The Perfect Server - Debian 10 (Buster) with Apache, BIND, Dovecot, PureFTPD and ISPConfig 3.2

This tutorial shows how to prepare a Debian 10 server (with Apache2, BIND, Dovecot) for the installation of ISPConfig 3.2, and how to install ISPConfig. The web hosting control panel ISPConfig 3 allows you to configure the following services through a web browser: Apache or nginx web server, Postfix mail server, Courier or Dovecot IMAP/POP3 server, MySQL, BIND or MyDNS nameserver, PureFTPd, SpamAssassin, ClamAV, and many more. This setup covers Apache (instead of nginx), BIND, and Dovecot.

On this page

1 Preliminary Note

In this tutorial, I will use the hostname *server1.example.com* with the IP address *192.168.0.100* and the gateway *192.168.0.1*. These settings might differ for you, so you have to replace them where appropriate. Before proceeding further you need to have a minimal installation of Debian 10. This might be a Debian minimal image from your Hosting provider or you use the Minimal Debian Server tutorial to set up the base system.

All commands below are run as root user. Either log in as root user directly or log in as your normal user and then use the command

```
su -
```

to become root user on your server before you proceed. **IMPORTANT**: You must use 'su -' and not just 'su', otherwise your PATH variable is set wrong by Debian.

2 Install the SSH server (Optional)

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

```
apt-get install ssh openssh-server
```

From now on you can use an SSH client such as PuTTY and connect from your workstation to your Debian 9 server and follow the remaining steps from this tutorial.

3 Install a shell text editor (Optional)

We will use *nano* text editor in this tutorial. Some users prefer the classic vi editor, therefore we will install both editors here. The default *vi* program has some strange behavior on Debian and Ubuntu; to fix this, we install *vim-nox*:

```
apt-get install nano vim-nox
```

If vi is your favorite editor, then replace nano with vi in the following commands to edit files.

4 Configure the Hostname

The hostname of your server should be a subdomain like "server1.example.com". Do not use a domain name without subdomain part like "example.com" as hostname as this will cause problems later with your mail setup. First, you should check the hostname in */etc/hosts* and change it when necessary. The line should be: "IP Address - space - full hostname incl. domain - space - subdomain part". For our hostname server1.example.com, the file shall look like this:

```
nano /etc/hosts
```

```
127.0.0.1       localhost.localdomain   localhost
192.168.0.100   server1.example.com     server1

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Then edit the /etc/hostname file:

```
nano /etc/hostname
```

It shall contain only the subdomain part, in our case:

```
server1
```

Finally, reboot the server to apply the change:

```
systemctl reboot
```

Log in again and check if the hostname is correct now with these commands:

```
hostname
hostname -f
```

The output shall be like this:

```
root@server1:/tmp# hostname
server1
root@server1:/tmp# hostname -f
server1.example.com
```

5 Update your Debian Installation

First, make sure that your `/etc/apt/sources.list` contains the `buster/updates` repository (this makes sure you always get the newest security updates), and that the `contrib` and `non-free` repositories are enabled as some required packages are not in the main repository.

```
nano /etc/apt/sources.list
```

```
deb http://deb.debian.org/debian/ buster main contrib non-free
deb-src http://deb.debian.org/debian/ buster main contrib non-free

deb http://security.debian.org/debian-security buster/updates main co
ntrib non-free
deb-src http://security.debian.org/debian-security buster/updates mai
n contrib non-free
```

Run:

```
apt-get update
```

To update the apt package database

```
apt-get upgrade
```

and to install the latest updates (if there are any).

6. 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
```

Use dash as the default system shell (/bin/sh)? **<-- No**

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

7 Synchronize the System Clock

It is a good idea to synchronize the system clock with an NTP (**n**etwork **t**ime **p**rotocol) server over the Internet. Simply run

```
apt-get -y install ntp
```

and your system time will always be in sync.

8 Install Postfix, Dovecot, MariaDB, rkhunter, and Binutils

We can install Postfix, Dovecot, MariaDB as MySQL alternative, rkhunter, and Binutils with a single command:

```
apt-get -y install postfix postfix-mysql postfix-doc mariadb-client
mariadb-server openssl getmail4 rkhunter binutils dovecot-imapd dov
ecot-pop3d dovecot-mysql dovecot-sieve dovecot-lmtpd sudo curl
```

You will be asked the following questions:

```
General type of mail configuration: <-- Internet Site
System mail name: <-- server1.example.com
```

To secure the MariaDB installation and to disable the test database, run this command:

```
mysql_secure_installation
```

Answer the questions as follows:

```
Change the root password? [Y/n] <-- y
New password: <-- Enter a new MariaDB root password
Re-enter new password: <-- Repeat the MariaDB root password
Remove anonymous users? [Y/n] <-- y
Disallow root login remotely? [Y/n] <-- y
Remove test database and access to it? [Y/n] <-- y
Reload privilege tables now? [Y/n] <-- y
```

Next, open the TLS/SSL and submission ports in Postfix:

```
nano /etc/postfix/master.cf
```

Uncomment the *submission* and *smtps* sections as follows and add lines where necessary so that this section of the master.cf file looks exactly like the one below. **IMPORTANT:** Remove the # in front of the lines that start with smtps and submission too and not just from the -o lines after these lines!

```
[...]
submission inet n - - - - smtpd
 -o syslog_name=postfix/submission
 -o smtpd_tls_security_level=encrypt
 -o smtpd_sasl_auth_enable=yes
 -o smtpd_client_restrictions=permit_sasl_authenticated,reject
# -o smtpd_reject_unlisted_recipient=no
# -o smtpd_client_restrictions=$mua_client_restrictions
# -o smtpd_helo_restrictions=$mua_helo_restrictions
# -o smtpd_sender_restrictions=$mua_sender_restrictions
# -o smtpd_recipient_restrictions=
# -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
# -o milter_macro_daemon_name=ORIGINATING
smtps inet n - - - - smtpd
 -o syslog_name=postfix/smtps
 -o smtpd_tls_wrappermode=yes
 -o smtpd_sasl_auth_enable=yes
 -o smtpd_client_restrictions=permit_sasl_authenticated,reject
# -o smtpd_reject_unlisted_recipient=no
# -o smtpd_client_restrictions=$mua_client_restrictions
# -o smtpd_helo_restrictions=$mua_helo_restrictions
# -o smtpd_sender_restrictions=$mua_sender_restrictions
# -o smtpd_recipient_restrictions=
# -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
# -o milter_macro_daemon_name=ORIGINATING
[...]
```

Restart Postfix afterwards:

```
systemctl restart postfix
```

We want MySQL to listen on all interfaces, not just localhost. Therefore, we edit `/etc/mysql/mariadb.conf.d/50-server.cnf` and comment out the line `bind-address = 127.0.0.1` by adding a # in front of it.

```
nano /etc/mysql/mariadb.conf.d/50-server.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

[...]
```

Set the password authentication method in MariaDB to native so we can use PHPMyAdmin later to connect as root user:

```
echo "update mysql.user set plugin = 'mysql_native_password' where
user='root';" | mysql -u root
```

Edit the file /etc/mysql/debian.cnf and set the MYSQL / MariaDB root password there twice in the rows that start with the word password.

```
nano /etc/mysql/debian.cnf
```

The MySQL root password that needs to be added is shown in red. In this example, the password is "howtoforge".

```
# Automatically generated for Debian scripts. DO NOT TOUCH!
[client]
host = localhost
user = root
password = howtoforge
socket = /var/run/mysqld/mysqld.sock
[mysql_upgrade]
host = localhost
```

```
user = root
password = howtoforge
socket = /var/run/mysqld/mysqld.sock
basedir = /usr
```

To prevent the error '**Error in accept: Too many open files**' we will set higher open file limits for MariaDB now.

Open the file /etc/security/limits.conf with an editor:

```
nano /etc/security/limits.conf
```

and add these lines at the end of the file.

```
mysql soft nofile 65535
mysql hard nofile 65535
```

Next, create a new directory /etc/systemd/system/mysql.service.d/ with the mkdir command.

```
mkdir -p /etc/systemd/system/mysql.service.d/
```

and add a new file inside:

```
nano /etc/systemd/system/mysql.service.d/limits.conf
```

paste the following lines into that file:

```
[Service]
LimitNOFILE=infinity
```

Save the file and close the nano editor.

Then we reload systemd and restart MariaDB:

```
systemctl daemon-reload
systemctl restart mariadb
```

Now check that networking is enabled. Run

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

The output should look like this:

```
root@server1:/home/administrator# netstat -tap | grep mysql
```

```
tcp6 0 0 [::]:mysql [::]:* LISTEN 16623/mysqld
```

9 Install Amavisd-new, SpamAssassin, and ClamAV

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

```
apt-get install amavisd-new spamassassin clamav clamav-daemon unzip
bzip2 arj nomarch lzop cabextract p7zip p7zip-full unrar lrzip apt-
listchanges libnet-ldap-perl libauthen-sasl-perl clamav-docs daemon
libio-string-perl libio-socket-ssl-perl libnet-ident-perl zip libne
t-dns-perl libdbd-mysql-perl postgrey
```

The ISPConfig 3 setup uses amavisd which loads the SpamAssassin filter library internally, so we can stop SpamAssassin to free up some RAM:

```
systemctl stop spamassassin
systemctl disable spamassassin
```

10 Install Apache Web Server and PHP

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

```
apt-get -y install apache2 apache2-doc apache2-utils libapache2-mod
-php php7.3 php7.3-common php7.3-gd php7.3-mysql php7.3-imap php7.3
-cli php7.3-cgi libapache2-mod-fcgid apache2-suexec-pristine php-pe
ar mcrypt  imagemagick libruby libapache2-mod-python php7.3-curl ph
p7.3-intl php7.3-pspell php7.3-recode php7.3-sqlite3 php7.3-tidy ph
p7.3-xmlrpc php7.3-xsl memcached php-memcache php-imagick php-gette
xt php7.3-zip php7.3-mbstring memcached libapache2-mod-passenger ph
p7.3-soap php7.3-fpm php7.3-opcache php-apcu libapache2-reload-perl
```

Then run the following command to enable the Apache modules *suexec*, *rewrite*, *ssl*, *actions*, and *include* (plus *dav*, *dav_fs*, and *auth_digest* if you want to use WebDAV):

```
a2enmod suexec rewrite ssl actions include dav_fs dav auth_digest c
gi headers actions proxy_fcgi alias
```

To ensure that the server cannot be attacked through the HTTPOXY vulnerability, we will disable the HTTP_PROXY header in apache globally by adding the configuration file /etc/apache2/conf-available/httpoxy.conf.

Note: The vulnerability is named httpoxy (without 'r') and therefore the file where we add the config to prevent it is named httpoxy.conf and not httproxy.conf, so there is no 'r' missing in the filename.

```
nano /etc/apache2/conf-available/httpoxy.conf
```

Paste the following content to the file:

```
<IfModule mod_headers.c>
    RequestHeader unset Proxy early
</IfModule>
```

And enable the module by running:

```
a2enconf httpoxy
systemctl restart apache2
```

11 Install Let's Encrypt

ISPConfig is using acme.sh now as Let's Encrypt client. Install acme.sh using the following command:

```
curl https://get.acme.sh | sh -s
```

12 Install Mailman

ISPConfig allows you to manage (create/modify/delete) Mailman mailing lists. If you want to make use of this feature, install Mailman as follows:

```
apt-get install mailman
```

Select at least one language, e.g.:

```
Languages to support: <-- en (English)
Missing site list <-- Ok
```

Before we can start Mailman, a first mailing list called `mailman` must be created:

```
newlist mailman
```

```
root@server1:~# newlist mailman
Enter the email of the person running the list: <-- admin email address, e.g. listadmin@example.com
Initial mailman password: <-- admin password for the mailman list
To finish creating your mailing list, you must edit your /etc/aliases (or
equivalent) file by adding the following lines, and possibly running the
```

```
`newaliases' program:

## mailman mailing list
mailman:              "|/var/lib/mailman/mail/mailman post mailman"
mailman-admin:        "|/var/lib/mailman/mail/mailman admin mailman"
mailman-bounces:      "|/var/lib/mailman/mail/mailman bounces mailma
n"
mailman-confirm:      "|/var/lib/mailman/mail/mailman confirm mailma
n"
mailman-join:         "|/var/lib/mailman/mail/mailman join mailman"
mailman-leave:        "|/var/lib/mailman/mail/mailman leave mailman"
mailman-owner:        "|/var/lib/mailman/mail/mailman owner mailman"
mailman-request:      "|/var/lib/mailman/mail/mailman request mailma
n"
mailman-subscribe:    "|/var/lib/mailman/mail/mailman subscribe mailm
an"
mailman-unsubscribe:  "|/var/lib/mailman/mail/mailman unsubscribe mai
lman"

Hit enter to notify mailman owner... <-- ENTER

root@server1:~#
```

Open `/etc/aliases` afterwards...

```
nano /etc/aliases
```

... and add the following lines:

```
[...]
## mailman mailing list
mailman:              "|/var/lib/mailman/mail/mailman post mailman"
mailman-admin:        "|/var/lib/mailman/mail/mailman admin mailman"
mailman-bounces:      "|/var/lib/mailman/mail/mailman bounces mailma
n"
mailman-confirm:      "|/var/lib/mailman/mail/mailman confirm mailma
n"
mailman-join:         "|/var/lib/mailman/mail/mailman join mailman"
mailman-leave:        "|/var/lib/mailman/mail/mailman leave mailman"
mailman-owner:        "|/var/lib/mailman/mail/mailman owner mailman"
mailman-request:      "|/var/lib/mailman/mail/mailman request mailma
n"
mailman-subscribe:    "|/var/lib/mailman/mail/mailman subscribe mailm
an"
mailman-unsubscribe:  "|/var/lib/mailman/mail/mailman unsubscribe mai
lman"
```

Run:

```
newaliases
```

and restart Postfix:

```
systemctl restart postfix
```

Finally, we must enable the Mailman Apache configuration:

```
ln -s /etc/mailman/apache.conf /etc/apache2/conf-enabled/mailman.conf
```

This defines the alias `/cgi-bin/mailman/` for all Apache vhosts, which means you can access the Mailman admin interface for a list at `http://server1.example.com/cgi-bin/mailman/admin/`, and the web page for users of a mailing list can be found at `http://server1.example.com/cgi-bin/mailman/listinfo/`.

Under `http://server1.example.com/pipermail` you can find the mailing list archives.

Restart Apache afterwards:

```
systemctl restart apache2
```

Then start the Mailman daemon:

```
systemctl restart mailman
```

13 Install PureFTPd and Quota

PureFTPd and quota can be installed with the following command:

```
apt-get install pure-ftpd-common pure-ftpd-mysql quota quotatool
```

Create the dhparam file for pure-ftpd:

```
openssl dhparam -out /etc/ssl/private/pure-ftpd-dhparams.pem 2048
```

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

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

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

```
[...]
```

```
STANDALONE_OR_INETD=standalone
[...]
VIRTUALCHROOT=true
[...]
```

Now we configure PureFTPd to allow FTP and TLS sessions. FTP is a very insecure protocol because all passwords and all data are transferred in clear text. By using TLS, the whole communication can be encrypted, thus making FTP much more secure.

If you want to allow FTP and TLS sessions, run

```
echo 1 > /etc/pure-ftpd/conf/TLS
```

In order to use TLS, we must create an SSL certificate. I create it in `/etc/ssl/private/`, therefore I create that directory first:

```
mkdir -p /etc/ssl/private/
```

Afterwards, we can generate the SSL certificate as follows:

```
openssl req -x509 -nodes -days 7300 -newkey rsa:2048 -keyout /etc/ssl/private/pure-ftpd.pem -out /etc/ssl/private/pure-ftpd.pem
```

```
Country Name (2 letter code) [AU]: <-- Enter your Country Name (e.g., "DE").
State or Province Name (full name) [Some-State]: <-- Enter your State or Province Name.
Locality Name (eg, city) []: <-- Enter your City.
Organization Name (eg, company) [Internet Widgits Pty Ltd]: <-- Enter your Organization Name (e.g., the name of your company).
Organizational Unit Name (eg, section) []: <-- Enter your Organizational Unit Name (e.g. "IT Department").
Common Name (eg, YOUR name) []: <-- Enter the Fully Qualified Domain Name of the system (e.g. "server1.example.com").
Email Address []: <-- Enter your Email Address.
```

Change the permissions of the SSL certificate:

```
chmod 600 /etc/ssl/private/pure-ftpd.pem
```

Then restart PureFTPd:

```
systemctl restart pure-ftpd-mysql
```

Edit `/etc/fstab`. Mine looks like this (I added `,usrjquota=quota.user,grpjquota=quota.group,jqfmt=vfsv0` to the partition with the mount point `/`):

```
nano /etc/fstab
```

```
# /etc/fstab: static file system information.
#
# Use 'blkid' 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>
# / was on /dev/sda1 during installation
UUID=45576b38-39e8-4994-b8c1-ea4870e2e614 / ext4 errors=remount-ro,usrjquota=quota.user,grpjquota=quota.group,jqfmt=vfsvo 0 1
# swap was on /dev/sda5 during installation
UUID=8bea0d1e-ec37-4b20-9976-4b7daaa3eb69 none swap sw 0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
```

To enable quota, run these commands:

```
mount -o remount /
```

```
quotacheck -avugm
quotaon -avug
```

14 Install BIND DNS Server

BIND can be installed as follows:

```
apt-get install bind9 dnsutils
```

If your server is a virtual machine, then it is highly recommended to install the haveged daemon to get a higher entropy for DNSSEC signing. You can install haveged on nonvirtual servers as well, it should not hurt.

```
apt-get install haveged
```

An explanation on that topic can be found here.

15 Install Webalizer, AWStats and GoAccess

Webalizer and AWStats can be installed as follows:

```
apt-get install webalizer awstats geoip-database libclass-dbi-mysql-perl libtimedate-perl
```

Open `/etc/cron.d/awstats` afterwards...

```
nano /etc/cron.d/awstats
```

... and comment out everything in that file:

```
#MAILTO=root

#*/10 * * * * www-data [ -x /usr/share/awstats/tools/update.sh ] && /usr/share/awstats/tools/update.sh

# Generate static reports:
#10 03 * * * www-data [ -x /usr/share/awstats/tools/buildstatic.sh ] && /usr/share/awstats/tools/buildstatic.sh
```

Installing the latest GoAccess version directly from the GoAccess repository:

```
echo "deb https://deb.goaccess.io/ $(lsb_release -cs) main" | sudo tee -a /etc/apt/sources.list.d/goaccess.list
wget -O - https://deb.goaccess.io/gnugpg.key | sudo apt-key --keyring /etc/apt/trusted.gpg.d/goaccess.gpg add -
apt-get update
apt-get install goaccess
```

16 Install Jailkit

Jailkit is needed only if you want to chroot SSH users. It can be installed as follows:

```
apt-get install build-essential autoconf automake libtool flex bison debhelper binutils
```

```
cd /tmp
wget http://olivier.sessink.nl/jailkit/jailkit-2.20.tar.gz
tar xvfz jailkit-2.20.tar.gz
cd jailkit-2.20
echo 5 > debian/compat
./debian/rules binary
```

You can now install the Jailkit `.deb` package as follows:

```
cd ..
dpkg -i jailkit_2.20-1_*.deb
rm -rf jailkit-2.20*
```

17 Install fail2ban and UFW Firewall

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

```
apt-get install fail2ban
```

To make fail2ban monitor PureFTPd and Dovecot, create the file */etc/fail2ban/jail.local*:

```
nano /etc/fail2ban/jail.local
```

And add the following configuration to it.

```
[pure-ftpd]
enabled = true
port = ftp
filter = pure-ftpd
logpath = /var/log/syslog
maxretry = 3

[dovecot]
enabled = true
filter = dovecot
logpath = /var/log/mail.log
maxretry = 5

[postfix-sasl]
enabled = true
port = smtp
filter = postfix[mode=auth]
logpath = /var/log/mail.log
maxretry = 3
```

Restart fail2ban afterwards:

```
systemctl restart fail2ban
```

To install the UFW firewall, run this apt command:

```
apt-get install ufw
```

18 Install PHPMyAdmin Database Administration Tool

Since Debian 10, PHPMyAdmin is not available as .deb package anymore. Therefore we will install it from source.

Create folders for PHPMyadmin:

```
mkdir /usr/share/phpmyadmin
mkdir /etc/phpmyadmin
mkdir -p /var/lib/phpmyadmin/tmp
chown -R www-data:www-data /var/lib/phpmyadmin
touch /etc/phpmyadmin/htpasswd.setup
```

Go to the /tmp directory and download the PHPMyAdmin sources:

```
cd /tmp
wget https://files.phpmyadmin.net/phpMyAdmin/4.9.0.1/phpMyAdmin-4.
9.0.1-all-languages.tar.gz
```

Unpack the downloaded archive file and move the files to the /usr/share/phpmyadmin folder and clean up the /tmp directory.

```
tar xfz phpMyAdmin-4.9.0.1-all-languages.tar.gz
mv phpMyAdmin-4.9.0.1-all-languages/* /usr/share/phpmyadmin/
rm phpMyAdmin-4.9.0.1-all-languages.tar.gz
rm -rf phpMyAdmin-4.9.0.1-all-languages
```

Create a new config file for PHPMyaAdmin based on the provided sample file:

```
cp /usr/share/phpmyadmin/config.sample.inc.php  /usr/share/phpmyadm
in/config.inc.php
```

Open the config file with nano editor:

```
nano /usr/share/phpmyadmin/config.inc.php
```

Set a secure password (blowfish secret) which must be 32 chars long:

```
$cfg['blowfish_secret'] = 'bD3e6wva9fnd93jVsb7SDgeiBCd452Dh'; /* YOU
MUST FILL IN THIS FOR COOKIE AUTH! */
```

Don't use my example blowfish secret, set your own one!

Then add a line to set the directory which PHPMyAdmin shall use to store temporary files:

```
$cfg['TempDir'] = '/var/lib/phpmyadmin/tmp';
```

Next, we create the Apache configuration file for PHPMyAdmin by opening a new file in nano editor:

```
nano /etc/apache2/conf-available/phpmyadmin.conf
```

Paste the following config into the file and save it.

```
# phpMyAdmin default Apache configuration

Alias /phpmyadmin /usr/share/phpmyadmin

<Directory /usr/share/phpmyadmin>
 Options FollowSymLinks
 DirectoryIndex index.php

 <IfModule mod_php7.c>
 AddType application/x-httpd-php .php

 php_flag magic_quotes_gpc Off
 php_flag track_vars On
 php_flag register_globals Off
 php_value include_path .
 </IfModule>

</Directory>

# Authorize for setup
<Directory /usr/share/phpmyadmin/setup>
 <IfModule mod_authn_file.c>
 AuthType Basic
 AuthName "phpMyAdmin Setup"
 AuthUserFile /etc/phpmyadmin/htpasswd.setup
 </IfModule>
 Require valid-user
</Directory>

# Disallow web access to directories that don't need it
<Directory /usr/share/phpmyadmin/libraries>
 Order Deny,Allow
 Deny from All
</Directory>
<Directory /usr/share/phpmyadmin/setup/lib>
 Order Deny,Allow
 Deny from All
</Directory>
```

Activate the configuration and restart Apache.

```
a2enconf phpmyadmin
systemctl restart apache2
```

In the next step, we will configure the phpMyadmin configuration store (database).

Log into MariaDB as root user:

```
mysql -u root -p
```

In the MariaDB shell, create a new database for PHPMyAdmin:

```
MariaDB [(none)]> CREATE DATABASE phpmyadmin;
```

Then create a new user:

```
MariaDB [(none)]> CREATE USER 'pma'@'localhost' IDENTIFIED BY 'mypass
word';
```

Replace the word mypassword with a secure password of your choice in the commands above and below, use the same password both times. Then grant the user access to this database and reload database permissions.

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON phpmyadmin.* TO 'pma'@'loca
lhost' IDENTIFIED BY 'mypassword' WITH GRANT OPTION;
MariaDB [(none)]> FLUSH PRIVILEGES;
MariaDB [(none)]> EXIT;
```

Finally, load the SQL tables into the database:

```
mysql -u root -p phpmyadmin < /usr/share/phpmyadmin/sql/create_tabl
es.sql
```

Enter the MariaDB root password on request.

All we have to do now is to set the phpmyadmin user details in the configuration file. Open the file in nano editor again:

```
nano /usr/share/phpmyadmin/config.inc.php
```

Scroll down until you see the lines below and edit them:

```
/* User used to manipulate with storage */
$cfg['Servers'][$i]['controlhost'] = 'localhost';
$cfg['Servers'][$i]['controlport'] = '';
$cfg['Servers'][$i]['controluser'] = 'pma';
$cfg['Servers'][$i]['controlpass'] = 'mypassword';

/* Storage database and tables */
$cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';
$cfg['Servers'][$i]['bookmarktable'] = 'pma__bookmark';
$cfg['Servers'][$i]['relation'] = 'pma__relation';
$cfg['Servers'][$i]['table_info'] = 'pma__table_info';
$cfg['Servers'][$i]['table_coords'] = 'pma__table_coords';
$cfg['Servers'][$i]['pdf_pages'] = 'pma__pdf_pages';
$cfg['Servers'][$i]['column_info'] = 'pma__column_info';
$cfg['Servers'][$i]['history'] = 'pma__history';
$cfg['Servers'][$i]['table_uiprefs'] = 'pma__table_uiprefs';
$cfg['Servers'][$i]['tracking'] = 'pma__tracking';
$cfg['Servers'][$i]['userconfig'] = 'pma__userconfig';
$cfg['Servers'][$i]['recent'] = 'pma__recent';
$cfg['Servers'][$i]['favorite'] = 'pma__favorite';
$cfg['Servers'][$i]['users'] = 'pma__users';
```

```
$cfg['Servers'][$i]['usergroups'] = 'pma__usergroups';
$cfg['Servers'][$i]['navigationhiding'] = 'pma__navigationhiding';
$cfg['Servers'][$i]['savedsearches'] = 'pma__savedsearches';
$cfg['Servers'][$i]['central_columns'] = 'pma__central_columns';
$cfg['Servers'][$i]['designer_settings'] = 'pma__designer_settings';
$cfg['Servers'][$i]['export_templates'] = 'pma__export_templates';
```

I've marked the lines in red which I've edited. Replace mypassword with the password that you've chosen for the phpmyadmin user. Note that the // in front of the lines have been removed as well!

19 Install RoundCube Webmail (optional)

In this chapter, we will install the RoundCube webmail client. First, we have to create the database for Roundcube manually as there is currently an issue in the RoundCube Debian installer which causes it to fail to create the database automatically. Run this command to create the database:

```
echo "CREATE DATABASE roundcube;" | mysql --defaults-file=/etc/mysql/debian.cnf
```

Then install RoundCube with this command:

```
apt-get install roundcube roundcube-core roundcube-mysql roundcube-plugins
```

The installer will ask the following questions:

```
Configure database for roundcube with dbconfig.common? <-- yes
MySQL application password for roundcube: <-- press enter
```

Then edit the RoundCube /etc/roundcube/config.inc.php file and adjust a few settings:

```
nano /etc/roundcube/config.inc.php
```

Set the default_host to localhost and the smtp_server.

```
$config['default_host'] = 'localhost';
$config['smtp_server'] = 'tls://%h';
$config['smtp_port']  = 587;
```

Then edit the Apache RoundCube configuration file /etc/apache2/conf-enabled/roundcube.conf:

```
nano /etc/apache2/conf-enabled/roundcube.conf
```

And add an alias line for the apache /webmail alias and one for /roundcube, you can add the line right at the beginning of the file. NOTE: Do not use /mail as alias or the ispconfig email module will stop working!

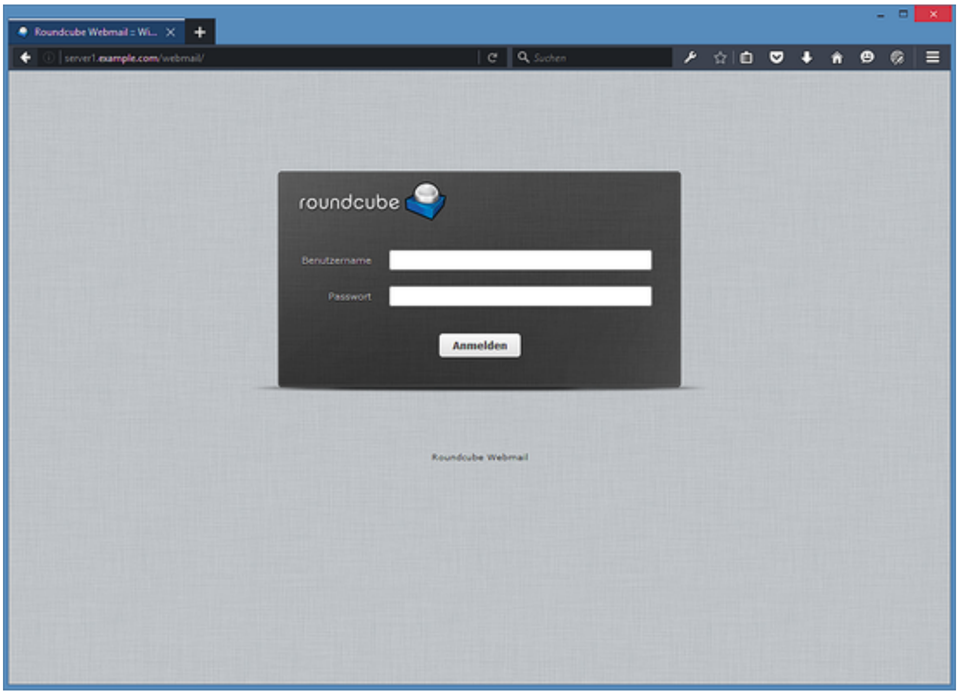
```
Alias /roundcube /var/lib/roundcube
Alias /webmail /var/lib/roundcube
```

Then reload Apache:

```
systemctl reload apache2
```

Now you can access RoundCube as follows:

http://192.168.0.100/webmail
http://www.example.com/webmail
http://server1.example.com:8080/webmail (after you have installed ISPConfig, see the next chapter)



There exist some plugins to integrate RoundCube Webmail with ISPConfig, have a look here for the ISPConfig RoundCube plugin installation instructions.

20 Download ISPConfig 3

20 Download the ISPConfig stable release (recommended)

To install ISPConfig 3 from the latest released version, do this:

```
cd /tmp
```

```
wget http://www.ispconfig.org/downloads/ISPConfig-3-stable.tar.gz
tar xfz ISPConfig-3-stable.tar.gz
cd ispconfig3_install/install/
```

21 Install ISPConfig

The next step is to run the ISPConfig installer.

```
php -q install.php
```

This will start the ISPConfig 3 installer. The installer will configure all services like Postfix, Dovecot, etc. for you. A manual setup as required for ISPConfig 2 (perfect setup guides) is not necessary.

```
# php -q install.php
```

```
-----------------------------------------------------------------------
-----------
_____ ___________ _____ __ _ ____
|_ _/ ___| ___ \ / __ \ / _(_) /__ \
| | \ `--.| |_/ / | / \/ ___ _ __ | |_ _ __ _ _/ /
| | `--. \ __/ | | / _ \| '_ \| _| |/ _` | |_ |
_| |_/\__/ / | | \__/\ (_) | | | | | | | (_| | ___\ \
\___/\____/\_| \____/\___/|_| |_|_| |_|\__, | \____/
__/ |
|___/
-----------------------------------------------------------------------
-----------
```

```
>> Initial configuration
```

```
Operating System: Debian 10.0 (Buster) or compatible
```

```
Following will be a few questions for primary configuration so be car
eful.
Default values are in [brackets] and can be accepted with <ENTER>.
Tap in "quit" (without the quotes) to stop the installer.
```

```
Select language (en,de) [en]: <-- Hit Enter
```

```
Installation mode (standard,expert) [standard]: <-- Hit Enter
```

```
Full qualified hostname (FQDN) of the server, eg server1.domain.tld
[server1.example.com]: <-- Hit Enter
```

```
MySQL server hostname [localhost]: <-- Hit Enter
MySQL server port [3306]: <-- Hit Enter
```

```
MySQL root username [root]: <-- Hit Enter
```

```
MySQL root password []: <-- Enter your MySQL root password
```

```
MySQL database to create [dbispconfig]: <-- Hit Enter
```

```
MySQL charset [utf8]: <-- Hit Enter
```

```
Configuring Postgrey
Configuring Postfix
Generating a 4096 bit RSA private key
.....................................................................
..++
.....................................................................
...................................................................++
writing new private key to 'smtpd.key'
-----
You are about to be asked to enter information that will be incorpora
ted
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]: <-- Enter 2 letter country code
State or Province Name (full name) [Some-State]: <-- Enter the name of the s
tate
Locality Name (eg, city) []: <-- Enter your city
Organization Name (eg, company) [Internet Widgits Pty Ltd]: <-- Enter co
mpany name or press enter
Organizational Unit Name (eg, section) []: <-- Hit Enter
Common Name (e.g. server FQDN or YOUR name) []: <-- Enter the server hostna
me, in my case: server1.example.com
Email Address []: <-- Hit Enter
Configuring Mailman
Configuring Dovecot
Configuring Spamassassin
Configuring Amavisd
Configuring Getmail
Configuring BIND
Configuring Jailkit
Configuring Pureftpd
Configuring Apache
Configuring vlogger
[INFO] service Metronome XMPP Server not detected
Configuring Ubuntu Firewall
Configuring Fail2ban
[INFO] service OpenVZ not detected
Configuring Apps vhost
Installing ISPConfig
ISPConfig Port [8080]:
```

```
Admin password [admin]:
```

```
Do you want a secure (SSL) connection to the ISPConfig web interface (y,n) [y]: <-- Hit Enter
```

```
Generating RSA private key, 4096 bit long modulus
.......................++
.....................................................................
...........................................................++
e is 65537 (0x10001)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]: <-- Enter 2 letter country code
State or Province Name (full name) [Some-State]: <-- Enter the name of the state
Locality Name (eg, city) []: <-- Enter your city
Organization Name (eg, company) [Internet Widgits Pty Ltd]: <-- Enter company name or press enter
Organizational Unit Name (eg, section) []: <-- Hit Enter
Common Name (e.g. server FQDN or YOUR name) []: <-- Enter the server hostname, in my case: server1.example.com
Email Address []: <-- Hit Enter
```

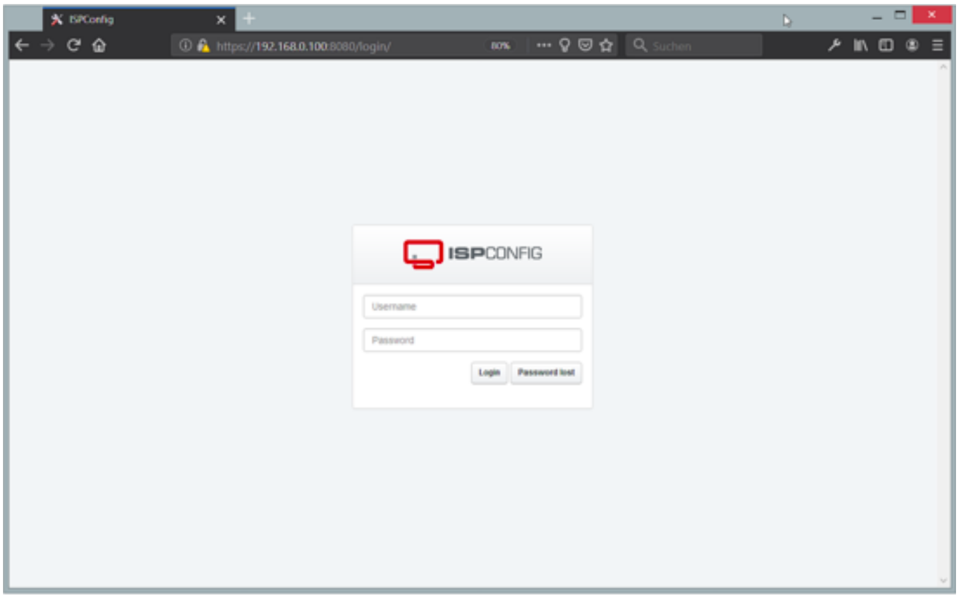
```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: <-- Hit Enter
An optional company name []: <-- Hit Enter
writing RSA key
```

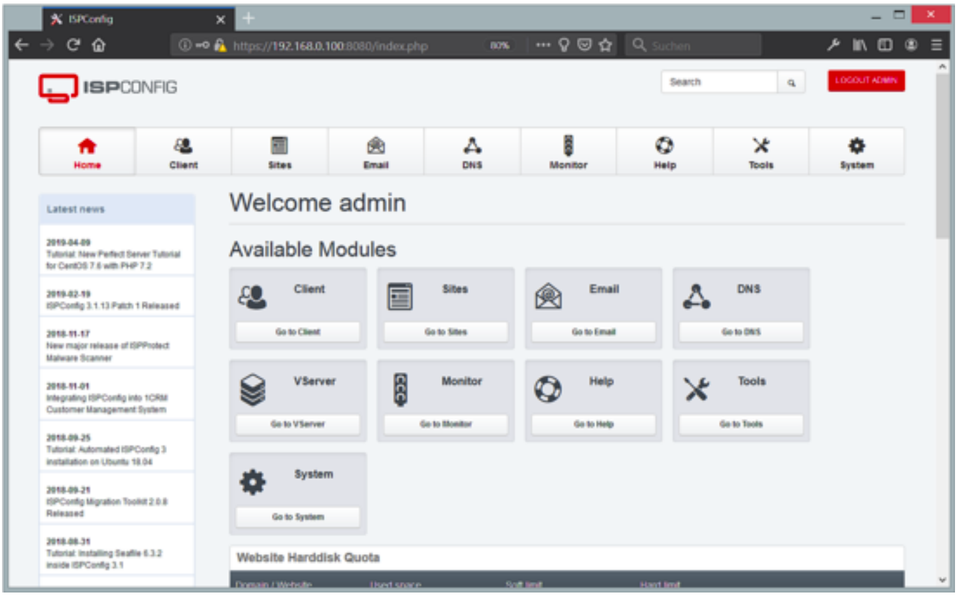
```

Configuring DBServer
Installing ISPConfig crontab
no crontab for root
no crontab for getmail
Detect IP addresses
Restarting services ...
Installation completed.
```

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

Afterwards you can access ISPConfig 3 under `http(s)://server1.example.com:8080/` or `http(s)://192.168.0.100:8080/` (http or https depends on what you chose during installation). Log in with the username `admin` and the password `admin` (you should change the default password after your first login):





Welcome admin

Available Modules

Latest news

2019-04-09
Tutorial: New Perfect Server Tutorial for CentOS 7.6 with PHP 7.2

2019-02-19
ISPConfig 3.1.13 Patch 1 Released

2018-11-17
New major release of ISPProtect Malware Scanner

2018-11-01
Integrating ISPConfig into 1CRM Customer Management System

2018-09-25
Tutorial: Automated ISPConfig 3 installation on Ubuntu 18.04

2018-09-21
ISPConfig Migration Toolkit 2.0.8 Released

2018-08-31
Tutorial: Installing Seafile 6.3.2 inside ISPConfig 3.1

The system is now ready to be used.

21.1 ISPConfig 3 Manual

In order to learn how to use ISPConfig 3, I strongly recommend downloading the ISPConfig 3 Manual.

On more than 300 pages, it covers the concept behind ISPConfig (admin, resellers, clients), explains how to install and update ISPConfig 3, includes a reference for all forms and form fields in ISPConfig together with examples of valid inputs, and provides tutorials for the most common tasks in ISPConfig 3. It also lines out how to make your server more secure and comes with a troubleshooting section at the end.

22 Virtual Machine Image Download of this Tutorial

This tutorial is available as ready to use virtual machine image in ovf/ova format that is compatible with VMWare and Virtualbox. The virtual machine image uses the following login details:

SSH / Shell Login

Username: administrator
Password: howtoforge

Username: root
Password: howtoforge

ISPConfig Login

Username: admin
Password: admin

MySQL Login

Username: root
Password: howtoforge

The IP of the VM is 192.168.0.100, it can be changed in the file /etc/network/interfaces. Please change all the above passwords to secure the virtual machine.

23 Links

- Debian: http://www.debian.org/
- ISPConfig: http://www.ispconfig.org/

About Till Brehm

Over 20 years experience as Software Developer and Linux System Administrator. Till Brehm is the founder and lead developer of the ISPConfig Hosting Control Panel software (since 2000) and he founded HowtoForge in 2005 as a place to share Linux knowledge with other Linux enthusiasts.

view as pdf | print

Share this page:

Recommend Tweet Follow G+

Suggested articles