

Cloud-Init Fix - Deep Analysis

Problem History

We've tried to fix cloud-init boot delays multiple times:

Previous Attempts (FAILED)

1. ****cloud-init-unblock.service**** - Tried to "unblock" cloud-init by forcing it to complete
 - Problem: Still allows cloud-init to run, just forces completion
 - Doesn't prevent delays
 - Cloud-init can still block boot
2. ****cloud-init-force-unblock.service**** - More aggressive unblocking
 - Same problem: Tries to work WITH cloud-init instead of disabling it

Root Cause

- Cloud-init is enabled by default in some Raspberry Pi OS images
- It waits for network configuration or other services
- If it waits for something that never comes, boot hangs
- Our custom build doesn't need cloud-init at all

Current Solution (06-disable-cloud-init.service)

Approach: DISABLE, not unblock

- Stop all cloud-init services
- Disable all cloud-init services
- MASK all cloud-init services (prevents ANY start)
- MASK cloud-init.target (prevents target from blocking)

Service Configuration

Dependencies:

```
DefaultDependencies=no          # Prevents systemd from adding unwanted dependencies
After=local-fs.target           # Runs after filesystem is ready
Before=multi-user.target         # Runs before full boot
Before=cloud-init.target        # Runs before cloud-init target
Before=cloud-init.service       # Runs before cloud-init service
Conflicts=cloud-init.target     # Prevents cloud-init.target from running
Conflicts=cloud-init.service    # Prevents cloud-init.service from running
```

Actions:

1. Stop all cloud-init services
2. Disable all cloud-init services
3. Mask all cloud-init services (systemd can't start them)
4. Mask cloud-init.target (prevents target from blocking boot)

Build Script Integration

Also disables cloud-init during image build:

- Stops cloud-init services
- Disables cloud-init services
- Masks cloud-init services
- Ensures cloud-init won't run on first boot

Why This Should Work

1. ****Early execution****: Runs after local-fs.target, before cloud-init
2. ****Conflicts****: Systemd won't start cloud-init if our service conflicts with it
3. ****Masking****: Even if something tries to start cloud-init, mask prevents it
4. ****Target masking****: cloud-init.target is also masked, preventing boot blocking
5. ****DefaultDependencies=no****: Prevents systemd from adding dependencies that might delay our service

Potential Issues

1. ****Service runs too late****: If cloud-init starts before our service, it might still block
 - Mitigation: `Before=cloud-init.target` and `Conflicts=` should prevent this
2. ****Cloud-init required by something else****: If another service requires cloud-init
 - Mitigation: Masking should prevent cloud-init from starting even if required
3. ****Different service names****: If cloud-init has different names in different OS versions
 - Mitigation: We mask all known cloud-init service names
4. ****Systemd ignores mask****: In rare cases, systemd might ignore mask if service is critical
 - Mitigation: `Conflicts=` should still prevent it

Testing

After applying fix:

1. Boot Pi
2. Check if boot stops at cloud-init (should NOT)
3. Verify services: `systemctl status cloud-init` (should show "masked")
4. Check logs: `journalctl -u 06-disable-cloud-init.service`
5. Verify target: `systemctl status cloud-init.target` (should show "masked")

Files

- `moode-source/lib/systemd/system/06-disable-cloud-init.service` - Service file
- `imgbuild/moode-cfg/stage3_03-ghettoblaster-custom_00-run-chroot.sh` - Build script
- `docs/CLOUD_INIT_FIX.md` - Basic documentation
- `docs/CLOUD_INIT_ANALYSIS.md` - This deep analysis

Status

- Service created with robust dependencies
- Build script updated
- SD card updated
- Comprehensive documentation

This is the most robust fix possible. If this doesn't work, the problem is likely something else entirely.