

Configure Postfix to Send Mail Using Gmail and Google Apps on Debian or Ubuntu

Postfix is a Mail Transfer Agent (MTA) that can act as an SMTP server or client to send or receive email. There are many reasons why you would want to configure Postfix to send email using Google Apps and Gmail. One reason is to avoid getting your mail flagged as spam if your current server's IP has been added to a blacklist.

In this guide, you will learn how to install and configure a Postfix server on Debian or Ubuntu to send email through Gmail and Google Apps. For information on configuring Postfix with other external SMTP servers, see our [Configure Postfix to Send Mail Using an External SMTP Server](#) guide.

Before You Begin

1. Complete our [Getting Started](#) and [Securing Your Server](#) guides and ensure that the Linode's [hostname is set](#).

2. Update your system:

3.

```
sudo apt-get update && sudo apt-get upgrade
```

4. Use your web browser to confirm your email login credentials by logging in to [Gmail](#).

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:

2.

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

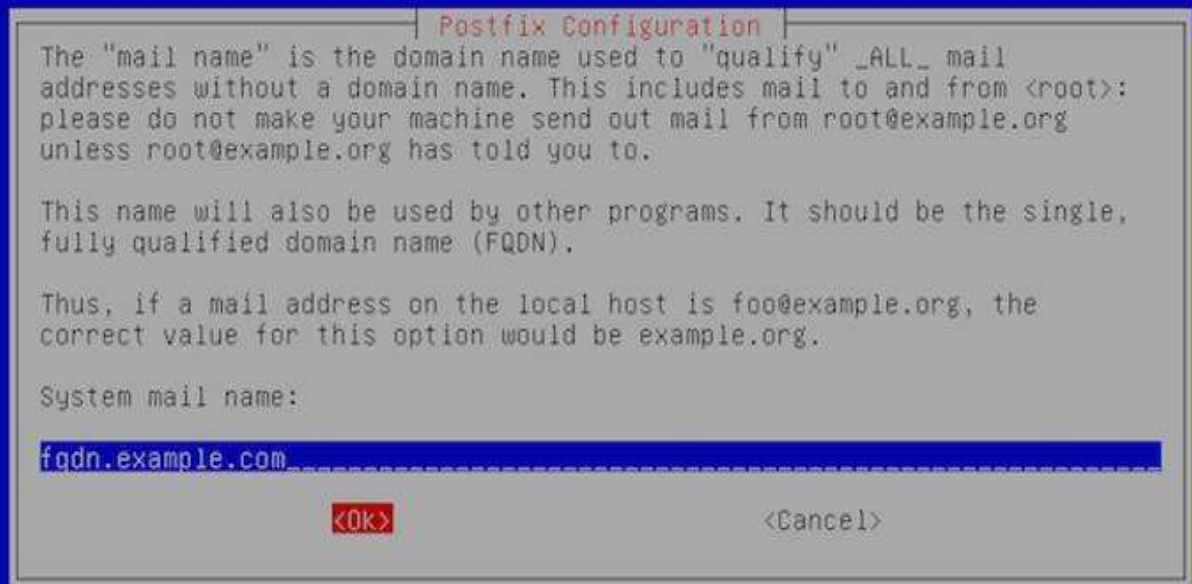
3. During the Postfix installation, a prompt will appear asking for your **General type of mail configuration**. Select **Internet Site**:

Package configuration



4. Enter the fully qualified name of your domain. In this example, **fqdn.example.com**:

Package configuration



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

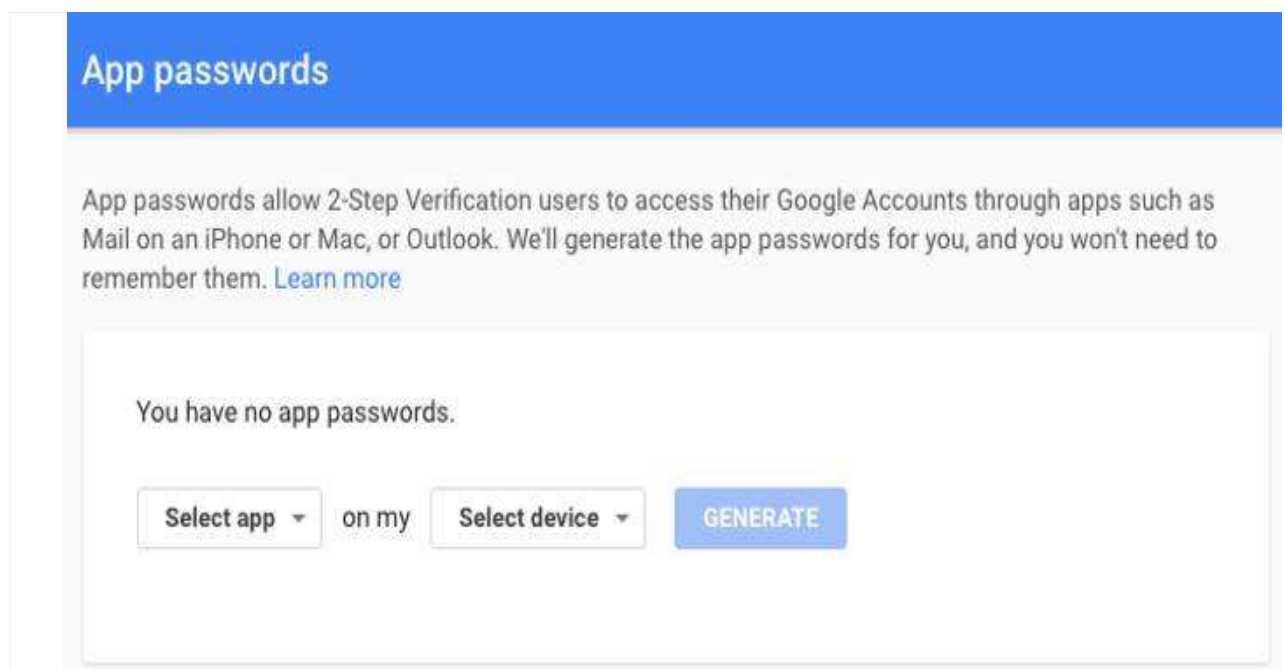
File: `/etc/postfix/main.cf`

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

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:



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**:

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

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
```

Configure the Postfix Relay Server

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

1. 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.

File: `/etc/postfix/main.cf`

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

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

File: `/etc/postfix/main.cf`

```
1 # Enable SASL authentication
2 smtp_sasl_auth_enable = yes
3 # Disallow methods that allow anonymous authentication
4 smtp_sasl_security_options = noanonymous
5 # Location of sasl_passwd
6 smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd
7 # Enable STARTTLS encryption
8 smtp_tls_security_level = encrypt
9 # Location of CA certificates
10 smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
```

3. Save your changes and close the file.

4. Restart Postfix:

```
5. sudo systemctl restart postfix
```

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.

![Enable “Less Secure Apps”](postfix-gmail-less-secure-apps.png “Enable “Less Secure Apps””)

2. Test Postfix as shown in the following section. If your test emails don’t appear after a few minutes, [disable captcha from new application login attempts](#) and click **Continue**.

Test Postfix

Use Postfix’s sendmail implementation to send a test email. Enter lines similar to those shown below, and note that there is no prompt between lines until the `.` ends the process:

```
sendmail recipient@elsewhere.com
From: you@example.com
Subject: Test mail
This is a test email
.
```

Check the destination email account for the test email. Open `syslog` using the `tail -f` command to show changes as they appear live:

```
sudo tail -f /var/log/syslog
```

CTRL + C to exit the log.