

```

# GOES-19 Live Full-Disk Web UI (RaspiNOAA + SatDump)
## Purpose
This document describes, **step by step**, how to build a reliable, live-updating
GOES-19 Full Disk web interface on a Raspberry Pi running RaspiNOAA with SatDump.
Nothing here is optional. Nothing is implied. Every step exists because something
breaks if it is skipped.

This system:
- Receives GOES-19 HRIT via SatDump
- Publishes *only Full Disk ABI products*
- Mirrors the newest Full Disk directory to a web root
- Pushes **real-time browser refresh** using Server-Sent Events (SSE)
- Does **not** interfere with RaspiNOAA's existing LEO tracking UI
---

## Assumptions (Non-Negotiable)
- Raspberry Pi OS (Bookworm)
- RaspiNOAA already installed and running
- SatDump built and working
- RTL-SDR working at 1.6941 GHz
- You are comfortable with `systemctl`, `nginx`, and shell scripts
---

## Filesystem Layout (Authoritative)
SatDump output **must** look like this:
```
/home/pi/sat/GOES-19/IMAGES/GOES-19/Full Disk/
 YYYY-MM-DD_HH-MM-SS/
 G19_*_YYYYMMDDTHHMMSSZ.png
 product.cbor
```

If this layout is wrong, nothing downstream will work.
---

## SatDump Service (GOES-19 HRIT)
### Systemd service
```
[Service]
ExecStart **must** include:
--fill_missing
--output-directory /home/pi/sat

Restart and verify:
```
```
systemctl restart satdump-goes19
systemctl is-active satdump-goes19
```

### USB Stability (Critical)
SatDump **will silently stall** unless usbfs memory is disabled:
```
echo 0 | sudo tee /sys/module/usbcore/parameters/usbfs_memory_mb
```

Make this persistent via `/etc/rc.local` or a systemd unit.
---

## Web Publish Script
### Script Location
```
/usr/local/bin/update_goes_fd_web.sh
```

```

```

#### Script (Minimal, Correct)
```
#!/bin/bash
set -euo pipefail
SATDUMP_FD_ROOT="/home/pi/sat/GOES-19/IMAGES/GOES-19/Full Disk"
WEB_ROOT="/var/www/goes"
LATEST_DIR=$(ls -1 "$SATDUMP_FD_ROOT" | sort -r | head -n 1)"
SRC="$SATDUMP_FD_ROOT/$LATEST_DIR"
mkdir -p "$WEB_ROOT/current"
rm -f "$WEB_ROOT/current/*"
cp -f "$SRC"/* "$WEB_ROOT/current/"
cat > "$WEB_ROOT/meta.json" <<EOF
{
 "timestamp_dir": "$LATEST_DIR",
 "updated_utc": "$(date -u +%FT%TZ)"
}
EOF
touch "$WEB_ROOT/.trigger"
```
Permissions:
```
chmod +x /usr/local/bin/update_goes_fd_web.sh
```
```
Systemd Timer
Service
```
/etc/systemd/system/update-goes-fd-web.service
```
```
[Service]
Type=oneshot
ExecStart=/usr/local/bin/update_goes_fd_web.sh
```
Timer
```
/etc/systemd/system/update-goes-fd-web.timer
```
```
[Timer]
OnBootSec=60
OnUnitActiveSec=60
```
Enable:
```
systemctl enable --now update-goes-fd-web.timer
```
```
## nginx (DO NOT BREAK RaspiNOAA)
### Dedicated GOES site
```
/etc/nginx/sites-available/goes
```
```
server {
listen 8080;
root /var/www/goes;

```

```
index index.html;
location /goes/current/ {
alias /var/www/goes/current/;
autoindex on;
add_header Cache-Control "no-store";
}
location /goes/events {
default_type text/event-stream;
add_header Cache-Control no-cache;
add_header Connection keep-alive;
chunked_transfer_encoding on;
}
}

Enable:
```
In -s /etc/nginx/sites-available/goes /etc/nginx/sites-enabled/goes
nginx -t && systemctl reload nginx
```

```
## Server■Sent Events (SSE)
A lightweight Python watcher monitors `trigger` and emits updates.
```
python3 sse_watch.py
```

This **must** be running or the browser will not auto-refresh.
```
Browser UI Behavior
- Page loads newest image
- Dropdown lists all images in `/current`
- New image arrival:
- Script updates `trigger`
- SSE emits event
- Browser reloads image and steals focus
- Footer shows UTC timestamp parsed from filename
```
```
Known Failure Modes (Read This)
Symptom	Cause	Fix
Images frozen	usbfs buffer	set usbfs_memory_mb=0
JSON updates but images don't	Wrong SATDUMP_FD_ROOT	fix path
SSE connects, no updates	`trigger` not touched	fix script
RaspiNOAA broken	You edited port 80	undo it
```
## Philosophy
This system exists because static pages are useless for live satellites.
If it stops updating, something **upstream** is broken.
Do not debug the web UI first.
```
License
Public domain. Share it. Improve it. Do not simplify it.
```