<http://www.cnblogs.com/ggjucheng/archive/2012/08/19/2646032.html>

基本上，所有的 sshd 服务器详细设定都放在 /etc/ssh/sshd\_config 里面！不过，每个 Linux distribution 的预设设定都不太相同，所以我们有必要来了解一下整个设定值的意义为何才好！ 同时请注意，在预设的档案内，只要是预设有出现且被批注的设定值 (设定值前面加 #)，即为『默认值！』，你可以依据它来修改的哩

[复制代码](javascript:void(0);)

[root@www ~]# vim /etc/ssh/sshd\_config

# 1. 关于 SSH Server 的整体设定，包含使用的 port 啦，以及使用的密码演算方式

# Port 22

# SSH 预设使用 22 这个port，也可以使用多个port，即重复使用 port 这个设定项目！

# 例如想要开放 sshd 在 22 与 443 ，则多加一行内容为：『 Port 443 』

# 然后重新启动 sshd 这样就好了！不过，不建议修改 port number 啦！

Protocol 2

# 选择的 SSH 协议版本，可以是 1 也可以是 2 ，CentOS 5.x 预设是仅支援 V2。

# 如果想要支持旧版 V1 ，就得要使用『 Protocol 2,1 』才行。

# ListenAddress 0.0.0.0

# 监听的主机适配器！举个例子来说，如果你有两个 IP，分别是 192.168.1.100 及

# 192.168.100.254，假设你只想要让 192.168.1.100 可以监听 sshd ，那就这样写：

# 『 ListenAddress 192.168.1.100 』默认值是监听所有接口的 SSH 要求

# PidFile /var/run/sshd.pid

# 可以放置 SSHD 这个 PID 的档案！上述为默认值

# LoginGraceTime 2m

# 当使用者连上 SSH server 之后，会出现输入密码的画面，在该画面中，

# 在多久时间内没有成功连上 SSH server 就强迫断线！若无单位则默认时间为秒！

# Compression delayed

# 指定何时开始使用压缩数据模式进行传输。有 yes, no 与登入后才将数据压缩 (delayed)

# 2. 说明主机的 Private Key 放置的档案，预设使用下面的档案即可！

# HostKey /etc/ssh/ssh\_host\_key # SSH version 1 使用的私钥

# HostKey /etc/ssh/ssh\_host\_rsa\_key # SSH version 2 使用的 RSA 私钥

# HostKey /etc/ssh/ssh\_host\_dsa\_key # SSH version 2 使用的 DSA 私钥

# 还记得我们在主机的 SSH 联机流程里面谈到的，这里就是 Host Key ～

# 3. 关于登录文件的讯息数据放置与 daemon 的名称！

SyslogFacility AUTHPRIV

# 当有人使用 SSH 登入系统的时候，SSH 会记录信息，这个信息要记录在什么 daemon name

# 底下？预设是以 AUTH 来设定的，即是 /var/log/secure 里面！什么？忘记了！

# 回到 Linux 基础去翻一下。其他可用的 daemon name 为：DAEMON,USER,AUTH,

# LOCAL0,LOCAL1,LOCAL2,LOCAL3,LOCAL4,LOCAL5,

# LogLevel INFO

# 登录记录的等级！嘿嘿！任何讯息！同样的，忘记了就回去参考！

# 4. 安全设定项目！极重要！

# 4.1 登入设定部分

# PermitRootLogin yes

# 是否允许 root 登入！预设是允许的，但是建议设定成 no！

# StrictModes yes

# 是否让 sshd 去检查用户家目录或相关档案的权限数据，

# 这是为了担心使用者将某些重要档案的权限设错，可能会导致一些问题所致。

# 例如使用者的 ~.ssh/ 权限设错时，某些特殊情况下会不许用户登入

# PubkeyAuthentication yes

# AuthorizedKeysFile .ssh/authorized\_keys

# 是否允许用户自行使用成对的密钥系统进行登入行为，仅针对 version 2。

# 至于自制的公钥数据就放置于用户家目录下的 .ssh/authorized\_keys 内

PasswordAuthentication yes

# 密码验证当然是需要的！所以这里写 yes 啰！

# PermitEmptyPasswords no

# 若上面那一项如果设定为 yes 的话，这一项就最好设定为 no ，

# 这个项目在是否允许以空的密码登入！当然不许！

# 4.2 认证部分

# RhostsAuthentication no

# 本机系统不使用 .rhosts，因为仅使用 .rhosts太不安全了，所以这里一定要设定为 no

# IgnoreRhosts yes

# 是否取消使用 ~/.ssh/.rhosts 来做为认证！当然是！

# RhostsRSAAuthentication no #

# 这个选项是专门给 version 1 用的，使用 rhosts 档案在 /etc/hosts.equiv

# 配合 RSA 演算方式来进行认证！不要使用啊！

# HostbasedAuthentication no

# 这个项目与上面的项目类似，不过是给 version 2 使用的！

# IgnoreUserKnownHosts no

# 是否忽略家目录内的 ~/.ssh/known\_hosts 这个档案所记录的主机内容？

# 当然不要忽略，所以这里就是 no 啦！

ChallengeResponseAuthentication no

# 允许任何的密码认证！所以，任何 login.conf 规定的认证方式，均可适用！

# 但目前我们比较喜欢使用 PAM 模块帮忙管理认证，因此这个选项可以设定为 no 喔！

UsePAM yes

# 利用 PAM 管理使用者认证有很多好处，可以记录与管理。

# 所以这里我们建议你使用 UsePAM 且 ChallengeResponseAuthentication 设定为 no

# 4.3 与 Kerberos 有关的参数设定！因为我们没有 Kerberos 主机，所以底下不用设定！

# KerberosAuthentication no

# KerberosOrLocalPasswd yes

# KerberosTicketCleanup yes

# KerberosTgtPassing no

# 4.4 底下是有关在 X-Window 底下使用的相关设定！

X11Forwarding yes

# X11DisplayOffset 10

# X11UseLocalhost yes

# 比较重要的是 X11Forwarding 项目，他可以让窗口的数据透过 ssh 信道来传送喔！

# 在本章后面比较进阶的 ssh 使用方法中会谈到。

# 4.5 登入后的项目：

# PrintMotd yes

# 登入后是否显示出一些信息呢？例如上次登入的时间、地点等等，预设是 yes

# 亦即是打印出 /etc/motd 这个档案的内容。但是，如果为了安全，可以考虑改为 no ！

# PrintLastLog yes

# 显示上次登入的信息！可以啊！预设也是 yes ！

# TCPKeepAlive yes

# 当达成联机后，服务器会一直传送 TCP 封包给客户端藉以判断对方式否一直存在联机。

# 不过，如果联机时中间的路由器暂时停止服务几秒钟，也会让联机中断喔！

# 在这个情况下，任何一端死掉后，SSH可以立刻知道！而不会有僵尸程序的发生！

# 但如果你的网络或路由器常常不稳定，那么可以设定为 no 的啦！

UsePrivilegeSeparation yes

# 是否权限较低的程序来提供用户操作。我们知道 sshd 启动在 port 22 ，

# 因此启动的程序是属于 root 的身份。那么当 student 登入后，这个设定值

# 会让 sshd 产生一个属于 sutdent 的 sshd 程序来使用，对系统较安全

MaxStartups 10

# 同时允许几个尚未登入的联机画面？当我们连上 SSH ，但是尚未输入密码时，

# 这个时候就是我们所谓的联机画面啦！在这个联机画面中，为了保护主机，

# 所以需要设定最大值，预设最多十个联机画面，而已经建立联机的不计算在这十个当中

# 4.6 关于用户抵挡的设定项目：

DenyUsers \*

# 设定受抵挡的使用者名称，如果是全部的使用者，那就是全部挡吧！

# 若是部分使用者，可以将该账号填入！例如下列！

DenyUsers test

DenyGroups test

# 与 DenyUsers 相同！仅抵挡几个群组而已！

# 5. 关于 SFTP 服务与其他的设定项目！

Subsystem sftp /usr/lib/ssh/sftp-server

# UseDNS yes

# 一般来说，为了要判断客户端来源是正常合法的，因此会使用 DNS 去反查客户端的主机名

# 不过如果是在内网互连，这项目设定为 no 会让联机达成速度比较快。

[复制代码](javascript:void(0);)

预设的 sshd 服务已经算是挺安全的了，不过还不够！建议你 (1)将 root 的登入权限取消； (2)将 ssh 版本设定为 2 。其他的设定值就请你依照自己的喜好来设定了。 通常不建议进行随便修改啦！另外，如果你修改过上面这个档案(/etc/ssh/sshd\_config)，那么就必需要重新启动一次 sshd 这个 daemon 才行！亦即是：

/etc/init.d/sshd restart