Web server management

- Apache setup & Virtual servers
- SQL
- SSL & Certificate management
- PostgreSQL
- SQL Reference
- Mail Server

Apache setup & Virtual servers

Initial setup

Install packages `sudo apt install apache2 mariadb-server libapache2-mod-php php-mysql -y`

Open ports at `cd /etc/apache2/ports.conf` and add the line `Listen NEW_PORT_NUMBER`

Creating virtual servers

Create a file named NAME.conf and add something like this

```
<VirtualHost *:80>
        ServerName pdb.conorbriggs.com.au
        DocumentRoot /mnt/drives/10tb/10tb/web_servers/pdb/

<Directory /mnt/drives/10tb/10tb/web_servers/pdb/>
        Options -Indexes -FollowSymLinks
        AllowOverride All
        Require all granted
</Directory>
ErrorLog /var/log/apache2/pdb-error.log
CustomLog /var/log/apache2/pdb-access.log combined

RewriteEngine on
</VirtualHost>
```

Enable the site & restart apache with `sudo a2ensite site.conf && systemctl restart apache2`

SSL Certificate management

Install certbot and run setup

```
sudo apt install certbot python3-certbot-apache
a2enmod ssl
sudo certbot --apache
```

List all certificates with `certbot certificates`
Delete a certificaite `certbot delete --cert-name CERT_NAME_HERE`

Managing the apache2 service

Start, stop & restart `sudo service apache2 start/restart/stop`

Enable a config `sudo a2ensite SITE_CONF_FILE.conf`
Disable a config `sudo a2dissite SITE_CONF_FILE.conf`

SQL

After install

ensure secure access `sudo mysql_secure_installation`

Logging in

`mysql -u USERNAME -p`

add `-h HOSTNAME IP` when logging in remotely

Creating a user

`CREATE USER 'NEW_USERNAME'@'localhost' IDENTIFIED BY 'NEW_PASSWORD';`

Creating a database

`CREATE DATABASE DATABASE_NAME`

Allow user to modify Database

`GRANT ALL PRIVILEGES ON DATABASE_NAME.* TO 'USERNAME'@'localhost';`

Update permissions

`FLUSH PRIVILEGES;`

Open Database to the world

1. Open file at `/etc/mysql/maridadb.conf.d/50-server.cnf`
2. Edit line `bind-address = 127.0.0.1` to -> `bind-address = 0.0.0.0`
3. Restart db service `systemctl restart mariadb`
4. Allow port through firewall with `ufw allow 3306`

Automated backup script

```
#!/bin/bash

# === Config ===
BACKUP_DIR="/var/backups/mysql"
DATE=$(date +"%Y-%m-%d_%H-%M")
DB_USER="backupuser"
DB_PASS="yourpassword"

# Create backup dir if not exist
mkdir -p "$BACKUP_DIR"

# Dump all databases
mysqldump -u "$DB_USER" -p"$DB_PASS" --all-databases --single-transaction > "$BACKUP_DIR/mariadb-$DATE.sql"

# Optional: Compress the backup
gzip "$BACKUP_DIR/mariadb-$DATE.sql"

# Delete backups older than 7 days
find "$BACKUP_DIR" -name "*.sql.gz" -type f -mtime +7 -delete
```

SSL & Certificate management

Install certbot and run setup

```
sudo apt install certbot python3-certbot-apache
a2enmod ssl
sudo certbot --apache
```

List all certificates

```
certbot certificates
```

Delete a certificaite

```
certbot delete --cert-name CERT_NAME_HERE
```

PostgreSQL

Install

Install postgresql and postgis for coordinates data

```
sudo apt install postgresql postgis -y
```

Login

```
sudo -u postgres psql
```

Change password to default account

```
ALTER USER postgres WITH PASSWORD 'new password';
```

Restrict remote access to localhost only

Open the config file

```
nano /etc/postgresql/15/main/postgresql.conf
```

uncomment this line

```
listen_addresses = 'localhost'
```

if you need remote access

```
listen_addresses = '0.0.0.0'
```

and Edit `/etc/postgresql/15/main/pg_hba.conf` and add a line like:

```
host    all             all             192.168.1.0/24          scram-sha-256
```

SQL Reference

Create a table that auto deletes on other table deletion

```
CREATE TABLE tradie_postcode_covered (
    tradie_id INT,
    postcode VARCHAR(10),
    PRIMARY KEY (tradie_id, postcode), -- primary key ensures no tradie can be in the same postcode twice
    FOREIGN KEY (tradie_id) REFERENCES tradies(id) ON DELETE CASCADE
);
```

Mail Server

Initial setup

1 - Updates

```
sudo apt update && sudo apt upgrade -y
```

2 - Install docker and docker compose

```
# Install required packages
sudo apt install -y apt-transport-https ca-certificates curl gnupg lsb-release

# Add Docker's official GPG key
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

# Set up the stable repository
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Install Docker
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin

# Add your user to docker group
sudo usermod -aG docker $USER
```

3 - Create DNS Records

- **A record**: `mail.yourdomain.com` → your server IP

- **MX record**: `@` → `mail.yourdomain.com` (priority 10)
- **TXT record (SPF)**: `v=spf1 mx ~all`

4 - Install and configure mailserver

```
# Create directory for mail server
mkdir -p ~/mailserver
cd ~/mailserver

# Download docker-compose.yml and .env template
wget https://raw.githubusercontent.com/docker-mailserver/docker-mailserver/master/compose.yaml
wget https://raw.githubusercontent.com/docker-mailserver/docker-mailserver/master/mailserver.env

# Rename for easier use
mv compose.yaml docker-compose.yml
mv mailserver.env .env
```

5 - Edit .env variables

- `HOSTNAME=mail`
- `DOMAINNAME=yourdomain.com`
- `OVERRIDE_HOSTNAME=mail.yourdomain.com`
- `ENABLE_SPAMASSASSIN=1`
- `ENABLE_CLAMAV=1`
- `ENABLE_FAIL2BAN=1`
- `SSL_TYPE=letsencrypt` (or `manual` if you have your own certs)
- `ACCOUNT_PROVISIONER=FILE`

6 - Edit docker-compose.yml

```
services:
  mailserver:
    image: ghcr.io/docker-mailserver/docker-mailserver:latest
    container_name: mailserver
    # Provide the FQDN of your mail server here (Your DNS MX record should point to this value)
    hostname: mx.home.conorbriggs.com.au
```

```yaml
    env_file: .env
    # More information about the mail-server ports:
    # https://docker-mailserver.github.io/docker-mailserver/latest/config/security/understanding-the-ports/
    ports:
      - "25:25"    # SMTP  (explicit TLS => STARTTLS, Authentication is DISABLED => use port 465/587 instead)
      - "143:143"  # IMAP4 (explicit TLS => STARTTLS)
      - "465:465"  # ESMTP (implicit TLS)
      - "587:587"  # ESMTP (explicit TLS => STARTTLS)
      - "993:993"  # IMAP4 (implicit TLS)
    volumes:
      - ./mail-data/:/var/mail/
      - ./mail-state/:/var/mail-state/
      - ./mail-logs/:/var/log/mail/
      - ./config/:/tmp/docker-mailserver/
      - /etc/localtime:/etc/localtime:ro
      - /etc/letsencrypt:/etc/letsencrypt:ro
    restart: always
    stop_grace_period: 1m
    # Uncomment if using `ENABLE_FAIL2BAN=1`:
    cap_add:
      - NET_ADMIN
    healthcheck:
      test: "ss --listening --ipv4 --tcp | grep --silent ':smtp' || exit 1"
      timeout: 3s
      retries: 0
    networks:
      - mailserver-network
networks:
  mailserver-network:
    driver: bridge
```

Connecting

IMAP (Incoming):

- Server: `mx.home.conorbriggs.com.au`

- Port: `993` (IMAPS with SSL/TLS)
- Security: SSL/TLS
- Authentication: Normal password

SMTP (Outgoing):

- Server: `mx.home.conorbriggs.com.au`
- Port: `587` (with STARTTLS) or `465` (with SSL/TLS)
- Security: STARTTLS (port 587) or SSL/TLS (port 465)
- Authentication: Normal password

Container Management

Basic Container Operations

Start the mailserver

```
docker compose up -d
```

Stop the mailserver

```
docker compose down
```

Restart the mailserver

```
docker compose restart
```

View logs (live)

```
docker compose logs -f
```

View logs (last 100 lines)

```
docker compose logs --tail=100
```

Check container status

```
docker ps -a
```

Check container health

```
docker inspect mailserver | grep -A 10 Health
```

Email Account Management

Create Email Accounts

Create a new email account (will prompt for password)

```
docker exec -it mailserver setup email add user@home.conorbriggs.com.au
```

Create account with password in command

```
docker exec -it mailserver setup email add user@home.conorbriggs.com.au password123
```

Create account with quota (e.g., 500MB)

```
docker exec -it mailserver setup email add user@home.conorbriggs.com.au password123 500M
```

List Email Accounts

List all email accounts

```
docker exec mailserver setup email list
```

View the accounts file directly

```
docker exec mailserver cat /tmp/docker-mailserver/postfix-accounts.cf
```

Update/Change Passwords

Update password for existing account

```
docker exec -it mailserver setup email update user@home.conorbriggs.com.au new_password
```

Change password (alternative method - will prompt)

```
docker exec -it mailserver setup email update user@home.conorbriggs.com.au
```

Delete Email Accounts

Delete an email account

```
docker exec -it mailserver setup email del user@home.conorbriggs.com.au
```

Delete account and remove mailbox data

```
docker exec mailserver setup email del user@home.conorbriggs.com.au
rm -rf ./mail-data/home.conorbriggs.com.au/user
```

Alias Management

Create Aliases

Create an alias (forward emails from alias to recipient)

```
docker exec mailserver setup alias add alias@home.conorbriggs.com.au recipient@home.conorbriggs.com.au
```

Create alias with multiple recipients

```
docker exec mailserver setup alias add sales@home.conorbriggs.com.au
"user1@home.conorbriggs.com.au,user2@home.conorbriggs.com.au"
```

List Aliases

List all aliases

```
docker exec mailserver setup alias list
```

View aliases file

```
docker exec mailserver cat /tmp/docker-mailserver/postfix-virtual.cf
```

Delete Aliases

Delete an alias

```
docker exec mailserver setup alias del alias@home.conorbriggs.com.au recipient@home.conorbriggs.com.au
```

Quota Management

Set Quotas

Set quota for a user (e.g., 1GB)

```
docker exec mailserver setup quota set user@home.conorbriggs.com.au 1G
```

Set unlimited quota

```
docker exec mailserver setup quota set user@home.conorbriggs.com.au 0
```

Check Quotas

Check quota for specific user

```
docker exec mailserver setup quota get user@home.conorbriggs.com.au
```

List all quotas

```
docker exec mailserver setup quota list
```

Check quota usage

```
docker exec mailserver doveadm quota get -u user@home.conorbriggs.com.au
```

Delete Quotas

Remove quota (sets to default)

```
docker exec mailserver setup quota del user@home.conorbriggs.com.au
```

DKIM (Email Signing)

Generate DKIM Keys

Generate DKIM key for domain

```
docker exec mailserver setup config dkim
```

Generate for specific domain

```
docker exec mailserver setup config dkim domain home.conorbriggs.com.au
```

Generate with custom key size

```
docker exec mailserver setup config dkim keysize 2048
```

View DKIM Public Key

Show DKIM DNS record

```
docker exec mailserver setup config dkim help
```

View the public key directly

```
docker exec mailserver cat /tmp/docker-mailserver/opendkim/keys/home.conorbriggs.com.au/mail.txt
```

Fail2Ban (Security)

Fail2Ban Status

Check fail2ban status

```
docker exec mailserver setup fail2ban status
```

Check banned IPs

```
docker exec mailserver setup fail2ban
```

Unban an IP address

```
docker exec mailserver setup fail2ban unban <IP_ADDRESS>
```

Ban an IP address

```
docker exec mailserver setup fail2ban ban <IP_ADDRESS>
```

Debugging & Diagnostics

Service Status

Check all listening ports

```
docker exec mailserver ss -tlnp
```

Check specific service status

```
docker exec mailserver supervisorctl status
```

Check Postfix status

```
docker exec mailserver postfix status
```

Check Dovecot status

```
docker exec mailserver doveadm service status
```

Mail Queue

View mail queue

```
docker exec mailserver postqueue -p
```

Flush mail queue (retry sending)

```
docker exec mailserver postqueue -f
```

Delete all queued mail

```
docker exec mailserver postsuper -d ALL
```

Delete specific message from queue

```
docker exec mailserver postsuper -d <QUEUE_ID>
```

Logs

View mail logs

```
docker exec mailserver tail -f /var/log/mail/mail.log
```

View mail errors

```
docker exec mailserver tail -f /var/log/mail/mail.err
```

View specific log files

```
docker exec mailserver ls -la /var/log/mail/
```

Search logs for specific email

```
docker exec mailserver grep "user@domain.com" /var/log/mail/mail.log
```

Test Email Delivery

Test SMTP connection

```
docker exec mailserver nc -zv localhost 25
```

Send test email from command line

```
echo "Test email body" | docker exec -i mailserver sendmail test@home.conorbriggs.com.au
```

Test with swaks (if installed)

```
docker exec mailserver swaks --to user@home.conorbriggs.com.au --from test@home.conorbriggs.com.au
```

Connection Testing

Test IMAP connection

```
docker exec mailserver nc -zv localhost 143
docker exec mailserver nc -zv localhost 993
```

Test SMTP connection

```
docker exec mailserver nc -zv localhost 25
docker exec mailserver nc -zv localhost 587
docker exec mailserver nc -zv localhost 465
```

Check TLS/SSL certificates

```
docker exec mailserver openssl s_client -connect localhost:993 -showcerts
docker exec mailserver openssl s_client -connect localhost:465 -showcerts
```

Configuration Management

Reload Configuration

Reload postfix configuration

```
docker exec mailserver postfix reload
```

Reload dovecot configuration

```
docker exec mailserver doveadm reload
```

Restart all services

```
docker compose restart
```

View Configuration

View postfix configuration

```
docker exec mailserver postconf
```

View dovecot configuration

```
docker exec mailserver doveconf
```

View specific postfix setting

```
docker exec mailserver postconf | grep smtp_tls
```

Check all environment variables

```
docker exec mailserver env | grep -E '(SMTP|IMAP|SSL|TLS)'
```

Backup Configuration

Backup all mail data

```
tar -czf mailserver-backup-$(date +%Y%m%d).tar.gz ./mail-data ./mail-state ./config
```

Backup just configuration

```
tar -czf mailserver-config-$(date +%Y%m%d).tar.gz ./config
```

Backup specific user's mailbox

```
tar -czf user-backup-$(date +%Y%m%d).tar.gz ./mail-data/home.conorbriggs.com.au/user
```

Database/User Management

User Database

List all users in Dovecot

```
docker exec mailserver doveadm user '*'
```

Check if user exists

```
docker exec mailserver doveadm user user@home.conorbriggs.com.au
```

View user's mailbox location

```
docker exec mailserver doveadm mailbox status -u user@home.conorbriggs.com.au all '*'
```

Mailbox Management

List mailboxes for user

```
docker exec mailserver doveadm mailbox list -u user@home.conorbriggs.com.au
```

Create mailbox for user

```
docker exec mailserver doveadm mailbox create -u user@home.conorbriggs.com.au Folder.Name
```

Delete mailbox

```
docker exec mailserver doveadm mailbox delete -u user@home.conorbriggs.com.au Folder.Name
```

Rebuild mailbox index

```
docker exec mailserver doveadm force-resync -u user@home.conorbriggs.com.au INBOX
```

Performance & Monitoring

Check Resource Usage

Check container stats

```
docker stats mailserver
```

Check disk usage

```
docker exec mailserver df -h
```

Check memory usage

```
docker exec mailserver free -h
```

Check mail directory size

```
du -sh ./mail-data/*
```

Connection Monitoring

Show active connections

```
docker exec mailserver ss -tn | grep -E ':(25|587|465|143|993)'
```

Count connections by port

```
docker exec mailserver ss -tn | grep -E ':(25|587|465|143|993)' | wc -l
```

Show who's connected to IMAP

```
docker exec mailserver doveadm who
```

SSL/TLS Certificate Management

Check Certificates

Check SSL certificate expiry

```
docker exec mailserver openssl x509 -in /etc/letsencrypt/live/mx.home.conorbriggs.com.au/fullchain.pem -noout -dates
```

View certificate details

```
docker exec mailserver openssl x509 -in /etc/letsencrypt/live/mx.home.conorbriggs.com.au/fullchain.pem -noout -text
```

Test SSL/TLS for SMTP

```
openssl s_client -connect mx.home.conorbriggs.com.au:465 -showcerts
```

Test STARTTLS for SMTP

```
openssl s_client -connect mx.home.conorbriggs.com.au:587 -starttls smtp
```

Troubleshooting

Common Issues

Check if services are running

```
docker exec mailserver supervisorctl status
```

Restart specific service

```
docker exec mailserver supervisorctl restart postfix
docker exec mailserver supervisorctl restart dovecot
```

Check for permission issues

```
docker exec mailserver ls -la /var/mail/
docker exec mailserver ls -la /tmp/docker-mailserver/
```

Verify DNS records

```
dig mx home.conorbriggs.com.au
dig txt _dmarc.home.conorbriggs.com.au
dig txt mail._domainkey.home.conorbriggs.com.au
```

Test email authentication

```
docker exec mailserver opendkim-testkey -d home.conorbriggs.com.au -s mail
```

Reset and Clean Up

Remove all mail data (WARNING: deletes all emails)

```
docker compose down
rm -rf ./mail-data/*
rm -rf ./mail-state/*
docker compose up -d
```

Clear logs

```
docker exec mailserver truncate -s 0 /var/log/mail/mail.log
```

Rebuild entire container

```
docker compose down
docker compose pull
docker compose up -d --force-recreate
```

Quick Reference

Setup Script Help

Show all setup commands

```
docker exec mailserver setup help
```

Help for specific command

```
docker exec mailserver setup email help
docker exec mailserver setup alias help
docker exec mailserver setup config help
```

File Locations Inside Container

```
/tmp/docker-mailserver/          - Configuration files (mapped to ./config/)
/var/mail/                       - Mail data (mapped to ./mail-data/)
/var/mail-state/                 - State files (mapped to ./mail-state/)
/var/log/mail/                   - Mail logs (mapped to ./mail-logs/)
/etc/letsencrypt/                - SSL certificates (read-only)
/etc/postfix/                    - Postfix configuration
/etc/dovecot/                    - Dovecot configuration
```

Important Configuration Files

```
./config/postfix-accounts.cf      - Email accounts
./config/postfix-virtual.cf       - Aliases
./config/dovecot-quotas.cf        - User quotas
./config/opendkim/                - DKIM keys
```

Advanced Operations

Database Operations

Export all accounts

```
docker exec mailserver cat /tmp/docker-mailserver/postfix-accounts.cf > accounts-backup.txt
```

Import accounts

```
cat accounts-backup.txt | docker exec -i mailserver tee /tmp/docker-mailserver/postfix-accounts.cf
docker compose restart
```

Verify account database

```
docker exec mailserver postmap -q user@home.conorbriggs.com.au /tmp/docker-mailserver/postfix-accounts.cf
```

Custom Scripts

Run custom maintenance script

```
docker exec mailserver /bin/bash -c "your-script-here"
```

Execute interactive shell

```
docker exec -it mailserver /bin/bash
```

Monitoring & Alerts

Set Up Monitoring

Watch logs in real-time

```
docker compose logs -f --tail=100
```

Monitor for failed logins

```
docker exec mailserver tail -f /var/log/mail/mail.log | grep "authentication failed"
```

Monitor mail queue size

```
watch -n 60 'docker exec mailserver postqueue -p | tail -1'
```

Check for errors

```
docker exec mailserver grep -i error /var/log/mail/mail.log | tail -20
```

Common Workflows

Adding a New User

1. Create account

```
docker exec -it mailserver setup email add newuser@home.conorbriggs.com.au
```

2. Set quota (optional)

```
docker exec mailserver setup quota set newuser@home.conorbriggs.com.au 2G
```

3. Verify account created

```
docker exec mailserver setup email list
```

4. Test login: Use an email client to connect via IMAP (993) or SMTP (587)

Migrating Mail

1. Backup old server

```
tar -czf old-mailserver-backup.tar.gz ./mail-data
```

2. Copy to new server

```
scp old-mailserver-backup.tar.gz newserver:/path/to/mailserver/
```

3. Extract on new server

```
tar -xzf old-mailserver-backup.tar.gz
```

4. Fix permissions

```
chown -R 5000:5000 ./mail-data
```

5. Restart mailserver

```
docker compose restart
```

Security Hardening

1. Enable Fail2Ban (in .env file)

```
ENABLE_FAIL2BAN=1
```

2. Check banned IPs regularly

```
docker exec mailserver fail2ban-client status postfix-sasl
```

3. Monitor authentication attempts

```
docker exec mailserver grep "authentication failed" /var/log/mail/mail.log
```

4. Review SSL/TLS settings

```
docker exec mailserver postconf | grep tls
```

Tips & Best Practices

1. **Always backup before major changes:** `tar -czf backup.tar.gz ./mail-data ./config`
2. **Test email flow after changes:** Send test emails in/out
3. **Monitor disk space:** Check `df -h` regularly
4. **Keep certificates updated:** Let's Encrypt certs expire every 90 days
5. **Review logs periodically:** Look for authentication failures or delivery issues
6. **Set appropriate quotas:** Prevent users from filling up disk
7. **Use strong passwords:** Minimum 12 characters for email accounts
8. **Enable DKIM/SPF/DMARC:** Improves deliverability
9. **Regular updates:** `docker compose pull && docker compose up -d`
10. **Document your changes:** Keep notes on custom configurations