



More Raspberry Pi Projects

Nine Advanced Projects for the Serious Pi Enthusiast

Nextcloud · Email Server · Grafana · Docker · Tor Relay ADS-B Flight Tracker · Pi AI Kit · RetroPie

osztromok.com

Contents

Project 1 — Personal Cloud with Nextcloud

Apache · PHP · MariaDB · SSL · Sync Clients

Project 2 — Email Server Part 1

Postfix · Dovecot · Roundcube Webmail

Project 3 — Email Server Part 2

TLS · SPF · DKIM · DMARC · SpamAssassin

Project 4 — Network Monitoring with Grafana

InfluxDB · Telegraf · Flux Queries · Alerts

Project 5 — Docker and Containers

Images · Compose · Volumes · ARM64 Compatibility

Project 6 — Tor Relay

Middle Relay · torrc · Bandwidth · Nyx Monitor

Project 7 — ADS-B Flight Tracker

RTL-SDR · readsb · tar1090 · FlightAware Feeder

Project 8 — Machine Learning at the Edge

Pi AI Kit · Hailo NPU · YOLOv8 · Object Detection

Project 9 — Retro Gaming with RetroPie

EmulationStation · Shaders · RetroAchievements · HA

Project 1 — Personal Cloud with Nextcloud

Your own Google Drive, running at home

What you build	A self-hosted cloud with file sync, calendar, contacts, and sharing
Difficulty	Intermediate — several components to configure correctly
Time	2 – 4 hours for a production-ready installation
Pi requirement	Pi 4 (4 GB recommended); SSD strongly recommended for performance

Install the Stack

```
sudo apt install -y apache2 mariadb-server php php-xml php-gd \
    php-json php-imagick php-apcu php-redis redis-server
sudo mysql_secure_installation
```

MariaDB Setup

```
CREATE DATABASE nextcloud CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE USER 'nextcloud'@'localhost' IDENTIFIED BY 'strong_password';
GRANT ALL PRIVILEGES ON nextcloud.* TO 'nextcloud'@'localhost';
FLUSH PRIVILEGES;
```

Key occ Commands

```
sudo -u www-data php occ maintenance:mode --on/--off
sudo -u www-data php occ files:scan --all
sudo -u www-data php occ app:install calendar
sudo -u www-data php occ update:check
```

Cron Job (performance)

```
# Run as www-data — add via: sudo crontab -u www-data -e
*/5 * * * * php -f /var/www/nextcloud/cron.php
```

■ Sync clients for Nextcloud are available free for Windows, Mac, Linux, iOS, and Android. CalDAV and CardDAV URLs let you sync calendars and contacts with native device apps.

Project 2 — Email Server Part 1

Postfix · Dovecot · Roundcube Webmail

What you build	A working mail server that can send and receive email, with a webmail UI
Difficulty	Advanced — the most complex project in this course
Time	Parts 1 and 2 together: 4 – 6 hours
Requires	A domain name you own, port 25 not blocked by ISP, port forwarding

The Mail Stack

Postfix	MTA — receives and sends email between servers (ports 25 and 587)
Dovecot	IMAP — stores and serves mail to clients (ports 143 and 993)
Roundcube	Webmail — browser-based email client, talks to Dovecot via IMAP

Installation

```
sudo apt install -y postfix postfix-mysql mailutils
# Choose: Internet Site → enter your domain (not mail.yourdomain.com)
sudo apt install -y dovecot-core dovecot-imapd dovecot-lmtpd
sudo apt install -y roundcube roundcube-mysql
```

DNS Records Required

A record	mail.yourdomain.com → your public IP
MX record	@ → 10 mail.yourdomain.com

■ Check that your ISP does not block inbound port 25 before starting. Many residential ISPs block this port. Contact them to request it be opened, or use a mail relay service.

Project 3 — Email Server Part 2

TLS · SPF · DKIM · DMARC · SpamAssassin

What you add	TLS encryption, email authentication (SPF/DKIM/DMARC), spam filtering
Why it matters	Without this, email goes to spam and passwords travel in plaintext
Time	2 – 3 hours plus DNS propagation time
Prerequisite	Project 2 complete — Postfix, Dovecot, and Roundcube all working

Email Authentication Summary

SPF	DNS TXT record listing authorised sending IPs. Tells receivers which servers can send from your domain.
DKIM	Cryptographic signature on every outgoing email. Verified using a public key in DNS.
DMARC	Policy record telling receivers what to do when SPF/DKIM fail. Requests aggregate reports.

DNS Records

```
# SPF (TXT record at @)
v=spf1 ip4:YOUR.IP.ADDRESS -all

# DKIM (TXT record at mail._domainkey)
v=DKIM1; h=sha256; k=rsa; p=<your-public-key>

# DMARC (TXT record at _dmarc)
v=DMARC1; p=none; rua=mailto:dmarc@yourdomain.com; fo=1
```

Install OpenDKIM

```
sudo apt install -y opendkim opendkim-tools
sudo opendkim-genkey -t -s mail -d yourdomain.com \
  -D /etc/opendkim/keys/yourdomain.com/
# Publish the public key in DNS, then restart Postfix + OpenDKIM
```

- Test your full email score at mail-tester.com. A correctly configured server should score 9–10/10. Check IP blocklist status at mxtoolbox.com/blacklists.aspx

Project 4 — Network Monitoring with Grafana

InfluxDB · Telegraf · Live dashboards

What you build	Live dashboards showing CPU, memory, network, Pi-hole stats, and sensor data
Difficulty	Intermediate — installation straightforward, dashboard building takes experimentation
Time	2 – 3 hours for core stack
Pi requirement	Pi 4 (2 GB+ recommended); SSD recommended for InfluxDB write performance

The TIG Stack

Telegraf	Metrics collection agent — CPU, memory, disk, network, Pi-hole, ping
InfluxDB	Time-series database — stores all metrics with timestamps
Grafana	Visualisation — queries InfluxDB and renders charts, graphs, and gauges

Installation

```
# Add InfluxData repository, then:
sudo apt install -y influxdb2 influxdb2-cli telegraf
# Add Grafana repository, then:
sudo apt install -y grafana
sudo systemctl enable --now influxdb telegraf grafana-server
```

Example Flux Query

```
from(bucket: "telegraf")
  |> range(start: -1h)
  |> filter(fn: (r) => r._measurement == "cpu"
                    and r._field == "usage_percent"
                    and r.cpu == "cpu-total")
  |> aggregateWindow(every: 1m, fn: mean)
```

■ Access Grafana at <http://raspberrypi.local:3000> (admin/admin). Import community dashboards by ID at grafana.com/grafana/dashboards. Dashboard IDs 10578 and 15650 are excellent starting points.

Project 5 — Docker and Containers

Package apps in isolation — upgrade, remove, and rebuild with ease

What you learn	How Docker works, running containerised apps, managing stacks with Docker Compose
Difficulty	Intermediate — concepts take time to internalise; commands are straightforward
Time	2 – 3 hours for installation and core concepts
Pi requirement	Pi 4 (2 GB+); SSD recommended; 64-bit Raspberry Pi OS required

Installation

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
sudo usermod -aG docker $USER
newgrp docker
sudo systemctl enable docker containerd
```

Docker Compose — Pi-hole Example

```
# ~/docker/pihole/compose.yaml
services:
  pihole:
    image: pihole/pihole:latest
    ports:
      - "53:53/tcp"
      - "53:53/udp"
      - "8053:80/tcp"
    environment:
      TZ: 'Europe/London'
      WEBPASSWORD: 'change_me'
    volumes:
      - 'pihole_data:/etc/pihole'
    restart: unless-stopped
```

Essential Commands

```
docker compose up -d      # start all services
docker compose down       # stop (volumes preserved)
docker compose pull       # pull latest images
docker compose logs -f    # follow logs
docker ps                 # list running containers
docker exec -it name bash # shell inside a container
docker system prune -a    # clean unused images/containers
```

■ Always use 64-bit Raspberry Pi OS for Docker. Many images are published for linux/arm64 only. Check with:
`docker buildx imagetools inspect image:tag`

Project 6 — Tor Relay

Contribute bandwidth to the privacy network

What you build	A Tor middle relay — your Pi passes encrypted traffic for privacy-seeking users
Difficulty	Beginner-friendly — installation is simple
Time	1 – 2 hours to set up; 2 – 4 days to reach full speed
Important	Run a MIDDLE relay only — not an exit relay — on a home connection

Relay Types

Guard relay	First hop — knows user IP. Low-moderate risk.
Middle relay	* Recommended. Passes encrypted traffic. Knows neither user nor destination. Minimal risk.
Exit relay	Last hop — makes connections to websites. Receives abuse complaints. DO NOT run at home.

Installation & Configuration

```
# Add Tor Project repository, then:
sudo apt install -y tor

# /etc/tor/torrc — key settings:
Nickname      MyPiRelay
ContactInfo    you_AT_example.com
ORPort        9001
ExitPolicy     reject *:.*    # CRITICAL — middle relay only
RelayBandwidthRate 1 MB
RelayBandwidthBurst 2 MB
```

Monitor with Nyx

```
sudo apt install -y nyx
# Add to torrc: ControlPort 9051, CookieAuthentication 1
sudo usermod -aG debian-tor $USER
nyx # launch the terminal dashboard
```

■ Find your relay at metrics.torproject.org/rs.html after 24 hours. It takes 2–4 days to reach full traffic as the network gradually builds trust in new relays.

Project 7 — ADS-B Flight Tracker

Watch live aircraft overhead with a £20 USB dongle

What you build	A live radar map showing every ADS-B aircraft within 200–300 km
Difficulty	Beginner-friendly — one of the most satisfying Pi projects
Time	1 – 2 hours from unboxing to watching planes
Hardware	RTL-SDR USB dongle (~£20) + 1090 MHz antenna (included or ~£10 extra)

Installation

```
# Install readsb (ADS-B decoder):
curl -L https://github.com/wiedehopf/adsb-wiki/raw/master/readsb-install.sh | sudo bash

# Install tar1090 (radar web interface):
curl -L https://github.com/wiedehopf/tar1090/raw/master/install.sh | sudo bash

# Install graphs1090 (historical statistics):
curl -L https://github.com/wiedehopf/graphs1090/raw/master/install.sh | sudo bash
```

Access Your Radar

```
http://raspberrypi.local:8080          # live radar map
http://raspberrypi.local:8080/graphs1090/ # statistics
```

Feeding Services (earn free premium accounts)

FlightAware	Free Enterprise subscription (~\$89/mo). Install: PiAware package
Flightradar24	Free Business subscription (~\$500/yr). Install: fr24feed package
ADSBexchange	Community-run, unfiltered (incl. military). Free feeder package

■ Antenna placement is everything. Every metre of additional height doubles your detection horizon. A loft or rooftop antenna typically sees 2–3× more aircraft than one on a windowsill.

Project 8 — Machine Learning at the Edge

Real-time object detection with the Pi AI Kit and Hailo NPU

What you build	A real-time camera system identifying people, animals, and vehicles locally
Difficulty	Intermediate-Advanced — good tooling makes it accessible
Time	2 – 4 hours (AI Kit path); 1 – 2 hours (software-only)
Hardware	Pi AI Kit (~£65) for Pi 5, or software-only on Pi 4/5

Install Hailo Software Stack (AI Kit)

```
sudo apt install -y hailo-all
sudo reboot

# Verify the NPU is detected:
hailortcli fw-control identify
# Should show: Board Name: Hailo-8L
```

Run the Built-in Demo

```
rplicam-hello -t 0 \
  --post-process-file /usr/share/rpi-camera-assets/hailo_yolov8_inference.json \
  --lores-width 640 --lores-height 640
# Live preview with bounding boxes drawn on detected objects
```

Software-only (Pi 4 / Pi 5 without AI Kit)

```
pip3 install --break-system-packages ultralytics picamera2
from ultralytics import YOLO
model = YOLO('yolov8n.pt') # nano – fastest on Pi 4
results = model(frame, conf=0.5)
```

- The Hailo NPU runs YOLOv8s at 30+ FPS, leaving the CPU free for notifications, video recording, and web serving. The Pi 4 CPU-only path runs at ~2–4 FPS (nano model).

Project 9 — Retro Gaming with RetroPie

EmulationStation · Save states · Shaders · Achievements

What you build	A polished retro gaming console with beautiful artwork, save states, and achievements
Difficulty	Beginner-friendly for basics; deep rabbit hole if you want to
Time	1 – 2 hours to a working console; a pleasant afternoon to polish
Pi requirement	Pi 3B+ for NES/SNES/PS1; Pi 4 for N64/Dreamcast

Installation

Flash the RetroPie image for your Pi model from retropie.org.uk/download using Raspberry Pi Imager (Choose OS → Use custom).

Hotkey Reference

Select + R1	Save state
Select + L1	Load state
Select + Start	Exit game
Select + X	RetroArch menu (shaders, settings)
Select + L2	Rewind (hold)
Select + Left/Right	Change save slot

Scrape Artwork with Skyscraper

```
# Gather resources from ScreenScraper:
Skyscraper -p snes -s screenscraper -u "username:password"

# Generate gamelist and composite artwork:
Skyscraper -p snes
```

Home Assistant Integration

RetroPie fires scripts on game start/end. Use them to call Home Assistant webhooks and automate your living room:

```
# /opt/retropie/configs/all/runcommand-onstart.sh
curl -s -X POST "http://homeassistant.local:8123/api/webhook/retropie_start" \
  -H "Content-Type: application/json" \
  -d "{\"system\": \"$1\", \"game\": \"$(basename $3)\"}"
```

[*] Congratulations — More Raspberry Pi Projects complete! You have built a personal cloud, a mail server, a monitoring stack, a container platform, a Tor relay, an aircraft radar, an AI camera, and a retro gaming console. Nine projects, one tiny computer.