

# How to Install iRedMail on Ubuntu 24.04

Learn how to install iRedMail on Ubuntu 24.04 with our step-by-step guide. Set up a complete, secure email server solution quickly and efficiently.

---

# Contents

|    |                            |    |
|----|----------------------------|----|
| 01 | Introduction               | 3  |
| 02 | Prerequisites              | 3  |
| 03 | Install iRedMail           | 3  |
| 04 | iRedMail Post Installation | 6  |
| 05 | Test the Mail Server       | 11 |
| 06 | Conclusion                 | 17 |

# Introduction

iRedMail is a free, open-source mail server solution designed to enable the deployment of a fully functional email system. It integrates multiple components for mail transfer, retrieval, storage, spam filtering, virus scanning, spam detection, and secure communication. In addition, iRedMail provides a web administration interface to manage email accounts, domains, and server settings.

This article explains how to install the iRedMail server on Ubuntu 24.04.

## Prerequisites

Before you begin:

- [Deploy an Ubuntu 24.04](#) instance on Vultr with at least 4 GB RAM.
- Set up at least 2 [domain DNS records pointing to the instance's IP address](#). For example, `mail.example.com` and `example.com`.
- Access the server [using SSH](#) as a non-root user with sudo privileges.
- [Open a new support ticket](#) and request the SMTP port 25 to be unblocked on your instance.
- [Update the server](#).

## Install iRedMail

Follow the steps below to configure the hostname for your server, install iRedMail, and configure it.

1. Open the terminal and run the following command to set the hostname.

CONSOLE

```
$ sudo hostnamectl set-hostname mail.example.com
```

2. Open the `/etc/hosts` file.

CONSOLE

```
$ sudo nano /etc/hosts
```

3. Update the `127.0.1.1` and `127.0.0.1` entries to include your mail domain to enable correct name resolution on your server.

INI

```
# Part of file: /etc/hosts  
127.0.1.1 mail.example.com mail
```

4. Verify the active server hostname.

CONSOLE

```
$ hostname -f
```

Output:

```
mail.example.com
```

5. Download the latest iRedMail release file.

CONSOLE

```
$ wget https://github.com/iredmail/iRedMail/archive/  
1.7.0.tar.gz
```

The above command downloads the iRedMail version `1.7.1` release file. Visit the [official download page](#) to verify the latest version to install on your server.

6. Extract files from the `tar` archive.

```
CONSOLE
```

```
$ tar -xvf 1.7.0.tar.gz
```

7. Switch to the extracted iRedMail files directory.

```
CONSOLE
```

```
$ cd iRedMail-1.7.0
```

8. Run the iRedMail installation script.

```
CONSOLE
```

```
$ sudo bash iRedMail.sh
```

9. Press Enter when prompted to start the installation wizard. Then, reply each of the following prompts to install iRedMail on your server.

- **Default mail storage path:** Press Enter to use the default `/var/vmail` directory.
- **Preferred web server:** Keep `Nginx` selected and press Enter to enable the web server.
- **Choose Preferred backend uses to store mail accounts:** Press the Down arrow key and press Space to select MariaDB as the database backend.
- **LDAP suffix:** Press Enter to use the default domain format to create a directory tree.
- **Password for MySQL administrator:** Enter a new strong password to assign the `root` database user account on your server.
- **Your first domain name:** Enter your domain to use when sending and receiving emails on the server. For example, `example.com`

- **Password for the mail domain administrator:** Enter a new strong password to assign the `postmaster` mail administrator account.
- **Optional components:** Select optional components to install on your server and press Enter to apply changes.
- Enter Y when prompted to continue and save your iRedMail installation options.

```
< Question > Continue? [y|N]
```

- Enter Y to allow SSH connections on the default port `22` through the firewall.

```
< Question > Restart firewall now (with ssh ports: 22)? [y|N]
```

10. Reboot the server to apply the iRedMail installation changes.

```
CONSOLE
```

```
$ sudo systemctl reboot
```

## iRedMail Post Installation

iRedMail requires valid MX, SPF, DKIM and DMARC records on your mail domain to send and receive emails on your server. Log in to your domain registrar and set the following DNS records to enable iRedMail to send and deliver emails.

1. Create a new MX (Mail Exchange) record with the following details.

- Type: `MX`
- Name: `@`
- Priority: `10`
- Value: `mail.example.com`

2. Create a new SPF (Sender Policy Framework) record with the following details. Replace `192.0.2.1` with your actual server IP address.

- Type: `TXT`
- Name: `@`
- Value: `"v=spf1 a mx ip4:192.0.2.1 -all"`

3. Create a new DKIM (DomainKeys Identified Mail) record:

- Run the following command in your server's terminal to view your domain's DKIM records generated by the `amavisd` iRedMail utility.

CONSOLE

```
$ amavisd showkeys
```

Your output should be similar to the one below.

```
; key#1 2048 bits, s=dkim, d=example.com, /var/lib/dkim/example.com.pem
dkim._domainkey.example.com. 3600 TXT (
"v=DKIM1; p="
"MIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEAlzj9jaznT73JQMGC5ag"
"iSj0EEYqMQqSHRAXMHUm05ZDzy5ypzLdLYech8R+T+5S7+We0tQ0tWMAj07rB4/U"
"N6BX+J8ycp5Tcsg34lxEuYh34DENoQ8n0x4YLldr0/4+L5xk5pjJpsjiG6JigLPv"
"VDDzQSB+PA8Sn8NNczxRVhTFZ4AQgma7p9AnraTV+d4zjD24RD2pAvHL45BjH6NS"
"A4B3V3NfTf+0VbecI5dD29RsK8DIzhGW/XGsoo7G68fGwBpZdYkww3EhjpIojpgbn"
"uM7Je07rIDbczehroujPCxn52c+AuUC+pnfJuBBMXoeQ0jqabsLfmll0+KPeSuFR"
"EwIDAQAB")
```

Copy all contents in bracelets `()` to use as your DKIM record's data.

- Create a new `TXT` record using the generated DKIM information with the following details.
  - Type: `TXT`
  - Name: `dkim._domainkey`
  - Value: `"<ioredmail-dkim-data>"`
- Test your DKIM keys configuration using `amavisd` to verify the DNS record.

## CONSOLE

```
$ amavisd testkeys
```

Output:

```
TESTING#1 example.com: dkim._domainkey.example.com => pass
```

4. Create a new DMARC (Domain-based Message Authentication, Reporting, and Conformance) record with the following details to send aggregate reports to your administrative email `dmarc-reports@example.com`.
  - Type: `TXT`
  - Name: `_dmarc`
  - Value: `"v=DMARC1; p=reject; rua=mailto:dmarc-reports@example.com; pct=100"`

## Configure iRedMail to Use Trusted Let's Encrypt SSL Certificates

iRedMail uses a self-signed SSL certificate by default. Follow the steps below to configure iRedMail to use trusted Let's Encrypt SSL certificates on your server.

1. Install the Certbot Let's Encrypt client tool.

## CONSOLE

```
$ sudo snap install certbot --classic
```

2. Generate a new Let's Encrypt SSL certificate for your mail domain. Replace `mail.example.com` and `hello@example.com` with your actual details.

## CONSOLE

```
$ sudo certbot certonly --webroot -w /var/www/html -d mail.example.com -m hello@example.com --agree-tos
```

3. Open the default iRedMail virtual host configuration for Nginx.

## CONSOLE

```
$ sudo nano /etc/nginx/sites-available/00-default-ssl.conf
```

4. Add the following directives to the file after `server_name _;`. Replace `/etc/letsencrypt/live/mail.example.com/` with your actual Let's Encrypt directory.

## INI

```
ssl_certificate /etc/letsencrypt/live/mail.example.com/
fullchain.pem;

ssl_certificate_key /etc/letsencrypt/live/mail.example.com/
privkey.pem;
```

Save and close the file.

The above configuration directives enable Nginx to use your domain's trusted Let's Encrypt certificates to deliver requests on your server.

5. Open the `/etc/nginx/templates/ssl.tpl` file.

## CONSOLE

```
$ sudo nano /etc/nginx/templates/ssl.tpl
```

6. Find the following configuration directives.

## INI

```
ssl_certificate /etc/ssl/certs/iRedMail.crt;
ssl_certificate_key /etc/ssl/private/iRedMail.key;
```

7. Replace the SSL certificate and private key paths with your actual Let's Encrypt details.

## INI

```
ssl_certificate /etc/letsencrypt/live/mail.example.com/
fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/mail.example.com/
privkey.pem;
```

Save and close the file.

- Open the main `Postfix` configuration file `/etc/postfix/main.cf`.

CONSOLE

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

- Find the following TLS settings section.

INI

```
# SSL key, certificate, CA
#
smtpd_tls_key_file = /etc/ssl/private/iRedMail.key
smtpd_tls_cert_file = /etc/ssl/certs/iRedMail.crt
smtpd_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
smtpd_tls_CAspath = /etc/ssl/certs
```

- Replace the `smtpd_tls_key_file` and `smtpd_tls_cert_file` with your actual Let's Encrypt certificate details.

INI

```
smtpd_tls_key_file = /etc/letsencrypt/live/mail.example.com/
privkey.pem
smtpd_tls_cert_file = /etc/letsencrypt/live/mail.example.com/
fullchain.pem
#smtpd_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
#smtpd_tls_CAspath = /etc/ssl/certs
```

- Open the Dovecot configuration file `/etc/dovecot/dovecot.conf`.

CONSOLE

```
$ sudo nano /etc/dovecot/dovecot.conf
```

12. Find the following SSL configuration section.

INI

```
ssl_cert = </etc/ssl/certs/iRedMail.crt  
ssl_key = </etc/ssl/private/iRedMail.key
```

13. Replace the `ssl_cert` and `ssl_key` values with your actual Let's Encrypt details.

INI

```
ssl_cert = </etc/letsencrypt/live/mail.example.com/  
fullchain.pem  
ssl_key = </etc/letsencrypt/live/mail.example.com/privkey.pem
```

Save and close the file.

14. Restart Nginx, Postfix and Dovecot to apply your configuration changes.

CONSOLE

```
$ sudo systemctl restart nginx postfix dovecot
```

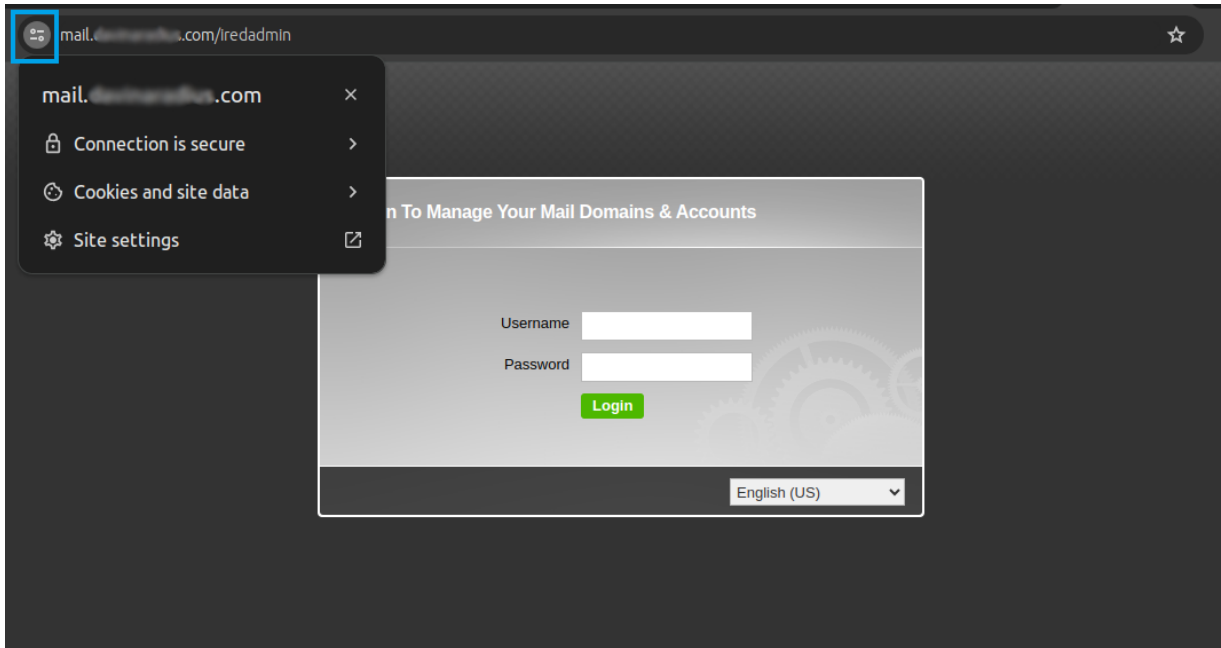
## Test the Mail Server

Follow the steps below to test the iRedMail server functionalities.

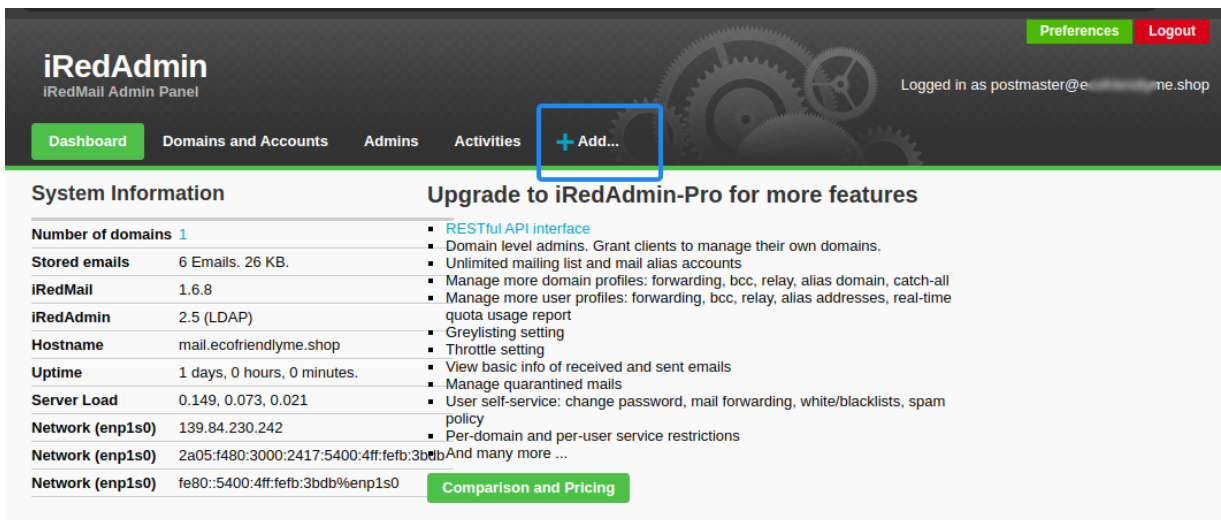
1. Access iRedMail Administration Console (iRedAdmin) using your domain in a web browser such as Chrome.

```
https://mail.example.com/iredadmin
```

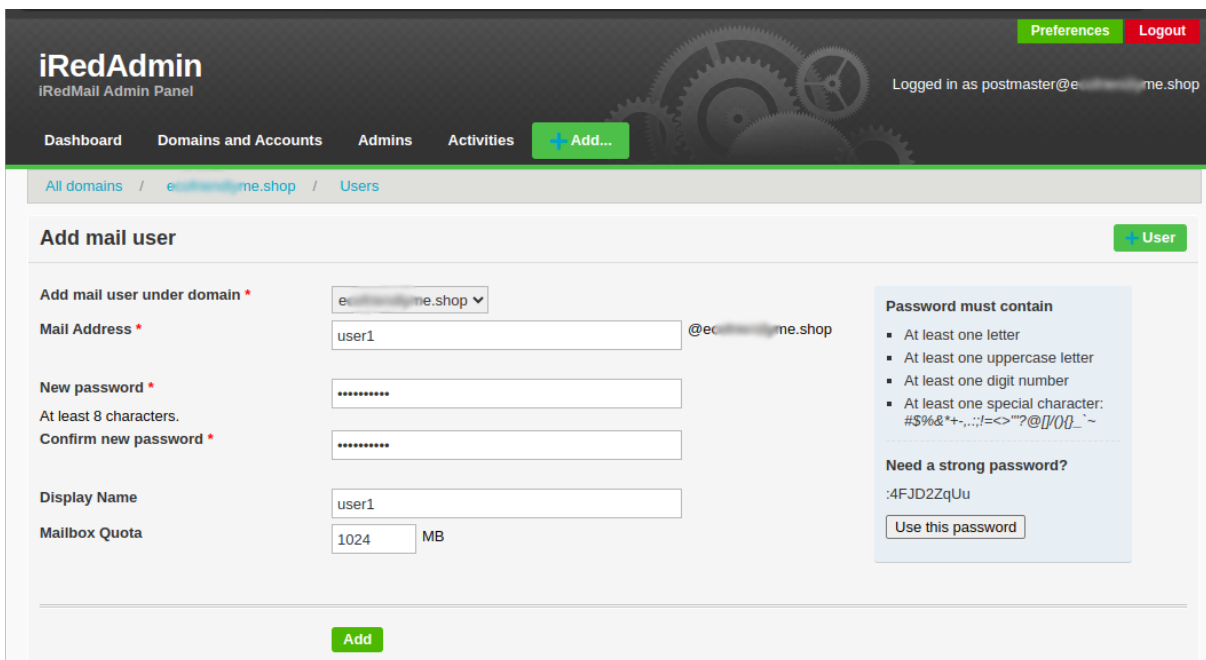
Log in to the iRedMail web administration dashboard using the `postmaster` administrator email and password you set during the installation process.



2. Click the **Add** drop-down and select **User** from the list to set up a new email account.



3. Enter your desired email information and click **Add** to create the new user.



The screenshot shows the iRedAdmin web interface. At the top, there's a navigation bar with 'Dashboard', 'Domains and Accounts', 'Admins', 'Activities', and '+ Add...'. The user is logged in as 'postmaster@e...me.shop'. The main content area is titled 'Add mail user' and contains the following fields:

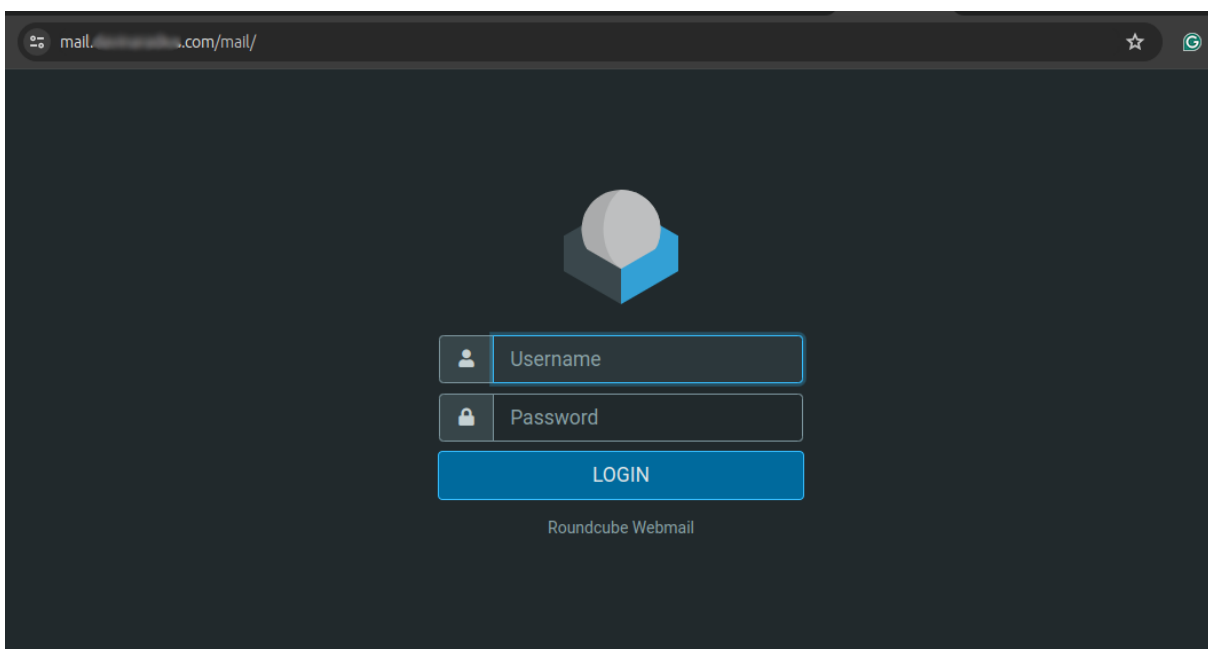
- Add mail user under domain \***: A dropdown menu showing 'e...me.shop'.
- Mail Address \***: A text input field containing 'user1', followed by '@e...me.shop'.
- New password \***: A text input field with masked characters (dots).
- Confirm new password \***: A text input field with masked characters (dots).
- Display Name**: A text input field containing 'user1'.
- Mailbox Quota**: A text input field containing '1024' and a unit dropdown set to 'MB'.

On the right side, there's a 'Password must contain' box with the following requirements:

- At least one letter
- At least one uppercase letter
- At least one digit number
- At least one special character: # \$ % & \* + , . : ; ! = < > " ? @ [ \ / { } \_ ' ~

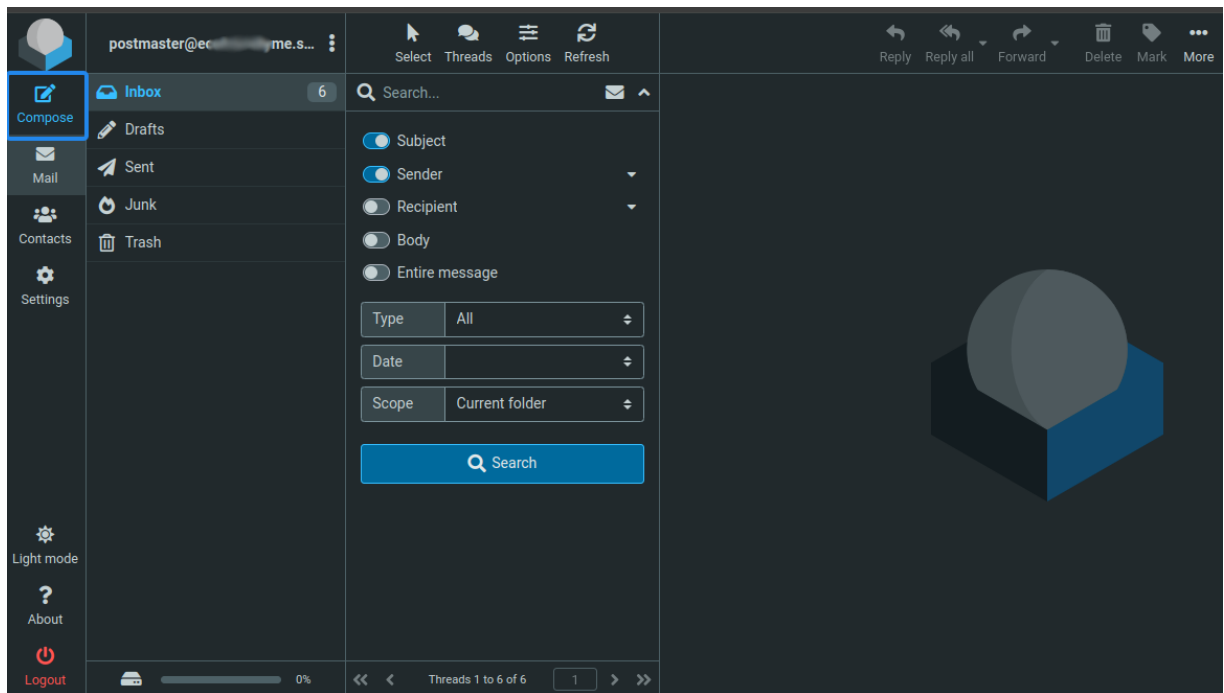
Below this, there's a 'Need a strong password?' box showing a generated password ':4FJD2ZqUu' and a 'Use this password' button. At the bottom of the form is an 'Add' button.

4. Add the `/mail` path to your domain in new web browser window to access the Roundcube Webmail interface and manage your email accounts.

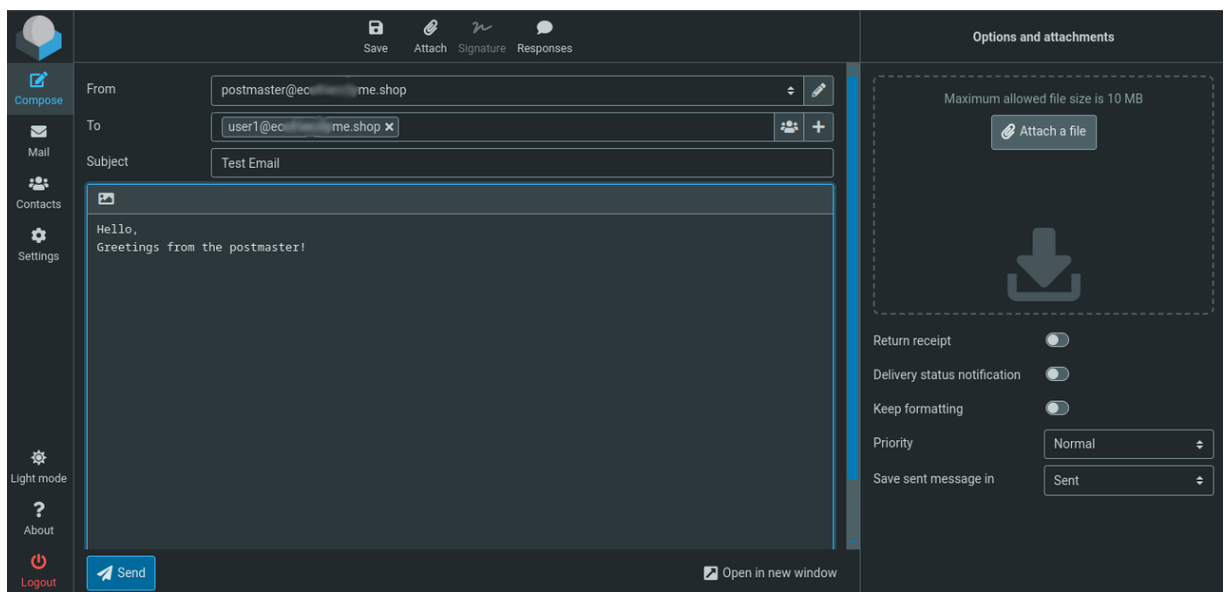


Enter your Postmaster user and password you set to log in to its web mail interface.

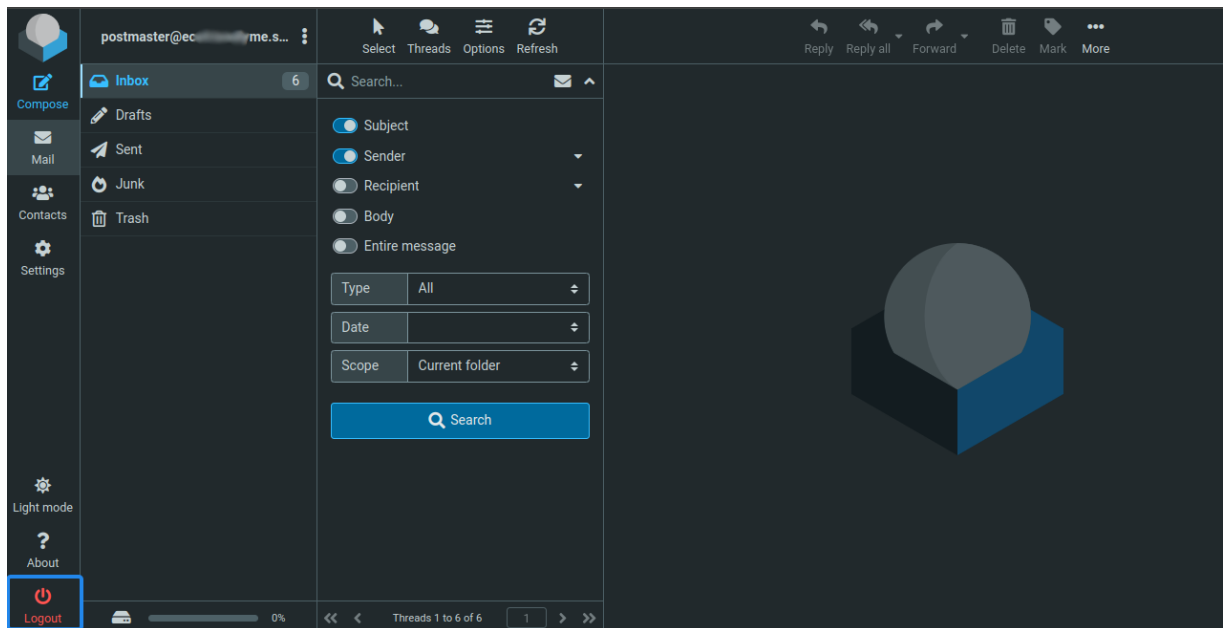
5. Click **Compose** to create a new test email to your new user's email. For example, `user1@example.com`.



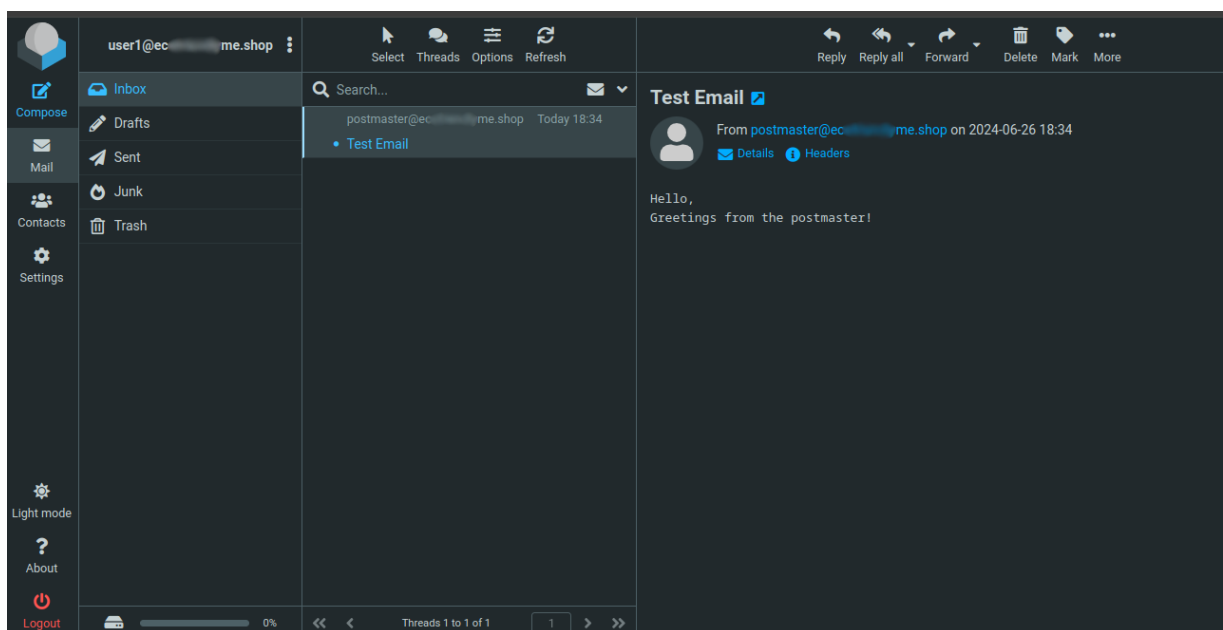
6. Click **Send** to send the test email and verify that it's successful.



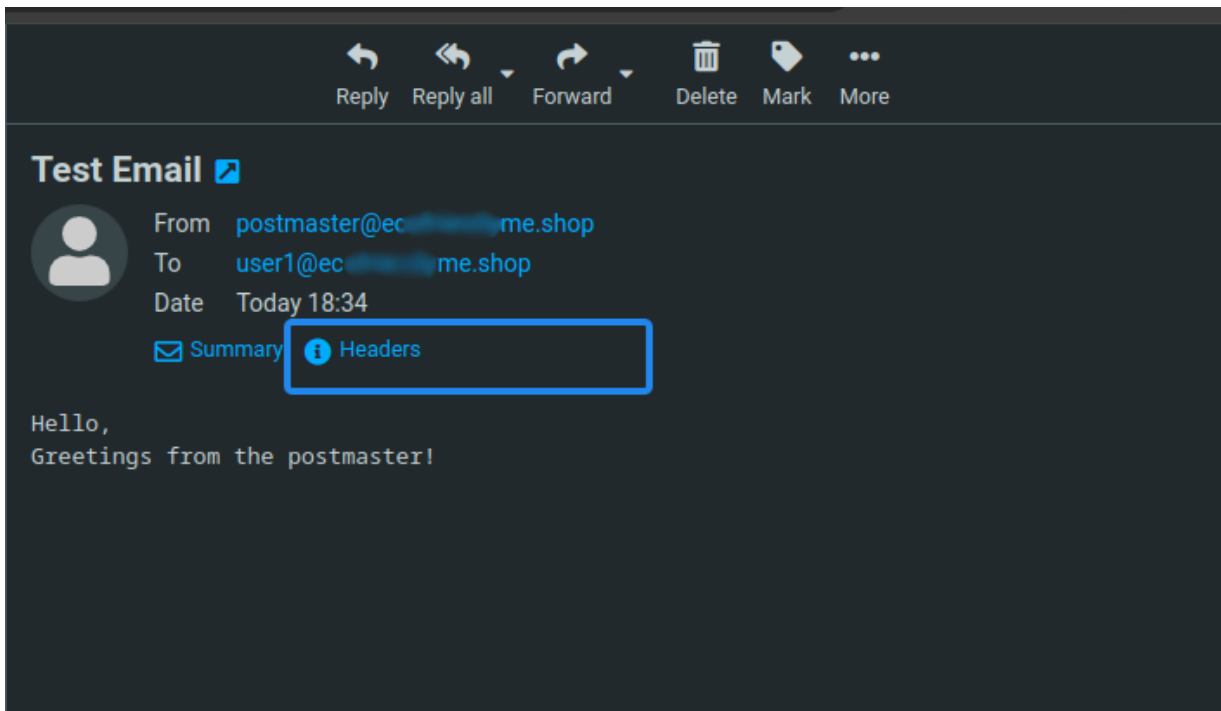
7. Click **Logout** in the bottom left to exit the Roundcube Webmail interface.



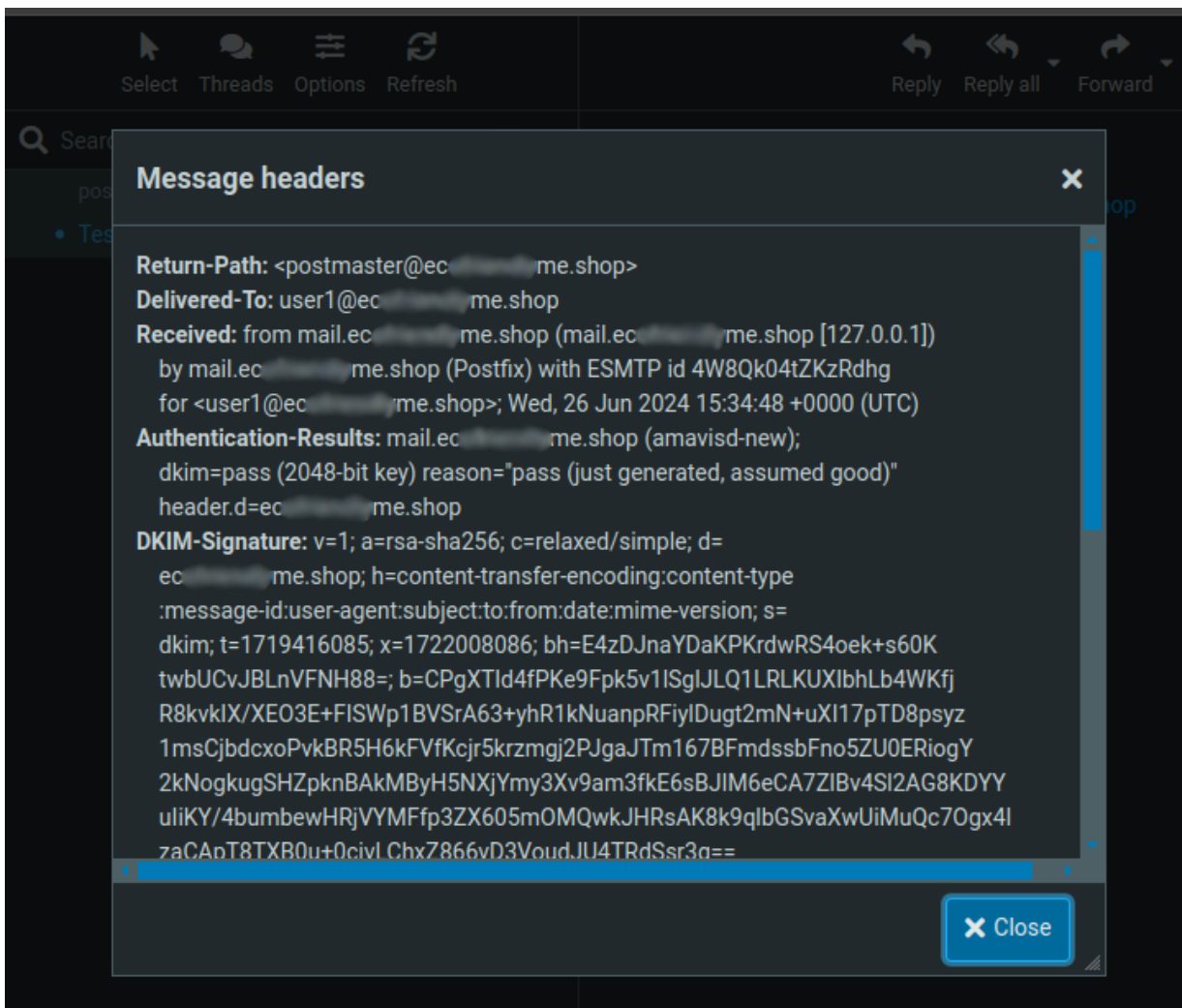
8. Log in to Roundcube using the new email user you created earlier to access the email sent from the `postmaster` account.



9. Click the `header` link within **Headers**.



10. Verify your domain's DKIM values sent to all mail domains.



## Conclusion

---

You have installed iRedMail on Ubuntu 24.04 server and enabled secure access to email accounts using valid SPF, DKIM, and DMARC records. iRedMail supports multiple email domains and enables you to set up a secure mail server you can use to send and receive emails. For more information, visit the [iRedMail documentation](#).



VULTR

