

# Linux????

- [ubuntu14.04 BIND配置](#)
- [配置 zookeeper](#)
- [OpenSSH\\_6.6.1p1 配置 openssh-9.1p1](#)
- [配置 CORS](#)
- [nginx-1.24.0 配置 1.27.0](#)

# ubuntu14.04 BIND????

```
named -v
BIND 9.9.5-3-Ubuntu (Extended Support Version)
```

```
apt-cache policy bind9
sudo apt-get install bind9
```

```
9.11.19
```

```
Ubuntu 14.04 Bind9
Bind9
Bind9
plaintext
sudo service bind9 stop
Bind9
apt-get
plaintext
sudo apt-get remove bind9
```

```
Bind9
purge
plaintext
sudo apt-get purge bind9
```

```
Bind9
Bind9
/etc/bind
plaintext
sudo rm -rf /etc/bind
```

```
Bind9
```

# ??zookeeper???

#####

```
# [] zookeeper[]  
# /usr/local/zookeeper-3.4.13/bin/zkCli.sh -server 127.0.0.1:2181  
> setAcl /zookeeper ip:127.0.0.1:cdrwa  
> setAcl /dubbo ip:127.0.0.1:cdrwa  
> getAcl /zookeeper  
> getAcl /dubbo
```

#####

```
1 []  
# echo -n admin:shuncom2004 | openssl dgst -binary -sha1 | openssl base64  
j9pfb5lKAqerlC8/RmfV2lq+1HQ=
```

2 []

```
# /usr/local/zookeeper-3.4.13/bin/zkCli.sh -server 127.0.0.1:2181  
> setAcl /zookeeper digest:admin:j9pfb5lKAqerlC8/RmfV2lq+1HQ=:cdrwa
```

```
> getAcl /zookeeper  
'digest,'admin:j9pfb5lKAqerlC8/RmfV2lq+1HQ=  
: cdrwa
```

```
> ls /zookeeper  
Authentication is not valid : /zookeeper
```

```
# []  
> addauth digest admin:shuncom2004  
> ls /zookeeper  
[quota]
```

#####

```
# vsftpd[]  
firewall-cmd --add-port=33880/tcp --permanent  
firewall-cmd --add-port=10001-10005/tcp --permanent  
firewall-cmd --add-port=30080/tcp --permanent
```

```
# []  
firewall-cmd --add-port=6011/tcp --permanent  
firewall-cmd --add-port=6013/tcp --permanent  
firewall-cmd --add-port=6014/tcp --permanent  
firewall-cmd --add-port=6012/udp --permanent
```

```
# tomcat  
firewall-cmd --add-port=8080/tcp --permanent
```

```
# shuncom-connect.jar
firewall-cmd --add-port=8088/tcp --permanent
firewall-cmd --add-port=8089/tcp --permanent
firewall-cmd --add-port=8094/tcp --permanent
firewall-cmd --add-port=8099/tcp --permanent
firewall-cmd --add-port=8091/tcp --permanent
```

```
# connect2-server.jar
firewall-cmd --add-port=8074/tcp --permanent
firewall-cmd --add-port=9097/tcp --permanent
```

```
firewall-cmd --reload
firewall-cmd --list-all
```

# OpenSSH\_6.6.1p1???openssh-9.1p1

## 1.1 安装 zlib

```
tar -xzf zlib-1.2.11.tar.gz && cd zlib-1.2.11
```

```
./configure --prefix=/usr/local/zlib && make -j 8 && make install
```

更新ld.so.conf

```
echo "/usr/local/zlib/lib" >> /etc/ld.so.conf
ldconfig -v
```

## 1.2 安装 openssl

```
tar -xzf openssl-OpenSSL_1_1_0k.tar.gz
```

```
./config --shared --prefix=/usr/local/openssl --libdir=/usr/local/openssl/lib && make -j 8 && make install
mv /usr/bin/openssl /usr/bin/openssl_bak && mv /usr/include/openssl/ /usr/include/openssl_bak/
ln -s /usr/local/openssl/bin/openssl /usr/bin/openssl && ln -s /usr/local/openssl/include/openssl/ /usr/include/openssl
```

更新ld.so.conf.d

创建openssl.conf

```
echo "/usr/local/openssl/lib" > /etc/ld.so.conf.d/openssl.conf
ldconfig -v
```

## 1.3 安装 openssh

```
tar -xzf openssh-9.1p1.tar.gz
```

```
mv /etc/ssh/ /etc/ssh_bak/ && mv /etc/init.d/ssh /etc/init.d/ssh_bak && mv /usr/bin/ssh /usr/bin/ssh_bak && cd openssh-9.1p1/
```

```
./configure --with-pam --sysconfdir=/etc/ssh --with-md5-passwords --with-zlib=/usr/local/zlib --with-ssl-dir=/usr/local/openssl --with-privsep-path=/var/lib/ssh
make -j 8 && make install
```

## 1.4 安装 openssh 9.8p1

```
tar -xzf openssl-1.1.1w.tar.gz
```

```
cd openssl-1.1.1w
```

```
./config --prefix=/usr/local/openssl --shared && make -j 8 && make install
```



Web

```

# CORS
access_log off;
server {
    listen 80;
    server_name localhost;

    # Add trusted domain here
    Access-Control-Allow-Origin: *

    # Add trusted domain here
    Access-Control-Allow-Origin: https://trusted-domain.com


    # Add trusted domain here
    cookies { HTTP
        Access-Control-Allow-Credentials: true

    Access-Control-Allow-Credentials: true
    Access-Control-Allow-Origin: https://trusted-domain.com


    # Add trusted domain here
    HTTP {
        HTTP

    Access-Control-Allow-Methods: GET, POST


    # Add trusted domain here
    Content-Type application/javascript; charset=utf-8;

    Access-Control-Allow-Headers: Content-Type, Authorization


    # Add trusted domain here
    Access-Control-Max-Age 600

    # Add trusted domain here
    CORS
}

Nginx
```

```
server {
    listen 80;
    server_name api.example.com;

    location /api/ {
        add_header 'Access-Control-Allow-Origin' 'https://trusted-domain.com';
        add_header 'Access-Control-Allow-Methods' 'GET, POST';
        add_header 'Access-Control-Allow-Headers' 'Content-Type, Authorization';
        add_header 'Access-Control-Allow-Credentials' 'true';
        add_header 'Access-Control-Max-Age' '600';

        if ($request_method = 'OPTIONS') {
            return 204;
        }

        proxy_pass http://backend-server;
    }
}
```

□□□ Apache □□□□ CORS □□□□

□□□□ CORS □□□□

Python Flask





# nginx-1.24.0???1.27.0

nginx-1.24.0 nginx-1.27.0

nginx-1.24.0

sudo /usr/local/nginx-1.24.0/sbin/nginx -s stop

Ubuntu 14.04

Nginx 1.27.0

SSL

1.

sudo apt-get update

sudo apt-get install -y build-essential libpcre3 libpcre3-dev zlib1g zlib1g-dev

2. Nginx

cd /usr/local/src

wget http://nginx.org/download/nginx-1.27.0.tar.gz

tar -zxvf nginx-1.27.0.tar.gz

cd nginx-1.27.0

3.

./configure --user=www-data --group=www-data --with-http\_ssl\_module --with-http\_stub\_status\_module --prefix=/usr/local/nginx-1.27.0 --with-openssl=/usr/local/openssl

./configure --user=www-data --group=www-data --with-http\_stub\_status\_module --prefix=/usr/local/nginx-1.27.4

./configure --prefix=/usr/local/nginx-1.27.0

./configure

SSL

./configure \

--prefix=/usr/local/nginx-1.27.0 \

--with-http\_ssl\_module \

--with-openssl=/usr/local/openssl \

--with-pcre \

--with-zlib=/usr/include

--prefix=/usr/local/nginx-1.27.0

--with-http\_ssl\_module SSL

```
--with-openssl=/usr/local/openssl OpenSSL
--with-pcre
--with-zlib=/usr/include zlib  gzip
```

4. vim auto/lib/openssl/conf

```
make -j 8
make -j$(nproc)
sudo make install
```

```
mkdir -p /usr/local/nginx-1.27.4/conf/conf.d
```

```
vi /usr/local/nginx-1.27.4/conf/nginx.conf
```

```
http {
```

```
http {
```

```
    include      mime.types;
    default_type  application/octet-stream;
```

```
#      nginx  conf.d/      .conf
include /usr/local/nginx-1.27.0/conf/conf.d/*.conf;
```

```
-j$(nproc) CPU
```

```
18  access_log  logs/light-access.log main;
19  error_log   logs/light-error.log warn;
```

5.

```
 Nginx
```

```
/usr/local/nginx-1.27.4/sbin/nginx -V
```

```
--with-http_ssl_module
```

6. Nginx

```
 Nginx
```

```
sudo /usr/local/nginx-1.27.0/sbin/nginx -s stop
```

```
 Nginx
```

```
sudo /usr/local/nginx-1.27.4/sbin/nginx
```

```
ps aux | grep nginx
netstat -tulnp | grep nginx
```

## 7. 配置 SSL 证书

编辑 SSL 配置文件 `/usr/local/nginx-1.27.0/conf/nginx.conf`

```
server {  
    listen 443 ssl;  
    server_name yourdomain.com;  
  
    ssl_certificate /etc/nginx/ssl/your_cert.pem;  
    ssl_certificate_key /etc/nginx/ssl/your_key.pem;  
  
    location / {  
        root /var/www/html;  
        index index.html index.htm;  
    }  
}
```

重启 Nginx

`sudo /usr/local/nginx-1.27.4/sbin/nginx -s reload`

检查 Nginx 1.27.4 是否支持 SSL 证书

检查 SSL 证书是否安装

`cat /usr/local/nginx-1.27.4/logs/error.log`

检查错误日志

检查 nginx-1.24.0 与 nginx-1.27.0 的依赖关系

`sudo ln -sf /usr/local/nginx-1.27.0/sbin/nginx /usr/local/bin/nginx`

`ln -sf /usr/local/nginx-1.27.4/sbin/nginx /usr/sbin/nginx`

检查 nginx 是否安装

检查 nginx-1.24.0 是否安装

`sudo ln -sf /usr/local/nginx-1.24.0/sbin/nginx /usr/local/bin/nginx`