

## Jumpserver堡垒机管理

### 一、堡垒机简介

#### 1.运维常见背黑锅场景

#### 2.背黑锅的主要原因

#### 3.解决背黑锅的方法

- 1、统一入口、规范管理
- 2、利用手机APP动态口令等验证机制
- 3、托管服务器密码，实现自动改密
- 4、事中控制，防止违规操作
- 5、精细化审计，追溯整个运维过程

### 二、Jumpserver简介

#### 1、支持的操作系统

#### 2、功能介绍

#### 3、Jumpserver组件说明

Jumpserver:

Coco: .

Luna;

Guacamole:

Nginx:

Redis:

Mysql;

#### 4、Jumpserver功能说明

### 三、部署Jumpserver环境

### 四、jumpserver配置应用

- 1.系统配置
- 2.创建管理用户
- 3.创建资产/添加资产
- 4.创建系统用户
- 5.用户管理
- 6.资产授权
- 7.用新建的账号登录跳板机
- 8.命令行xhell连接服务器

### Jumpserver堡垒机管理

#### 一、堡垒机简介

- 1.运维常见背黑锅场景
- 2.背黑锅的主要原因
- 3.解决背黑锅的方法
  - 1、统一入口、规范管理
  - 2、利用手机APP动态口令等验证机制
  - 3、托管服务器密码，实现自动改密
  - 4、事中控制，防止违规操作
  - 5、精细化审计，追溯整个运维过程

#### 二、Jumpserver简介

- 1、支持的操作系统
- 2、功能介绍

### 3、Jumpserver组件说明

Jumpserver:

Coco: .

Luna;

Guacamole:

Nginx:

Redis:

Mysql;

### 4、Jumpserver功能说明

### 三、部署Jumpserver环境

## Jumpserver堡垒机管理

### 一、堡垒机简介

#### 1.运维常见背黑锅场景

- 1、由于不明身份用户利用远程运维通道攻击服务器造成业务系统出现异常：  
但是运维人员无法明确攻击来源，那么领导很生气、后果很严重。
- 2、只有张三能管理的服务器，被李四登录过并且做了违规操作：  
但是没有证据是李四登录的，那么张三只能背黑锅了。
- 3、运维人员不小心泄露了服务器的密码。一旦发生安全事故，那么后果不堪设想。
- 4、某服务器的重要数据被窃。但是数据文件无法挽回，  
那么面临的是无法估量的经济损失。

运维工作中由于远程登录来源身份不明、越权操作、密码泄露、数据被窃、违规操作等因素都可能会使运营的业务系统面临严重威胁，一旦发生事故，如果不能快速定位事故原因，运维人员往往就会背黑锅

#### 2.背黑锅的主要原因

其实运维工作，出现各种问题是在所难免的不仅要有很好的分析处理能力，而且还要避免问题再次发生。

要清楚认识到出现问题的真实原因：

- 没有规范管理，人与服务器之间的界限不清晰
- 没有实名机制，登录服务器前没有实名验证
- 没有密码托管，服务器的密码太多，很难做到定期修改，自己保管怕丢失
- 没有操作预警，对高危、敏感的操作无法做到事前防御
- 没有传输控制，对重要服务器无法控制文件传输
- 没有回溯过程，不能完整还原运维过程

### 3.解决背黑锅的方法

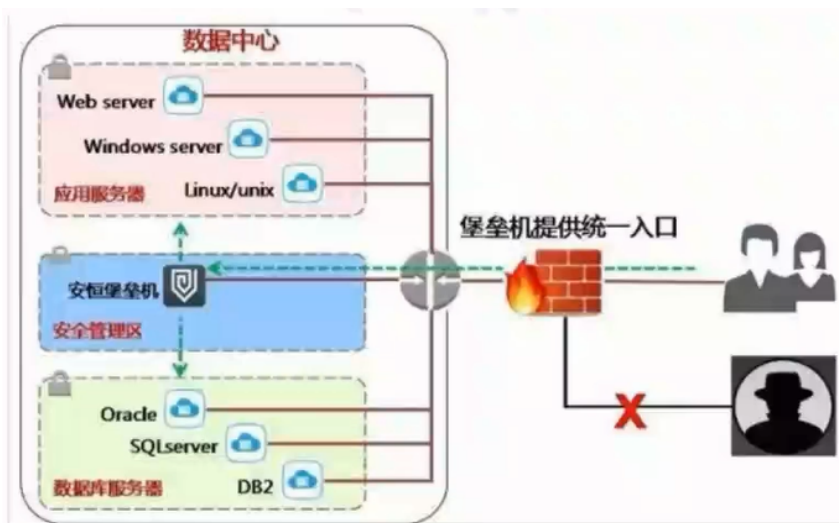
作为运维人员，如何摆脱以上背黑锅的尴尬局面呢？

也许堡垒机是一个破解此局面的方法。

#### 1、统一入口、规范管理

提供统一入口，所有运维人员只能登录堡垒机才能访问服务器，梳理“人与服务器”之

间的关系，防止越权登录



#### 2、利用手机APP动态口令等验证机制

采用手机APP动态口令、OTP 动态令牌、USBKEY、短信口令等双因素身份实名鉴别机制防止密码被暴力破解，解决访问身份模糊的问题。



### 3、托管服务器密码，实现自动改密

通过堡垒机定期自动修改服务器的密码，解决手工修改密码、密码泄露和记住密码的烦恼。

- 1、可自动修改Windows、Linux、 Unix、 网络设备等操作系统的密码
- 2、可以设置周期或指定时间执行改密任务
- 3、可设定密码的复杂度、随机密码、指定密码、固定密码格式等
- 4、可通过邮件、SFTP、 FTP 方式自动发送密码文件给管理员
- 5、提供密码容错机制:改密前自动备份、备份失败不改密、改密后自动备份、自动恢复密码等

### 4、事中控制，防止违规操作

作为运维人员，如何摆脱以上背黑锅的尴尬局面呢?也许堡垒机是一个破解此局面的必杀技。

- 1、通过命令控制策略，拦截高危、敏感的命令
- 2、通过命令审核策略，审批需要执行但又不能随意执行的命令
- 3、通过文件传输控制策略，防止数据、文件的泄露



### 5、精细化审计，追溯整个运维过程

堡垒机要做到文件记录、视频回放等精细化完整审计，快速定位运维过程：

- 1、不仅要对所有操作会话的在线监控、实时阻断、日志回放、起止时间、来源用户、来源地址、目标地址、协议、命令、操作(如对文件的上传、下载、删除、修改等操作等)等行为记录。
- 2、还要能保存SFTP/FTP/SCP/RDP/RZ/SZ传输的文件为上传恶意文件、拖库、窃取数据等危

险行为起到了追踪依据。

## 二、Jumpserver简介

Jumpserver (官网<http://www.jumpserver.org/>)是全球首款完全开源的堡垒机(跳板机),

使用GNU GPL v2.0开源协议,是符合4A(认证Authentication、授权Authorization、记账Accounting、审计Audit)的专业运维审计系统。Jumpserver使用Python/ Django进行开发,遵循Web2.0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、无并发限制。

基于ssh协议来管理。客户端无需安装agent。助力互联网企业高效的用户、资产、权限、审计管理。而且管理界面是中文的,适应窗口功能还是使用便捷度上来讲,都是非常不错的选择。

### 1、支持的操作系统

- Redhat CentOS
- Debian
- SUSE Ubuntu
- FreeBSD
- 其他ssh协议硬件设备(如交换机).

### 2、功能介绍

1. 精确记录操作命令
2. 支持批量文件上传下载
3. 支持主机搜索登录
4. 支持批量命令执行(Ansible完成)
5. 支持WebTerminal连接主机
6. 支持Web端批量命令执行
7. 支持录像回放
8. 支持硬件信息如cpu  
内存等抓取
9. 支持资产Excel导入导出
10. 支持资产批量更改
11. 支持系统用户的批量推送(Ansible实现)
12. 支持用户,主机,用户组,主机组,系统用户混合细颗粒授权
13. 支持sudo管理
14. 支持命令统计和命令搜索

- 15. 支持上传下载文件审计
- 16. 支持终止用户连接
- 17. 支持各种搜索
- 18. 其他

### 3、Jumpserver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、资产授权等操作默认端口为8080/tcp 配置文件在jumpserver/config.yml

**Coco: .**

为SSHServer和Web Terminal Server。用户可以通过使用自己的账户登录SSH 或者WebTerminal直接访问被授权的资产。不需要知道服务器的账户密码, 默认SSH 端口为2222/tcp, 默认Web Terminal端口为5000/tcp 配置文件在coco/config.yml

**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), 默认端口为8081/tcp

**Nginx:**

默认端口为80/tcp, 前端代理服务

**Redis:**

默认端口为6379/tcp, 数据库缓存服务

**Mysql;**

默认端口为3306/tcp, 数据库服务

### 4、Jumpserver功能说明

Jumpserver提供的堡垒机必备功能		
身份验证Authentication	登录认证 多因子认证	资源统一登录和认证 LDAP认证 支持OpenID,实现单点登录 MFA ( Google Authenticator )
账号管理Account	集中账号管理 统一密码管理 批量密码变更(X-PACK) 多云环境的资产纳管(X-PACK)	管理用户管理 系统用户管理 资产密码托管 自动生成密码 密码自动推送 密码过期设置 定期批量修改密码 生成随机密码

		对私有云、公有云及广域网统一纳管
授权控制Authorization	资产授权管理 组织管理(X-PACK) 多维度授权 指令限制 统一文件传输 文件管理	资产树 资产或资产组灵活授权 节点内资产自动继承授权 实现多租户管理，权限隔离 可对用户、用户组或系统角色授权 限制特权指令使用，支持黑白名单 SFTP文件上传/下载 Web SFTP文件管理
安全审计Audit	会话管理 录像管理 指令审计 文件传输审计	在线会话管理 历史会话管理 Linux录像支持 Windows录像支持  指令记录 上传/下载记录审计

### 三、部署Jumpserver环境

官网推荐安装环境

CPU:64位双核处理器

内存：4G

数据库：mysql 版本大于等于5.6 mariadb版本大于等于5.5.6

操作系统	IP地址	主机名	角色
CentOS7.7	192.168.200.111	jumpserver	堡垒机
CentOS7.7	192.168.200.108	server108	被管理服务器
CentOS7.7	192.168.200.109	server109	被管理服务器

环境：

- 系统: CentOS 7
- IP: 192.168.244.144
- 目录: /opt
- 数据库: mariadb
- 代理: nginx

### CentOS 7 安装文档

```
$ yum update -y
```

# 防火墙 与 selinux 设置说明, 如果已经关闭了 防火墙 和 Selinux 的用户请跳过设置

```
$ systemctl start firewalld
```

```
$ 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
□ koko
  --permanent 永久生效, 没有此参数重启后失效

$ firewall-cmd --reload # 重新载入规则

$ setenforce 0
$ sed -i "s/SELINUX=enforcing/SELINUX=disabled/g" /etc/selinux/config

# 安装依赖包
$ yum -y install wget gcc epel-release git

# 安装 Redis, JumpServer 使用 Redis 做 cache 和 celery broke
$ yum -y install redis
$ systemctl enable redis
$ systemctl start redis

# 安装 MySQL, 如果不使用 Mysql 可以跳过相关 Mysql 安装和配置, 支持sqlite3, mysql,
postgres等
$ yum -y install mariadb mariadb-devel mariadb-server MariaDB-shared # centos7
下叫mariadb, 用法与mysql一致
$ systemctl enable mariadb
$ systemctl start mariadb
# 创建数据库 JumpServer 并授权
$ DB_PASSWORD=`cat /dev/urandom | tr -dc A-Za-z0-9 | head -c 24` # 生成随机数
数据库密码
$ echo -e "\033[31m 你的数据库密码是 $DB_PASSWORD \033[0m"
$ mysql -uroot -e "create database jumpserver default charset 'utf8' collate
'utf8_bin'; grant all on jumpserver.* to 'jumpserver'@'127.0.0.1' identified by
'$DB_PASSWORD'; flush privileges;"

# 安装 Nginx, 用作代理服务器整合 JumpServer 与各个组件
$ vi /etc/yum.repos.d/nginx.repo

[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/7/$basearch/
gpgcheck=0
enabled=1

$ yum -y install nginx
$ systemctl enable nginx

# 安装 Python3.6
$ yum -y install python36 python36-devel

# 配置并载入 Python3 虚拟环境
```

```
$ cd /opt
$ python3.6 -m venv py3 # py3 为虚拟环境名称, 可自定义
$ source /opt/py3/bin/activate # 退出虚拟环境可以使用 deactivate 命令

# 看到下面的提示符代表成功, 以后运行 JumpServer 都要先运行以上 source 命令, 载入环境后默认以下所有命令均在该虚拟环境中运行
(py3) [root@localhost py3]

# 下载 JumpServer
$ cd /opt/
$ git clone --depth=1 https://github.com/jumpserver/jumpserver.git

# 安装依赖 RPM 包
$ yum -y install $(cat /opt/jumpserver/requirements/rpm_requirements.txt)

# 安装 Python 库依赖
$ pip install wheel
$ pip install --upgrade pip setuptools
$ pip install -r /opt/jumpserver/requirements/requirements.txt
# 修改 JumpServer 配置文件
$ cd /opt/jumpserver
$ cp config_example.yml config.yml

$ SECRET_KEY=`cat /dev/urandom | tr -dc A-Za-z0-9 | head -c 50` # 生成随机 SECRET_KEY
$ echo "SECRET_KEY=$SECRET_KEY" >> ~/.bashrc
$ BOOTSTRAP_TOKEN=`cat /dev/urandom | tr -dc A-Za-z0-9 | head -c 16` # 生成随机 BOOTSTRAP_TOKEN
$ echo "BOOTSTRAP_TOKEN=$BOOTSTRAP_TOKEN" >> ~/.bashrc

$ sed -i "s/SECRET_KEY:/SECRET_KEY: $SECRET_KEY/g" /opt/jumpserver/config.yml
$ sed -i "s/BOOTSTRAP_TOKEN:/BOOTSTRAP_TOKEN: $BOOTSTRAP_TOKEN/g" /opt/jumpserver/config.yml
$ sed -i "s/# DEBUG: true/DEBUG: false/g" /opt/jumpserver/config.yml
$ sed -i "s/# LOG_LEVEL: DEBUG/LOG_LEVEL: ERROR/g" /opt/jumpserver/config.yml
$ sed -i "s/# SESSION_EXPIRE_AT_BROWSER_CLOSE: false/SESSION_EXPIRE_AT_BROWSER_CLOSE: true/g" /opt/jumpserver/config.yml
$ sed -i "s/DB_PASSWORD: /DB_PASSWORD: $DB_PASSWORD/g" /opt/jumpserver/config.yml

$ echo -e "\033[31m 你的SECRET_KEY是 $SECRET_KEY \033[0m"
$ echo -e "\033[31m 你的BOOTSTRAP_TOKEN是 $BOOTSTRAP_TOKEN \033[0m"

$ vi config.yml # 确认内容有没有错误
# SECURITY WARNING: keep the secret key used in production secret!
# 加密秘钥 生产环境中请修改为随机字符串, 请勿外泄, PS: 纯数字不可以
SECRET_KEY:
```

```
# SECURITY WARNING: keep the bootstrap token used in production secret!
# 预共享Token koko和guacamole用来注册服务账号, 不在使用原来的注册接受机制
BOOTSTRAP_TOKEN:

# Development env open this, when error occur display the full process track,
# Production disable it
# DEBUG 模式 开启DEBUG后遇到错误时可以看到更多日志
DEBUG: false

# DEBUG, INFO, WARNING, ERROR, CRITICAL can set. See
# https://docs.djangoproject.com/en/1.10/topics/logging/
# 日志级别
LOG_LEVEL: ERROR
# LOG_DIR:

# Session expiration setting, Default 24 hour, Also set expired on on browser close
# 浏览器Session过期时间, 默认24小时, 也可以设置浏览器关闭则过期
# SESSION_COOKIE_AGE: 86400
SESSION_EXPIRE_AT_BROWSER_CLOSE: true

# Database setting, Support sqlite3, mysql, postgres ....
# 数据库设置
# See https://docs.djangoproject.com/en/1.10/ref/settings/#databases

# SQLite setting:
# 使用单文件sqlite数据库
# DB_ENGINE: sqlite3
# DB_NAME:

# MySQL or postgres setting like:
# 使用Mysql作为数据库
DB_ENGINE: mysql
DB_HOST: 127.0.0.1
DB_PORT: 3306
DB_USER: jumpserver
DB_PASSWORD:
DB_NAME: jumpserver

# When Django start it will bind this host and port
# ./manage.py runserver 127.0.0.1:8080
# 运行时绑定端口
HTTP_BIND_HOST: 0.0.0.0
HTTP_LISTEN_PORT: 8080

# Use Redis as broker for celery and web socket
# Redis配置
```

```
REDIS_HOST: 127.0.0.1
REDIS_PORT: 6379
# REDIS_PASSWORD:
# REDIS_DB_CELERY: 3
# REDIS_DB_CACHE: 4

# Use OpenID authorization
# 使用OpenID 来进行认证设置
# BASE_SITE_URL: http://localhost:8080
# AUTH_OPENID: false # True or False
# AUTH_OPENID_SERVER_URL: https://openid-auth-server.com/
# AUTH_OPENID_REALM_NAME: realm-name
# AUTH_OPENID_CLIENT_ID: client-id
# AUTH_OPENID_CLIENT_SECRET: client-secret

# OTP settings
# OTP/MFA 配置
# OTP_VALID_WINDOW: 0
# OTP_ISSUER_NAME: Jumpserver
# 运行 JumpServer
$ cd /opt/jumpserver
$ ./jms start -d # 后台运行使用 -d 参数./jms start -d
# 新版本更新了运行脚本, 使用方式./jms start|stop|status all 后台运行请添加 -d 参数

$ wget -O /usr/lib/systemd/system/jms.service
https://demo.jumpserver.org/download/shell/centos/jms.service
$ chmod 755 /usr/lib/systemd/system/jms.service
$ systemctl enable jms # 配置自启
# 安装 docker 部署 koko 与 guacamole
$ yum install -y yum-utils device-mapper-persistent-data lvm2
$ yum-config-manager --add-repo http://mirrors.aliyun.com/docker-
ce/linux/centos/docker-ce.repo
$ yum makecache fast
$ rpm --import https://mirrors.aliyun.com/docker-ce/linux/centos/gpg
$ yum -y install docker-ce
$ systemctl enable docker
$ mkdir /etc/docker
$ wget -O /etc/docker/daemon.json
http://demo.jumpserver.org/download/docker/daemon.json
$ systemctl restart docker

# 允许 容器ip 访问宿主 8080 端口, (容器的 ip 可以进入容器查看)
$ firewall-cmd --permanent --add-rich-rule="rule family="ipv4" source
address="172.17.0.0/16" port protocol="tcp" port="8080" accept"
$ firewall-cmd --reload
# 172.17.0.x 是docker容器默认的IP池, 这里偷懒直接授权ip段了, 可以根据实际情况单独授
权IP
```

```

# 获取当前服务器 IP
$ 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`
$ echo -e "\033[31m 你的服务器IP是 $Server_IP \033[0m"

# http://<Jumpserver_url> 指向 jumpserver 的服务端口, 如
http://192.168.244.144:8080
# BOOTSTRAP_TOKEN 为 Jumpserver/config.yml 里面的 BOOTSTRAP_TOKEN
$ 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
-e LOG_LEVEL=ERROR --restart=always jumpserver/jms_koko:1.5.8
$ 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 -e GUACAMOLE_LOG_LEVEL=ERROR --
restart=always jumpserver/jms_guacamole:1.5.8
# 安装 Web Terminal 前端: Luna 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
$ cd /opt
$ wget https://github.com/jumpserver/luna/releases/download/1.5.8/luna.tar.gz

# 如果网络有问题导致下载无法完成可以使用下面地址
$ wget https://demo.jumpserver.org/download/luna/1.5.8/luna.tar.gz

$ tar xf luna.tar.gz
$ chown -R root:root luna
# 配置 Nginx 整合各组件
$ rm -rf /etc/nginx/conf.d/default.conf
$ vi /etc/nginx/conf.d/jumpserver.conf

server {
    listen 80;
    # server_name _;

    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;  
}
```

```

}
# 运行 Nginx
$ nginx -t # 确保配置没有问题, 有问题请先解决
$ systemctl start nginx

# 访问 http://192.168.244.144 (注意 没有 :8080 通过 nginx 代理端口进行访问)
# 默认账号: admin 密码: admin 到会话管理-终端管理 接受 koko Guacamole 等应用的注册
# 测试连接
$ ssh -p2222 admin@192.168.244.144
$ sftp -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
$ sftp admin@192.168.244.144 2222
  密码: admin
如果能登陆代表部署成功

# sftp默认上传的位置在资产的 /tmp 目录下
# windows拖拽上传的位置在资产的 Guacamole RDP上的 G 目录下

```

```

Administrator, 欢迎使用Jumpserver开源堡垒机系统

1) 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 进行搜索, 如: /192.168.
3) 输入 p 进行显示您有权限的主机.
4) 输入 g 进行显示您有权限的节点.
5) 输入 d 进行显示您有权限的数据库.
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.

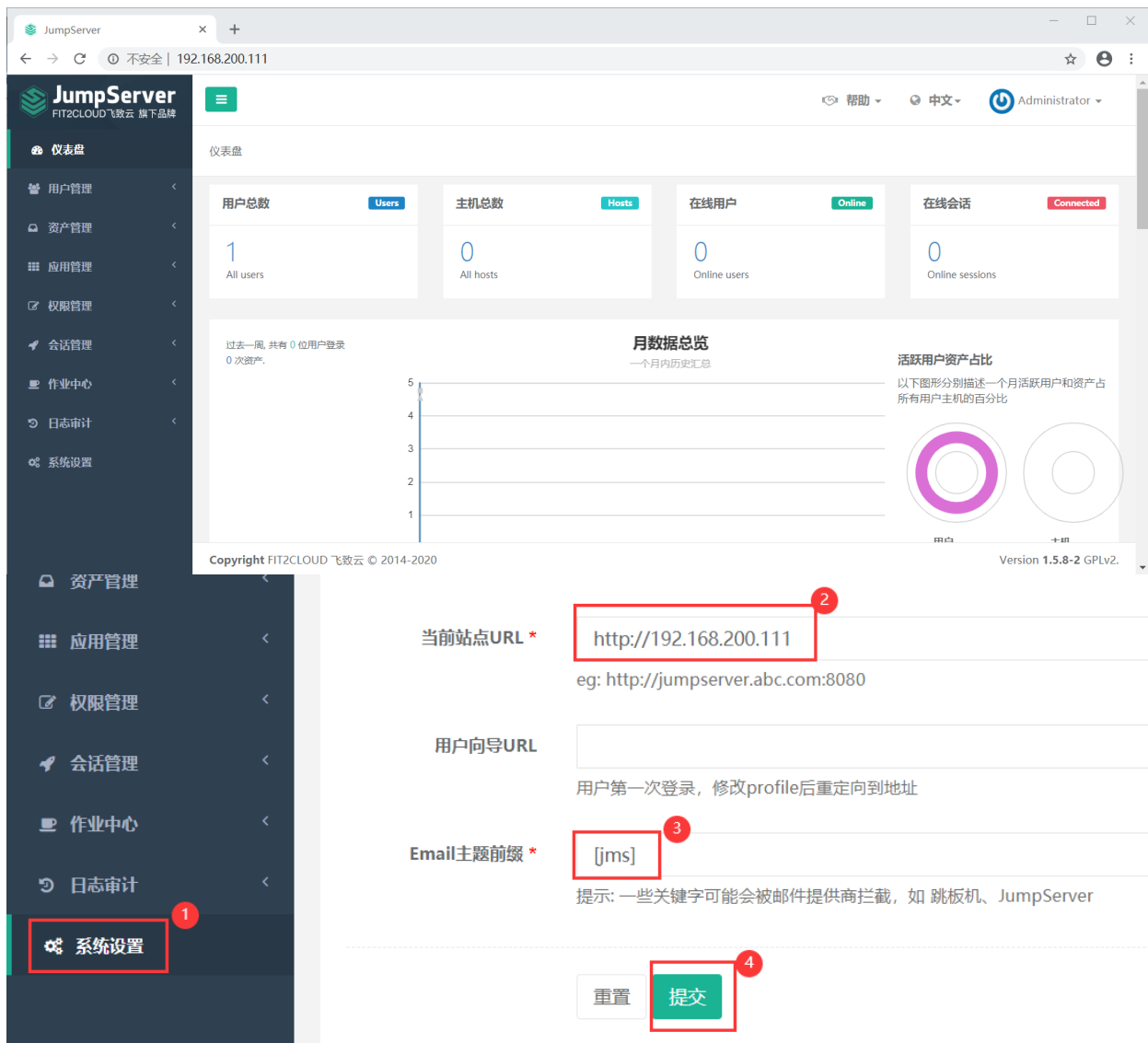
Opt> 

```

不容易啊, 终于从坑里爬出来了

## 四、jumpserver配置应用

### 1.系统配置



想哭，为什么不能保存快照

为什么手贱的要去拍快照哎

网易授权码：FESKUWIUCPODHMPR

完了再接着弄吧

还好还好，就是保存慢了点，没太大问题

## 2.创建管理用户



用户管理

资产管理

资产列表

网域列表

管理用户

系统用户

标签管理

命令过滤

平台列表

应用管理

权限管理

会话管理

创建管理用户

名称 \*

root

用户名

root

密码

.....

123456

密码或密钥密码

ssh私钥

选择文件

未选择任何文件

备注

重置

提交

服务器上的管理员信息，所有后台管理服务器的root用户及密码

### 3.创建资产/添加资产

用户管理

资产管理

资产列表

网域列表

管理用户

系统用户

标签管理

命令过滤

平台列表

应用管理

权限管理

会话管理

作业中心

日志审计

系统设置

创建资产

基本

主机名 \*

server107

IP \*

192.168.200.107

系统平台 \*

Linux

Windows 2016的RDP协议与之前不同，如果是请设置

公网IP

网域

网域

如果有多多个的互相隔离的网络，设置资产属于的网域，使用网域网关跳转登录

协议组

ssh

22

-

+

认证

管理用户 \*

root

root或其他拥有NOPASSWD: ALL权限的用户，如果是windows或其它硬件可以随意设置一个，更多信息查看左侧`管理用户`菜单

节点

节点 \*

× Default

标签管理

未找到结果

标签

其它

备注

被管理服务器107

激活

重置

提交

以同样的方式创建109服务器

创建资产

每页 15 搜索

标签 CSV

<input type="checkbox"/>	主机名 ↑↓	IP ↑↓	硬件	可连接	动作
<input type="checkbox"/>	server107	192.168.200.107	1 Core 974.0 M 100.0 G	<div></div>	<div>更新</div> <div>删除</div>
<input type="checkbox"/>	server109	192.168.200.109	1 Core 974.0 M 100.0 G	<div></div>	<div>更新</div> <div>删除</div>

批量删除

提交

显示第 1 至 2 项结果; 总共 2 项

< 1 >

## 4.创建系统用户

用来管理批量服务器的账号

仪表盘

用户管理

资产管理

资产列表

网域列表

管理用户

系统用户

标签管理

命令过滤

资产管理 / 创建系统用户

创建系统用户

基本

名称 \*

sysadmin

登录模式 \*

自动登录

用户名

sysadmin

如果选择手动登录模式，用户名和密码可以不填写

## 5.用户管理

给每一个运维创建的用户

都有哪些用户可以登录jumpserver平台

用户管理

用户列表

用户组

资产管理

应用管理

权限管理

会话管理

用户列表

创建用户

<input type="checkbox"/>	名称 ↑↓	用户名 ↑↓	角色 ↑↓
<input type="checkbox"/>	Administrator	admin	管理员
<input type="checkbox"/>	soifa	soifa	用户

批量删除

提交

显示第 1 至 2

用户详情

用户授权

soifa



名称:	soifa
用户名:	soifa
邮件:	liruifangtote@163.com
角色:	用户
多因子认证:	禁用
用户来源:	数据库
失效日期:	2090-04-22 05:50:00
创建者:	Administrator
创建日期:	2020-05-9 05:53:21
最后登录:	
最后更新密码:	2020-05-9 05:53:22
备注:	

## 6.资产授权

排错从这张图入手吧 ~

创建权限规则

基本

名称 \* server107

用户

用户 × soifa(soifa)

用户组 用户组

资产

资产 × server107(192.168.200.107) × server109(192.168.200.109)

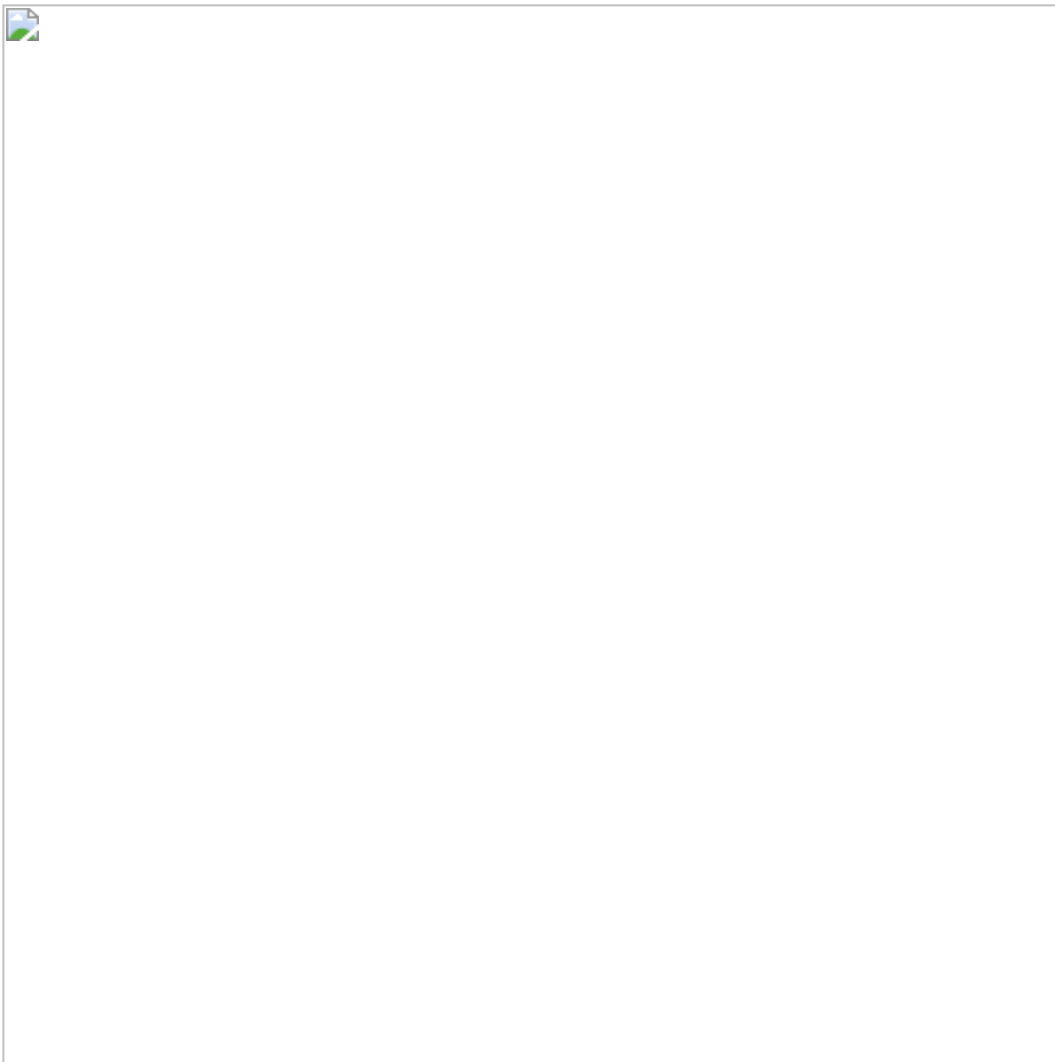
节点 节点

系统用户 \* × sysadmin(sysadmin)



JMS

## 7.用新建的账号登录跳板机



JumpServer

FIT2CLOUD 飞致云 旗下品牌

我的资产

我的应用

个人信息


Web终端

文件管理

个人信息

个人信息

soifa



用户名soifa

名称soifa

角色用户

邮件liruifangtote@163.com

激活中Yes

ssh公钥

多因子认证禁用

用户来源数据库

创建日期2020-05-09 05:53:21

快速修改

设置多因子认证:启用

更改密码:更新

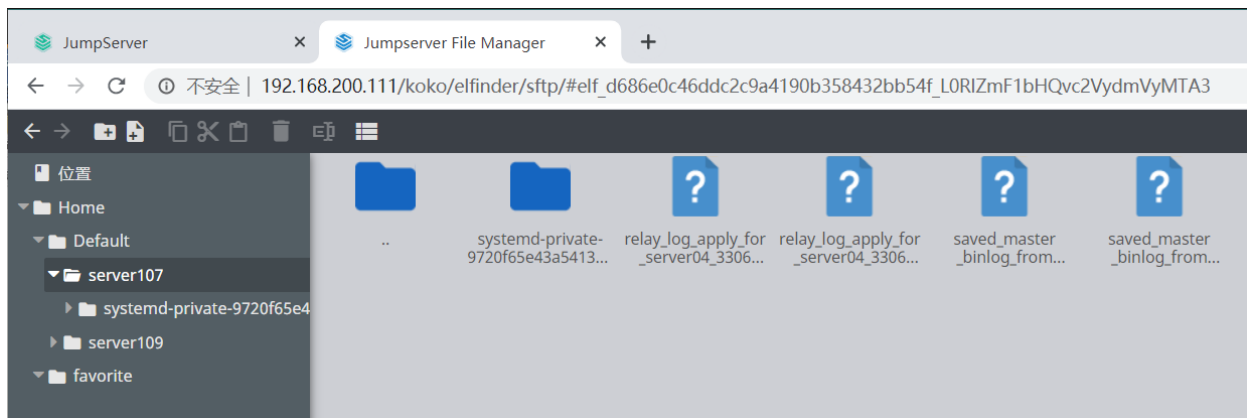
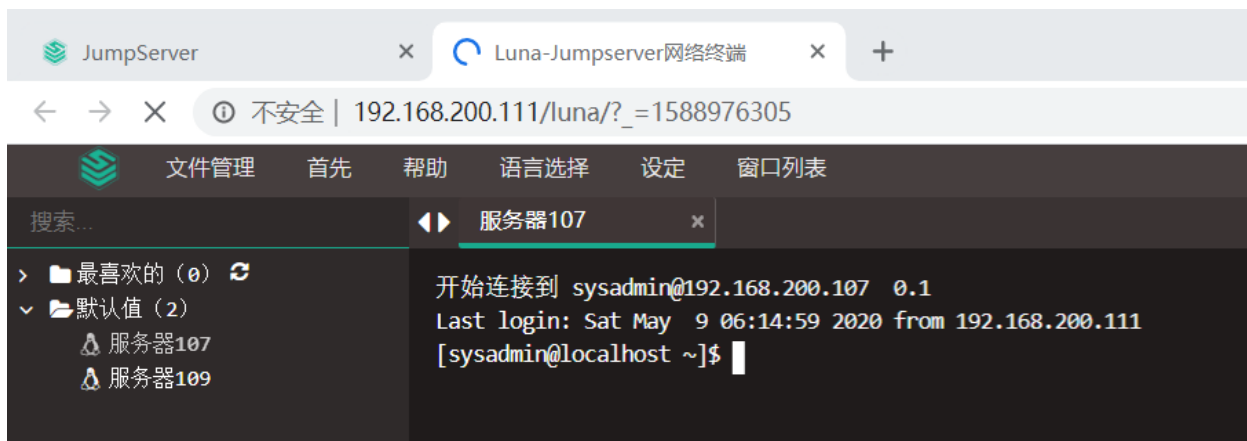
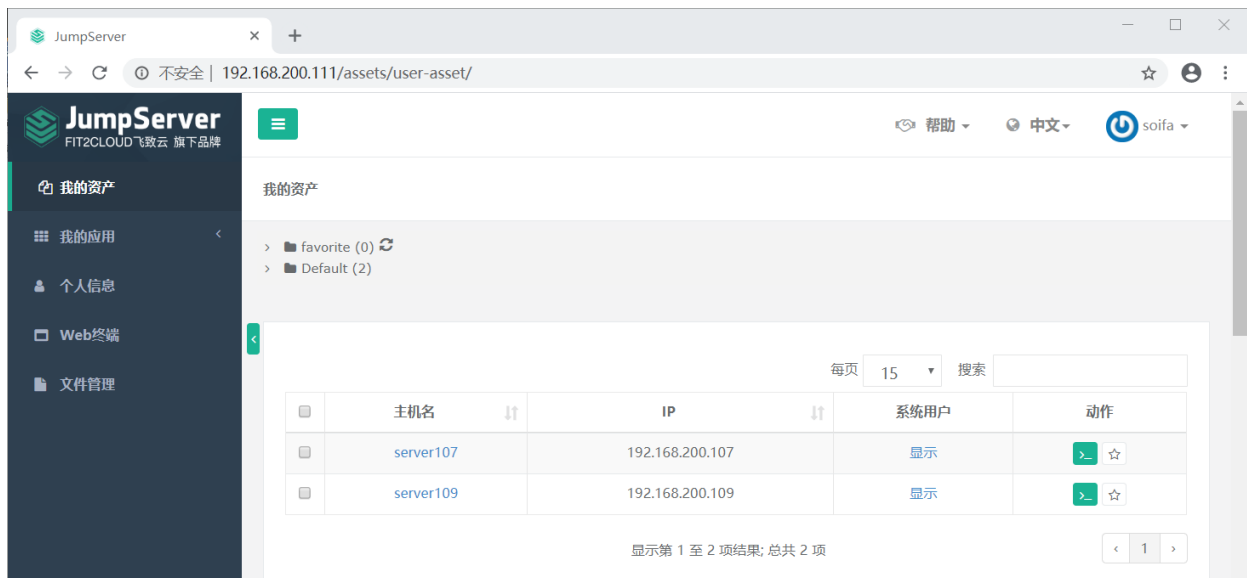
更改SSH密钥:更新

重置并下载SSH密钥:重置

设置

Copyright FIT2CLOUD 飞致云 © 2014-2020

Version 1.5.8-2 GPLv2.



## 8.命令行xhell连接服务器

sofia, 欢迎使用Jumpserver开源堡垒机系统

- 1) 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** 进行搜索, 如: /192.168.
- 3) 输入 **p** 进行显示您有权限的主机。
- 4) 输入 **g** 进行显示您有权限的节点。
- 5) 输入 **d** 进行显示您有权限的数据库。
- 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> █

注意：用户账号      资产授权      这两处容易出错，而导致ssh无法登录

ID	主机名	IP	备注
1	server107	192.168.200.107	被管理服务器107
2	server109	192.168.200.109	被管理服务器109

页码: 1, 每页行数: 31, 总页数: 1, 总数量: 2

提示: 输入资产ID直接登录, 二级搜索使用 // + 字段, 如: //192 上一页: b 下一页: n

搜索: 所有

Opt> █

ID	主机名	IP	备注
1	server107	192.168.200.107	被管理服务器107
2	server109	192.168.200.109	被管理服务器109

页码: 1, 每页行数: 31, 总页数: 1, 总数量: 2

提示: 输入资产ID直接登录, 二级搜索使用 // + 字段, 如: //192 上一页: b 下一页: n

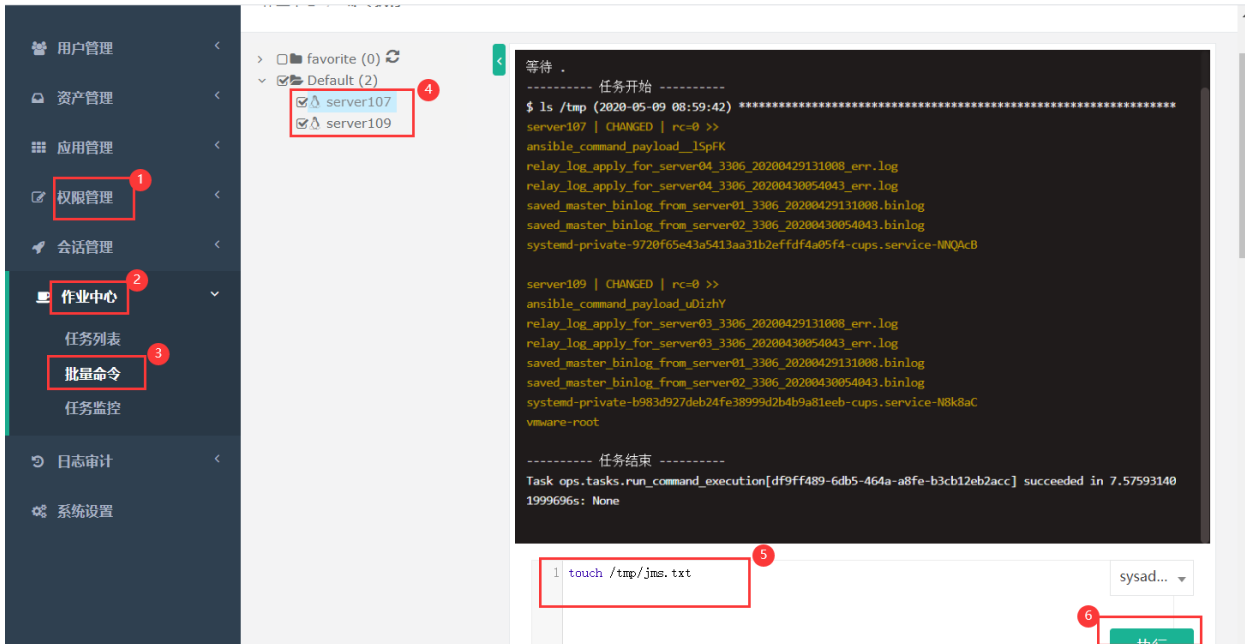
搜索: 所有

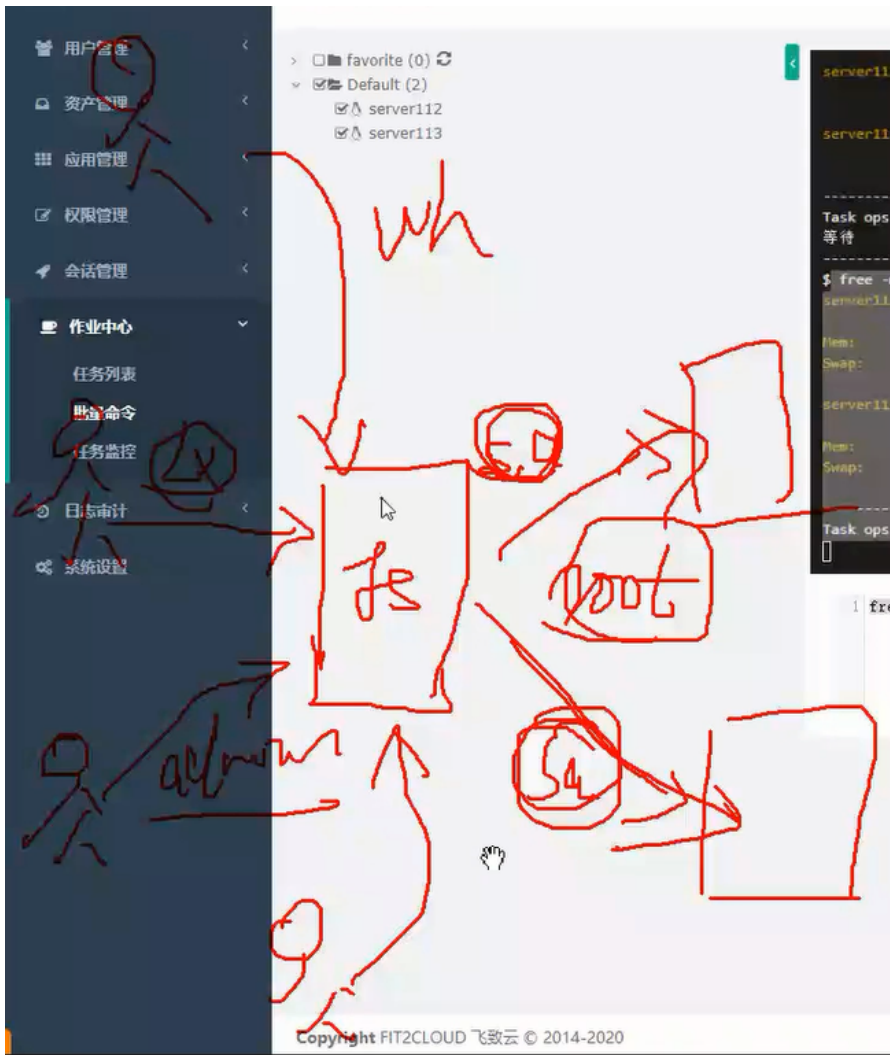
Opt> 1

开始连接到 sysadmin@192.168.200.107 0.1

Last login: Sat May 9 06:18:30 2020 from 192.168.200.111

[sysadmin@localhost ~]\$ █





查看命令

```
[root@localhost ~]# cat /etc/sudoers
```



[http://192.168.200.111/static/img/logo\\_text.png](http://192.168.200.111/static/img/logo_text.png)

```
(py3) [root@jumpserver opt]# ls jumpserver/data/static/img/
authenticator_android.png  avatar          header-profile.png  logo.png
otp_auth.png
authenticator_iphone.png   facio.ico      login_image.png     logo_text.png
root.png

(py3) [root@jumpserver opt]# cd jumpserver/data/static/img/
(py3) [root@jumpserver img]# mv logo_text.png logo_text.png.bak
(py3) [root@jumpserver img]# rz -E
rz waiting to receive.
```

```
(py3) [root@jumpserver img]# systemctl restart nginx
```

```
(py3) [root@jumpserver img]#
```



这就是公司的自动化平台啦

```
(py3) [root@jumpserver img]# mv logo.png logo.png.bak
```

您在 /var/spool/mail/root 中有新邮件

```
(py3) [root@jumpserver img]# rz -E
```

rz waiting to receive.

```
(py3) [root@jumpserver img]# systemctl restart nginx
```

```
(py3) [root@jumpserver img]#
```



<http://192.168.200.111/static/img/logo.png>





## 登录

登录

[忘记密码?](#)