

# **Projekt: Lokalen Mailserver auf RaspberryPi installieren**

Autor: Thomas Dauner  
Matrikelnummer: 3138283  
Studiengang: Informatik  
Fach: Betriebssysteme  
Professor: Prof. Traub  
Hochschule: Technische Hochschule Ulm

## Inhaltsverzeichnis

1. Zielstellung: .....	2
2. Voraussetzungen: .....	2
3. Installation:.....	2
3.1. SSH auf dem RaspberryPi aktivieren .....	2
3.2. IP-Adresse im Router festlegen .....	4
3.3. Mailserver auf RaspberryPi installieren .....	5
4. Einrichtung in Thunderbird .....	13
5. E-Mails versenden .....	15
6. E-Mails empfangen.....	15
7. Virtuelle Mailadresse.....	16
8. Fazit/Schlussbemerkung.....	17
9. Quellen: .....	17

## 1. Zielstellung:

Es soll auf einem RaspberryPi ein Mailserver installiert werden, welcher im lokalen Netzwerk erreichbar ist. Es sollen Mails innerhalb des lokalen Netzwerks versendet und empfangen werden können.

## 2. Voraussetzungen:

Es existiert ein vorkonfiguriertes Netzwerk sowie ein vorinstalliertes Raspbian auf einem RaspberryPi. Dieser RaspberryPi ist mit einem Monitor, einer Maus und Tastatur sowie mit dem Netzwerk verbunden.

Zudem existiert ein vorinstalliertes Mailprogramm „Thunderbird“ auf einem Windows PC.

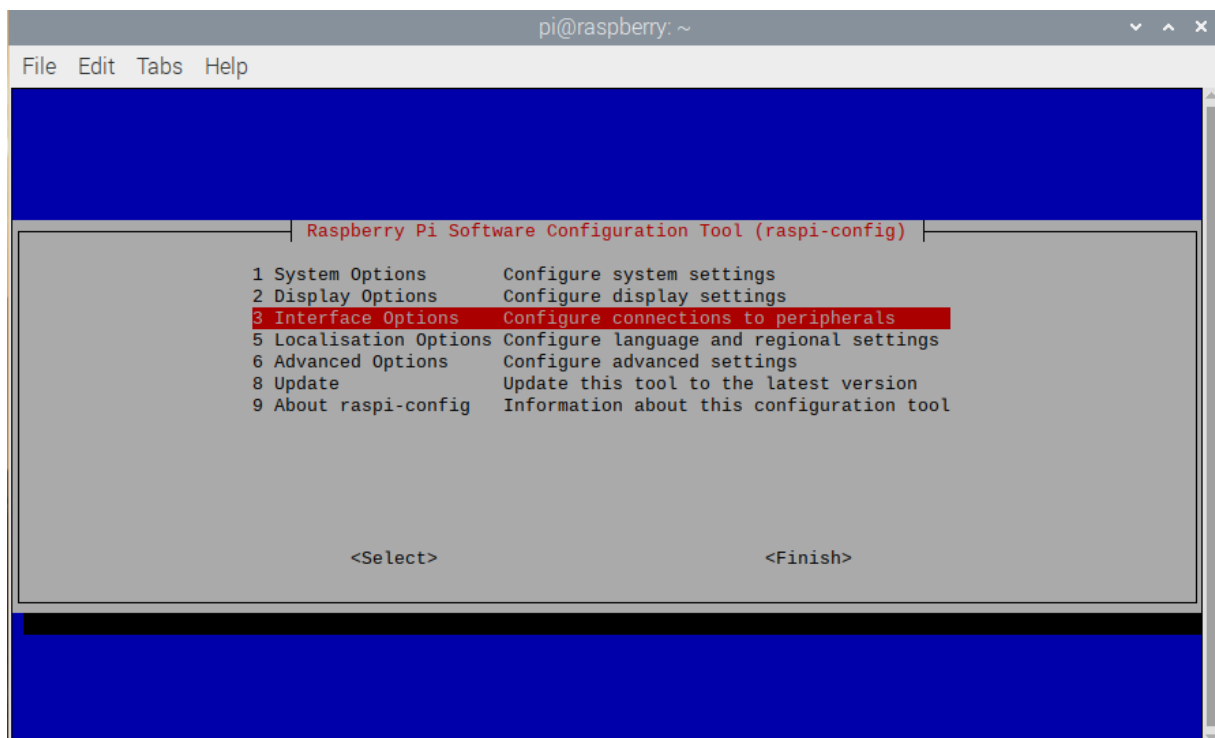
## 3. Installation:

### 3.1. SSH auf dem RaspberryPi aktivieren

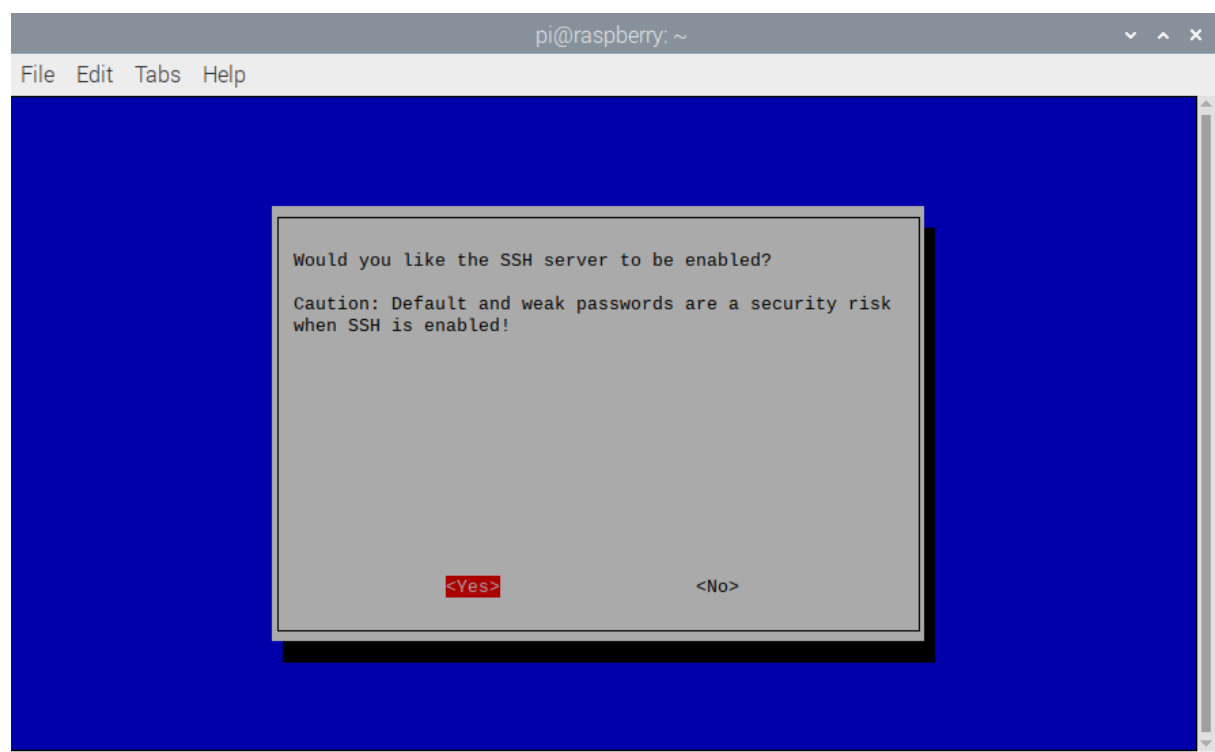
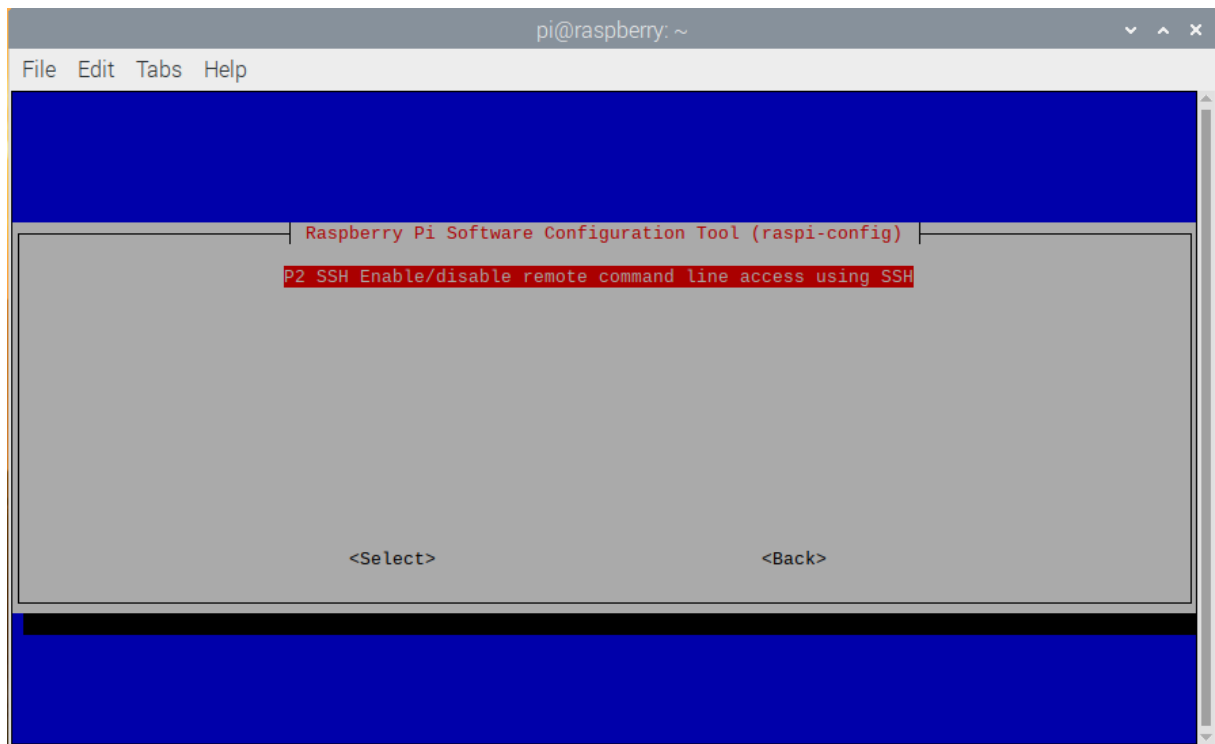
Ein Aktiviertes SSH ermöglicht später einfachen Zugriff um z.B. Wartungsarbeiten durchzuführen. Um SSH auf dem RaspberryPi zu aktivieren, muss im Terminal über den Befehl

```
sudo raspi-config
```

die Konfiguration des RaspberryPi's gestartet werden.



Hier wird nun über „Interface Options -> SSH Enable/Disable -> YES“ der SSH Zugriff aktiviert.



Über „Finish“ wird das Menü verlassen. Anschließend kann über z.B. PUTTY über SSH mittels der IP-Adresse des RaspberryPi's und dem Port 22 auf den RaspberryPi zugegriffen werden.

### 3.2. IP-Adresse im Router festlegen

Es wird der Befehl

```
ifconfig
```

im Terminal des RaspberryPi eingegeben. Dadurch werden alle Netzwerkadapter aufgelistet. Im eth0 Adapter kann nach „inet“ die IP-Adresse abgelesen werden. In diesem Fall die „192.168.178.86“.

```
pi@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.178.86 netmask 255.255.255.0 broadcast 192.168.178.255
    inet6 2003:c9:870b:8a00:1ea4:6358:c000:3585 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::3683:8f03:2247:6dbe prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:d5:9e:b1 txqueuelen 1000 (Ethernet)
    RX packets 5601 bytes 988569 (965.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2883 bytes 248118 (242.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 19 base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4615 bytes 1113324 (1.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4615 bytes 1113324 (1.0 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Nun wird die Konfigurationsseite des WLAN-Routers geöffnet. Unter Heimnetz -> Netzwerk werden alle verbundenen Geräte angezeigt. Es wird nun nach der entsprechenden IP des RaspberryPi's gesucht.

FRITZ!Box 7490			
Heimnetz > Netzwerk			
Netzwerkverbindungen    Netzwerkeinstellungen			
Die Tabelle zeigt alle per LAN oder WLAN mit der FRITZ!Box verbundenen Netzwerkgeräte sowie VPN-Verbindungen ins Heimnetz, die von FRITZ!Box-Benutzern und Apps (z.B. MyFRITZ!App, FRITZ!Mobile) im Heimnetz sind alle in einem Computer-Netzwerk verbunden und können Daten, Bilder, Musik und Videos miteinander austauschen. Netzwerkgeräte im Heimnetz können üi werden.			
Name	Verbindung	IP-Adresse	Eigenschaften
Diese FRITZ!Box			
fritz.box	DSL , ↓ 63,7 Mbit/s, ↑ 12,7 Mbit/s	192.168.178.1	WLAN 2,4 / 5 GHz
Aktive Verbindungen			
raspberrypi	LAN 2 mit 1 Gbit/s	192.168.178.86	
raspberrypi	LAN 2 mit 1 Gbit/s	192.168.178.87	Portfreigabe

Der entsprechende Eintrag wird nun konfiguriert. Über „Diesem Netzwerkgerät immer die gleiche IPv4-Adresse zuweisen“ wird festgelegt, dass auch bei zukünftigen Reboots des RaspberryPi die zugeteilte IP-Adresse erhalten bleibt. Dies ermöglicht, dass der Mailserver in Zukunft immer unter derselben IP-Adresse im lokalen Netzwerk erreicht werden kann.

### 3.3. Mailserver auf RaspberryPi installieren

Zunächst führen wir ein Update der Paketlisten und Pakete durch. Hierfür verwende ich die Befehle

```
sudo apt update  
sudo apt upgrade
```

Dies aktualisiert die Paketlisten und lädt die aktuellsten Updates herunter und installiert diese.

Anschließend beginne ich mit der Installation des Mailservers.

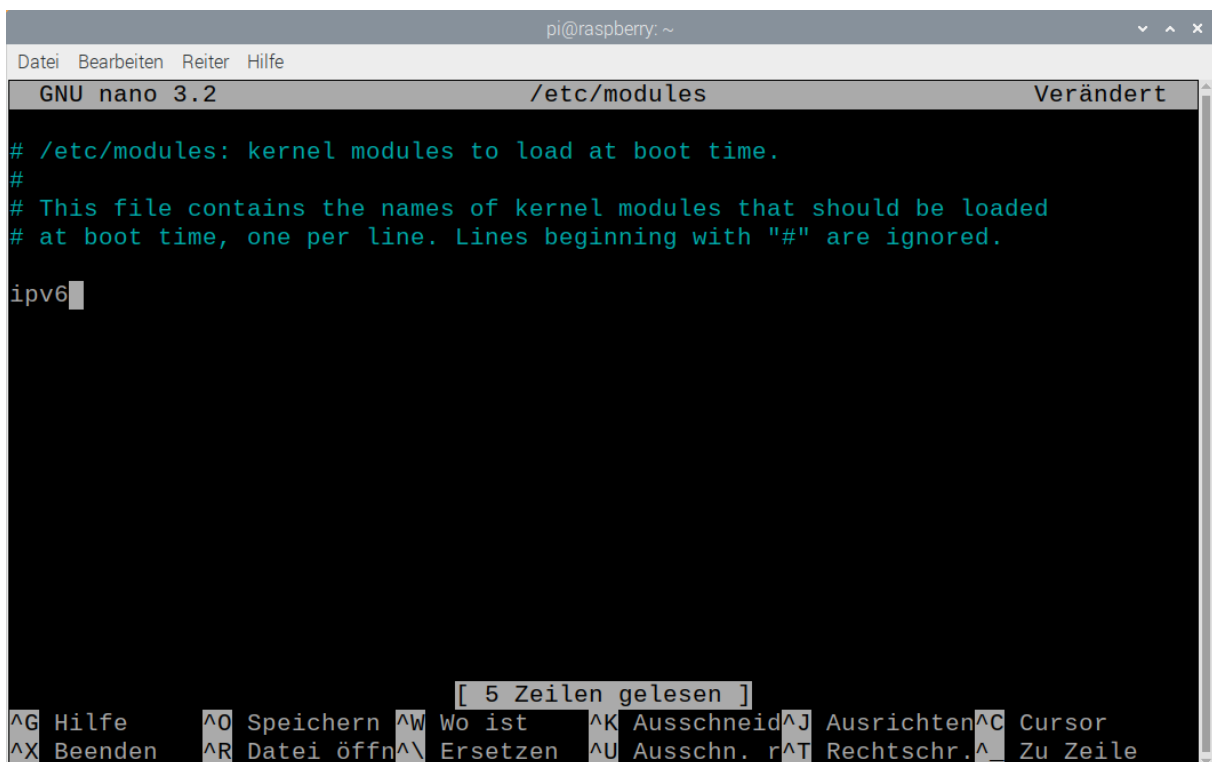
Zunächst muss mittels

```
sudo modprobe ipv6
```

die Nutzung der ipv6-Adresse am RaspberryPi aktiviert werden.  
Zudem wird mittels

```
sudo nano /etc/modules
```

die Datei „/etc/modules“ verändert. Hier wird „ipv6“ am Ende hinzugefügt, um das ipv6 Kernel Modul beim Booten mitzuladen. Mittels „STRG+X“ wird die Datei gespeichert und verlassen.

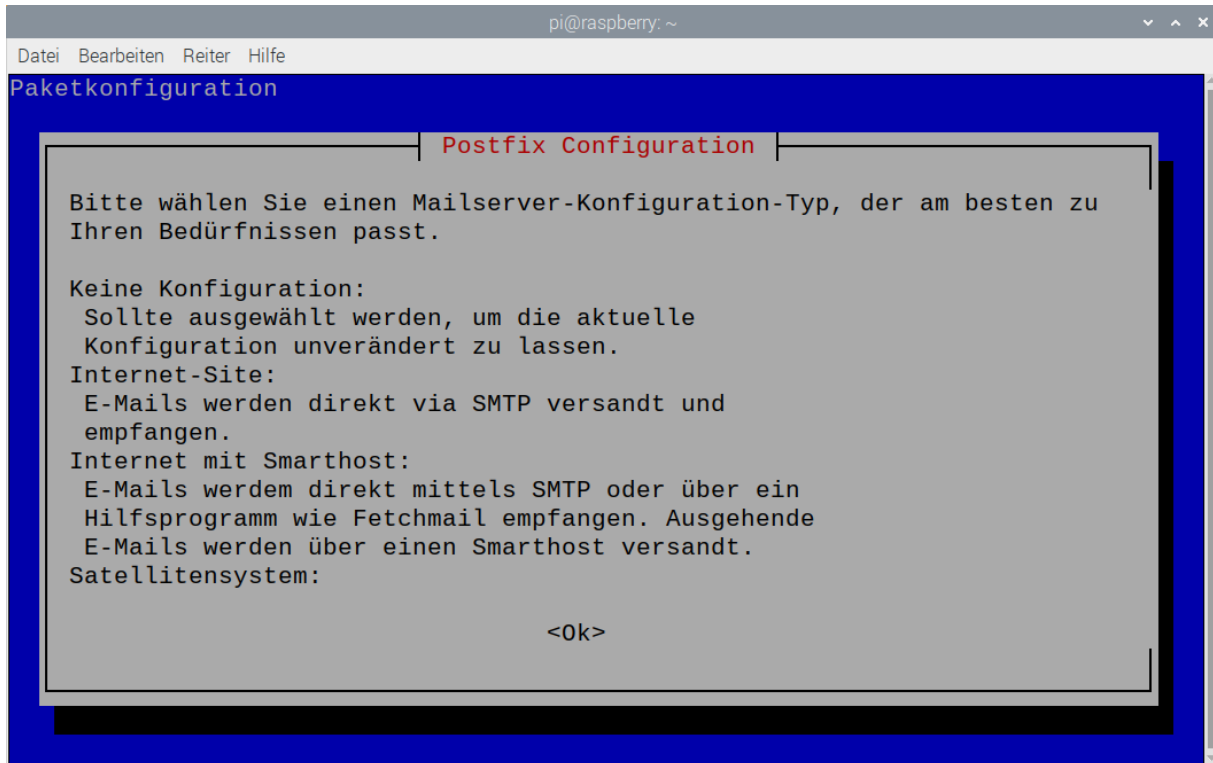


```
pi@raspberrypi: ~  
Datei Bearbeiten Reiter Hilfe  
GNU nano 3.2 /etc/modules Verändert  
# /etc/modules: kernel modules to load at boot time.  
#  
# This file contains the names of kernel modules that should be loaded  
# at boot time, one per line. Lines beginning with "#" are ignored.  
ipv6  
[ 5 Zeilen gelesen ]  
^G Hilfe ^O Speichern ^W Wo ist ^K Ausschneid ^J Ausrichten ^C Cursor  
^X Beenden ^R Datei öffnen ^\ Ersetzen ^U Ausschn. r ^T Rechtschr. ^_ Zu Zeile
```

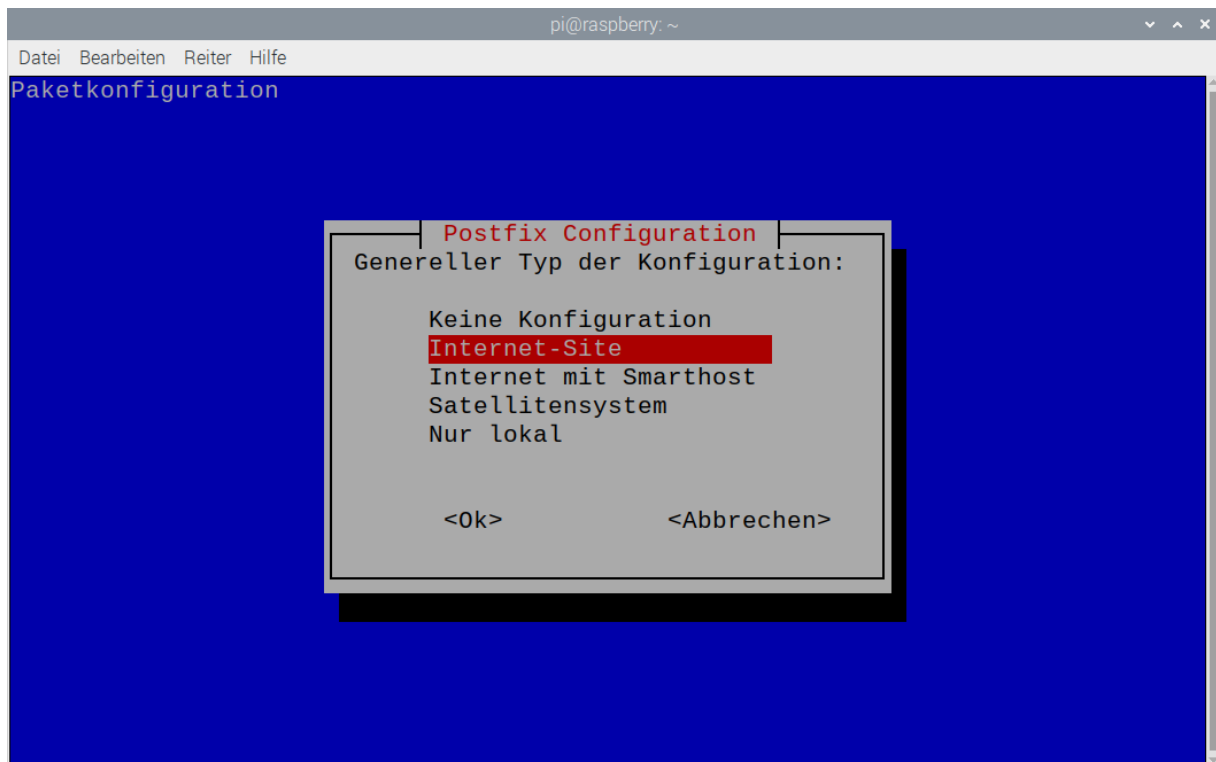
Nun kann die Software installiert werden. Hierfür wird folgender Befehl ausgeführt:

```
sudo apt-get install dovecot-common dovecot-imapd dovecot-pop3d dovecot-sieve dovecot-managesieved postfix
```

Dieser Befehl lädt diverse Pakete herunter und installiert diese. Es öffnet sich die Konfiguration des Mailservers.



Ich wähle als Standardkonfiguration die „internet-site“. Außerdem vergebe ich einen E-Mail Namen des Systems. Diese Einstellungen sind aktuell nicht relevant, da sie zu einem späteren Zeitpunkt noch geändert werden.



Ist die Software installiert können die Accounts der Nutzer erstellt werden, welche den Mailserver später nutzen möchten. Dies wird mit

```
sudo useradd -d /home/ACCOUNTNAME -m ACCOUNTNAME
```

durchgeführt. In diesem Fall habe ich zwei Accounts namens „Thomas“ und „Nina“ erstellt. Die Befehle lauten dann

```
sudo useradd -d /home/Thomas -m Thomas  
sudo useradd -d /home/Nina -m Nina
```

Nun müssen den Accounts noch Passwörter zugewiesen werden. Dies geschieht mit den Befehlen:

```
sudo passwd Thomas  
sudo passwd Nina
```

Es öffnet sich dann eine Eingabeaufforderung, bei welcher das Passwort neu gesetzt werden kann. Hier sollten selbstverständlich sichere Passwörter verwendet werden.



```
pi@raspberrypi:~ $ sudo useradd -d /home/Thomas -m Thomas
pi@raspberrypi:~ $ sudo useradd -d /home/Nina -m Nina
pi@raspberrypi:~ $ sudo passwd Thomas
Geben Sie ein neues Passwort ein:
Geben Sie das neue Passwort erneut ein:
passwd: Passwort erfolgreich geändert
pi@raspberrypi:~ $ sudo passwd Nina
Geben Sie ein neues Passwort ein:
Geben Sie das neue Passwort erneut ein:
passwd: Passwort erfolgreich geändert
pi@raspberrypi:~ $
```

Nun wird in jedem der beiden Accounts das entsprechende Mailverzeichnis erstellt, in welchem die E-Mails gespeichert werden. Hierfür wird zunächst in den entsprechenden Account gewechselt. Das geschieht mit

```
sudo login Thomas bzw. sudo login Nina
```

Beim Login wird nun das zuvor definierte Passwort gefordert. Nach der Eingabe dessen lautet nun der Benutzer nicht mehr „pi@raspberrypi“ sondern „thomas@raspberrypi“ bzw. „nina@raspberrypi“. Der Login auf den entsprechenden Account hat also funktioniert.

```
pi@raspberrypi:~ $ sudo login Thomas
Passwort:
Linux raspberrypi 4.19.0-13-amd64 #1 SMP Debian 4.19.160-2 (2020-11-28) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Thomas@raspberrypi:~ $
```

Nun wird mittels

```
mkdir Maildir
```

das Verzeichnis erstellt, in welchem die Mails gespeichert werden sollen.

Mittels

```
exit
```

kann der aktuelle Benutzeraccount verlassen werden. Dies wird bei beiden Accounts durchgeführt.

```
Nina@raspberrypi:~ $ mkdir Maildir
Nina@raspberrypi:~ $ exit
Abgemeldet
pi@raspberrypi:~ $
```

Nun sind beide Konten fertig konfiguriert. Es muss nur noch der Mailserver entsprechend konfiguriert werden. Hierfür muss die Konfigurationsdatei mittels

```
sudo nano /etc/dovecot/conf.d/10-ssl.conf
```

Verändert werden. Es wird die Einstellung „ssl“ von „yes“ zu „required“ geändert.

```
##  
## SSL settings  
##  
# SSL/TLS support: yes, no, required. <doc/wiki/SSL.txt>  
ssl = required
```

Mittels „STRG+O“ und „STRG+X“ wird die Datei gespeichert und verlassen. Nun wird eine weitere Datei verändert. Hierfür wird der Befehl

```
sudo nano /etc/dovecot/conf.d/10-mail.conf
```

Hier wird die Einstellung „mail\_location = mbox:~/mail:INBOX=/var/mail/%u“ auskommentiert, indem an den Beginn der Zeile eine „#“ gesetzt wird. Stattdessen wird die Zeile „mail\_location = maildir:~/Maildir“ einkommentiert, indem das „#“ zum Zeilenbeginn entfernt wird.

```
#  
# See doc/wiki/Variables.txt for full list. Some examples:  
#  
# mail_location = maildir:~/Maildir  
# mail_location = mbox:~/mail:INBOX=/var/mail/%u  
# mail_location = mbox:/var/mail/%d/%n/%n:INDEX=/var/indexes/%d/%n/%n  
#  
# <doc/wiki/MailLocation.txt>  
#  
#mail_location = mbox:~/mail:INBOX=/var/mail/%u
```

Auch diese Datei wird mittels „STRG+O“ und „STRG+X“ gespeichert und verlassen.

Nun wird die Datei „main.cf“ verändert. Hierfür wird sie mittels

```
sudo nano /etc/postfix/main.cf
```

geöffnet.

```
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache  
  
# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for  
# information on enabling SSL in the smtp client.  
  
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination  
myhostname = raspberry.fritz.box  
alias_maps = hash:/etc/aliases  
alias_database = hash:/etc/aliases  
mydestination = $myhostname, raspberrypi, raspberry.fritz.box, localhost.fritz.box, localhost  
relayhost =  
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128  
mailbox_size_limit = 0  
recipient_delimiter = +  
inet_interfaces = all  
inet_protocols = all
```

Ab Zeile „myhostname = raspberrypi.fritz.box“ wird alles entfernt und durch folgenden Text ersetzt:

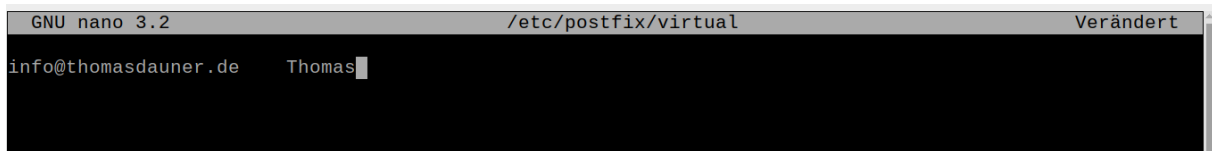
```
myhostname = PiMailServer
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = PiMailServer, localhost.localdomain,
localhost, thomasdauner.de
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
message_size_limit = 50000000
recipient_delimiter = +
home_mailbox = Maildir/
smtpd_sasl_auth_enable = yes
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/dovecot-auth
smtpd_sasl_authenticated_header = yes
smtpd_sasl_local_domain = $myhostname
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions =
reject_unknown_sender_domain
reject_unknown_recipient_domain reject_unauth_pipelining
permit_mynetworks permit_sasl_authenticated
reject_unauth_destination
smtpd_sender_restrictions = reject_unknown_sender_domain
mailbox_command = /usr/lib/dovecot/deliver -c
/etc/dovecot/conf.d/01-mail-stack-delivery.conf -m
"${EXTENSION}"
smtp_use_tls = yes
smtpd_tls_received_header = yes
smtpd_tls_mandatory_protocols = SSLv3, TLSv1
smtpd_tls_mandatory_ciphers = medium
smtpd_tls_auth_only = yes
tls_random_source = dev:/dev/urandom
virtual_alias_maps = hash:/etc/postfix/virtual
inet_interfaces = all
```

Auch diese Datei wird mittels „STRG+O“ und „STRG+X“ gespeichert und verlassen.

In der Datei „/etc/postfix/virtual“ können zudem Mail-Adressen spezifiziert werden, zu denen es keinen eigenen Account auf dem RaspberryPi gibt. Auch diese Datei wird mittels

```
sudo nano /etc/postfix/virtual
```

bearbeitet. Hier trage ich nun beispielhaft eine E-Mail Adresse ein, welche keinem Benutzer auf dem RaspberryPi zugeordnet ist. Ich wähle [info@thomasdauner.de](mailto:info@thomasdauner.de). Hinter der Mail Adresse wird spezifiziert, welchem RaspberryPi bzw. Mailbenutzer diese E-Mail Adresse zugeordnet werden soll.



Auch diese Datei wird mittels „STRG+O“ und „STRG+X“ gespeichert und verlassen.

Um die Änderungen zu übernehmen muss nun der Befehl

```
sudo postmap /etc/postfix/virtual
```

ausgeführt werden. Nun sind die Einstellungen der virtuellen Adressen übernommen.

Nun wird mittels

```
sudo nano /etc/dovecot/conf.d/01-mail-stack-delivery.conf
```

eine weitere Konfigurationsdatei erstellt, in die folgender Inhalt gespeichert wird:

```
# Some general options
protocols = imap pop3 sieve
disable_plaintext_auth = yes
ssl = yes
ssl_cert = </etc/ssl/certs/ssl-cert-snakeoil.pem
ssl_key = </etc/ssl/private/ssl-cert-snakeoil.key
ssl_cipher_list =
ALL:!LOW:!SSLv2:ALL:!aNULL:!ADH:!eNULL:!EXP:RC4+RSA:+HIGH:
+MEDIUM
mail_location = maildir:~/Maildir
auth_username_chars =
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ012345
67890.-_@

# IMAP configuration
protocol imap {
    mail_max_userip_connections = 10
    imap_client_workarounds = delay-newmail
}

# POP3 configuration
protocol pop3 {
    mail_max_userip_connections = 10
    pop3_client_workarounds = outlook-no-nuls oe-ns-
eoh
}

# LDA configuration
protocol lda {
    postmaster_address = postmaster
    mail_plugins = sieve
```

```
        quota_full_tempfail = yes
        deliver_log_format = msgid=%m: %$
        rejection_reason = Your message to <%t> was
automatically rejected:%n%r
    }

# Plugins configuration
plugin {
    sieve=~/.dovecot.sieve
    sieve_dir=~/.sieve
}

# Authentication configuration
auth_mechanisms = plain login

service auth {
    # Postfix smtp-auth
    unix_listener /var/spool/postfix/private/dovecot-auth {
        mode = 0660
        user = postfix
        group = postfix
    }
}
```

Die Datei wird gespeichert und verlassen.

Um nun alle Änderungen anzuwenden müssen die Services „postfix“ sowie „dovecot“ neu gestartet werden. Hierzu werden die Befehle

```
sudo service postfix restart
sudo service dovecot restart
```

verwendet.

## 4. Einrichtung in Thunderbird

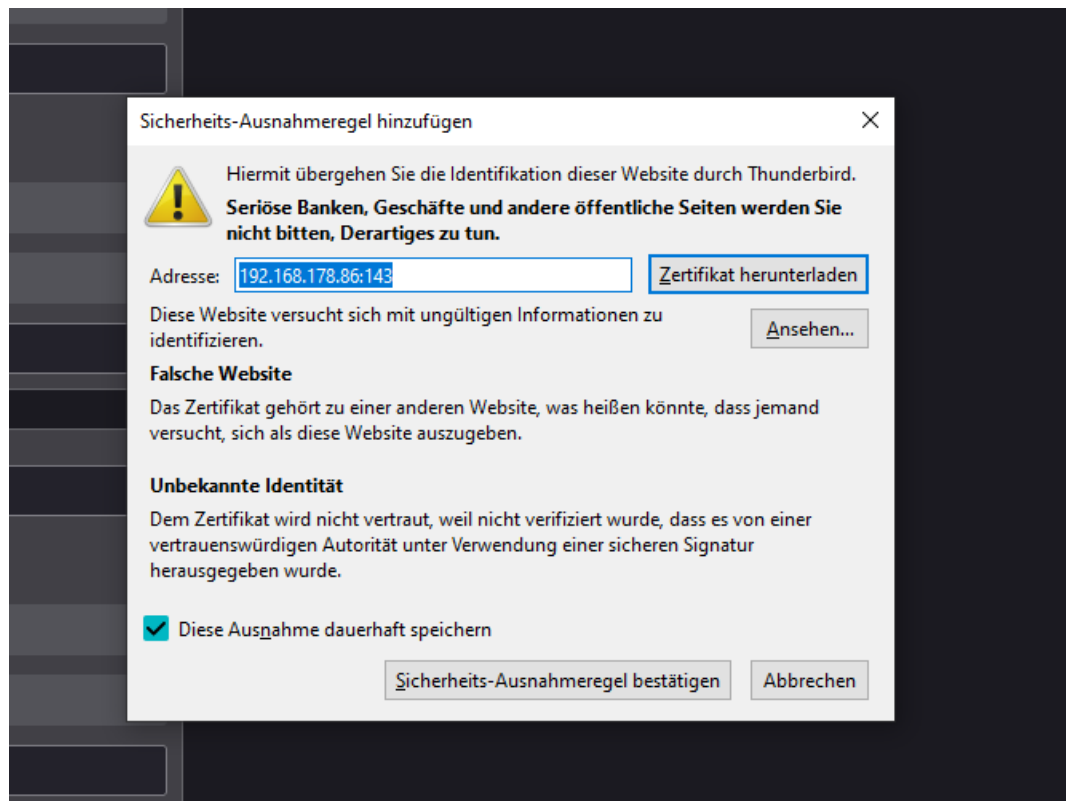
Um nun Mails empfangen und versenden zu können wird das E-Mail Programm Thunderbird verwendet. Dieses kann kostenfrei von der Thunderbird Website heruntergeladen und installiert werden. Dies habe ich bereits im Vorfeld durchgeführt, da ich das Programm privat nutze.

Nun wird über Extras -> Konteneinstellungen ein neues Konto hinzugefügt.

Als Name gebe ich hier „Thomas“ ein. Die E-Mail Adresse lautet [thomas@thomasdauner.de](mailto:thomas@thomasdauner.de). Dies ist die E-Mail Adresse, welche ich im vorherigen Schritt auf dem RaspberryPi angelegt habe. Als Passwort trage ich das definierte Passwort ein.

Über den Button „manuelle Einrichtung“ lege ich nun noch einige Einstellungen fest. Als „Hostname“ wird die IP des RaspberryPi angegeben. Über „Erneut Testen“ wird die Verbindung zum RaspberryPi getestet. Eine grüne Nachricht zeigt an, dass der Mailserver erfolgreich gefunden wurde. Über den Button „Fertig“ wird die Eingabe bestätigt.

Nun wird ein Zertifikat benötigt. Über „Sicherheits-Ausnahmeregel bestätigen“ wird bestätigt, dass auch ohne gültiges Zertifikat der Mailserver benutzt werden darf. Das Konto wurde nun erfolgreich hinzugefügt.

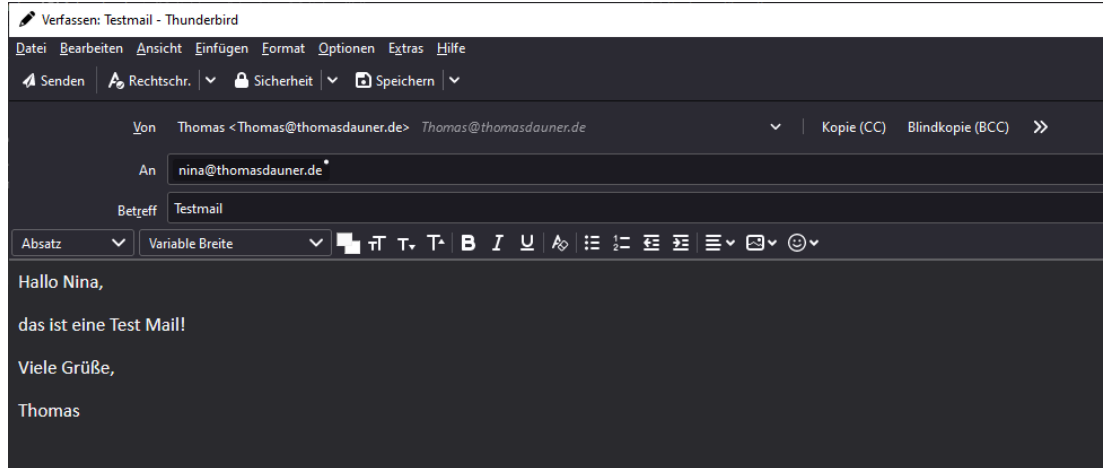


Den selben Vorgang wiederhole ich nun und füge das Konto [nina@thomasdauner.de](mailto:nina@thomasdauner.de) zu Thunderbird hinzu. Auch dieses Konto konnte erfolgreich hinzugefügt werden.

## 5. E-Mails versenden

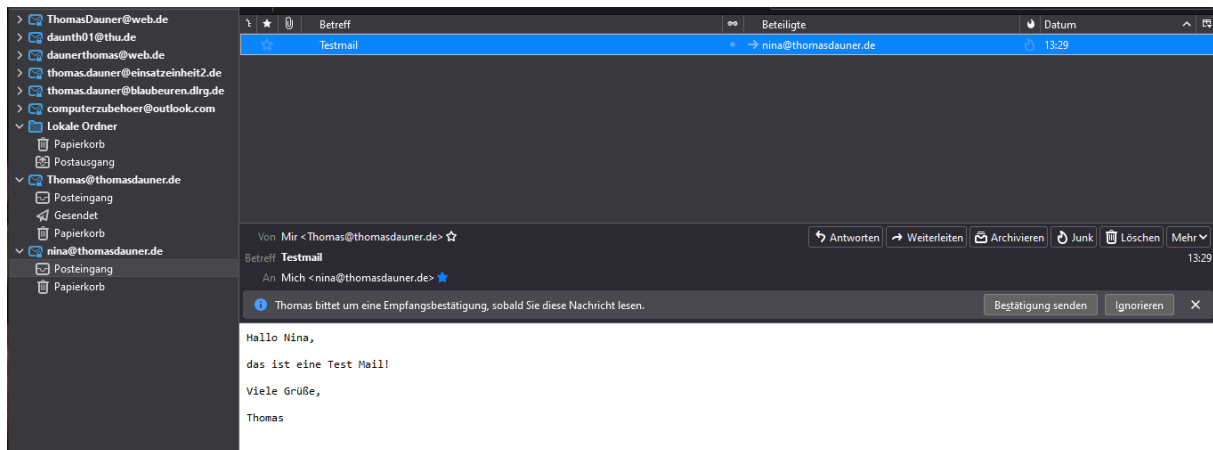
Um nun eine E-Mail zu versenden wähle ich den Mail-Account [Thomas@thomasdauner.de](mailto:Thomas@thomasdauner.de) aus und klicke auf „verfassen“ um eine neue Mail zu verfassen. Als Empfänger wähle ich [nina@thomasdauner.de](mailto:nina@thomasdauner.de).

Über „Senden“ wird die Mail versendet.



## 6. E-Mails empfangen

Direkt nach dem versenden der E-Mail von [thomas@thomasdauner.de](mailto:thomas@thomasdauner.de) wurde mir der Eingang der E-Mail auf dem Account [nina@thomasdauner.de](mailto:nina@thomasdauner.de) bestätigt. Die Mail wird im Postfach angezeigt und kann gelesen werden.

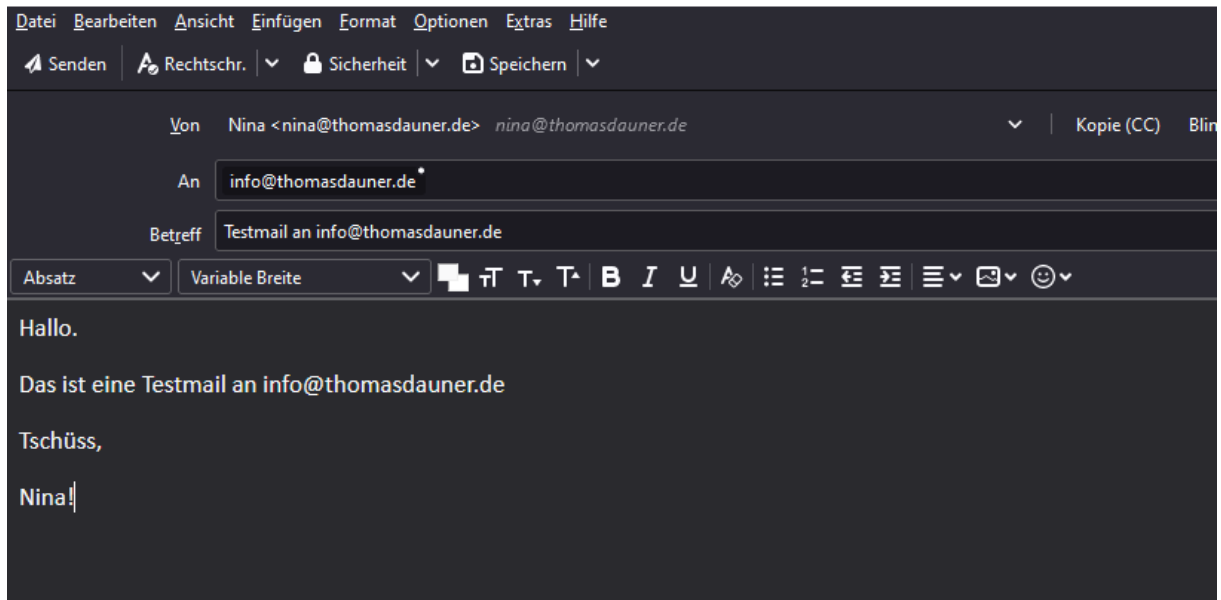




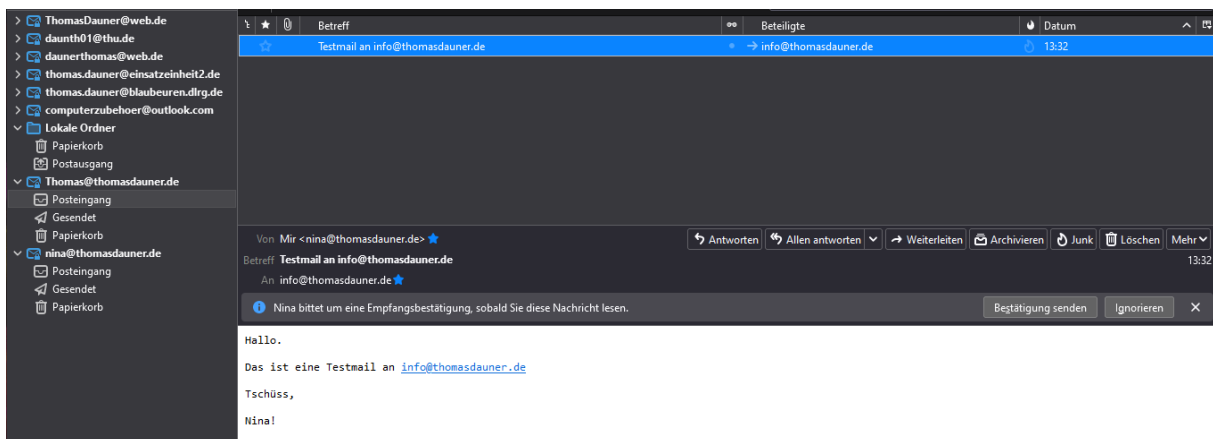
## 7. Virtuelle Mailadresse

Nun teste ich noch die virtuelle Mailadresse [info@thomasdauner.de](mailto:info@thomasdauner.de). Hierfür verfasse ich von dem Account [nina@thomasdauner.de](mailto:nina@thomasdauner.de) eine E-Mail an [info@thomasdauner.de](mailto:info@thomasdauner.de) und sende diese ab.

Verfassen: Testmail an info@thomasdauner.de - Thunderbird



Da im Vorfeld festgelegt wurde, dass E-Mails an die Adresse [info@thomasdauner.de](mailto:info@thomasdauner.de) an die Adresse [thomas@thomasdauner.de](mailto:thomas@thomasdauner.de) weitergeleitet werden, kam diese E-Mail im Postfach von [thomas@thomasdauner.de](mailto:thomas@thomasdauner.de) an und kann gelesen werden.



## 8. Fazit/Schlussbemerkung

Die Installation eines lokalen Mailservers ist nicht sehr schwer und kann auf einem RaspberryPi ohne Probleme durchgeführt werden. Allerdings können mit diesem Mailserver nur im lokalen Netzwerk E-Mails versendet werden.

Um E-Mails ins Internet zu versenden wird eine Domain benötigt, bei welcher A-Records sowie MX-Records angepasst werden können. Hier wird die IP des eigenen Servers als Adresse des Mailservers festgelegt.

Möchte man diesen Server jedoch privat Zuhause betreiben, besteht das Problem der wechselnden IP-Adresse. Der Provider teilt dem Haushalt regelmäßig eine neue IP-Adresse zu. Um dennoch über einen festen Link immer die korrekte IP-Adresse des privaten Haushalts zu erreichen, wird eine sogenannte „Dynamic DNS“-Adresse verwendet. Hierbei läuft auf dem Server im privaten Haushalt ein kleines Skript, welches regelmäßig an den DDNS Server die aktuelle eigene IP-Adresse meldet. Somit kann über eine feste URL dauerhaft der korrekte Haushalt bzw. der korrekte Server erreicht werden, auch wenn dessen IP sich laufend ändert.

Bei Webservern ist dies kein Problem, bei Mailservern stößt man hier allerdings sehr schnell auf Probleme. Oft sind solche DDNS-Server auf Backlists, um Spam zu verhindern. Meistens können bei DDNS Adressen keine weiteren Einstellungen zum MX-Record vorgenommen werden.

Generell gestaltet sich die Konfiguration eines privaten Mailservers welcher mit dem Internet verbunden werden soll als sehr schwierig, wenn nicht eine feste IP seitens des Servers existiert. Daher ist es eher unsinnig einen realen Mailserver auf einem RaspberryPi zu betreiben. Als kleines Projekt um den prinzipiellen Umgang mit einem Mailserver zu lernen ist es dennoch ein interessantes Thema und hat mir Spaß bereitet.

## 9. Quellen:

<https://www.pc-magazin.de/ratgeber/raspberry-pi-als-mailserver-einrichten-installieren-1836244.html>

[www.noip.com](http://www.noip.com)

<https://www.ionos.de/digitalguide/server/konfiguration/einen-eigenen-raspberry-pi-mailserver-einrichten/>

<https://www.raspberrypi.com/software/>

<https://www.citadel.org/easyinstall.html>

<https://www.electromaker.io/tutorial/blog/how-to-build-a-raspberry-pi-email-server>

<https://www.youtube.com/watch?v=FESyyWMFJvM&t=691s>