jump-server 安装部署

配置清单

主机名称	IP地址	最低配置	软件名称	版本
jump-server	192.168.1.251	4CPU，8G内存	jumpserver	v2.10.2

jumpserver概述

> Jumpserver是一款开源的堡垒机，可使系统的管理员和开发人员安全的连接到企业内部服务器上执行操作，并且支持大部分操作系统，是一款非常安全的远程连接工具

常见支持的系统：

> CentOS，RedHat，Fedora，Amazon LinuxDebianSUSE，UbuntuFreeBSD其他ssh协议硬件设备

安装部署

下载软件，部署基础环境，从跳板机拷贝project3/jumpserver目录 到 jumpserver主机 解压此文件

```
[root@js ~]# scp -r project3/jumpserver 192.168.1.251:/root/

jump-server主机操作
[root@jump-server ~]# cd jumpserver/
[root@jump-server jumpserver]# md5sum docker-compose jumpserver-installer-v2.10.2.tar.gz
bec660213f97d788d129410d047f261f  docker-compose
223415d3cd9777a58fc0dc71c0b579cf  jumpserver-installer-v2.10.2.tar.gz
[root@jump-server jumpserver]# yum install -y firewalld docker-ce-18.06.3.ce-3.el7.x86_64.rpm
[root@jump-server jumpserver]# systemctl enable --now docker firewalld
[root@jump-server jumpserver]# cp docker-compose /usr/bin/
[root@jump-server jumpserver]# chmod 755 /usr/bin/docker-compose
```

导入jumpserver所需要的镜像

```
[root@jump-server jumpserver]# cd images/
[root@jumpserver images]# for i in *.tar; do docker load -i $i; done
[root@jump-server images]# cd ..        #返回到上一级
[root@jump-server jumpserver]# tar -xf jumpserver-installer-v2.10.2.tar.gz -C /opt/
[root@jump-server jumpserver]# cd /opt/jumpserver-installer-v2.10.2/
[root@jump-server jumpserver-installer-v2.10.2]# cat static.env
export VERSION="v2.10.2"
[root@jump-server jumpserver-installer-v2.10.2]# ./jmsctl.sh install #一路回车
语言 Language  (cn/en)  (default cn):

1. 检查配置文件
配置文件位置: /opt/jumpserver/config
/opt/jumpserver/config/config.txt  [ ✓ ]
/opt/jumpserver/config/nginx/lb_ssh_server.conf  [ ✓ ]
完成
```

```
2. 配置 Nginx
配置文件: /opt/jumpserver/config/nginx/cert
/opt/jumpserver/config/nginx/cert/server.crt  [ ✓ ]
/opt/jumpserver/config/nginx/cert/server.key  [ ✓ ]
完成

3. 备份配置文件
备份至 /opt/jumpserver/config/backup/config.txt.2021-05-27_20-09-29
完成

>>> 安装配置 Docker
1. 安装 Docker
完成

2. 配置 Docker
是否需要自定义 docker 存储目录，默认将使用目录 /var/lib/docker? (y/n)  (默认为 n):
完成

3. 启动 Docker
完成

>>> 加载 Docker 镜像
[jumpserver/redis:6-alpine]
[jumpserver/mysql:5]
[jumpserver/nginx:alpine2]
[jumpserver/luna:v2.10.2]
[jumpserver/core:v2.10.2]
[jumpserver/koko:v2.10.2]
[jumpserver/lion:v2.10.2]
[jumpserver/lina:v2.10.2]

>>> 安装配置 JumpServer
1. 配置网络
是否需要支持 IPv6? (y/n)  (默认为 n):
完成

2. 配置加密密钥
SECRETE_KEY:     ZTAwOWYzNDctMjA1ZS00NzM4LThlZDMtYjEwYmY3NDJkZjA4
BOOTSTRAP_TOKEN: ZTAwOWYzNDctMjA1
完成

3. 配置持久化目录
是否需要自定义持久化存储，默认将使用目录 /opt/jumpserver? (y/n)  (默认为 n):
完成

4. 配置 MySQL
是否使用外部 MySQL? (y/n)  (默认为 n):
完成

5. 配置 Redis
是否使用外部 Redis? (y/n)  (默认为 n):
完成

安装完成之后，可以启动jumpserver
先重启docker
[root@jump-server jumpserver-installer-v2.10.2]# systemctl restart docker
[root@jump-server jumpserver-installer-v2.10.2]# ./jmsctl.sh start
Creating network "jms_net" with driver "bridge"
Creating jms_redis  ... done
Creating jms_mysql  ... done
```

```
Creating jms_core   ... done
Creating jms_lina   ... done
Creating jms_lion   ... done
Creating jms_koko   ... done
Creating jms_celery ... done
Creating jms_luna   ... done
Creating jms_nginx  ... done
[root@jump-server jumpserver-installer-v2.10.2]# ./jmsctl.sh status
Name           Command                    State          Ports
-------------------------------------------------------------------------
jms_celery     ./entrypoint.sh start task  Up (healthy)   8070/tcp, 8080/tcp
jms_core       ./entrypoint.sh start web   Up (healthy)   8070/tcp, 8080/tcp
jms_koko       ./entrypoint.sh             Up (healthy)   0.0.0.0:2222->2222/tcp, 5000/tcp
jms_lina       /docker-entrypoint.sh ngin  Up (healthy)   80/tcp
jms_lion       /usr/bin/supervisord        Up (healthy)   4822/tcp
jms_luna       /docker-entrypoint.sh ngin  Up (healthy)   80/tcp
jms_mysql      docker-entrypoint.sh --cha  Up (healthy)   3306/tcp, 33060/tcp
jms_nginx      sh -c crond -b -d 8 && ngi  Up (healthy)   0.0.0.0:8443->443/tcp, 0.0.0.0:8080-
>80/tcp
jms_redis      docker-entrypoint.sh redis  Up (healthy)   6379/tcp
[root@jump-server jumpserver-installer-v2.10.2]#

2. 其它一些管理命令
./jmsctl.sh stop        #停止 JumpServer
./jmsctl.sh restart     #重启 JumpServer
./jmsctl.sh upgrade     #升级 JumpServer
更多还有一些命令，你可以 ./jmsctl.sh --help 来了解

3. Web 访问
http://公网IP:8080
https://公网IP:8443
默认用户: admin  默认密码: admin

4. SSH/SFTP 访问
ssh admin@192.168.1.251 -p2222
sftp -P2222 admin@192.168.1.251

5. 更多信息
我们的官网: https://www.jumpserver.org/
我们的文档: https://docs.jumpserver.org/
```

访问jumpserver的web页面，可以使用http或者https访问

```
没有公网IP需要购买，并进行绑定才能正常访问
http://49.4.15.231:8080/
访问页面如下：
使用用户名admin，密码admin登录，会让修改密码，使用新密码登录即可
```

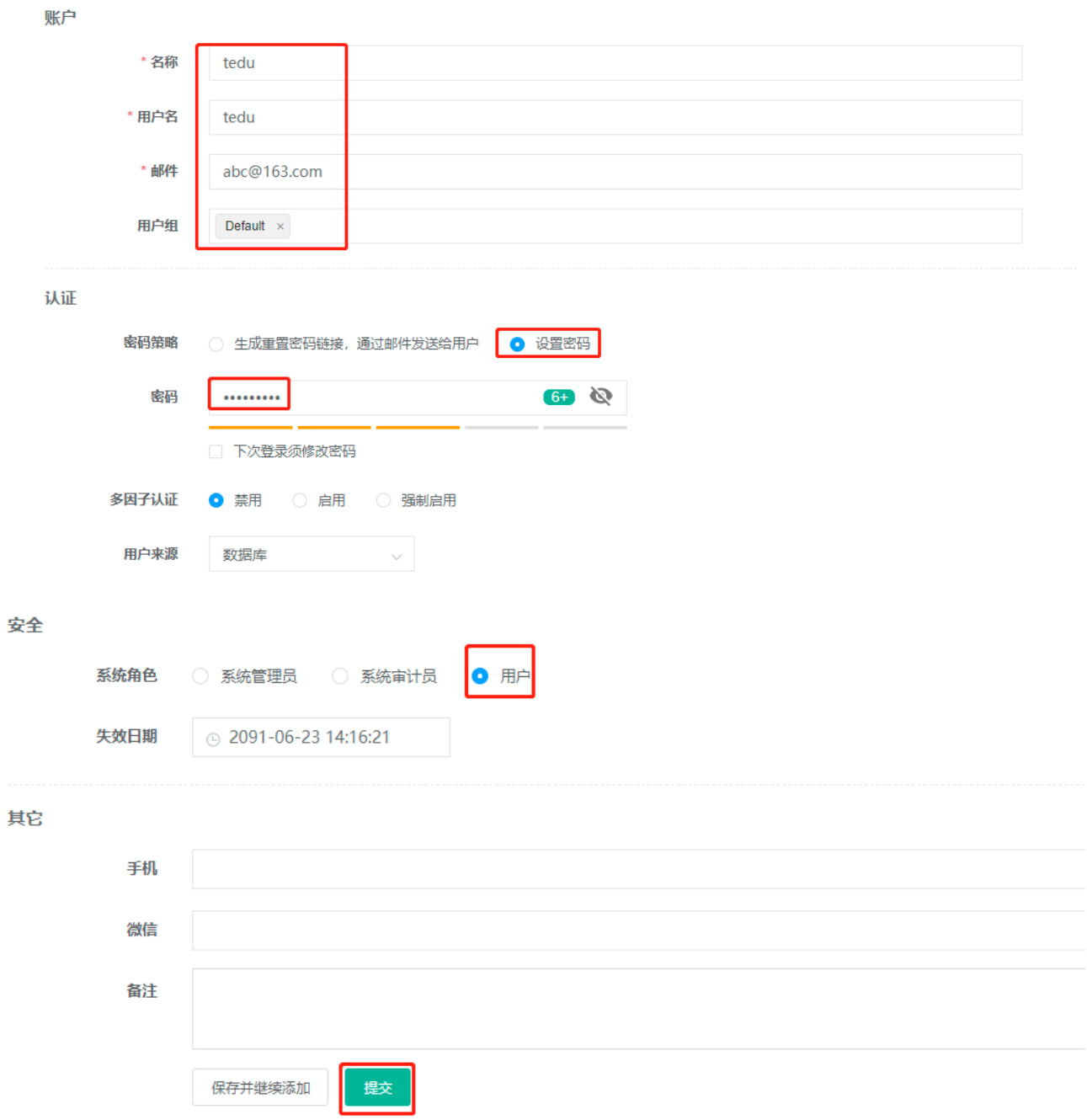
在jumpserver里面有三种用户：

用户管理里面的用户

资产管理里面的用户：管理用户 和 系统用户

用户： 这个用户指堡垒机账号， 就是你能用这个账号登录web页面， 登录跳板机服务器的用户
管理用户： 这个用户是你的目标资产主机上面拥有很高权限的用户比如root, 这个用户的目的是用来通过ansible接口推送下面将要说的系统用户
系统用户： 指你需要登录的目标资产主机的普通用户， 这个用户是给你登录目标主机的用户， 批量执行命令的用户

创建登录jumpserver的普通用户tedu：



用户创建好之后，需要使用这个用户进行资产的管理，但是我们还没有资产。
在华为云上面创建三台web集群的机器，只需要把机器创建出来即可，模拟用户管理资产，不用配置web集群

找到资产管理，点击管理用户

管理用户是资产（被控服务器）上的 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

点击资产管理，创建系统用户

> 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


资产管理
资产列表
网域列表
管理用户
系统用户
命令过滤
平台列表
标签管理
创建

创建系统用户web


资产管理
资产列表
网域列表
管理用户
系统用户
命令过滤
平台列表
标签管理
创建
主机协议
SSH
Telnet
RDP
VNC

创建系统用户

基本

* 名称 web

登录模式 自动登录 手动登录

如果选择手动登录模式，用户名和密码可以不填写

* 用户名 web

用户名与用户相同

用户名是动态的，登录资产时使用当前用户的用户名登录

优先级 81

优先级可选范围为 1-100 (数值越小越优先)

* 协议 ssh

系统用户创建完成之后，还没有资产，现在需要添加资产（此资产就是后面需要使用jumpserver管理的集群机器，如web，数据库集群等）

创建资产

基本

* 主机名 web-0001

* IP(域名) 192.168.1.10

* 系统平台 Linux

公网IP

网域 请选择

协议组

协议组 ssh 22

认证

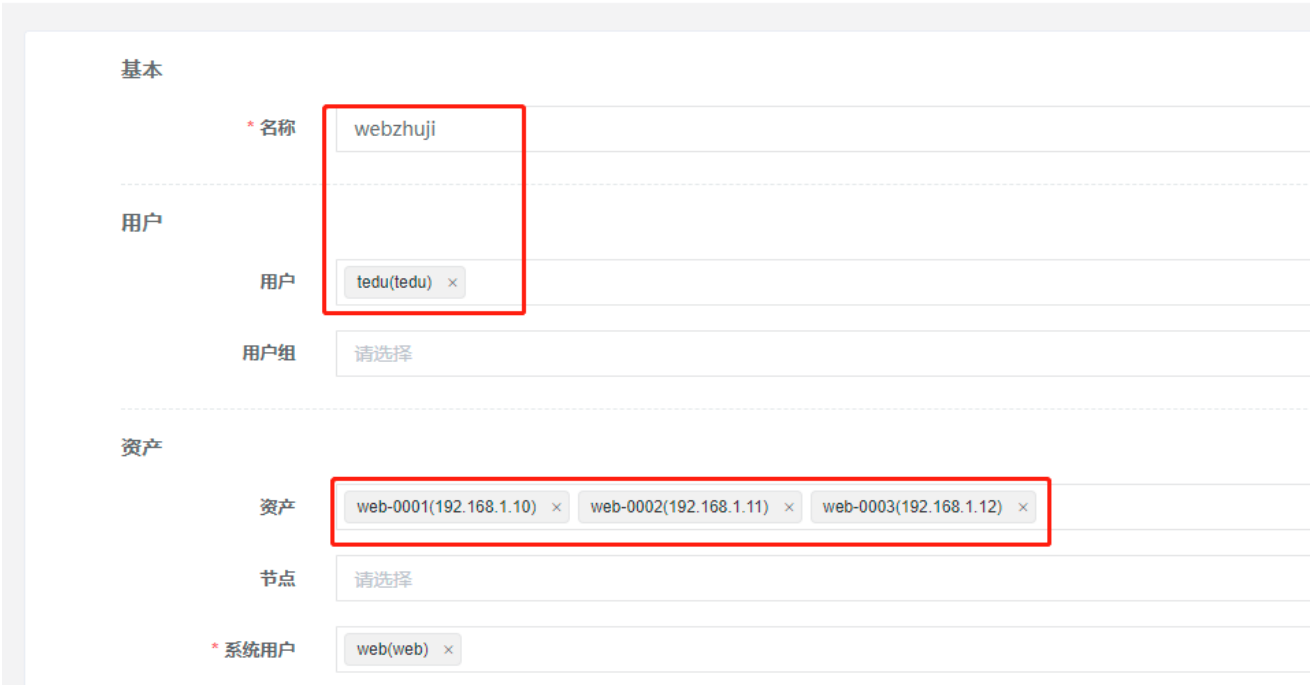
* 管理用户 root(root)

点击保存并继续添加，可以继续添加其他的机器，如果没有其他机器需要添加，点击提交即可

提交结果如下：

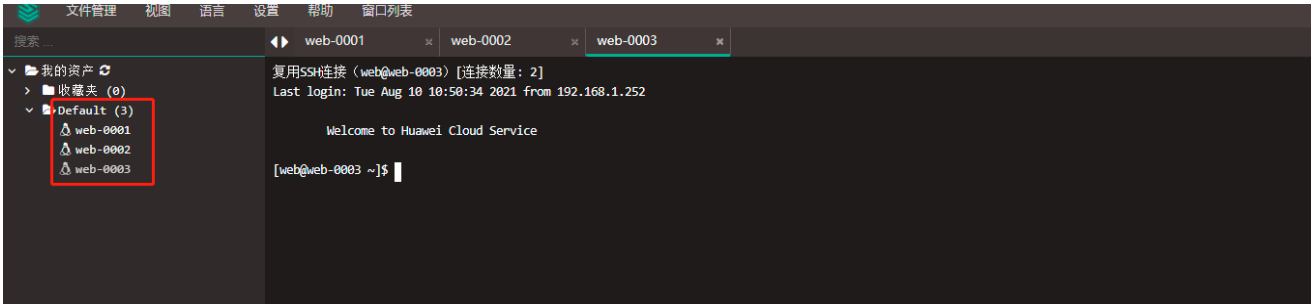
资产创建完成之后，需要授权给哪个系统用户进行管理

创建资产授权规则



此时退出admin管理用户，使用tedu用户登录，测试web系统用户的资产

点击web终端，可以正常登陆资产，使用sudo -s切换到root用户



如果jumpserver有多个用户，并且授权不同的用户管理不同的资产，此时每个用户只能看到自己的资产，别人的资产是看不到的