PurrBrews Project Report

Project Summary

This report provides daily updates on the status and progress of the project, including tasks completed, challenges encountered, and next steps. The goal is to ensure all stakeholders are informed about the project's development and any potential roadblocks.

Daily Summary

Day 01

Here is the comprehensive documentation for the **Purrbrews.cc** infrastructure project as of December 31, 2025. This covers the architecture, node configuration, standard operating procedures, and the current status of each component.

☕ Purrbrews Infrastructure Documentation

Date: December 31, 2025

Project: Purrbrews.cc Eco-system

Status: Phase 1 Successfully Concluded (Core Foundation Established)

Architecture: Bare-Metal Docker Implementation (Storage and Compute Resources Decoupled)

1. Executive Summary

The server infrastructure has been established as a professional-grade home environment emphasizing high availability, security, and scalability. The architecture is logically partitioned into specialized nodes: "Mill" (dedicated to Network and DNS services), "Barista" (serving as the Compute and Administrative core), and "Bean" (operating as the Voice Satellite).Key Achievements

- **Centralized Infrastructure Repository:** The `purrbrews-infra` repository was created, enforcing strict separation between operational code (`compose/`) and sensitive configuration data (`.env.example`).

- **Network Core Deployment (Mill):** Pi-hole v6 was successfully deployed on the `mill` node (192.168.0.11), functioning as the primary DNS resolver and advertisement blocker.
- **Compute Core Implementation (Barista):** Portainer was successfully deployed on the `barista` node (192.168.0.10) to facilitate centralized container management.
- **Security Architecture:** A robust security framework was implemented, utilizing "Deploy Keys" for authenticated GitHub access and a "Symlink Bridge" methodology for managing secret files.
- **Satellite Node Preparation (Bean):** Compatibility challenges related to the underlying OS on the Raspberry Pi Zero 2 W were successfully resolved, enabling the installation of the Docker engine.

2. Ecosystem Inventory

1. Server and Infrastructure Nodes (The Backend)

Hostname	IP Address	Hardware	Role	Operating System / Status
Barista	`192.168.0.10`	**HP Pavilion x360**	**Primary Compute Server**	Debian 12 (Bookworm) / 🟢 Operational
Mill	`192.168.0.11`	**Raspberry Pi 3B+**	**Network and DNS Services**	Debian 12 (Bookworm) / 🟢 Operational
Bean	`192.168.0.12`	**Pi Zero 2 W**	**Satellite Monitoring Node**	Raspberry Pi OS / 🟡 Suspended

2. Client Devices (The Customers)

Hostname	Hardware	Role	Security Classification (Trust Level)
Roastery	Lenovo Legion 5 Pro	**Primary Engineering Workstation**	High (Administrative Privileges)
Espresso	Samsung S24	**Primary Mobile Communication Device**	High
Cortado	iPhone 12 Mini	**Auxiliary/Office Mobile Device**	Medium
Mocha	Realme Pad 2	**Media Consumption Terminal**	Medium
Decaf	Dell Generic Laptop	**Corporate/External Access Device**	Low (Zero Trust/Guest Access)

---

3. Standard Operating Procedures (SOPs)

The "Pet vs. Cattle" Storage Standard

All computing nodes shall strictly adhere to the separation of persistent data from the operating system files.

- **Code Repository:** `/opt/purrbrews` (Git-Controlled, designated as "Cattle")
- **Persistent Data:** `/media/storage/data` (Routinely backed up, designated as "Pet")
- **Secrets Vault:** `/media/storage/secrets` (Secured storage, designated as "Pet")

Secret Management Methodology (Via Symbolic Linkage)

Sensitive configuration data (secrets) shall under no circumstances be committed to the code repository. Access to the secrets vault will be facilitated through a symbolic link.

1. Creation of the Vault File (On Host System):

```
sudo nano /media/storage/secrets/global.env
# Populate this file with actual configuration values (e.g.,
PIHOLE_PASSWORD=...)
```

2. Establishment of Secure Permissions:

```
sudo chmod 600 /media/storage/secrets/global.env
```

3. Linking to the Repository Directory:

```
cd /opt/purrbrews
rm -f .env
ln -s /media/storage/secrets/global.env .env
```

4. Node Configuration Details

🟢 *Node 1: Mill (DNS/Network)*

- **OS:** Debian 12 (Bookworm) - Headless
- **Role:** Network Shield
- **Status:** Online

Network Config (/etc/network/interfaces):

```
auto wlan0
iface wlan0 inet static
    address 192.168.0.11
    netmask 255.255.255.0
    gateway 192.168.0.1
    # Points to localhost (itself) then Cloudflare
    dns-nameservers 127.0.0.1 1.1.1.1
```

Deployment:

```
cd /opt/purrbrews
docker compose --env-file .env -f mill/compose/core.yml up
-d --force-recreate
```

🟢 *Node 2: Barista (Compute/Admin)*

- **OS:** Debian 12 (Bookworm)
- **Role:** Heavy Lifting / Admin
- **Status:** Online

Network Config (/etc/network/interfaces):

```
auto wlan0
iface wlan0 inet static
    address 192.168.0.10
    netmask 255.255.255.0
    gateway 192.168.0.1
    # Points to Mill (11), then Cloudflare (Backup)
    dns-nameservers 192.168.0.11 1.1.1.1
```

Deployment:

```
cd /opt/purrbrews
docker compose --env-file .env -f barista/compose/core.yml up -d
```

🟡 *Node 3: Bean (Satellite)*

- **OS:** Raspberry Pi OS (Trixie/Unstable)
- **Hardware:** Pi Zero 2 W
- **Status:** Paused (Waiting for Hardware)

Critical Fix: Manual Docker Install

Because the OS identified as "Trixie" (Debian 13), the standard script failed. We forced the "Bookworm" (Debian 12) repository.

Critical Fix: Manual Docker Installation Required

Due to the operating system identifying as "Trixie" (Debian 13), the standard installation script failed. Consequently, the "Bookworm" (Debian 12) repository was manually specified.

Installation Commands Used:

```
# 1. Clean previous attempts
sudo rm /etc/apt/sources.list.d/docker.list

# 2. Add Key manually
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL
[https://download.docker.com/linux/raspbian/gpg](https://download.docker.com/linux/raspbian/gpg) -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# 3. Add Repo (Forcing 'bookworm')
echo \
  "deb [arch=armhf signed-by=/etc/apt/keyrings/docker.asc]
[https://download.docker.com/linux/raspbian](https://download.docker.com/linux/raspbian) \
  bookworm stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# 4. Install
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io
docker-buildx-plugin docker-compose-plugin
```

5. Security Protocols

Git Access (Deploy Keys)

All nodes utilize **Deploy Keys** (Read-Only) for code retrieval, thereby ensuring system isolation.

SSH Config (~/.ssh/config):

```
Host github.com
  IdentityFile ~/.ssh/id_deploy
  StrictHostKeyChecking accept-new
```

Zero Trust Access (Future)

The selection for providing external access has been determined as **Cloudflare Tunnel**.

- **Objective:** To securely expose internal services, such as `portainer.purrbrews.cc`, without necessitating the opening of router ports.
- **Deployment:** The requisite agent is scheduled for deployment on the server `'mill'`