Welcome to another 'single-topic special'

In response to reader requests, we are assembling the content of some of our serialised articles into dedicated editions.

For now, this is a straight reprint of the series 'The Perfect Server' from issues 31 through 34; nothing fancy, just the facts.

Please bear in mind the original publication date; current versions of hardware and software may differ from those illustrated, so check your hardware and software versions before attempting to emulate the tutorials in these special editions. You may have later versions of software installed or available in your distributions' repositories.

Enjoy!

Please note: this Special Edition is provided with absolutely no warranty whatsoever; neither the contributors nor Full Circle Magazine accept any responsibility or liability for loss or damage resulting from readers choosing to apply this content to theirs or others' computers and equipment.

The articles contained in this magazine are released under the Creative Commons Attribution-Share Alike 3.0 Unported license. This means you can adapt, copy, distribute and transmit the articles but only under the following conditions: You must attribute the work to the original author in some way (at least a name, email or URL) and to this magazine by name ('full circle magazine') and the URL www.fullcirclemagazine.org (but not attribute the article(s) in any way that suggests that they endorse you or your use of the work). If you alter, transform, or build upon this work, you must distribute the resulting work under the same, similar or a compatible license.

Full Circle Magazine is entirely independent of Canonical, the sponsor of Ubuntu projects and the views and opinions in the magazine should in no way be assumed to have Canonical endorsement.
This tutorial shows how to prepare an Ubuntu 9.10 (Karmic Koala) server for ISPConfig 3, and how to install ISPConfig 3 on it. ISPConfig 3 is a webhosting control panel that allows you to configure the following services through a web browser: Apache web server, Postfix mail server, MySQL, MyDNS name server, PureFTPd, SpamAssassin, ClamAV, and many more.

Please note that this setup does not work for ISPConfig 2. It is valid for ISPConfig 3 only!

Requirements

To install such a system you will need the Ubuntu 9.10 server CD, available here:
http://releases.ubuntu.com/releases/9.10/ubuntu-9.10-server-i386.iso (32-bit) or:
http://releases.ubuntu.com/releases/9.10/ubuntu-9.10-server-amd64.iso (64-bit)

Preliminary Note

In this tutorial, I use the host name server1.example.com, with IP address 192.168.0.100 and gateway 192.168.0.1. These settings might differ for you, so you have to replace them where appropriate.

The Base System

Insert your Ubuntu install CD into your system and boot from it. Select your language then select Install Ubuntu Server:

Now you have to partition your hard disk. For simplicity's sake, I select Guided, use entire disk and set up LVM. This will create one volume group with two logical volumes—one for the / file system, and another one for swap. Of course, the partitioning is totally up to you—if you know what you're doing, you can also set up your partitions manually. You may find it helpful in future months if you set up separate /home and /var partitions.
Select the disk that you want to partition, and, when you're asked 'Write the changes to disk and configure LVM?', select Yes.

If you have selected Guided, use entire disk and set up LVM, the partitioner will create one big volume group that uses all the disk space. You can now specify how much of that disk space should be used by the logical volumes for / and swap. It makes sense to leave some space unused, so later on you can expand your existing logical volumes, or create new ones. This gives you more flexibility.

Write the changes to disk:

Then the base system is installed:

I don't need an encrypted private directory, so I choose No here:

Next, the package manager apt gets configured. Leave the HTTP proxy line empty unless you're using a proxy server to connect to the Internet:

I'm a little bit old-fashioned, and I like to update my servers manually to have more control, therefore I select No automatic updates. Of course, it's up to you what you select there.

We need DNS, mail, and LAMP servers, but, nevertheless, I don't select any of them now because I like to have full control over what gets installed on my system. We will install the needed packages manually later on. The only item I select here is OpenSSH server, so that I can immediately connect to the system with an SSH client such as PuTTY after the installation has finished:

The installation continues, then the GRUB boot loader gets installed.

The base system installation is now finished. Remove the installation CD from the CD drive and select Continue to reboot the system:

Next month, we use our administrator account to install SSH Server and vim-nox, and also configure the network itself.
tutorial with root privileges, we can either prepend all commands in this tutorial with the string sudo, or we become root right now by typing:

```
sudo su
```

You can also enable the root login by running:

```
sudo passwd root
```

and giving root a password. You can then directly log in as root, but this is frowned upon by the Ubuntu developers and community for various reasons. (See http://ubuntuforums.org/showthread.php?t=765414)

### Install The SSH Server (Optional)

If you did not install the OpenSSH server during the system installation, you can do it now:

```
aptitude install ssh openssh-server
```

From now on, you can use an SSH client such as PuTTY and connect from your workstation to your Ubuntu 9.10 server and follow the remaining steps in this tutorial.

### Install vim-nox (Optional)

I'll use vi as my text editor in this tutorial. The default vi program has some strange behaviour on Ubuntu and Debian; to fix this, we install vim-nox:

```
aptitude install vim-nox
```

You don't have to do this if you use a different text editor such as joe or nano.

### Configure The Network

Because the Ubuntu installer has configured our system to get its network settings via DHCP, we have to change that now because a server should have a static IP address. Edit /etc/network/interfaces and adjust it to your needs (in this example setup I will use the IP address 192.168.0.100):

```
vi /etc/network/interfaces
```

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

```
# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.0.100
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
```

Restart your network with:

```
/etc/init.d/networking restart
```

Then edit /etc/hosts:

```
vi /etc/hosts
```

Last month, we did the basic Ubuntu Server installation from CD, and got to the point of rebooting into the installed system.

### Get Root Privileges

After the reboot you can login with your previously created username (e.g. administrator). Because we must run all the steps from this
and make it look like the text shown in Fig.1.

Now run

echo server1.example.com > /etc/hostname

and reboot the server with:

reboot

 Afterwards, run:

hostname
hostname -f

Both should show server1.example.com now.

Edit sources.list And Update Your Linux Installation

Edit /etc/apt/sources.list:

vi /etc/apt/sources.list

Comment out or remove the installation CD from the file, and make sure that the universe and multiverse repositories are enabled.

Then run

aptitude update

to update the apt package database, and

aptitude safe-upgrade

to install the latest updates (if there are any). If you see that a new kernel gets installed as part of the updates, you should reboot the system afterwards with:

reboot

Change The Default Shell

/bin/sh is a symlink to /bin/dash, however we need /bin/bash, not /bin/dash.
Therefore we do this:

dpkg-reconfigure dash

Install dash as /bin/sh?, Choose: No

If you don't do this, the ISPConfig installation will fail.

Disable AppArmor

AppArmor is a security extension (similar to SELinux) that should provide extended security. In my opinion, you don't need it to configure a secure system, and it usually causes more problems than it has advantages (think of this - after you have done a week of trouble-shooting because some service wasn't working as expected, and then you find out that everything was OK, only AppArmor was causing the problem). Therefore, I disable it (this is a must if you want to install ISPConfig later on).

We can disable it like this:

/etc/init.d/apparmor stop
update-rc.d -f apparmor remove
aptitude remove apparmor apparmor-utils

Synchronize the System Clock

It is a good idea to synchronize the system clock with an NTP (network time protocol) server over the Internet. Simply run

aptitude install ntp ntpdate

and your system time will always be in sync.

Next month, we will install Postfix, SpamAssassin, Webalizer and much, much, more!
We can install Postfix, Courier, Saslauthd, MySQL, rkhunter, and binutils - with a single command:

```
( Prefix each command with sudo, if appropriate).
```

```
aptitude install postfix
postfix-mysql postfix-doc
mysql-client mysql-server
courier-authdaemon courier-authlib-mysql courier-pop
courier-pop-ssl courier-imap
```

You will be asked the following questions:

- New password for the MySQL "root" user
- Repeat password for the MySQL "root" user
- Create directories for web-based administration? Enter: No
- General type of mail configuration: Enter: Internet Site
- System mail name: Enter: server1.example.com (but using your .com)
- SSL certificate required Enter: OK

Next we install maildrop as follows:

```
update-alternatives --remove-all maildir.5
update-alternatives --remove-all maildirquota.7
aptitude install maildrop
```

You will ask yourself why we didn’t install maildrop together with all the other packages. The reason for this is a bug in the courier-base package - if you install maildrop together with courier-pop, courier-pop-ssl, courier-imap, and courier-imap-ssl, you will get the following error:

```
update-alternatives: error: alternative link /usr/share/man/man5/maildir.5.gz is already managed by maildir.5.gz.
```

We want MySQL to listen on all interfaces, not just localhost. Therefore we edit /etc/mysql/my.cnf and comment out the line bind-address = 127.0.0.1:

```
vi /etc/mysql/my.cnf
```

```
# Instead of skip-networking
the default is now to listen only on
# localhost which is more compatible and is not less secure.

#bind-address = 127.0.0.1
[...]
```

Then we restart MySQL:

```
/etc/init.d/mysql restart
```

Now check that networking is enabled. Run:

```
netstat -tap | grep mysql
```

The output should look like this:

```
root@server1:~# netstat -tap | grep mysql
```

```
tcp 0 0 *:mysql *:* LISTEN 6267/mysqld
root@server1:~#
```

During the installation, the SSL certificates for IMAP-SSL and POP3-SSL are created with the hostname localhost. To
THE PERFECT SERVER - PART 3

and Courier-POP3-SSL:

```
/etc/init.d/courier-imap-ssl restart
/etc/init.d/courier-pop-ssl restart
```

Install Amavisd-new, SpamAssassin, And Clamav

To install amavisd-new, SpamAssassin, and ClamAV, we run:

```
aptitude install amavisd-new spamassassin clamav clamav-daemon zoo unzip bzip2 arj nomarch lzip cabextract apt-listchanges libnet-ldap-perl libauthen-sasl-perl clamav-docs daemon libio-string-perl libio-socket-ssl-perl libnet-ident-perl zip libnet-dns-perl
```

Install Apache2, PHP5, phpMyAdmin, FCGI, suExec, Pear, And mcrypt

Apache2, PHP5, phpMyAdmin, FCGI, suExec, Pear, and mcrypt can be installed as follows:

```
aptitude install apache2 apache2.2-common apache2-doc apache2-mpm-prefork apache2-utils libexpat1 ssl-cert libapache2-mod-php5 php5 php5-common php5-gd php5-mysql php5-imap phpmyadmin php5-cli php5-cgi libapache2-mod-fcgid apache2-suexec php-pear php-auth php5-mcrypt mcrypt php5-imagick imagemagick libapache2-mod-suphp
```

You will see the following question:

Web server to reconfigure automatically:
Enter: **apache2**

Configure database for phpmyadmin with dbconfig-common?
Enter: **No**

Then run the following command to enable the Apache modules suexec, rewrite, ssl, actions, and include:

```
a2enmod suexec rewrite ssl actions include
```

Then restart Apache afterwards:

```
/etc/init.d/apache2 restart
```

Install PureFTPd And Quota

PureFTPd and quota can be installed with the following command:

```
aptitude install pure-ftpd-common pure-ftpd-mysql quota quotatool
```

Edit the file `/etc/default/pure-ftpd-common`:

```
vi /etc/default/pure-ftpd-common
```

and make sure that the start mode is set to standalone and set VIRTUALCHROOT=true:

```
[...]
VIRTUALCHROOT=true
[...]
```

Then restart PureFTPd:

```
/etc/init.d/pure-ftpd-mysql restart
```

Install Apache2, PHP5, phpMyAdmin, FCGI, suExec, Pear, And mcrypt

Apache2, PHP5, phpMyAdmin, FCGI, suExec, Pear, and mcrypt can be installed as follows:

```
aptitude install apache2 apache2.2-common apache2-doc apache2-mpm-prefork apache2-utils libexpat1 ssl-cert libapache2-mod-php5 php5 php5-common php5-gd php5-mysql php5-imap phpmyadmin php5-cli php5-cgi libapache2-mod-fcgid apache2-suexec php-pear php-auth php5-mcrypt mcrypt php5-imagick imagemagick libapache2-mod-suphp
```

Then recreate the certificates:

```
mkimapdcert
mkpop3dcert
```

Then restart Courier-IMAP-SSL:

```
/etc/init.d/courier-imap-ssl restart
```

and Courier-POP3-SSL:

```
/etc/init.d/courier-pop-ssl restart
```

```
vi /etc/courier/imapd.cnf
[...]
CN=server1.example.com
[...]
vi /etc/courier/pop3d.cnf
[...]
CN=server1.example.com
[...]```
To enable quota, run these commands:

```
touch /aquota.user
/touch /aquota.group
chmod 600 /aquota.*
mount -o remount /
quotacheck -avugm
quotaon -avug
```

### Install MyDNS

Before we install MyDNS, we need to install a few prerequisites:

```
aptitude install g++ libc6
gcc gawk make texinfo
libmysqlclient15-dev
```

MyDNS is not available in the Ubuntu 9.10 repositories, therefore we have to build it ourselves as follows:

```
cd /tmp
wget http://heanet.dl.sourceforge.net/sourceforge/mydns-ng/mydns-1.2.8.27.tar.gz
tar xvfz mydns-1.2.8.27.tar.gz
cd mydns-1.2.8
./configure
make
make install
```

Then we make the script executable, and create the system startup links for it:

```
chmod +x /etc/init.d/mydns
update-rc.d mydns defaults
```

### Install Vlogger and Webalizer

Vlogger and webalizer can be installed as follows:

```
aptitude install vlogger
webalizer
```

### Install Jailkit

Jailkit is needed only if you want to chroot SSH users. It can be installed as follows (important: Jailkit must be installed before ISPConfig - it cannot be installed afterwards!):

```
aptitude install build-essential autoconf
automake1.9 libtool flex
bison
cd /tmp
wget http://olivier.sessink.nl/jailkit/jailkit-2.10.tar.gz
tar xvfz jailkit-2.10.tar.gz
```

---

# /etc/fstab: static file system information.
#
# Use `blkid --value -s UUID` to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See `fstab(5)`.
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
/dev/mapper/server1-root / ext4 errors=remount-ro,usrjquota=aquota.user,grpjquota=aquota.group,jqfmt=vfsv0 0 1
# /boot was on /dev/sda5 during installation
UUID=9ea34148-31b7-4d5c-baee-c2e2022562ea /boot ext2 defaults 0 2
/dev/mapper/server1-swap_1 none swap sw 0 0
/dev/scd0 /media/cdrom0 udf,iso9660 user,noauto,exec(utf8 0 0
/dev/fd0 /media/floppy0 auto rw,user,noauto,exec(utf8 0 0
```

---

**Fig. 1**
#!/bin/sh
#
# mydns Start the MyDNS server
#
# Author: Philipp Kern <phil@phil kern.de>. Based upon skeleton 1.9.4 by Miquel van
# Smoorenburg <miquels@cistron.nl> and Ian Murdock <imurdock@gnu.ai.mit.edu>.
#
set -e

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:
/usr/bin
DAEMON=/usr/local/sbin/mydns
NAME=mydns
DESC="DNS server"
SCRIPTNAME=/etc/init.d/$NAME

# Gracefully exit if the package has been removed.
test -x $DAEMON || exit 0

case "$1" in
  start)
    echo -n "Starting $DESC: $NAME"
    start-stop-daemon --start --quiet --oknodo \
                 --exec $DAEMON
    sleep 1
    start-stop-daemon --start --quiet \
                 --exec $DAEMON -- -b
    echo "."
    ;;
  stop)
    echo -n "Stopping $DESC: $NAME"
    start-stop-daemon --stop --oknodo --quiet \
                      --exec $DAEMON
    echo "."
    ;;
  reload|force-reload)
    echo -n "Reloading $DESC configuration..."
    start-stop-daemon --stop --signal HUP --quiet \
                      --exec $DAEMON
    echo "done."
    ;;
  *)
    echo "Usage: $SCRIPTNAME
{start|stop|restart|reload|force-reload}" >&2
    exit 1
    esac
    exit 0

cd jailkit-2.10
./configure
make
make install
cd ..
rm -rf jailkit-2.10*

Install fail2ban

This is optional but recommended, because the ISPConfig monitor tries to show the fail2ban log:

aptitude install fail2ban

Next month, in the final installment, we will install SquirrelMail and ISPConfig3, giving you the perfect server, ready to go!
To install the SquirrelMail webmail client, run:

```
aptitude install squirrelmail
```

Then, create the following symlink...

```
ln -s /usr/share/squirrelmail/ /var/www/webmail
```

... and configure SquirrelMail:

```
squirrelmail-configure
```

We must tell SquirrelMail that we are using Courier-IMAP/POP3:

```
SquirrelMail Configuration:
Read: config.php (1.4.0)
Main Menu
1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers
C. Turn color on
S. Save data
Q. Quit

Command >>
```
Enter: D
```

Now, you will see a list of IMAP server options entitled:

```
Please select your IMAP server:
```

Enter the word: courier

```
imap_server_type = courier
default_folder_prefix = INBOX.
trash_folder = Trash
sent_folder = Sent
draft_folder = Drafts
show_prefix_option = false
default_sub_of_inbox = false
show_contain_subfolders_option = false
optional_delimiter = .
delete_folder = true
```

Press any key to continue...

Next, you will see a list of options and their settings; press the Enter key to continue.

Back at the Main Menu, enter: to save data, and you will see:

```
Data saved in config.php
Press enter to continue
```

Back at the Main Menu, enter Q to quit.

Afterwards you can access SquirrelMail under:

```
http://server1.example.com/webmail
```

or:

```
http://192.168.0.100/webmail
```

Install ISPConfig 3

To install ISPConfig 3 from the latest released version, do this (replacing ISPConfig-3.0.1.6.tar.gz with the latest version):

```
cd /tmp
wget http://downloads.sourceforge.net/ispconfig/ISPConfig-3.0.1.6.tar.gz?use_mirror=
tar xvfz ISPConfig-3.0.1.6.tar.gz
```
The next step is to run:

```bash
cd ispconfig3_install/install/
php -q install.php
```

This will start the ISPConfig 3 installer. Press **Enter** for each option - except when asked for your MySQL root password.

The installer automatically configures all underlying services, so no manual configuration is needed.

Afterwards you can access ISPConfig 3 under:

```
http://server1.example.com:80 80/
```

or:

```
http://192.168.0.100:8080/
```

Log in with the username **admin** and the password **admin** (you should change the default password after your first login):

The system is now ready to be used.