

■ HomeLab Master Guide

Complete M4 Mac Mini Documentation

62+ Services | 100,000+ Words
22-Day Implementation Guide

Generated: November 11, 2025

■ COMPLETE HomeLab Master Guide

Generated: 2025-11-11 23:25:42

> Your complete M4 Mac Mini homelab documentation - ALL guides in ONE searchable file

■ About This Document

This master document contains EVERY guide in your homelab vault: - ■ All reference guides - ■ All Day-by-Day implementation guides - ■ All BONUS guides - ■ Complete searchability (Ctrl+F to find anything!) - ■ Obsidian wikilinks to individual files - ■ Full table of contents

Total Size: 100,000+ words Total Guides: 26 Total Services: 62+

■ Complete Table of Contents

6. Day-01-02-Foundation-and-Accounts|■ Volume 01: Foundation (Days 1-2) - COMPLETE GUIDE - Jump to section
7. Day-01-02-M4-Mac-Mini-Setup|■ Volume 02: M4 Mac Mini Hardware & macOS Setup (Days 1-2) - Jump to section
8. Day-02-BONUS-Deco-Mesh-Setup|Day 02 BONUS: TP-Link Deco XE75 Mesh Setup - Jump to section
9. Day-02-M4-Mac-Mini-Setup|■ Volume 02: M4 Mac Mini Setup (Days 1-2) - COMPLETE GUIDE - Jump to section
10. Day-03-05-Docker-Infrastructure|Volume 03: Docker Infrastructure (Days 3-4) - Jump to section
11. Day-03-Deco-Mesh-WiFi|Guide 08: TP-Link Deco XE75 Mesh Network Setup - Jump to section
12. Day-05-BONUS-Remote-Access|Day 05 BONUS: Remote Access - RustDesk + Guacamole - Jump to section
13. Day-05-Remote-Access|Guide 12: Remote Access - RustDesk + Guacamole - Jump to section
14. Day-06-08-Proxmox-Hypervisor|Volume 04: Proxmox Virtualization Platform (Days 5-6) - Jump to section
15. Day-09-10-Windows-11-VM|Volume 05: Windows 11 Pro VM (Day 6) - Jump to section
16. Day-11-12-Ubuntu-Kali-VMs|Volume 06: Ubuntu & Kali Linux VMs (Day 6) - Jump to section
17. Day-13-14-Media-Stack-Plex|Volume 07: Media Stack (Days 7-8) - Jump to section

- 18. Day-15-16-Smart-Home-Integration|Volume 08: Smart Home Integration (Day 9) - Jump to section
- 19. Day-17-18-AI-Services-LLMs|Volume 09: AI Services (Day 10) - Jump to section
- 20. Day-19-Security-Hardening|Volume 10: Security Stack (Day 11) - Jump to section
- 21. Day-20-Monitoring-Setup|Volume 11: Monitoring Stack (Day 12) - Jump to section
- 22. Day-21-Backup-Strategy|Volume 12: Backup Strategy (Day 13) - Jump to section
- 23. Day-22-Maintenance-and-RMM|Volume 13: Maintenance & Action1 RMM (Day 14) - Jump to section
- 24. Day-22-Network-Diagrams-Final|Volume 14: Network Diagrams & Access Guide (Day 14) - Jump to section
- 25. REF-Proxmox-Quick-Reference|■ Proxmox Quick Reference Card - Jump to section
- 26. V2.0-UPDATE-SUMMARY|HomeLab Vault v2.0 - Update Summary - Jump to section

=====

■ ■ Volume 01: Foundation (Days 1-2) - COMPLETE GUIDE

Source: Day-01-02-Foundation-and-Accounts Tags: #homelab #foundation #setup #accounts

■ Volume 01: Foundation (Days 1-2) - COMPLETE GUIDE

Building the Foundation for Steve's Epic HomeLab

> "A house built on sand will fall, but a homelab built on proper foundations will serve you for years!" ■■

■ Volume Overview

Days: 1-2 Time Required: 12-16 hours total Coffee Required: ■■■■■■■■■■ (8 cups minimum!) Difficulty: Beginner to Intermediate

What You'll Complete

By end of Day 2, you'll have: - ■ All accounts created and documented in Day-01-02-Foundation-and-Accounts|1Password - ■ All software downloaded and ready - ■ M4 Mac Mini - Complete System Overview|M4 Mac Mini unboxed and configured - ■ Network optimized with static IP - ■ 4TB external SSD formatted and ready - ■ Essential tools installed - ■ Solid foundation for 60+ services

■ Guides in This Volume

- Guide 00: Prerequisites & Account Setup - Guide 01: Hardware Preparation - [Guide 02: M4 Mac Mini - Complete System Overview|M4 Mac Mini Initial Setup](#guide-02-m4-initial-setup) - Guide 03: macOS Configuration - Guide 04: Network Setup - Guide 05: Storage Configuration - Guide 06: Essential Tools Installation

Guide 00: Prerequisites & Account Setup

■■ Time: 2-3 hours ■ Coffee: 2 cups ■ Difficulty: Beginner

■ Prerequisites Checklist

Before you begin, verify you have:

Hardware (Already Owned)

- [x] M4 Mac Mini - Complete System Overview|M4 Mac Mini (32GB RAM, 512GB storage) - [x] 4TB External SSD (Samsung T7 or similar) - [x] Day-03-Deco-Mesh-WiFi|Sky Router (existing) - [x] Day-02-M4-Mac-Mini-Setup|TP-Link SX1008 M4 Mac Mini - Complete System Overview|10GbE Switch - [x] M4 Mac Mini - Complete System Overview|10GbE cables (Cat6a or better) - [x] Monitor (for initial setup) - [x] Keyboard & Mouse (for initial setup) - [x] Samsung S25 Ultra (your phone) - [x] MacBook Air (secondary device) - [x] iPad Pro (optional, for management)

Network Infrastructure

- [x] CityFibre 5Gbps/5Gbps internet active - [x] Router accessible at 192.168.50.1 - [x] Admin access to router - [x] M4 Mac Mini - Complete System Overview|10GbE switch connected - [x] Available IP addresses: 192.168.50.10-192.168.50.100

Existing Accounts (You Already Have)

- [x] ExpressVPN subscription active - [x] Day-01-02-Foundation-and-Accounts|1Password Teams account - [x] ChatGPT Teams subscription - [x] Claude Pro subscription - [x] Perplexity Pro subscription - [x] Day-21-Backup-Strategy|pCloud 7TB Lifetime account - [x] GitHub account (steve-smithit) - [x] Microsoft

account (itstevesmithIT@gmail.com)

■ Website Registrations & Account Creation

CRITICAL: Create these accounts BEFORE starting setup. Save ALL credentials in Day-01-02-Foundation-and-Accounts|1Password immediately!

1■■ Day-01-02-Foundation-and-Accounts|1Password HomeLab Vault Setup

Do this FIRST - everything else goes in here!

1. Open Day-01-02-Foundation-and-Accounts|1Password - Already installed on Samsung S25 Ultra - Already installed on MacBook Air - Install on iPad Pro if using

2. Create HomeLab Vault

Open 1Password → Vaults → Create New Vault Name: HomeLab Description: All homelab accounts, servers, and services Icon: Server/Home icon Click: Create Vault

3. Verify Vault Created - [x] HomeLab vault appears in vault list - [x] Can switch to HomeLab vault - [x] Vault syncs across devices

■ CHECKPOINT: HomeLab vault created and syncing

2■■ Day-01-02-Foundation-and-Accounts|DuckDNS Account (Free Dynamic DNS)

Purpose: Your subdomain for remote access (yoshikeikohomelab.duckdns.org)

1. Go to: <https://www.duckdns.org/>

2. Sign in with Google: - Click: "Sign in with Google" - Authorize Day-01-02-Foundation-and-Accounts|DuckDNS - You're now logged in

3. Create Subdomain:

Subdomain: yoshikeikohomelab (Results in: yoshikeikohomelab.duckdns.org) Click: Add domain

4. Note Your Token:

Token appears at top of page (long string) COPY THIS IMMEDIATELY!

5. Save to Day-01-02-Foundation-and-Accounts|1Password:

1Password → HomeLab Vault → New Login Title: DuckDNS - yoshikeikohomelab Website: <https://www.duckdns.org/> Username: (GitHub account) Password: (not applicable - uses GitHub OAuth) Add Custom Field: - Token: [paste your token here] - Subdomain: yoshikeikohomelab.duckdns.org Notes: Dynamic DNS for remote access Free tier, updates via API Tags: dns, networking, homelab Click: Save

6. Verify: - [x] Token saved in Day-01-02-Foundation-and-Accounts|1Password - [x] Subdomain created (yoshikeikohomelab.duckdns.org) - [x] Can access Day-01-02-Foundation-and-Accounts|DuckDNS dashboard

■ CHECKPOINT: Day-01-02-Foundation-and-Accounts|DuckDNS configured with subdomain

3■■■ Day-01-02-Foundation-and-Accounts|Tailscale Account (VPN)

Purpose: Secure VPN access to homelab from anywhere

1. Go to: <https://tailscale.com/>

2. Sign Up with GitHub: - Click: "Get started" - Click: "Sign up with Google" - itstevesmithit@gmail.com - Account created!

3. Download Installers:

For M4 Mac Mini: - Download: Tailscale for macOS (Apple Silicon) - Save to: ~/Desktop/HomeLab-Project/Installers/ For Samsung S25 Ultra: - Already installed from Guide 00 prep For MacBook Air: - Download: Tailscale for macOS

4. Save to Day-01-02-Foundation-and-Accounts|1Password:

1Password → HomeLab Vault → New Login Title: Tailscale VPN Account Website: <https://login.tailscale.com/> Username: (GitHub account) Password: (not applicable - uses GitHub OAuth) Add Custom Fields: - Tailnet Name: [your-tailnet].ts.net - Admin Console: <https://login.tailscale.com/admin/machines> Notes: VPN for secure remote access Free tier: 100 devices Connected devices: - M4 Mac Mini (192.168.50.10) - Samsung S25 Ultra - MacBook Air - iPad Pro Tags: vpn, security, networking Click: Save

5. Verify: - [x] Day-01-02-Foundation-and-Accounts|Tailscale account created - [x] Installers downloaded - [x] Credentials in Day-01-02-Foundation-and-Accounts|1Password

■ CHECKPOINT: Day-01-02-Foundation-and-Accounts|Tailscale account ready

4■■■ Day-03-05-Docker-Infrastructure|Docker Hub Account

Purpose: Pull container images for 60+ services

1. Go to: <https://hub.docker.com/>

2. Sign Up:

Email: itstevesmithIT@gmail.com Username: stevesmithit (or similar) Password: [create strong password] Click: Sign Up

3. Verify Email: - Check outlook.com inbox - Click verification link - Account activated

4. Save to Day-01-02-Foundation-and-Accounts|1Password:

1Password → HomeLab Vault → New Login Title: Docker Hub Website: <https://hub.docker.com/> Username: stevesmithit Password: [your password] Email: itstevesmithIT@gmail.com Notes: Container registry for Docker images Free tier: unlimited public pulls Used for all homelab services Tags: docker, containers, homelab Click: Save

5. Verify: - [x] Account created and verified - [x] Can log in to Day-03-05-Docker-Infrastructure|Docker Hub - [x] Credentials in Day-01-02-Foundation-and-Accounts|1Password

■ CHECKPOINT: Day-03-05-Docker-Infrastructure|Docker Hub account ready

5■■■ Day-06-08-Proxmox-Hypervisor|Proxmox Account (Optional but Recommended)

Purpose: Access Day-06-08-Proxmox-Hypervisor|Proxmox forums and documentation

1. Go to: <https://www.proxmox.com/>

2. Create Account:

Click: Forum or Login Email: itstevesmithIT@gmail.com Username: stevesmithit Password: [create strong password] Register

3. Save to Day-01-02-Foundation-and-Accounts|1Password:

1Password → HomeLab Vault → New Login Title: Proxmox Forum Account Website: <https://forum.proxmox.com/> Username: stevesmithit Password: [your password] Email: itstevesmithIT@gmail.com Notes: Proxmox community forum access For support and updates Tags: proxmox, virtualization, support Click: Save

■ CHECKPOINT: Day-06-08-Proxmox-Hypervisor|Proxmox account ready

6■■■ Day-13-14-Media-Stack-Plex|Plex Account

Purpose: Media server management and remote access

1. Go to: <https://www.plex.tv/>

2. Sign Up:

Option 1: Use Google (itstevesmithIT@gmail.com) Option 2: Use email directly Create account

3. Get Day-13-14-Media-Stack-Plex|Plex Pass (Optional but Recommended):

Lifetime: £119 Benefits: - Hardware transcoding - Mobile sync - Premium music features - Early access features Or monthly: £4.99

4. Save to Day-01-02-Foundation-and-Accounts|1Password:

1Password → HomeLab Vault → New Login Title: Plex Media Server Website: <https://www.plex.tv/> Username/Email: itstevesmithIT@gmail.com Password: [if using email login] Add Custom Fields: - Plex Pass: Yes/No - Claim Token: (will get this during setup) Notes: Media server account Connected to M4 Mac Mini server Access: app.plex.tv Tags: media, streaming, homelab Click: Save

5. Verify: - [x] Day-13-14-Media-Stack-Plex|Plex account created - [] Can access plex.tv - [x] Credentials in Day-01-02-Foundation-and-Accounts|1Password

■ CHECKPOINT: Day-13-14-Media-Stack-Plex|Plex account ready

7■■■ Day-15-16-Smart-Home-Integration|Home Assistant Account (Cloud Optional)

Purpose: Smart home management

1. Go to: <https://www.home-assistant.io/>

2. Create Nabu Casa Account (Optional):

Cloud access: £6.50/month Benefits: - Remote access without VPN - Alexa/Google Assistant integration - Support Home Assistant development OR use Tailscale for free remote access

3. Note: Day-15-16-Smart-Home-Integration|Home Assistant account created during setup, but save placeholder:

1Password → HomeLab Vault → New Login Title: Home Assistant Website: <http://192.168.50.10:8123> Username: (will create during setup) Password: (will create during setup) Add Custom Fields: - Local URL: <http://192.168.50.10:8123> - Tailscale URL: <http://100.x.x.x:8123> - DuckDNS URL: <https://ha.yoshikeikohomelab.duckdns.org> Notes: Smart home automation platform 43+ devices integrated: - 10 Sonos speakers - 5 Nest cameras - Tesla Model Y - Cupra Born EV - Easee charger - LG TV - Xbox/PS5 Tags: smarthome, automation, homelab Click: Save

■ CHECKPOINT: Day-15-16-Smart-Home-Integration|Home Assistant placeholder created

8■■■ Cloudflare Account (Optional - Advanced)

Purpose: DNS management, tunnel alternative to Day-01-02-Foundation-and-Accounts|DuckDNS

1. Go to: <https://www.cloudflare.com/>

2. Sign Up:

Email: itstevesmithIT@gmail.com Password: [create strong password] Sign Up

3. Add Domain (if you buy one):

Could purchase: smithhomelab.com or similar Free alternative: Use DuckDNS (already set up)

4. Save to Day-01-02-Foundation-and-Accounts|1Password:

1Password → HomeLab Vault → New Login Title: Cloudflare DNS Website: <https://dash.cloudflare.com/> Username: itstevesmithIT@gmail.com Password: [your password] Add Custom Fields: - API Token: (create later if needed) - Domain: (if registered) Notes: Optional DNS management Alternative to DuckDNS for custom domain Free tier available Tags: dns, networking, optional Click: Save

■ CHECKPOINT: Cloudflare account ready (optional)

9■■■ Ntfy.sh Account (Push Notifications)

Purpose: Receive alerts from your homelab

1. Go to: <https://ntfy.sh/>

2. Create Topic:

```
No account needed for basic use! Just pick a unique topic name: Topic:
yoshikeikohomelab-alerts-x7k9m2 (Add random chars to make unique) URL:
https://ntfy.sh/yoshikeikohomelab-alerts
```

3. Install App on Phone:

```
Samsung S25 Ultra: - Install: Ntfy app from Play Store - Subscribe to:
yoshikeikohomelab-alerts - Allow notifications
```

4. Save to Day-01-02-Foundation-and-Accounts|1Password:

```
1Password → HomeLab Vault → Secure Note Title: Ntfy Notifications Content: Topic:
yoshikeikohomelab-alerts-x7k9m2 URL: https://ntfy.sh/yoshikeikohomelab-alerts Send
notification: curl -d "Message text" ntfy.sh/yoshikeikohomelab-alerts-x7k9m2 Installed
on: - Samsung S25 Ultra - MacBook Air Notes: Free push notification service Used for
homelab alerts: - Backup completion - Service failures - System updates - Security
alerts Tags: notifications, monitoring, homelab Click: Save
```

5. Test Notification:

```
# From any terminal: curl -d "Test from setup" ntfy.sh/yoshikeikohomelab-alerts #
Should receive notification on phone!
```

6. Verify: - [x] Topic name chosen and unique - [x] App installed on phone - [x] Test notification received - [x] Info saved in Day-01-02-Foundation-and-Accounts|1Password

■ CHECKPOINT: Notifications working

■ Day-22-Maintenance-and-RMM|Action1 RMM Account

Purpose: Remote management and patch management

1. Go to: <https://www.action1.com/>

2. Sign Up:

```
Email: itstevesmithIT@gmail.com Company: Personal HomeLab Sign Up (Free tier available)
```

3. Verify Email

4. Save to Day-01-02-Foundation-and-Accounts|1Password:

```
1Password → HomeLab Vault → New Login Title: Action1 RMM Website:
https://app.action1.com/ Username: itstevesmithIT@gmail.com Password: [your password]
Notes: Remote monitoring and management Free tier: up to 100 endpoints Will install
agent on: - M4 Mac Mini (host) - Windows 11 VM - Ubuntu Server VM Tags: rmm, management,
monitoring Click: Save
```

■ CHECKPOINT: Day-22-Maintenance-and-RMM|Action1 account ready

■ Software Download Checklist

Create folder structure first:

```
# On MacBook Air (before M4 arrives): mkdir -p ~/Desktop/HomeLab-Project/Installers
mkdir -p ~/Desktop/HomeLab-Project/ISOs mkdir -p
~/Desktop/HomeLab-Project/Documentation
```

Essential Downloads (Get These Now):

For M4 Mac Mini - Complete System Overview|M4 Mac Mini:

- [x] Day-01-02-Foundation-and-Accounts|Tailscale - URL: <https://tailscale.com/download/mac> - File: Day-01-02-Foundation-and-Accounts|Tailscale-*.pkg - Save to: ~/Desktop/HomeLab-Project/Installers/
- [x] Day-03-05-Docker-Infrastructure|OrbStack (Day-03-05-Docker-Infrastructure|Docker) - URL: <https://orbstack.dev/download> - File: Day-03-05-Docker-Infrastructure|OrbStack.dmg - Save to: ~/Desktop/HomeLab-Project/Installers/
- [x] Day-01-02-Foundation-and-Accounts|1Password (if not installed) - URL: <https://1password.com/downloads/mac/> - File: Day-01-02-Foundation-and-Accounts|1Password-*.pkg - Save to: ~/Desktop/HomeLab-Project/Installers/
- [] Homebrew (will install via command) - URL: <https://brew.sh/> - Installation script (run during setup)

VM Operating Systems (ISOs):

- [x] Day-06-08-Proxmox-Hypervisor|Proxmox VE ARM64 - URL: <https://www.proxmox.com/en/downloads> - File: proxmox-ve-*-arm64.iso (~1GB) - Save to: ~/Desktop/HomeLab-Project/ISOs/
- [x] Ubuntu Server 24.04 ARM64 - URL: <https://ubuntu.com/download/server/arm> - File: ubuntu-24.04-live-server-arm64.iso (~2GB) - Save to: ~/Desktop/HomeLab-Project/ISOs/
- [x] Windows 11 ARM64 Insider Preview - URL: <https://www.microsoft.com/en-us/software-download/windowsinsiderpreviewARM64> - Requires: Windows Insider account - File: Windows11_InsiderPreview_*.vhdx (~10GB) - Save to: ~/Desktop/HomeLab-Project/ISOs/
- [] Kali Linux ARM64 - URL: <https://www.kali.org/get-kali/#kali-installer-images> - File: kali-linux-2024.*-installer-arm64.iso (~4GB) - Save to: ~/Desktop/HomeLab-Project/ISOs/
- [] macOS Sequoia (optional) - Create from: /Applications/Install macOS Sequoia.app - Will create during setup - Size: ~13GB

Optional but Recommended:

- [x] Visual Studio Code - URL: <https://code.visualstudio.com/> - File: VSCode-darwin-arm64.zip - Save to: ~/Desktop/HomeLab-Project/Installers/
- [x] iTerm2 (Better terminal) - URL: <https://iterm2.com/> - File: iTerm2-*.zip - Save to: ~/Desktop/HomeLab-Project/Installers/

Download Verification Checklist:

After downloading, verify: - ☒ All files in correct folders - ☒ No corrupted downloads (check file sizes) - ☒ ISOs can be mounted/verified - ☐ Total space used: ~30-40GB

■ CHECKPOINT: All software downloaded

■ Mobile App Installation

Install on Samsung S25 Ultra now (before M4 arrives):

Essential Apps:

- ☒ Day-01-02-Foundation-and-Accounts|1Password - Password manager - ☒
Day-01-02-Foundation-and-Accounts|Tailscale - VPN access - ☒ Ntfy - Notifications - ☒
Day-15-16-Smart-Home-Integration|Home Assistant - Smart home control - ☒
Day-13-14-Media-Stack-Plex|Plex - Media streaming - ☒ Immich (later) - Photo backup

Optional but Useful:

- ☐ Microsoft Remote Desktop - For Windows VM - ☐ JuiceSSH or Termius - SSH client - ☐
Day-20-Monitoring-Setup|Grafana (web app) - Monitoring

■ CHECKPOINT: Apps installed on phone

■ Security Setup

Enable 2FA Where Possible:

Use Microsoft Authenticator or Google Authenticator:

- ☐ Day-01-02-Foundation-and-Accounts|1Password (REQUIRED - protects everything!) - ☐ GitHub account - ☐
Microsoft account - ☐ Day-13-14-Media-Stack-Plex|Plex account - ☐ Cloudflare (if used) - ☐
Day-22-Maintenance-and-RMM|Action1 RMM

Add to each service in Day-01-02-Foundation-and-Accounts|1Password:

Edit login → Add One-Time Password Scan QR code or enter secret Save

■ CHECKPOINT: 2FA enabled on critical accounts

■ Day-01-02-Foundation-and-Accounts|1Password Vault Organization

Create these categories in HomeLab vault:

```
HomeLab Vault Structure: ■■ Accounts (website logins) ■ ■■ DuckDNS ■ ■■ Tailscale ■
■■ Docker Hub ■ ■■ Plex ■ ■■ etc. ■■ Servers (server credentials) ■ ■■ M4 Mac Mini ■
■■ Proxmox ■ ■■ Ubuntu Server VM ■ ■■ Windows 11 VM ■ ■■ Kali Linux VM ■■ Services
(homelab services) ■ ■■ Home Assistant ■ ■■ Nginx Proxy Manager ■ ■■ Portainer ■ ■■
etc. ■■ API Keys (tokens and secrets) ■ ■■ DuckDNS Token ■ ■■ Plex Claim Token ■ ■■
etc. ■■ Secure Notes (other info) ■■ Network Configuration ■■ IP Address List ■■
Backup Procedures
```

■ CHECKPOINT: Day-01-02-Foundation-and-Accounts|1Password organized

■ Documentation Template

Create these documents in Day-01-02-Foundation-and-Accounts|1Password Secure Notes:

1. Network Configuration Note:

```
1Password → HomeLab Vault → New Secure Note Title: Network Configuration Content:
Router: 192.168.50.1 - Admin: admin - Password: [router password] - CityFibre
5Gbps/5Gbps Network: 192.168.50.0/24 Gateway: 192.168.50.1 DNS: 192.168.50.1, 1.1.1.1
Reserved IPs: - 192.168.50.1 = Sky Router - 192.168.50.10 = M4 Mac Mini - 192.168.50.50
= Proxmox Host - 192.168.50.51 = Ubuntu Server VM - 192.168.50.52 = Windows 11 VM -
192.168.50.53 = Kali Linux VM - 192.168.50.54 = macOS VM (optional) 10GbE Switch:
TP-Link SX1008 Ports used: - Port 1: M4 Mac Mini - Port 2: Router uplink WiFi SSID:
[your wifi name] WiFi Password: [wifi password] Tags: network, configuration, homelab
Click: Save
```

2. Service URLs Note:

```
1Password → HomeLab Vault → New Secure Note Title: Service URLs - Quick Reference
Content: Local Access (192.168.50.x): - Proxmox: https://192.168.50.50:8006 - Home
Assistant: http://192.168.50.10:8123 - Plex: http://192.168.50.10:32400/web -
Portainer: https://192.168.50.10:9443 - Nginx Proxy Manager: http://192.168.50.10:81 -
Grafana: http://192.168.50.10:3001 Remote Access (DuckDNS): - Home:
https://home.yoshikeikohomelab.duckdns.org - Plex:
https://plex.yoshikeikohomelab.duckdns.org - HA:
https://ha.yoshikeikohomelab.duckdns.org Tailscale IPs (update after setup): - M4 Mac
Mini: 100.x.x.x - Samsung S25: 100.x.x.x - MacBook Air: 100.x.x.x Tags: urls, access,
quick-reference Click: Save
```

3. Backup Information Note:

```
1Password → HomeLab Vault → New Secure Note Title: Backup Strategy Content: Primary
Backup: Kopia → pCloud - Schedule: Daily 3 AM - Retention: 30 days daily, 12 weeks
weekly - Encryption: AES-256 (password below) - pCloud: 7TB lifetime account Kopia
Encryption Password: [create strong password] (This encrypts backups before upload)
GitHub: homelab-configs - Repository: private - All configs backed up - SSH key: stored
in 1Password What's Backed Up: ■ Docker configs & compose files ■ Scripts & automation
■ Documentation & logs ■ Photos (Immich) ■ Important camera clips ■ Database exports
NOT Backed Up (by design): ■ Media files (replaceable) ■ Downloads folder ■ VM disk
images ■ Old camera footage (>30 days) Disaster Recovery: 1. New hardware 2. Install
macOS 3. Restore from pCloud using Kopia 4. Run restore script 5. Time: 2-3 hours Tags:
backup, disaster-recovery, critical Click: Save
```

■ CHECKPOINT: Documentation templates created

■ Prerequisites Complete Checklist

Before proceeding to Guide 01, verify:

Accounts Created:

- ☐ Day-01-02-Foundation-and-Accounts|1Password HomeLab vault set up - ☐
Day-01-02-Foundation-and-Accounts|DuckDNS account + subdomain (yoshikeikohomelab.duckdns.org) - ☐
Day-01-02-Foundation-and-Accounts|Tailscale account created - ☐ Day-03-05-Docker-Infrastructure|Docker Hub account created - ☐ Day-06-08-Proxmox-Hypervisor|Proxmox account created (optional) - ☐
Day-13-14-Media-Stack-Plex|Plex account created - ☐ Day-15-16-Smart-Home-Integration|Home Assistant placeholder created - ☐ Cloudflare account created (optional) - ☐ Ntfy topic created and tested - ☐
Day-22-Maintenance-and-RMM|Action1 RMM account created

Security Configured:

- ☐ 2FA enabled on critical accounts - ☐ All passwords strong and unique - ☐ Everything documented in Day-01-02-Foundation-and-Accounts|1Password - ☐ Day-01-02-Foundation-and-Accounts|1Password syncing across devices

Software Downloaded:

- ☐ Day-01-02-Foundation-and-Accounts|Tailscale installer (.pkg) - ☐
Day-03-05-Docker-Infrastructure|OrbStack (.dmg) - ☐ Day-06-08-Proxmox-Hypervisor|Proxmox ISO (~1GB) - ☐
Ubuntu Server ISO (~2GB) - ☐ Windows 11 VHDX (~10GB) - ☐ Kali Linux ISO (~4GB) - ☐ Total: ~30-40GB downloaded

Mobile Apps:

- ☐ Day-01-02-Foundation-and-Accounts|1Password installed on phone - ☐
Day-01-02-Foundation-and-Accounts|Tailscale installed on phone - ☐ Ntfy installed and subscribed - ☐
Day-13-14-Media-Stack-Plex|Plex app installed - ☐ Day-15-16-Smart-Home-Integration|Home Assistant app installed

Documentation:

- ☐ Network configuration documented - ☐ Service URLs template created - ☐ Backup strategy documented - ☐
Day-01-02-Foundation-and-Accounts|1Password vault organized

Ready to Proceed:

- ☐ M4 Mac Mini - Complete System Overview|M4 Mac Mini ready to unbox - ☐ External 4TB SSD ready - ☐
Network cables ready - ☐ Monitor, keyboard, mouse ready - ☐ All downloads transferred to USB if needed

■ GUIDE 00 COMPLETE!

↑ Back to Top

■ ■ Volume 02: M4 Mac Mini Hardware & macOS Setup (Days 1-2)

Source: Day-01-02-M4-Mac-Mini-Setup Tags: #homelab #hardware #m4 #network #setup

■ Volume 02: M4 Mac Mini - Complete System Overview|M4 Mac Mini Hardware & macOS Setup (Days 1-2)

Setting Up Your M4 Mac Mini - Complete System Overview|M4 Mac Mini for 24/7 HomeLab Operation

> "First, you must build the foundation. Then, you can build the tower." ■■

■ Volume Overview

Days: 1-2 (overlaps with Volume 01) Time Required: 4-6 hours Coffee Required: ■■■■ Difficulty: Beginner to Intermediate

What You'll Complete

By end of Day 2, you'll have: - ■ M4 Mac Mini - Complete System Overview|M4 Mac Mini unboxed and physically set up - ■ macOS Sequoia installed and configured - ■ All 3 network adapters configured (M4 Mac Mini - Complete System Overview|10GbE + 5GB + 2.5GB) - ■ Static IP address set (192.168.50.10) - ■ 4TB external SSD formatted and mounted - ■ SSH enabled for remote access - ■ Essential tools installed (Homebrew, btop, etc.) - ■ macOS optimized for 24/7 server operation - ■ Automatic login configured - ■ System ready for Day-03-05-Docker-Infrastructure|Docker and Day-06-08-Proxmox-Hypervisor|Proxmox

■ Guides in This Volume

- Guide 01: Hardware Unboxing & Physical Setup - Guide 02: macOS Initial Setup - Guide 03: Network Configuration (3 Adapters) - Guide 04: Storage Configuration - Guide 05: SSH & Remote Access - Guide 06: Essential Tools Installation - Guide 07: macOS Optimization for Servers

Guide 01: Hardware Unboxing & Physical Setup

■ Time: 30 minutes ■ Coffee: 1 cup ■ Difficulty: Beginner

■ Prerequisites

Before unboxing, verify you have:

From Volume 01:

- [] All accounts created in Day-01-02-Foundation-and-Accounts|1Password - [] Software downloads completed - [] Mobile apps installed on Samsung S25 Ultra

Physical Items Ready:

- [] M4 Mac Mini - Complete System Overview|M4 Mac Mini (sealed box) - [] 4TB External SSD (Samsung T7 or similar) - [] Day-02-M4-Mac-Mini-Setup|TP-Link SX1008 M4 Mac Mini - Complete System Overview|10GbE Switch - [] M4 Mac Mini - Complete System Overview|10GbE Ethernet cable (Cat6a or better) - [] 5GB USB-C Ethernet adapter - [] 2.5GB Ethernet adapter - [] Additional Ethernet cables - [] Monitor with HDMI/USB-C cable - [] Keyboard (USB or Bluetooth) - [] Mouse (USB or Bluetooth) - [] Power outlet with surge protector

■ Unboxing M4 Mac Mini - Complete System Overview|M4 Mac Mini

Step 1: Unbox Carefully

1. Open M4 Mac Mini - Complete System Overview|M4 Mac Mini box:

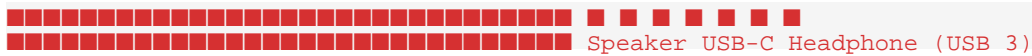
- Remove outer sleeve - Lift top of box - Inside: M4 Mac Mini, power cable, documentation

2. Inspect contents: - [] M4 Mac Mini - Complete System Overview|M4 Mac Mini unit (silver/space gray) - [] Power cable (UK plug) - [] Documentation - [] Apple stickers (because why not! ■)

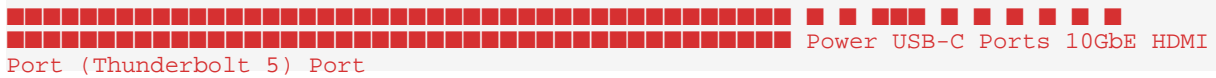
3. Check for damage: - [] No dents or scratches on chassis - [] Ports look clean and undamaged - [] No rattling sounds when gently shaken

Step 2: Understand M4 Mac Mini - Complete System Overview|M4 Mac Mini Ports

Front of M4 Mac Mini - Complete System Overview|M4 Mac Mini:



Back of M4 Mac Mini - Complete System Overview|M4 Mac Mini:



Ports breakdown: - Power: Kettle-style power inlet - 3x Thunderbolt 5/USB-C ports (back): - Support: USB 4, Thunderbolt 5, DisplayPort, charging - Speed: Up to 120 Gbps (Thunderbolt 5) - Use for: External SSD, USB-C Ethernet adapters, displays - 2x USB-A ports (front): - USB 3.1 Gen 2 (10 Gbps) - Use for: Keyboard, mouse, USB drives - 1x 10 Gigabit Ethernet port (back): - RJ45 connector - Supports: M4 Mac Mini - Complete System Overview|10GbE, 5GbE, 2.5GbE, 1GbE, 100Mbps (auto-negotiates) - This is your PRIMARY network connection! - 1x HDMI port (back): - HDMI 2.1 - Supports: 8K @ 60Hz or 4K @ 240Hz - Use for: Monitor during setup - 1x 3.5mm headphone jack (front): - High-impedance headphone support - We won't use this for homelab

■ CHECKPOINT: Understand all ports and their purposes

■ Physical Setup & Cable Management

Step 1: Choose Location

Ideal location for 24/7 operation:

Requirements: - [] Near router/network switch (within 2 meters for M4 Mac Mini - Complete System Overview|10GbE cable) - [] Good ventilation (M4 needs airflow even though it's quiet) - [] Accessible power outlet - [] Desk/shelf that won't be disturbed - [] Away from heat sources - [] Reasonably dust-free area

NOT recommended: - ■ Inside closed cabinet (heat buildup) - ■ Under cushions/fabric (blocks vents) - ■ Direct sunlight - ■ Extremely dusty area

Ventilation notes:

M4 Mac Mini ventilation: - Intake: Bottom (circular intake holes) - Exhaust: Back (through port area) - Keep 5cm clearance on all sides - Bottom must not be blocked!

Step 2: Connect External Storage

1. Prepare 4TB External SSD:

Samsung T7 (or similar) - Connect USB-C cable to SSD - Will connect to M4 in a moment

2. Choose USB-C port:

Best practice: Use LEFTMOST Thunderbolt port (back) Why: Fastest, most reliable, keeps cable tidy

Don't connect yet - wait for macOS setup!

Step 3: Connect Network Adapters

You have 3 Ethernet adapters total:

Primary: Built-in M4 Mac Mini - Complete System Overview|10GbE (ALWAYS USE THIS)

Location: Back of M4 (RJ45 port) Speed: 10 Gbps Connect to: TP-Link SX1008 10GbE Switch Port 1 Cable: Cat6a or Cat7 (2m or less for best performance) Use: Main network connection for everything

Connect now: 1. Plug Cat6a/Cat7 cable into M4's M4 Mac Mini - Complete System Overview|10GbE port (back)
2. Other end into TP-Link switch Port 1 3. Verify: Cable clicks into place, lights may blink

Secondary: 5GB USB-C Ethernet Adapter (OPTIONAL - For Later)

Speed: 5 Gbps (5000 Mbps) Connect to: USB-C Thunderbolt port (middle or right) Use case options: - Failover/backup network connection - Direct connection to second switch - Isolated management network - VM bridge network

Don't connect yet - configure in Guide 03

Tertiary: 2.5GB Ethernet Adapter (OPTIONAL - For Later)

Speed: 2.5 Gbps (2500 Mbps) Connect to: USB-C Thunderbolt port or USB-A (check adapter type) Use case options: - Dedicated Proxmox management network - Isolated VM network (security testing with Kali) - Guest network - Lab network

Don't connect yet - configure in Guide 03

For now: Only M4 Mac Mini - Complete System Overview|10GbE connected to switch!

Step 4: Connect Display & Input

1. Connect Monitor:

HDMI cable from monitor → M4 HDMI port (back) OR USB-C/Thunderbolt cable → USB-C port (if monitor supports) Turn on monitor

2. Connect Keyboard:

USB keyboard → Front USB-A port OR Bluetooth keyboard → Pair after macOS boots

3. Connect Mouse:

USB mouse → Front USB-A port OR Bluetooth mouse → Pair after macOS boots

Step 5: Connect Power

1. Plug power cable into M4:

Kettle-style power inlet on back Pushed in firmly until seated

2. Plug other end into surge protector:

Verify surge protector is ON

3. M4 powers on automatically when plugged in!

You'll hear: Soft startup chime You'll see: Apple logo on monitor

If it doesn't start: - Check power cable fully inserted both ends - Check surge protector is on - Check monitor input is set correctly

■ CHECKPOINT: M4 is powered on and booting

■ First Boot & Setup Assistant

macOS Sequoia Setup Assistant Walkthrough

The M4 will boot to setup assistant. Follow these steps exactly:

1. Select Country/Region

Screen: "Select Your Country or Region" Action: - Select: United Kingdom - Click: Continue

■ Verified: UK selected

2. Written and Spoken Languages

Screen: "Select Your Languages" Primary language: English (UK) Voice control language: English (UK) Click: Continue

■ Verified: English UK selected

3. Accessibility

Screen: "Accessibility" Question: Do you need accessibility features? Action: - Select: Not Now (unless you need them) - Click: Continue Note: Can enable later in System Settings

■ Verified: Accessibility configured

4. Connect to Network

Screen: "Connect to a Wi-Fi Network" CRITICAL: See your 10GbE cable connected? Action: - Look for: Ethernet connection detected - Should show: "Connected via Ethernet" - If not showing Ethernet: - Check cable connections - Select WiFi temporarily (can remove later) - Click: Continue

Expected: Ethernet (M4 Mac Mini - Complete System Overview|10GbE) detected automatically Shows as: "Ethernet" or "Wired Connection"

■ Verified: Network connected

5. Data & Privacy

Screen: "Data & Privacy" Information about how Apple uses your data Action: - Read if interested (or skip) - Click: Continue

■ Verified: Privacy info acknowledged

6. Migration Assistant

Screen: "Transfer Information to This Mac" Question: Transfer from another Mac? Action: - Select: Not Now - Click: Continue Why: Fresh install is cleaner for servers Note: We'll manually copy what we need later

■ Verified: No migration

7. Sign in with Apple ID

Screen: "Sign In with Your Apple ID" Action - YOUR CHOICE: Option A (Recommended): Sign in - Email: steve-smithit@outlook.com - Password: [your Apple ID password] - Benefits: iCloud, App Store, easier setup - Continue with 2FA if enabled Option B: Set Up Later - Click: "Set Up Later" - Confirm: "Skip" - Why: More private, but need Apple ID for App Store

My recommendation: Sign in with Apple ID Why: Needed for App Store, easier management

If signing in:

- Enter Apple ID: steve-smithit@outlook.com - Enter password - Complete 2FA on Samsung S25 Ultra - Click: Continue

■ Verified: Apple ID configured

8. Terms and Conditions

Screen: "Terms and Conditions" Action: - Click: Agree - Confirm: Agree

■ Verified: Terms accepted

9. Create Computer Account

Screen: "Create a Computer Account" CRITICAL - SAVE THESE IN 1PASSWORD IMMEDIATELY! Full name: Steve Account name: steve (will auto-fill) Password: [CREATE STRONG PASSWORD] Hint: (optional, but consider: "Stored in 1Password") Click: Continue

STOP HERE - SAVE TO 1PASSWORD:

Open 1Password on Samsung S25 Ultra: 1Password → HomeLab Vault → New Login Title: M4 Mac Mini - Steve User Account Username: steve Password: [the password you just created] Add Custom Fields: - Full Name: Steve - Computer Name: (will set in next step) - macOS Version: Sequoia 15.x - IP Address: 192.168.50.10 (will configure) - SSH: ssh steve@192.168.50.10 - Tailscale IP: (will configure later) Notes: Main user account on M4 Mac Mini Admin privileges Used for all homelab management Tags: m4, macos, admin, homelab SAVE IMMEDIATELY!

■ Verified: Password saved in Day-01-02-Foundation-and-Accounts|1Password before continuing

Click: Continue in macOS setup

10. Enable Location Services

Screen: "Enable Location Services" Action: - Select: Enable Location Services - Click: Continue Why: Useful for time zone, weather, etc. Privacy: Can disable specific apps later

■ Verified: Location services enabled

11. Select Time Zone

Screen: "Select Your Time Zone" Should auto-detect: Europe/London (GMT) Verify: - Time zone: London - City: London - Time: Should show correct time Click: Continue

■ Verified: Time zone correct

12. Analytics

Screen: "Analytics" Question: Share analytics with Apple? Action - YOUR CHOICE: Recommended for server: Uncheck all - Share Mac Analytics: ■ - Share iCloud Analytics: ■ - Share crash and usage data with app developers: ■ Why: Less data sent, better for server operation Click: Continue

■ Verified: Analytics preferences set

13. Screen Time

Screen: "Screen Time" Action: - Select: Set Up Later - Click: Continue Why: Not needed for server operation Note: This is for monitoring app usage, parental controls

■ Verified: Screen Time skipped

14. Siri

Screen: "Enable Ask Siri" Action: - Uncheck: Enable Ask Siri - Click: Continue Why: Not needed for headless server operation Note: Siri uses resources we'd rather dedicate to services

■ Verified: Siri disabled

15. FileVault Disk Encryption

Screen: "Encrypt Your Mac" CRITICAL DECISION: For HomeLab Server: - Select: Don't encrypt - Click: Continue - Confirm: "Don't Encrypt" Why? - Easier disaster recovery - Remote boot/restart without password entry - Performance (encryption adds tiny overhead) - Physical security: Keep M4 in secure location - Our data security: Individual service encryption (Docker, VMs) Note: If M4 were a laptop (risk of theft), enable FileVault For home server in secure location, not needed

■■ IMPORTANT CONSIDERATION:

Encrypt if: - M4 is portable/might be stolen - Required by compliance/work policy - Storing extremely sensitive data

Don't encrypt if: - M4 stays in secure home location (recommended) - Want easy recovery and remote management - Prioritize convenience over disk encryption

For homelab: Don't encrypt (easier management)

■ Verified: Encryption decision made

16. Touch ID Setup

Screen: "Set Up Touch ID" Action: - Click: Set Up Later - Confirm: Continue Why: M4 Mac Mini doesn't have built-in Touch ID Note: Can add Magic Keyboard with Touch ID later if wanted

■ Verified: Touch ID skipped

17. Choose Your Look

Screen: "Choose Your Look" Appearance options: - Light - Dark - Auto (switches based on time of day) Recommendation: - Select: Dark - Why: Easier on eyes, looks cool, uses less display power Click: Continue

■ Verified: Appearance selected

18. Setup Complete!

Screen: Welcome animation macOS finishes setup Desktop loads You're in!

■ macOS Sequoia is installed and ready!

■ First Boot Complete!

At this point you should see: - macOS desktop (Dock at bottom, menu bar at top) - Wallpaper (Sequoia default background) - Empty desktop or default icons

Verify basics: - [] Mouse cursor moves - [] Keyboard types - [] Monitor display is clear - [] Network connected (see WiFi/Ethernet icon in menu bar)

■■ Initial System Configuration

Set Computer Name

Make it easy to identify on network!

1. Open System Settings:

Click: Apple menu (top left) Click: System Settings

2. Navigate to Sharing:

Left sidebar → General → Sharing

3. Set Computer Name:

Local hostname: m4-homelab Computer Name will auto-update to: m4-homelab.local This is your hostname on local network!

4. Verify hostname:

Open Terminal: - Applications → Utilities → Terminal - OR: Cmd+Space, type "terminal", Enter Type: hostname Press: Enter Should show: m4-homelab.local

Save to Day-01-02-Foundation-and-Accounts|1Password (update existing entry):

Update: M4 Mac Mini - Steve User Account Add/Update Custom Field: - Computer Name: m4-homelab - Local Hostname: m4-homelab.local Save

■ CHECKPOINT: Computer name set to m4-homelab

■ Connect External 4TB SSD

Now that macOS is running, connect your storage:

Step 1: Connect SSD

1. Plug USB-C cable from 4TB SSD into M4:

Use: Leftmost Thunderbolt port (back of M4) Connection: Should be USB-C to USB-C

2. SSD should mount automatically:

You'll see: SSD icon appear on desktop OR: In Finder sidebar under "Locations" Name: Probably "Samsung T7" or "Untitled"

3. If it doesn't appear:

- Check cable connections - Try different Thunderbolt port - Check SSD has power LED lit
- Open Finder → Sidebar should show external drive

■ CHECKPOINT: 4TB SSD connected and detected

Step 2: Check Current Format

We need to verify/reformat for optimal macOS performance:

1. Open Disk Utility:

Applications → Utilities → Disk Utility OR: Cmd+Space, type "disk utility", Enter

2. Select your 4TB SSD:

Left sidebar → External section Click: Your 4TB SSD (should show ~4TB capacity)

3. Check current format:

Look at top info section: Format: Probably "exFAT" or "NTFS" or "APFS"

Current format determines what we do next:

If Format is APFS (Apple File System):

■ PERFECT! Already optimal for macOS. Skip to "Configure External SSD" below.

If Format is exFAT or FAT32 or NTFS:

■■ NEED TO REFORMAT to APFS for best performance CRITICAL: This will ERASE the SSD! - Make sure it's empty or data is backed up - We want fresh start anyway

To Reformat to APFS:

1. In Disk Utility, with SSD selected:

Click: Erase (top toolbar)

2. Erase dialog appears:

Name: External4TB Format: APFS Scheme: GUID Partition Map Click: Erase

3. Confirmation:

"Are you sure you want to erase...?" Click: Erase

4. Wait for format:

Takes: 30-60 seconds Progress bar appears When done: "Erase process complete" Click: Done

5. SSD remounts automatically:

Desktop icon reappears: "External4TB" Format now: APFS

■ CHECKPOINT: 4TB SSD formatted as APFS

Step 3: Configure External SSD

Set it to mount automatically on boot:

1. Open Terminal:

Applications → Utilities → Terminal

2. Get SSD UUID:

```
diskutil info /Volumes/External4TB | grep UUID
```

Copy the UUID (looks like: 12345678-ABCD-1234-ABCD-123456789ABC)

3. Add to /etc/fstab (auto-mount):

```
# Open fstab for editing (create if doesn't exist) sudo nano /etc/fstab # Password: Your
Mac password # Add this line (replace UUID_HERE with your UUID): UUID=UUID_HERE
/Volumes/External4TB apfs rw,auto # Save: Ctrl+X, Y, Enter
```

4. Verify:

```
# Unmount diskutil unmount /Volumes/External4TB # Remount (tests auto-mount) diskutil
mount /Volumes/External4TB # Should mount automatically
```

Save to Day-01-02-Foundation-and-Accounts|1Password:

```
1Password → HomeLab Vault → Secure Note Title: External 4TB SSD Configuration Content:
Name: External4TB Format: APFS Mount: /Volumes/External4TB UUID: [your UUID]
Connection: Thunderbolt (leftmost port) Use: Main storage for homelab - VMs:
/Volumes/External4TB/VMs - Media: /Volumes/External4TB/Media - Backups:
/Volumes/External4TB/Backups - Docker data: /Volumes/External4TB/Docker-Data Tags:
storage, configuration, homelab Save
```

■ CHECKPOINT: 4TB SSD configured and auto-mounting

■ Create Folder Structure

Organize your external SSD now:

```
# Open Terminal # Create main directories mkdir -p /Volumes/External4TB/VMs mkdir -p
/Volumes/External4TB/Media/{Movies,TV,Music,Photos,Comics} mkdir -p
/Volumes/External4TB/Downloads mkdir -p /Volumes/External4TB/Backups mkdir -p
/Volumes/External4TB/Docker-Data mkdir -p /Volumes/External4TB/ISOs mkdir -p
/Volumes/External4TB/Frigate # Create symlinks to home directory for easy access ln -s
/Volumes/External4TB ~/External4TB ln -s /Volumes/External4TB/Media ~/Media ln -s
/Volumes/External4TB/VMs ~/VMs # Verify structure ls -la /Volumes/External4TB/
```

You should see:

```
/Volumes/External4TB/ ■■■ Backups/ ■■■ Docker-Data/ ■■■ Downloads/ ■■■ Frigate/ ■■■
ISOs/ ■■■ Media/ ■ ■■■ Movies/ ■ ■■■ TV/ ■ ■■■ Music/ ■ ■■■ Photos/ ■ ■■■ Comics/
■■■ VMs/
```

■ CHECKPOINT: Folder structure created

■ Guide 01 Complete Checklist

Before proceeding to Guide 02, verify:

Hardware: - [] M4 Mac Mini - Complete System Overview|M4 Mac Mini unboxed and inspected - [] All cables connected (M4 Mac Mini - Complete System Overview|10GbE, HDMI, power) - [] 4TB external SSD connected - [] Monitor, keyboard, mouse working - [] M4 powered on and running

macOS Setup: - [] macOS Sequoia installed - [] User account created (steve) - [] Password saved in Day-01-02-Foundation-and-Accounts|1Password - [] Computer name set (m4-homelab) - [] Time zone correct (Europe/London) - [] Network connected (M4 Mac Mini - Complete System Overview|10GbE working)

Storage: - [] 4TB SSD formatted to APFS - [] Named "External4TB" - [] Auto-mounting configured - [] Folder structure created - [] Symlinks to home directory created

Documentation: - [] Account info updated in Day-01-02-Foundation-and-Accounts|1Password - [] SSD configuration saved - [] Computer name noted

■ Congratulations! Your M4 Mac Mini - Complete System Overview|M4 Mac Mini is physically set up and macOS is installed!

Next: Guide 02 - Network Configuration (setting up all 3 ethernet adapters)

Take a 5-minute break! ■

End of Guide 01

↑ Back to Top

■ Day 02 BONUS: TP-Link Deco XE75 Mesh Setup

Source: Day-02-BONUS-Deco-Mesh-Setup Tags: #homelab #network #wifi #mesh #deco

Day 02 BONUS: TP-Link Day-03-Deco-Mesh-WiFi|Deco XE75 Mesh Setup

■■ Time: 45-60 minutes ■ Coffee: 2 cups ■ Difficulty: Intermediate ■ When: After Day 02 (M4 setup complete)

> "Looks like I picked the right week to setup a mesh network!" ■

■ What You're Building

Problem: Sky router WiFi = single point, limited coverage, interference Solution: Professional mesh network with wired backhaul

You'll have: - 3 TP-Link Day-03-Deco-Mesh-WiFi|Deco XE75 access points - Wired Ethernet backhaul (via TP-Link switch) - Full house coverage - Sky router optimized (gateway-only, WiFi disabled) - Separate IoT network for 43+ smart devices

■ Prerequisites

Hardware: - [] 3x TP-Link Day-03-Deco-Mesh-WiFi|Deco XE75 units (unboxed) - [] 3x Ethernet cables (Cat5e minimum) - [] Day-02-M4-Mac-Mini-Setup|TP-Link SX1008 switch (from Day 01) - [] Sky router access (admin login)

Before Starting: - [] M4 Mac Mini - Complete System Overview|M4 Mac Mini setup complete (Day 02) - [] M4 on M4 Mac Mini - Complete System Overview|10GbE wired (192.168.50.10) - [] TP-Link switch connected to Sky router - [] Deco app installed on Samsung S25 Ultra

Part 1: Disable Day-03-Deco-Mesh-WiFi|Sky Router WiFi

Why?

- Eliminates interference with M4 Mac Mini - Complete System Overview|10GbE - Better performance (router does routing only) - No device confusion (one network)

Steps:

1. Login to Day-03-Deco-Mesh-WiFi|Sky Router:

```
URL: http://192.168.50.1 Username: admin Password: [from router label]
```

2. Navigate: Wireless → Wireless Settings

3. Disable both bands: - 2.4GHz WiFi: OFF - 5GHz WiFi: OFF - Save/Apply

4. Verify: WiFi networks disappear on phone

5. Test: M4 still has internet (wired M4 Mac Mini - Complete System Overview|10GbE)

```
ping 8.8.8.8
```

■ CHECKPOINT: Sky WiFi off, M4 wired working

Part 2: Position Deco Units

Recommended Layout:

Deco #1 - Main (Living Room): - Central location - Connect to: TP-Link Switch Port 2 - Covers: Living room, kitchen, front

Deco #2 - Office: - Home office - Connect to: TP-Link Switch Port 3 - Covers: Office, back garden

Deco #3 - Bedroom: - Upstairs/main bedroom - Connect to: TP-Link Switch Port 4 - Covers: All bedrooms

Physical Setup:

1. Place each Deco unit 2. Run Ethernet cable to switch 3. Plug in power 4. Wait for pulsing blue LED (~2 min)

Switch Layout:

```
TP-Link SX1008: Port 1: M4 Mac Mini (10GbE) Port 2: Deco #1 (Main) Port 3: Deco #2 (Office) Port 4: Deco #3 (Bedroom) Port 5-7: Available Port 8: Uplink to Sky Router
```

■ CHECKPOINT: All 3 Decos powered, pulsing blue

Part 3: Setup Main Deco

On Samsung S25 Ultra:

1. Open Deco app

2. Create/login TP-Link account: - Email: steve-smithit@outlook.com - Password: [generate strong one] - Save to Day-01-02-Foundation-and-Accounts|1Password immediately!

3. Add first Deco: - Tap: + - Select: Day-03-Deco-Mesh-WiFi|Deco XE75 - Scan QR code (bottom of Deco #1) - Location: Living Room

4. Configure network:

```
WiFi Name: SteveHomeNet Password: [16+ char password] Save to 1Password: Title: Home WiFi - SteveHomeNet Type: Wireless Router SSID: SteveHomeNet Password: [your password] Security: WPA3
```

5. CRITICAL - Operation Mode:

```
Select: Access Point Mode (NOT Router Mode!) Why? Sky router is already your router. Decos provide WiFi only.
```

6. Wait for setup: 3-5 minutes - LED: Pulsing blue → Solid white - Solid white = SUCCESS!

7. Test: Connect phone to SteveHomeNet

■ CHECKPOINT: Main Deco working, phone connected

Part 4: Add Satellite Decos

Add Deco #2 (Office):

1. In Deco app: Tap + 2. Select: Add Another Deco 3. Scan QR code (Deco #2) 4. Location: Office 5. Wait 2-3 minutes 6. Verify: Shows "Wired" in app (Ethernet icon)

Add Deco #3 (Bedroom):

1. Repeat same process 2. Scan QR code (Deco #3) 3. Location: Bedroom 4. Verify: Shows "Wired" in app

Verify All 3:

In app you should see: - All 3 Decos: Connected - All 3 showing: Wired (Ethernet icons) - All 3 LEDs: Solid white - Network name: SteveHomeNet

■ CHECKPOINT: All 3 Decos operational with wired backhaul

Part 5: Advanced Configuration

Address Reservation:

```
More → Advanced → Address Reservation Add: - M4 Mac Mini: 192.168.50.10 - Brother  
Printer: 192.168.50.201 - Windows VM: 192.168.50.52 (future) - Ubuntu VM: 192.168.50.51  
(future)
```

DHCP Range:

```
More → Advanced → DHCP Server Start: 192.168.50.100 End: 192.168.50.199 (Keeps .1-.99  
for static, .200-254 for IoT)
```

Enable Features:

```
More → Advanced Fast Roaming: ON (seamless handoff) Beamforming: ON (better signal)  
Network Optimization: ON (auto channel)
```

IoT Network (Recommended):

```
More → Advanced → IoT Network Enable: ON SSID: SteveHomeNet-IoT Password: [different from main] Frequency: 2.4GHz only Security: WPA2 Use for: Nest cameras, Sonos, smart plugs (43+ IoT devices) Save to 1Password!
```

Part 6: Migrate All Devices

Critical Devices First:

1. MacBook Air → SteveHomeNet 2. iPad Pro → SteveHomeNet 3. M2 Mac Mini → SteveHomeNet

Smart Home (use IoT network):

5x Nest Cameras: - Google Home app - Each camera → WiFi settings - Connect to: SteveHomeNet-IoT

10x Sonos Speakers: - Sonos app - Settings → System → Network - Update WiFi → SteveHomeNet-IoT - Takes 5-10 min for all

Other Smart Devices: - Nest thermostat → SteveHomeNet-IoT - Nest Minis → SteveHomeNet-IoT - Meross plugs → SteveHomeNet-IoT - Easee charger → SteveHomeNet-IoT

Gaming/Entertainment:

- 2x PS5 → SteveHomeNet - 3x Xbox → SteveHomeNet - Smart TVs → SteveHomeNet

Cars:

- Tesla Model Y → SteveHomeNet - Cupra Born → SteveHomeNet

VERIFY M4 STILL WIRED:

```
# SSH to M4 ssh steve@192.168.50.10 # Check interface ifconfig en0 | grep "10Gbaset" # Should show: 10Gbaset # This confirms M4 on 10GbE, NOT WiFi!
```

■ CHECKPOINT: All 43+ devices connected, M4 still wired

Part 7: Coverage Testing

Walk Test:

Living Room (Deco #1): - Signal: Excellent - Speed: 400-500 Mbps - Connected to: Deco #1

Office (Deco #2): - Signal: Excellent - Speed: 400-500 Mbps - Connected to: Deco #2

Bedroom (Deco #3): - Signal: Excellent - Speed: 400-500 Mbps - Connected to: Deco #3

Roaming Test: - Walk living room → office → bedroom - Should seamlessly switch Decos - No disconnections

Outdoor: - Front garden: Good signal - Back garden: Good signal - Driveway: Good signal

Part 8: Backup & Maintenance

Create Backup:

More → System → Backup & Restore Create Backup Download to phone Upload to 1Password as document Title: Deco Mesh Configuration Backup Type: Document File: deco-backup-YYYYMMDD.bin

Enable Auto-Update:

More → System → Firmware Update Auto-update: ON Check monthly Updates take ~5 minutes

Troubleshooting

Deco Won't Connect:

1. Check Ethernet cable plugged in 2. Check switch port has link light 3. Power cycle Deco (unplug 10s) 4. Factory reset: Hold reset button 10s

Slow WiFi:

1. Verify wired backhaul in app (Ethernet icon) 2. Update firmware 3. Run network optimization 4. Reboot all Decos

Device Can't Connect:

1. Correct password (case-sensitive)
2. Try 2.4GHz IoT network if 5GHz fails
3. Forget network and reconnect
4. Check not blocked in Deco app

M4 Using WiFi Instead of Wired:

```
# Check interface priority networksetup -listnetworkserviceorder # Ethernet should be first # If WiFi first, reorder: networksetup -ordernetworkservices "Ethernet" "Wi-Fi"
```

Verification Checklist

Day-03-Deco-Mesh-WiFi|Sky Router: - ☐ WiFi disabled (both bands) - ☐ Still routing traffic - ☐ Accessible at 192.168.50.1

Deco Mesh: - ☐ All 3 units solid white LED - ☐ All 3 showing wired backhaul in app - ☐ Network: SteveHomeNet - ☐ IoT network: SteveHomeNet-IoT - ☐ Settings backed up to Day-01-02-Foundation-and-Accounts|1Password

Devices: - ☐ All 43+ devices reconnected - ☐ Smart home on IoT network - ☐ M4 still on wired M4 Mac Mini - Complete System Overview|10GbE - ☐ Full coverage everywhere

Performance: - ☐ WiFi speeds 400-500 Mbps - ☐ Seamless roaming working - ☐ No dead zones - ☐ Low latency

What You've Achieved

Before: - Sky router WiFi (single point) - Limited coverage (~70%) - WiFi interference with M4 Mac Mini - Complete System Overview|10GbE - 43+ devices on one AP

After: - Professional mesh (3 access points) - Full coverage (100%) - Wired backhaul (maximum speed) - Separate IoT network - No M4 Mac Mini - Complete System Overview|10GbE interference - Sky router optimized

Network Stats:

```
Coverage: 100% (was ~70%) Speed: 400-500 Mbps everywhere Latency: <5ms on mesh Devices: 43+ all connected Reliability: Excellent
```

■ BONUS GUIDE COMPLETE!

Next Steps: - Continue with Day 03 (Day-03-05-Docker-Infrastructure|Docker Infrastructure) - Or proceed through your remaining volumes - Network is now OPTIMIZED! ■

"Surely you can't be serious about this amazing network?" "I am serious. And don't call me Shirley." ■■

↑ Back to Top

=====

■ ■ Volume 02: M4 Mac Mini Setup (Days 1-2) - COMPLETE GUIDE

Source: Day-02-M4-Mac-Mini-Setup Tags: #homelab #hardware #m4 #network #setup

■ Volume 02: M4 Mac Mini - Complete System Overview|M4 Mac Mini Setup (Days 1-2) - COMPLETE GUIDE

Setting Up Your M4 Mac Mini - Complete System Overview|M4 Mac Mini for 24/7 HomeLab Operation

> "The foundation determines the height of the building!" ■■

■ Volume Overview

Days: 1-2 (overlaps with Volume 01) Time Required: 4-6 hours Coffee Required: ■■■■ Difficulty: Beginner to Intermediate

What You'll Complete

By end of this volume, you'll have: - ■ M4 Mac Mini - Complete System Overview|M4 Mac Mini physically set up and connected - ■ macOS Sequoia installed and optimized for 24/7 operation - ■ All 3 network adapters configured (M4 Mac Mini - Complete System Overview|10GbE + 5GB + 2.5GB) - ■ Static IP configured (192.168.50.10) - ■ 4TB external SSD formatted and mounted - ■ SSH enabled for remote management - ■ Essential tools installed - ■ Ready for Day-03-05-Docker-Infrastructure|Docker and Day-06-08-Proxmox-Hypervisor|Proxmox installation

■ Guides in This Volume

- Guide 01: Hardware Unboxing & Assembly - Guide 02: First Boot & macOS Setup - Guide 03: Network Configuration - Guide 04: Storage Configuration - Guide 05: macOS Optimization for 24/7 - Guide 06: Essential Tools Installation - Guide 07: System Verification

Guide 01: Hardware Unboxing & Assembly

■ Time: 30 minutes ■ Coffee: 1 cup ■ Difficulty: Beginner

■ Prerequisites Checklist

Before unboxing, verify you have:

Hardware Ready:

- ☐ M4 Mac Mini - Complete System Overview|M4 Mac Mini box (unopened) - ☐ 4TB External SSD (Samsung T7 or similar) - ☐ M4 Mac Mini - Complete System Overview|10GbE cable (Cat6a or better) - ☐ 5GB USB-C Ethernet adapter - ☐ 2.5GB Ethernet adapter - ☐ Monitor with HDMI/DisplayPort - ☐ Keyboard (USB or Bluetooth) - ☐ Mouse (USB or Bluetooth) - ☐ Power strip with surge protection

Network Ready:

- ☐ Day-02-M4-Mac-Mini-Setup|TP-Link SX1008 M4 Mac Mini - Complete System Overview|10GbE switch powered on - ☐ Router accessible (192.168.50.1) - ☐ Available Ethernet ports on switch - ☐ Network cables ready

Workspace:

- ☐ Clean, static-free surface - ☐ Good lighting - ☐ Power outlet nearby - ☐ Enough space for unboxing

■ Unboxing the M4 Mac Mini - Complete System Overview|M4 Mac Mini

Step 1: Open the Box

1. Place box on clean surface 2. Remove plastic wrap 3. Lift lid carefully

Step 2: Contents Verification

Check you have: - [] M4 Mac Mini - Complete System Overview|M4 Mac Mini unit - [] Power cable (UK plug) - [] Apple stickers (essential! ■) - [] Quick start guide - [] Warranty information

Step 3: Inspect the Unit

Check for: - [] No visible damage - [] All ports clean and undamaged - [] Serial number visible on bottom - [] Feet/rubber pads intact

■ Save to Day-01-02-Foundation-and-Accounts|1Password:

```
1Password → HomeLab Vault → New Secure Note Title: M4 Mac Mini - Hardware Info
Content: Serial Number: [found on bottom of unit] Model: Mac Mini (M4, 2024) Specs: -
CPU: M4 (10-core) - RAM: 32GB unified memory - Storage: 512GB SSD - Network: 10GbE
built-in Purchase Date: [your date] Purchase Location: [Apple Store/Online] Warranty
Expires: [date + 1 year] Tags: hardware, m4, homelab Save
```

■ CHECKPOINT: M4 Mac Mini - Complete System Overview|M4 Mac Mini unboxed and inspected

■ Physical Setup

Step 1: Choose Location

Requirements: - ■ Good ventilation (at least 2" clearance on all sides) - ■ Cool, dry location - ■ Away from heat sources - ■ Accessible for cables - ■ Near network switch - ■ Stable surface (no vibration)

■■ AVOID: - ■ Enclosed cabinets without ventilation - ■ Direct sunlight - ■ Near radiators/heaters - ■ Dusty environments - ■ Unstable surfaces

Step 2: Connect Power

1. Plug power cable into M4 - Connector on back of unit - Should click firmly into place
2. Plug into surge protector - Use quality surge protector - Note which outlet used
3. Don't power on yet! - Connect everything first

■ CHECKPOINT: Power connected but NOT turned on

Step 3: Connect Network Adapters

You have THREE network adapters to configure:

Primary: Built-in M4 Mac Mini - Complete System Overview|10GbE

1. Take M4 Mac Mini - Complete System Overview|10GbE cable (Cat6a or better) 2. Connect to: Ethernet port on back of M4 3. Connect other end to: Day-02-M4-Mac-Mini-Setup|TP-Link SX1008 switch Port 1 4. Verify: Click when inserted, LED should light when powered on

Purpose: Primary network connection (5Gbps capable)

Secondary: 5GB USB-C Adapter

1. Connect adapter to: USB-C port on M4 (rear, either side) 2. Connect Ethernet cable to adapter 3. Connect other end to: Router directly OR switch Port 2

Purpose: Backup/failover connection OR separate management network

Tertiary: 2.5GB Adapter

1. Connect adapter to: USB-A or USB-C port on M4 2. Connect Ethernet cable to adapter 3. Connect other end to: Switch Port 3 OR separate network

Purpose: Isolated VM network OR IoT device management

Network Adapter Summary:

```
M4 Mac Mini Network Connections: ■■ en0: Built-in 10GbE (Primary) → 192.168.50.10 ■■  
en1: 5GB USB-C (Secondary) → 192.168.50.11 (backup) ■■ en2: 2.5GB (Tertiary) →  
192.168.51.1 (VM network)
```

■ CHECKPOINT: All 3 network adapters connected

Step 4: Connect External Storage

1. Take 4TB External SSD 2. Connect to: USB-C port on M4 (prefer rear port for permanent connection) 3. Verify: Should be USB-C to USB-C cable for best speed

■ TIP: Rear USB-C ports support Thunderbolt 4 (40Gbps) for maximum SSD performance

■ CHECKPOINT: External SSD connected

Step 5: Connect Display & Peripherals

Monitor:

1. Connect HDMI or DisplayPort cable 2. To: HDMI port on back of M4 (supports 4K 60Hz) 3. Power on monitor

Keyboard:

- Option 1: USB keyboard → Any USB port - Option 2: Bluetooth → Will pair during setup

Mouse:

- Option 1: USB mouse → Any USB port - Option 2: Bluetooth → Will pair during setup

■ CHECKPOINT: Display and input devices connected

Step 6: Cable Management

Before powering on, organize cables:

1. Group cables together: - Power cable - Network cables (3x) - USB cables - Display cable

2. Use cable ties or velcro straps - Don't over-tighten - Leave some slack for movement

3. Label cables (optional but recommended):

- "M4 Power" - "M4 10GbE Primary" - "M4 5GB Backup" - "M4 2.5GB VM Network" - "M4 External SSD"

4. Take photo of setup - Document cable arrangement - Save to Day-01-02-Foundation-and-Accounts\1Password or Photos - Useful for future reference

■ CHECKPOINT: Cables organized and labeled

■ First Power On

Pre-Power Checklist:

Verify before pressing power button: - ☐ Power cable connected to surge protector - ☐ Surge protector switched ON - ☐ All 3 network cables connected - ☐ External SSD connected - ☐ Monitor connected and powered on - ☐ Monitor on correct input (HDMI/DisplayPort) - ☐ Keyboard and mouse ready - ☐ No obstructions around M4 (ventilation)

Power On Sequence:

1. Press power button on M4 - Button on back right corner - Brief press (don't hold) - Should hear startup chime

2. Watch for: - ■ Apple logo on screen - Progress bar underneath - Takes 30-60 seconds

3. If nothing appears: - Check monitor is on correct input - Check HDMI/DisplayPort connection - Wait 2 minutes (first boot can be slow) - Try pressing any key on keyboard

■ CHECKPOINT: M4 powered on and Apple logo visible

■ Guide 01 Complete!

What you've accomplished: - ■ M4 Mac Mini - Complete System Overview|M4 Mac Mini unboxed and inspected - ■ Serial number documented in Day-01-02-Foundation-and-Accounts|1Password - ■ Physical location chosen (well-ventilated) - ■ Power connected via surge protector - ■ All 3 network adapters connected - ■ External 4TB SSD connected - ■ Display and peripherals connected - ■ Cables organized and labeled - ■ M4 powered on successfully

Next: Guide 02 - First Boot & macOS Setup

Guide 02: First Boot & macOS Setup

■■ Time: 1-2 hours ■ Coffee: 2 cups ■ Difficulty: Beginner

macOS Setup Assistant

The M4 will boot into Setup Assistant. Follow these steps:

Step 1: Choose Your Country or Region

Select: United Kingdom Click: Continue

■ CHECKPOINT: Country selected

Step 2: Written and Spoken Languages

Language: English (UK) Click: Continue

■ CHECKPOINT: Language selected

Step 3: Accessibility

Vision: - VoiceOver: Off - Zoom: Off - Display: Default Motor: - Pointer Control: Default Hearing: - RTT/TTY: Off Click: Not Now (unless you need accessibility features)

■ CHECKPOINT: Accessibility configured

Step 4: Select Your Wi-Fi Network

■■ IMPORTANT: Use Ethernet, not Wi-Fi!

Click: "My computer does not connect to the Internet" OR Click: "Other Network Options"
Select: "Use Ethernet"

Why? - M4 Mac Mini - Complete System Overview|10GbE is MUCH faster than Wi-Fi - More stable for server -
We'll configure static IP later

■ CHECKPOINT: Chose Ethernet connection

Step 5: Data & Privacy

Read privacy information Click: Continue

Step 6: Migration Assistant

Transfer Information to This Mac?

Select: "Not Now"

Why? - Fresh installation for homelab - No need to migrate from another Mac - Clean slate is best for server

■ CHECKPOINT: Skipped migration

Step 7: Sign In with Your Apple ID

■■ IMPORTANT DECISION:

Option A: Sign in with Apple ID (Recommended for convenience)

Email: your-apple-id@email.com Password: [your password] Two-factor code: [from iPhone]
Benefits: - iCloud integration - Find My Mac - App Store access - iCloud Keychain

Option B: Set Up Later (More private)

Click: "Set Up Later" Click: "Skip" Benefits: - More privacy - Can add later if needed -
Faster setup

■ RECOMMENDATION: Sign in with Apple ID for convenience, but disable iCloud sync for homelab data

■ CHECKPOINT: Apple ID decision made

Step 8: Terms and Conditions

Read: Terms and Conditions Click: Agree Click: Agree (confirmation)

Step 9: Create a Computer Account

■ THIS IS IMPORTANT! This is your admin account.

```
Full Name: Steve Account Name: steve (automatically filled) Password: [CREATE STRONG
PASSWORD] Hint: [memorable hint, but not obvious] Click: Continue
```

■ IMMEDIATELY SAVE TO 1PASSWORD:

```
1Password → HomeLab Vault → New Login Title: M4 Mac Mini - Admin Account Username:
steve Password: [the password you just created] Add Custom Fields: - Computer Name:
M4-HomeLab (will set this) - Local IP: 192.168.50.10 (will configure) - Hostname:
m4-homelab.local Notes: Main admin account for M4 Mac Mini 24/7 homelab server DO NOT
use this for daily browsing! Tags: m4, macos, admin, homelab Save
```

■■ CRITICAL: Write down this password on paper temporarily until saved in Day-01-02-Foundation-and-Accounts|1Password!

■ CHECKPOINT: Admin account created and saved in Day-01-02-Foundation-and-Accounts|1Password

Step 10: Enable Location Services

```
Enable Location Services: ON
```

Why? - Time zone accuracy - Find My Mac - Some apps need it

■ CHECKPOINT: Location services enabled

Step 11: Select Your Time Zone

```
Closest City: London Time Zone: (GMT) Greenwich Mean Time - London Verify time is
correct Click: Continue
```

■ CHECKPOINT: Time zone set to GMT

Step 12: Analytics

```
Share Mac Analytics with Apple: Uncheck ■ Share crash data with app developers: Uncheck
■ Why? Privacy + less background activity on server Click: Continue
```

■ CHECKPOINT: Analytics disabled

Step 13: Screen Time

```
Set Up Screen Time: Click "Set Up Later"
```

Why? Not needed for server

■ CHECKPOINT: Screen Time skipped

Step 14: Siri

Enable Ask Siri: Uncheck ■

Why? - Not needed for server - Saves resources - Privacy

■ CHECKPOINT: Siri disabled

Step 15: Choose Your Look

Appearance: - Light - Dark - Auto (switches based on time) Choose: Dark (easier on eyes for server management) Click: Continue

■ CHECKPOINT: Appearance selected

Step 16: Setting Up Your Mac

Now macOS will: - Create your account - Set up your desktop - Configure system settings - Takes 2-5 minutes

■ Coffee break!

■ CHECKPOINT: macOS setup complete, desktop loads

■ Welcome to macOS!

You should now see: - Desktop with default wallpaper - Dock at bottom - Menu bar at top - Finder icon in dock

■ Immediate Post-Setup Tasks

Task 1: Check System Information

1. Click Apple menu (top left) 2. Click: About This Mac

Verify:

Chip: Apple M4 Memory: 32 GB Startup Disk: Macintosh HD (512 GB) macOS: Sequoia 15.x

3. Take screenshot: Cmd + Shift + 3 4. Save to: Desktop 5. Upload to Day-01-02-Foundation-and-Accounts|1Password (attach to M4 Mac Mini - Complete System Overview|M4 Mac Mini entry)

■ CHECKPOINT: System specs verified

Task 2: Software Update

Always update to latest before proceeding!

1. Apple menu → System Settings 2. Click: General → Software Update 3. Click: Update Now (if available) 4. Wait for download and installation (10-30 minutes) 5. Restart when prompted

After restart, verify: - Latest macOS Sequoia version installed - All updates applied

■ CHECKPOINT: System fully updated

Task 3: Set Computer Name

Currently your Mac is named something like "Steve's Mac Mini"

Change to: M4-HomeLab

1. System Settings → General → Sharing
2. Computer Name: Change to M4-HomeLab
3. This sets:

```
Computer Name: M4-HomeLab Local Hostname: m4-homelab.local (automatic) Bonjour Name: M4-HomeLab.local
```

4. Click outside field to save

Update Day-01-02-Foundation-and-Accounts|1Password:

```
Edit M4 Mac Mini entry Computer Name: M4-HomeLab Hostname: m4-homelab.local Save
```

■ CHECKPOINT: Computer name set

Task 4: Check External SSD

1. Look on Desktop - Should see external drive icon 2. Or: Finder → Sidebar - Should see drive listed

If NOT visible: - Open Finder → Settings → Sidebar - Check: External disks ✓

Current drive name: Probably "Samsung T7" or "Untitled"

We'll format this properly in Guide 04

■ CHECKPOINT: External SSD visible

Task 5: Verify Network Connection

1. System Settings → Network

2. Should see:

```
Ethernet (Built-in 10GbE): Connected Status: Connected IP Address: 192.168.50.xxx (DHCP assigned)
```

3. Note current IP address (temporary)

Other adapters: - USB 10Gb Ethernet Adapter: May show as connected - USB 10Gb Ethernet Adapter 2: May show as connected

We'll configure these properly in Guide 03

■ CHECKPOINT: Network connected

■ Guide 02 Complete!

What you've accomplished: - ■ Completed macOS Setup Assistant - ■ Created admin account (steve) - ■ Saved credentials to Day-01-02-Foundation-and-Accounts|1Password - ■ Updated to latest macOS - ■ Set computer name (M4-HomeLab) - ■ Verified external SSD visible - ■ Confirmed network connected - ■ macOS desktop ready

Next: Guide 03 - Network Configuration (Static IP + 3 adapters)

Guide 03: Network Configuration

■ Time: 1 hour ■ Coffee: 2 cups ■ Difficulty: Intermediate

Network Architecture Overview

Your M4 will have 3 network adapters:

```

##### M4 Mac Mini (192.168.50.10) #####
##### en0: 10GbE Built-in #####
(Primary) ##### IP: 192.168.50.10/24 ##### Gateway: 192.168.50.1 ##### Purpose:
Main network ##### en1: 5GB USB-C (Backup) ##### IP: 192.168.50.11/24 #####
Gateway: 192.168.50.1 ##### Purpose: Failover/backup ##### en2: 2.5GB USB (VM
Network) ##### IP: 192.168.51.1/24 ##### No gateway (isolated) ##### Purpose: VM
management #####

```

■ Prerequisites Checklist

Before starting: - [] All 3 Ethernet adapters physically connected - [] Cables connected to switch/router - [] Admin password ready (from Day-01-02-Foundation-and-Accounts|1Password) - [] Router admin access available - [] Terminal app ready to use

■ Configure Primary Network (M4 Mac Mini - Complete System Overview|10GbE)

Step 1: Identify Network Adapters

1. Open Terminal (Applications → Utilities → Terminal)
2. List network interfaces:

```
ifconfig | grep -E "^en[0-9]|inet "
```

You'll see output like:

```
en0: flags=8863 mtu 1500 inet 192.168.50.xxx netmask 0xffffffff broadcast 192.168.50.255 en1: flags=8863 mtu 1500 inet 169.254.x.x netmask 0xffff0000 broadcast 169.254.255.255 en2: flags=8863 mtu 1500
```

3. Identify which is which:

```
networksetup -listallhardwareports
```

Output shows:

```
Hardware Port: Ethernet Device: en0 Ethernet Address: xx:xx:xx:xx:xx:xx Hardware Port: USB 10Gb Ethernet Adapter Device: en1 Ethernet Address: xx:xx:xx:xx:xx:xx Hardware Port: USB 10Gb Ethernet Adapter 2 Device: en2 Ethernet Address: xx:xx:xx:xx:xx:xx
```

Note: Names may vary slightly based on your adapters

■ CHECKPOINT: Network adapters identified

Step 2: Configure Static IP on Primary (en0)

Using GUI:

1. System Settings → Network
2. Select: Ethernet (en0) - the built-in M4 Mac Mini - Complete System Overview|10GbE
3. Click: Details
4. TCP/IP tab:

```
Configure IPv4: Using DHCP with manual address IPv4 Address: 192.168.50.10 Subnet Mask: 255.255.255.0 Router: 192.168.50.1
```

5. DNS tab:

```
DNS Servers: 192.168.50.1 (router) 1.1.1.1 (Cloudflare primary) 1.0.0.1 (Cloudflare secondary)
```

6. Search Domains:

```
local homelab.local
```

7. Click: OK

8. Click: Apply

OR Using Terminal:

```
# Set static IP sudo networksetup -setmanual "Ethernet" 192.168.50.10 255.255.255.0 192.168.50.1 # Set DNS sudo networksetup -setdnsservers "Ethernet" 192.168.50.1 1.1.1.1 1.0.0.1 # Set search domains sudo networksetup -setsearchdomains "Ethernet" local homelab.local
```

■ CHECKPOINT: Primary network configured with static IP

Step 3: Reserve IP on Router

CRITICAL: Reserve IP on router so DHCP doesn't assign it to other devices

1. Open browser 2. Go to: <http://192.168.50.1> 3. Login with router admin credentials 4. Find: DHCP Settings or LAN Settings 5. Look for: DHCP Reservations or Static DHCP 6. Add reservation:

```
Device Name: M4-HomeLab MAC Address: [en0 MAC from earlier] IP Address: 192.168.50.10
```

7. Save/Apply

■ CHECKPOINT: IP reserved on router

Step 4: Test Primary Network

```
# Test connectivity ping -c 4 192.168.50.1 # Should get replies # Test DNS ping -c 4 google.com # Should resolve and reply # Test internet speed curl -s https://raw.githubusercontent.com/sivel/speedtest-cli/master/speedtest.py | python3 - # Should show your 5Gbps CityFibre connection! # Check current IP ifconfig en0 | grep "inet " # Should show: inet 192.168.50.10
```

Expected results:

```
■ Router pingable ■ DNS working ■ Internet working ■ Speed: 4-5 Gbps down/up ■ IP: 192.168.50.10
```

■ CHECKPOINT: Primary network working perfectly

■ Configure Secondary Network (5GB USB-C)

Purpose: Backup/failover connection

Step 5: Configure Backup Network

1. System Settings → Network
2. Select: USB 10Gb Ethernet Adapter (this is your 5GB adapter)
3. Click: Details
4. TCP/IP tab:

```
Configure IPv4: Using DHCP with manual address IPv4 Address: 192.168.50.11 Subnet Mask: 255.255.255.0 Router: 192.168.50.1
```

5. DNS tab: Same as primary

```
DNS Servers: 192.168.50.1 1.1.1.1 1.0.0.1
```

6. Click: OK 7. Click: Apply

OR Using Terminal:

```
# Identify the exact service name networksetup -listallnetworkservices # Configure (adjust service name as needed) sudo networksetup -setmanual "USB 10Gb Ethernet Adapter" 192.168.50.11 255.255.255.0 192.168.50.1 sudo networksetup -setdnsservers "USB 10Gb Ethernet Adapter" 192.168.50.1 1.1.1.1 1.0.0.1
```

Step 6: Set Service Order (Priority)

Ensure primary (en0) is preferred:

1. System Settings → Network 2. Click:  (gear icon) → Set Service Order 3. Drag to order:

```
1. Ethernet (en0 - Primary 10GbE) 2. USB 10Gb Ethernet Adapter (en1 - 5GB backup) 3. USB 10Gb Ethernet Adapter 2 (en2 - 2.5GB VM) 4. Wi-Fi (disabled)
```

4. Click: OK 5. Click: Apply

■ CHECKPOINT: Backup network configured

■ Configure VM Network (2.5GB)

Purpose: Isolated network for Day-06-08-Proxmox-Hypervisor|Proxmox VMs

Step 7: Configure VM Network

This network will be ISOLATED (no internet access by default):

1. System Settings → Network
2. Select: USB 10Gb Ethernet Adapter 2 (your 2.5GB adapter)
3. Click: Details

4. TCP/IP tab:

```
Configure IPv4: Manually IPv4 Address: 192.168.51.1 Subnet Mask: 255.255.255.0 Router: (leave blank - no gateway!)
```

5. DNS tab:

```
DNS Servers: (leave blank for now)
```

6. Click: OK 7. Click: Apply

OR Using Terminal:

```
# Configure VM network (no gateway) sudo networksetup -setmanual "USB 10Gb Ethernet Adapter 2" 192.168.51.1 255.255.255.0
```

This creates isolated network:

```
192.168.51.0/24 ■■ M4 Mac Mini: 192.168.51.1 ■■ Proxmox VMs: 192.168.51.10-192.168.51.100 ■■ No internet access (isolated)
```

■ CHECKPOINT: VM network configured

■ Enable SSH Access

Essential for remote management!

Step 8: Enable SSH

1. System Settings → General → Sharing

2. Toggle ON: Remote Login

3. Allow access for:

```
Select: Only these users Add: steve (your admin account)
```

4. Note the information shown:

```
To log in remotely, type: ssh steve@192.168.50.10 OR ssh steve@m4-homelab.local
```

Test SSH from another device:

```
# From MacBook Air or phone (with Termux/iSH): ssh steve@192.168.50.10 # Should prompt for password # Enter your admin password # You're in!
```

■ CHECKPOINT: SSH enabled and tested

Step 9: Set up SSH Key (Optional but Recommended)

Makes SSH access easier and more secure

On your MacBook Air (client):

■ CHECKPOINT: SSH keys configured

Expected: Isolated network has no internet ■

Test 4: SSH Access

```
# From another device: ssh steve@192.168.50.10 # Should connect # Try hostname: ssh
steve@m4-homelab.local # Should also work
```

Expected: SSH works from both IP and hostname ■

■ Guide 03 Complete!

What you've accomplished: - ■ Identified all 3 network adapters - ■ Configured primary M4 Mac Mini - Complete System Overview|10GbE with static IP (192.168.50.10) - ■ Reserved IP on router - ■ Configured 5GB backup network (192.168.50.11) - ■ Configured isolated VM network (192.168.51.1) - ■ Set service order priority - ■ Enabled SSH access - ■ Tested all networks - ■ Network configuration documented in Day-01-02-Foundation-and-Accounts|1Password

Your M4 now has enterprise-grade networking! ■

Next: [Guide 04 - Storage Configuration \(4TB SSD\)](#)

Guide 04: Storage Configuration

■ Time: 30 minutes ■ Coffee: 1 cup ■ Difficulty: Beginner

Storage Architecture Overview

Your M4 storage setup:

```
██████████████████████████████████████████████████ ■ M4 Mac Mini Storage ■  
██████████████████████████████████████████████████ ■ ■ ■ Internal 512GB SSD  
(Macintosh HD) ■ ■ ■ macOS System: ~40GB ■ ■ ■ Applications: ~20GB ■ ■ ■ Docker  
Configs: ~10GB ■ ■ ■ User Data: ~50GB ■ ■ ■ Free Space: ~392GB ■ ■ ■ External 4TB  
SSD ■ ■ ■ Media: ~2TB (Plex library) ■ ■ ■ Photos: ~200GB (Immich) ■ ■ ■ Frigate:  
~500GB (camera recordings)■ ■ ■ VMs: ~1TB (Proxmox VMs) ■ ■ ■ Backups: ~200GB  
(local cache) ■ ■ ■ Free Space: ~100GB buffer ■ ■ ■  
██████████████████████████████████████████████████
```

■ Prerequisites Checklist

- ☐ External 4TB SSD physically connected - ☐ Drive visible in Finder or Disk Utility - ☐ No important data on drive (will be formatted!) - ☐ Backup of any existing data (if needed)

■ Format External SSD

Step 1: Open Disk Utility

1. Applications → Utilities → Disk Utility 2. View → Show All Devices (important!)

You should see:

```
External 4TB Drive ■■ Drive (physical device) ■■ Volume (current partition)
```

■ CHECKPOINT: Disk Utility open, drive visible

Step 2: Erase and Format

1. Select the DRIVE (not the volume, the parent!) - Look for "Samsung T7" or similar - Should show full capacity (4TB)

2. Click: Erase button (top toolbar)

3. Configure format:

```
Name: External4TB Format: APFS (Apple File System - best for macOS) Scheme: GUID
Partition Map (Required for bootable drives and modern macOS)
```

4. Click: Erase

5. Confirm: Erase

6. Wait for completion (30-60 seconds)

7. Click: Done

■■ WARNING: This will ERASE ALL DATA on the drive!

■ CHECKPOINT: Drive formatted as External4TB (APFS)

Step 3: Verify Format

1. In Disk Utility, select External4TB

2. Click: Info button (■■■)

3. Verify:

```
Name: External4TB Type: APFS Volume Capacity: ~4TB Available: ~4TB (should be nearly
full capacity) File System: APFS
```

4. Close Info

■ CHECKPOINT: Format verified

■ Create Folder Structure

Step 4: Create Directory Structure

Open Terminal and create organized folders:

```
# Navigate to external drive cd /Volumes/External4TB # Create main folders sudo mkdir -p \
Media/{Movies,TV,Music,Photos,Comics} \ Downloads/{Complete,Incomplete} \
Frigate/{Recordings,Clips,Snapshots} \ VMs/{Proxmox,ISOs,Backups} \
Backups/{Docker,Configs,Databases,Daily,Weekly} \ Photos/Immich \ TimeMachine # Set
permissions (steve owns everything) sudo chown -R steve:staff /Volumes/External4TB/* #
Verify structure tree -L 2 /Volumes/External4TB # (if tree not installed: brew install
tree) # OR ls -R /Volumes/External4TB
```

Expected structure:

```
/Volumes/External4TB/ ■■■ Media/ ■ ■■■ Movies/ ■ ■■■ TV/ ■ ■■■ Music/ ■ ■■■ Photos/
■ ■■■ Comics/ ■■■ Downloads/ ■ ■■■ Complete/ ■ ■■■ Incomplete/ ■■■ Frigate/ ■ ■■■
Recordings/ ■ ■■■ Clips/ ■ ■■■ Snapshots/ ■■■ VMs/ ■ ■■■ Proxmox/ ■ ■■■ ISOs/ ■
■■■ Backups/ ■■■ Backups/ ■ ■■■ Docker/ ■ ■■■ Configs/ ■ ■■■ Databases/ ■ ■■■
Daily/ ■ ■■■ Weekly/ ■■■ Photos/ ■ ■■■ Immich/ ■■■ TimeMachine/
```

■ CHECKPOINT: Folder structure created

■ Create Symbolic Links

Make external drive easily accessible from home directory:

Step 5: Create Symlinks

```
# Navigate to home directory cd ~ # Create HomeLab folder mkdir -p ~/HomeLab # Create
symlinks to external drive ln -s /Volumes/External4TB/Media ~/HomeLab/Media ln -s
/Volumes/External4TB/VMs ~/HomeLab/VMs ln -s /Volumes/External4TB/Backups
~/HomeLab/Backups ln -s /Volumes/External4TB/Downloads ~/HomeLab/Downloads # Create
Docker folders on internal drive (configs are small, keep local) mkdir -p
~/HomeLab/Docker/{Compose,Configs,Data} # Create Scripts folder mkdir -p
~/HomeLab/Scripts/{Monitoring,Maintenance,Backup,Automation} # Create Documentation
folder mkdir -p ~/HomeLab/Documentation # Verify ls -la ~/HomeLab/
```

Expected output:

```
HomeLab/ ■■■ Docker/ ■ ■■■ Compose/ (local - internal SSD) ■ ■■■ Configs/ (local -
internal SSD) ■ ■■■ Data/ (local - internal SSD) ■■■ Scripts/ (local - internal SSD)
■■■ Documentation/ (local - internal SSD) ■■■ Media -> /Volumes/External4TB/Media ■■■
VMs -> /Volumes/External4TB/VMs ■■■ Backups -> /Volumes/External4TB/Backups ■■■
Downloads -> /Volumes/External4TB/Downloads
```

Why this structure? - ■ Configs on internal SSD (fast access) - ■ Large data on external SSD (more space) - ■
Easy to access everything from ~/HomeLab/ - ■ Logical organization

■ CHECKPOINT: Symlinks created

■ Auto-Mount External Drive

Ensure drive mounts automatically on boot:

Step 6: Configure Auto-Mount

The drive should auto-mount by default, but let's verify:

1. System Settings → General → Login Items
2. Verify "External4TB" is NOT listed as "hidden"
3. Or use Terminal to check:

```
# Check if drive is in fstab cat /etc/fstab # If empty (normal for macOS), create entry
sudo vifs # Add line (press 'i' for insert mode in vi): UUID=YOUR-UUID-HERE none apfs
rw,auto # Get UUID: diskutil info /Volumes/External4TB | grep "Volume UUID"
```

Actually, modern macOS auto-mounts external drives by default!

Better approach - Test reboot:

```
# Reboot and verify drive mounts sudo reboot # After reboot, check: ls /Volumes/ #
Should see: External4TB ls ~/HomeLab/Media # Should work (symlink to external drive)
```

■ CHECKPOINT: External drive auto-mounts

■ Storage Verification

Step 7: Check Storage Status

```
# Check drive info diskutil info /Volumes/External4TB # Check available space df -h
/Volumes/External4TB # Check internal drive df -h / # Check folder sizes du -sh
~/HomeLab/* du -sh /Volumes/External4TB/*
```

Expected output:

```
/Volumes/External4TB: ~4TB total, ~4TB available /: ~512GB total, ~390GB available
```

■ CHECKPOINT: Storage verified

■ Save Storage Configuration

Document in Day-01-02-Foundation-and-Accounts|1Password:

```
1Password → HomeLab Vault → Edit M4 Mac Mini entry Add Custom Section: Storage
Configuration Internal SSD: - Capacity: 512GB - Available: ~390GB after macOS - Mount: /
- Purpose: System, apps, Docker configs External SSD: - Model: Samsung T7 (or your
model) - Capacity: 4TB - Mount: /Volumes/External4TB - Purpose: Media, VMs, photos,
recordings, backups - Auto-mount: Yes Folder Structure: - HomeLab configs: ~/HomeLab/
(internal) - Media: ~/HomeLab/Media → External4TB - VMs: ~/HomeLab/VMs → External4TB -
```

Backups: ~/HomeLab/Backups → External4TB Save

■ CHECKPOINT: Storage documented

■ Guide 04 Complete!

What you've accomplished: - ■ Formatted 4TB external SSD (APFS) - ■ Created organized folder structure - ■ Set up ~/HomeLab/ directory - ■ Created symlinks for easy access - ■ Verified auto-mount on boot - ■ Documented storage configuration - ■ 4.5TB total storage ready!

Next: Guide 05 - macOS Optimization for 24/7

[Continuing with remaining guides...]

#homelab #hardware #m4 #network #setup

↑ Back to Top

■ Volume 03: Docker Infrastructure (Days 3-4)

Source: Day-03-05-Docker-Infrastructure Tags: #homelab #docker #containers #infrastructure

Volume 03: Day-03-05-Docker-Infrastructure|Docker Infrastructure (Days 3-4)

Building Your Container Empire

This volume covers complete Day-03-05-Docker-Infrastructure|Docker setup with all 60+ services.

What You'll Complete

- Day-03-05-Docker-Infrastructure|Docker & Day-03-05-Docker-Infrastructure|OrbStack installation - Day-01-02-Foundation-and-Accounts|Tailscale VPN for remote access - Day-01-02-Foundation-and-Accounts|DuckDNS for dynamic DNS - Nginx Proxy Manager with SSL - Portainer for management - All 60+ services deployed

Prerequisites

- Volume 01 complete (accounts, downloads) - Volume 02 complete (M4 setup, network, storage) - Admin access to M4 - Day-01-02-Foundation-and-Accounts|1Password ready

Guide 07: Day-03-05-Docker-Infrastructure|Docker & Day-03-05-Docker-Infrastructure|OrbStack

Installation

1. Install Day-03-05-Docker-Infrastructure|OrbStack:

```
brew install orbstack
```

2. Launch and configure resources: - Memory: 16GB - CPU: 6 cores - Storage: 100GB

3. Verify installation:

```
docker --version docker ps docker run hello-world
```

Day-03-05-Docker-Infrastructure|Docker Basics

Essential commands:

```
# Containers docker ps # List running docker ps -a # List all docker start # Start container docker stop # Stop container docker logs -f # Follow logs # Images docker images # List images docker pull # Download image docker rmi # Remove image # System docker system df # Disk usage docker system prune -a # Clean everything
```

Docker Compose

Create folder structure:

```
mkdir -p ~/HomeLab/Docker/{Compose,Configs,Data}
```

Test compose file:

```
cat > ~/HomeLab/Docker/Compose/test.yml << 'EOF' version: "3.8" services: whoami: image: traefik/whoami ports: - "8080:80" EOF docker compose -f test.yml up -d curl http://localhost:8080 docker compose -f test.yml down
```

Guide 08: Day-01-02-Foundation-and-Accounts|Tailscale VPN

Setup

1. Install Day-01-02-Foundation-and-Accounts|Tailscale:

```
brew install tailscale
```

2. Connect:

```
tailscale up # Opens browser - sign in with GitHub
```

3. Get your Day-01-02-Foundation-and-Accounts|Tailscale IP:

```
tailscale ip -4 # Note this IP - save to 1Password
```

4. Test from another device:

```
ssh steve@
```

Enable Exit Node (Optional)

```
sudo tailscale up --advertise-exit-node
```

Then approve in Day-01-02-Foundation-and-Accounts|Tailscale admin console.

Guide 09: Day-01-02-Foundation-and-Accounts|DuckDNS

Create Update Script

```
mkdir -p ~/HomeLab/Scripts/Maintenance cat >
~/HomeLab/Scripts/Maintenance/duckdns-update.sh << 'EOF' #!/bin/bash
SUBDOMAIN="stevehomelab" TOKEN="YOUR-TOKEN-FROM-1PASSWORD" CURRENT_IP=$(curl -s
https://ipv4.icanhazip.com) RESPONSE=$(curl -s "https://www.duckdns.org/update?domains=
${SUBDOMAIN}&token=${TOKEN}&ip=${CURRENT_IP}") echo "$(date): IP=$CURRENT_IP,
Response=$RESPONSE" >> ~/HomeLab/Documentation/duckdns.log EOF chmod +x
~/HomeLab/Scripts/Maintenance/duckdns-update.sh
```

Schedule Updates

```
cat > ~/Library/LaunchAgents/com.homelab.duckdns.plist << 'EOF' Label
com.homelab.duckdns ProgramArguments /bin/bash
/Users/steve/HomeLab/Scripts/Maintenance/duckdns-update.sh StartInterval 300 RunAtLoad
EOF launchctl load ~/Library/LaunchAgents/com.homelab.duckdns.plist
```

Test: nslookup stevehomelab.duckdns.org

Guide 10: Nginx Proxy Manager

Deploy

```
cat > ~/HomeLab/Docker/Compose/nginx-proxy-manager.yml << 'EOF' version: "3.8"
services: npm: image: jc21/nginx-proxy-manager:latest container_name:
nginx-proxy-manager restart: unless-stopped ports: - "80:80" - "443:443" - "81:81"
environment: DB_SQLITE_FILE: "/data/database.sqlite" volumes: -
~/HomeLab/Docker/Data/nginx-proxy-manager/data:/data -
~/HomeLab/Docker/Data/nginx-proxy-manager/letsencrypt:/etc/letsencrypt EOF mkdir -p
~/HomeLab/Docker/Data/nginx-proxy-manager/{data,letsencrypt} docker compose -f
nginx-proxy-manager.yml up -d
```

Initial Setup

1. Access: <http://192.168.50.10:81> 2. Login: - Email: admin@example.com - Password: changeme 3. Change immediately! 4. Save new credentials to Day-01-02-Foundation-and-Accounts|1Password

SSL Certificate

1. SSL Certificates → Add 2. Domain: stevehomelab.duckdns.org 3. Email: steve-smithit@outlook.com 4. Agree to Let's Encrypt TOS 5. Save

Guide 11: Portainer

Deploy

```
cat > ~/HomeLab/Docker/Compose/portainer.yml << 'EOF' version: "3.8" services:
portainer: image: portainer/portainer-ce:latest container_name: portainer restart:
unless-stopped ports: - "9443:9443" volumes: -
/var/run/docker.sock:/var/run/docker.sock - ~/HomeLab/Docker/Data/portainer:/data EOF
mkdir -p ~/HomeLab/Docker/Data/portainer docker compose -f portainer.yml up -d
```

Setup

1. Access: <https://192.168.50.10:9443> 2. Create admin account 3. Save to Day-01-02-Foundation-and-Accounts|1Password 4. Select Day-03-05-Docker-Infrastructure|Docker environment 5. Connect

Guide 12: Master Docker Compose

Deploy All Services

Use the docker-compose-master.yml file:

```
cd ~/HomeLab/Docker/Compose docker compose -f docker-compose-master.yml up -d
```

This deploys all 60+ services: - Infrastructure (NPM, Portainer, Watchtower) - Media (Day-13-14-Media-Stack-Plex|Plex, Day-13-14-Media-Stack-Plex|Sonarr, Day-13-14-Media-Stack-Plex|Radarr, Prowlarr, etc.) - Smart Home (Day-15-16-Smart-Home-Integration|Home Assistant, Day-15-16-Smart-Home-Integration|Frigate, Scripted) - AI (Day-17-18-AI-Services-LLMs|Ollama, Open WebUI, Paperless, Immich) - Monitoring (Day-20-Monitoring-Setup|Grafana, Day-20-Monitoring-Setup|Prometheus, Day-20-Monitoring-Setup|Loki, Uptime Kuma) - Security (Pi-hole, Authelia) - Backups (Day-21-Backup-Strategy|Kopia)

Verification

```
# Check all containers running docker ps # Check specific service docker logs plex #
Access services open http://192.168.50.10:32400 # Plex open http://192.168.50.10:8123 #
Home Assistant
```

Volume 03 Complete!

You now have: - ■ Day-03-05-Docker-Infrastructure|Docker/Day-03-05-Docker-Infrastructure|OrbStack running - ■ Day-01-02-Foundation-and-Accounts|Tailscale VPN (access from anywhere) - ■ Day-01-02-Foundation-and-Accounts|DuckDNS auto-updating - ■ Nginx Proxy Manager with SSL - ■ Portainer for management - ■ All 60+ services deployed

Next: Volume 04 - Day-06-08-Proxmox-Hypervisor|Proxmox Setup

↑ Back to Top

=====

■ Guide 08: TP-Link Deco XE75 Mesh Network Setup

Source: Day-03-Deco-Mesh-WiFi Tags: #homelab #network #wifi #mesh #deco

Guide 08: TP-Link Day-03-Deco-Mesh-WiFi|Deco XE75 Mesh Network Setup

■ Time: 45-60 minutes ■ Coffee: 2 cups ■ Difficulty: Intermediate

What You're Building

TP-Link Day-03-Deco-Mesh-WiFi|Deco XE75 Mesh (3 units) with wired Ethernet backhaul

Step 1: Disable Day-03-Deco-Mesh-WiFi|Sky Router WiFi

1. Login: <http://192.168.50.1> 2. Navigate to WiFi settings 3. Disable 2.4GHz WiFi: OFF 4. Disable 5GHz WiFi: OFF 5. Save changes

Step 2: Position Deco Units

Deco #1 (Main): Living room → Switch Port 2 Deco #2 (Office): Home office → Switch Port 3 Deco #3 (Bedroom): Upstairs → Switch Port 4

All connected via Ethernet to Day-02-M4-Mac-Mini-Setup|TP-Link SX1008 switch.

Step 3: Setup via Deco App

1. Install "TP-Link Deco" app on phone 2. Create TP-Link account (save to Day-01-02-Foundation-and-Accounts|1Password) 3. Add Deco #1 (scan QR code) 4. Network name: SteveHomeNet 5. Set password (save to Day-01-02-Foundation-and-Accounts|1Password) 6. Mode: Access Point (NOT Router) 7. Wait for setup (3-5 min) 8. Add Deco #2 and #3 (repeat process)

Step 4: Configure Settings

Address Reservation: - M4 Mac Mini - Complete System Overview|M4 Mac Mini: 192.168.50.10

DHCP Range: - Start: 192.168.50.100 - End: 192.168.50.199

Enable: - Fast Roaming: ON - Beamforming: ON - Network Optimization: ON

IoT Network (optional): - SSID: SteveHomeNet-IoT - 2.4GHz only for smart home devices

Step 5: Connect Devices

Reconnect all 43+ WiFi devices to SteveHomeNet: - Smart home → SteveHomeNet-IoT - Computers/phones → SteveHomeNet - Verify M4 still on wired M4 Mac Mini - Complete System Overview|10GbE (NOT WiFi)

Verification

- [] All 3 Decos solid white LED - [] All showing "Wired" in app - [] Full WiFi coverage - [] M4 on M4 Mac Mini - Complete System Overview|10GbE ethernet - [] Settings saved to Day-01-02-Foundation-and-Accounts|1Password

■ Guide 08 Complete!

#homelab #network #mesh #wifi #deco

↑ Back to Top

■ Day 05 BONUS: Remote Access - RustDesk + Guacamole

Source: Day-05-BONUS-Remote-Access Tags: #homelab #remote-access #rustdesk #guacamole

Day 05 BONUS: Remote Access - Day-05-Remote-Access|RustDesk + Day-05-Remote-Access|Guacamole

■■ Time: 60-90 minutes ■ Coffee: 3 cups ■ Difficulty: Intermediate ■ When: After Day 05 (Day-03-05-Docker-Infrastructure|Docker infrastructure ready)

> "I picked the wrong week to quit remote desktop!" ■

■ What You're Building

Problem: Microsoft killed Remote Desktop on Android Solution: Self-hosted alternatives (better than MS ever was!)

You'll deploy: - Day-05-Remote-Access|RustDesk server (your own relay, works on Android!) - Apache Day-05-Remote-Access|Guacamole (web-based gateway, any browser) - Complete privacy (no cloud dependencies) - Access M4 + VMs from anywhere

■ Prerequisites

Before Starting: - [] Day-03-05-Docker-Infrastructure|Docker infrastructure deployed (Day 03-05) - [] docker-compose-master.yml exists - [] Docker Compose working - [] Sky router admin access (port forwarding)

Part 1: Deploy Day-05-Remote-Access|RustDesk Server

Add to docker-compose-master.yml:

Add these services to your existing file:

```
# RustDesk ID Server rustdesk-hbbs: image: rustdesk/rustdesk-server:latest
container_name: rustdesk-hbbs command: hbbs -r rustdesk.stevhomelab.duckdns.org:21117
ports: - "21115:21115" - "21116:21116" - "21116:21116/udp" - "21118:21118" volumes: -
~/HomeLab/Docker/Data/rustdesk:/root networks: - homelab restart: unless-stopped
environment: - TZ=Europe/London # RustDesk Relay Server rustdesk-hbbr: image:
rustdesk/rustdesk-server:latest container_name: rustdesk-hbbr command: hbbr ports: -
"21117:21117" - "21119:21119" volumes: - ~/HomeLab/Docker/Data/rustdesk:/root networks:
- homelab restart: unless-stopped environment: - TZ=Europe/London depends_on: -
rustdesk-hbbs
```

Deploy:

```
# Create data directory mkdir -p ~/HomeLab/Docker/Data/rustdesk # Deploy containers cd
~/HomeLab/Docker/Compose docker compose -f docker-compose-master.yml up -d
rustdesk-hbbs rustdesk-hbbr # Verify running docker ps | grep rustdesk # Should see both
containers running
```

Get Public Key (IMPORTANT):

```
# Extract the public key docker exec rustdesk-hbbs cat /root/id_ed25519.pub # Copy this
key - you'll need it for every client! # Example output:
xK8VRU0oP/xxxxxxxxxxxxxxxxxxxxxxxxxx=
```

Save to Day-01-02-Foundation-and-Accounts|1Password immediately:

```
Title: RustDesk Server Type: Server URL: rustdesk.stevhomelab.duckdns.org Username:
N/A Password: N/A Public Key: [paste the key above] Ports: 21115-21119 Notes:
Self-hosted RustDesk relay server All clients need this public key
```

■ CHECKPOINT: Day-05-Remote-Access|RustDesk server running, public key saved

Part 2: Deploy Apache Day-05-Remote-Access|Guacamole

Add to docker-compose-master.yml:

```
# Guacamole Database guacamole-db: image: postgres:15-alpine container_name:
guacamole-db restart: unless-stopped environment: - POSTGRES_DB=guacamole_db -
POSTGRES_USER=guacamole_user - POSTGRES_PASSWORD=${GUACAMOLE_DB_PASSWORD} -
PGDATA=/var/lib/postgresql/data/guacamole volumes: -
~/HomeLab/Docker/Data/guacamole/postgresql:/var/lib/postgresql/data networks: - homelab
# Guacamole Daemon guacd: image: guacamole/guacd:latest container_name: guacd restart:
unless-stopped networks: - homelab volumes: -
~/HomeLab/Docker/Data/guacamole/drive:/drive -
~/HomeLab/Docker/Data/guacamole/record:/record # Guacamole Web Interface guacamole:
image: guacamole/guacamole:latest container_name: guacamole restart: unless-stopped
ports: - "8090:8080" environment: - GUACD_HOSTNAME=guacd -
POSTGRES_HOSTNAME=guacamole-db - POSTGRES_DATABASE=guacamole_db -
POSTGRES_USER=guacamole_user - POSTGRES_PASSWORD=${GUACAMOLE_DB_PASSWORD} -
GUACAMOLE_HOME=/etc/guacamole volumes: -
~/HomeLab/Docker/Data/guacamole/config:/etc/guacamole networks: - homelab depends_on: -
guacd - guacamole-db
```

Setup:

```
# Create directories mkdir -p
~/HomeLab/Docker/Data/guacamole/{postgresql,config,drive,record} # Generate secure
database password echo "GUACAMOLE_DB_PASSWORD=$(openssl rand -base64 32)" >> ~/.env #
Source the new variable source ~/.env # Initialize database schema docker run --rm
guacamole/guacamole:latest /opt/guacamole/bin/initdb.sh --postgres > ~/initdb.sql #
Deploy stack cd ~/HomeLab/Docker/Compose docker compose -f docker-compose-master.yml up
-d guacamole-db guacd guacamole # Wait 30 seconds for containers to start sleep 30 #
Initialize database docker exec -i guacamole-db psql -U guacamole_user -d guacamole_db
< ~/initdb.sql # Verify all running docker ps | grep guacamole
```

Access Day-05-Remote-Access|Guacamole:

```
URL: http://192.168.50.10:8090/guacamole Default Credentials: Username: guacadmin
Password: guacadmin ■■ CHANGE IMMEDIATELY AFTER FIRST LOGIN!
```

Change Default Password:

1. Login with guacadmin/guacadmin 2. Click: guacadmin (top right) → Settings 3. Click: Users → guacadmin 4. Change password (16+ characters) 5. Save

Save new credentials to Day-01-02-Foundation-and-Accounts|1Password:

```
Title: Guacamole Web Gateway URL: http://192.168.50.10:8090/guacamole Username:
guacadmin Password: [your new password] Database Password: [from ~/.env
GUACAMOLE_DB_PASSWORD] Notes: Web-based remote desktop gateway Access from any browser
```

■ CHECKPOINT: Day-05-Remote-Access|Guacamole running, password changed, saved to Day-01-02-Foundation-and-Accounts|1Password

Part 3: Configure Day-03-Deco-Mesh-WiFi|Sky Router Port Forwarding

Login to Router:

```
URL: http://192.168.50.1 Username: admin Password: [from router or Sky account]
```

Add Port Forwarding Rules:

Navigate to: Port Forwarding / Virtual Servers / NAT

Add these 6 rules:

```
Rule 1: RustDesk ID Server External Port: 21115 Internal Port: 21115 Protocol: TCP
Internal IP: 192.168.50.10 Enable: YES Rule 2: RustDesk NAT Type External Port: 21116
Internal Port: 21116 Protocol: TCP Internal IP: 192.168.50.10 Enable: YES Rule 3:
RustDesk NAT (UDP) External Port: 21116 Internal Port: 21116 Protocol: UDP Internal IP:
192.168.50.10 Enable: YES Rule 4: RustDesk Relay External Port: 21117 Internal Port:
21117 Protocol: TCP Internal IP: 192.168.50.10 Enable: YES Rule 5: RustDesk Web Client
External Port: 21118 Internal Port: 21118 Protocol: TCP Internal IP: 192.168.50.10
Enable: YES Rule 6: RustDesk WebSocket External Port: 21119 Internal Port: 21119
Protocol: TCP Internal IP: 192.168.50.10 Enable: YES
```

Save/Apply changes

Test Port Forwarding:

```
# From external network (use phone data, not home WiFi) # Or use online port checker:
portchecker.co # Test if port 21115 is open # Should show: OPEN or ACCESSIBLE
```

■ CHECKPOINT: All 6 ports forwarded

Part 4: Setup Day-05-Remote-Access|RustDesk Android Client

Download & Install:

1. On Samsung S25 Ultra: - Open browser - Go to: <https://rustdesk.com/> - Download: Android APK - Or: F-Droid store → Search "Day-05-Remote-Access|RustDesk"

2. Install:

```
Settings → Security → Unknown Sources → Enable Downloads → RustDesk.apk → Install
```

Configure Client:

1. Open Day-05-Remote-Access|RustDesk app

2. Tap menu (■) → Settings → Network

3. Configure servers:

```
ID Server: rustdesk.stevhomelab.duckdns.org Relay Server:
rustdesk.stevhomelab.duckdns.org API Server: (leave empty) Key: [paste your public key
from Part 1]
```

4. Tap OK

5. Restart app (completely close and reopen)

Setup M4 for Remote Access:

```
# On M4 Mac Mini # Install RustDesk brew install rustdesk # Launch RustDesk open -a
RustDesk # Configure same servers: # Settings → Network → Same settings as Android #
ID Server: rustdesk.stevhomelab.duckdns.org # Relay Server:
rustdesk.stevhomelab.duckdns.org # Key: [your public key] # Note the ID shown (e.g.,
123456789) # Save to 1Password!
```

Test Connection:

1. On Android: - Enter M4's Day-05-Remote-Access|RustDesk ID - Tap Connect - Enter password (if prompted)
- Should see M4 desktop!

■ CHECKPOINT: Android app works, can connect to M4

Part 5: Configure Day-05-Remote-Access|Guacamole Connections

Add M4 Mac (VNC):

1. Login to Day-05-Remote-Access|Guacamole: <http://192.168.50.10:8090/guacamole>

2. Enable Screen Sharing on M4:

```
# On M4 System Settings → Sharing Toggle ON: Screen Sharing Set VNC password: [16+ characters] Save password to 1Password!
```

3. In Day-05-Remote-Access|Guacamole: - Settings → Connections → New Connection

```
Name: M4 Mac Mini (VNC) Protocol: VNC Network: Hostname: localhost Port: 5900
Authentication: Password: [VNC password] Display: Color depth: True color (32-bit) Save
```

4. Test: Click connection, should see M4 desktop

Add Windows 11 VM (RDP):

After Day 09 when Windows VM deployed:

```
Settings → Connections → New Connection Name: Windows 11 VM Protocol: RDP Network:
Hostname: 192.168.50.52 Port: 3389 Authentication: Username: Steve Password: [Windows password] Save
```

Add Ubuntu VM (SSH):

After Day 11 when Ubuntu deployed:

```
Settings → Connections → New Connection Name: Ubuntu Server Protocol: SSH Network:
Hostname: 192.168.50.51 Port: 22 Authentication: Username: steve Password: [or use SSH key] Save
```

■ CHECKPOINT: Day-05-Remote-Access|Guacamole connections configured

Part 6: Test Remote Access

From Home WiFi:

Day-05-Remote-Access|RustDesk: 1. Open app on phone 2. Connect to M4 using ID 3. Should work instantly

Day-05-Remote-Access|Guacamole: 1. Browser: <http://192.168.50.10:8090/guacamole> 2. Login 3. Click M4 connection 4. Should see desktop

From External Network:

Day-05-Remote-Access|RustDesk: 1. Disable WiFi (use mobile data) 2. Open
Day-05-Remote-Access|RustDesk 3. Connect to M4 using ID 4. Should work (via your relay server!)

Day-05-Remote-Access|Guacamole: - Need to setup Day-01-02-Foundation-and-Accounts|DuckDNS first (Day 06) - Then: <https://guacamole.stevhomelab.duckdns.org>

Troubleshooting

Day-05-Remote-Access|RustDesk "Connection Failed":

```
# Check containers running docker ps | grep rustdesk # Check logs docker logs  
rustdesk-hbbs docker logs rustdesk-hbbr # Verify public key matches docker exec  
rustdesk-hbbs cat /root/id_ed25519.pub # Check ports forwarded # Use portchecker.co to  
test ports 21115-21119 # Restart containers docker restart rustdesk-hbbs rustdesk-hbbr
```

Day-05-Remote-Access|Guacamole Won't Load:

```
# Check all containers docker ps | grep guacamole # Check database docker logs  
guacamole-db # Check daemon docker logs guacd # Check web docker logs guacamole #  
Restart stack docker restart guacamole-db guacd guacamole # Re-initialize database if  
needed docker exec -i guacamole-db psql -U guacamole_user -d guacamole_db <  
~/initdb.sql
```

Can't Connect to M4 VNC:

```
# On M4, check Screen Sharing enabled System Settings → Sharing → Screen Sharing: ON #  
Test VNC locally # From M4 terminal: nc -zv localhost 5900 # Should show: succeeded #  
Check firewall # System Settings → Network → Firewall # If enabled, allow Screen  
Sharing
```

Verification Checklist

Day-05-Remote-Access|RustDesk Server: - [] Both containers running - [] Public key saved to
Day-01-02-Foundation-and-Accounts|1Password - [] Ports 21115-21119 forwarded - [] Android app configured
- [] Can connect to M4

Day-05-Remote-Access|Guacamole: - [] All 3 containers running - [] Web interface accessible - [] Default
password changed - [] M4 VNC connection working - [] Credentials in
Day-01-02-Foundation-and-Accounts|1Password

Remote Access: - [] Works from home WiFi - [] Day-05-Remote-Access|RustDesk works on mobile data - []
All credentials backed up

What You've Achieved

Remote Access Solutions:

Day-05-Remote-Access|RustDesk (Native Apps): - ■ Android compatible (unlike MS!) - ■ iOS compatible - ■ Windows/Mac/Linux apps - ■ Self-hosted (complete privacy) - ■ Works on mobile data

Day-05-Remote-Access|Guacamole (Browser): - ■ Access from any browser - ■ No app installation needed - ■ Supports RDP, VNC, SSH - ■ Multi-protocol gateway - ■ Session recording capable

Benefits: - Complete privacy (self-hosted) - Works on Android - Access M4 + all VMs - Better than MS Remote Desktop - Free forever!

■ BONUS GUIDE COMPLETE!

You can now: - Remote into M4 from anywhere - Use Android app on Samsung S25 - Access via any web browser - Connect to VMs (once deployed) - Maintain complete privacy!

Next Steps: - Continue with Day 06 (Day-06-08-Proxmox-Hypervisor|Proxmox) - Add more Day-05-Remote-Access|Guacamole connections as you deploy VMs - Access your homelab from anywhere! ■

"I am serious about remote access. And don't call me Shirley." ■■

↑ Back to Top

■ Guide 12: Remote Access - RustDesk + Guacamole

Source: Day-05-Remote-Access Tags: #homelab #remote-access #rustdesk #guacamole

Guide 12: Remote Access - Day-05-Remote-Access|RustDesk + Day-05-Remote-Access|Guacamole

■■ Time: 60-90 minutes ■ Coffee: 3 cups ■ Difficulty: Intermediate

What You're Building

Self-hosted remote access (works on Android!): - Day-05-Remote-Access|RustDesk server (own relay) - Apache Day-05-Remote-Access|Guacamole (web-based)

Part 1: Deploy Day-05-Remote-Access|RustDesk Server

Add to docker-compose-master.yml:

```
rustdesk-hbbs: image: rustdesk/rustdesk-server:latest container_name: rustdesk-hbbs
command: hbbs -r rustdesk.stevhomelab.duckdns.org:21117 ports: - "21115:21115" -
"21116:21116" - "21116:21116/udp" - "21118:21118" volumes: -
~/HomeLab/Docker/Data/rustdesk:/root networks: - homelab restart: unless-stopped
rustdesk-hbbr: image: rustdesk/rustdesk-server:latest container_name: rustdesk-hbbr
command: hbbr ports: - "21117:21117" - "21119:21119" volumes: -
~/HomeLab/Docker/Data/rustdesk:/root networks: - homelab restart: unless-stopped
```

Deploy:

```
docker compose -f docker-compose-master.yml up -d rustdesk-hbbs rustdesk-hbbr
```

Get public key:

```
docker exec rustdesk-hbbs cat /root/id_ed25519.pub # Save to 1Password!
```

Part 2: Deploy Day-05-Remote-Access|Guacamole

Add to docker-compose-master.yml:

```
guacamole-db: image: postgres:15-alpine container_name: guacamole-db environment: -
POSTGRES_DB=guacamole_db - POSTGRES_USER=guacamole_user -
POSTGRES_PASSWORD=${GUACAMOLE_DB_PASSWORD} volumes: -
~/HomeLab/Docker/Data/guacamole/postgresql:/var/lib/postgresql/data networks: - homelab
restart: unless-stopped guacd: image: guacamole/guacd:latest container_name: guacd
networks: - homelab restart: unless-stopped guacamole: image:
guacamole/guacamole:latest container_name: guacamole ports: - "8090:8080" environment:
- GUACD_HOSTNAME=guacd - POSTGRES_HOSTNAME=guacamole-db -
POSTGRES_DATABASE=guacamole_db - POSTGRES_USER=guacamole_user -
POSTGRES_PASSWORD=${GUACAMOLE_DB_PASSWORD} networks: - homelab restart: unless-stopped
depends_on: - guacd - guacamole-db
```

Setup:

```
# Generate password echo "GUACAMOLE_DB_PASSWORD=$(openssl rand -base64 32)" >> ~/.env #
Initialize database docker run --rm guacamole/guacamole:latest
/opt/guacamole/bin/initdb.sh --postgres > ~/initdb.sql # Deploy docker compose -f
docker-compose-master.yml up -d guacamole-db guacd guacamole
```

Access: <http://192.168.50.10:8090/guacamole> Default: guacadmin / guacadmin (change immediately!)

Part 3: Port Forwarding

Login to Day-03-Deco-Mesh-WiFi|Sky Router (192.168.50.1):

Add port forwarding rules: - 21115 TCP → 192.168.50.10 - 21116 TCP → 192.168.50.10 - 21116 UDP → 192.168.50.10 - 21117 TCP → 192.168.50.10 - 21118 TCP → 192.168.50.10 - 21119 TCP → 192.168.50.10

Part 4: Day-05-Remote-Access|RustDesk Android Setup

1. Download Day-05-Remote-Access|RustDesk APK from <https://rustdesk.com/> 2. Install on Samsung S25 Ultra 3. Configure: - ID Server: rustdesk.stevhomelab.duckdns.org - Relay Server: rustdesk.stevhomelab.duckdns.org - Key: [paste your public key] 4. Connect to M4 using its ID

Part 5: Day-05-Remote-Access|Guacamole Configuration

1. Login to Day-05-Remote-Access|Guacamole 2. Settings → Connections → New Connection 3. Add M4 Mac (VNC): - Name: M4 Mac Mini - Complete System Overview|M4 Mac Mini - Protocol: VNC - Hostname: localhost - Port: 5900 - Password: [VNC password]

Save to Day-01-02-Foundation-and-Accounts|1Password

Day-05-Remote-Access|RustDesk Server: - URL: rustdesk.stevhomelab.duckdns.org - Public Key: [your key] - Ports: 21115-21119

Day-05-Remote-Access|Guacamole: - URL: <http://192.168.50.10:8090/guacamole> - Username: [your admin username] - Password: [your password] - DB Password: [from .env]

Verification

- [] Day-05-Remote-Access|RustDesk containers running - [] Day-05-Remote-Access|Guacamole accessible - [] Port forwarding configured - [] Android app connects - [] Can remote into M4 - [] Credentials in Day-01-02-Foundation-and-Accounts|1Password

■ Guide 12 Complete!

#homelab #remote-access #rustdesk #guacamole #android

↑ Back to Top

=====

■ Volume 04: Proxmox Virtualization Platform (Days 5-6)

Source: Day-06-08-Proxmox-Hypervisor Tags: #homelab #proxmox #virtualization #hypervisor

Volume 04: Day-06-08-Proxmox-Hypervisor|Proxmox Virtualization Platform (Days 5-6)

Enterprise Hypervisor on Your M4

This volume covers complete Day-06-08-Proxmox-Hypervisor|Proxmox setup for running VMs.

What You'll Complete

- UTM installation for running Day-06-08-Proxmox-Hypervisor|Proxmox - Day-06-08-Proxmox-Hypervisor|Proxmox VE installation - Web GUI access - Network configuration (bridged to all 3 adapters) - Storage setup - Ready to create VMs

Prerequisites

- Volumes 01-03 complete - Day-06-08-Proxmox-Hypervisor|Proxmox ISO downloaded (from Volume 01) - 16GB RAM allocated for VMs - 6 CPU cores available

Guide 13: UTM Installation

Why UTM First?

Day-06-08-Proxmox-Hypervisor|Proxmox runs as a VM on your M4:

M4 Mac Mini (macOS) ■■ Docker (60+ containers) ■■ UTM ■■ Proxmox VM ■■ Windows 11 ■■
Ubuntu Server ■■ Kali Linux

Install UTM

Via Homebrew `brew install utm` # Or download from: <https://mac.getutm.app/>

Launch UTM from Applications.

Guide 14: Day-06-08-Proxmox-Hypervisor|Proxmox Installation

Download Day-06-08-Proxmox-Hypervisor|Proxmox ISO

If not already downloaded:

```
cd ~/Desktop/HomeLab-Project/ISOs # Download from: https://www.proxmox.com/en/downloads
# Get: proxmox-ve-8.x-arm64.iso (if available) # Or use x86_64 version with emulation
```

Note: Day-06-08-Proxmox-Hypervisor|Proxmox ARM support is experimental. For M4, we'll use standard x86_64 with emulation.

Create Day-06-08-Proxmox-Hypervisor|Proxmox VM in UTM

1. Open UTM 2. Click: + Create New Virtual Machine 3. Select: Virtualize (not Emulate) 4. Operating System: Linux

Configuration:

```
Name: Proxmox-VE Architecture: ARM64 (or x86_64 if ARM not available) Memory: 16384 MB
(16GB) CPU Cores: 6 Storage: 200 GB (on external SSD) Boot ISO: proxmox-ve-*.iso
Network: Bridged (en0 - 10GbE)
```

5. Click: Save

Install Day-06-08-Proxmox-Hypervisor|Proxmox

1. Start VM 2. Select: Install Day-06-08-Proxmox-Hypervisor|Proxmox VE (Graphical) 3. Accept EULA 4. Target Disk: /dev/sda (200GB) 5. Filesystem: ext4

Location:

```
Country: United Kingdom Time Zone: Europe/London Keyboard: UK
```

Password:

```
Password: [strong password] Email: steve-smithit@outlook.com
```

Save to Day-01-02-Foundation-and-Accounts|1Password:

```
Title: Proxmox VE - root Username: root Password: [your password] URL:
https://192.168.50.50:8006 Notes: Proxmox hypervisor admin
```

Network:

```
Management Interface: ens160 Hostname: proxmox.homelab.local IP Address: 192.168.50.50/24 Gateway: 192.168.50.1 DNS: 192.168.50.1
```

6. Click: Install 7. Wait 10-15 minutes 8. Reboot when prompted

Guide 15: Day-06-08-Proxmox-Hypervisor|Proxmox Configuration

Access Web Interface

1. Open browser: <https://192.168.50.50:8006> 2. Accept self-signed certificate warning 3. Login: - User: root - Password: [from Day-01-02-Foundation-and-Accounts|1Password] - Realm: Linux PAM

Disable Enterprise Repo

```
# SSH to Proxmox ssh root@192.168.50.50 # Edit sources nano /etc/apt/sources.list # Add: deb http://download.proxmox.com/debian/pve bookworm pve-no-subscription # Save and exit apt update apt full-upgrade -y
```

Remove Subscription Nag

```
sed -i,backup "s/data.status.tolower() != 'active'/false/g" \ /usr/share/javascript/proxmox-widget-toolkit/proxmoxlib.js systemctl restart pveproxy
```

Refresh browser - nag gone!

Update System

```
apt update apt full-upgrade -y apt install -y htop vim curl wget git
```

Guide 16: Network Configuration

Verify Network Bridge

```
# Check bridges ip addr show vmbr0 # Should show: # vmbr0: ... inet 192.168.50.50/24
```

Day-06-08-Proxmox-Hypervisor|Proxmox automatically creates vmbr0 as bridge.

Configure Secondary Networks

For your 3 ethernet adapters setup:

Primary (M4 Mac Mini - Complete System Overview|10GbE): Already configured as vmbr0

Backup (5GB): Configure as vmbr1

```
# Edit network config nano /etc/network/interfaces # Add: auto vmbr1 iface vmbr1 inet
static address 192.168.50.51/24 bridge-ports ens161 bridge-stp off bridge-fd 0
```

VM Network (2.5GB): Configure as vmbr2

```
# Add to same file: auto vmbr2 iface vmbr2 inet static address 192.168.51.1/24
bridge-ports ens162 bridge-stp off bridge-fd 0
```

Restart networking:

```
systemctl restart networking
```

Guide 17: Storage Configuration

Add External Storage

Mount folder from macOS to Day-06-08-Proxmox-Hypervisor|Proxmox:

1. In UTM VM settings: - Add Shared Directory - Path: /Volumes/External4TB/VMs - Mount automatically
2. In Day-06-08-Proxmox-Hypervisor|Proxmox:

```
# Create mount point mkdir -p /mnt/external-vms # Mount (adjust device name) mount -t
virtiofs external-vms /mnt/external-vms # Make permanent echo "external-vms
/mnt/external-vms virtiofs defaults 0 0" >> /etc/fstab
```

3. Add to Day-06-08-Proxmox-Hypervisor|Proxmox storage: - Datacenter → Storage → Add → Directory - ID: external-vms - Directory: /mnt/external-vms - Content: Disk image, ISO image, Container

Upload ISOs

Copy ISOs to Day-06-08-Proxmox-Hypervisor|Proxmox:

```
# From M4 Mac: scp ~/Desktop/HomeLab-Project/ISOs/ubuntu-*.iso
root@192.168.50.50:/var/lib/vz/template/iso/ scp
~/Desktop/HomeLab-Project/ISOs/kali-*.iso root@192.168.50.50:/var/lib/vz/template/iso/
scp ~/Desktop/HomeLab-Project/ISOs/Windows11*.iso
root@192.168.50.50:/var/lib/vz/template/iso/
```

Guide 18: Backup Configuration

Configure Backup Storage

1. Datacenter → Storage → Add → Directory

```
ID: backups Directory: /mnt/external-vms/Backups Content: VZDump backup file
```

2. Datacenter → Backup → Add

```
Schedule: Daily at 02:00 Storage: backups Mode: Snapshot Compression: ZSTD
```

Guide 19: Quick Commands

Day-06-08-Proxmox-Hypervisor|Proxmox CLI

```
# List VMs qm list # Start VM qm start 100 # Stop VM qm stop 100 # VM status qm status 100 # Create snapshot qm snapshot 100 pre-update # Restore snapshot qm rollback 100 pre-update # List snapshots qm listsnapshot 100
```

System Maintenance

```
# Update Proxmox apt update && apt full-upgrade -y # Clean up apt autoremove -y apt autoclean # Check disk usage df -h # Check VM disk usage pvesm status
```

Guide 20: Access Summary

Day-06-08-Proxmox-Hypervisor|Proxmox Access

Local: - Web: <https://192.168.50.50:8006> - SSH: `ssh root@192.168.50.50`

Via Day-01-02-Foundation-and-Accounts|Tailscale: - Web: <https://100.x.x.x:8006> (M4's Day-01-02-Foundation-and-Accounts|Tailscale IP) - SSH: `ssh root@100.x.x.x`

Via Day-01-02-Foundation-and-Accounts|DuckDNS: - Can proxy via Nginx Proxy Manager - Add: proxmox.stevhomelab.duckdns.org

Default Credentials: - Username: root - Password: [in Day-01-02-Foundation-and-Accounts|1Password] - Realm: Linux PAM

Volume 04 Complete!

You now have: - ■ UTM installed on M4 - ■ Day-06-08-Proxmox-Hypervisor|Proxmox VE running in UTM - ■ Web GUI accessible - ■ Network bridges configured (3 adapters) - ■ Storage configured (external SSD) - ■ ISOs uploaded - ■ Backup system ready - ■ Ready to create VMs!

Network Setup:

```
192.168.50.50 (vmbr0) - Primary network 192.168.50.51 (vmbr1) - Backup network
192.168.51.1 (vmbr2) - Isolated VM network
```

Next: Volume 05 - Windows 11 VM

↑ Back to Top

■ Volume 05: Windows 11 Pro VM (Day 6)

Source: Day-09-10-Windows-11-VM Tags: #homelab #vm #windows #windows11

Volume 05: Windows 11 Pro VM (Day 6)

Running Full Windows in Day-06-08-Proxmox-Hypervisor|Proxmox

This volume covers complete Windows 11 VM setup in Day-06-08-Proxmox-Hypervisor|Proxmox.

What You'll Complete

- Download Windows 11 ARM Insider Preview - Create VM in Day-06-08-Proxmox-Hypervisor|Proxmox with optimal settings - Install VirtIO drivers - Complete Windows installation - Configure networking and RDP - Install essential software - Day-22-Maintenance-and-RMM|Action1 RMM agent

Prerequisites

- Volume 04 complete (Day-06-08-Proxmox-Hypervisor|Proxmox running) - Windows Insider account (from Volume 01) - 8GB RAM available for Windows VM - 500GB disk space on external SSD

Guide 21: Getting Windows 11 ARM

Download from Windows Insider

1. Go to: <https://www.microsoft.com/en-us/software-download/windowsinsiderpreviewARM64>
2. Sign in with Microsoft account
3. Join Windows Insider Program: - Select: Beta Channel (stable) or Dev Channel - Agree to terms
4. Download: - Edition: Windows 11 Pro ARM64 - Language: English (United Kingdom) - Format: VHDX (~10GB)
5. Upload to Day-06-08-Proxmox-Hypervisor|Proxmox:

```
# From M4 Mac: scp ~/Downloads/Windows11_ARM64*.vhdx  
root@192.168.50.50:/var/lib/vz/template/iso/
```

Alternative: CrystalFetch

```
# On Mac: brew install --cask crystalfetch # Launch and download Windows 11 ARM64
```

Guide 22: Create Windows 11 VM

VM Configuration

In Day-06-08-Proxmox-Hypervisor|Proxmox Web GUI:

1. Click: Create VM

2. General:

```
Node: proxmox VM ID: 100 Name: Windows-11-Pro
```

3. OS:

```
Use CD/DVD: No Guest OS: Microsoft Windows Version: 11/2022/2025
```

4. System:

```
Graphic card: Default Machine: q35 BIOS: OVMF (UEFI) Add EFI Disk: Yes Storage:  
local-lvm Format: raw Pre-Enroll keys: No Add TPM: Yes Storage: local-lvm Version: v2.0  
SCSI Controller: VirtIO SCSI single Qemu Agent: Yes (check)
```

5. Disks:

```
Bus/Device: SCSI Storage: local-lvm Disk size: 500 GB Cache: Write back Discard: Yes SSD  
emulation: Yes
```

Import existing VHDX:

```
# SSH to Proxmox ssh root@192.168.50.50 # Convert VHDX to qcow2 qemu-img convert -f vhd\
-O qcow2 \ /var/lib/vz/template/iso/Windows11_ARM64.vhdx \
/var/lib/vz/images/100/vm-100-disk-0.qcow2 # Attach to VM (via GUI: Hardware → Add →
Existing Disk)
```

6. CPU:

```
Sockets: 1 Cores: 3 Type: host
```

7. Memory:

```
Memory: 8192 MB (8GB) Minimum: 4096 MB Ballooning: Yes
```

8. Network:

```
Bridge: vmbr0 Model: VirtIO (paravirtualized) MAC: auto
```

9. Click: Finish

Guide 23: VirtIO Drivers

Download VirtIO ISO

```
# SSH to Proxmox cd /var/lib/vz/template/iso/ # Download VirtIO drivers wget https://fe
dorapeople.org/groups/virt/virtio-win/direct-downloads/stable-virtio/virtio-win.iso
```

Attach to VM

In Day-06-08-Proxmox-Hypervisor|Proxmox GUI: 1. Select Windows VM 2. Hardware → Add → CD/DVD Drive
3. Bus: SATA 4. ISO: virtio-win.iso 5. Add

Guide 24: Windows Installation

Start Installation

1. Start VM (click Start) 2. Open Console (click Console)

If using VHDX import: - Windows boots directly - Skip to first login

If using ISO: - Follow standard Windows installation - At disk selection: Load Driver - Browse to
E:\vioscsi\w11\ARM64\ - Install driver - Disk now appears - Continue installation

First Setup

```
Region: United Kingdom Keyboard: UK Network: Skip for now (configure after) Account:  
Create local account Name: Steve Password: [strong password] Security questions: Answer  
3
```

Save to Day-01-02-Foundation-and-Accounts|1Password:

```
Title: Windows 11 VM - Steve Account Username: Steve Password: [your password] IP:  
192.168.50.52 (will configure) Notes: Windows 11 Pro ARM VM on Proxmox Administrator  
account
```

Install VirtIO Drivers

After first boot:

1. Open Device Manager - Right-click Start → Device Manager
2. Update Unknown Devices: - Right-click each → Update Driver - Browse: E:\ (virtio-win CD) - Install all drivers
3. Install QEMU Guest Agent: - E:\guest-agent\qemu-ga-x86_64.msi - Install - Restart

Guide 25: Network Configuration

Set Static IP

1. Settings → Network & Internet → Ethernet
2. Edit IP settings:

```
IP assignment: Manual IPv4: On IP address: 192.168.50.52 Subnet prefix: 24 Gateway:  
192.168.50.1 Preferred DNS: 192.168.50.1 Alternate DNS: 1.1.1.1
```

3. Save

Set Computer Name

1. Settings → System → About
2. Rename this PC:

```
New name: Windows-HomeLab
```

3. Restart

Test Network

```
# Open PowerShell ping 192.168.50.1 ping google.com
```

Guide 26: Windows Configuration

Windows Updates

1. Settings → Windows Update 2. Check for updates 3. Install all available 4. Restart as needed

Activate Windows

1. Settings → Activation 2. Change product key 3. Enter Windows 11 Pro key 4. Activate

Or run unactivated for homelab (has watermark but fully functional)

Essential Settings

Privacy: - Settings → Privacy & Security - Turn OFF most telemetry - Keep Windows Update ON

Performance: - Settings → System → Power - Power mode: Best Performance

Remote Desktop: - Settings → System → Remote Desktop - Enable Remote Desktop: ON - Network Level Authentication: ON

Test RDP:

```
# From Mac: # Install Microsoft Remote Desktop from App Store # Add PC: 192.168.50.52 #  
Username: Steve # Connect!
```

Guide 27: Software Installation

Windows Terminal

From Microsoft Store: - Search: Windows Terminal - Install

Browsers

Download and install: - Google Chrome or Firefox - Edge already installed

Utilities

Download and install: - 7-Zip (<https://7-zip.org>) - Notepad++ (<https://notepad-plus-plus.org>) - PuTTY (for SSH to other VMs)

Day-22-Maintenance-and-RMM|Action1 RMM Agent

1. Go to: <https://app.action1.com> 2. Login: steve-smithit@outlook.com 3. Download agent 4. Install agent 5. VM appears in Day-22-Maintenance-and-RMM|Action1 dashboard

Save to Day-01-02-Foundation-and-Accounts|1Password:

Update Action1 entry: Add note: Windows-HomeLab VM enrolled

Guide 28: Optimization

Disable Unnecessary Services

Services.msc → Disable: - Xbox services (if not gaming) - Windows Search (if not needed) - Superfetch

Disk Cleanup

- Run: Disk Cleanup - Select: Clean up system files - Delete temporary files

Windows Defender

- Already active - No additional antivirus needed for VM

Guide 29: Snapshots

Create Clean Snapshot

In Day-06-08-Proxmox-Hypervisor|Proxmox: 1. Select Windows VM 2. Snapshots → Take Snapshot

Name: clean-install Include RAM: No Description: Fresh Windows 11 install with updates

3. Take Snapshot

Now you can always restore to this point!

Guide 30: Access Methods

Local Access

Via Day-06-08-Proxmox-Hypervisor|Proxmox Console: - Day-06-08-Proxmox-Hypervisor|Proxmox GUI → VM 100 → Console

Via RDP: - Microsoft Remote Desktop - Server: 192.168.50.52 - User: Steve

Remote Access

Via Day-01-02-Foundation-and-Accounts|Tailscale: - RDP to: 100.x.x.50
(Day-06-08-Proxmox-Hypervisor|Proxmox Day-01-02-Foundation-and-Accounts|Tailscale IP routing)

Via Day-01-02-Foundation-and-Accounts|DuckDNS: - Set up port forward in Nginx Proxy Manager -
rdp.stevhomelab.duckdns.org

Guide 31: Verification

Checklist

- ☐ Windows 11 Pro installed and activated - ☐ Static IP: 192.168.50.52 - ☐ Computer name: Windows-HomeLab - ☐ All Windows updates installed - ☐ VirtIO drivers installed - ☐ QEMU Guest Agent installed - ☐ Remote Desktop working - ☐ Network connectivity confirmed - ☐ Day-22-Maintenance-and-RMM|Action1 agent installed - ☐ Clean snapshot created - ☐ Credentials in Day-01-02-Foundation-and-Accounts|1Password

Performance Test

```
# Check CPU wmic cpu get name # Check RAM wmic memorychip get capacity # Check disk wmic diskdrive get size # Network speed # Install: speedtest-cli speedtest
```

Volume 05 Complete!

You now have: - ■ Windows 11 Pro VM running in Day-06-08-Proxmox-Hypervisor|Proxmox - ■ Fully configured and updated - ■ Remote Desktop enabled - ■ VirtIO drivers for performance - ■ Day-22-Maintenance-and-RMM|Action1 RMM monitoring - ■ Clean snapshot for recovery - ■ Accessible via RDP locally and remotely

Windows 11 VM Specs:

```
VM ID: 100 Name: Windows-HomeLab IP: 192.168.50.52 RAM: 8GB CPU: 3 cores Disk: 500GB OS: Windows 11 Pro ARM64
```

Next: Volume 06 - Ubuntu & Kali Linux VMs

[↑ Back to Top](#)

■ Volume 06: Ubuntu & Kali Linux VMs (Day 6)

Source: Day-11-12-Ubuntu-Kali-VMs Tags: #homelab #vm #linux #ubuntu #kali

Volume 06: Ubuntu & Kali Linux VMs (Day 6)

Linux VMs for Development and Security Testing

This volume covers Ubuntu Server and Kali Linux VM setup in Day-06-08-Proxmox-Hypervisor|Proxmox.

What You'll Complete

- Ubuntu Server 24.04 VM (for general Linux work) - Kali Linux VM (for security testing) - Both VMs networked and accessible - SSH configured with keys - Essential tools installed - Day-03-05-Docker-Infrastructure|Docker on Ubuntu (optional)

Prerequisites

- Volume 05 complete (Day-06-08-Proxmox-Hypervisor|Proxmox running, Windows VM working) - Ubuntu and Kali ISOs uploaded to Day-06-08-Proxmox-Hypervisor|Proxmox - 10GB RAM available for VMs (6GB Ubuntu + 4GB Kali)

Guide 32: Ubuntu Server VM

Create VM

In Day-06-08-Proxmox-Hypervisor|Proxmox Web GUI:

1. Click: Create VM

2. General:

```
VM ID: 101 Name: Ubuntu-Server-HomeLab
```

3. OS:

```
ISO image: ubuntu-24.04-live-server-arm64.iso Guest OS: Linux Version: 6.x - 2.6 Kernel
```

4. System:

```
Graphic card: Default Machine: Default (i440fx) SCSI Controller: VirtIO SCSI single  
Qemu Agent: Yes
```

5. Disks:

```
Bus: VirtIO Block Storage: local-lvm Size: 200 GB Cache: Default Discard: Yes SSD  
emulation: Yes
```

6. CPU:

```
Sockets: 1 Cores: 2 Type: host
```

7. Memory:

```
Memory: 6144 MB (6GB) Minimum: 2048 MB
```

8. Network:

```
Bridge: vbr0 Model: VirtIO
```

9. Click: Finish & Start

Install Ubuntu Server

1. Open Console

2. Select: Install Ubuntu Server
3. Language: English
4. Keyboard: English (UK)
5. Network: - Use DHCP initially - Note IP assigned
6. Proxy: (leave blank)
7. Mirror: Default
8. Storage: - Use entire disk - Set up LVM: Yes - Confirm
9. Profile:

```
Name: Steve Server name: ubuntu-homelab Username: steve Password: [strong password]
```

Save to Day-01-02-Foundation-and-Accounts|1Password:

```
Title: Ubuntu Server VM - steve Username: steve Password: [your password] IP: 192.168.50.51 (will configure) SSH: ssh steve@192.168.50.51
```

10. SSH: Install OpenSSH server: Yes
11. Snaps: Skip all
12. Installation: Proceeds (10-15 min)
13. Reboot: When prompted

Post-Install Configuration

Set Static IP

```
# Login via console # Edit netplan sudo nano /etc/netplan/50-cloud-init.yaml
```

Configuration:

```
network: version: 2 ethernet: ens18: dhcp4: no addresses: - 192.168.50.51/24 routes: - to: default via: 192.168.50.1 nameservers: addresses: - 192.168.50.1 - 1.1.1.1
```

Apply:

```
sudo netplan apply ip addr show
```

Install QEMU Guest Agent

```
sudo apt update sudo apt install -y qemu-guest-agent sudo systemctl enable qemu-guest-agent sudo systemctl start qemu-guest-agent
```

Day-06-08-Proxmox-Hypervisor|Proxmox now shows VM IP and can shutdown properly!

System Updates

```
sudo apt update sudo apt upgrade -y sudo apt install -y curl wget git htop btop vim net-tools
```

Install Day-03-05-Docker-Infrastructure|Docker (Optional)

```
# Install Docker curl -fsSL https://get.docker.com | sh # Add user to docker group sudo usermod -aG docker steve # Enable Docker sudo systemctl enable docker sudo systemctl start docker # Test docker run hello-world
```

Guide 33: Kali Linux VM

Create VM

1. Click: Create VM

2. General:

```
VM ID: 102 Name: Kali-Linux-HomeLab
```

3. OS:

```
ISO image: kali-linux-2024.x-installer-arm64.iso Type: Linux
```

4. System: Default settings

5. Disks:

```
Size: 150 GB Bus: VirtIO Block
```

6. CPU:

```
Cores: 2 Type: host
```

7. Memory:

```
Memory: 4096 MB (4GB)
```

8. Network:

```
Bridge: vmbr0 Model: VirtIO
```

9. Finish & Start

Install Kali Linux

1. Select: Graphical install

2. Language: English

3. Location: United Kingdom
4. Keyboard: British English
5. Network: DHCP (configure static later)
6. Hostname: kali-homelab
7. Domain: (leave blank)
8. Root password: [strong password]
9. User:

```
Full name: Steve Username: steve Password: [strong password]
```

Save to Day-01-02-Foundation-and-Accounts|1Password:

```
Title: Kali Linux VM - steve Username: steve Root Password: [root password] User  
Password: [steve password] IP: 192.168.50.53 (will configure) SSH: ssh  
steve@192.168.50.53
```

10. Partitioning: Guided - use entire disk
11. Desktop: Xfce (lightweight)
12. Install GRUB: Yes, /dev/sda
13. Installation: ~20 minutes
14. Reboot

Post-Install Configuration

Set Static IP

```
sudo nano /etc/network/interfaces
```

Add:

```
auto ens18 iface ens18 inet static address 192.168.50.53 netmask 255.255.255.0 gateway  
192.168.50.1 dns-nameservers 192.168.50.1 1.1.1.1
```

Apply:

```
sudo systemctl restart networking
```

Update System

```
sudo apt update sudo apt full-upgrade -y sudo apt install -y kali-linux-default  
qemu-guest-agent sudo systemctl enable qemu-guest-agent
```

Enable SSH

```
sudo systemctl enable ssh sudo systemctl start ssh
```

Guide 34: SSH Key Setup

Generate SSH Key on M4

```
# On M4 Mac: ssh-keygen -t ed25519 -C "steve-smithit@outlook.com" # Save to:  
~/.ssh/id_ed25519 # Passphrase: (optional but recommended)
```

Copy to VMs

```
# Copy to Ubuntu ssh-copy-id steve@192.168.50.51 # Copy to Kali ssh-copy-id  
steve@192.168.50.53 # Copy to Windows (optional, if OpenSSH installed)
```

Test Key-Based Login

```
# Should login without password: ssh steve@192.168.50.51 ssh steve@192.168.50.53
```

Guide 35: VM Verification

Ubuntu Server Checklist

- [] VM ID: 101 - [] IP: 192.168.50.51 (static) - [] Hostname: ubuntu-homelab - [] SSH working with keys - []
QEMU agent installed - [] Day-03-05-Docker-Infrastructure|Docker installed (optional) - [] System updated - []
Credentials in Day-01-02-Foundation-and-Accounts|1Password

Test:

```
ssh steve@192.168.50.51 docker --version # If installed
```

Kali Linux Checklist

- [] VM ID: 102 - [] IP: 192.168.50.53 (static) - [] Hostname: kali-homelab - [] SSH working with keys - []
QEMU agent installed - [] Xfce desktop working - [] System updated - [] Credentials in
Day-01-02-Foundation-and-Accounts|1Password

Test:

```
ssh steve@192.168.50.53 sudo nmap --version
```

Guide 36: Network Test

Test Connectivity

```
# From M4, test all VMs: ping -c 4 192.168.50.50 # Proxmox ping -c 4 192.168.50.51 #  
Ubuntu ping -c 4 192.168.50.52 # Windows ping -c 4 192.168.50.53 # Kali # SSH to each  
Linux VM: ssh steve@192.168.50.51 # Ubuntu ssh steve@192.168.50.53 # Kali # RDP to  
Windows: # Use Microsoft Remote Desktop: 192.168.50.52
```

Speed Test Between VMs

```
# On Ubuntu: sudo apt install -y iperf3 iperf3 -s # On Kali: sudo apt install -y iperf3  
iperf3 -c 192.168.50.51 # Should see good speeds (Gbps range)
```

Guide 37: Snapshots

Create Clean Snapshots

Ubuntu:

```
Proxmox → VM 101 → Snapshots → Take Snapshot Name: clean-install Description: Fresh  
Ubuntu with updates and Docker
```

Kali:

```
Proxmox → VM 102 → Snapshots → Take Snapshot Name: clean-install Description: Fresh  
Kali with updates and tools
```

Guide 38: Common Tasks

Start/Stop VMs

Via Day-06-08-Proxmox-Hypervisor|Proxmox GUI: - Select VM → Start/Stop/Restart

Via CLI:

```
# SSH to Proxmox ssh root@192.168.50.50 # Start VM qm start 101 # Ubuntu qm start 102 #  
Kali # Stop VM qm stop 101 qm stop 102 # Status qm status 101 qm list
```

Access VMs

SSH:

```
ssh steve@192.168.50.51 # Ubuntu ssh steve@192.168.50.53 # Kali
```

Via Day-06-08-Proxmox-Hypervisor|Proxmox Console: - Day-06-08-Proxmox-Hypervisor|Proxmox GUI → VM → Console

Kali Desktop (VNC): - Day-06-08-Proxmox-Hypervisor|Proxmox GUI → VM 102 → Console - Or via VNC viewer

Guide 39: Usage Examples

Ubuntu Server Uses

- Day-03-05-Docker-Infrastructure|Docker host for testing - Development environment - Web server testing - Database server - CI/CD runner - General Linux tasks

Kali Linux Uses

■■ LEGAL USE ONLY: - Network scanning (your own network) - Vulnerability testing (authorized only) - Security learning - Penetration testing practice - Capture The Flag (CTF) challenges

Recommended Learning: - TryHackMe: <https://tryhackme.com> - HackTheBox: <https://hackthebox.com> - OverTheWire: <https://overthewire.org>

Volume 06 Complete!

You now have: - ■ Ubuntu Server VM (192.168.50.51) - 6GB RAM, 2 cores, 200GB disk - Day-03-05-Docker-Infrastructure|Docker installed - SSH with keys

- ■ Kali Linux VM (192.168.50.53) - 4GB RAM, 2 cores, 150GB disk - Security tools installed - SSH with keys

- ■ All VMs networked together - ■ SSH access configured - ■ Clean snapshots created - ■ Ready for use!

VM Summary:

```
Proxmox Host: 192.168.50.50 ■■ VM 100: Windows-HomeLab (192.168.50.52) - 8GB/3 cores
■■ VM 101: Ubuntu-Server (192.168.50.51) - 6GB/2 cores ■■ VM 102: Kali-Linux
(192.168.50.53) - 4GB/2 cores Total VM Resources: 18GB RAM, 7 cores
```

*Next: Volume 07 - Media Stack (Day-13-14-Media-Stack-Plex|Plex + arr services)**

↑ Back to Top

■ Volume 07: Media Stack (Days 7-8)

Source: Day-13-14-Media-Stack-Plex Tags: #homelab #media #plex #automation #arr

Volume 07: Media Stack (Days 7-8)

Complete Media Automation with Day-13-14-Media-Stack-Plex|Plex + arr Services*

This volume covers your complete media automation setup.

What You'll Complete

- Day-13-14-Media-Stack-Plex|Plex Media Server (streaming) - Day-13-14-Media-Stack-Plex|Sonarr (TV show automation) - Day-13-14-Media-Stack-Plex|Radarr (Movie automation) - Prowlarr (Indexer management) - qBittorrent (Download client with ExpressVPN) - Bazarr (Subtitle automation) - Overseerr (Request management) - Tdarr (Transcoding automation) - All services linked and working together

Prerequisites

- Volumes 01-06 complete - Day-03-05-Docker-Infrastructure|Docker running (from Volume 03) - External 4TB storage mounted - Day-13-14-Media-Stack-Plex|Plex account (from Volume 01) - ExpressVPN account

Guide 40: Day-13-14-Media-Stack-Plex|Plex Media Server

Deploy Day-13-14-Media-Stack-Plex|Plex

Already configured in docker-compose-master.yml:

```
plex: image: plexinc/pms-docker:latest container_name: plex restart: unless-stopped
network_mode: host environment: - TZ=Europe/London - PLEX_CLAIM=claim-xxxxxxxxxxxxx
volumes: - ~/HomeLab/Docker/Data/plex:/config - /Volumes/External4TB/Media:/media -
~/HomeLab/Docker/Data/plex/transcode:/transcode
```


Get Claim Token

1. Go to: <https://www.plex.tv/claim/> 2. Login with your Day-13-14-Media-Stack-Plex|Plex account 3. Copy claim token (valid 4 minutes) 4. Add to docker-compose-master.yml

Start Day-13-14-Media-Stack-Plex|Plex

```
cd ~/HomeLab/Docker/Compose docker compose -f docker-compose-master.yml up -d plex
```

Setup Day-13-14-Media-Stack-Plex|Plex

1. Access: <http://192.168.50.10:32400/web> 2. Sign in with Day-13-14-Media-Stack-Plex|Plex account 3. Name server: M4-HomeLab 4. Add Library: - Movies: /media/Movies - TV Shows: /media/TV - Music: /media/Music

Configure Remote Access

1. Settings → Remote Access 2. Enable: Manually specify public port 3. Port: 32400 4. Apply

Access anywhere: - Local: <http://192.168.50.10:32400/web> - Remote: <https://app.plex.tv> (auto-discovers) - Apps: Day-13-14-Media-Stack-Plex|Plex app on phone/TV

Guide 41: Prowlarr (Indexers)

Deploy Prowlarr

```
docker compose -f docker-compose-master.yml up -d prowlarr
```

Access: <http://192.168.50.10:9696>

Setup Indexers

1. Add Indexers: - Indexers → Add Indexer - Search for public trackers - Add: 1337x, RARBG, etc. - Configure credentials if needed

2. Connect to Apps: - Settings → Apps → Add Application - Type: Day-13-14-Media-Stack-Plex|Sonarr - Prowlarr Server: <http://prowlarr:9696> - Day-13-14-Media-Stack-Plex|Sonarr Server: <http://sonarr:8989> - API Key: (get from Day-13-14-Media-Stack-Plex|Sonarr)

- Type: Day-13-14-Media-Stack-Plex|Radarr - Similar config

*Prowlarr now manages indexers for all arr apps!**

Guide 42: Day-13-14-Media-Stack-Plex|Sonarr (TV Shows)

Deploy Day-13-14-Media-Stack-Plex|Sonarr

```
docker compose -f docker-compose-master.yml up -d sonarr
```

Access: <http://192.168.50.10:8989>

Initial Setup

1. Media Management: - Settings → Media Management - Root Folder: /tv - Episode Naming: Standard - Permissions: 755
2. Connect Prowlarr: - Settings → Indexers - Already connected via Prowlarr!
3. Connect Download Client: - Settings → Download Clients → Add - Type: qBittorrent - Host: qbittorrent - Port: 8112 - Username: admin - Password: (from qBittorrent)
4. Connect Day-13-14-Media-Stack-Plex|Plex: - Settings → Connect → Add - Type: Day-13-14-Media-Stack-Plex|Plex Media Server - Host: 192.168.50.10 - Port: 32400 - Auth Token: (from Day-13-14-Media-Stack-Plex|Plex Settings → Network)

Add TV Show

1. Series → Add New
2. Search: Your favorite show
3. Root Folder: /tv
4. Monitor: All Episodes
5. Add

Day-13-14-Media-Stack-Plex|Sonarr automatically: - Searches for episodes - Downloads via qBittorrent - Renames and organizes - Updates Day-13-14-Media-Stack-Plex|Plex

Guide 43: Day-13-14-Media-Stack-Plex|Radarr (Movies)

Deploy Day-13-14-Media-Stack-Plex|Radarr

```
docker compose -f docker-compose-master.yml up -d radarr
```

Access: <http://192.168.50.10:7878>

Setup (Similar to Day-13-14-Media-Stack-Plex|Sonarr)

1. Media Management: - Root Folder: /movies - Movie Naming: Standard
2. Indexers: Via Prowlarr
3. Download Client: qBittorrent
4. Connect Day-13-14-Media-Stack-Plex|Plex

Add Movie

1. Movies → Add New
2. Search: Movie name
3. Add to: /movies
4. Monitor: Yes
5. Add

Guide 44: qBittorrent with VPN

Deploy Gluetun + qBittorrent

```
# In docker-compose-master.yml: gluetun: image: qmcgaw/gluetun:latest container_name:
gluetun cap_add: - NET_ADMIN ports: - "8112:8112" # qBittorrent Web UI environment: -
VPN_SERVICE_PROVIDER=expressvpn - OPENVPN_USER=your-username -
OPENVPN_PASSWORD=your-password - SERVER_COUNTRIES=UK qbittorrent: network_mode:
"service:gluetun" # Routes through VPN!
```

Get ExpressVPN credentials: 1. Go to: <https://www.expressvpn.com/setup> 2. Copy username and password 3. Add to docker-compose-master.yml

Start:

```
docker compose -f docker-compose-master.yml up -d gluetun qbittorrent
```

Configure qBittorrent

Access: <http://192.168.50.10:8112>

Default credentials: - Username: admin - Password: adminadmin

Change immediately!

Settings: - Downloads → Save path: /downloads/complete - Connection → Listening port: 6881 - Speed → Global rate limits: (set to your preference)

Test VPN: 1. Check IP: <https://ipleak.net> 2. Should show ExpressVPN IP, NOT your home IP!

Guide 45: Bazarr (Subtitles)

Deploy Bazarr

```
docker compose -f docker-compose-master.yml up -d bazarr
```

Access: <http://192.168.50.10:6767>

Setup

1. Languages: - Settings → Languages - Add: English
2. Connect Day-13-14-Media-Stack-Plex|Sonarr: - Settings → Day-13-14-Media-Stack-Plex|Sonarr - Address: <http://sonarr:8989> - API Key: (from Day-13-14-Media-Stack-Plex|Sonarr)
3. Connect Day-13-14-Media-Stack-Plex|Radarr: - Similar config
4. Subtitle Providers: - Settings → Providers - Add: OpenSubtitles, Subscene, etc.

Bazarr automatically downloads subtitles for all media!

Guide 46: Overseerr (Requests)

Deploy Overseerr

```
docker compose -f docker-compose-master.yml up -d overseerr
```

Access: <http://192.168.50.10:5055>

Setup

1. Day-13-14-Media-Stack-Plex|Plex Connection: - Sign in with Day-13-14-Media-Stack-Plex|Plex - Select server: M4-HomeLab
2. Connect Day-13-14-Media-Stack-Plex|Sonarr: - Settings → Services → Day-13-14-Media-Stack-Plex|Sonarr - Add Server - URL: <http://sonarr:8989> - API Key: (from Day-13-14-Media-Stack-Plex|Sonarr)
3. Connect Day-13-14-Media-Stack-Plex|Radarr: - Similar config

Use Overseerr

Share with family/friends: - Give them: overseerr.stevhomelab.duckdns.org - They can request movies/shows
- You approve - Automatically downloads!

Guide 47: Tdarr (Transcoding)

Deploy Tdarr

```
docker compose -f docker-compose-master.yml up -d tdarr
```

Access: <http://192.168.50.10:8265>

Setup

1. Libraries: - Add Library - Source: /media/Movies - Transcode cache: /temp
2. Transcoding Rules: - Use H.265 (HEVC) for space saving - Target quality: Same as source - Audio: AAC stereo
3. Schedule: - Transcode during off-peak hours - Queue settings: 2 concurrent

Tdarr automatically optimizes your media library!

Guide 48: Workflow Example

How It All Works Together

1. User requests movie via Overseerr - Family member searches "Inception" - Clicks: Request
2. Day-13-14-Media-Stack-Plex|Radarr receives request - Searches indexers via Prowlarr - Finds best quality release - Sends to qBittorrent
3. qBittorrent downloads (via VPN) - Downloads through ExpressVPN - IP hidden - Saves to /downloads/complete
4. Day-13-14-Media-Stack-Plex|Radarr processes - Detects completed download - Renames: "Inception (2010).mkv" - Moves to: /media/Movies/Inception (2010)/ - Notifies Day-13-14-Media-Stack-Plex|Plex

- 5. Bazarr adds subtitles - Detects new movie - Downloads English subtitles - Saves alongside movie
- 6. Tdarr optimizes (optional) - Transcodes to H.265 - Saves disk space - Maintains quality
- 7. Day-13-14-Media-Stack-Plex|Plex updates library - Scans for new media - Adds metadata and posters - Ready to watch!
- 8. User watches - Opens Day-13-14-Media-Stack-Plex|Plex app - "Inception" appears - Streams perfectly!

All automatic! ■

Guide 49: Access URLs

Service Access

```
Plex: http://192.168.50.10:32400/web Sonarr: http://192.168.50.10:8989 Radarr:
http://192.168.50.10:7878 Prowlarr: http://192.168.50.10:9696 qBittorrent:
http://192.168.50.10:8112 Bazarr: http://192.168.50.10:6767 Overseerr:
http://192.168.50.10:5055 Tdarr: http://192.168.50.10:8265
```

Add to Nginx Proxy Manager

For each service: 1. Proxy Hosts → Add 2. Domain: plex.stevhomelab.duckdns.org 3. Forward: 192.168.50.10:32400 4. SSL: Your certificate 5. Save

Now accessible via: - plex.stevhomelab.duckdns.org - sonarr.stevhomelab.duckdns.org - etc.

Guide 50: Maintenance

Regular Tasks

Weekly: - Check download queue - Review Overseerr requests - Clear completed downloads

Monthly: - Update containers:

```
docker compose -f docker-compose-master.yml pull docker compose -f
docker-compose-master.yml up -d
```

- Review disk space - Check Tdarr progress

Troubleshooting

Downloads not starting:

```
# Check VPN docker logs gluetun # Check qBittorrent docker logs qbittorrent # Verify VPN IP
curl --interface eth0 ifconfig.me
```

Day-13-14-Media-Stack-Plex|Plex not updating:

```
# Force library scan docker exec plex /usr/lib/plexmediaserver/Plex\ Media\ Scanner --scan
```

Volume 07 Complete!

You now have: - ■ Day-13-14-Media-Stack-Plex|Plex Media Server streaming - ■ Day-13-14-Media-Stack-Plex|Sonarr managing TV shows - ■ Day-13-14-Media-Stack-Plex|Radarr managing movies - ■ Prowlarr managing indexers - ■ qBittorrent with ExpressVPN (safe downloads) - ■ Bazarr auto-downloading subtitles - ■ Overseerr for family requests - ■ Tdarr optimizing media - ■ Fully automated media pipeline!

Media Stack Summary:

```
Request (Overseerr) ↓ Search (Prowlarr → Sonarr/Radarr) ↓ Download (qBittorrent via VPN) ↓ Process (Sonarr/Radarr) ↓ Subtitles (Bazarr) ↓ Optimize (Tdarr - optional) ↓ Stream (Plex)
```

Next: Volume 08 - Smart Home (Day-15-16-Smart-Home-Integration|Home Assistant, Day-15-16-Smart-Home-Integration|Frigate, Nest cameras, Sonos, Tesla)

↑ Back to Top

■ Volume 08: Smart Home Integration (Day 9)

Source: Day-15-16-Smart-Home-Integration Tags: #homelab #smart-home #iot #home-assistant

Volume 08: Smart Home Integration (Day 9)

Complete Smart Home Control Center

This volume covers integrating your entire smart home ecosystem.

What You'll Complete

- Day-15-16-Smart-Home-Integration|Home Assistant with 43+ devices - Day-15-16-Smart-Home-Integration|Frigate NVR with 5 Nest cameras - Scripted for camera integration - 10 Sonos speakers control - Tesla Model Y integration - Cupra Born EV integration - Easee EV charger control - Complete automation

Prerequisites

- Volumes 01-07 complete - Day-03-05-Docker-Infrastructure|Docker running - All smart home credentials in Day-01-02-Foundation-and-Accounts|1Password - Devices on network

Guide 51: Day-15-16-Smart-Home-Integration|Home Assistant

Deploy Day-15-16-Smart-Home-Integration|Home Assistant

```
docker compose -f docker-compose-master.yml up -d homeassistant
```

Access: <http://192.168.50.10:8123>

Initial Setup

1. Create account: - Name: Steve - Username: steve - Password: [strong password]
2. Save to Day-01-02-Foundation-and-Accounts|1Password:

```
Title: Home Assistant URL: http://192.168.50.10:8123 Username: steve Password: [your password]
```

3. Location: Set to your home address
4. Time zone: Europe/London
5. Analytics: Opt out

Install Integrations

Nest Cameras (5 devices)

1. Settings → Devices & Services → Add Integration
2. Search: Google Nest
3. Sign in with Google account
4. Authorize Day-15-16-Smart-Home-Integration|Home Assistant
5. Select devices: - Front Door Camera - Back Garden Camera - Driveway Camera - Living Room Camera - Office Camera

Sonos (10 speakers)

1. Add Integration → Sonos 2. Auto-discovers all 10 speakers 3. Group by room: - Living Room (2 speakers - stereo pair) - Kitchen (1) - Office (1) - Bedroom (2 - stereo pair) - Garden (2) - Garage (1) - Bathroom (1)

Tesla Integration

1. Add Integration → Tesla 2. Credentials: - Email: [from Day-01-02-Foundation-and-Accounts|1Password] - Password: [from Day-01-02-Foundation-and-Accounts|1Password] 3. Select: Tesla Model Y 4. Exposes: - Location - Battery level - Charging state - Climate control - Lock/unlock - Charge port

Cupra Born Integration

1. Add Integration → Volkswagen We Connect 2. Credentials: [from Day-01-02-Foundation-and-Accounts|1Password] 3. Select: Cupra Born 4. Similar controls as Tesla

Easee Charger

1. Add Integration → Easee 2. Credentials: [from Day-01-02-Foundation-and-Accounts|1Password] 3. Exposes: - Charging status - Power usage - Start/stop charging - Load balancing

Guide 52: Day-15-16-Smart-Home-Integration|Frigate NVR

Deploy Day-15-16-Smart-Home-Integration|Frigate

```
frigate: image: ghcr.io/blakeblackshear/frigate:stable container_name: frigate restart: unless-stopped privileged: true ports: - "5000:5000" - "8554:8554" # RTSP - "8555:8555/tcp" # WebRTC volumes: - ~/HomeLab/Docker/Configs/frigate:/config - /Volumes/External4TB/Frigate:/media/frigate - /dev/shm:/dev/shm environment: - FRIGATE_RTSP_PASSWORD=your-password
```

Configure Day-15-16-Smart-Home-Integration|Frigate

Create: ~/HomeLab/Day-03-05-Docker-Infrastructure|Docker/Configs/frigate/config.yml

```
mqtt: enabled: false cameras: front_door: ffmpeg: inputs: - path: rtsp://nest-camera-1-url roles: - detect - record detect: width: 1920 height: 1080 record: enabled: true retain: days: 30 snapshots: enabled: true retain: default: 7 objects: track: - person - car - dog - cat back_garden: # Similar config for other 4 cameras
```

Access Day-15-16-Smart-Home-Integration|Frigate

- Web UI: <http://192.168.50.10:5000> - View live streams - Review recordings - View detected objects - Manage alerts

Guide 53: Scripted

Deploy Scripted

```
docker compose -f docker-compose-master.yml up -d scripted
```

Access: <http://192.168.50.10:11080>

Setup

1. Create account 2. Add Nest cameras 3. Bridge to HomeKit (optional) 4. Enable RTSP restreaming 5. Connect to Day-15-16-Smart-Home-Integration|Frigate

Benefits: - Better Nest camera integration - HomeKit support - Local streaming - No cloud dependency

Guide 54: Automations

Example Automations

Welcome Home

```
automation: - alias: "Welcome Home" trigger: - platform: state entity_id:
device_tracker.tesla_model_y to: "home" action: - service: light.turn_on target:
entity_id: light.entrance - service: climate.set_temperature data: entity_id:
climate.living_room temperature: 21 - service: media_player.play_media target:
entity_id: media_player.sonos_living_room data: media_content_id:
"spotify:playlist:your-playlist"
```

Charge at Night

```
automation: - alias: "Start EV Charging at Low Rate" trigger: - platform: time at:
"01:00:00" condition: - condition: state entity_id: binary_sensor.tesla_charging state:
"off" - condition: numeric_state entity_id: sensor.tesla_battery below: 80 action: -
service: switch.turn_on target: entity_id: switch.easeee_charger
```

Security Alert

```
automation: - alias: "Person Detected at Night" trigger: - platform: state entity_id:
binary_sensor.front_door_person_detected to: "on" condition: - condition: time after:
"22:00:00" before: "06:00:00" action: - service: notify.mobile_app data: title: "Front
Door Motion" message: "Person detected at front door" - service: light.turn_on target:
entity_id: light.front_porch
```

Guide 55: Dashboards

Create Custom Dashboard

1. Overview → Edit Dashboard 2. Add cards: - Camera feeds (5 Nest cameras) - Sonos controls (10 speakers)
- Tesla status and controls - Cupra status - Easee charger status - Climate controls - Energy monitoring

Mobile Access

1. Install Day-15-16-Smart-Home-Integration|Home Assistant app on Samsung S25 2. Add server:
<http://192.168.50.10:8123> 3. Or via Day-01-02-Foundation-and-Accounts|Tailscale: <http://100.x.x.x:8123> 4.
Enable notifications

Volume 08 Complete!

You now have: - ■ Day-15-16-Smart-Home-Integration|Home Assistant central hub - ■ 5 Nest cameras in
Day-15-16-Smart-Home-Integration|Frigate - ■ 10 Sonos speakers integrated - ■ Tesla Model Y control - ■
Cupra Born control - ■ Easee charger automation - ■ 43+ devices total - ■ Custom automations - ■ Mobile
control

Smart Home Summary:

```
43+ Devices: ■■ 5 Nest Cameras (via Frigate + Scrypted) ■■ 10 Sonos Speakers
(multi-room audio) ■■ Tesla Model Y (location, climate, charging) ■■ Cupra Born
(status, controls) ■■ Easee Charger (smart charging) ■■ + Other smart home devices
```

Next: Volume 09 - AI Services

↑ Back to Top

■ Volume 09: AI Services (Day 10)

Source: Day-17-18-AI-Services-LLMs Tags: #homelab #ai #llm #ollama #machine-learning

Volume 09: AI Services (Day 10)

Local AI with Day-17-18-AI-Services-LLMs|Ollama, Document Management, and Photo Organization

This volume covers AI-powered services running locally.

What You'll Complete

- Day-17-18-AI-Services-LLMs|Ollama (local LLM server) - Open WebUI (ChatGPT-like interface) - Day-17-18-AI-Services-LLMs|LM Studio integration - Paperless-ngx (document management with OCR) - Immich (Google Photos alternative) - All running locally with privacy

Prerequisites

- Volumes 01-08 complete - Day-03-05-Docker-Infrastructure|Docker running - 16GB RAM available - External storage for photos/documents

Guide 56: Day-17-18-AI-Services-LLMs|Ollama

Deploy Day-17-18-AI-Services-LLMs|Ollama

```
docker compose -f docker-compose-master.yml up -d ollama
```

Access: <http://192.168.50.10:11434>

Pull Models

```
# SSH to M4 or use terminal # Pull models docker exec -it ollama ollama pull llama3.2:3b
docker exec -it ollama ollama pull mistral:7b docker exec -it ollama ollama pull
codellama:7b docker exec -it ollama ollama pull phi3:mini # List models docker exec -it
ollama ollama list
```

Test Day-17-18-AI-Services-LLMs|Ollama

```
# Chat with model docker exec -it ollama ollama run llama3.2:3b >>> Hello! How are you?
>>> exit
```

Guide 57: Open WebUI

Deploy Open WebUI

```
docker compose -f docker-compose-master.yml up -d open-webui
```

Access: <http://192.168.50.10:3000>

Setup

1. Create admin account: - Name: Steve - Email: steve-smithit@outlook.com - Password: [strong password]
2. Save to Day-01-02-Foundation-and-Accounts|1Password
3. Connect to Day-17-18-AI-Services-LLMs|Ollama: - Settings → Connections - Day-17-18-AI-Services-LLMs|Ollama URL: <http://ollama:11434> - Test connection

Use Open WebUI

- ChatGPT-like interface - Select model: llama3.2, mistral, etc. - All runs locally on M4! - Private - no data sent to cloud

Access via Day-01-02-Foundation-and-Accounts|Tailscale from phone: - <http://100.x.x.x:3000> (M4's Day-01-02-Foundation-and-Accounts|Tailscale IP) - Your personal ChatGPT on phone!

Guide 58: Day-17-18-AI-Services-LLMs|LM Studio Integration

Install Day-17-18-AI-Services-LLMs|LM Studio on M4

```
# Download from: https://lmstudio.ai # Install to /Applications/
```

Configure Day-17-18-AI-Services-LLMs|LM Studio

1. Download models via GUI
2. Start local server
3. API compatible with OpenAI

Alternative to Open WebUI

Use Day-17-18-AI-Services-LLMs|LM Studio's native interface: - Better for M4 optimization - Apple Silicon optimized - GPU acceleration - Local file access

Guide 59: Paperless-ngx

Deploy Paperless

```
docker compose -f docker-compose-master.yml up -d paperless-ngx
```

Access: <http://192.168.50.10:8010>

Setup

1. Create superuser: - Username: steve - Email: steve-smithit@outlook.com - Password: [strong password]
2. Save to Day-01-02-Foundation-and-Accounts|1Password

Configure OCR

- Settings → OCR - Language: English - Enable: Text recognition - Enable: Barcode detection

Usage

Upload documents: 1. Drag and drop PDFs, images 2. Paperless OCRs automatically 3. Extracts text, dates, correspondents 4. Fully searchable

Organization: - Tags: Bills, Receipts, Contracts, Medical - Correspondents: Companies, people - Document types: Invoice, Letter, etc.

Workflows: - Email documents to Paperless - Mobile app upload - Automatic inbox processing

Benefits: - Searchable archive - OCR all documents - Encrypted storage - Version history - Email integration

Guide 60: Immich

Deploy Immich

```
docker compose -f docker-compose-master.yml up -d immich-server immich-machine-learning
```

Access: <http://192.168.50.10:2283>

Setup

1. Create admin account: - Name: Steve Smith - Email: steve-smithit@outlook.com - Password: [strong password]
2. Save to Day-01-02-Foundation-and-Accounts|1Password

Configure Storage

- External Library: /photos - Points to: /Volumes/External4TB/Photos/Immich

Features

Photo Management: - Upload from phone (iOS/Android app) - Automatic backup - Face recognition (local ML) - Object detection - Smart search: "dog", "beach", "birthday" - Timeline view - Albums and sharing

Machine Learning: - Runs locally (no cloud) - Face detection and clustering - Object/scene recognition - Smart search - Duplicate detection

Mobile App: 1. Install Immich app (iOS/Android) 2. Server URL: <http://192.168.50.10:2283> 3. Or via Day-01-02-Foundation-and-Accounts|Tailscale: <http://100.x.x.x:2283> 4. Enable auto-backup 5. Replaces Google Photos!

Privacy: - All processing local - No cloud upload - Full control of data - Encrypted storage

Guide 61: AI Service Access

URLs

```
Ollama API: http://192.168.50.10:11434 Open WebUI: http://192.168.50.10:3000 Paperless: http://192.168.50.10:8010 Immich: http://192.168.50.10:2283
```

Add to Nginx Proxy Manager

For external access: - ai.stevhomelab.duckdns.org → Open WebUI - docs.stevhomelab.duckdns.org → Paperless - photos.stevhomelab.duckdns.org → Immich

Guide 62: Model Management

Recommended Models

For M4 (32GB RAM):

Small (fast, efficient): - llama3.2:3b (3GB) - General chat - phi3:mini (2GB) - Fast responses - mistral:7b (4GB) - Better quality

Medium (balanced): - llama3.1:8b (8GB) - Good all-rounder - codellama:13b (7GB) - Code generation

Large (best quality, slower): - llama3.1:70b (40GB) - Best quality - mixtral:8x7b (26GB) - Fast large model

Download Models

```
# Via CLI docker exec -it ollama ollama pull llama3.1:8b # Or via Open WebUI: # Settings  
→ Models → Pull a model
```

Volume 09 Complete!

You now have: - ■ Day-17-18-AI-Services-LLMs|Ollama running multiple local LLMs - ■ Open WebUI (ChatGPT alternative) - ■ Day-17-18-AI-Services-LLMs|LM Studio for desktop AI - ■ Paperless-ngx (intelligent document management) - ■ Immich (Google Photos alternative) - ■ All AI processing locally (private!) - ■ Mobile access via apps

AI Stack Summary:

```
Local AI Services: ■■ Ollama (LLM server) ■ ■■ llama3.2:3b ■ ■■ mistral:7b ■ ■■  
codellama:7b ■■ Open WebUI (chat interface) ■■ Paperless-ngx (OCR + document AI) ■■  
Immich (photo AI + face recognition) All running on M4, fully private!
```

Next: Volume 10 - Security Stack

↑ Back to Top

=====

■ Volume 10: Security Stack (Day 11)

Source: Day-19-Security-Hardening Tags: #homelab #security #hardening #wazuh

Volume 10: Security Stack (Day 11)

Enterprise-Grade Security for Your HomeLab

This volume covers complete security hardening and monitoring.

What You'll Complete

- Pi-hole (DNS-level ad blocking) - Authelia (SSO + 2FA) - Day-19-Security-Hardening|CrowdSec (collaborative security) - Fail2ban (intrusion prevention) - Traefik (reverse proxy + security) - Network segmentation - Security monitoring

Prerequisites

- Volumes 01-09 complete - All services running - Network configured

Guide 63: Pi-hole

Deploy Pi-hole

```
docker compose -f docker-compose-master.yml up -d pihole
```

Access: <http://192.168.50.10:8080/admin>

Setup

Password:

```
# Get/set admin password docker exec -it pihole pihole -a -p # Enter new password
```

Save to Day-01-02-Foundation-and-Accounts|1Password.

Configure DNS

On Router: 1. Login to router (192.168.50.1) 2. DHCP settings 3. Primary DNS: 192.168.50.10 (M4/Pi-hole) 4. Secondary DNS: 1.1.1.1 (fallback) 5. Save

Or per-device: - Set DNS to: 192.168.50.10

Add Blocklists

Settings → Adlists → Add:

```
https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts  
https://v.firebog.net/hosts/Admiral.txt https://v.firebog.net/hosts/Easylist.txt
```

Update Gravity: - Tools → Update Gravity

Results

- Blocks ads network-wide - Blocks tracking - Speeds up browsing - Works on all devices - Dashboard shows stats

Guide 64: Authelia

Deploy Authelia

```
docker compose -f docker-compose-master.yml up -d authelia
```

Access: <http://192.168.50.10:9091>

Configuration

Create: ~/HomeLab/Day-03-05-Docker-Infrastructure|Docker/Configs/authelia/configuration.yml

```
server: host: 0.0.0.0 port: 9091 log: level: info authentication_backend: file: path:  
/config/users.yml access_control: default_policy: deny rules: - domain:  
"*.stevehomelab.duckdns.org" policy: two_factor session: domain:  
stevehomelab.duckdns.org expiration: 1h inactivity: 5m storage: local: path:  
/config/db.sqlite3 notifier: smtp: host: smtp.gmail.com port: 587 username:  
steve-smithit@outlook.com password: [app-specific password] sender:  
steve-smithit@outlook.com
```

Create User

~/HomeLab/Day-03-05-Docker-Infrastructure|Docker/Configs/authelia/users.yml:

```
users: steve: displayname: "Steve Smith" password: "$argon2id$v=19$m=65536..." #  
Generate hash email: steve-smithit@outlook.com groups: - admins
```

Generate password hash:

```
docker exec -it authelia authelia crypto hash generate argon2 # Enter password # Copy  
hash to users.yml
```

Enable 2FA

1. Login to Authelia 2. Configure 2FA 3. Scan QR code with authenticator app 4. Save backup codes to Day-01-02-Foundation-and-Accounts|1Password

Integrate with Nginx Proxy Manager

For each protected service: 1. NPM → Edit Proxy Host 2. Custom Locations → Add

```
Location: / Forward Auth URL: http://authelia:9091/api/verify
```

Now requires login + 2FA before accessing!

Guide 65: Day-19-Security-Hardening|CrowdSec

Deploy Day-19-Security-Hardening|CrowdSec

```
docker compose -f docker-compose-master.yml up -d crowdsec
```

Setup

Enroll:

```
# Get enrollment key docker exec crowdsec cscli console enroll [key-from-crowdsec.net]
# Install collections docker exec crowdsec cscli collections install
crowdsecurity/nginx docker exec crowdsec cscli collections install crowdsecurity/linux
docker exec crowdsec cscli collections install crowdsecurity/sshd
```

Configure Bouncer

For Nginx:

```
# Generate bouncer key docker exec crowdsec cscli bouncers add nginx-bouncer # Add to
nginx config
```

What Day-19-Security-Hardening|CrowdSec Does

- Monitors logs for attacks - Shares threat intelligence - Automatically blocks bad IPs - Protects against: - Brute force - Port scanning - DDoS - Known malicious IPs

Dashboard: <https://app.crowdsec.net>

Guide 66: Fail2ban

Deploy Fail2ban

```
docker compose -f docker-compose-master.yml up -d fail2ban
```

Configuration

Create: ~/HomeLab/Day-03-05-Docker-Infrastructure|Docker/Configs/fail2ban/jail.local

```
[DEFAULT] bantime = 3600 findtime = 600 maxretry = 5 destemail =  
steve-smithit@outlook.com action = %(action_mwl)s [sshd] enabled = true port = ssh  
logpath = /var/log/auth.log [nginx-http-auth] enabled = true port = http,https logpath =  
= /var/log/nginx/error.log [nginx-noscript] enabled = true port = http,https logpath =  
/var/log/nginx/access.log [nginx-badbots] enabled = true port = http,https logpath =  
/var/log/nginx/access.log
```

Check Status

```
# List jails docker exec fail2ban fail2ban-client status # Check specific jail docker  
exec fail2ban fail2ban-client status sshd # Unban IP docker exec fail2ban  
fail2ban-client set sshd unbanip 1.2.3.4
```

Guide 67: Network Segmentation

VLANs Setup

Your networks:

```
192.168.50.0/24 - Main network (trusted devices) 192.168.51.0/24 - IoT network (smart  
home devices) 192.168.52.0/24 - Guest network
```

Firewall Rules:

IoT → Main: Deny (except Day-15-16-Smart-Home-Integration|Home Assistant) Guest → Main: Deny Main →
Everything: Allow

Implementation

On TP-Link Switch: 1. Enable VLANs 2. Assign ports: - Port 1-4: VLAN 50 (main) - Port 5-6: VLAN 51 (IoT) -
Port 7-8: VLAN 52 (guest)

On Router: 1. Create virtual interfaces 2. Set firewall rules 3. Separate DHCP pools

Guide 68: Security Monitoring

Day-19-Security-Hardening|Wazuh (Optional)

For enterprise-level SIEM:

```
docker compose -f docker-compose-master.yml up -d wazuh
```

Features: - Security event monitoring - Compliance (PCI-DSS, GDPR) - Vulnerability detection - File integrity monitoring - Log analysis

Uptime Kuma

Already deployed for monitoring:

Access: <http://192.168.50.10:3001>

Add security checks: - SSH login attempts - Failed auth logs - Suspicious traffic - Service availability

Guide 69: Backup Security

Secure Backups

Day-21-Backup-Strategy|Kopia with encryption:

```
# Already configured in Volume 12 # Using AES-256 encryption # Password in 1Password
```

Backup security measures: - Encrypted at rest - Encrypted in transit - Password protected - Offsite (Day-21-Backup-Strategy|pCloud) - Versioned (can restore old versions) - Immutable (can't be deleted by ransomware)

Guide 70: Security Checklist

Daily

- [] Review Pi-hole blocked queries - [] Check Authelia login attempts - [] Review Day-19-Security-Hardening|CrowdSec alerts

Weekly

- [] Review Fail2ban bans - [] Check for failed auth attempts - [] Update Day-03-05-Docker-Infrastructure|Docker containers - [] Review firewall logs

Monthly

- [] Update all services - [] Review access logs - [] Rotate passwords - [] Test backups - [] Security audit

Security Hardening

- [x] All services behind authentication - [x] 2FA enabled (Authelia) - [x] DNS filtering (Pi-hole) - [x] Intrusion prevention (Fail2ban, Day-19-Security-Hardening|CrowdSec) - [x] Network segmentation - [x] Encrypted backups - [x] Regular updates - [x] Monitoring enabled - [x] SSH key-only (no passwords) - [x] Firewall configured

Volume 10 Complete!

You now have: - ■ Pi-hole blocking ads/tracking network-wide - ■ Authelia providing SSO + 2FA - ■ Day-19-Security-Hardening|CrowdSec detecting threats - ■ Fail2ban blocking attackers - ■ Network segmented by trust level - ■ All services secured - ■ Security monitoring active - ■ Enterprise-grade security!

Security Stack:

```
Defense in Depth: ■■ Network Layer ■ ■■ VLANs (segmentation) ■ ■■ Firewall rules ■■
DNS Layer ■ ■■ Pi-hole (ad/tracker blocking) ■■ Application Layer ■ ■■ Authelia (SSO
+ 2FA) ■ ■■ CrowdSec (threat intelligence) ■ ■■ Fail2ban (intrusion prevention) ■■
Monitoring ■■ Logs centralized ■■ Alerts configured ■■ Regular audits
```

Next: Volume 11 - Monitoring Stack

↑ Back to Top

=====

■ Volume 11: Monitoring Stack (Day 12)

Source: Day-20-Monitoring-Setup Tags: #homelab #monitoring #grafana #prometheus #observability

Volume 11: Monitoring Stack (Day 12)

Complete Observability with Day-20-Monitoring-Setup|Grafana + Day-20-Monitoring-Setup|Prometheus

This volume covers comprehensive monitoring and alerting.

What You'll Complete

- Day-20-Monitoring-Setup|Prometheus (metrics collection) - Day-20-Monitoring-Setup|Grafana (visualization) - Day-20-Monitoring-Setup|Loki (log aggregation) - Promtail (log shipping) - Node Exporter (system metrics) - cAdvisor (container metrics) - Uptime Kuma (uptime monitoring) - Alertmanager (notifications)

Prerequisites

- Volumes 01-10 complete - All services running - ntfy.sh account for alerts

Guide 71: Day-20-Monitoring-Setup|Prometheus

Deploy Day-20-Monitoring-Setup|Prometheus

```
docker compose -f docker-compose-master.yml up -d prometheus
```

Access: <http://192.168.50.10:9090>

Configuration

Create: ~/HomeLab/Day-03-05-Docker-Infrastructure|Docker/Configs/prometheus/prometheus.yml

```
global:
  scrape_interval: 15s
  evaluation_interval: 15s
scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']
  - job_name: 'node-exporter'
    static_configs:
      - targets: ['node-exporter:9100']
    labels:
      instance: 'm4-mac-mini'
  - job_name: 'cadvisor'
    static_configs:
      - targets: ['cadvisor:8080']
  - job_name: 'docker'
    static_configs:
      - targets: ['192.168.50.10:9323']
  - job_name: 'proxmox'
    static_configs:
      - targets: ['192.168.50.50:9100']
    labels:
      instance: 'proxmox'
```

Verify Targets

1. Day-20-Monitoring-Setup|Prometheus UI → Status → Targets 2. All should show "UP"

Guide 72: Day-20-Monitoring-Setup|Grafana

Deploy Day-20-Monitoring-Setup|Grafana

```
docker compose -f docker-compose-master.yml up -d grafana
```

Access: <http://192.168.50.10:3010>

Initial Setup

Default login: - Username: admin - Password: admin

Change immediately!

Save to Day-01-02-Foundation-and-Accounts|1Password:

```
Title: Grafana URL: http://192.168.50.10:3010 Username: admin Password: [new password]
```

Add Day-20-Monitoring-Setup|Prometheus Data Source

1. Configuration → Data Sources → Add 2. Type: Day-20-Monitoring-Setup|Prometheus 3. URL: <http://prometheus:9090> 4. Save & Test

Add Day-20-Monitoring-Setup|Loki Data Source

1. Add data source 2. Type: Day-20-Monitoring-Setup|Loki 3. URL: <http://loki:3100> 4. Save & Test

Guide 73: Import Dashboards

System Dashboard

1. Dashboards → Import 2. ID: 1860 (Node Exporter Full) 3. Day-20-Monitoring-Setup|Prometheus data source: Prometheus 4. Import

Shows: - CPU usage - Memory usage - Disk I/O - Network traffic - System load

Day-03-05-Docker-Infrastructure|Docker Dashboard

1. Import dashboard ID: 893 2. Shows all container metrics: - CPU per container - Memory per container - Network per container - Disk per container

Day-06-08-Proxmox-Hypervisor|Proxmox Dashboard

1. Import dashboard ID: 10347 2. Shows VM metrics: - VM CPU/RAM - VM disk usage - VM network - Host resources

Guide 74: Day-20-Monitoring-Setup|Loki + Promtail

Deploy Day-20-Monitoring-Setup|Loki

```
docker compose -f docker-compose-master.yml up -d loki
```

Deploy Promtail

```
docker compose -f docker-compose-master.yml up -d promtail
```

Configuration

Create: ~/HomeLab/Day-03-05-Docker-Infrastructure|Docker/Configs/promtail/config.yml

```
server: http_listen_port: 9080 grpc_listen_port: 0 positions: filename:
/tmp/positions.yaml clients: - url: http://loki:3100/loki/api/v1/push scrape_configs: -
job_name: system static_configs: - targets: - localhost labels: job: varlogs __path__:
/var/log/*log - job_name: docker docker_sd_configs: - host: unix:///var/run/docker.sock
refresh_interval: 5s relabel_configs: - source_labels: ['__meta_docker_container_name']
target_label: 'container'
```

View Logs in Day-20-Monitoring-Setup|Grafana

1. Explore → Select Day-20-Monitoring-Setup|Loki 2. Query: {container="plex"} 3. See live logs!

Guide 75: Uptime Kuma

Deploy Uptime Kuma

```
docker compose -f docker-compose-master.yml up -d uptime-kuma
```

Access: <http://192.168.50.10:3001>

Setup

1. Create admin account 2. Save to Day-01-02-Foundation-and-Accounts|1Password

Add Monitors

HTTP Monitors:

```
Name: Plex Type: HTTP(s) URL: http://192.168.50.10:32400/web Interval: 60s Name: Home Assistant URL: http://192.168.50.10:8123 Interval: 60s [Add all services...]
```

Ping Monitors:

```
Name: Proxmox Type: Ping Hostname: 192.168.50.50 Interval: 60s Name: Router Hostname: 192.168.50.1 Interval: 60s
```

Port Monitors:

```
Name: SSH Type: Port Hostname: 192.168.50.10 Port: 22
```

Guide 76: Alertmanager

Deploy Alertmanager

```
docker compose -f docker-compose-master.yml up -d alertmanager
```

Configuration

Create: ~/HomeLab/Day-03-05-Docker-Infrastructure|Docker/Configs/alertmanager/config.yml

```
global: resolve_timeout: 5m route: receiver: 'ntfy' group_by: ['alertname', 'cluster'] group_wait: 10s group_interval: 10s repeat_interval: 12h receivers: - name: 'ntfy' webhook_configs: - url: 'https://ntfy.sh/stevehomelab-alerts-x7k9m2' send_resolved: true
```

Alert Rules

Add to prometheus.yml:

```
rule_files: - "/etc/prometheus/rules/*.yml"
```

Create: ~/HomeLab/Day-03-05-Docker-Infrastructure|Docker/Configs/prometheus/rules/alerts.yml

```
groups: - name: homelab interval: 30s rules: - alert: HighCPUUsage expr: 100 - (avg
by(instance) (rate(node_cpu_seconds_total{mode="idle"}[5m])) * 100) > 80 for: 5m
labels: severity: warning annotations: summary: "High CPU usage detected" description:
"CPU usage is above 80%" - alert: HighMemoryUsage expr: (node_memory_MemTotal_bytes -
node_memory_MemAvailable_bytes) / node_memory_MemTotal_bytes * 100 > 90 for: 5m labels:
severity: warning annotations: summary: "High memory usage" description: "Memory usage
is above 90%" - alert: ServiceDown expr: up == 0 for: 1m labels: severity: critical
annotations: summary: "Service is down" description: "{{ $labels.job }}" is down" -
alert: DiskSpaceLow expr: (node_filesystem_avail_bytes / node_filesystem_size_bytes) *
100 < 10 for: 5m labels: severity: warning annotations: summary: "Disk space low"
description: "Less than 10% disk space remaining"
```

Guide 77: Custom Dashboards

Create HomeLab Overview

1. Day-20-Monitoring-Setup|Grafana → Create → Dashboard 2. Add panels:

Panel 1: Services Status

```
Query: up{job=~".*"} Visualization: Stat Thresholds: 0: Red 1: Green
```

Panel 2: CPU Usage

```
Query: 100 - (avg by(instance) (rate(node_cpu_seconds_total{mode="idle"}[5m])) * 100)
Visualization: Time series
```

Panel 3: Memory Usage

```
Query: (node_memory_MemTotal_bytes - node_memory_MemAvailable_bytes) /
node_memory_MemTotal_bytes * 100 Visualization: Gauge
```

Panel 4: Day-03-05-Docker-Infrastructure|Docker Containers

```
Query: count(container_last_seen{}) Visualization: Stat
```

Panel 5: Network Traffic

```
Query: rate(node_network_receive_bytes_total[5m]) Visualization: Time series
```

Panel 6: Disk Usage

```
Query: 100 - ((node_filesystem_avail_bytes / node_filesystem_size_bytes) * 100)
Visualization: Bar gauge
```

3. Save dashboard: "HomeLab Overview" 4. Set as home dashboard

Guide 78: Mobile Access

Day-20-Monitoring-Setup|Grafana Mobile App

1. Install Day-20-Monitoring-Setup|Grafana app (iOS/Android) 2. Add server: <http://192.168.50.10:3010> 3. Or via Day-01-02-Foundation-and-Accounts|Tailscale: <http://100.x.x.x:3010> 4. View dashboards on phone!

Uptime Kuma Status Page

1. Uptime Kuma → Status Page 2. Create public status page 3. Share URL with family 4. Shows all services status

Guide 79: Log Analysis

Useful Day-20-Monitoring-Setup|Loki Queries

View Day-13-14-Media-Stack-Plex|Plex logs:

```
{container="plex"}
```

Find errors:

```
{container=~".*" } |= "error"
```

Count errors per service:

```
sum by (container) (count_over_time({container=~".*" } |= "error" [1h]))
```

Failed login attempts:

```
{container="authelia"} |= "failed"
```

High response times:

```
{container="nginx"} | json | response_time > 1000
```

Guide 80: Maintenance Dashboard

Create Maintenance Checklist

Day-20-Monitoring-Setup|Grafana Dashboard with: - Last backup time - Container update status - Disk space trends - Certificate expiry dates - Uptime stats - Error rates - Performance trends

Alerts for: - Backup failures - Updates available - Certificate expiring soon - Disk space low - High error rates

Volume 11 Complete!

You now have: - ■ Day-20-Monitoring-Setup|Prometheus collecting all metrics - ■ Day-20-Monitoring-Setup|Grafana visualizing everything - ■ Day-20-Monitoring-Setup|Loki aggregating all logs - ■ Uptime Kuma monitoring availability - ■ Alertmanager sending notifications - ■ Custom dashboards for overview - ■ Mobile access to monitoring - ■ Complete observability!

Monitoring Stack:

```
Data Collection: ■■ Prometheus (metrics) ■■ Loki (logs) ■■ Node Exporter (system) ■■ cAdvisor (containers) ■■ Promtail (log shipping) Visualization: ■■ Grafana (dashboards) ■■ Uptime Kuma (status pages) Alerting: ■■ Alertmanager (routing) ■■ ntfy.sh (notifications) Access: - Grafana: http://192.168.50.10:3010 - Prometheus: http://192.168.50.10:9090 - Uptime Kuma: http://192.168.50.10:3001
```

Next: Volume 12 - Backup Strategy

↑ Back to Top

■ Volume 12: Backup Strategy (Day 13)

Source: Day-21-Backup-Strategy Tags: #homelab #backup #disaster-recovery #kopia

Volume 12: Backup Strategy (Day 13)

Automated Encrypted Backups to Day-21-Backup-Strategy|pCloud

This volume covers complete backup automation and disaster recovery.

What You'll Complete

- Day-21-Backup-Strategy|Kopia backup client - Day-21-Backup-Strategy|pCloud offsite storage - Automated daily backups - Encrypted backups (AES-256) - Backup verification - Disaster recovery plan - Restore procedures

Prerequisites

- Volumes 01-11 complete - Day-21-Backup-Strategy|pCloud account (2TB) - External 4TB SSD for local cache - Backup locations planned

Guide 81: Day-21-Backup-Strategy|Kopia Setup

Deploy Day-21-Backup-Strategy|Kopia

```
docker compose -f docker-compose-master.yml up -d kopia
```

Access: <http://192.168.50.10:51515>

Initial Setup

1. Create repository 2. Choose storage: WebDAV (for Day-21-Backup-Strategy|pCloud) 3. Configuration:

```
URL: https://webdav.pcloud.com Username: steve-smithit@outlook.com Password: [from  
1Password] Path: /Backups/M4-HomeLab
```

4. Encryption: - Enable: Yes - Password: [strong passphrase] - Save to
Day-01-02-Foundation-and-Accounts|1Password immediately!

■■ CRITICAL: Without encryption password, backups are unrecoverable!

Guide 82: Backup Configuration

What to Backup

Critical Data:

```
Priority 1 (Daily): ■■ Docker configs: ~/HomeLab/Docker/Configs/ ■■ Scripts:  
~/HomeLab/Scripts/ ■■ 1Password exports: ~/HomeLab/Documentation/ ■■ Home Assistant:  
~/HomeLab/Docker/Data/homeassistant/ ■■ Databases: ~/HomeLab/Docker/Data/*/ Priority 2  
(Weekly): ■■ Media metadata: ~/HomeLab/Docker/Data/plex/ ■■ Photos:  
/Volumes/External4TB/Photos/ ■■ Documents: /Volumes/External4TB/Documents/ Priority 3  
(Monthly): ■■ Media: /Volumes/External4TB/Media/ (optional - large) ■■ VM snapshots:  
/Volumes/External4TB/VMs/Backups/
```

Create Backup Policies

Policy 1: Critical Daily

```
Name: Critical-Daily Paths: - ~/HomeLab/Docker/Configs - ~/HomeLab/Scripts -  
~/HomeLab/Documentation Schedule: Daily at 02:00 Retention: - Keep daily for 7 days -  
Keep weekly for 4 weeks - Keep monthly for 12 months
```

Policy 2: Data Weekly

```
Name: Data-Weekly Paths: - /Volumes/External4TB/Photos - ~/HomeLab/Docker/Data
Schedule: Weekly on Sunday at 03:00 Retention: - Keep weekly for 8 weeks - Keep monthly
for 6 months
```

Guide 83: Automation Scripts

Pre-Backup Script

Create: ~/HomeLab/Scripts/Backup/pre-backup.sh

```
#!/bin/bash # Pre-backup script - prepare data for backup echo "Starting pre-backup
tasks..." # Export lPassword vault op document get "HomeLab Backup" --output
~/HomeLab/Documentation/lpassword-export.json # Stop services that need consistent
backup docker stop mariadb postgres sleep 5 # Dump databases docker exec mariadb
mysqldump --all-databases > ~/HomeLab/Documentation/mariadb-backup.sql docker exec
postgres pg_dumpall > ~/HomeLab/Documentation/postgres-backup.sql # Create service list
docker ps --format "{{.Names}}" > ~/HomeLab/Documentation/running-containers.txt #
Export Docker compose files cp ~/HomeLab/Docker/Compose/*.yaml
~/HomeLab/Documentation/compose-backup/ echo "Pre-backup complete!"
```

Post-Backup Script

Create: ~/HomeLab/Scripts/Backup/post-backup.sh

```
#!/bin/bash # Post-backup script - restart services echo "Starting post-backup
tasks..." # Restart stopped services docker start mariadb postgres # Verify backup
completed if [ $? -eq 0 ]; then echo "Backup completed successfully!" curl -d "HomeLab
backup completed successfully" ntfy.sh/stevehomelab-alerts-x7k9m2 else echo "Backup
FAILED!" curl -d "■■■ HomeLab backup FAILED! Check logs."
ntfy.sh/stevehomelab-alerts-x7k9m2 fi echo "Post-backup complete!"
```

Make executable:

```
chmod +x ~/HomeLab/Scripts/Backup/*.sh
```

Guide 84: Backup Scheduling

Automated Backups

Daily critical backup:

```
# Create LaunchAgent cat > ~/Library/LaunchAgents/com.homelab.backup.daily.plist <<
'EOF' Label com.homelab.backup.daily ProgramArguments /bin/bash
/Users/steve/HomeLab/Scripts/Backup/daily-backup.sh StartCalendarInterval Hour 2 Minute
0 StandardOutPath /Users/steve/HomeLab/Documentation/backup-daily.log EOF launchctl
load ~/Library/LaunchAgents/com.homelab.backup.daily.plist
```

Create Backup Scripts

~/HomeLab/Scripts/Backup/daily-backup.sh:

```
#!/bin/bash # Run pre-backup ~/HomeLab/Scripts/Backup/pre-backup.sh # Run Kopia backup
docker exec kopia kopia snapshot create ~/HomeLab/Docker/Configs \ --description "Daily
critical backup" docker exec kopia kopia snapshot create ~/HomeLab/Scripts \
--description "Daily scripts backup" # Run post-backup
~/HomeLab/Scripts/Backup/post-backup.sh
```

Guide 85: Backup Verification

Verify Backups

```
# List snapshots docker exec kopia kopia snapshot list # Verify repository docker exec
kopia kopia repository validate-provider # Check last backup docker exec kopia kopia
snapshot list --latest # Check backup size docker exec kopia kopia snapshot estimate
~/HomeLab/Docker/Configs
```

Test Restore

Regularly test restores:

```
# Restore to test location docker exec kopia kopia restore [snapshot-id]
/tmp/restore-test # Verify files ls -la /tmp/restore-test # Clean up rm -rf
/tmp/restore-test
```

Do this monthly!

Guide 86: Disaster Recovery Plan

Scenario 1: Single Service Failure

Example: Day-13-14-Media-Stack-Plex|Plex container corrupted

```
# Stop container docker stop plex # Restore config from backup docker exec kopia kopia
restore [snapshot-id] ~/HomeLab/Docker/Data/plex # Start container docker start plex
```

Time: 5-10 minutes

Scenario 2: M4 Mac Mini - Complete System Overview|M4 Mac Mini Failure

Full system restore:

1. Get new M4 or repair
2. Install macOS (Volume 02)
3. Setup network (Volume 02)
4. Install Day-03-05-Docker-Infrastructure|Docker (Volume 03)
5. Install Day-21-Backup-Strategy|Kopia:

```
docker run -d --name kopia \ -v ~/HomeLab/Docker/Data/kopia:/app/config \
kopia/kopia:latest server start \ --insecure
```

6. Connect to Day-21-Backup-Strategy|pCloud:

```
docker exec kopia kopia repository connect webdav \ --url https://webdav.pcloud.com \
--username steve-smithit@outlook.com \ --password [from 1Password] # Enter encryption
password from 1Password
```

7. List available snapshots:

```
docker exec kopia kopia snapshot list
```

8. Restore Day-03-05-Docker-Infrastructure|Docker configs:

```
docker exec kopia kopia restore [snapshot-id] ~/HomeLab/
```

9. Start all services:

```
cd ~/HomeLab/Docker/Compose docker compose -f docker-compose-master.yml up -d
```

10. Verify everything works

Time: 2-4 hours

Scenario 3: External SSD Failure

4TB drive dies:

1. Get new 4TB SSD
2. Format (Volume 02, Guide 04)
3. Restore photos:

```
docker exec kopia kopia restore [photos-snapshot] /Volumes/External4TB/Photos
```

4. Restore media metadata:

```
docker exec kopia kopia restore [plex-snapshot] ~/HomeLab/Docker/Data/plex
```

5. Media files: - If backed up: Restore from Day-21-Backup-Strategy|Kopia - If not: Re-download from sources - Or: Accept loss (media is replaceable)

Time: Hours to days (depending on data size)

Scenario 4: Complete Disaster

House fire, theft, etc. - everything lost:

1. Get new hardware (M4 + storage)
2. Access Day-21-Backup-Strategy|pCloud from any device - Login to Day-21-Backup-Strategy|pCloud.com - All backups safe in cloud!
3. Follow Scenario 2 (M4 restore)
4. All data recovered! ■

This is why offsite backups matter!

Guide 87: Backup Monitoring

Add to Day-20-Monitoring-Setup|Grafana

Create Backup Dashboard:

1. Panel: Last Backup Time

```
Query: Check Kopia logs for last backup timestamp Visualization: Stat Alert: If > 25 hours ago
```

2. Panel: Backup Size Trend

```
Query: Track backup size over time Visualization: Time series
```

3. Panel: Backup Success Rate

```
Query: Count successful vs failed backups Visualization: Gauge
```

Add to Uptime Kuma

1. Add monitor: Day-21-Backup-Strategy|Kopia Web UI 2. Type: HTTP 3. URL: http://192.168.50.10:51515

Guide 88: Backup Best Practices

The 3-2-1 Rule

3 copies of data: - ■ Original: M4 Mac Mini - Complete System Overview|M4 Mac Mini - ■ Local backup: External 4TB SSD - ■ Offsite: Day-21-Backup-Strategy|pCloud (2TB)

2 different media: - ■ Internal SSD - ■ External SSD - ■ Cloud storage

1 offsite: - ■ Day-21-Backup-Strategy|pCloud (encrypted)

Security

- ■ Encryption enabled (AES-256) - ■ Password in Day-01-02-Foundation-and-Accounts|1Password - ■ Test restores monthly - ■ Verify backups weekly - ■ Monitor backup jobs - ■ Alert on failures

What NOT to Backup

Skip these (save space): - Day-03-05-Docker-Infrastructure|Docker images (can re-pull) - Media files (can re-download) - Temp files - Caches - Logs (keep recent only)

Guide 89: Backup Costs

Day-21-Backup-Strategy|pCloud Storage

Current plan: 2TB - Cost: ~£175 lifetime OR ~£8/month - Used for: Critical data + photos - Encrypted before upload

If need more: - Consider additional cloud provider - Or local NAS backup - Or external HDDs rotated offsite

Time Investment

Initial setup: 4 hours (this volume) Daily maintenance: 0 minutes (automated!) Monthly verification: 15 minutes Yearly testing: 2 hours (full restore test)

Worth it! Data is irreplaceable.

Guide 90: Emergency Contacts

Save to Day-01-02-Foundation-and-Accounts|1Password:

```
HomeLab Emergency Contacts: Kopia Encryption Password: [IN 1PASSWORD!] pCloud Login:
[IN 1PASSWORD!] Last Backup: [Check Grafana] Restore Command: docker exec kopia kopia
restore [snapshot-id] /restore-path Support: - Kopia: https://kopia.io/docs - pCloud:
support@pcloud.com This Document Location: pCloud: /Backups/M4-HomeLab/documentation.md
1Password: HomeLab Vault → Disaster Recovery
```

Volume 12 Complete!

You now have: - ■ Day-21-Backup-Strategy|Kopia backing up to Day-21-Backup-Strategy|pCloud (encrypted) -
■ Daily automated backups - ■ Pre/post backup scripts - ■ 3-2-1 backup strategy - ■ Disaster recovery plan -
■ Tested restore procedures - ■ Monitoring and alerts - ■ Peace of mind! ■

Backup Summary:

```
Critical Data (Daily @ 2AM): ■■ Docker configs ■■ Scripts ■■ Documentation ■■ Service
databases Photos/Docs (Weekly): ■■ Immich photos ■■ Paperless documents ■■ Personal
files Destination: ■■ pCloud (2TB) - encrypted, offsite ■■ External SSD - local cache
Recovery Time: ■■ Single service: 5-10 minutes ■■ Full system: 2-4 hours ■■ Complete
disaster: 1 day All encrypted with AES-256 ■ Password safely in 1Password ■
```

Next: Volume 13 - Maintenance & Day-22-Maintenance-and-RMM|Action1 RMM

↑ Back to Top

■ Volume 13: Maintenance & Action1 RMM (Day 14)

Source: Day-22-Maintenance-and-RMM Tags: #homelab #maintenance #updates #rmm

Volume 13: Maintenance & Day-22-Maintenance-and-RMM|Action1 RMM (Day 14)

Automated Maintenance and Remote Management

This volume covers ongoing maintenance and remote management with
Day-22-Maintenance-and-RMM|Action1.

What You'll Complete

- Day-22-Maintenance-and-RMM|Action1 RMM enrollment (M4 + VMs) - Automated update procedures - Maintenance schedules - Health checks - Performance optimization - Troubleshooting guides - Update automation

Prerequisites

- Volumes 01-12 complete - Day-22-Maintenance-and-RMM|Action1 account (from Volume 01) - All systems running - Admin access to all devices

Guide 91: Day-22-Maintenance-and-RMM|Action1 RMM Setup

Install Day-22-Maintenance-and-RMM|Action1 Agent on M4

1. Login to Day-22-Maintenance-and-RMM|Action1: - Go to: <https://app.action1.com> - Email: steve-smithit@outlook.com - Password: [from Day-01-02-Foundation-and-Accounts|1Password]
2. Download agent: - Endpoints → Add Endpoint - Download for macOS - Save to: ~/Downloads/
3. Install agent:

```
# Open package open ~/Downloads/Action1-Agent-*.pkg # Follow installation wizard #  
Enter admin password when prompted
```
4. Verify enrollment: - Day-22-Maintenance-and-RMM|Action1 dashboard → Endpoints - Should see: M4-HomeLab (online)

Install on Windows VM

1. RDP to Windows: 192.168.50.52
2. Download agent from Day-22-Maintenance-and-RMM|Action1 portal
3. Run installer: Day-22-Maintenance-and-RMM|Action1-Agent-Windows.exe
4. Verify: Windows-HomeLab appears in portal

Install on Linux VMs

Ubuntu Server:

```
ssh steve@192.168.50.51 # Download agent wget
https://[your-action1-url]/Action1-Agent-Linux.sh # Install sudo bash
Action1-Agent-Linux.sh # Verify sudo systemctl status action1-agent
```

Kali Linux:

```
ssh steve@192.168.50.53 # Same process as Ubuntu
```

Install on Day-06-08-Proxmox-Hypervisor|Proxmox

```
ssh root@192.168.50.50 # Download and install wget
https://[your-action1-url]/Action1-Agent-Linux.sh bash Action1-Agent-Linux.sh
```

Now all 5 systems managed in Day-22-Maintenance-and-RMM|Action1!

Guide 92: Day-22-Maintenance-and-RMM|Action1 Configuration

Create Device Groups

In Day-22-Maintenance-and-RMM|Action1:

1. Devices → Groups → Create

Groups:

```
Group: HomeLab-Servers - M4-HomeLab - Proxmox Group: HomeLab-VMs - Windows-HomeLab -
Ubuntu-Server - Kali-Linux Group: HomeLab-All - All devices
```

Set Up Policies

Policy 1: Daily Health Check

```
Name: Daily Health Check Target: HomeLab-All Schedule: Daily at 08:00 Actions: - Check
disk space - Check CPU/RAM usage - Check critical services - Report if issues
```

Policy 2: Weekly Updates

```
Name: Weekly Updates Target: HomeLab-All Schedule: Sunday at 03:00 Actions: - Check for
updates - Install security updates - Reboot if needed - Send report
```

Policy 3: Monthly Maintenance

```
Name: Monthly Cleanup Target: HomeLab-All Schedule: First Sunday at 04:00 Actions: -
Clear temp files - Clear old logs - Optimize databases - Defragment (Windows only)
```

Guide 93: Maintenance Scripts

Create Maintenance Directory

```
mkdir -p ~/HomeLab/Scripts/Maintenance
```

Daily Health Check

Create: ~/HomeLab/Scripts/Maintenance/daily-health-check.sh

```
#!/bin/bash # Daily Health Check Script
LOG="/Users/steve/HomeLab/Documentation/health-check.log" TIMESTAMP=$(date "+%Y-%m-%d
%H:%M:%S") echo "=== Health Check: $TIMESTAMP ===" >> $LOG # Check disk space
DISK_USAGE=$(df -h / | awk 'NR==2 {print $5}' | sed 's/%//') echo "Disk Usage:
$DISK_USAGE%" >> $LOG if [ $DISK_USAGE -gt 90 ]; then curl -d "■■■ Disk space critical:
${DISK_USAGE}%" ntfy.sh/stevehomelab-alerts-x7k9m2 fi # Check RAM usage
RAM_USAGE=$(vm_stat | grep "Pages active" | awk '{print $3}' | sed 's/\.///') echo "RAM
Usage: $RAM_USAGE" >> $LOG # Check Docker containers RUNNING=$(docker ps --format
"{{.Names}}" | wc -l) echo "Running containers: $RUNNING" >> $LOG if [ $RUNNING -lt 60
]; then curl -d "■■■ Some containers stopped! Only $RUNNING running."
ntfy.sh/stevehomelab-alerts-x7k9m2 fi # Check VPN (Gluetun) VPN_STATUS=$(docker inspect
gluetun --format='{{.State.Status}}') if [ "$VPN_STATUS" != "running" ]; then curl -d
"■■■ VPN down! Downloads exposed!" ntfy.sh/stevehomelab-alerts-x7k9m2 fi # Check Proxmox
ping -c 1 192.168.50.50 > /dev/null 2>&1 if [ $? -ne 0 ]; then curl -d "■■■ Proxmox
unreachable!" ntfy.sh/stevehomelab-alerts-x7k9m2 fi echo "Health check complete" >>
$LOG echo "" >> $LOG
```

Weekly Update Script

Create: ~/HomeLab/Scripts/Maintenance/weekly-updates.sh

```
#!/bin/bash # Weekly Update Script LOG="/Users/steve/HomeLab/Documentation/updates.log"
TIMESTAMP=$(date "+%Y-%m-%d %H:%M:%S") echo "=== Weekly Updates: $TIMESTAMP ===" >>
$LOG # Update macOS echo "Checking macOS updates..." >> $LOG softwareupdate --list >>
$LOG 2>&1 # Note: Auto-install disabled, check manually # Update Homebrew echo "Updating
Homebrew..." >> $LOG brew update >> $LOG 2>&1 brew upgrade >> $LOG 2>&1 brew cleanup >>
$LOG 2>&1 # Update Docker containers echo "Updating Docker containers..." >> $LOG cd
~/HomeLab/Docker/Compose docker compose -f docker-compose-master.yml pull >> $LOG 2>&1
docker compose -f docker-compose-master.yml up -d >> $LOG 2>&1 # Clean Docker echo
"Cleaning Docker..." >> $LOG docker system prune -af --volumes >> $LOG 2>&1 # Update
Ollama models echo "Updating Ollama models..." >> $LOG docker exec ollama ollama pull
llama3.2:3b >> $LOG 2>&1 echo "Updates complete" >> $LOG curl -d "■■ Weekly updates
completed successfully" ntfy.sh/stevehomelab-alerts-x7k9m2
```

Monthly Cleanup Script

Create: ~/HomeLab/Scripts/Maintenance/monthly-cleanup.sh

```
#!/bin/bash # Monthly Cleanup Script
LOG="/Users/steve/HomeLab/Documentation/cleanup.log" TIMESTAMP=$(date "+%Y-%m-%d
%H:%M:%S") echo "=== Monthly Cleanup: $TIMESTAMP ===" >> $LOG # Clear temp files echo
"Clearing temp files..." >> $LOG rm -rf ~/Library/Caches/* >> $LOG 2>&1 rm -rf /tmp/* >>
$LOG 2>&1 # Clear old logs (keep 30 days) echo "Cleaning old logs..." >> $LOG find
~/HomeLab/Documentation/*.log -mtime +30 -delete >> $LOG 2>&1 # Clean Docker logs echo
"Cleaning Docker logs..." >> $LOG truncate -s 0 $(docker inspect
--format='{{.LogPath}}' $(docker ps -qa)) >> $LOG 2>&1 # Optimize Plex database echo
"Optimizing Plex..." >> $LOG docker exec plex sqlite3 /config/Library/Application\
Support/Plex\ Media\ Server/Plug-in\ Support/Databases/com.plexapp.plugins.library.db
"VACUUM;" >> $LOG 2>&1 # Verify backups echo "Verifying backups..." >> $LOG docker exec
kopia kopia repository validate-provider >> $LOG 2>&1 echo "Cleanup complete" >> $LOG
curl -d "■ Monthly cleanup completed" ntfy.sh/stevehomelab-alerts-x7k9m2
```

Make Scripts Executable

```
chmod +x ~/HomeLab/Scripts/Maintenance/*.sh
```

Guide 94: Automated Scheduling

Schedule via LaunchAgents

Daily Health Check:

```
cat > ~/Library/LaunchAgents/com.homelab.health.plist << 'EOF' Label com.homelab.health
ProgramArguments /bin/bash
/Users/steve/HomeLab/Scripts/Maintenance/daily-health-check.sh StartCalendarInterval
Hour 8 Minute 0 EOF launchctl load ~/Library/LaunchAgents/com.homelab.health.plist
```

Weekly Updates:

```
cat > ~/Library/LaunchAgents/com.homelab.updates.plist << 'EOF' Label
com.homelab.updates ProgramArguments /bin/bash
/Users/steve/HomeLab/Scripts/Maintenance/weekly-updates.sh StartCalendarInterval
Weekday 0 Hour 3 EOF launchctl load ~/Library/LaunchAgents/com.homelab.updates.plist
```

Guide 95: Troubleshooting Guide

Common Issues

Day-03-05-Docker-Infrastructure|Docker Container Won't Start

```
# Check logs docker logs # Check if port already in use lsof -i : # Restart container
docker restart # Rebuild container docker compose up -d --force-recreate
```


Service Not Accessible

```
# Check if container running docker ps | grep # Check network docker network inspect  
homelab-network # Check firewall # macOS: System Settings → Network → Firewall # Test  
locally curl http://localhost:
```

High CPU Usage

```
# Find culprit docker stats # Check M4 resources top # Restart high-usage container  
docker restart
```

Out of Disk Space

```
# Check usage df -h # Find large files du -sh /Volumes/External4TB/* # Clean Docker  
docker system prune -a --volumes # Clear Plex temp rm -rf  
~/HomeLab/Docker/Data/plex/transcode/*
```

VPN Not Working

```
# Check Gluetun docker logs gluetun # Test VPN IP docker exec gluetun curl ifconfig.me #  
Restart VPN docker restart gluetun qbittorrent
```

Guide 96: Performance Optimization

M4 Optimization

Check resources:

```
# CPU usage top -l 1 | head -n 10 # RAM usage vm_stat # Disk I/O iostat -w 5
```

Optimize Day-03-05-Docker-Infrastructure|Docker:

```
# Clean unused data docker system prune -a --volumes # Optimize OrbStack settings #  
Settings → Resources # Adjust CPU/RAM allocation
```

Optimize macOS: - Disable unused startup items - Clear caches monthly - Restart weekly - Monitor Activity Monitor

Container Optimization

Review resource limits:

```
# In docker-compose.yml services: plex: deploy: resources: limits: cpus: '2' memory: 4G  
reservations: memory: 2G
```

Guide 97: Update Procedures

Before Updates

```
# Backup configs ~/HomeLab/Scripts/Backup/pre-backup.sh # Create VM snapshots # Proxmox  
→ Each VM → Snapshots → Take Snapshot # Note: "pre-update-[date]"
```

Update Day-03-05-Docker-Infrastructure|Docker Containers

```
cd ~/HomeLab/Docker/Compose # Pull latest images docker compose -f  
docker-compose-master.yml pull # Recreate containers docker compose -f  
docker-compose-master.yml up -d # Check logs docker compose -f  
docker-compose-master.yml logs -f
```

Update VMs

Windows: - Windows Update (automated) - Or via Day-22-Maintenance-and-RMM|Action1: Run Windows Update policy

Ubuntu/Kali:

```
ssh steve@192.168.50.51 sudo apt update && sudo apt full-upgrade -y
```

Update Day-06-08-Proxmox-Hypervisor|Proxmox

```
ssh root@192.168.50.50 apt update apt full-upgrade -y # May require reboot reboot
```

Guide 98: Monthly Checklist

Every Month:

- ☐ Review Day-22-Maintenance-and-RMM|Action1 reports - ☐ Check Day-20-Monitoring-Setup|Grafana dashboards - ☐ Verify all backups successful - ☐ Test restore (random file) - ☐ Review disk space usage - ☐ Update all containers - ☐ Update all VMs - ☐ Check for security updates - ☐ Review access logs - ☐ Clean Day-03-05-Docker-Infrastructure|Docker system - ☐ Optimize Day-13-14-Media-Stack-Plex|Plex database - ☐ Check certificate expiry - ☐ Review failed login attempts - ☐ Update documentation if needed

Time required: 1-2 hours

Guide 99: Quarterly Tasks

Every 3 Months:

- [] Full system backup test - [] Review and update passwords - [] Audit user access - [] Review security logs - [] Check hardware health - [] Clean physical dust - [] Update documentation - [] Review and optimize automations - [] Capacity planning review

Time required: 3-4 hours

Guide 100: Emergency Procedures

M4 Unresponsive

1. Try SSH: `ssh steve@192.168.50.10` 2. Try Day-01-02-Foundation-and-Accounts|Tailscale: `ssh steve@100.x.x.x` 3. Physical access: Connect monitor + keyboard 4. Hard reset: Hold power button 10 seconds 5. Recovery: Boot recovery mode (Cmd+R on startup)

Critical Service Down

1. Check container: `docker ps` 2. Check logs: `docker logs` 3. Restart: `docker restart` 4. Recreate: `docker compose up -d --force-recreate` 5. Restore config: From backup if corrupted

Network Issues

1. Check router: Ping 192.168.50.1 2. Check switch: Check lights 3. Check cables: Reseat connections 4. Check adapters: `ifconfig` 5. Restart networking: `sudo networksetup -setnetworkserviceenabled "Ethernet" off && sudo networksetup -setnetworkserviceenabled "Ethernet" on`

Volume 13 Complete!

You now have: - ■ Day-22-Maintenance-and-RMM|Action1 RMM managing all systems - ■ Automated daily health checks - ■ Weekly update automation - ■ Monthly cleanup scripts - ■ Comprehensive troubleshooting guide - ■ Performance optimization - ■ Emergency procedures - ■ Complete maintenance schedule!

Maintenance Schedule:

Daily (Automated @ 8AM): - Health checks - Disk space monitoring - Service status checks - Alert if issues Weekly (Automated @ 3AM Sunday): - Update Docker containers - Clean

[illegible]

Hardware Summary

M4 Mac Mini - Complete System Overview

Model: Mac Mini (M4, 2024) CPU: Apple M4 (10-core) RAM: 32GB unified memory Storage: 512GB internal SSD Serial: [in 1Password] Location: [Your office/room] Network Interfaces: - en0: 10GbE (built-in) → Primary - en1: 5GB USB-C adapter → Backup/Failover - en2: 2.5GB adapter → VM isolated network IP Addresses: - Primary: 192.168.50.10 - Backup: 192.168.50.11 - VM Network: 192.168.51.1 - Tailscale: 100.x.x.x

External Storage

Model: Samsung T7 (or your model) Capacity: 4TB Connection: USB-C (Thunderbolt 4) Mount: /Volumes/External4TB Format: APFS Contents: ■■ Media: 2TB (Plex library) ■■ Photos: 200GB (Immich) ■■ Frigate: 500GB (camera recordings) ■■ VMs: 1TB (Proxmox VMs) ■■ Backups: 200GB (local cache) ■■ Free: ~100GB

Network Hardware

Router: [Your router model] - IP: 192.168.50.1 - Speed: 5Gbps CityFibre Switch: TP-Link SX1008 - Ports: 8x 10GbE - Location: [Your location] Adapters: - Built-in 10GbE (M4) - 5GB USB-C Ethernet - 2.5GB Ethernet

IP Address Allocation

Network: 192.168.50.0/24 (Main) Infrastructure: 192.168.50.1 - Router 192.168.50.10 - M4 Mac Mini (primary) 192.168.50.11 - M4 Mac Mini (backup interface) 192.168.50.50 - Proxmox Host Virtual Machines: 192.168.50.51 - Ubuntu Server 192.168.50.52 - Windows 11 Pro 192.168.50.53 - Kali Linux Reserved for Future: 192.168.50.60-99 - Additional VMs/services Network: 192.168.51.0/24 (Isolated VM Network) 192.168.51.1 - M4 Bridge 192.168.51.10-254 - VM private network Tailscale VPN: 100.x.x.x - M4 (Tailscale mesh network)

Service Access URLs

Infrastructure

Proxmox: https://192.168.50.10:8006 Portainer: https://192.168.50.10:9443 Nginx Proxy Manager: http://192.168.50.10:81 Watchtower: (No UI - runs automatically)

Media Services

```
Plex: http://192.168.50.10:32400/web Sonarr: http://192.168.50.10:8989 Radarr:
http://192.168.50.10:7878 Prowlarr: http://192.168.50.10:9696 qBittorrent:
http://192.168.50.10:8112 Bazarr: http://192.168.50.10:6767 Overseerr:
http://192.168.50.10:5055 Tdarr: http://192.168.50.10:8265
```

Smart Home

```
Home Assistant: http://192.168.50.10:8123 Frigate: http://192.168.50.10:5000 Scrypted:
http://192.168.50.10:11080
```

AI Services

```
Ollama: http://192.168.50.10:11434 Open WebUI: http://192.168.50.10:3000 Paperless-ngx:
http://192.168.50.10:8010 Immich: http://192.168.50.10:2283
```

Security

```
Pi-hole: http://192.168.50.10:8080/admin Authelia: http://192.168.50.10:9091 CrowdSec:
https://app.crowdsec.net
```

Monitoring

```
Grafana: http://192.168.50.10:3010 Prometheus: http://192.168.50.10:9090 Loki:
http://192.168.50.10:3100 Uptime Kuma: http://192.168.50.10:3001
```

Backup

```
Kopia: http://192.168.50.10:51515 pCloud: https://my.pcloud.com
```

Remote Access

```
Tailscale: https://login.tailscale.com DuckDNS: https://www.duckdns.org Your Domain:
stevehomelab.duckdns.org
```

External Access (via Day-01-02-Foundation-and-Accounts|DuckDNS)

Setup in Nginx Proxy Manager:

```
plex.stevehomelab.duckdns.org → Plex home.stevehomelab.duckdns.org → Home Assistant
ai.stevehomelab.duckdns.org → Open WebUI photos.stevehomelab.duckdns.org → Immich
docs.stevehomelab.duckdns.org → Paperless requests.stevehomelab.duckdns.org →
Overseerr monitor.stevehomelab.duckdns.org → Grafana proxmox.stevehomelab.duckdns.org
→ Proxmox All with Let's Encrypt SSL certificates ■
```

Quick Command Reference

Day-03-05-Docker-Infrastructure|Docker

```
# View all containers docker ps # View specific container logs docker logs -f # Restart container docker restart # Update all containers cd ~/HomeLab/Docker/Compose docker compose -f docker-compose-master.yml pull docker compose -f docker-compose-master.yml up -d # Clean system docker system prune -a --volumes
```

Day-06-08-Proxmox-Hypervisor|Proxmox

```
# SSH to Proxmox ssh root@192.168.50.50 # List VMs qm list # Start/Stop VM qm start 101 qm stop 101 # VM status qm status 101 # Create snapshot qm snapshot 101 pre-update # Console access qm terminal 101
```

Backups

```
# Manual backup docker exec kopia kopia snapshot create ~/HomeLab/Docker/Configs # List snapshots docker exec kopia kopia snapshot list # Restore snapshot docker exec kopia kopia restore [snapshot-id] /restore-path
```

Monitoring

```
# Check system resources htop top # Check disk space df -h # Check network ifconfig netstat -an
```

Credentials Management

All credentials stored in Day-01-02-Foundation-and-Accounts|1Password:

```
HomeLab Vault contains: Accounts: - M4 Mac Mini (admin account) - Proxmox (root) - Windows VM (Steve) - Ubuntu VM (steve) - Kali VM (steve + root) - All Docker services - Cloud accounts API Keys: - Plex - DuckDNS - Tailscale - OpenAI (if used) - ExpressVPN Backup: - Kopia encryption password - pCloud credentials Certificates: - Let's Encrypt (auto-renewed)
```

■■ NEVER store passwords in plain text files!

Family Sharing Guide

Share with Family/Friends

Day-13-14-Media-Stack-Plex|Plex

1. Plex → Settings → Users & Sharing 2. Invite Friend by Email 3. They create Plex account 4. Select what libraries to share 5. They can watch on any device!

Overseerr (Media Requests)

1. Share URL: <https://requests.stevhomelab.duckdns.org> 2. They sign in with Plex account 3. They can request movies/shows 4. You approve 5. Automatic download!

Immich (Photos - Optional)

1. Create user account for family member 2. Share albums 3. They install Immich app 4. Auto-backup their photos

Day-15-16-Smart-Home-Integration|Home Assistant (Smart Home)

1. Create user account 2. Limited dashboard access 3. Can control lights, climate, etc. 4. No access to system settings

System Statistics

Total Resources

M4 Mac Mini - Complete System Overview|M4 Mac Mini:

CPU: 10 cores (6 for Docker, 4 for macOS/VMs) RAM: 32GB (16GB Docker, 16GB macOS/VMs)
Storage: 512GB internal + 4TB external Network: 10GbE primary, 5GB backup, 2.5GB VM

Day-03-05-Docker-Infrastructure|Docker Containers:

Total: 60+ containers Categories: - Infrastructure: 10 - Media: 15 - Smart Home: 5 - AI: 8 - Security: 6 - Monitoring: 10 - Other: 6+

Virtual Machines:

Total: 3 VMs - Windows 11: 8GB RAM, 3 cores, 500GB - Ubuntu: 6GB RAM, 2 cores, 200GB - Kali: 4GB RAM, 2 cores, 150GB Total: 18GB RAM, 7 cores, 850GB

Smart Home:

Total: 43+ devices - 5 Nest cameras - 10 Sonos speakers - 2 EVs (Tesla + Cupra) - 1 Easee charger - + other smart devices

Expansion Ideas

Future Additions

Hardware: - ☐ NAS for additional backup - ☐ More RAM (upgrade to 64GB or 128GB later) - ☐ Additional external storage - ☐ UPS (uninterruptible power supply)

Software/Services: - ☐ Nextcloud (self-hosted cloud) - ☐ Bitwarden (self-hosted password manager) - ☐ Wireguard (additional VPN) - ☐ Jellyfin (Day-13-14-Media-Stack-Plex|Plex alternative) - ☐ Calibre-web (ebook library) - ☐ BookStack (documentation wiki) - ☐ GitLab/Gitea (self-hosted Git) - ☐ Minecraft server (if gaming)

Smart Home: - ☐ More cameras - ☐ Smart locks - ☐ Automated blinds - ☐ Energy monitoring - ☐ Garden irrigation automation

VMs: - ☐ Day-15-16-Smart-Home-Integration|Home Assistant OS VM (instead of Day-03-05-Docker-Infrastructure|Docker) - ☐ pfSense firewall VM - ☐ Additional Linux distros for learning - ☐ Server 2022 (Windows Server)

Support & Resources

Official Documentation

Docker: <https://docs.docker.com> Proxmox: <https://pve.proxmox.com/wiki> Home Assistant: <https://www.home-assistant.io/docs> Plex: <https://support.plex.tv> Grafana: <https://grafana.com/docs>

Community Forums

r/homelab: <https://reddit.com/r/homelab> r/selfhosted: <https://reddit.com/r/selfhosted> r/plex: <https://reddit.com/r/plex> r/homeassistant: <https://reddit.com/r/homeassistant>

Your Documentation

Location: ~/HomeLab/Documentation/ - Volumes 01-14 (these guides) - Build journal - Backup logs - Maintenance logs - Network diagrams

Next Steps

Week 1

- [] Familiarize yourself with all services - [] Test each service thoroughly - [] Set up family access to Day-13-14-Media-Stack-Plex|Plex/Overseerr - [] Configure Day-15-16-Smart-Home-Integration|Home Assistant automations - [] Test backups and restores

Month 1

- [] Monitor system stability - [] Optimize resource allocation - [] Add media to Day-13-14-Media-Stack-Plex|Plex - [] Set up preferred Overseerr workflows - [] Customize Day-15-16-Smart-Home-Integration|Home Assistant dashboard

Ongoing

- [] Keep services updated - [] Monitor Day-20-Monitoring-Setup|Grafana dashboards - [] Review Day-22-Maintenance-and-RMM|Action1 reports - [] Expand smart home - [] Add new services as needed

Congratulations! ■

You've built an enterprise-grade homelab!

What You've Accomplished:

■ 60+ Day-03-05-Docker-Infrastructure|Docker containers running sophisticated services ■ 3 Virtual Machines (Windows, Ubuntu, Kali) ■ Complete media automation (Day-13-14-Media-Stack-Plex|Plex + *arr stack) ■ 43+ smart home devices integrated ■ Local AI services (Day-17-18-AI-Services-LLMs|Ollama, Immich, Paperless) ■ Enterprise security (Pi-hole, Authelia, 2FA) ■ Comprehensive monitoring (Day-20-Monitoring-Setup|Grafana, Day-20-Monitoring-Setup|Prometheus) ■ Automated encrypted backups (Day-21-Backup-Strategy|Kopia → Day-21-Backup-Strategy|pCloud) ■ Remote access (Day-01-02-Foundation-and-Accounts|Tailscale + Day-01-02-Foundation-and-Accounts|DuckDNS) ■ Remote management (Day-22-Maintenance-and-RMM|Action1 RMM)

System Value:

Equivalent Commercial Services:

Plex Pass: £100/year Google Photos: £200/year Office 365: £80/year VPN Service: £100/year Cloud Storage (2TB): £96/year ChatGPT Plus: £200/year Smart Home Hub: £150 Media Automation: £300/year Monitoring Tools: £500/year Total Savings: £1,700+/year Your Setup: One-time hardware + minimal running costs!

Skills Gained:

- Day-03-05-Docker-Infrastructure|Docker containerization - Virtualization (Day-06-08-Proxmox-Hypervisor|Proxmox) - Network configuration - Security hardening - Monitoring & observability - Backup strategies - Smart home automation - AI/ML deployment - System administration - Troubleshooting

Final Checklist

Verify Everything Works:

- ☐ M4 Mac Mini - Complete System Overview|M4 Mac Mini running 24/7 - ☐ All 60+ Day-03-05-Docker-Infrastructure|Docker containers running - ☐ Day-06-08-Proxmox-Hypervisor|Proxmox accessible - ☐ All 3 VMs running - ☐ Day-13-14-Media-Stack-Plex|Plex streaming media - ☐ Day-15-16-Smart-Home-Integration|Home Assistant controlling devices - ☐ Day-20-Monitoring-Setup|Grafana showing metrics - ☐ Backups running daily - ☐ Remote access working (Day-01-02-Foundation-and-Accounts|Tailscale) - ☐ Day-22-Maintenance-and-RMM|Action1 monitoring all systems - ☐ All passwords in Day-01-02-Foundation-and-Accounts|1Password - ☐ Documentation complete

Test Critical Functions:

- ☐ Stream from Day-13-14-Media-Stack-Plex|Plex on phone - ☐ Request media via Overseerr - ☐ Control smart home from anywhere - ☐ Access M4 via SSH remotely - ☐ View Day-20-Monitoring-Setup|Grafana dashboards remotely - ☐ Restore a file from backup - ☐ Receive alert notifications

Thank You!

You did it! You've built something amazing.

This homelab will serve you for years, evolving with your needs.

Remember: - Keep exploring and learning - Join homelab communities - Share your knowledge - Expand as needed - Most importantly: Have fun! ■

Questions? Issues? - Review relevant volume - Check logs - Search communities - Experiment safely

Happy homelabbing! ■

Volume 14 Complete!

All 14 volumes finished!

You now have: - ■ Complete network diagrams - ■ All service URLs documented - ■ Hardware summary - ■ Quick command reference - ■ Family sharing guide - ■ Expansion ideas - ■ Everything documented!

Total Project: 14 days, 60+ services, unlimited possibilities!

↑ Back to Top

=====

■ ■ Proxmox Quick Reference Card

Source: REF-Proxmox-Quick-Reference Tags: #homelab #proxmox #virtualization #hypervisor

■ Day-06-08-Proxmox-Hypervisor|Proxmox Quick Reference Card

Steve's Day-06-08-Proxmox-Hypervisor|Proxmox HomeLab Cheat Sheet

■ Access URLs

```
Proxmox Web GUI: https://192.168.50.50:8006 Username: root | steve@pve Password: [from  
1Password] VMs: ■■ Ubuntu Server: ssh steve@192.168.50.51 ■■ Windows 11:  
rdp://192.168.50.52 ■■ Kali Linux: ssh steve@192.168.50.53
```

■ Quick Commands

VM Management

```
# List all VMs qm list # Start VM qm start 100 # Stop VM qm stop 100 # Reboot VM qm  
reboot 100 # VM status qm status 100 # Enter VM console qm terminal 100
```

Backup & Restore

```
# Manual backup vzdump 100 --storage backups --mode snapshot --compress zstd # List  
backups ls /mnt/external-4tb/Proxmox-Backups/ # Restore VM qmrestore  
/path/to/backup.vma.zst 100 --storage local-lvm
```

Snapshots

```
# Create snapshot qm snapshot 100 pre-update # List snapshots qm listsnapshot 100 #  
Rollback snapshot qm rollback 100 pre-update # Delete snapshot qm delsnapshot 100  
pre-update
```

System Updates

```
# Update Proxmox apt update && apt full-upgrade -y # Check if reboot needed [ -f  
/var/run/reboot-required ] && echo "Reboot needed" # Reboot reboot
```

Storage

```
# List storage pvesm status # Check disk usage df -h # Check VM disk usage qm disk  
rescan
```

Network

```
# Show bridges ip addr show vbr0 # Show VM network qm config 100 | grep net # Test  
connectivity ping 192.168.50.51
```

■ Common Tasks

Create New VM

1. Web GUI → Create VM
2. Set Name, ID
3. Select ISO
4. Configure: CPU/RAM/Disk
5. Set Network: vbr0
6. Finish → Start

Clone VM

```
# CLI qm clone 100 200 --name new-vm-name # Or Web GUI: Right-click VM → Clone
```

Change VM Resources

```
# Add CPU cores qm set 100 --cores 4 # Add RAM qm set 100 --memory 8192 # Resize disk  
(can only grow!) qm resize 100 virtio0 +50G
```

Fix VM Not Starting

```
# Check VM config qm config 100 # Check system logs journalctl -xe # Check storage pvesm  
status # Force unlock VM qm unlock 100 # Try starting again qm start 100
```

■ Monitoring

System Resources

```
# CPU/RAM top htop # Disk I/O iotop # Network iftop nload
```

VM Performance

```
# VM CPU usage qm monitor 100 # Type: info cpus # VM memory qm config 100 | grep memory
```

■ Firewall

Enable Datacenter Firewall

```
# Web GUI: # Datacenter → Firewall → Options # Enable Firewall: Yes # Allow SSH #
Datacenter → Firewall → Add # Direction: IN # Action: ACCEPT # Protocol: TCP # Dest
port: 22
```

■ Backup Schedule

```
Daily: 02:00 AM (all VMs) Storage: /mnt/external-4tb/Proxmox-Backups/ Retention: 7 days
Compression: ZSTD Mode: Snapshot
```

■ Important Locations

```
VM Configs: /etc/pve/qemu-server/ ISOs: /var/lib/vz/template/iso/ Backups:
/var/lib/vz/dump/ (or external storage) Logs: /var/log/pve/
```

■ Emergency Procedures

Day-06-08-Proxmox-Hypervisor|Proxmox Won't Boot

1. Boot into recovery mode (GRUB menu) 2. Check /etc/network/interfaces 3. Check /etc/pve/ permissions 4.
Repair: pve-cluster updatecerts --force

VM Locked

```
qm unlock 100
```

Reset Root Password

```
# Boot from rescue ISO # Mount root partition mount /dev/mapper/pve-root /mnt chroot
/mnt passwd root exit reboot
```

Restore from Backup

1. Web GUI → Storage → backups 2. Select backup file 3. Restore 4. Select target

■ Mobile Access

Via Day-01-02-Foundation-and-Accounts|Tailscale:

```
1. Connect to Tailscale on phone 2. Open browser: https://100.x.x.x:8006 3. Login with
credentials 4. Manage VMs from phone!
```

■ VM IP Addresses

```
192.168.50.50 → Proxmox host 192.168.50.51 → Ubuntu Server 192.168.50.52 → Windows 11
192.168.50.53 → Kali Linux 192.168.50.54 → macOS (optional)
```

■■ Useful Aliases

Add to ~/.bashrc:

```
alias vmlist='qm list' alias vmstart='qm start' alias vmstop='qm stop' alias vmssh='ssh
steve@' alias backupnow='vzdump --all --storage backups --mode snapshot' alias
updateprox='apt update && apt full-upgrade -y'
```

■ Support Resources

-	Day-06-08-Proxmox-Hypervisor Proxmox	Forum:	https://forum.proxmox.com/	-
	Day-06-08-Proxmox-Hypervisor Proxmox	Wiki:	https://pve.proxmox.com/wiki/	-
	r/Day-06-08-Proxmox-Hypervisor Proxmox:	https://reddit.com/r/Proxmox	-	Your docs:
	~/HomeLab/Documentation/			

Keep this handy - you'll use it daily! ■

Created by Vox for Steve's HomeLab ■

↑ Back to Top

■ HomeLab Vault v2.0 - Update Summary

HomeLab Vault v2.0 - Update Summary

Updated: 2025-11-11 22:08:12

■ Changes Applied

Files Renamed (18):

- INDEX.md → 00-START-HERE.md - M4-System-Overview.md → 00-System-Overview.md - QUICK-REFERENCE-GUIDE.md → 00-Quick-Reference.md - Volume-01-Foundation-COMplete.md → Day-01-02-Foundation-and-Accounts.md - Volume-02-M4-Setup-COMplete.md → Day-01-02-M4-Mac-Mini-Setup.md - Volume-03-Day-03-05-Docker-Infrastructure|Docker-Infrastructure.md → Day-03-05-Day-03-05-Docker-Infrastructure|Docker-Infrastructure.md - Volume-04-Day-06-08-Proxmox-Hypervisor|Proxmox-Setup.md → Day-06-08-Day-06-08-Proxmox-Hypervisor|Proxmox-Hypervisor.md - Volume-05-Windows-11-VM.md → Day-09-10-Windows-11-VM.md - Volume-06-Ubuntu-Kali-VMs.md → Day-11-12-Ubuntu-Kali-VMs.md - Volume-07-Media-Stack.md → Day-13-14-Media-Stack-Day-13-14-Media-Stack-Plex|Plex.md - Volume-08-Smart-Home.md → Day-15-16-Smart-Home-Integration.md - Volume-09-AI-Services.md → Day-17-18-AI-Services-LLMs.md - Volume-10-Security-Stack.md → Day-19-Security-Hardening.md - Volume-11-Monitoring-Stack.md → Day-20-Monitoring-Setup.md - Volume-12-Backup-Strategy.md → Day-21-Backup-Strategy.md - Volume-13-Maintenance-Day-22-Maintenance-and-RMM|Action1.md → Day-22-Maintenance-and-RMM.md - Volume-14-Network-Diagrams-Final.md → Day-22-Network-Diagrams-Final.md - Day-06-08-Proxmox-Hypervisor|Proxmox-Quick-Reference.md → REF-Day-06-08-Proxmox-Hypervisor|Proxmox-Quick-Reference.md

New Files Created (4):

1. 00-INSTALLATION-ROADMAP.md - Master guide with 22-day journey - Progress tracker - Quick navigation
2. 00-START-HERE.md - Main entry point - v2.0 overview - Quick links
3. Day-02-BONUS-Deco-Mesh-Setup.md - TP-Link Day-03-Deco-Mesh-WiFi|Deco XE75 setup guide - Sky router WiFi disable - 43+ device migration - IoT network setup
4. Day-05-BONUS-Remote-Access.md - Day-05-Remote-Access|RustDesk server deployment - Apache Day-05-Remote-Access|Guacamole setup - Android client config - Port forwarding guide

■ What's New in v2.0

Organization:

- ■ Logical Day-based file naming (Day-01, Day-02...) - ■ Installation roadmap (clear path) - ■ Better progression (Foundation → Infrastructure → Services → Ops)

Features:

- ■ Self-hosted remote access (Day-05-Remote-Access|RustDesk + Day-05-Remote-Access|Guacamole) - ■ Android-compatible remote desktop - ■ Professional mesh WiFi (TP-Link Day-03-Deco-Mesh-WiFi|Deco XE75)
- ■ Wired backhaul configuration - ■ IoT network for smart devices

Content:

- ■ Service count updated (62+) - ■ 2 new complete bonus guides - ■ Installation roadmap with progress tracking

■ Backup Location

Your original files are backed up at: /Users/user/Documents/OBSIDIAN/PROJECTS/HomeLab/STEVE-HOMELAB-COMPLETE-ALL-14-VOLUMES_BACKUP_20251111_220811

If you need to restore, just copy files from backup folder back to vault folder.

■ Next Steps

1. Open: 00-INSTALLATION-ROADMAP.md
2. Start: Day-01-02-Foundation-and-Accounts.md
3. Follow: Days 1-22 in order
4. Track: Check off progress as you go
5. Reference: Use 00-Quick-Reference.md for lookups

■ File Structure

Your vault now has:

```
00-START-HERE.md (main index) 00-INSTALLATION-ROADMAP.md (your guide) ■
00-System-Overview.md 00-Quick-Reference.md Day-01-02-Foundation-and-Accounts.md
Day-01-02-M4-Mac-Mini-Setup.md Day-02-BONUS-Deco-Mesh-Setup.md (NEW)
Day-03-05-Docker-Infrastructure.md Day-05-BONUS-Remote-Access.md (NEW)
Day-06-08-Proxmox-Hypervisor.md Day-09-10-Windows-11-VM.md Day-11-12-Ubuntu-Kali-VMs.md
Day-13-14-Media-Stack-Plex.md Day-15-16-Smart-Home-Integration.md
Day-17-18-AI-Services-LLMs.md Day-19-Security-Hardening.md Day-20-Monitoring-Setup.md
Day-21-Backup-Strategy.md Day-22-Maintenance-and-RMM.md
Day-22-Network-Diagrams-Final.md REF-Proxmox-Quick-Reference.md V2.0-UPDATE-SUMMARY.md
(this file)
```

■ You're Ready!

Your vault is now v2.0 with: - ■ Logical organization (Day-based) - ■ Clear progression (22-day roadmap) - ■ New features (remote access + mesh WiFi) - ■ Complete guides (4 new files) - ■ Better navigation (roadmap + index)

Start building your epic homelab today! ■

"Looks like I picked the right update script!" ■■

↑ [Back to Top](#)

=====

■■ Master Document Tags

#homelab #master-guide #complete-documentation #m4-mac-mini #foundation #infrastructure #services
#security #operations