Host Node Setup and Deployment Guide

# 1. Overview

This document outlines the complete flow and logic behind the Host Node setup process for Raspberry Pi 4B systems used in the distributed 3D printing environment. The setup is handled entirely locally using a script named `host\_setup.sh` located in `/home/makerspace/host\_setup/`.

# 2. Setup Script Flow

The script prompts the user to specify whether this is a New or Replacement Host Node:

* New Host Node Flow:
* • User is prompted for Host Node ID (e.g., Node3).
* • User enters the number of printers (1 or 2).
* • For each printer, the user enters a unique Printer ID (e.g., Printer3, Printer4).
* • A static IP address is requested and applied using `nmcli`.
* • Hostname is set and updated system-wide.
* • Template files are patched and written based on user inputs (e.g., service files, rules, config.yaml).
* • Udev rules are copied and triggered.
* • Setup data is logged in `setup\_result.json` for future replacements.
* Replacement Host Node Flow:
* • Pulls previous configuration data from `setup\_result.json`.
* • Re-applies static IP, hostname, and symlinks using the saved configuration.

# 3. Template Substitution

Template files reside in `/home/makerspace/host\_setup/node\_templates/` and include:

* • Systemd services: octoprint.service.template, mjpg-streamer.service.template
* • Autoconnect services and paths
* • OctoPrint config.yaml templates
* • Udev rules: 99-3dprinters.rules.template, 99-webcam.rules.template

Variables like `{{PRINTER\_ID}}`, `{{WEBCAM\_ID}}`, `\_\_WEBCAM\_URL\_\_`, and `\_\_SNAPSHOT\_URL\_\_` are patched dynamically.

# 4. Services Enabled

* Based on printer count, the following services are enabled dynamically:
* • OctoPrint instances (octoprint.service, octoprint2.service)
* • MJPG Streamer instances (mjpg-streamer.service, mjpg-streamer2.service)
* • Autoconnect services and paths for each printer

# 5. Logging and State Tracking

* • Logs are saved to `/home/makerspace/host\_setup/logs/setup.log`.
* • Setup metadata is stored in `setup\_result.json`, used for replacement mode and tracking.

# 6. Udev Rule Injection

* • 99-3dprinters.rules and 99-webcam.rules are patched to reflect chosen Printer IDs and WebCam IDs.
* • Custom ID\_PATH or USB KERNELS values are retained and only symlink names are changed.

# 7. Dry Run Checklist

Use this guide to verify setup logic without executing any destructive or system-altering operations.

1. 1. Run the script with `bash host\_setup.sh`.
2. 2. Choose 'New Node' and supply test values (e.g., NodeTest, Printer99).
3. 3. When prompted for static IP, use a non-conflicting, unused IP address (e.g., 192.168.1.250).
4. 4. Check that:

* - Hostname changes in `/etc/hostname` and `/etc/hosts`
* - Correct service files are generated in `/etc/systemd/system/`
* - Correct YAML and udev rules are patched and placed

1. 5. Ensure `setup\_result.json` reflects the inputs correctly.
2. 6. Reboot the Pi manually if needed, or inspect the `/var/log/syslog` for any issues.