Creating a Raspberry Pi Mail server

Contents

Introduction

If you have seen my previous 2 articles; **creating a Raspberry Pi web server** and **creating a Raspberry Pi LAMP server**, you will almost definately want to be able to allow your websites to send emails.

You will also want your server to be able to handle emails being sent to something@yourdomain.com?

This article will details exactly how to achieve this.

Postfix to allow outgoing emails

We need to ensure that outgoing emails can be sent from the Raspberry Pi by installing Postfix; a mail transport agent.

You will need to open Port 25 – SMTP to you Raspberry Pi via your router.

Run the following command to make sure you're running the latest software updates: `sudo apt-get update`.

Begin the Postfix Installation by entering `sudo apt-get install postfix` and confirm `Y` when prompted.

The Postfix setup will begin and you will see a blue screen with some information on the various configuration types; use the `Tab` key to highlight and click `OK`.

The final step is to configure the FQDN. It is important that you've already set the hostname as we're asked to confirm it at this stage. Set the system mail name to your fully qualified domain name, which may be pre-filled for you.

Once installation is complete, you will be able to receive mail from your Raspberry Pi web server.

If you are getting warnings in your log files that look similar to the following:

```
postfix: warning: inet_protocols: disabling IPv6 name/address
support: Address family not supported by protocol
```

You need to disable IPv6 by modifying `sudo nano /etc/postfix/main.cf` to include the line:

```
inet_protocols = ipv4
```

...at the bottom of the file if not already there and then reboot.

Additional configuration settings

Add your domain to the config files, so others can't abuse your mail system:

```
sudo postconf -e 'myorigin = example.com'
```

Add your hostname (computer name), use command `hostname` to display your hostname if not sure:

```
sudo postconf -e 'myhostname = tariqkhan.co.uk'
```

Now add the domain names that your system will handle:

```
sudo postconf -e 'relay_domains = example.com, example2.com,
example3.com'
```

Reload Postfix Server: `sudo postfix reload`

Test the mail server

Type `telnet tariqkhan.co.uk 25` and you should see:

```
220 tariqkhan.co.uk ESMTP Postfix (Debian/GNU)
```

Send an email to yourself:

```
mail from:<mail@tariqkhan.co.uk>

rcpt to:<tariq.uk@gmail.com>

data

To: tariq.uk@gmail.com

From: mail@tariqkhan.co.uk

Subject: This is a telnet email

This is an email on debian postfix over telnet.

Test number XX.
```

To end data hit `enter`, type in a dot (`.`) and hit `enter` again, then type `quit`.

You're done. Type `mail` in the command-line terminal and see if you have mail.

Postfix mail forwarding domains

This is useful if you do not have local mail boxes and your server is not acting as MX backup; do not use the following if `relay_domains`, `sql map` or `virtual map` is configured. The main purpose of these domains is to forward mail elsewhere.

The MX record of the domain you want to forward must be set to the VPS with Postfix installed, to check this you can use the command `dig` or use **http://www.geektools.com/digtool.php**. The result should be something like mail.yourdomain.com that is a A or CNAME record to your VPS.

This configuration is useful if you do not have local mailboxes and just want to use Postfix to forward emails to somewhere. The following example shows how to set up tariqkhan.co.uk as a mail forwarding domain. Open `/etc/postfix/main.cf` file and ensure the following lines exists:

```
virtual_alias_domains = tariqkhan.co.uk example.com moredomains.com

virtual_alias_maps = hash:/etc/postfix/virtual
```

This is the goal of these 2 directives:

virtual_alias_domains: Postfix is final destination for the specified list of virtual alias domains, that is, domains for which all addresses are aliased to addresses in other local or remote domains.

virtual_alias_maps: Optional lookup tables that alias specific mail addresses or domains to other local or remote address

Now create `/etc/postfix/virtual` file and add the following:

```
mail@tariqkhan.co.uk your.real@email.com
```

You can also implement a catch-all address i.e. email sent to anything@example.com should be forwarded to somewhereelse@another.domain.com:

```
@example.com  somewhereelse@another.domain.com
```

As we are using Debian, we need to create the `virtual.db` file. Run `sudo postmap virtual` from within the `/etc/postfix` folder.

Save and close the file, then reload postfix: `sudo /etc/init.d/postfix reload`.

More information available at: **http://wiki.debian.org/Postfix**

Forward Emails

Forwarding emails can be done via alias file located in `/etc/aliases`. Run this command to add alias maps:

```
postconf -e "alias_maps = hash:/etc/aliases"
```

You can now add your user to `/etc/aliases` like this:

```
root: tariqkhan
```

You can forward your emails to a different email address:

```
tariqkhan: myemail@example.com
```

Or you could forward your email while still getting a copy in your local mailbox

tariqkhan: tariqkhan myemail@example.com

When done adding aliases run this command which will create a database like file.

```
newaliases
```

Reload postfix: `sudo /etc/init.d/postfix reload`.

Relaying emails through GMail

If you run a Postfix mail server in your local network and have a dynamic IP address, chances are that it is blacklisted. By relaying your emails through another mail server that is hosted on a static IP address in a data centre (e.g. your ISP's mail server or GMail) you can prevent your emails from being categorised as spam.

To configure relaying on your Postfix mail server, you will need a valid email account with Gmail, which includes a username, password and provided that this mail server makes use of SMTP-AUTH, which it does.

Enter the following commands:

```
sudo postconf -e 'relayhost = [smtp.gmail.com]:587'

sudo postconf -e 'smtp_sasl_auth_enable = yes'

sudo postconf -e 'smtp_sasl_password_maps =
hash:/etc/postfix/sasl/passwd'

sudo postconf -e 'smtp_sasl_security_options = noanonymous'

sudo postconf -e 'smtp_tls_CAfile = /etc/postfix/cacert.pem'

sudo postconf -e 'smtp_use_tls = yes'
```

Your `/etc/postfix/main.cf` should look similar to:

```
relayhost = [smtp.gmail.com]:587

smtp_sasl_auth_enable = yes

smtp_sasl_password_maps = hash:/etc/postfix/sasl/passwd

smtp_sasl_security_options = noanonymous

smtp_tls_CAfile = /etc/postfix/cacert.pem

smtp_use_tls = yes
```

Create `sudo nano /etc/postfix/sasl/passwd` with your username and password values:

```
[smtp.gmail.com]:587 tariq.uk@gmail.com:your_password
```

Secure your new file `passwd` and make it usable for Postfix only. It must be owned by root and no one else should have read access to that file:

```
sudo chown root:root /etc/postfix/sasl/passwd

sudo chmod 600 /etc/postfix/sasl/passwd
```

Now we must convert `/etc/postfix/sasl/passwd` into a format that Postfix can read:

```
sudo postmap /etc/postfix/sasl/passwd
```

This will create the file `/etc/postfix/sasl_passwd.db`. You can run `ls -l /etc/postfix/sasl/` to verify that the results look similar to the following:

```
total 12

-rw------- 1 root root    50 May 13 16:34 passwd

-rw------- 1 root root 12288 May 13 16:44 passwd.db
```

Make sure you have the right certification authorities available to Postfix

```
•    cat /etc/ssl/certs/Thawte_Premium_Server_CA.pem >>
/etc/postfix/cacert.pem

•    cat /etc/ssl/certs/Equifax_Secure_Global_eBusiness_CA.pem >>
/etc/postfix/cacert.pem
```

- **This did not work for me: Permission denied**

- Now you can restart postfix but it will complain about not being able to authenticate the certificate. In order to fix the problem we will use the `ca-certificate` package we installed earlier to tell it where it can validate the certificate.

```
cat /etc/ssl/certs/Thawte_Premium_Server_CA.pem | sudo tee -a
/etc/postfix/cacert.pem
```

This one worked for me!

All that is left to do is restart Postfix: `/etc/init.d/postfix restart`

That's it. You can now test by sending emails over your mailserver and having a look at your mail log. You should see that all your emails are now passed on to smtp.gmail.com (except the ones that have a local recipient).