Helicarrier Network: Roadmap & Blueprint

Overview:

The Helicarrier Network is a secure, hybrid cloud and local infrastructure, integrating encrypted communication, edge device control, and advanced AI assistance. This roadmap guides you through

setting up the local systems, connecting the remote node at your dad's place, utilizing OpenAI credits to build out a temporary JARVIS, and establishing the foundations for future enhancements.

Phase 1: Setting Up the Core Infrastructure

1. Proxmox Deployment on iMac

- Objective: Virtualize your 2017 iMac into multiple servers.
- Tools Needed: Proxmox, bootable USB, external storage for backups.
- Steps:
    - Install Proxmox onto the iMac using the bootable USB.
    - Create VMs for core services: Home Assistant, Encrypted Storage, Secure Gateway.
    - Set up full-disk encryption on each VM using LUKS or FileVault.
    - Configure regular snapshot backups to an external drive.
- Note: Use the eero Pro 6 for VLANs to isolate Proxmox traffic from general home traffic.

2. Home Assistant Finalization on Pi 4B

- Objective: Get your Home Assistant instance fully functional with OpenAI integration.
- Tools Needed: Home Assistant, OpenAI API key, Nabu Casa.
- Steps:
    - Update Home Assistant to the latest version.
    - Install the OpenAI integration via Home Assistant's UI.

    - Set up a custom assistant using OpenAI for commands like controlling lights.
    - Configure DuckDNS as a fallback for remote access.

Phase 2: Establishing Secure Connections

1. VPN and WireGuard Setup
   - Objective: Create secure tunnels between your home and your dad's network.
   - Tools Needed: Surfshark VPN, WireGuard.
   - Steps:
     - Set up a VPN client on the Proxmox VM using Surfshark's dedicated IP.
     - Install WireGuard on a separate Proxmox VM and configure a site-to-site tunnel.
     - Configure WireGuard on your dad's VPN router for stable connection to the main hub.

2. Deploy Encrypted Data Sync
   - Objective: Synchronize sensitive data between nodes securely.
   - Tools Needed: Syncthing or rsync with SSH.
   - Steps:
     - Set up Syncthing on both the iMac VM and a lightweight Linux instance at your dad's place.
     - Configure folder encryption on Syncthing for data protection.
     - Set automated synchronization schedules and monitor for sync errors.

Phase 3: Temporary AI Integration (JARVIS)

1. Spinning Up JARVIS using OpenAI API
   - Objective: Utilize OpenAI credit to create a temporary JARVIS assistant.
   - Tools Needed: OpenAI API key, Home Assistant, Python.
   - Steps:
     - Write a Python script to interact with OpenAI's API.
     - Integrate with Home Assistant to enable JARVIS to control home devices.

- Set up automations for spoken commands through Alexa or Siri.
- Configure access levels to prevent unintended system changes.

Phase 4: Long-term Resilience and Future-proofing

1. MacBook Air as Secondary Node
   - Objective: Set up the MacBook Air as a secondary node for redundancy.
   - Tools Needed: Proxmox (if feasible), Docker for lightweight containers.
   - Steps:
     - Test compatibility of Proxmox on the M2 chip.
     - Use Syncthing for data sync between the MacBook Air and iMac's Proxmox VMs.
     - Set up Tailscale for remote management of the MacBook Air.

2. Build Out Redundancies and Testing
   - Objective: Create test scenarios for resilience against failures.
   - Steps:
     - Test failover scenarios and battery backups.
     - Conduct regular security audits for VPN and WireGuard.