SETTING UP MAIL ON A RASPBERRY PI

Works as of December 17, 2019

STEP 1: Install mail-utils

```
$  sudo apt install mailutils
```

STEP 2: Install mutt

```
$  sudo apt install mutt
```

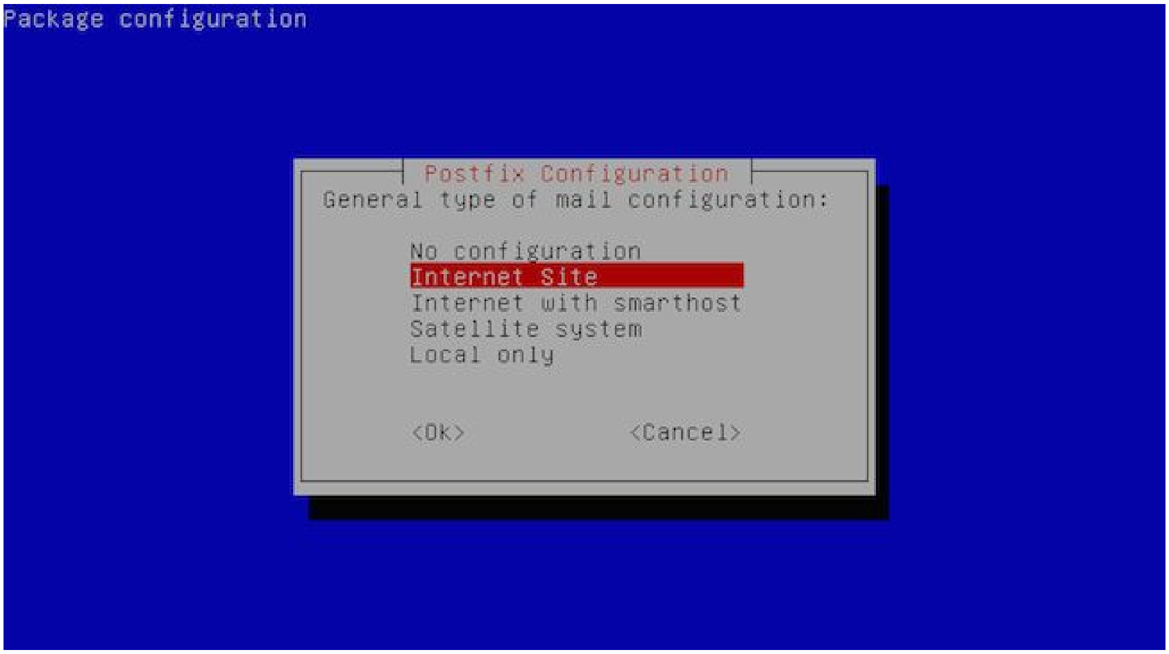
STEP 3: Install postfix

In this section, you will install Postfix as well as *libsasl2*, a package which helps manage the Simple Authentication and Security Layer (SASL).

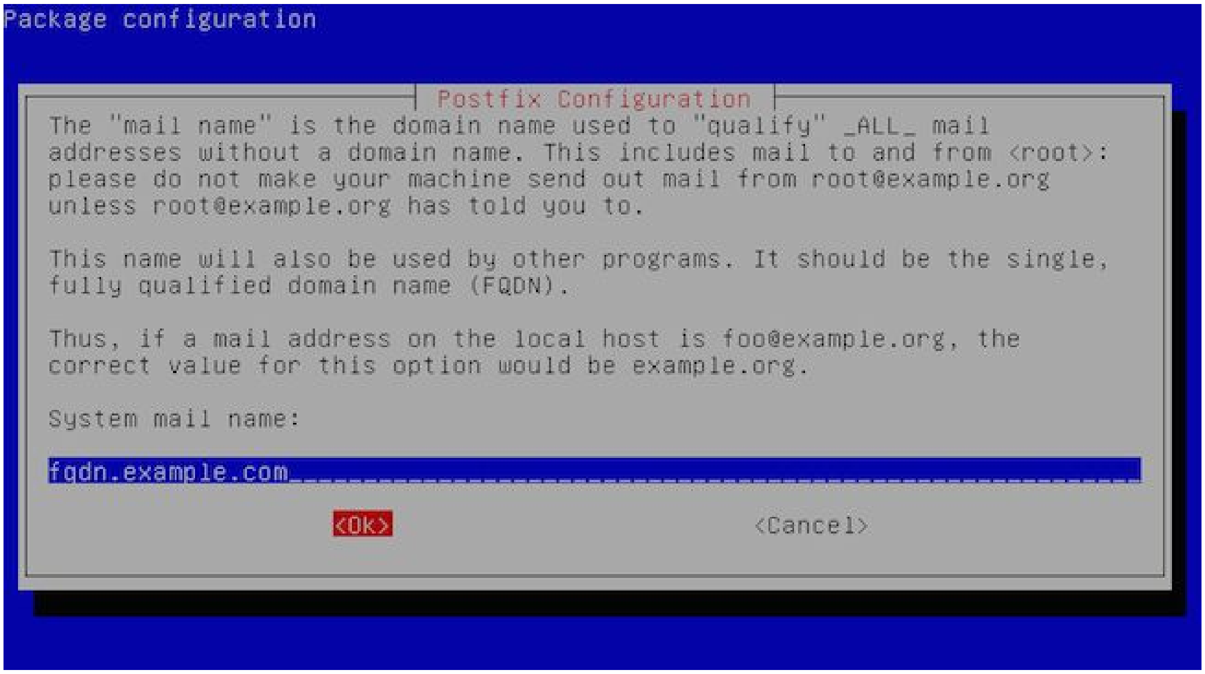
1. Install Postfix and the `libsasl2-modules` package:

```
$  sudo apt-get install libsasl2-modules postfix
```

2. During the Postfix installation, a prompt will appear asking for your **General type of mail configuration**. Select **Internet Site**:



3. Enter the fully qualified name of your domain. In this example, **fqdn.example.com**: use `smtp.gmail.com` for gmail



```
The "mail name" is the domain name used to "qualify" _ALL_ mail
addresses without a domain name. This includes mail to and from <root>:
please do not make your machine send out mail from root@example.org
unless root@example.org has told you to.

This name will also be used by other programs. It should be the single,
fully qualified domain name (FQDN).

Thus, if a mail address on the local host is foo@example.org, the
correct value for this option would be example.org.

System mail name:

fqdn.example.com
```

4. Once the installation is complete, confirm that the `myhostname` parameter is configured with your server's FQDN:

/etc/postfix/main.cf

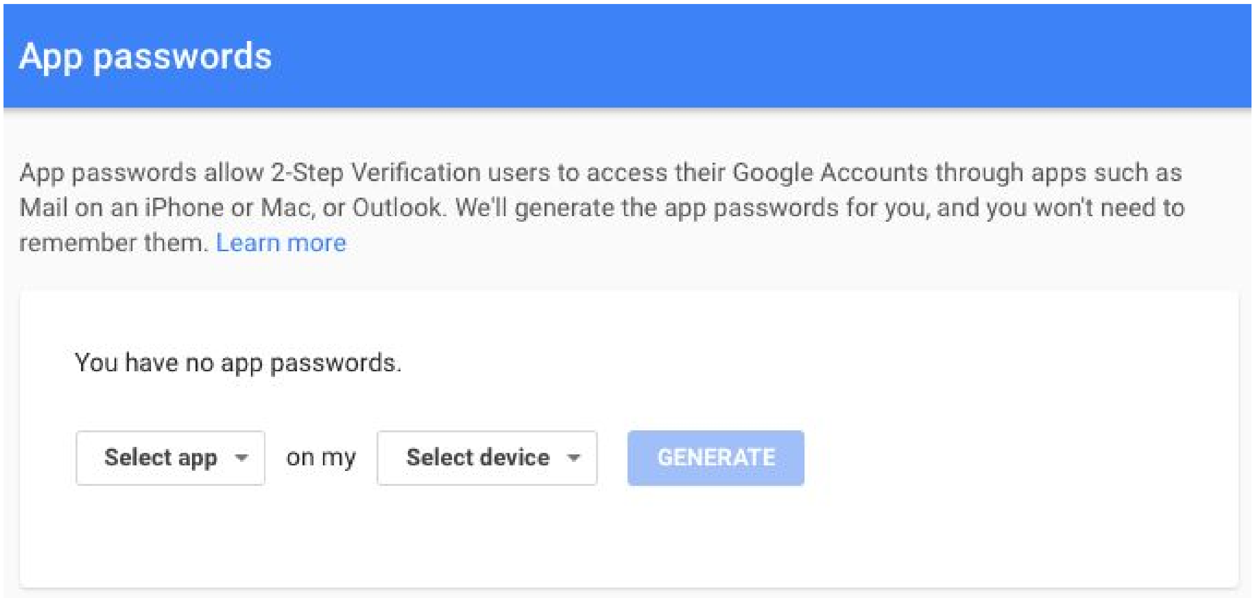
```
myhostname = fqdn.example.com
```

STEP 4: Generate an App Password for Postfix

When Two-Factor Authentication (2FA) is enabled, Gmail is preconfigured to refuse connections from applications like Postfix that don't provide the second step of authentication. While this is an important security measure that is designed to restrict unauthorized users from accessing your account, it hinders sending mail through some SMTP clients as you're doing here. Follow these steps to configure Gmail to create a Postfix-specific password:

1. Log in to your email, then click the following link: Manage your account access and security settings. Scroll down to "Password & sign-in method" and click **2-Step Verification**. You may be asked for your password and a verification code before continuing. Ensure that 2-Step Verification is enabled.

2. Click the following link to Generate an App password for Postfix:

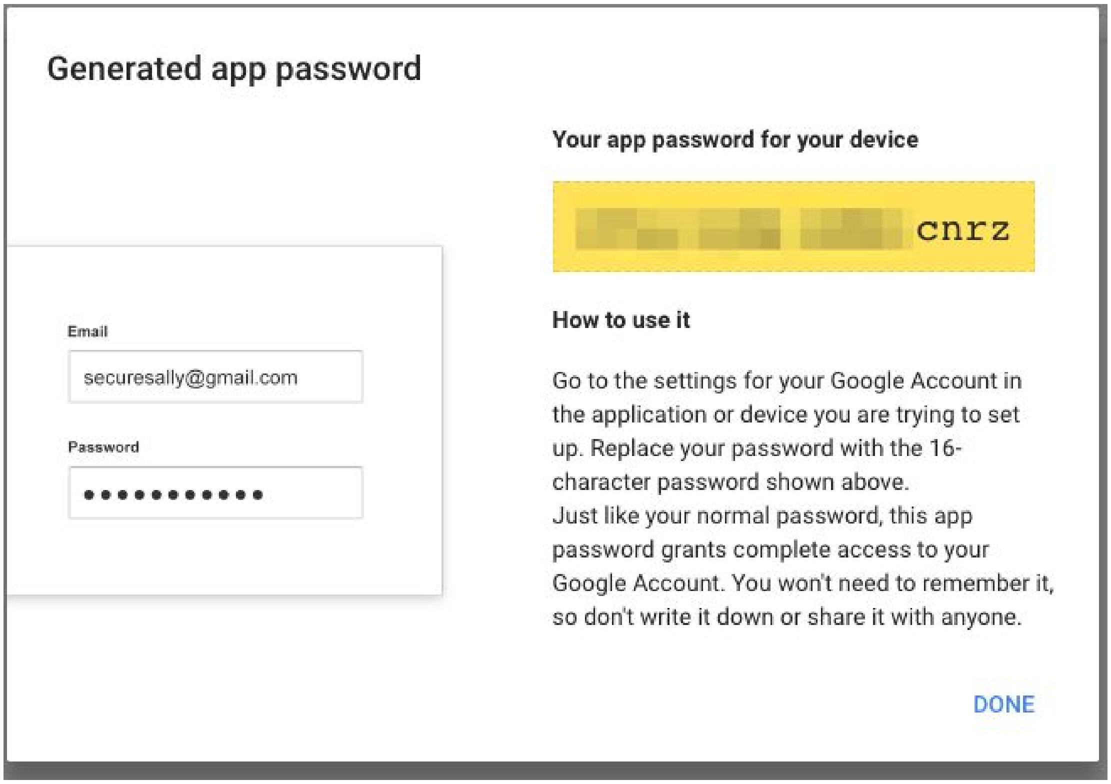


App passwords

App passwords allow 2-Step Verification users to access their Google Accounts through apps such as Mail on an iPhone or Mac, or Outlook. We'll generate the app passwords for you, and you won't need to remember them. Learn more

You have no app passwords.

3. Click **Select app** and choose **Other (custom name)** from the dropdown. Enter “Postfix” and click **Generate**.
4. The newly generated password will appear. Write it down or save it somewhere secure that you’ll be able to find easily in the next steps, then click **Done**:



Generated app password

Email

securesally@gmail.com

Password

●●●●●●●●●●●●

Your app password for your device

[redacted] cnrz

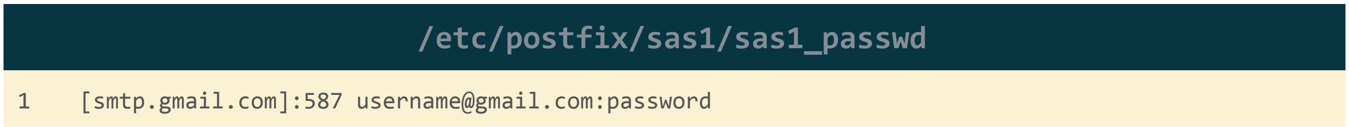
How to use it

Go to the settings for your Google Account in the application or device you are trying to set up. Replace your password with the 16-character password shown above.
Just like your normal password, this app password grants complete access to your Google Account. You won't need to remember it, so don't write it down or share it with anyone.

STEP 5: Add Gmail Username and Password to Postfix

Usernames and passwords are stored in `sasl_passwd` in the `/etc/postfix/sasl/` directory. In this section, you'll add your email login credentials to this file and to Postfix.

1. Open or create the `/etc/postfix/sasl/sasl_passwd` file and add the SMTP Host, username, and password information:



/etc/postfix/sas1/sas1_passwd

```
[smtp.gmail.com]:587 username@gmail.com:password
```

> **Note**
> The SMTP server address configuration `smtp.gmail.com` supports message submission over port **587** (StartTLS) and port **465** (SSL). Whichever protocol you choose, be sure the port number is the same in `/etc/postfix/sasl/sasl_passwd` and `/etc/postfix/main.cf` files. See Google's G Suite Administrator Help for more information.

2. Create the hash db file for Postfix by running the **`postmap`** command:

```
$  sudo postmap /etc/postfix/sasl/sasl_passwd
```

If all went well, you should have a new file named `sasl_passwd.db` in the `/etc/postfix/sasl/` directory.

STEP 6: Secure Your Postfix Hash Database and Email Password Files

The `/etc/postfix/sasl/sasl_passwd` and the `/etc/postfix/sasl/sasl_passwd.db` files created in the previous steps contain your SMTP credentials in plain text.

To restrict access to these files, change their permissions so that only the **root** user can read from or write to the file. Run the following commands to change the ownership to root and update the permissions for the two files:

```
$  sudo chown root:root /etc/postfix/sasl/sasl_passwd /etc/postfix/sasl/sasl_passwd.db
$  sudo chmod 0600 /etc/postfix/sasl/sasl_passwd /etc/postfix/sasl/sasl_passwd.db
```

STEP 7: Configure the Postfix Relay Server

In this section, you will configure the `/etc/postfix/main.cf` file to use Gmail’s SMTP server.

1. Configure the postfix information:

   ```
   $ sudo dpkg-reconfigure postfix
   ```

   Set hostname to `smtp.gmail.com`
   Set postmaster to `username@gmail.com` (replace `username` with the `username` used above)
   Leave everything else on defaults.

2. Find and modify `relayhost` in `/etc/postfix/main.cf` to match the following example. Be sure the port number matches what you specified in `/etc/postfix/sasl/sasl_passwd` above.

   /etc/postfix/main.cf

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

   At the end of the file, add the following parameters to enable authentication:

   /etc/postfix/main.cf

   ```
   # Enable SASL authentication
   smtp_sasl_auth_enable = yes
   # Disallow methods that allow anonymous authentication
   smtp_sasl_security_options = noanonymous
   # Location of sasl_passwd
   smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd
   # Enable STARTTLS encryption
   smtp_tls_security_level = encrypt
   # Location of CA certificates
   smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
   ```

   Save your changes and close the file.

3. Restart Postfix:

   ```
   $ sudo systemctl restart postfix
   ```

STEP 8: Send email

Here are two examples of how to send email. See the `mutt manual` for more information.

1. The first contains only a subject and body:

```
$  echo "Hello world" | mutt -s "PI2: $(date)" recipient@gmail.com -y
```

2. The second is an empty body, a subject, and a file attachment:

```
$  echo "" | mutt -s "PI2: $(date)" recipient@gmail.com -y -a attachment.csv
```

STEP 9: Optional. Change sender email and name. Create a **.muttrc** file in the home folder.

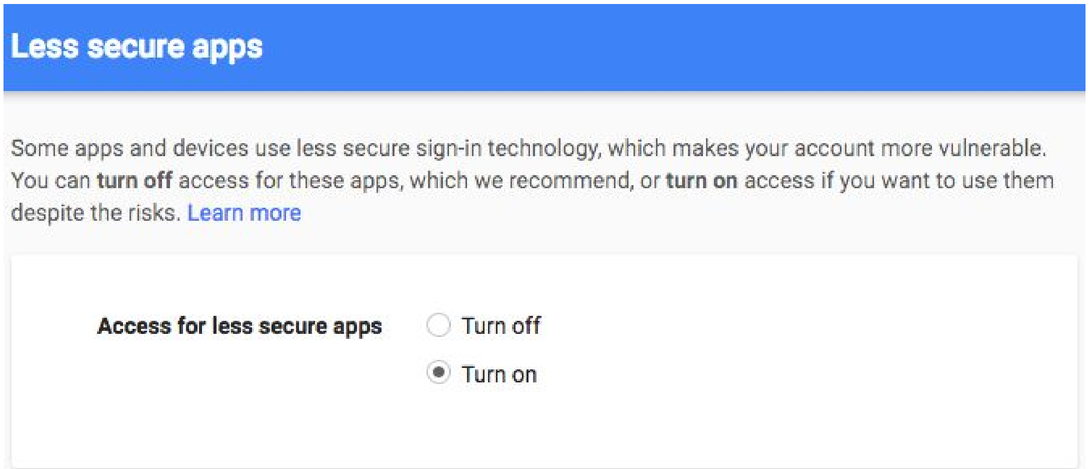
```
$  echo -e "set from = \"aname@something.net\"\nset realname = \"Yay\"" > ~/.muttrc
```

> **Note**
> This may cause GMail to mark messages from the pi as spam.

TROUBLESHOOTING: Enable "Less secure apps" access

In some cases, Gmail might still block connections from what it calls "Less secure apps." To enable access:

1. Enable "Less secure apps" access
   Select **Turn on**. A yellow "Updated" notice will appear at the top of the browser window and Gmail will automatically send a confirmation email.



Less secure apps

Some apps and devices use less secure sign-in technology, which makes your account more vulnerable. You can **turn off** access for these apps, which we recommend, or **turn on** access if you want to use them despite the risks. Learn more

Access for less secure apps ○ Turn off ● Turn on

2. If your test emails don't appear after a few minutes, disable captcha from new application login attempts and click **Continue**.