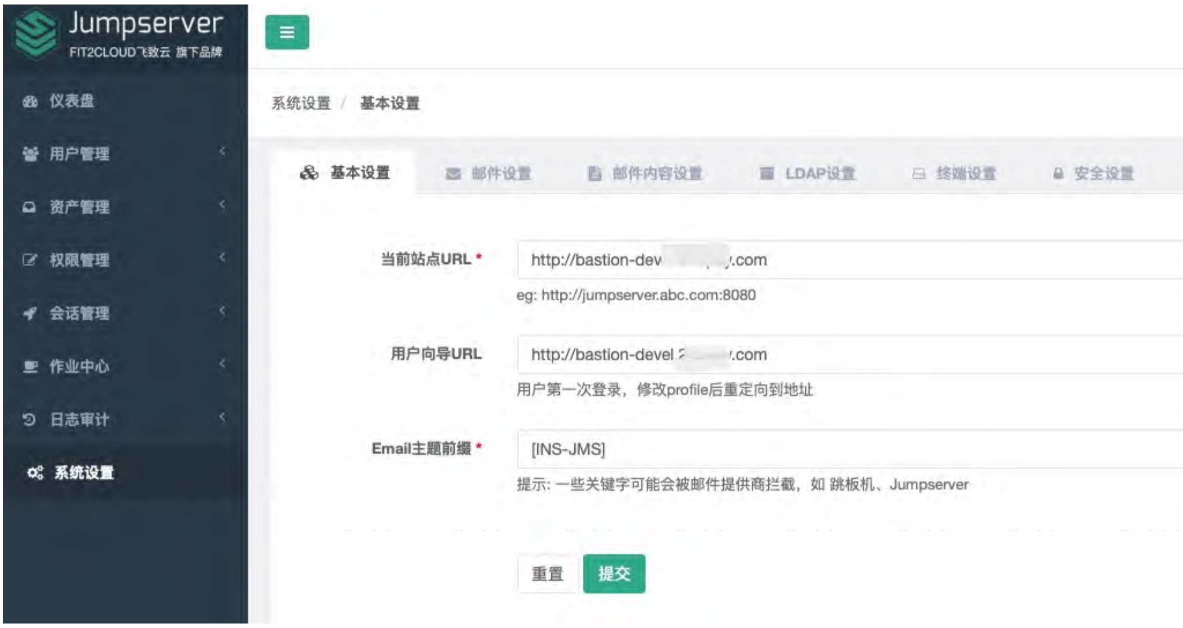
```
# 检查应用是否已经正常运行
# 服务全部启动后，访问 jumpserver 服务器 nginx 代理的 80 端口，不要通过8080端口访问
# 默认账号: admin 密码: admin
```

13、快速入门

参考：https://jumpserver.readthedocs.io/zh/master/quick_start.html

####系统设置

设置用户访问的URL

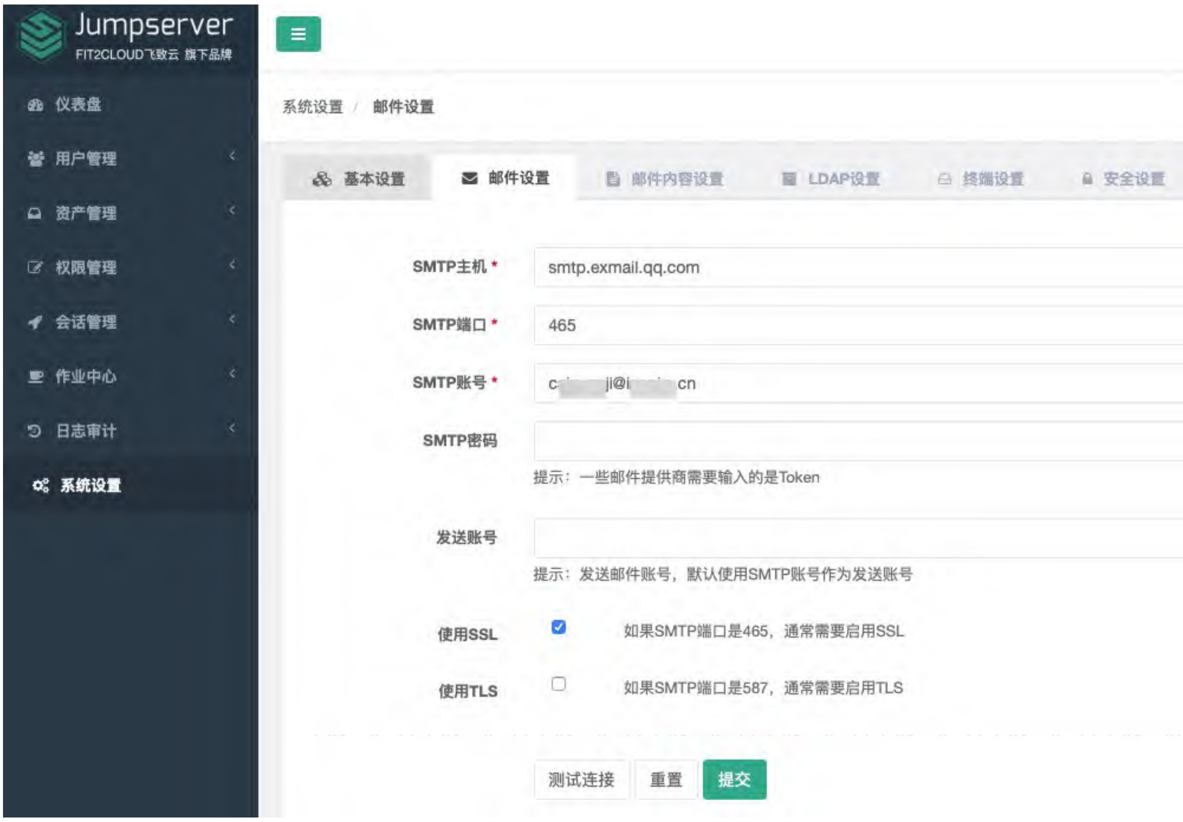


当前站点URL * http://bastion-dev[redacted].com
eg: http://jumpserver.abc.com:8080

用户向导URL http://bastion-devel[redacted].com
用户第一次登录，修改profile后重定向到地址

Email主题前缀 * [INS-JMS]
提示: 一些关键字可能会被邮件提供商拦截，如 跳板机、Jumpserver

设置邮件地址及验证



SMTP主机 * smtp.exmail.qq.com

SMTP端口 * 465

SMTP账号 * c[redacted]ji@i[redacted]cn

SMTP密码

提示：一些邮件提供商需要输入的是Token

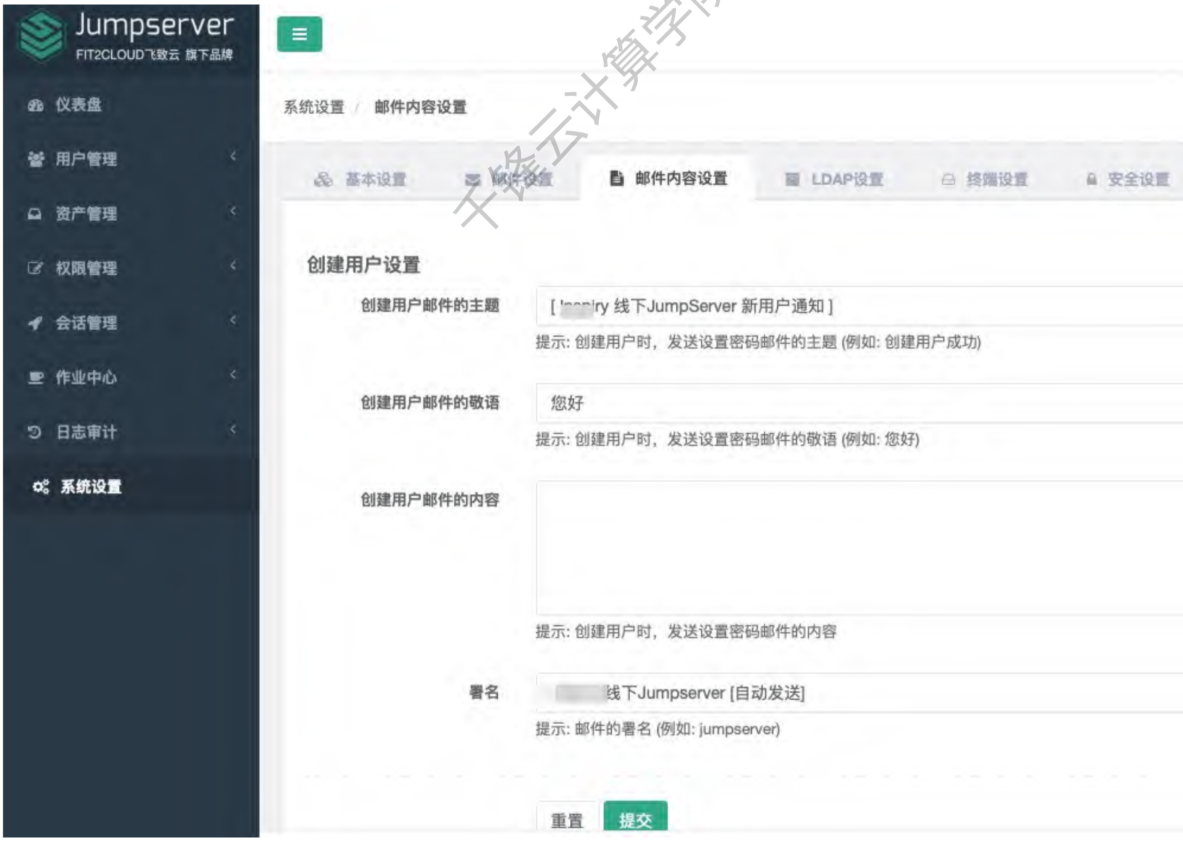
发送账号

提示：发送邮件账号，默认使用SMTP账号作为发送账号

使用SSL ☑ 如果SMTP端口是465，通常需要启用SSL

使用TLS □ 如果SMTP端口是587，通常需要启用TLS

设置邮件内容



创建用户设置

创建用户邮件的主题 [[redacted]ry 线下JumpServer 新用户通知]

提示: 创建用户时，发送设置密码邮件的主题 (例如: 创建用户成功)

创建用户邮件的敬语 您好

提示: 创建用户时，发送设置密码邮件的敬语 (例如: 您好)

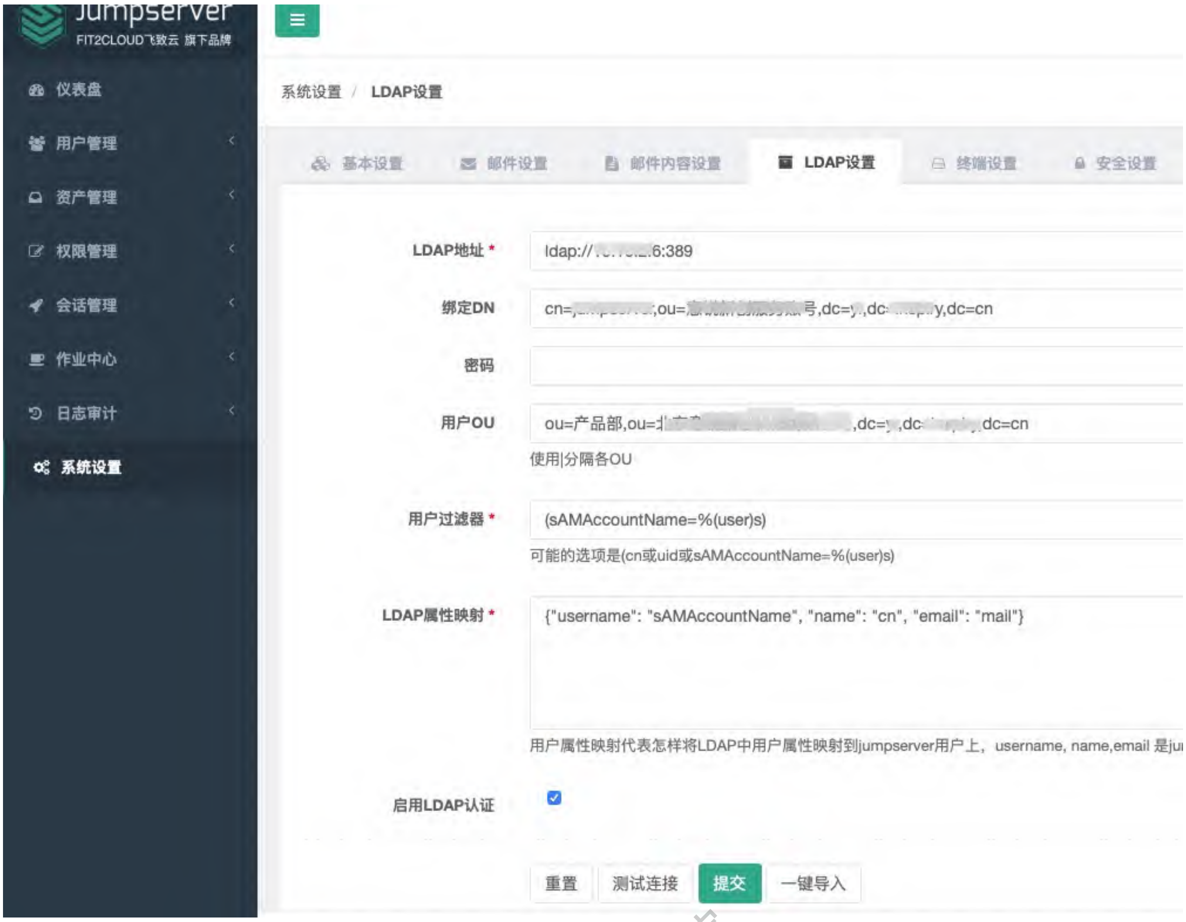
创建用户邮件的内容

提示: 创建用户时，发送设置密码邮件的内容

署名 [redacted]线下Jumpserver [自动发送]

提示: 邮件的署名 (例如: jumpserver)

配置LDAP



系统设置 / LDAP设置

LDAP地址 * ldap://[redacted]6:389

绑定DN cn=[redacted],ou=[redacted]号,dc=[redacted],dc=[redacted]y,dc=cn

密码

用户OU ou=产品部,ou=[redacted],dc=[redacted],dc=[redacted],dc=cn

使用|分隔各OU

用户过滤器 * (sAMAccountName=%(user)s)

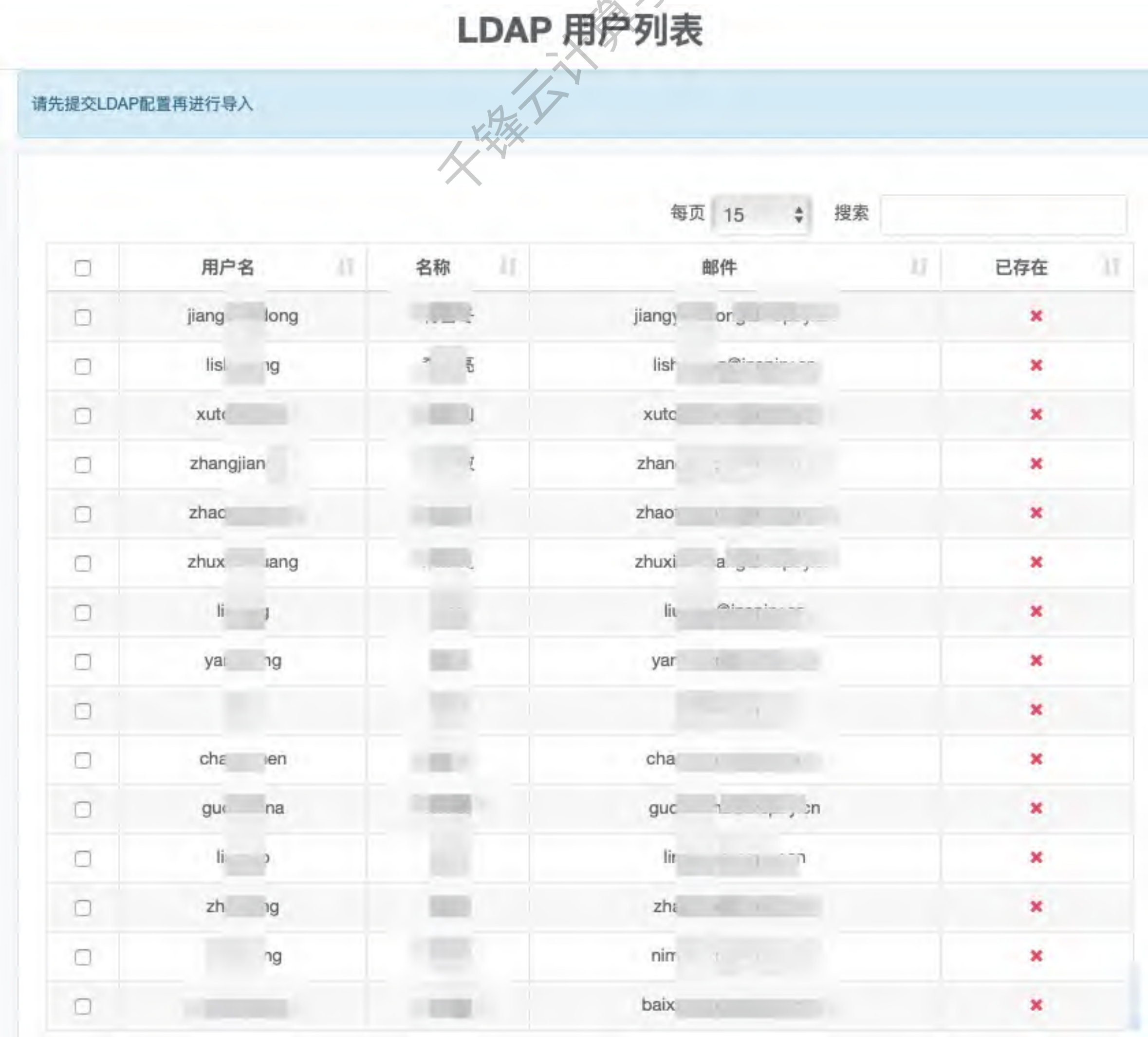
可能的选项是(cn或uid或sAMAccountName=%(user)s)

LDAP属性映射 * {"username": "sAMAccountName", "name": "cn", "email": "mail"}

用户属性映射代表怎样将LDAP中用户属性映射到jumpserver用户上，username, name,email 是ju

启用LDAP认证 ☑

导入LDAP用户



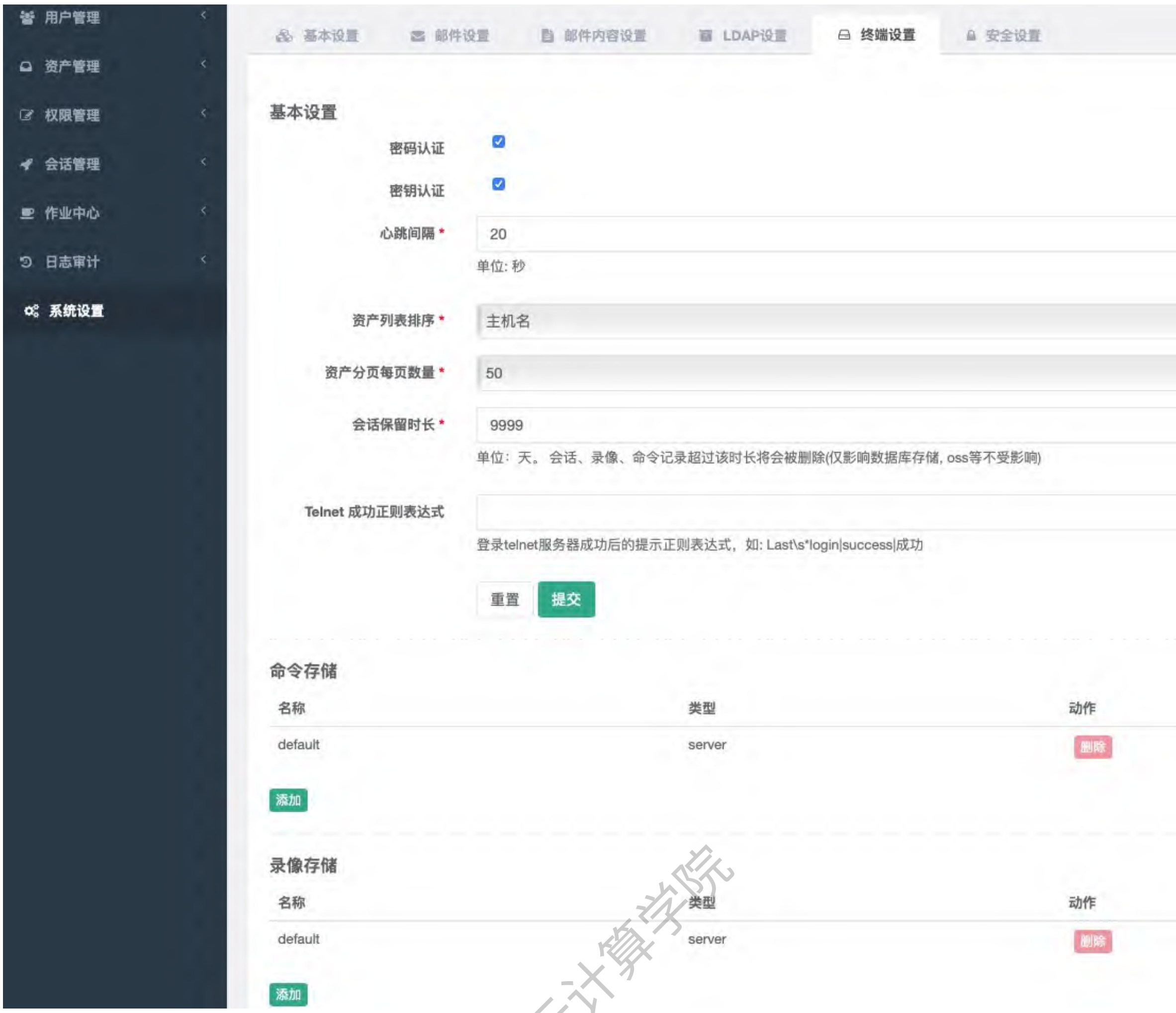
LDAP 用户列表

请先提交LDAP配置再进行导入

每页 15 搜索

□	用户名	名称	邮件	已存在
□	jiang[redacted]long	[redacted]	jiang[redacted]	✗
□	lis[redacted]ng	[redacted]	lish[redacted]	✗
□	xut[redacted]	[redacted]	xuto[redacted]	✗
□	zhangjian[redacted]	[redacted]	zhan[redacted]	✗
□	zhao[redacted]	[redacted]	zhao[redacted]	✗
□	zhux[redacted]iang	[redacted]	zhuxi[redacted]	✗
□	li[redacted]	[redacted]	liu[redacted]	✗
□	ya[redacted]ng	[redacted]	yar[redacted]	✗
□	[redacted]	[redacted]	[redacted]	✗
□	cha[redacted]en	[redacted]	cha[redacted]	✗
□	gu[redacted]na	[redacted]	guo[redacted]cn	✗
□	li[redacted]	[redacted]	lin[redacted]	✗
□	zh[redacted]ng	[redacted]	zha[redacted]	✗
□	[redacted]ng	[redacted]	nin[redacted]	✗
□	[redacted]	[redacted]	baix[redacted]	✗

终端设置



基本设置

密码认证 ☑

密钥认证 ☑

心跳间隔 * 20
单位: 秒

资产列表排序 * 主机名

资产分页每页数量 * 50

会话保留时长 * 9999
单位：天。 会话、录像、命令记录超过该时长将会被删除(仅影响数据库存储, oss等不受影响)

Telnet 成功正则表达式
登录telnet服务器成功后的提示正则表达式，如: Last\s*login|success|成功

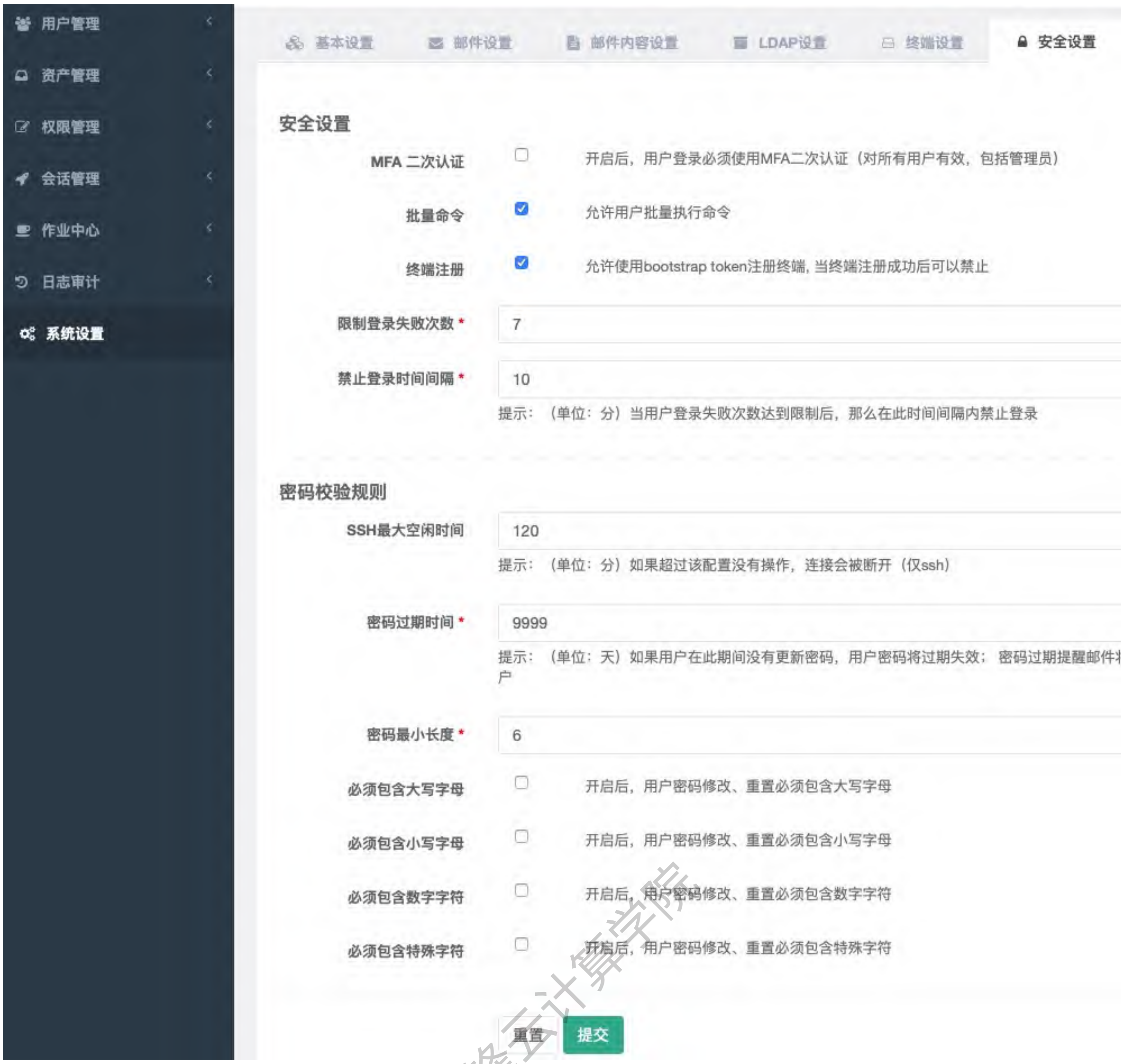
命令存储

名称	类型	动作
default	server	删除

录像存储

名称	类型	动作
default	server	删除

安全设置



安全设置

MFA 二次认证 □ 开启后，用户登录必须使用MFA二次认证（对所有用户有效，包括管理员）

批量命令 ☑ 允许用户批量执行命令

终端注册 ☑ 允许使用bootstrap token注册终端，当终端注册成功后可以禁止

限制登录失败次数 * 7

禁止登录时间间隔 * 10

提示：（单位：分）当用户登录失败次数达到限制后，那么在此时间间隔内禁止登录

密码校验规则

SSH最大空闲时间 120

提示：（单位：分）如果超过该配置没有操作，连接会被断开（仅ssh）

密码过期时间 * 9999

提示：（单位：天）如果用户在此期间没有更新密码，用户密码将过期失效；密码过期提醒邮件
户

密码最小长度 * 6

必须包含大写字母 □ 开启后，用户密码修改、重置必须包含大写字母

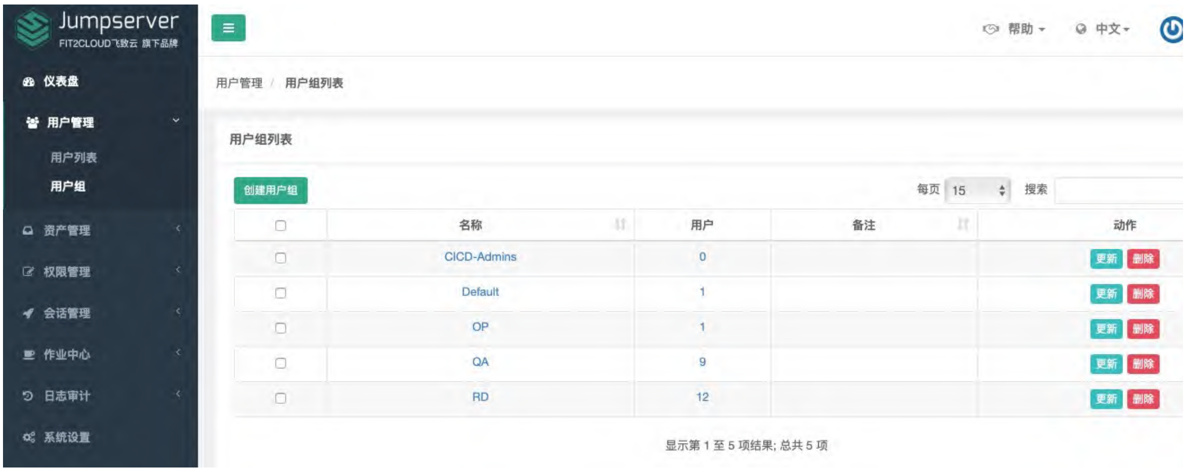
必须包含小写字母 □ 开启后，用户密码修改、重置必须包含小写字母

必须包含数字字符 □ 开启后，用户密码修改、重置必须包含数字字符

必须包含特殊字符 □ 开启后，用户密码修改、重置必须包含特殊字符

用户管理

用户组设置

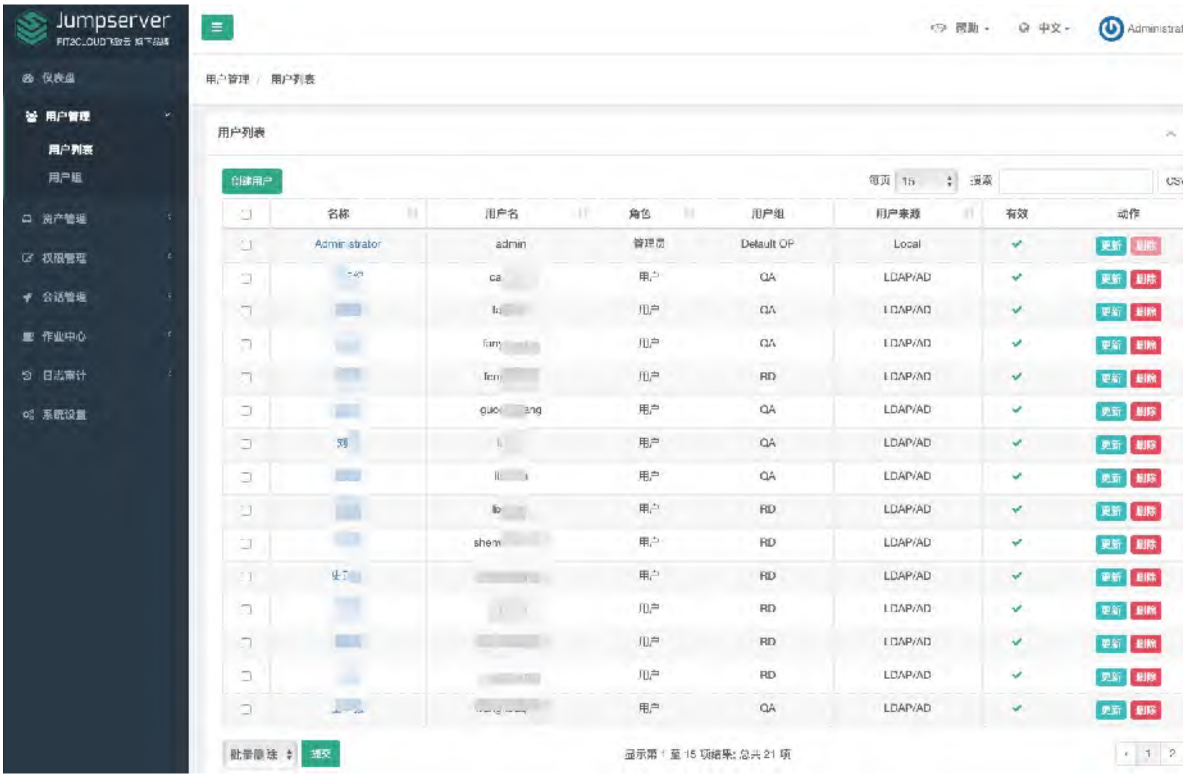


用户组列表

□	名称	用户	备注	动作
□	CICD-Admins	0		更新 删除
□	Default	1		更新 删除
□	OP	1		更新 删除
□	QA	9		更新 删除
□	RD	12		更新 删除

显示第 1 至 5 项结果；总共 5 项

用户设置



用户管理 / 用户列表

用户列表

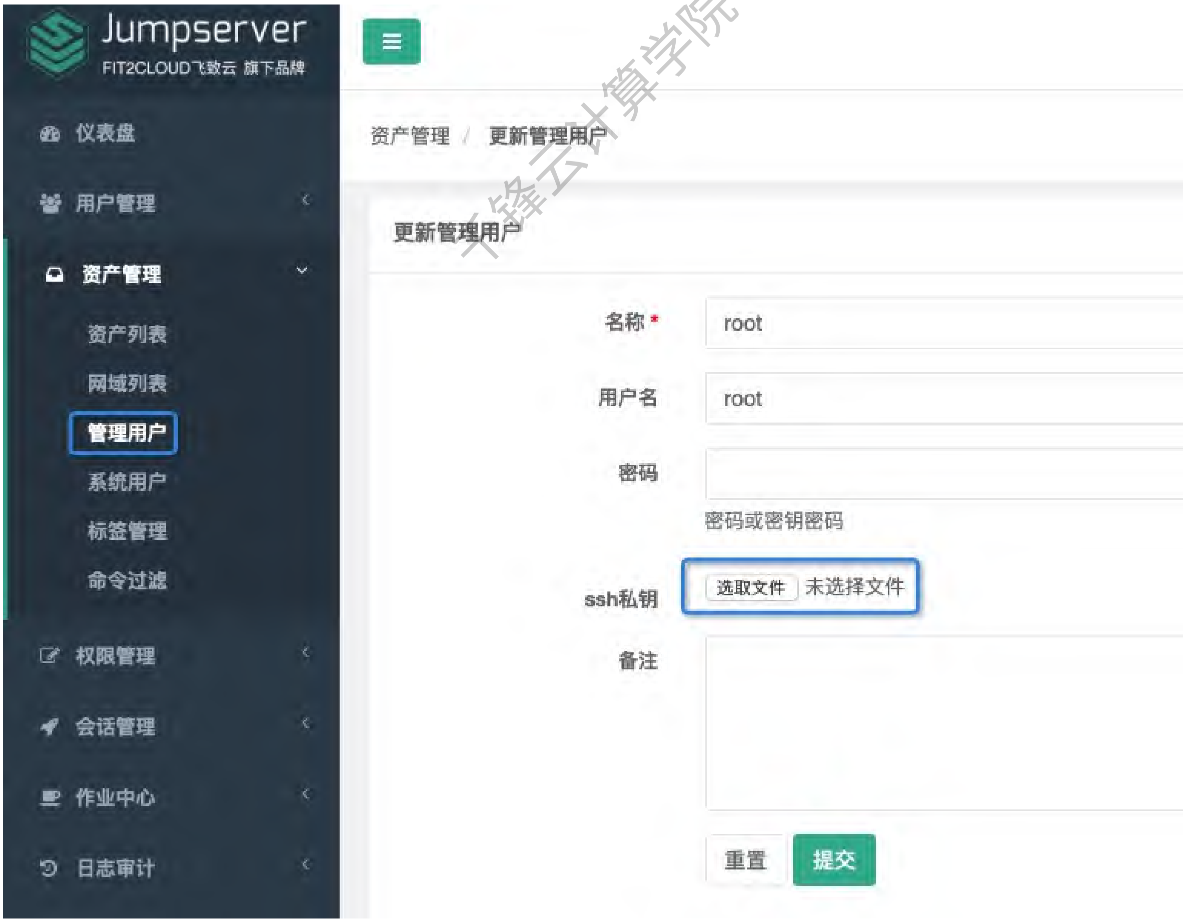
	名称	用户名	角色	用户组	用户来源	有效	动作
□	Administrator	admin	管理员	Default OP	Local	✓	更新 删除
□	[redacted]	[redacted]	用户	QA	LDAP/AD	✓	更新 删除
□	[redacted]	[redacted]	用户	QA	LDAP/AD	✓	更新 删除
□	[redacted]	[redacted]	用户	QA	LDAP/AD	✓	更新 删除
□	[redacted]	[redacted]	用户	RD	LDAP/AD	✓	更新 删除
□	[redacted]	[redacted]	用户	QA	LDAP/AD	✓	更新 删除
□	[redacted]	[redacted]	用户	QA	LDAP/AD	✓	更新 删除
□	[redacted]	[redacted]	用户	QA	LDAP/AD	✓	更新 删除
□	[redacted]	[redacted]	用户	RD	LDAP/AD	✓	更新 删除
□	[redacted]	[redacted]	用户	RD	LDAP/AD	✓	更新 删除
□	[redacted]	[redacted]	用户	RD	LDAP/AD	✓	更新 删除
□	[redacted]	[redacted]	用户	RD	LDAP/AD	✓	更新 删除
□	[redacted]	[redacted]	用户	RD	LDAP/AD	✓	更新 删除
□	[redacted]	[redacted]	用户	RD	LDAP/AD	✓	更新 删除
□	[redacted]	[redacted]	用户	QA	LDAP/AD	✓	更新 删除

显示第 1 至 15 项结果，总共 21 项

‹ 1 2

资产管理

资产管理--管理用户



资产管理 / 更新管理用户

更新管理用户

名称 * root

用户名 root

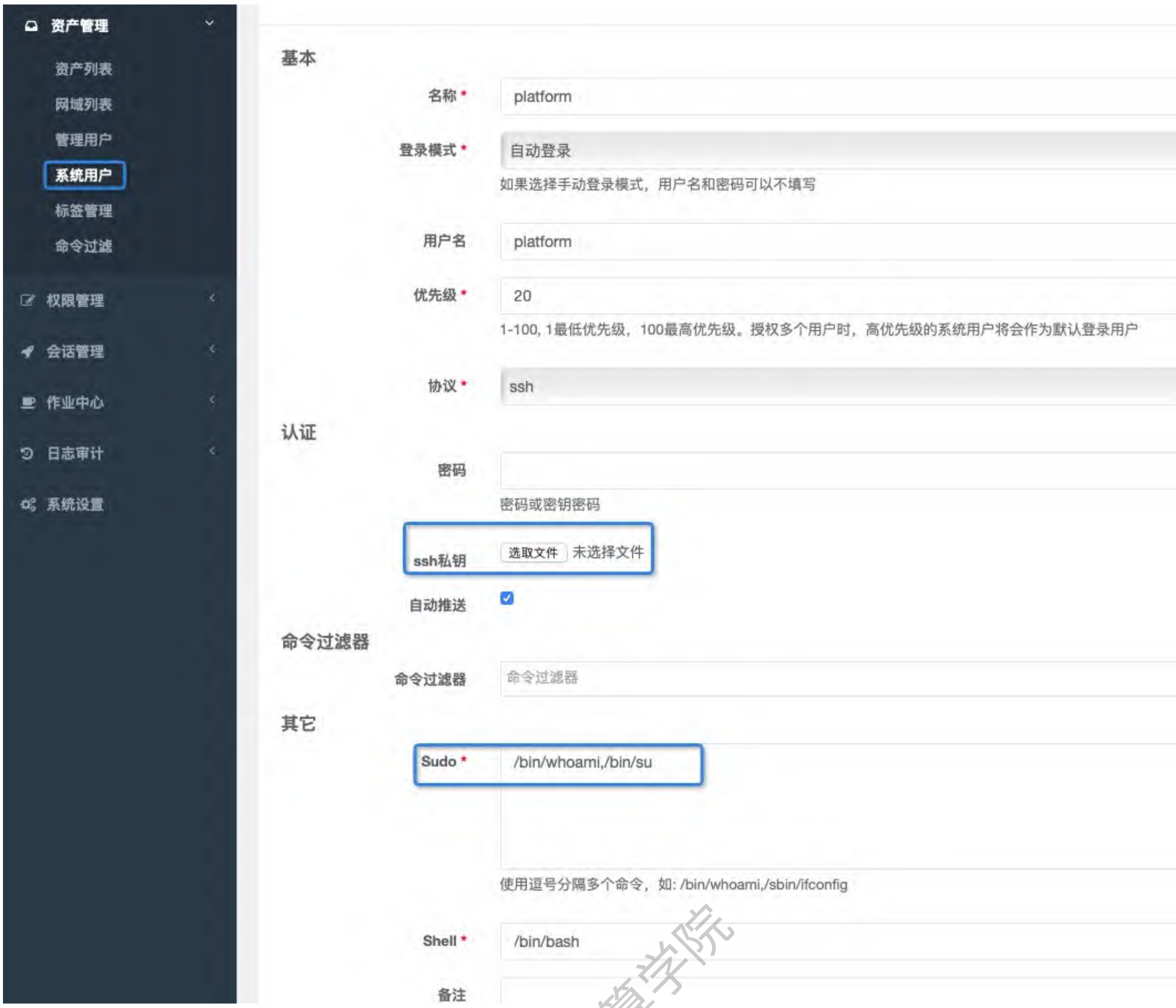
密码

密码或密钥密码

ssh私钥 选取文件 未选择文件

备注

资产管理--系统用户



基本

名称 * platform

登录模式 * 自动登录

如果选择手动登录模式，用户名和密码可以不填写

用户名 platform

优先级 * 20

1-100, 1最低优先级，100最高优先级。授权多个用户时，高优先级的系统用户将会作为默认登录用户

协议 * ssh

认证

密码

密码或密钥密码

ssh私钥 选取文件 未选择文件

自动推送 ☑

命令过滤器

命令过滤器 命令过滤器

其它

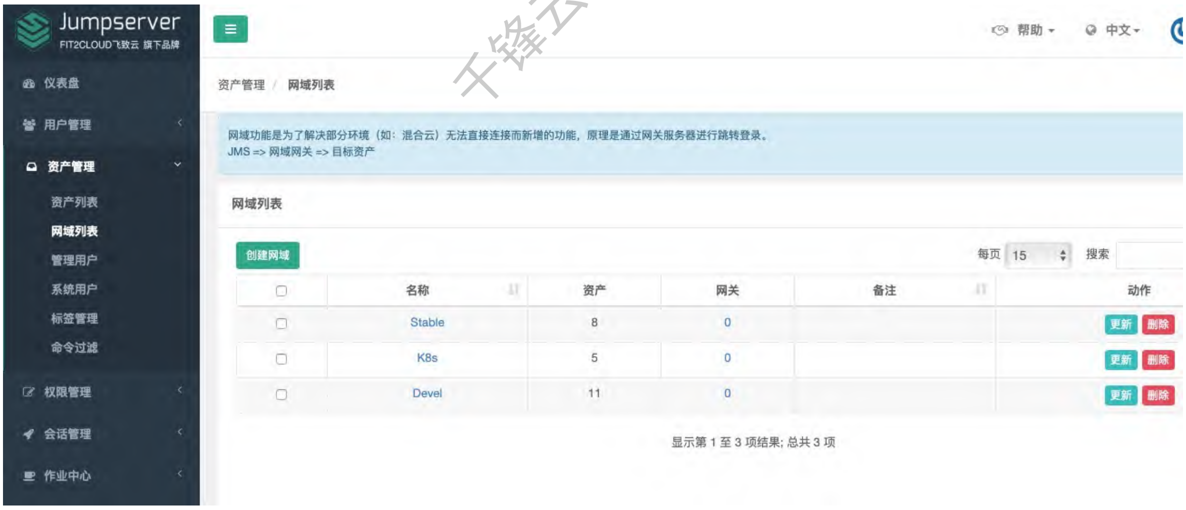
Sudo * /bin/whoami,/bin/su

使用逗号分隔多个命令，如: /bin/whoami,/sbin/ifconfig

Shell * /bin/bash

备注

资产管理--网域列表



资产管理 / 网域列表

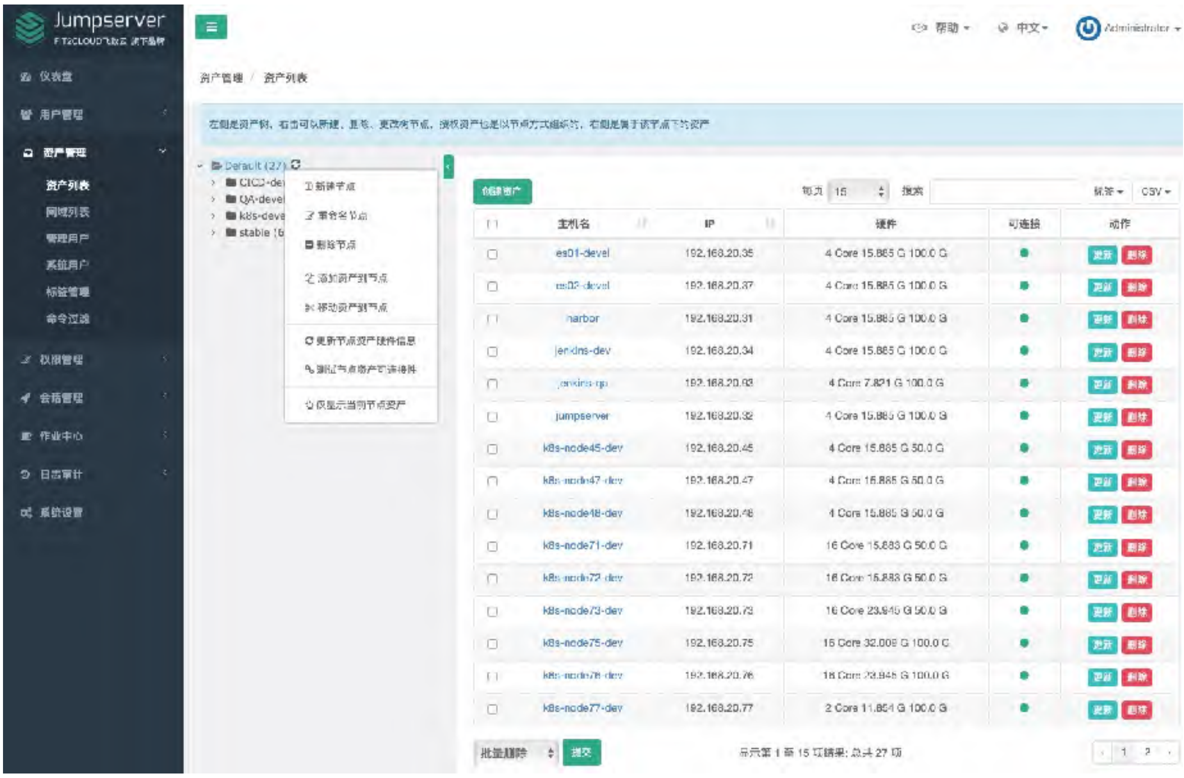
网域功能是为了解决部分环境（如：混合云）无法直接连接而新增的功能，原理是通过网关服务器进行跳转登录。
JMS => 网域网关 => 目标资产

网域列表

□	名称	资产	网关	备注	动作
□	Stable	8	0		更新 删除
□	K8s	5	0		更新 删除
□	Devel	11	0		更新 删除

显示第 1 至 3 项结果; 总共 3 项

资产管理--资产列表

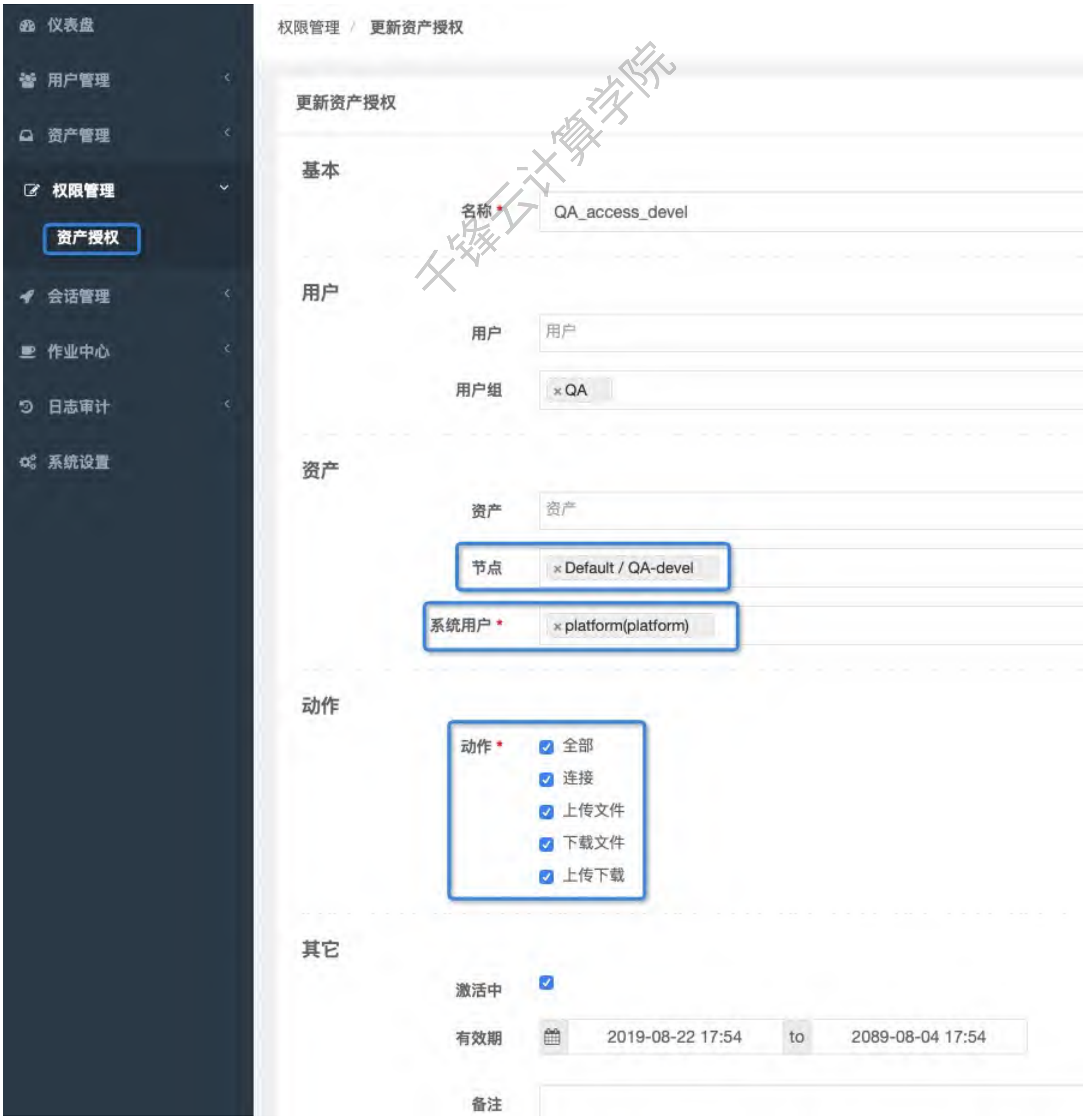


左侧是资产树，右击可以新建、删除、更改树节点，授权资产也是以节点方式组织的，右侧是属于该节点下的资产

□	主机名	IP	硬件	可连接	动作
□	es01-devel	192.168.20.35	4 Core 15.885 G 100.0 G	●	更新 删除
□	es02-devel	192.168.20.37	4 Core 15.885 G 100.0 G	●	更新 删除
□	harbor	192.168.20.31	4 Core 15.885 G 100.0 G	●	更新 删除
□	jenkins-dev	192.168.20.34	4 Core 15.885 G 100.0 G	●	更新 删除
□	jenkins-qa	192.168.20.93	4 Core 7.821 G 100.0 G	●	更新 删除
□	jumpserver	192.168.20.32	4 Core 15.885 G 100.0 G	●	更新 删除
□	k8s-node45-dev	192.168.20.45	4 Core 15.885 G 50.0 G	●	更新 删除
□	k8s-node47-dev	192.168.20.47	4 Core 15.885 G 50.0 G	●	更新 删除
□	k8s-node48-dev	192.168.20.48	4 Core 15.885 G 50.0 G	●	更新 删除
□	k8s-node71-dev	192.168.20.71	16 Core 15.883 G 50.0 G	●	更新 删除
□	k8s-node72-dev	192.168.20.72	16 Core 15.883 G 50.0 G	●	更新 删除
□	k8s-node73-dev	192.168.20.73	16 Core 23.845 G 50.0 G	●	更新 删除
□	k8s-node75-dev	192.168.20.75	16 Core 32.009 G 100.0 G	●	更新 删除
□	k8s-node76-dev	192.168.20.76	16 Core 23.845 G 100.0 G	●	更新 删除
□	k8s-node77-dev	192.168.20.77	2 Core 11.854 G 100.0 G	●	更新 删除

显示第 1 至 15 项结果，总共 27 项

权限管理



权限管理 / 更新资产授权

更新资产授权

基本

名称 * QA_access_devel

用户

用户 用户

用户组 × QA

资产

资产 资产

节点 × Default / QA-devel

系统用户 * × platform(platform)

动作

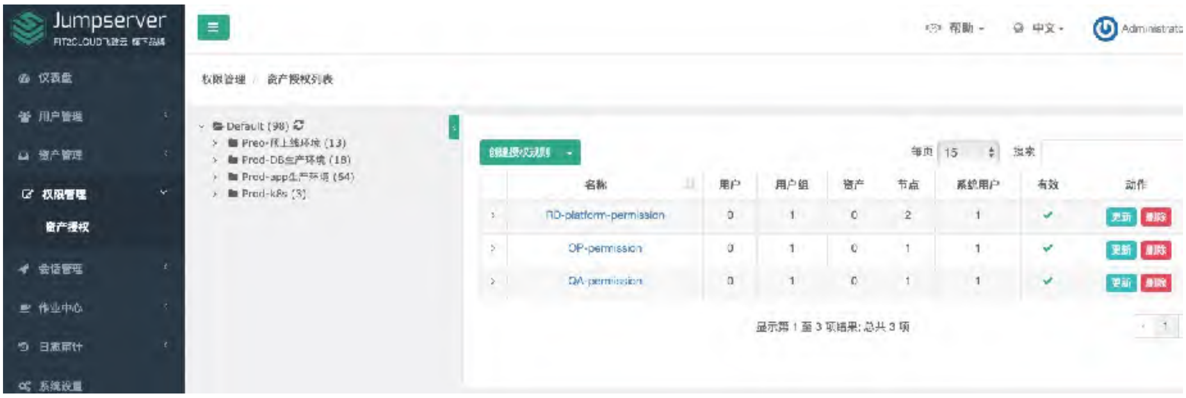
动作 * ☑ 全部 ☑ 连接 ☑ 上传文件 ☑ 下载文件 ☑ 上传下载

其它

激活中 ☑

有效期 2019-08-22 17:54 to 2089-08-04 17:54

备注

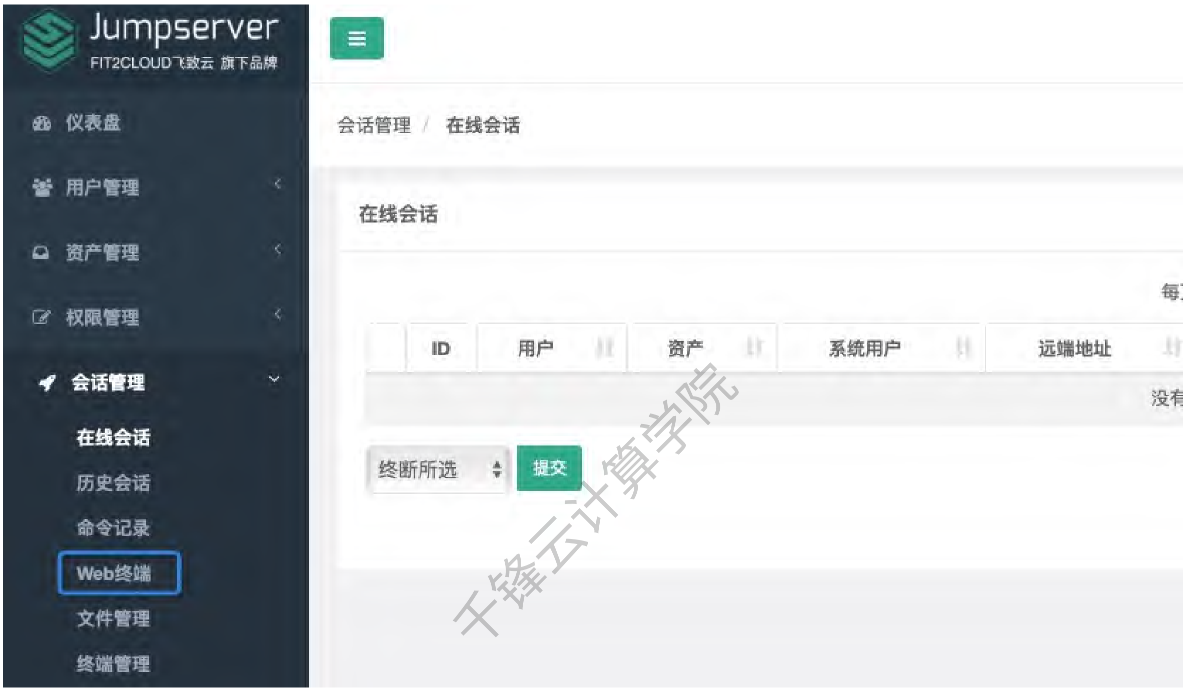


	名称	用户	用户组	资产	节点	系统用户	有效	动作
›	RD-platform-permission	0	1	0	2	1	✓	更新 删除
›	OP-permission	0	1	0	1	1	✓	更新 删除
›	QA-permission	0	1	0	1	1	✓	更新 删除

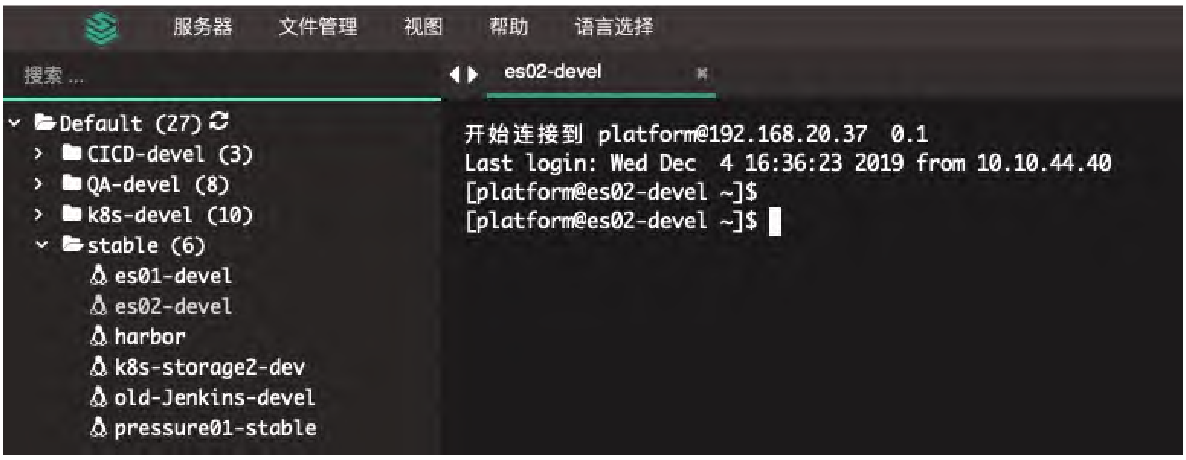
显示第 1 至 3 项结果，总共 3 项

会话管理

会话管理--Web终端

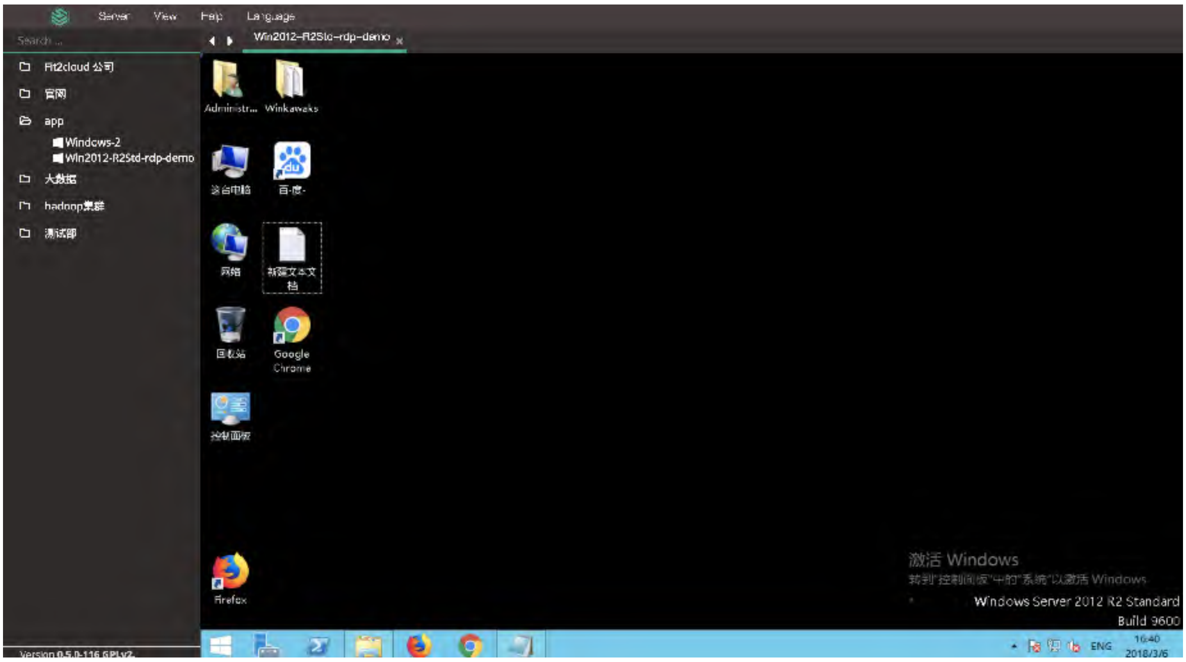


	ID	用户	资产	系统用户	远端地址
					没有

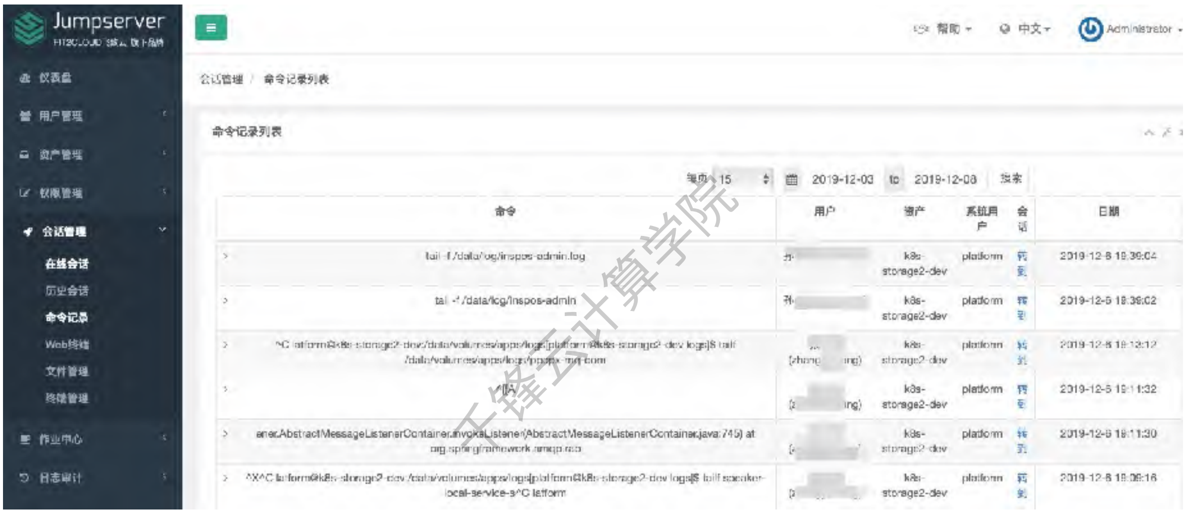


```
开始连接到 platform@192.168.20.37  0.1
Last login: Wed Dec  4 16:36:23 2019 from 10.10.44.40
[platform@es02-devel ~]$
[platform@es02-devel ~]$ 
```

windows终端



会话管理--命令记录



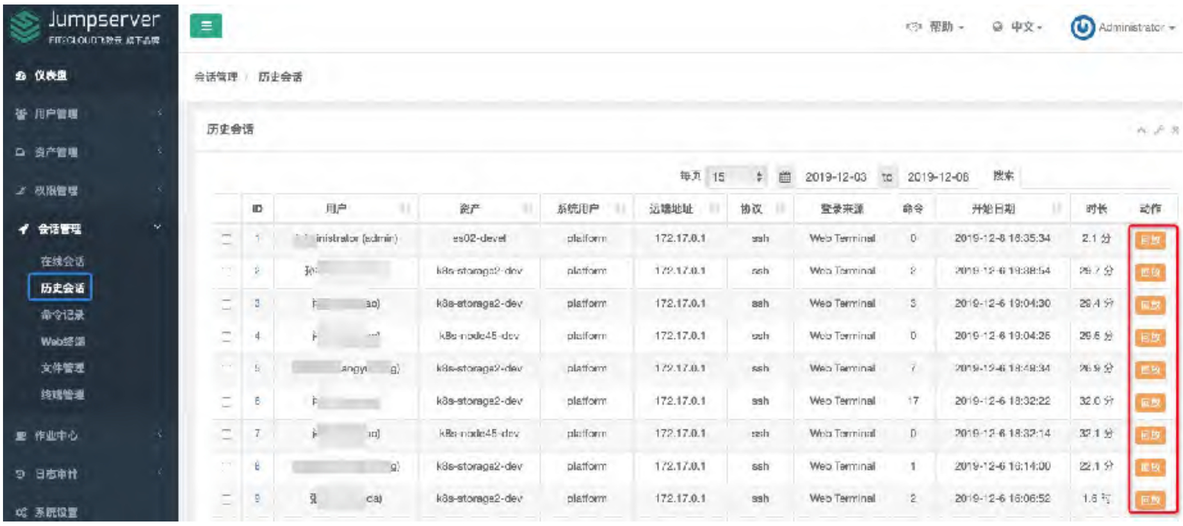
会话管理 / 命令记录列表

命令记录列表

每页 15 2019-12-03 to 2019-12-08 搜索

	命令	用户	资产	系统用户	会话	日期
>	tail -f /data/log/inspos-admin.log	孙[redacted]	k8s-storage2-dev	platform	跳到	2019-12-6 19:39:04
>	tail -f /data/log/inspos-admin	孙[redacted]	k8s-storage2-dev	platform	跳到	2019-12-6 19:39:02
>	^C latform@k8s-storage2-dev:/data/volumes/apps/logs[platform@k8s-storage2-dev logs]$ tail /data/volumes/apps/logs/ppppx-mq-com	[redacted] (zhang[redacted]ng)	k8s-storage2-dev	platform	跳到	2019-12-6 19:12:12
>	^[[A	[redacted] ([redacted]ng)	k8s-storage2-dev	platform	跳到	2019-12-6 19:11:32
>	ener.AbstractMessageListenerContainer.invokeListener(AbstractMessageListenerContainer.java:745) at org.springframework.amqp.rab	[redacted]	k8s-storage2-dev	platform	跳到	2019-12-6 19:11:30
>	^X^C latform@k8s-storage2-dev:/data/volumes/apps/logs[platform@k8s-storage2-dev logs]$ tail speaker-local-service-s^C latform	[redacted]	k8s-storage2-dev	platform	跳到	2019-12-6 19:09:16

会话管理--历史记录



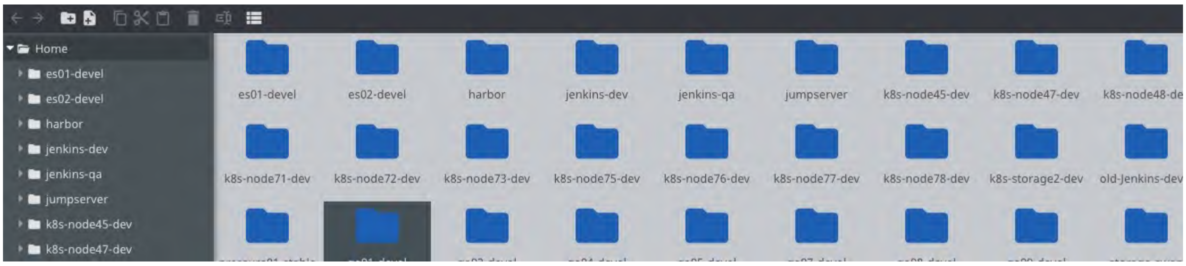
会话管理 / 历史会话

历史会话

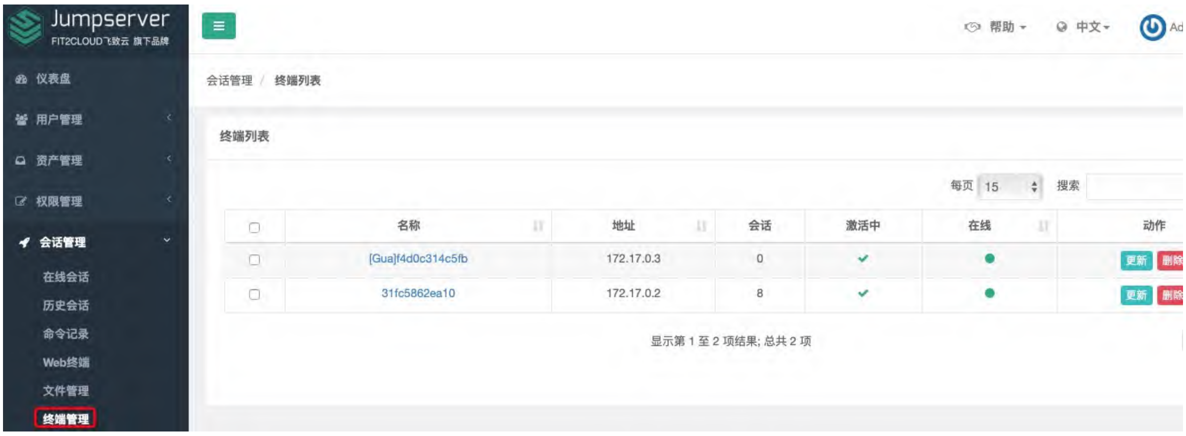
每页 15 2019-12-03 to 2019-12-06 搜索

	ID	用户	资产	系统用户	远端地址	协议	登录来源	命令	开始日期	时长	动作
□	1	[redacted]inistrator (admin)	es02-devel	platform	172.17.0.1	ssh	Web Terminal	0	2019-12-6 18:35:34	2.1 分	回放
□	2	孙[redacted]	k8s-storage2-dev	platform	172.17.0.1	ssh	Web Terminal	2	2019-12-6 19:38:54	29.7 分	回放
□	3	[redacted]ao)	k8s-storage2-dev	platform	172.17.0.1	ssh	Web Terminal	3	2019-12-6 19:04:30	29.4 分	回放
□	4	[redacted]	k8s-node45-dev	platform	172.17.0.1	ssh	Web Terminal	0	2019-12-6 19:04:26	29.6 分	回放
□	5	[redacted]angy[redacted]g)	k8s-storage2-dev	platform	172.17.0.1	ssh	Web Terminal	7	2019-12-6 18:48:34	26.9 分	回放
□	6	[redacted]	k8s-storage2-dev	platform	172.17.0.1	ssh	Web Terminal	17	2019-12-6 18:32:22	32.0 分	回放
□	7	[redacted]ao)	k8s-node45-dev	platform	172.17.0.1	ssh	Web Terminal	0	2019-12-6 18:32:14	32.1 分	回放
□	8	[redacted]g)	k8s-storage2-dev	platform	172.17.0.1	ssh	Web Terminal	1	2019-12-6 18:14:00	22.1 分	回放
□	9	张[redacted]cai)	k8s-storage2-dev	platform	172.17.0.1	ssh	Web Terminal	2	2019-12-6 16:06:52	1.6 分	回放

会话管理--文件管理



会话管理--终端管理



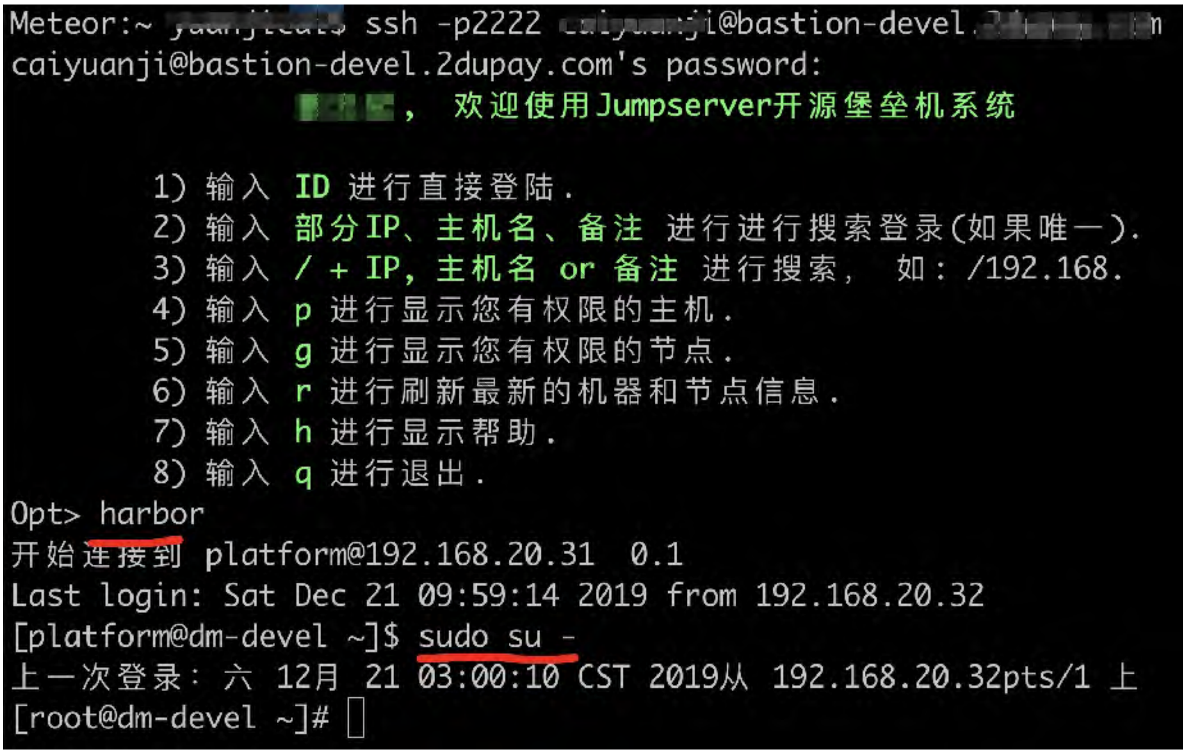
会话管理 / 终端列表

终端列表

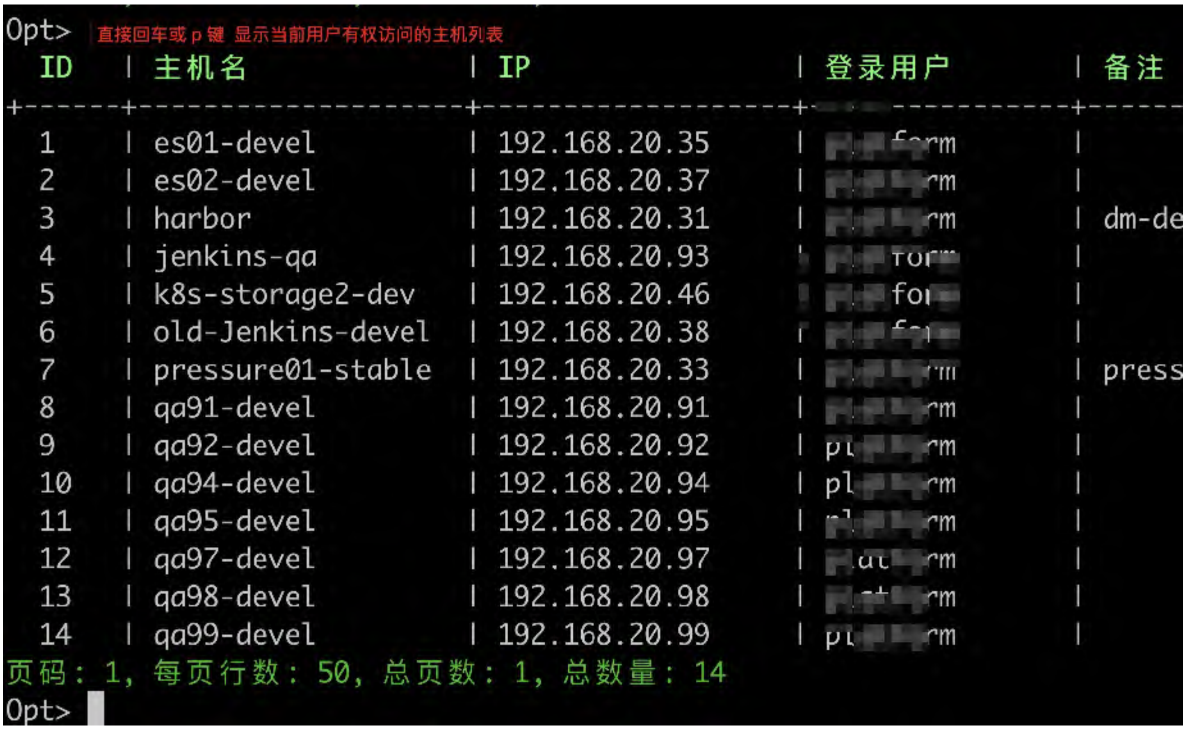
每页 15 搜索

□	名称	地址	会话	激活中	在线	动作
□	[Gua]f4d0c314c5fb	172.17.0.3	0	✓	●	更新 删除
□	31fc5862ea10	172.17.0.2	8	✓	●	更新 删除

显示第 1 至 2 项结果，总共 2 项



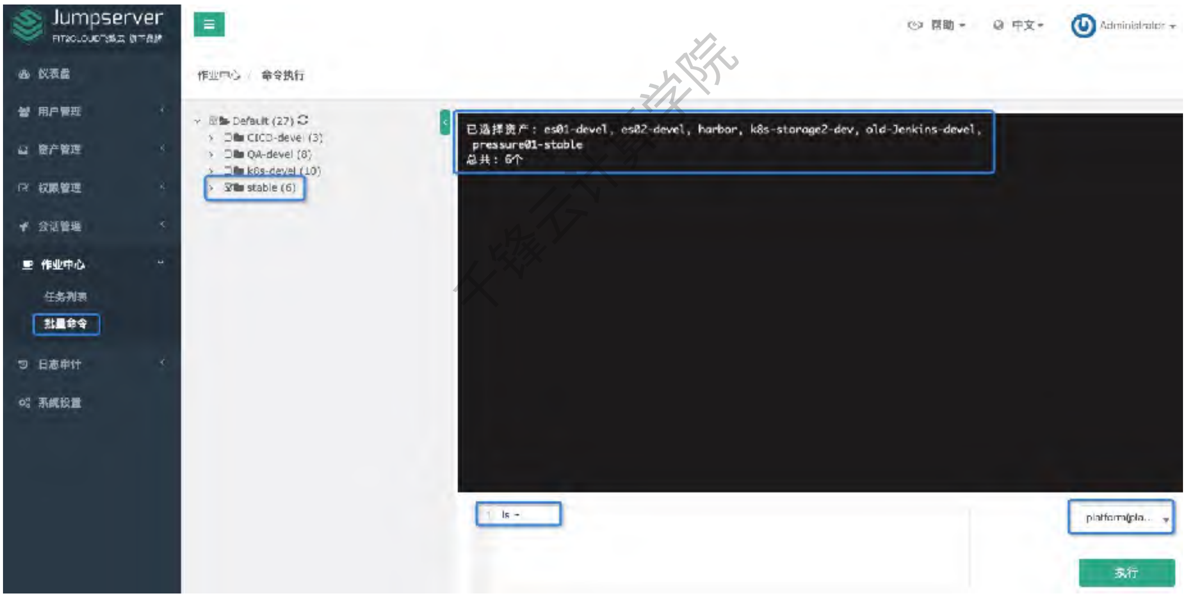
```
Meteor:~ [redacted] ssh -p2222 [redacted]i@bastion-devel.[redacted]m
caiyuanji@bastion-devel.2dupay.com's password:
            [redacted]， 欢迎使用Jumpserver开源堡垒机系统

        1) 输入 ID 进行直接登陆.
        2) 输入 部分IP、主机名、备注 进行进行搜索登录(如果唯一).
        3) 输入 / + IP, 主机名 or 备注 进行搜索, 如: /192.168.
        4) 输入 p 进行显示您有权限的主机.
        5) 输入 g 进行显示您有权限的节点.
        6) 输入 r 进行刷新最新的机器和节点信息.
        7) 输入 h 进行显示帮助.
        8) 输入 q 进行退出.
Opt> harbor
开始连接到 platform@192.168.20.31  0.1
Last login: Sat Dec 21 09:59:14 2019 from 192.168.20.32
[platform@dm-devel ~]$ sudo su -
上一次登录: 六 12月 21 03:00:10 CST 2019从 192.168.20.32pts/1 上
[root@dm-devel ~]#
```



```
Opt> 直接回车或 p 键 显示当前用户有权访问的主机列表
  ID   | 主机名                  | IP                      | 登录用户              | 备注
+------+-----------------------+-------------------------+---------------------+------
  1    | es01-devel             | 192.168.20.35          | [redacted]          |
  2    | es02-devel             | 192.168.20.37          | [redacted]          |
  3    | harbor                 | 192.168.20.31          | [redacted]          | dm-de
  4    | jenkins-qa             | 192.168.20.93          | [redacted]          |
  5    | k8s-storage2-dev       | 192.168.20.46          | [redacted]          |
  6    | old-Jenkins-devel      | 192.168.20.38          | [redacted]          |
  7    | pressure01-stable      | 192.168.20.33          | [redacted]          | press
  8    | qa91-devel             | 192.168.20.91          | [redacted]          |
  9    | qa92-devel             | 192.168.20.92          | [redacted]          |
  10   | qa94-devel             | 192.168.20.94          | [redacted]          |
  11   | qa95-devel             | 192.168.20.95          | [redacted]          |
  12   | qa97-devel             | 192.168.20.97          | [redacted]          |
  13   | qa98-devel             | 192.168.20.98          | [redacted]          |
  14   | qa99-devel             | 192.168.20.99          | [redacted]          |
页码：1，每页行数：50，总页数：1，总数量：14
Opt> 
```

作业中心

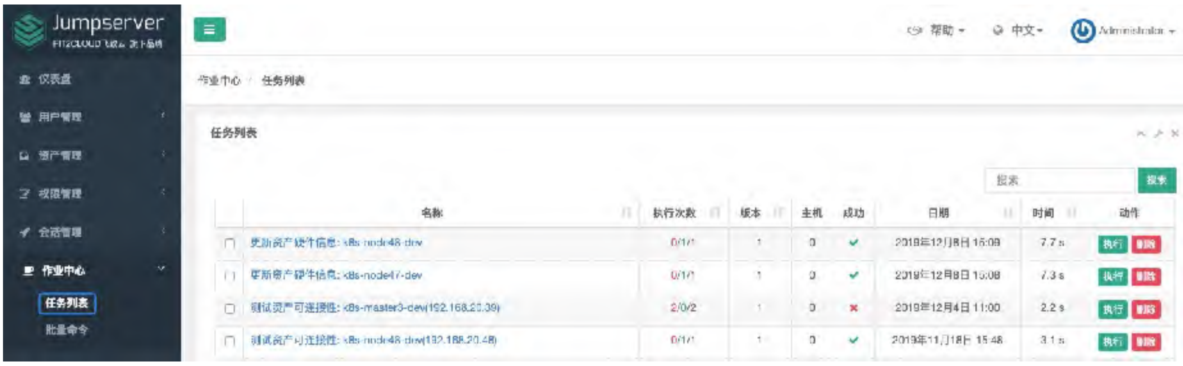
作业中心--批量命令



```
已选择资产：es01-devel, es02-devel, harbor, k8s-storage2-dev, old-Jenkins-devel,
 pressure01-stable
总共：6个
```

```
ls
```

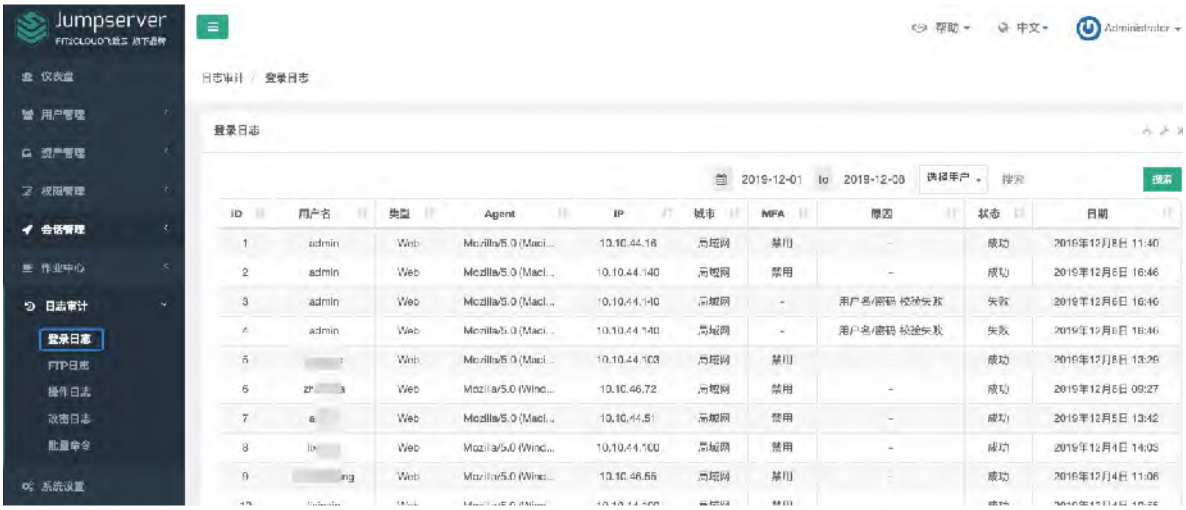
作业中心--任务列表



	名称	执行次数	版本	主机	成功	日期	时间	动作
□	更新资产硬件信息: k8s-node48-dev	0/1/1	1	0	✓	2019年12月8日 15:09	7.7 s	执行 删除
□	更新资产硬件信息: k8s-node47-dev	0/1/1	1	0	✓	2019年12月8日 15:08	7.3 s	执行 删除
□	测试资产可连接性: k8s-master3-dev(192.168.20.39)	2/0/2	1	0	✗	2019年12月4日 11:00	2.2 s	执行 删除
□	测试资产可连接性: k8s-node48-dev(192.168.20.48)	0/1/1	1	0	✓	2019年11月18日 15:48	3.1 s	执行 删除

日志审计

日志审计--登录日志



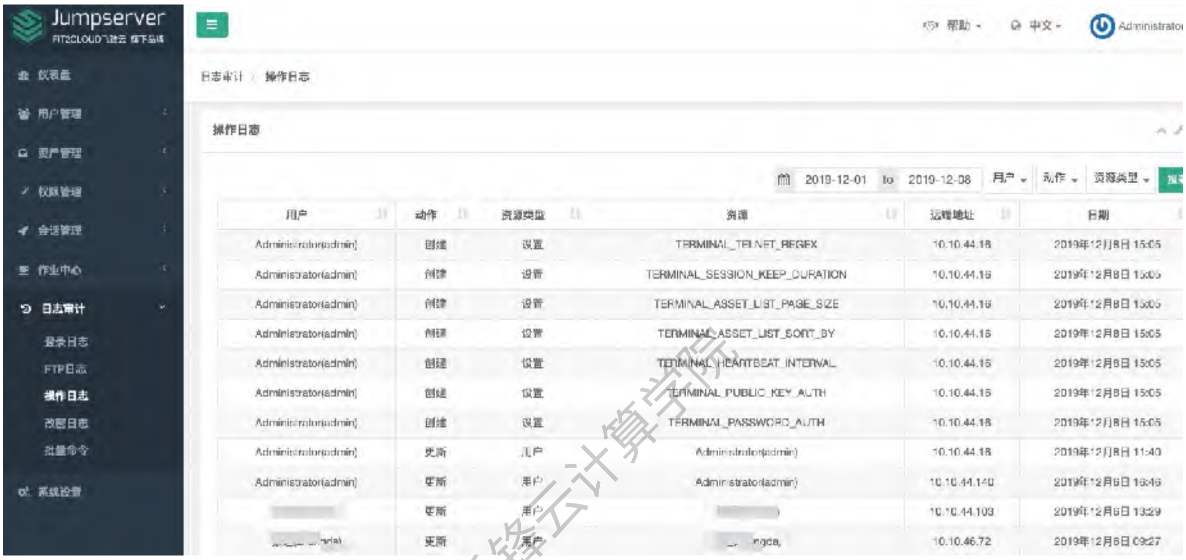
日志审计 / 登录日志

登录日志

2019-12-01 to 2019-12-08 选择用户 搜索 搜索

ID	用户名	类型	Agent	IP	城市	MFA	原因	状态	日期
1	admin	Web	Mozilla/5.0 (Maci...	10.10.44.16	局域网	禁用	-	成功	2019年12月8日 11:40
2	admin	Web	Mozilla/5.0 (Maci...	10.10.44.140	局域网	禁用	-	成功	2019年12月6日 16:46
3	admin	Web	Mozilla/5.0 (Maci...	10.10.44.140	局域网	-	用户名/密码 校验失败	失败	2019年12月6日 16:46
4	admin	Web	Mozilla/5.0 (Maci...	10.10.44.140	局域网	-	用户名/密码 校验失败	失败	2019年12月6日 16:46
5	[redacted]	Web	Mozilla/5.0 (Maci...	10.10.44.103	局域网	禁用	-	成功	2019年12月6日 13:29
6	zh[redacted]a	Web	Mozilla/5.0 (Winc...	10.10.46.72	局域网	禁用	-	成功	2019年12月6日 09:27
7	a[redacted]	Web	Mozilla/5.0 (Maci...	10.10.44.51	局域网	禁用	-	成功	2019年12月5日 13:42
8	lo[redacted]	Web	Mozilla/5.0 (Wind...	10.10.44.100	局域网	禁用	-	成功	2019年12月4日 14:03
9	[redacted]ng	Web	Mozilla/5.0 (Wind...	10.10.46.66	局域网	禁用	-	成功	2019年12月4日 11:06

日志审计--操作日志



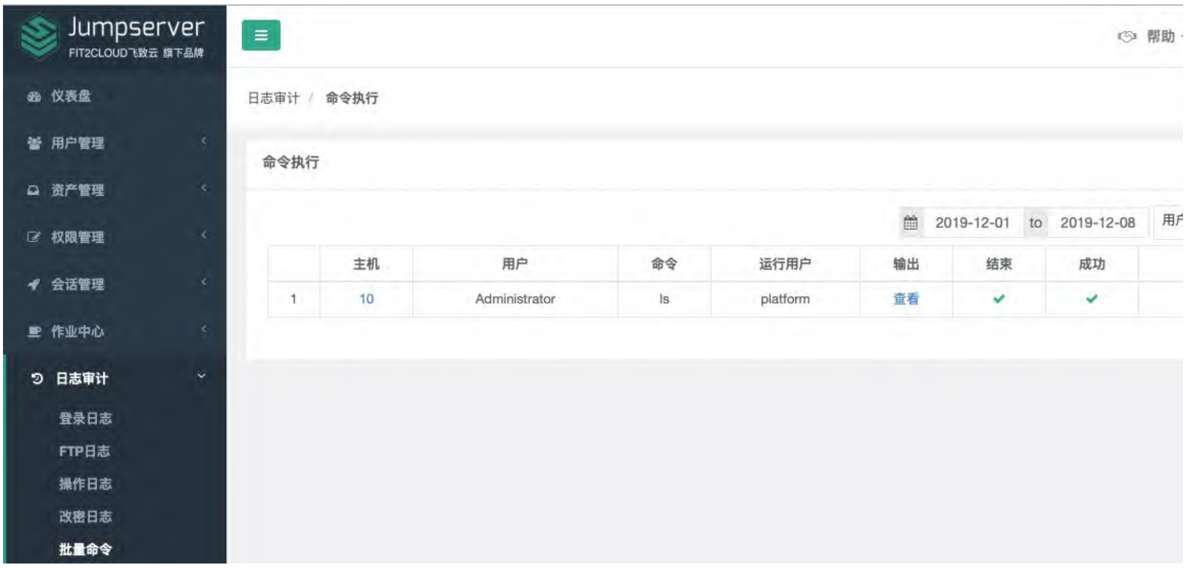
日志审计 / 操作日志

操作日志

2019-12-01 to 2019-12-08 用户 动作 资源类型

用户	动作	资源类型	资源	远端地址	日期
Administrator(admin)	创建	设置	TERMINAL_TELNET_REGEX	10.10.44.16	2019年12月8日 15:05
Administrator(admin)	创建	设置	TERMINAL_SESSION_KEEP_DURATION	10.10.44.16	2019年12月8日 15:05
Administrator(admin)	创建	设置	TERMINAL_ASSET_LIST_PAGE_SIZE	10.10.44.16	2019年12月8日 15:05
Administrator(admin)	创建	设置	TERMINAL_ASSET_LIST_SORT_BY	10.10.44.16	2019年12月8日 15:05
Administrator(admin)	创建	设置	TERMINAL_HEARTBEAT_INTERVAL	10.10.44.16	2019年12月8日 15:05
Administrator(admin)	创建	设置	TERMINAL_PUBLIC_KEY_AUTH	10.10.44.16	2019年12月8日 15:05
Administrator(admin)	创建	设置	TERMINAL_PASSWORD_AUTH	10.10.44.16	2019年12月8日 15:05
Administrator(admin)	更新	用户	Administrator(admin)	10.10.44.16	2019年12月8日 11:40
Administrator(admin)	更新	用户	Administrator(admin)	10.10.44.140	2019年12月6日 16:46
[redacted]	更新	用户	[redacted]	10.10.44.103	2019年12月6日 13:29
[redacted]nda)	更新	用户	[redacted]ngda,	10.10.46.72	2019年12月6日 09:27

日志审计--批量命令



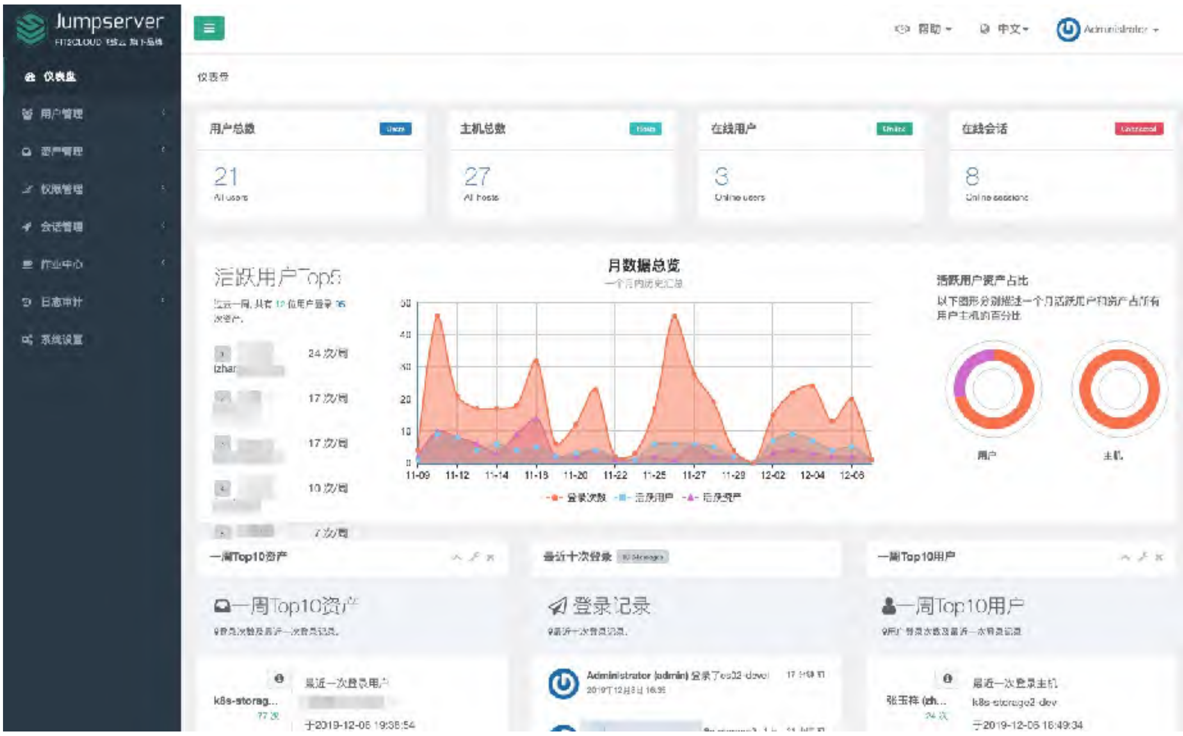
日志审计 / 命令执行

命令执行

2019-12-01 to 2019-12-08 用户

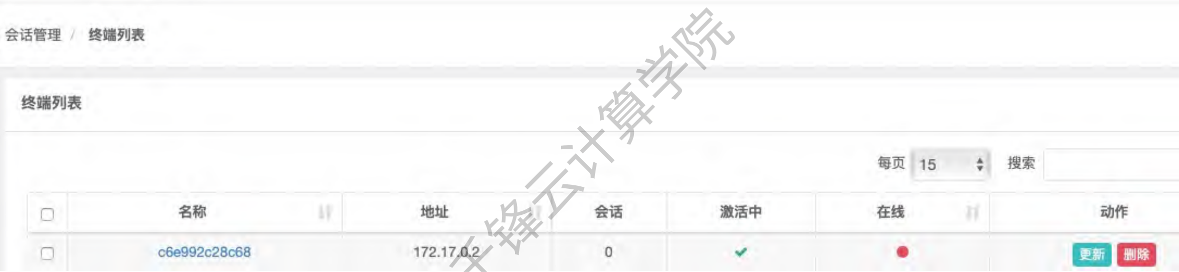
	主机	用户	命令	运行用户	输出	结束	成功
1	10	Administrator	ls	platform	查看	✓	✓

仪表盘



14、排错

（1）koko 不在线



会话管理 / 终端列表

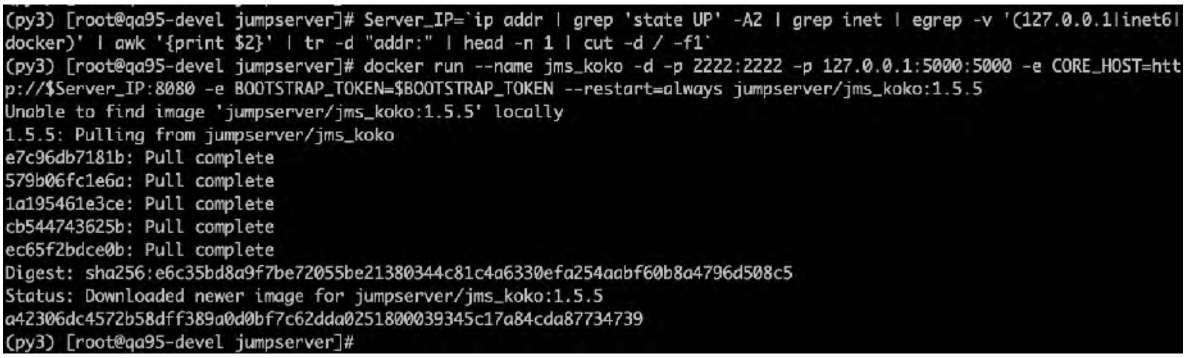
终端列表

每页 15 搜索

☐	名称	地址	会话	激活中	在线	动作
☐	c6e992c28c68	172.17.0.2	0	✓	●	更新 删除

原因：版本不匹配

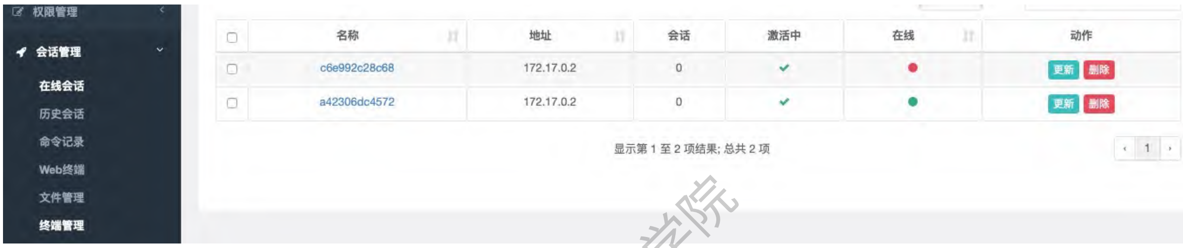
解决过程如下：



```
(py3) [root@qa95-devel jumpserver]# Server_IP=`ip addr | grep 'state UP' -A2 | grep inet | egrep -v '(127.0.0.1|inet6|
docker)' | awk '{print $2}' | tr -d "addr:" | head -n 1 | cut -d / -f1`
(py3) [root@qa95-devel jumpserver]# docker run --name jms_koko -d -p 2222:2222 -p 127.0.0.1:5000:5000 -e CORE_HOST=htt
p://$Server_IP:8080 -e BOOTSTRAP_TOKEN=$BOOTSTRAP_TOKEN --restart=always jumpserver/jms_koko:1.5.5
Unable to find image 'jumpserver/jms_koko:1.5.5' locally
1.5.5: Pulling from jumpserver/jms_koko
e7c96db7181b: Pull complete
579b06fc1e6a: Pull complete
1a195461e3ce: Pull complete
cb544743625b: Pull complete
ec65f2bdce0b: Pull complete
Digest: sha256:e6c35bd8a9f7be72055be21380344c81c4a6330efa254aabf60b8a4796d508c5
Status: Downloaded newer image for jumpserver/jms_koko:1.5.5
a42306dc4572b58dff389a0d0bf7c62dda0251800039345c17a84cda87734739
(py3) [root@qa95-devel jumpserver]#
```

```
(py3) [root@qa95-devel jumpserver]# Server_IP=`ip addr | grep 'state UP' -A2 |
grep inet | egrep -v '(127.0.0.1|inet6|docker)' | awk '{print $2}' | tr -d
"addr:" | head -n 1 | cut -d / -f1`
(py3) [root@qa95-devel jumpserver]# docker run --name jms_koko -d -p 2222:2222 -
p 127.0.0.1:5000:5000 -e CORE_HOST=http://$Server_IP:8080 -e
BOOTSTRAP_TOKEN=$BOOTSTRAP_TOKEN --restart=always jumpserver/jms_koko:1.5.5
Unable to find image 'jumpserver/jms_koko:1.5.5' locally
1.5.5: Pulling from jumpserver/jms_koko
e7c96db7181b: Pull complete
579b06fc1e6a: Pull complete
1a195461e3ce: Pull complete
cb544743625b: Pull complete
ec65f2bdce0b: Pull complete
Digest: sha256:e6c35bd8a9f7be72055be21380344c81c4a6330efa254aabf60b8a4796d508c5
Status: Downloaded newer image for jumpserver/jms_koko:1.5.5
a42306dc4572b58dff389a0d0bf7c62dda0251800039345c17a84cda87734739
(py3) [root@qa95-devel jumpserver]#
```

然后koko注册成功，并处于在线状态：



□	名称	地址	会话	激活中	在线	动作
□	c6e992c28c68	172.17.0.2	0	✓	●	更新 删除
□	a42306dc4572	172.17.0.2	0	✓	●	更新 删除

显示第 1 至 2 项结果; 总共 2 项

（ 2 ）Guacamole 注册失败

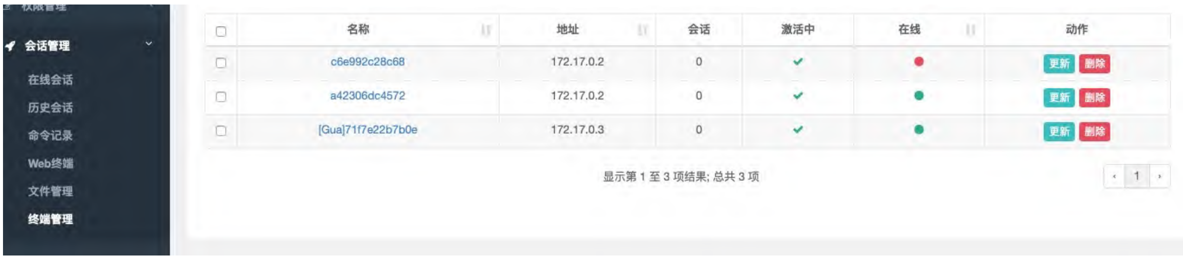
原因：版本不匹配

解决过程如下：

```
(py3) [root@qa95-devel jumpserver]# docker stop jms_guacamole
jms_guacamole
(py3) [root@qa95-devel jumpserver]# docker rm jms_guacamole
jms_guacamole
(py3) [root@qa95-devel jumpserver]# docker run --name jms_guacamole -d -p
127.0.0.1:8081:8080 -e JUMPSERVER_SERVER=http://$Server_IP:8080 -e
BOOTSTRAP_TOKEN=$BOOTSTRAP_TOKEN --restart=always jumpserver/jms_guacamole:1.5.5
Unable to find image 'jumpserver/jms_guacamole:1.5.5' locally
1.5.5: Pulling from jumpserver/jms_guacamole
ab5ef0e58194: Pull complete
edf7bc06322e: Pull complete
2034ec367e45: Pull complete
e75756b89a95: Pull complete
f04c5d071413: Pull complete
2599c3a6a821: Pull complete
52a073ddf64c: Pull complete
805616d72c12: Pull complete
3c40529b36f6: Pull complete
3044f8f99b07: Pull complete
d97561b081f3: Pull complete
Digest: sha256:667651fd4fe9836d6c4121c66cde25095dce966e9610035da512af25cbe00b79
Status: Downloaded newer image for jumpserver/jms_guacamole:1.5.5
71f7e22b7b0e1687b55f79063b3fe9c699286157c1efd361e4a68ee4ad141a95
(py3) [root@qa95-devel jumpserver]#  docker ps
```

```
CONTAINER ID        IMAGE                             COMMAND             CREATED
            STATUS              PORTS
   NAMES
71f7e22b7b0e        jumpserver/jms_guacamole:1.5.5   "./entrypoint.sh"   4
minutes ago        Up 4 minutes        127.0.0.1:8081->8080/tcp
        jms_guacamole
a42306dc4572        jumpserver/jms_koko:1.5.5        "./entrypoint.sh"   29
minutes ago       Up 29 minutes       0.0.0.0:2222->2222/tcp, 127.0.0.1:5000-
>5000/tcp    jms_koko
(py3) [root@qa95-devel jumpserver]#
```

然后Guacamole注册成功，如下所示：



□	名称	地址	会话	激活中	在线	动作
□	c6e992c28c68	172.17.0.2	0	✓	●	更新 删除
□	a42306dc4572	172.17.0.2	0	✓	●	更新 删除
□	[Gua]71f7e22b7b0e	172.17.0.3	0	✓	●	更新 删除

显示第 1 至 3 项结果; 总共 3 项

排错参考：https://docs.jumpserver.org/zh/docs/faq.html

五、远程管理实战

1、使用ssh管理远程机器

2、部署并使用jumpserver服务器