

Fine tuning 2026.10.26

CiCADA tuning, round 2

Based on the server info collected at 19:33 KST and the analyzer run at 19:38 on 26 Oct 2026.

Verdict

The server is idle and healthy: load 0.02, no CPU, memory or I/O pressure, PHP-FPM peaked at 3 of 7 workers per pool, OPcache hit rate 99.93%, InnoDB miss rate 0.16%, no slow queries and no nginx errors. More throughput tuning would not be measurable. What is left is one real latency item (cloud's data sits on a USB hard disk), a few configuration mistakes worth fixing, and housekeeping.

1. Heavy Nextcloud jobs run in the daytime

`maintenance_window_start` is 1, which means 01:00 to 05:00 UTC, or 10:00 to 14:00 KST. Move it to 02:00 to 06:00 KST (UTC 17):

```
for d in cloud.onni.me air.meson.in; do
    sudo -u www-data php8.4 /var/www/$d/occ config:system:set maintenance_window_start --
type=integer --value=17
done
```

The Monday backups (03:00 to 03:30) overlap with it once a week. Use 19 instead for 04:00 to 08:00 KST if you prefer.

2. Two cron lines probably fail every 5 minutes

`previewgenerator` is disabled on both instances, but the `www-data` crontab still calls `occ preview:pre-generate` every 5 minutes. Test it:

```
sudo -u www-data php8.4 /var/www/cloud.onni.me/occ preview:pre-generate
```

If it says the command is not defined, either run `occ app:enable previewgenerator` on that instance, or delete the two lines with `sudo crontab -u www-data -e`. Each failing run boots the whole Nextcloud framework for nothing, and the output goes to `/dev/null` so you never see it.

3. cloud.onni.me data lives on the slow disk

`iostat` (average since boot) shows `sd`, the USB hard disk mounted at `/home/web-data`, at 38 ms read and 31 ms write latency, against 0.5 to 0.9 ms for your SSDs. The cloud instance keeps its data and temp directory there, while its database is on the SSD. Few reads were sampled, so treat the figure as indicative, but it is the only place a user could feel slowness.

a) Monitor the disk. SMART output for `sd` was blank, so that disk is not being checked. USB bridges need SAT mode:

```
sudo smartctl -d sat -H -A /dev/sdc
```

b) Stop atime writes. In `/etc/fstab`, add `noatime` to the options of the `/home/web-data` line (keep your UUID and the other fields, for example `defaults,noatime,commit=60`), then:

```
sudo mount -o remount,noatime /home/web-data
findmnt /home/web-data
```

c) Larger read-ahead for the rotational disks (128 KB today) helps large file downloads:

```
sudo tee /etc/udev/rules.d/60-hdd-readahead.rules >/dev/null <<'EOF'
ACTION=="add|change", KERNEL=="sd[a-z]", ATTR{queue/rotational}=="1",
ATTR{queue/read_ahead_kb}="1024"
EOF
```

```
sudo udevadm control --reload && sudo udevadm trigger --subsystem-match=block --action=change
cat /sys/block/sdc/queue/read_ahead_kb
```

d) Keep cloud's temp files off that disk (air already uses the SSD):

```
sudo install -d -o www-data -g www-data -m 750 /home/www-data/.tempdir-cloud
sudo -u www-data php8.4 /var/www/cloud.onni.me/occ config:system:set tempdirectory --
value=/home/www-data/.tempdir-cloud
```

e) Measure before deciding anything bigger. If the previews are small, they could later move to the SSD:

```
sudo du -sh /home/web-data/cloud.onni.me/appdata_*/preview
```

4. Kernel settings

```
sudo tee /etc/sysctl.d/99-local.conf >/dev/null <<'EOF'
vm.swappiness = 10
vm.vfs_cache_pressure = 50
vm.overcommit_memory = 1
net.ipv4.tcp_congestion_control = bbr
net.ipv4.tcp_slow_start_after_idle = 0
net.core.rmem_max = 2500000
net.core.wmem_max = 2500000
EOF
echo tcp_bbr | sudo tee /etc/modules-load.d/bbr.conf
sudo modprobe tcp_bbr && sudo sysctl --system
sysctl net.ipv4.tcp_congestion_control vm.swappiness
```

Why each line:

- `vm.swappiness = 10` keeps the database and workers in RAM (54 MB is swapped today).
- `vm.vfs_cache_pressure = 50` keeps file metadata cached longer, which matters for the hard disks.
- `vm.overcommit_memory = 1` is what Redis asks for.
- BBR and no slow start after idle help remote clients over lossy links. The effect is modest and workload dependent; revert by deleting the line and running `sudo sysctl --system`.
- The larger UDP buffers keep HTTP/3 from dropping bursts.

5. MariaDB: remove one line

`log_bin` is off, so `binlog_format = ROW` does nothing and logs a warning at every start ('You need to use `--log-bin` to make `--binlog-format` work'):

```
sudo sed -i 's/^binlog_format/#binlog_format/' /etc/mysql/mariadb.conf.d/50-server.cnf
```

It takes effect at the next restart, so there is no need to restart just for this. Optional: reclaim the 45 MB of free space in cloud's activity table with `OPTIMIZE TABLE oc_activity;` in the cloud database.

Everything else matches what the numbers need: the buffer pool is 65% free with a 99.8% hit rate, and at most 4 of 60 connections were ever used.

6. PHP: two tiny trims

Do these only when you restart PHP-FPM for another reason:

- `/etc/php/8.4/mods-available/opcache.ini`: `opcache.jit_buffer_size=8M` (31.1 of 32 MB is unused)
- `/etc/php/8.4/mods-available/apcu.ini`: `apc.shm_size=32M` (1.8 of 128 MB is used)

Then run `sudo systemctl restart php8.4-fpm`. Keep the pool sizes (peak 3 of 7) and OPcache at 192 MB (72% used).

7. Redis eviction policy

Change `maxmemory-policy allkeys-lru` to `volatile-lru` in `/etc/redis/redis.conf`. Nextcloud locks and cache entries carry expiry times, so they stay evictable, and nothing without an expiry can be dropped by accident. Redis holds 1.9 MB of 256 MB, so this is protection, not tuning:

```
sudo sed -i 's/^maxmemory-policy .*/maxmemory-policy volatile-lru/' /etc/redis/redis.conf
sudo systemctl restart redis-server
```

Restart at a quiet moment, because file locks are dropped.

8. Docker

The torrent containers have no resource limits and unbounded logs. In `/home/www-data/bit.onni.me/docker-compose.yaml`, add this to each transmission service:

```
mem_limit: 512m
cpus: 1.5
cpu_shares: 256
logging:
  driver: json-file
  options:
    max-size: 10m
    max-file: "3"
```

Add only the `logging:` block to the `gluetun` service. Then apply:

```
cd /home/www-data/bit.onni.me && docker compose up -d
```

Compose recreates only the services that changed. Per-service logging avoids restarting `dockerd`, and a daemon restart would stop every container because live restore is off. `cpu_shares: 256` (default 1024) lets Nextcloud win under CPU contention. Optional: `docker image prune` frees 1.36 GB of unused images.

9. Journal size

The journal is 1.3 GB and `journald` holds 105 MB of RAM:

```
sudo mkdir -p /etc/systemd/journald.conf.d
sudo tee /etc/systemd/journald.conf.d/size.conf >/dev/null <<'EOF'
[Journal]
SystemMaxUse=200M
MaxRetentionSec=1month
EOF
sudo systemctl restart systemd-journald
sudo journalctl --vacuum-size=200M
```

10. Services you may not need

- **NFS and rpcbind** listen on every interface (the analyzer warning). UFW blocks them, but check whether anything is exported first with `sudo exportfs -v`. If the list is empty: `sudo`

`systemctl disable --now nfs-server rpc-statd rpcbind.socket rpcbind`. If you do use NFS, bind it to the Tailscale address instead (`host=` under `[nfsd]` in `/etc/nfs.conf`).

- **multipathd** (27 MB) is for multi-path SAN storage. If `sudo multipath -ll` prints nothing: `sudo systemctl disable --now multipathd.service multipathd.socket`.
- **snap lxd**: two revisions are mounted. Run `sudo snap remove lxd` if you do not use LXD.
- **Nextcloud apps**: with one user per instance, `firstrunwizard`, `recommendations`, `weather_status`, `survey_client`, `support` and `nextcloud_announcements` are loaded on every request. Run `occ app:disable <name>` one at a time and check the admin pages after each.

11. Watch list (no change now)

- **Nextcloud log**. `loglevel` is 2, yet about 4,700 of the last 5,000 lines on cloud are level 0 or 1, and 166 are errors (the setup check counts 43 errors since 19 Oct). Find the source:

```
cd /var/www/cloud.onni.me
LOG="$(sudo -u www-data php8.4 occ config:system:get datadirectory)/nextcloud.log"
sudo jq -r 'select(.level>=3)|[.app,.message]|@tsv' "$LOG" | cut -c1-150 | sort | uniq -c |
sort -rn | head
sudo jq -r 'select(.level<2)|.app' "$LOG" | sort | uniq -c | sort -rn | head
```

- **oc_job_runs** has about 190k rows on both instances (15 MB each). Count it again next week and report back if it grows quickly.
- **eno1** reports 689,191 received packets dropped (1.3%) with 0 missed. This is probably broadcast or unknown-protocol noise from the ISP segment. Check with `sudo ethtool -S eno1 | grep -iE 'drop|err|fifo|miss'`.
- **Backups**. The database dump is weekly (Monday 03:00). For a 230 MB medical database, daily is nearly free: change `00 03 * * 1` to `00 03 * * *` for `backup_mariadb.sh` in root's crontab.
- **fail2ban** only has the `sshd` jail, while the `nginx` logs show scanner bursts (`/.git`, `/.env`). A jail on the 'access forbidden by rule' lines would cut the noise.

Leave as is

PHP-FPM pool sizes and memory limits, OPcache at 192M, the InnoDB buffer pool (512M), the temp-table and full-join figures (they come from schema checks), `nginx` (HTTP/3, buffers, gzip, headers, zero errors), the CPU governor (`schedutil`) and the I/O schedulers.

Revision #1

Created 2026-10-26 13:04:47 UTC by meson

Updated 2026-10-26 13:05:25 UTC by meson