



链滴

# Jumpserver 环境安装与使用

作者: [someone33881](#)

原文链接: <https://ld246.com/article/1542274562283>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



```
python3.6 -m venv py3
source /opt/py3/bin/activate
# 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,以下所有命令均在
虚拟环境中运行
(py3) [root@localhost py3]

#自动载入 Python 虚拟环境配置
#此项仅为懒癌晚期的人员使用,防止运行 Jumpserver 时忘记载入 Python 虚拟环境导致程序无法运
。使用autoenv
cd /opt
git clone https://github.com/kennethreitz/autoenv.git
echo 'source /opt/autoenv/activate.sh' >> ~/.bashrc
source ~/.bashrc
```

## 三、安装jumpserver

### 1、clone项目

```
cd /opt/
git clone https://github.com/jumpserver/jumpserver.git
echo "source /opt/py3/bin/activate" > /opt/jumpserver/.env # 进入 jumpserver 目录时将自动
入 python 虚拟环境
# 首次进入 jumpserver 文件夹会有提示,按 y 即可
# Are you sure you want to allow this? (y/N) y
```

### 2、安装依赖RPM包

```
cd /opt/jumpserver/requirements
yum -y install $(cat rpm_requirements.txt) # 如果没有任何报错请继续
```

### 3、安装python依赖包

```
pip install --upgrade pip setuptools
pip install -r requirements.txt
```

### 4、安装 Redis, Jumpserver 使用 Redis 做 cache 和 celery broke\*\*

```
#centos7-yum方式安装
yum -y install redis
systemctl enable redis
systemctl start redis
```

```
#centos6-yum方式安装
yum -y install redis
chkconfig redis on
service redis start
```

#源码安装, 参考

[Redis 环境安装](#)

## 5、安装MySQL

#参考

[MySQL 环境安装](#)

#centos7

```
yum -y install mariadb mariadb-devel mariadb-server # centos7下安装的是mariadb
systemctl enable mariadb
systemctl start mariadb
```

# centos6 自带的 mysql5.1 不支持,请在其他服务器上创建 jumpserver 数据库连接

## 6、创建Jumpserver数据库并授权

```
mysql -uroot
> create database jumpserver default charset 'utf8';
> grant all on jumpserver.* to 'jumpserver'@'127.0.0.1' identified by '复杂密码大小写字母数字
殊符号';
> flush privileges;
> quit
```

## 7、修改Jumpserver配置文件

```
cd /opt/jumpserver
cp config_example.py config.py
vim config.py
# 注意对齐,不要直接复制本文档的内容,实际内容以文件为准,本文仅供参考,修改配置内容如下:
```

```
"""
```

```
jumpserver.config
```

```
~~~~~
```

```
Jumpserver project setting file
```

```
:copyright: (c) 2014-2017 by Jumpserver Team
```

```
:license: GPL v2, see LICENSE for more details.
```

```
"""
```

```
import os
```

```
BASE_DIR = os.path.dirname(os.path.abspath(__file__))
```

```
class Config:
```

```
    # Use it to encrypt or decrypt data
```

```
    # Jumpserver 使用 SECRET_KEY 进行加密,请务必修改以下设置
```

```
    # SECRET_KEY = os.environ.get('SECRET_KEY') or '2vym+ky!997d5kkcc64mnz06y1mmui3lut
```

```

(^wd=%s_qj$1%x'
SECRET_KEY = '请随意输入随机字符串(推荐字符大于等于 50位)'

# Django security setting, if your disable debug model, you should setting that
ALLOWED_HOSTS = ['*']

# DEBUG 模式 True为开启 False为关闭,默认开启,生产环境推荐关闭
# 注意: 如果设置了DEBUG = False,访问8080端口页面会显示不正常,需要搭建 nginx 代理才可以常访问
DEBUG = os.environ.get("DEBUG") or True

# 日志级别,默认为DEBUG,可调整为INFO, WARNING, ERROR, CRITICAL,默认INFO
LOG_LEVEL = os.environ.get("LOG_LEVEL") or 'WARNING'
LOG_DIR = os.path.join(BASE_DIR, 'logs')

# 使用的数据库配置,支持sqlite3, mysql, postgres等,默认使用sqlite3
# See https://docs.djangoproject.com/en/1.10/ref/settings/#databases

# 默认使用SQLite3,如果使用其他数据库请注释下面两行
# DB_ENGINE = 'sqlite3'
# DB_NAME = os.path.join(BASE_DIR, 'data', 'db.sqlite3')

# 如果需要使用mysql或postgres,请取消下面的注释并输入正确的信息,本例使用mysql做演示(maradb也是mysql)
DB_ENGINE = os.environ.get("DB_ENGINE") or 'mysql'
DB_HOST = os.environ.get("DB_HOST") or '127.0.0.1'
DB_PORT = os.environ.get("DB_PORT") or 3306
DB_USER = os.environ.get("DB_USER") or 'jumpserver'
DB_PASSWORD = os.environ.get("DB_PASSWORD") or '复杂密码大小写字母数字特殊符号'
DB_NAME = os.environ.get("DB_NAME") or 'jumpserver'

# Django 监听的ip和端口,生产环境推荐把0.0.0.0修改成127.0.0.1,这里的意思是允许x.x.x.x访问,127.0.0.1表示仅允许自身访问
# ./manage.py runserver 127.0.0.1:8080
HTTP_BIND_HOST = '0.0.0.0'
HTTP_LISTEN_PORT = 9091

# Redis 相关设置
REDIS_HOST = os.environ.get("REDIS_HOST") or '127.0.0.1'
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

def __init__(self):
    pass

def __getattr__(self, item):
    return None

class DevelopmentConfig(Config):
    pass

class TestConfig(Config):

```





### 3、修改配置文件并运行

```
cd /opt/coco
mkdir keys logs
cp conf_example.py conf.py # 如果 coco 与 jumpserver 分开部署,请手动修改 conf.py
vim conf.py
```

#注意对齐,不要直接复制本文档的内容

\*\*注意: 配置文件是 Python 格式,不要用 TAB,而要用空格\*\*

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#
```

```
import os
```

```
BASE_DIR = os.path.dirname(__file__)
```

```
class Config:
```

```
    """
```

```
    Coco config file, coco also load config from server update setting below
```

```
    """
```

```
    # 项目名称, 会用来向Jumpserver注册, 识别而已, 不能重复
```

```
    # NAME = "localhost"
```

```
    NAME = "coco"
```

```
    # Jumpserver项目的url, api请求注册会使用, 如果Jumpserver没有运行在127.0.0.1:9091,请修改处
```

```
    # CORE_HOST = os.environ.get("CORE_HOST") or 'http://127.0.0.1:9091'
```

```
    CORE_HOST = 'http://127.0.0.1:9091'
```

```
    # 启动时绑定的ip, 默认 0.0.0.0
```

```
    # BIND_HOST = '0.0.0.0'
```

```
    # 监听的SSH端口号, 默认2222
```

```
    # SSHD_PORT = 2222
```

```
    # 监听的HTTP/WS端口号,默认5000
```

```
    # HTTPD_PORT = 5000
```

```
    # 项目使用的ACCESS KEY, 默认会注册,并保存到 ACCESS_KEY_STORE中,
```

```
    # 如果有需求, 可以写到配置文件中, 格式 access_key_id:access_key_secret
```

```
    # ACCESS_KEY = None
```

```
    # ACCESS KEY 保存的地址, 默认注册后会保存到该文件中
```

```
    # ACCESS_KEY_STORE = os.path.join(BASE_DIR, 'keys', '.access_key')
```

```
    # 加密密钥
```

```
    # SECRET_KEY = None
```

```
    # 设置日志级别 ['DEBUG', 'INFO', 'WARN', 'ERROR', 'FATAL', 'CRITICAL']
```

```
    # LOG_LEVEL = 'INFO'
```

```
    LOG_LEVEL = 'WARN'
```

```
# 日志存放的目录
# LOG_DIR = os.path.join(BASE_DIR, 'logs')

# Session录像存放目录
# SESSION_DIR = os.path.join(BASE_DIR, 'sessions')

# 资产显示排序方式, ['ip', 'hostname']
# ASSET_LIST_SORT_BY = 'ip'

# 登录是否支持密码认证
# PASSWORD_AUTH = True

# 登录是否支持秘钥认证
# PUBLIC_KEY_AUTH = True

# SSH白名单
# ALLOW_SSH_USER = 'all' # ['test', 'test2']

# SSH黑名单, 如果用户同时在白名单和黑名单,黑名单优先生效
# BLOCK_SSH_USER = []

# 和Jumpserver 保持心跳时间间隔
# HEARTBEAT_INTERVAL = 5

# Admin的名字,出问题会提示给用户
# ADMINS = ""
COMMAND_STORAGE = {
    "TYPE": "server"
}
REPLAY_STORAGE = {
    "TYPE": "server"
}

# SSH连接超时时间 (default 15 seconds)
# SSH_TIMEOUT = 15

# 语言 = en
LANGUAGE_CODE = 'zh'
```

```
config = Config()
```

```
./cocod start # 后台运行使用 -d 参数./cocod start -d
```

```
# 新版本更新了运行脚本,使用方式./cocod start|stop|status|restart 后台运行请添加 -d 参数
```

启动成功后去Jumpserver 会话管理-终端管理(<http://ip:9091/terminal/terminal/>)(<http://ip:9091/terminal/terminal/>)接受coco的注册!!

注意: : : 在web页面接收coco的注册的时候,填入的ip地址不能是127.0.0.1, 需要是主机实际的网ip地址

## 五、安装web terminal前端Luna







```
vim /usr/local/nginx/conf/nginx.conf
#添加如下server配置
server {
    listen 9092; # 代理端口,以后将通过此端口进行访问,不再通过9091端口
    server_name caizhaoke.cn; # 修改成你的域名

    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 /socket.io/ {
        proxy_pass http://localhost:5000/socket.io/; # 如果coco安装在别的服务器,请填写它的ip
        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 /coco/ {
        proxy_pass http://localhost:5000/coco/; # 如果coco安装在别的服务器,请填写它的ip
        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/; # 如果guacamole安装在别的服务器,请填写它的ip
        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://localhost:9091; # 如果jumpserver安装在别的服务器,请填写它的ip
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}
}
```

运行nginx

```
nginx -t # 确保配置没有问题, 有问题请先解决
```

```
# CentOS 7
systemctl start nginx
systemctl enable nginx
```

```
# CentOS 6
service nginx start
chkconfig nginx on
```

## 八、测试使用jumpserver

检查应用是否已经正常运行

服务全部启动后,访问 <http://ip:9092>,访问nginx代理的端口9092,不要再通过9091端口访问

默认账号: admin 密码: admin

如果部署过程中没有接受应用的注册,需要到Jumpserver 会话管理-终端管理 接受 Coco Guacamole 等应用的注册。

#测试连接

```
#如果登录客户端是 macOS 或 Linux ,登录语法如下
ssh -p2222 admin@ip
sftp -P2222 admin@ip
#密码: admin
```

```
#如果登录客户端是 Windows ,Xshell Terminal 登录语法如下
ssh admin@ip 2222
sftp admin@ip 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
```

## 问题

1、若jumpserver管理页面及终端管理页面均可以访问,但是点击某台主机终端跳转登录终端时,一

卡住或者说没有响应，则应该是coco应用挂掉了，重启即可解决！

```
cd /opt/coco  
./cocod start -d
```