使用创建一个 python3 容器

** 注意，一定要使用 python3.6 版本

https://github.com/celery/kombu/issues/841

```
docker run -dit --restart=always --name jump -p 8280:80 -v /var/www:/storage
registry.cn-shenzhen.aliyuncs.com/itcp/my-python36:latest /usr/sbin/init
```

mysql

redis

```
docker run -dit --restart=always -p 6379:6379 --name my-redis \
-v /var/docker_storage/redis:/storage \
registry.cn-shenzhen.aliyuncs.com/itcp/my-redis /usr/sbin/init
```

建立 Python 虚拟环境

```
[root@centos7_node1 ~]# docker run -dit --restart=always --name jump
-p 8280:80 -v /var/www:/storage
registry.cn-shenzhen.aliyuncs.com/itcp/my-python37:latest
/usr/sbin/init
bfb6899c85ce2638d51400a739c79c50a5bd29a91e34211958fc70331499c801
[root@centos7_node1 ~]#
[root@centos7_node1 ~]# docker exec -ti jump /bin/bash
[root@bfb6899c85ce /]#
[root@bfb6899c85ce /]# cd /storage/
[root@bfb6899c85ce storage]# ls
[root@bfb6899c85ce storage]#
[root@bfb6899c85ce storage]# python3 -m venv py3
[root@bfb6899c85ce storage]# source /storage/py3/bin/activate
(py3) [root@bfb6899c85ce storage]#
```

下载源码

```
(py3) [root@bfb6899c85ce storage]#  git clone
https://github.com/jumpserver/jumpserver.git
```

安装依赖 RPM 包

```
(py3) [root@bfb6899c85ce storage]# cd jumpserver/requirements/
(py3) [root@bfb6899c85ce requirements]# yum -y install $(cat
rpm_requirements.txt)
```

安装 Python 库依赖

```
pip3 install -r requirements.txt
...
...
  Running setup.py install for aliyun-python-sdk-ecs ... done
Successfully installed Django-2.1 ForgeryPy-0.1 Jinja2-2.10
MarkupSafe-1.0 Pillow-4.3.0 PyNaCl-1.2.1 PyYAML-3.12
Werkzeug-0.14.1 aliyun-python-sdk-core-v3-2.9.1
aliyun-python-sdk-ecs-4.10.1 amqp-2.1.4 ansible-2.4.2.0
asn1crypto-0.24.0 azure-common-1.1.16 azure-nspkg-3.0.2
azure-storage-blob-1.3.1 azure-storage-common-1.3.0
azure-storage-nspkg-3.0.0 bcrypt-3.1.4 billiard-3.5.0.3
boto-2.49.0 boto3-1.6.5 botocore-1.9.5 celery-4.1.0
certifi-2018.1.18 cffi-1.11.5 chardet-3.0.4 configparser-3.5.0
coreapi-2.3.3 coreschema-0.0.4 crcmod-1.7 cryptography-2.3.1
decorator-4.1.2 django-auth-ldap-1.3.0 django-bootstrap3-9.1.0
django-celery-beat-1.1.1 django-filter-2.0.0 django-formtools-2.1
django-ranged-response-0.2.0 django-redis-cache-1.7.1
django-rest-swagger-2.1.2 django-simple-captcha-0.5.6
djangorestframework-3.8.2 djangorestframework-bulk-0.2.1
dnspython-1.15.0 docutils-0.14 drf-nested-routers-0.90.2
drf-yasg-1.9.1 ecdsa-0.13 elasticsearch-6.1.1 enum-compat-0.0.2
ephem-3.7.6.0 eventlet-0.24.1 future-0.17.1 greenlet-0.4.14
gunicorn-19.9.0 idna-2.7 inflection-0.3.1 itsdangerous-0.24
itypes-1.1.0 jmespath-0.9.3 jms-storage-0.0.19 kombu-4.0.2
ldap3-2.4 monotonic-1.5 mysqlclient-1.3.12 olefile-0.44
openapi-codec-1.3.2 oss2-2.4.0 paramiko-2.4.1 passlib-1.7.1
pyasn1-0.4.2 pycparser-2.19 pycrypto-2.6.1 pycryptodome-3.7.0
pyldap-2.4.45 pyotp-2.2.6 python-dateutil-2.6.1
python-gssapi-0.6.4 pytz-2018.3 redis-2.10.6 requests-2.18.4
ruamel.yaml-0.15.76 s3transfer-0.1.13 simplejson-3.13.2 six-1.11.0
sshpubkeys-3.1.0 uritemplate-3.0.0 urllib3-1.22 vine-1.1.4
(py3) [root@17cbb6c594e1 requirements]#
```

创建数据库 Jumpserver 并授权

```
$ mysql
> create database jumpserver default charset 'utf8';
> grant all on jumpserver.* to 'jumpserver'@'%' identified by
'some#1103';
```

修改 Jumpserver 配置文件

cd/opt/jumpserver cp config_example.py config.py
$ vi config.py # 我们计划修改 DevelopmentConfig 中的配置，因为默认 jumpserver 是使用该配置，它继承自 Config

注意: 配置文件是 Python 格式，不要用 TAB，而要用空格

```
class Config:
...
      # Use Redis as broker for celery and web socket
      REDIS_HOST = os.environ.get("REDIS_HOST") or '172.17.0.1'
      REDIS_PORT = os.environ.get("REDIS_PORT") or 6379
      REDIS_PASSWORD = os.environ.get("REDIS_PASSWORD") or ''
      REDIS_DB_CELERY = os.environ.get('REDIS_DB') or 3
      REDIS_DB_CACHE = os.environ.get('REDIS_DB') or 4
...
...
class DevelopmentConfig(Config):
      DEBUG = True
      DB_ENGINE = 'mysql'
      DB_HOST = '172.17.0.1'
      DB_PORT = 3306
      DB_USER = 'jumpserver'
      DB_PASSWORD = 'some#1103'
      DB_NAME = 'jumpserver'

...

config = DevelopmentConfig() # 确保使用的是刚才设置的配置文件
```

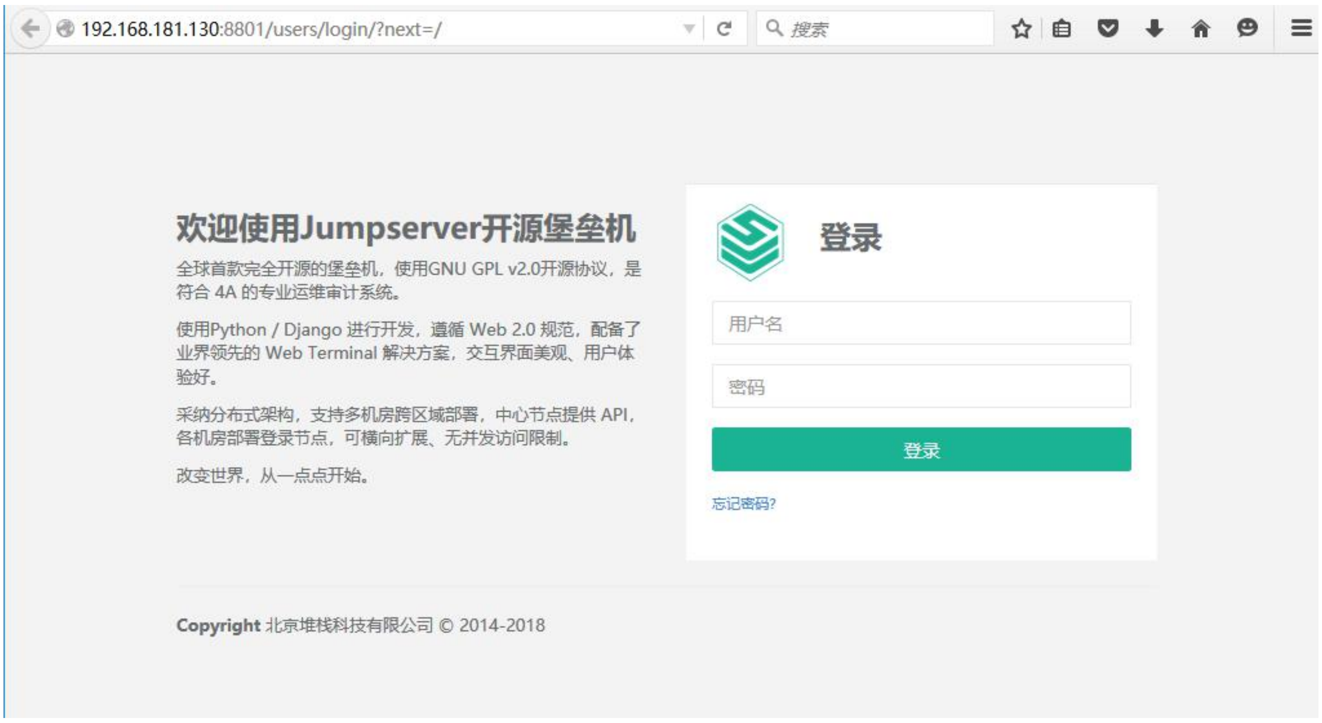
生成数据库表结构和初始化数据

```
$ cd jumpserver/utils
$ bash make_migrations.sh
.....
  Applying perms.0002_auto_20181103_1230... OK
  Applying sessions.0001_initial... OK
  Applying terminal.0001_initial... OK
  Applying terminal.0002_auto_20181103_1230... OK
2018-11-03 12:30:38 [signals_handler DEBUG] Receive django ready
signal
2018-11-03 12:30:38 [signals_handler DEBUG] - fresh all settings
No conflicts detected to merge.
(py3) [root@17cbb6c594e1 utils]#
```

运行 Jumpserver

```
cdjumpserver python3 run_server.py all
....
...
Task terminal.tasks.clean_orphan_session[f2511f8d-2171-4913-8ccd-d022ef56a50e]
succeeded in 0.13815198899982883s: None
beat: Synchronizing schedule...
Writing entries...
beat: Waking up in 1.00 minute.
```

运行不报错，请浏览器访问 http://192.168.244.144:8080/ (这里只是 Jumpserver，没有 Web Terminal，所以访问 Web Terminal 会报错)


192.168.181.130:8801/users/login/?next=/

欢迎使用Jumpserver开源堡垒机

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。

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。

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。

改变世界，从一点点开始。

Copyright 北京堆栈科技有限公司 © 2014-2018

三. 安装 SSH Server 和 WebSocket Server: Coco

3.1 下载或 Clone 项目

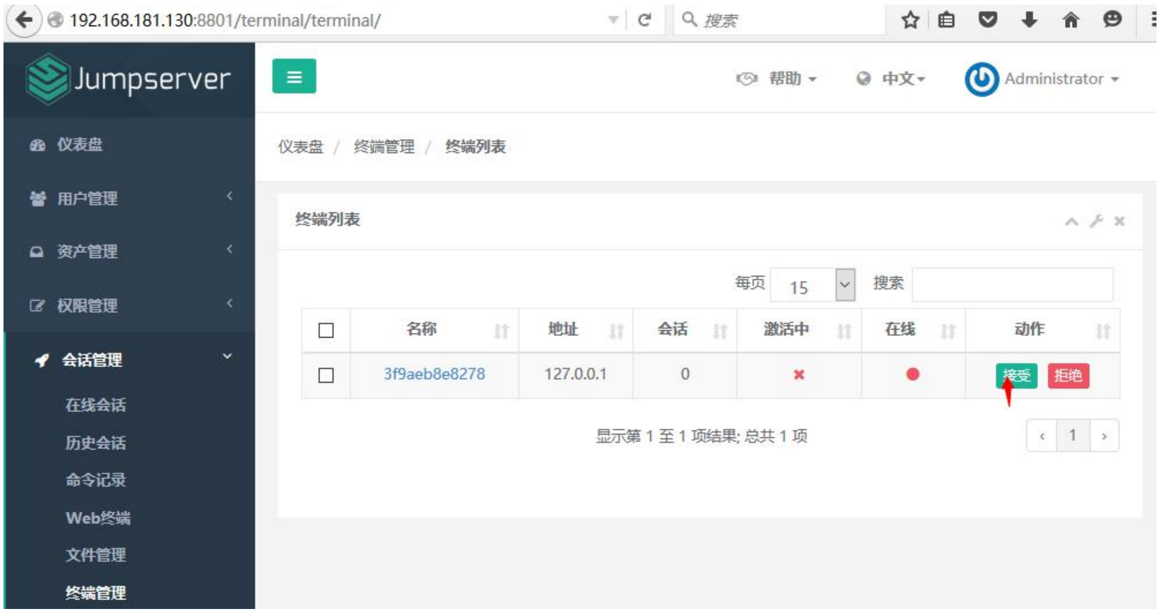
新开一个终端，连接测试机，别忘了 source /opt/py3/bin/activate

cd/storage source py3/bin/activate
$ git clone https://github.com/jumpserver/coco.git && cd coco && git checkout master

3.2 安装依赖
cd/opt/coco/requirements yum -y install (catrpmrequirements.txt) pip3 install -r requirements.txt
...
Running setup.py install for tornado ... done
Successfully installed Flask-1.0.2 Flask-SocketIO-2.9.2 cachetools-2.0.1 cffi-1.11.2 click-6.7 dotmap-1.2.20 idna-2.6 jumpserver-python-sdk-0.0.50 psutil-5.4.1 pycparser-2.18 pyte-0.8.0 python-engineio-2.1.0 python-socketio-1.8.3 tornado-4.5.2 wcwidth-0.1.7
(py3) [root@3f9aeb8e8278 requirements]#

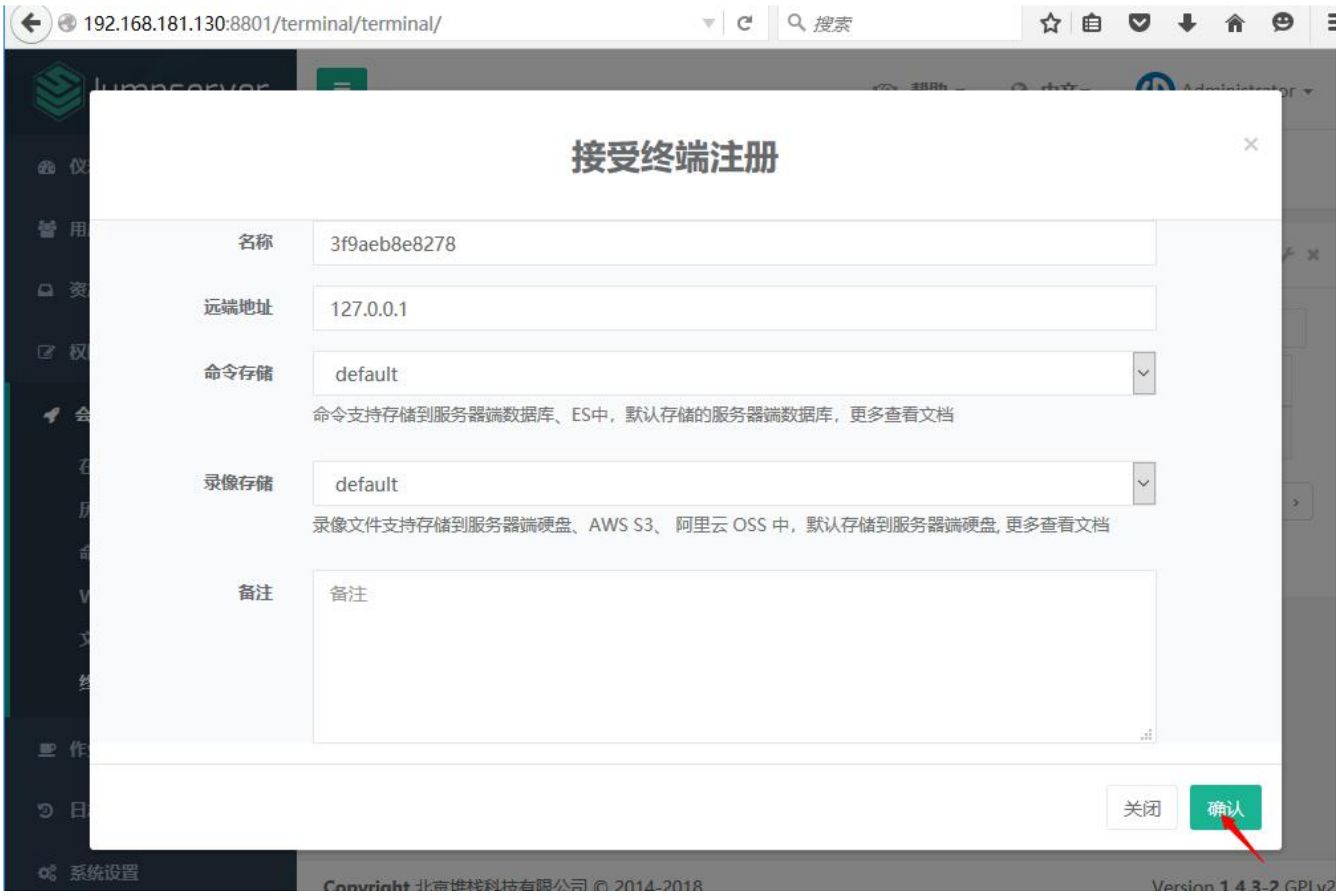
3.3 查看配置文件并运行
cd/opt/coco cp conf_example.py conf.py
$ python3 run_server.py
(py3) [root@3f9aeb8e8278 coco]# python3 run_server.py
2018-11-03 05:16:04 [service DEBUG] Initial app service
2018-11-03 05:16:04 [service DEBUG] Load access key
2018-11-03 05:16:04 [service INFO] No access key found, register it
2018-11-03 05:16:05 [service INFO] "Terminal was not accepted yet"
2018-11-03 05:16:08 [service INFO] "Terminal was not accepted yet"

时需要去 Jumpserver 管理后台-会话管理-终端管理（http://192.168.244.144:8080/terminal/terminal/）接受 Coco 的注册



名称	地址	会话	激活中	在线	动作
3f9aeb8e8278	127.0.0.1	0	✗	●	接受 拒绝

显示第 1 至 1 项结果; 总共 1 项


192.168.181.130:8801/terminal/terminal/
搜索
系统设置
Copyright 北京堆栈科技有限公司 © 2014-2018
关闭
确认

接受终端注册

名称	3f9aeb8e8278
远端地址	127.0.0.1
命令存储	default 命令支持存储到服务器端数据库、ES中，默认存储的服务器端数据库，更多查看文档
录像存储	default 录像文件支持存储到服务器端硬盘、AWS S3、 阿里云 OSS 中，默认存储到服务器端硬盘, 更多查看文档
备注	备注

成功后终端是这样的

```
2018-11-03 05:18:49 [service DEBUG] Service http auth:
<jms.auth.AccessKeyAuth object at 0x7f37f23caef0>
Start coco process
2018-11-03 05:18:49 [app DEBUG] Loading config from server:
{"COMMAND_STORAGE": {"TYPE": "server"}, "REPLAY_STORAGE": {"TYPE":
"server"}, "SECURITY_MAX_IDLE_TIME": 30}
Sat Nov 3 05:18:49 2018
Coco version 1.4.3, more see https://www.jumpserver.org
Quit the server with CONTROL-C.
Starting ssh server at 0.0.0.0:2222
Starting websocket server at 0.0.0.0:5000
``
```

3.4 测试连接

```
$ 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
如果能登陆代表部署成功

~~~
```shell
boeving@boeving-mint:~$ ssh -p2222 admin@192.168.181.130
admin@192.168.181.130's password:

    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>
~~~

四. 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
wget https://github.com/jumpserver/luna/releases/download/1.4.3/luna.tar.gz
4.1 解压 Luna

$ pwd
/opt/

tarxvfluna.tar.gz ls /opt/luna
…
\========

五. 安装 Windows 支持组件(这步我没搞)

因为手动安装 guacamole 组件比较复杂，这里提供打包好的 docker 使用, 启动 guacamole

注意：这里一定要改写一下本机的 IP 地址，否则会出错

```
    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://<填写本机的 IP 地址>:8080 \
      registry.jumpserver.org/public/guacamole:latest
```

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一下。

再次强调：修改 JUMPSERVER_SERVER 环境变量的配置，填上 Jumpserver 的内网地址, 这时 去 Jumpserver-会话管理-终端管理 接受[Gua]开头的一个注册
\========

docker run -dit --restart=always -p 80:80 --name nginx-node1 \
-v /var/docker_storage/nginx:/storage -v /var/docker_storage/jump3:/jump \
registry.cn-shenzhen.aliyuncs.com/itcp/nginx:v1 /usr/sbin/init

vhost/jump.conf
```
server {
listen 80;
server_name jump.test;
proxy_set_header X-Real-IP remoteaddr;proxysetheaderHosthost;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

location /luna/ {
    try_files $uri / /index.html;
    alias /jump/luna/;
}

location /media/ {
    add_header Content-Encoding gzip;
    root /jump/jumpserver/data/;
}

location /static/ {
    root /jump/jumpserver/data/;
}

location /socket.io/ {
    proxy_pass http://localhost:5000/socket.io/;
    proxy_buffering off;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}

location /guacamole/ {
    proxy_pass http://localhost:8081/;
    proxy_buffering off;
    proxy_http_version 1.1;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection $http_connection;
    access_log off;
}

location / {
    proxy_pass http://172.17.0.1:8801;
}

}
```

用 docker 部署好后的运行启动

首次启动

```
source /var/www/py3/bin/activate
cd /var/www/jumpserver/requirements/ && yum -y install $(cat
rpm_requirements.txt) && pip3 install -r requirements.txt
vi /var/www/jumpserver/config.py
cd /var/www/jumpserver/utils && bash make_migrations.sh
cd /var/www/jumpserver/ && python3 run_server.py all
```

以后启动

```
source /var/www/py3/bin/activate
cd /var/www/jumpserver/requirements/ && yum -y install $(cat
rpm_requirements.txt) && pip3 install -r requirements.txt
cd /var/www/jumpserver/ && python3 run_server.py all
```