



FreeBSD 安装 配置 网络 防火墙 数据库 用户管理 。

- [PostgreSQL 安装](#)
- [MySQL 安装 配置 DB 用户 , 备份 还原](#)
- [FreeBSD 安装 apcupsd 配置](#)
- [Certbot : 安装 配置 证书](#)
- [Munge 安装](#)
- [Shell 安装 sh 配置 bash 用户](#)
- [alias 安装 bash shell 配置 用户 脚本 \(.bashrc 安装 .bash_profile\)](#)
- [exFAT 格式化 挂载](#)
- [ssh 安装 配置](#)
- [GitHub Pull 合并 分支 仓库](#)
- [tar 打包](#)

PostgreSQL 安装

[FreeBSD Wiki](#) 提供了详细的安装指南。

2024年01月14日，在FreeBSD 16上安装PostgreSQL 16。

以root用户身份运行以下命令。
\$ sudo

安装

首先，安装PostgreSQL 16的服务器和客户端包。

```
$ pkg install postgresql16-server postgresql16-client
```

安装完成后，需要配置PostgreSQL 16的配置文件。

```
$ pkg search postgresql
```

通过以下命令查看PostgreSQL 16的配置文件位置。

```
$ sysrc postgresql_enable="YES"
```

然后，启动PostgreSQL 16服务。

```
$ /usr/local/etc/rc.d/postgresql initdb
```

```
$ service postgresql start
```

安装完成后，可以通过以下命令验证PostgreSQL 16是否安装成功。

```
$ sockstat -4 | grep 5432
```

```
<pid pid pid pid>
```

```
$ sockstat -4 | grep 5432 -> IPv4 pid pid .
```

```
$ sockstat -6 | grep 5432 -> IPv6 pid pid .
```

最后，创建PostgreSQL 16的数据库。


```
$ sudo passwd postgres
```

Enter new password:
Enter it again:
postgres admin password:
postgres admin password: . foo_db
postgres admin password:
postgres admin password: .

```
# su - postgres (or sudo su - postgres if you have sudo)
$ createuser admin
$ createdb foo_db -O admin
```

Now you can connect to the database (psql command) as the postgres user.
To exit the psql command, type **exit** or press Ctrl+D.
To connect to the database, type **psql foo_db** and press Enter.

```
$ psql foo_db
foo_db=# alter role admin with encrypted password 'yourpassword';
foo_db=# grant all privileges on database foo_db to admin;
foo_db=# exit
$ exit
```

Now you can connect to the database as the admin user.
To connect to the database, type **psql foo_db** and press Enter.
To exit the psql command, type **exit** or press Ctrl+D.

```
$ dropdb foo_db
```

Now you can connect to the database as the postgres user.
To connect to the database, type **psql foo_db** and press Enter.



Bind address

The bind address is the IP address that the database listens on.
To change the bind address, edit the postgresql.conf file.
data16: The bind address is the IP address that the database listens on.
To change the bind address, edit the postgresql.conf file.

```
sudo nano /var/db/postgres/data16/postgresql.conf
```

The listen_addresses parameter is the IP address that the database listens on.
To change the listen_addresses parameter, edit the postgresql.conf file.
listen_addresses: The IP address that the database listens on.
To change the listen_addresses parameter, edit the postgresql.conf file.


```
listen_addresses = '*'
```



```
sudo nano /var/db/postgres/data16/pg_hba.conf
```

```

def md5(filename):
    """Return the MD5 hash of the file at filename.

    The MD5 hash is a 32-character hexadecimal string.

    """
    import hashlib
    m = hashlib.md5()
    with open(filename, 'rb') as f:
        for chunk in iter(lambda: f.read(4096), b''):
            m.update(chunk)
    return m.hexdigest()

```

```
# TYPE DATABASE USER ADDRESS METHOD

# "local" is for Unix domain socket connections only
local all all trust

# IPv4 local connections:
host all all 127.0.0.1/32 trust

# IPv6 local connections:
host all all ::1/128 trust

# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all trust
host replication all 127.0.0.1/32 trust
host replication all ::1/128 trust
```

1. 在 `admin` 表中，添加一个名为 `foo_db` 的列，数据类型为 `varchar(255)`。
 2. 在 `admin` 表中，添加一个名为 `foo_db` 的列，数据类型为 `varchar(255)`。
 3. 在 `admin` 表中，添加一个名为 `foo_db` 的列，数据类型为 `varchar(255)`。

```
# Allow remote connections to foo_db as admin

host  foo_db      admin      0.0.0.0/0      md5 -> IPv4 [ ][ ][ ][ ]
host  foo_db      admin      ::/0           md5 -> IPv6 [ ][ ][ ][ ]
```

```
sudo service postgresql restart
```


我们 在 代码 中 , 使用 了 一个 叫 做 的 包 来 做 这 个 工 作 。 这 个 包 叫 做 `md5` , 它 是 用 来 做 这 个 工 作 的 。 这 个 包 叫 做 `trust` , 它 是 用 来 做 这 个 工 作 的 。 这 个 包 叫 做 `(4)` , 它 是 用 来 做 这 个 工 作 的 。

MySQL 数据库 数据库 ,

数据库 数据库

数据库 数据库 DB 数据库 数据库 数据库 .

数据库 数据库 .

数据库 数据库

数据库 数据库 数据库 数据库 数据库 数据库 数据库 数据库 数据库 数据库 数据库 数据库 数据库 数据库 数据库 数据库 .

```
create user 数据库 ID;
```

```
create user 数据库 ID@'localhost' identified by '数据库 ';
```

```
create user '数据库 ID'@'localhost' identified by 'PASSWORD';
```

数据库 **'localhost'**数据库 "数据库 数据库 数据库 数据库 数据库 数据库 数据库 . 数据库 数据库 IP 数据库 "数据库 IP数据库 数据库 数据库 数据库 .

数据库 数据库

数据库 数据库 数据库 数据库 character set utf8 数据库 数据库 数据库 数据库 .

```
mysql > create database DB default character set utf8;
```

数据库

mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY 'PASSWORD';

```
mysql> GRANT ALL privileges ON DB.* TO UserId@'localhost' IDENTIFIED BY 'PASSWORD';
```

mysql> FLUSH PRIVILEGES;

```
mysql > flush privileges;
```

MySQL

mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY 'PASSWORD';

mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY 'PASSWORD';

mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY 'PASSWORD';

FreeBSD 安装 apcupsd 教程

apcupsd 是 APC UPS 的守护进程，用于管理 UPS 设备。它可以在 FreeBSD 系统中安装和配置，以便通过 OS 接口管理 UPS 设备。

安装

APC UPS 的 AP9631 Network Management Card 可以通过网络接口管理。UPS 也可以通过 USB 接口连接到系统，通过 USB 接口管理。

安装和配置 apcupsd 的步骤如下：

安装

1. 更新系统软件包。

```
sudo pkg update
sudo pkg install apcupsd
```

2. 编辑配置文件 **/usr/local/etc/apcupsd/apcupsd.conf**，配置 UPS 设备。

```
sudo nano /usr/local/etc/apcupsd/apcupsd.conf
```

3. 重启服务。

```
# 配置 UPS 设备名称
UPSNAME APCUPS

# SNMP 设备名称
UPSCABLE ether

# SNMP 设备地址
UPSTYPE snmp

# UPS 设备 IP 地址
# 设备名称 161 设备名称 apc, 设备名称 public
DEVICE 192.168.0.255:161:apc:public
```



```
# 0.0.0.0 127.0.0.1 127.0.0.1 .
NETSERVER on

# 0.0.0.0 127.0.0.1 127.0.0.1 localhost 127.0.0.1 .
NISIP 127.0.0.1

# NIS 127.0.0.1 3551 127.0.0.1 .
NISPORT 3551

# UPS 127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 .
UPSCCLASS shareslave

# 127.0.0.1 127.0.0.1 .
UPSMODE disable
```

127.0.0.1 127.0.0.1 127.0.0.1 127.0.0.1 /etc/rc.conf 127.0.0.1 .

```
sudo service apcupsd enable
sudo service apcupsd start
or
sudo sysrc apcupsd_enable="YES"
sudo shutdown -r now
# 127.0.0.1 ...
```

127.0.0.1 127.0.0.1 127.0.0.1 apcaccess 127.0.0.1 NIS 127.0.0.1 .

```
$ apcaccess
APC      : 001,044,1028
DATE     : 2024-01-17 22:17:59 +0900
HOSTNAME : blurblur
VERSION  : 3.14.14 (31 May 2016) freebsd
UPSNAME  : APCUPS          # 127.0.0.1 127.0.0.1 .
CABLE    : Ethernet Link
DRIVER   : SNMP UPS Driver
UPSMODE  : ShareUPS Slave
STARTTIME: 2024-01-17 22:17:57 +0900
MODEL    : Smart-UPS 1500  # 127.0.0.1 127.0.0.1 .
STATUS   : ONLINE          # 127.0.0.1 ONLINE 127.0.0.1 127.0.0.1 .
LINEV    : 205.0 Volts
LOADPCT  : 24.0 Percent
BCHARGE  : 100.0 Percent
TIMELEFT : 52.0 Minutes
MBATTCHG : 5 Percent
MINTIMEL : 3 Minutes
```


MAXTIME : 0 Seconds
MAXLINEV : 205.0 Volts
MINLINEV : 204.0 Volts
OUTPUTV : 205.0 Volts
SENSE : High
DWAKE : 1000 Seconds
DSHUTD : 20 Seconds
DLOWBATT : 5 Minutes
LOTRANS : 195.0 Volts
HITRANS : 265.0 Volts
ITEMP : 23.0 C
ALARMDEL : 30 Seconds
BATTV : 27.0 Volts
LINEFREQ : 59.0 Hz
LASTXFER : No transfers since turnon
NUMXFERS : 0
TONBATT : 0 Seconds
CUMONBATT: 0 Seconds
XOFFBATT : N/A
SELFTTEST : OK
STESTI : OFF
STATFLAG : 0x05000008
MANDATE : 09/12/2019
SERIALNO : 3S1937X12596
BATTDATE : 12/15/2019
NOMOUTV : 230 Volts
FIRMWARE : UPS 09.3 (ID18)
END APC : 2024-01-17 22:18:20 +0900

apcupsd.conf

apcupsd.conf

```
## apcupsd.conf v1.1 ##  
#  
# for apcupsd release 3.14.14 (31 May 2016) - freebsd  
#
```



```
# "apcupsd" POSIX config file
```

#

□ □ □ □ □ □ □ apcupsd □ □ □ □ □ .

□ □□ □□□ □□□□□ □□□□□ .

#

#

===== □ □ □ □ =====

#

```
# UPSNAME xxx
```

UPS .

  UPS                                                  

EEPROM . 8 .

UPSNAMES FOO

```
# UPSCABLE < cable>
```

UPS .

#

<cabl> 1 2 3 4 5 6 7 :

```
# simple, smart, ether, usb
```

#

∴

940-0119A, 940-0127A, 940-0128A, 940-0020B,

940-0020C, 940-0023A, 940-0024B, 940-0024C,

940-1524C, 940-0024G, 940-0095A, 940-0095B,

940-0095C, 940-0625A, M-04-02-2000

#

UPSCABLE ether

apcupsd

UPS UPSTYPE ().

```
# 00000000000000000000 DEVICE 00000000000000000000 .
```

```
# USB UPS  1  DEVICE  1  1  1
```

UPS

#

#	UPSTYPE	DEVICE	Description
---	---------	--------	-------------

```
# apcsmart /dev/tty**
```


--	--	--

--	--

--	--

--	--

--	--	--	--	--

 .


```

#
# usb <BLANK>  1  UPS  USB  .
#
#      1  1  1  1  1  1
#
#      1  1  1  1  .
#
#
# net  hostname:port  apcupsd  1  1  1  1
#
#      apcupsd  1  1  1  .  1  1  1
#
#      UPS  1  1  1  1  1
#
#      1  1  .
#
#
# snmp  hostname:port:vendor:community
#
#      SNMP  1  UPS  1  SNMP  1  .
#
#      hostname  1  UPS  IP  1  1  1  .
#
#      vendor  "APC"  "APC_NOTRAP"  1  .
#
#      "APC_NOTRAP"  SNMP  1  1
#
#      1  "APC"  1  .  1  161  .
#
#      1  "private"  .
#
#
# netsnmp  hostname:port:vendor:community
#
#      OBSOLETE
#
#      1  SNMP  1  net-snmp  1  1  .
#
#      1  1  1  1  1  1
#
#      'snmp'  1  1  .
#
#
# dumb  /dev/tty**  simple-signaling UPS  1  1  1
#
#      1  1  1  .
#
#
# pcnet  ipaddr:username:passphrase:port
#
#      AP9617  1  1  1  SNMP  1
#
#      1  1  PowerChute  1  1
#
#      password  1  1  1  . port  UPS
#
#      1  1  1  (  3052 )  .  1
#
#      1  1  1  1  3052  1  .
#
#
# modbus  /dev/tty**  MODBUS  1  1  SmartUPS
#
#      1  1  1  1  1  .
#
# modbus  <BLANK>  MODBUS over USB  1  1  1  1
#
#      UPS  1  1  1  apcupsd  1  1
#
#      1  1  (USB UPS  1  1  1  ).
#
#

```


UPSTYPE snmp

DEVICE IP:PORT:APC:PRIVATE

POLLTIME <int>

apcupsd 的 UPS 的 名称 在 () 中。

在 名称 中 UPS(UPSTYPE apcsmart, usb, dumb) 的 名称。

UPS(UPSTYPE net, snmp) 的 名称。在 名称 中 CPU 的 名称。

在 名称 中 apcupsd 的 名称。在 名称 中 60 的 名称。

#POLLTIME 60

LOCKFILE <path to lockfile>

在 名称 中 名称。在 名称 中 名称。在 名称。

在 名称 中, apcupsd 的 名称 名称。

在 名称 中 名称 DEVICE 的 名称。

Win32 的 名称。

LOCKFILE /var/spool/lock

SCRIPTDIR <path to script directory>

apccontrol 的 名称 名称。

SCRIPTDIR /usr/local/etc/apcupsd

PWRFAILDIR <path to powerfail directory>

在 名称 中 名称 名称。在 名称 apcupsd 的

名称 名称 名称 名称, OS 的 名称 killpower

(UPS 的 名称) 的 名称 名称。

PWRFAILDIR /var/run

NOLOGINDIR <path to nologin directory>

nologin 的 名称 名称。在 名称 名称 名称

OS 的 名称 名称 名称。

NOLOGINDIR /var/run

#

===== 名称 名称 名称 =====

#

ONBATTERYDELAY 的 名称 名称

名称 名称 名称 名称 名称 () 名称。

#


```
# , 在 powerout 之前 apccontrol 被调用 。
# 在 onbattery 之前 onbatterydelay 被调用 apccontrol 被调用 。
# 在 之前 apccontrol powerout 被调用 之前 被调用
# 在 之前 被调用 。
```

ONBATTERYDELAY 6

```
#
# Note: BATTERYLEVEL, MINUTES 和 TIMEOUT 被调用
# 在 之前 被调用 。
```

```
# 在 之前 被调用 (UPS 被调用 ) 在 之前 被调用 ,
# apcupsd 在 之前 被调用 。
```

BATTERYLEVEL 5

```
# 在 之前 被调用 ( 在 ) (UPS 被调用 之前 )
# 在 , apcupsd 在 之前 被调用 。
```

MINUTES 3

```
# 在 UPS 被调用 之前 TIMEOUT 被调用
# 在 apcupsd 在 之前 被调用 。
# 在 0 之前 被调用 。
```

TIMEOUT 0

```
# 在 之前 被调用 signoff 在 之前 被调用 ( ) 。
```

ANNOY 300

```
# 在 之前 被调用 在 之前 被调用 在 之前 被调用 。
```

ANNOYDELAY 60

```
# 在 之前 被调用 在 之前 被调用 。
```

```
# NOLOGON <string> [ disable | timeout | percent | minutes | always ]
```

NOLOGON disable


```

# KILLDELAY 0 秒 间隔 , 每隔 秒 检查 ( ) 秒 间隔
# 每隔 秒 检查 秒 间隔 apcupsd 秒 间隔 .
# 每隔 秒 检查 秒 间隔 apcupsd 秒 间隔 秒 间隔 秒 间隔 .
# KILLDELAY <seconds> 0 disables
KILLDELAY 0

#

# ==== 秒 间隔 秒 间隔 秒 间隔 ====
#

# NETSERVER [ on | off ] on enables, off disables the network
# information server. If netstatus is on, a network information
# server process will be started for serving the STATUS and
# EVENT data over the network (used by CGI programs).
NETSERVER on

# NISIP <dotted notation ip address>
# IP address on which NIS server will listen for incoming connections.
# This is useful if your server is multi-homed (has more than one
# network interface and IP address). Default value is 0.0.0.0 which
# means any incoming request will be serviced. Alternatively, you can
# configure this setting to any specific IP address of your server and
# NIS will listen for connections only on that interface. Use the
# loopback address (127.0.0.1) to accept connections only from the
# local machine.
NISIP 0.0.0.0

# NISPORT <port> default is 3551 as registered with the IANA
# port to use for sending STATUS and EVENTS data over the network.
# It is not used unless NETSERVER is on. If you change this port,
# you will need to change the corresponding value in the cgi directory
# and rebuild the cgi programs.
NISPORT 3551

# If you want the last few EVENTS to be available over the network
# by the network information server, you must define an EVENTSFILE.
EVENTSFILE /var/log/apcupsd.events

# EVENTSFILEMAX <kilobytes>

```



```
# By default, the size of the EVENTSFILE will be not be allowed to exceed
# 10 kilobytes. When the file grows beyond this limit, older EVENTS will
# be removed from the beginning of the file (first in first out). The
# parameter EVENTSFILEMAX can be set to a different kilobyte value, or set
# to zero to allow the EVENTSFILE to grow without limit.
EVENTSFILEMAX 10

#
# ===== Configuration statements used if sharing =====
#      a UPS with more than one machine

#
# Remaining items are for ShareUPS (APC expansion card) ONLY
#

# UPSCLASS [ standalone | shareslave | sharemaster ]
# Normally standalone unless you share an UPS using an APC ShareUPS
# card.
UPSCLASS shareslave

# UPSMODE [ disable | share ]
# Normally disable unless you share an UPS using an APC ShareUPS card.
UPSMODE disable

#
# ===== Configuration statements to control apcupsd system logging =====
#

# Time interval in seconds between writing the STATUS file; 0 disables
STATTIME 0

# Location of STATUS file (written to only if STATTIME is non-zero)
STATFILE /var/log/apcupsd.status

# LOGSTATS [ on | off ] on enables, off disables
# Note! This generates a lot of output, so if
# you turn this on, be sure that the
# file defined in syslog.conf for LOG_NOTICE is a named pipe.
# You probably do not want this on.
LOGSTATS off
```



```
# Time interval in seconds between writing the DATA records to
# the log file. 0 disables.
DATETIME 0

# FACILITY defines the logging facility (class) for logging to syslog.
# If not specified, it defaults to "daemon". This is useful
# if you want to separate the data logged by apcupsd from other
# programs.
#FACILITY DAEMON

#
# ===== Configuration statements used in updating the UPS EPROM =====
#

#
# These statements are used only by apctest when choosing "Set EEPROM with conf
# file values" from the EEPROM menu. THESE STATEMENTS HAVE NO EFFECT ON APCUPSD.
#

# UPS name, max 8 characters
#UPSNAME UPS_IDEN

# Battery date - 8 characters
#BATTDATE mm/dd/yy

# Sensitivity to line voltage quality (H cause faster transfer to batteries)
# SENSITIVITY H M L (default = H)
#SENSITIVITY H

# UPS delay after power return (seconds)
# WAKEUP 000 060 180 300 (default = 0)
#WAKEUP 60

# UPS Grace period after request to power off (seconds)
# SLEEP 020 180 300 600 (default = 20)
#SLEEP 180

# Low line voltage causing transfer to batteries
# The permitted values depend on your model as defined by last letter
```


of FIRMWARE or APCMODEL. Some representative values are:

D 106 103 100 097

M 177 172 168 182

A 092 090 088 086

I 208 204 200 196 (default = 0 => not valid)

#LOTTRANSFER 208

High line voltage causing transfer to batteries

The permitted values depend on your model as defined by last letter

of FIRMWARE or APCMODEL. Some representative values are:

D 127 130 133 136

M 229 234 239 224

A 108 110 112 114

I 253 257 261 265 (default = 0 => not valid)

#HITTRANSFER 253

Battery charge needed to restore power

RETURNCHARGE 00 15 50 90 (default = 15)

#RETURNCHARGE 15

Alarm delay

0 = zero delay after pwr fail, T = power fail + 30 sec, L = low battery, N = never

BEEPSTATE 0 T L N (default = 0)

#BEEPSTATE T

Low battery warning delay in minutes

LOWBATT 02 05 07 10 (default = 02)

#LOWBATT 2

UPS Output voltage when running on batteries

The permitted values depend on your model as defined by last letter

of FIRMWARE or APCMODEL. Some representative values are:

D 115

M 208

A 100

I 230 240 220 225 (default = 0 => not valid)

#OUTPUTVOLTS 230

Self test interval in hours 336=2 weeks, 168=1 week, ON=at power on

SELFTEST 336 168 ON OFF (default = 336)

Certbot : 如何安装与使用

安装与配置

- 安装 Certbot 并配置 Nginx 或 Apache 服务器。
- 80 端口 监听 请求 并 响应。

80 端口 监听 请求 并 响应。

FreeBSD 系统 上 安装 并 配置。IPv6 支持 -4 选项 -6 选项。

```
sudo sockstat -4 -l
```

安装 net-tools 并 使用 sudo apt install net-tools 命令 安装。

```
netstat -tnlp | grep 80
```

Nginx 或 Apache 服务器 配置 并 启动。

使用 Certbot 获取证书

使用 命令。

```
sudo certbot certificates
```

安装 并 配置 服务器。安装 并 配置 服务器。

```
sudo certbot delete
```

安装 并 配置 服务器。

使用 Certbot 更新证书

使用 命令。

```
sudo certbot renew
```



```

[ ] [ ] [ ][ ][ ] [ ] [ ][ ][ ] [ ][ ][ ] [ ] --dry-run [ ][ ][ ] [ ][ ][ ] . [ ][ ][ ][ ] [ ][ ][ ][ ] 1[ ][ ][ ][ ] [ ][ ][ ][ ]
[ ][ ][ ][ ][ ] [ ][ ][ ][ ] [ ][ ][ ][ ] [ ][ ][ ][ ][ ] .

```


Munge 𐌿

𐌿 𐌿 𐌿 𐌿 𐌿𐌿𐌿𐌿 .

- 𐌿𐌿𐌿𐌿 𐌿𐌿𐌿 𐌿𐌿𐌿 𐌿 𐌿𐌿 𐌿𐌿𐌿𐌿 𐌿𐌿𐌿𐌿 𐌿𐌿 .
- (𐌿𐌿𐌿 𐌿𐌿)𐌿 𐌿𐌿 𐌿𐌿𐌿 𐌿𐌿 .
- 𐌿𐌿 𐌿𐌿𐌿 𐌿𐌿 𐌿𐌿𐌿 𐌿𐌿 𐌿𐌿𐌿𐌿 𐌿𐌿𐌿 𐌿𐌿 𐌿𐌿𐌿 𐌿𐌿𐌿𐌿𐌿 𐌿𐌿 .
- 𐌿𐌿𐌿 , 𐌿𐌿 , 𐌿𐌿𐌿𐌿 𐌿𐌿 𐌿𐌿𐌿𐌿 𐌿𐌿𐌿𐌿 𐌿𐌿𐌿𐌿𐌿𐌿 𐌿

Shell sh bash

bash shell

bash

```
sudo pkg install bash
```

bash is installed. To use bash, you need to change your shell to bash. You can do this by running the command `chsh`.

```
$ which bash
/usr/local/bin/bash
```

`/usr/local/bin/bash` is the path to the bash shell. To change your shell to bash, you can run the command `chsh` (change shell). You can also run `chsh -s /usr/local/bin/bash` to change your shell to bash. You can check your current shell by running `user id`.

```
$ chsh -s /usr/local/bin/bash # bash
$ chsh -s /usr/local/bin/bash ID # ID
```

chsh: user information updated

```
chsh: user information updated
```

bash is installed. To use bash, you need to change your shell to bash. You can do this by running the command `chsh`. You can also run `chsh -s /usr/local/bin/bash` to change your shell to bash. You can check your current shell by running `user id`.

alias 在 bash shell 中 使用 (.bashrc 与 .bash_profile)

在 终端 中 输入 以下 命令。
FreeBSD 14.0 中 bash shell 的 .bashrc 与 .bash_profile 文件 位于 /etc 目录下。

在 终端 中 输入 ls 命令 查看 文件 列表。在 文件 列表 中 找到 alias 文件。

```
$ nano .bash_profile
```

在 终端 中 输入 .bash_profile 命令 加载 配置文件。

```
alias ll='ls -G'
alias lsg='ls -alG'
```

在 终端 中 输入 ? 命令 查看 别名 列表。在 别名 列表 中 找到 ll 与 lsg 别名。

在 终端 中 输入 alias 命令 查看 别名 列表。

```
$ alias
```

在 终端 中 输入 alias 命令 查看 别名 列表。

```
$ ualias <alias 名称 >
```

在 终端 中 输入 .bash_profile 命令 加载 配置文件。

exFAT



<https://www.micski.dk/2021/04/10/how-to-mount-exfat-formatted-sd-memory-card-on-freebsd/>

FreeBSD GUI

```
$ sudo pkg install fusefs-exfat
$ sudo kldload fusefs
$ sudo nano /boot/loader.conf
#
fusefs_load="YES"
$ sudo shutdown -r now
```


ssh 키 생성 및 사용

<https://cho001.tistory.com/20>

1. ssh key pair 생성

```
# -b 4096 2048비트 키 생성
ssh-keygen -t rsa -b 4096
```

2. ssh-copy-id 사용하여 원격 호스트에 키 복사

```
# <remote-host>에 키 복사 (호스트명)
# : foobar@192.168.0.230
$ ssh-copy-id -i ~/.ssh/id_rsa.pub <remote-host>
```

3. scp를 사용하여 원격 호스트에 파일 복사

4. ssh로 원격 호스트에 접속

GitHub Pull merge



1. Clone the repository
2. Create a new branch
3. Make changes
4. Commit changes
5. Push changes to GitHub
6. Create a Pull Request
7. Review changes
8. Merge changes

```
$ sudo git fetch --all
$ sudo git reset --hard origin/master
# BookStack release/master
$ sudo git pull
# BookStack git pull origin release
```


tar 1111

11 1 11 1111 . LTO111 111 11111 1 1 1111 111 .
11111 -z 111 11 1111 . 1 111 111 111 111 1111 PDF, 111 111
1111 11 111 11 11 1111 111 1111 111 1111 111 1111 . 111 1
1111 11 111 111 111 11 1 1111 .

```
$ tar -cvf- TargetDir | split -b 1024m - DestinationFile.tar
```

- TargetDir : 111 11 1111
- 1024m : 111111 111 11 (1111 1GB11)
- DestinationFile.tar : 11 111 11

111 . 111 111 File.tara File.tarb File.tarc 11 111 1111 111111 .

111 1 1

```
$ cat CompFile.tar* | tar xvf -
```

111 111 111 .

1111 : <https://superad.tistory.com/74>